Global Information Assurance Certification Paper

SANS Darling Harbour (February 2001) Firewalls, Perimeter Protection, and VPNs GCFW Practical Assignment v1.5d

Jason Haar

April 2001

Contents

Assignment 1 – Security Architecture (25 Points)

Define a security architecture for GIAC Enterprises, a growing Internet startup that expects to earn $200 million per year in online sales of fortune cookie sayings, and which has just completed a merger/acquisition. Your architecture must specify filtering routers, firewalls, VPNs to partners, secure remote access, and internal firewalls. Be explicit about the brand and version of each perimeter defence component. Produce a diagram or set of diagrams with explanatory text that define how to use perimeter technologies to implement your security architecture. You must consider and define access for:

- *Customers (the companies that purchase bulk online fortunes);*
- *Suppliers (the authors of fortune cookie sayings that connect to supply fortunes);*
- *Partners (the international partners that translate and resell fortunes).*

1.1 Assumptions

In order to define a security architecture for GIAC Enterprises, I have made the following assumptions:

- As a successful Internet company, GIAC Enterprises depends on a reliable and high-performance network and computer infrastructure and therefore has a generous security architecture budget.
- All customers have individual accounts with GIAC Enterprises' Web servers.
- Staff perform a range of business functions, from sales, marketing and editorial, through to finance and software development. Staff from all business areas require remote access (from satellite sites, home, etc) to the GIAC Enterprises LAN.
- GIAC Enterprises wants the security architecture to protect against both incoming and outgoing security threats, e.g. viruses, trojans, data loss.
- Access requirements are as follows:
  - GIAC staff need access to the Web and to send and receive Internet email;
  - Customers need to access GIAC Enterprises' web servers; and
  - Suppliers/Partners need to access GIAC Enterprises' database servers.

1.2 Overall design

The main principle I have employed in designing the GIAC Enterprises security architecture is to achieve a balance between security and usability. The architecture is therefore split into 8 logical units corresponding to 8 physically separate networks joined by firewalls. This

- limits the reach of any compromise to the network on which it occurred;
- simplifies the firewall design; and
- reduces the chance of misconfiguration errors.

A mixture of vendors and equipment is used to limit any compromise. The entire network could be invaded if a single vendor is chosen for all firewall components, and a bug is found with that vendors products.

The networks reachable by the public use assigned Internet addresses (as expected), whereas the other GIAC Enterprises networks use private address spaces (192.168.0.0 and 10.0.0.0). This use of private address space again limits the depth that an attacker can reach into the GIAC Enterprises network (being unable to route packets from the Internet to private address spaces), even with the total collapse of the firewall systems being put in place.

The 8 networks are:

1. Perimeter network. Area where Internet interfaces with GIAC Enterprises.
2. Service subnet. GIAC Enterprises' email and DNS servers.
3. WebNet subnet. GIAC Enterprises' web servers.
4. SQLNet subnet. GIAC Enterprises' database servers (containing fortunes).
5. RemoteNet subnet. Area where all remote devices interface with the LAN.
6. Partner/Supplier VPN network. Network via which partners and suppliers access the SQLNet network.
7. IDS subnet. Intrusion detection area.
8. LAN.

Networks 1 through 6 above are here referred to as the 'Internet subnets'.

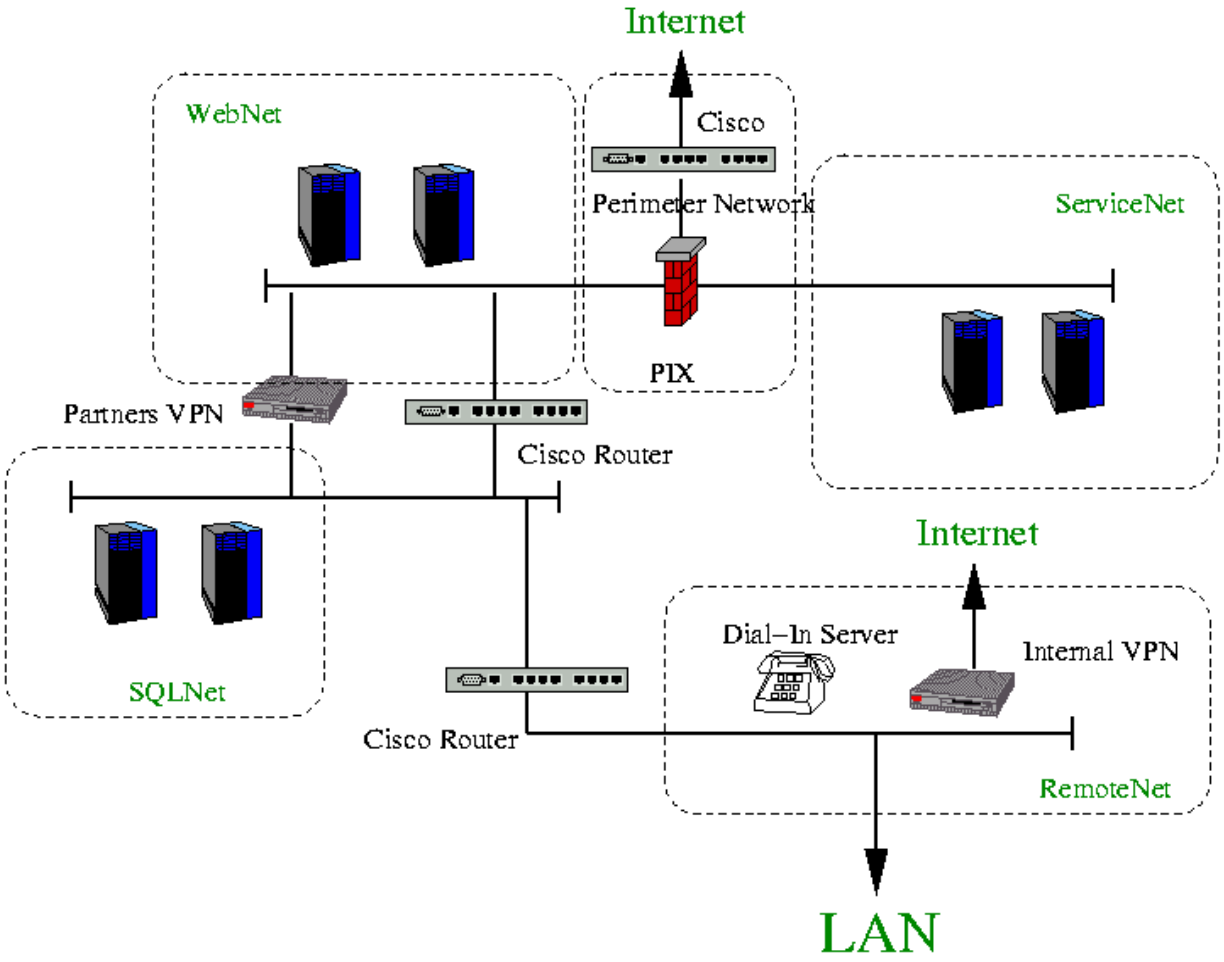


Figure 1: Network Architecture for GIAC Enterprises. For clarity, the IDS network is not shown here.

1.3 Characteristics common to all networks

1.3.1 Switches

Each network consists of an Extreme 24 switch, populated only with systems relevant to that subnet. Security holes are found even in switches, so here they are all installed as 'dumb' devices, with no management capabilities (including no IP address). This should not be a problem given the few devices that hang off each switch.

One port on each switch is allocated as the monitor port. All the traffic on that switch's ports is 'reflected' onto the monitor port so that an IDS system can 'sniff' all traffic on that switch's network.

1.3.2 Security Consoles

Each network contains a host that is referred to as the 'Subnet Security Console' (SSC). This host runs typical monitoring services such as syslog, arpwatch, snmptrapd, TFTP and SSH daemons for their network. The Security Console on the LAN is called the Central Security Console (CSC) and is used as the 'central repository' and main management interface.

For security reasons, the TFTP daemon is normally deactivated, but is turned on whenever reconfiguration of routers and firewalls is required. The arpwatch daemon continually monitors ARP packets on the subnet, and Emails alerts out whenever it notices a change in the ARP tables. As these are very static networks, any event as major as a change in MAC address needs to be treated seriously. It could be the sign of the operation of a tool such as ettercap (allows the 'sniffing' of traffic between two hosts on even a switched network by faking MAC addresses and proxying the results). Only SSH accepts connections from the LAN (only from the CSC, to be precise). The other daemons are firewalled such that they only accept packets from the subnet they are on. Ipchains is used to ensure no other traffic is accepted. The syslog files are rotated hourly and along with other application logs, rsync'ed via SSH back onto the CSC, that is pulled from the CSC. Local swatch processes monitor each SSC's syslog files and sends off alerts via email when appropriate. The SSCs are left alone to do their task — the CSC is the primary interface by which the IS Security staff monitor the external networks.

1.3.3 Access

All security components (routers and firewalls as well as hosts) are only accessible via SSH or their console (for when they don't support SSH). If such devices are present, console cables can be run through the building CAT-5 cabling system for some distance back into a switch box hanging off the back of the CSC.

1.3.4 Logging

All logging throughout the network is via syslog. Syslog writes records locally (where possible), as well as back onto the SSC (e.g. Web server writes to **/var/log/messages** and to **@ssc_host**).

All hosts and firewalls/routers have NTP enabled so that their clocks (and therefore the log files they generate) are synchronized.

SNMP is installed on all routers and firewalls, primarily because it is the only supported way to obtain traffic flow statistics – which GIAC Enterprises wants in order to monitor the quality of their services. To minimize the security risk, the SNMP agents are set to Read-Only, have good quality community strings, and only allow the CSC to access them. SNMP is also used to send traps (which show up in syslog) back to that subnet SSC. There are many reasons not to use SNMP — almost all of them security-related — but the risk is here deemed acceptable due to the business need.

MRTG is used on the CSC to graph network characteristics, bandwidth utilization, packet loss etc., as well as third-party modules, which also allow graphing of server characteristics.

Other logging cannot be forgotten. Web error logs especially must be monitored as religiously as the firewall 'alerts' as they can be indicative of an ongoing assault. All application logs should be automatically copied off by the CSC for backup and analysis.

1.3.5 Packet filtering policy

By default, all security components are configured to drop packets rather than reject them. TCP Resets and ICMP unreachables are only sent by host machines when it has been decided that not doing so would be detrimental to valid end-users. In all other cases they are dropped. As a consequence, many port scans will be slowed down considerably and some may fail totally. This doesn't stop determined crackers, but may discourage 'script-kiddies'.

The CSC is used primarily for batch-mode analysis of each network's activities and real-time analysis of

the IDS network. It contains the syslog files generated by every Security Console, and also uses swatch to set off alerts due to IDS events etc. The RADIUS accounting information resides here, and all this detail is made available via the secure Web interface. ACID is used as the front-end to the IDS SQL database (i.e. the IDS is Snort). ACID is an excellent interface by which to interrogate a Snort database, and can be used to view long-term attacks and patterns easily missed by less capable systems. Any syslog messages generated by routers and firewalls is pushed through logsnorter to subsume them into the Snort MySQL database – effectively turning those devices into 'mini' IDS systems too.

1.3.6 Server hosts

Linux is chosen as the OS on which to base the GIAC servers. It has a good reputation for being a secure OS (when implemented by competent people), and I personally have a lot of experience with it. Choice of OS should always take into account staff knowledge.

Software is only installed if it is needed. All setuid programs are scrutinized as to whether they are needed. For example, the **/bin/passwd** program is normally setuid root so that users can use it to change their passwords. As there will never be 'normal users' besides root, it doesn't need to be setuid anymore!

Denial of Service (DoS) by resource exhaustion is minimized by ensuring all daemons are started with their ulimits constrained as tightly as possible. DoS attacks cannot be completely stopped, but such measures can severely limit their impact. Instead of a whole system crashing, having tight resource limits can mean that a successful DoS is limited to just the application service under attack.

Where possible, all services are run as non-root accounts within chroot'ed environments. Chroot is a much-underused utility that can severely limit a successful system break-in, even making the break-in effectively a waste of time. Always try to run any application as non-root. Create separate accounts for each application, and run them with a 077 umask. That way compromise of one won't cause any harm to others. This layered approach illustrates 'defence in depth'.

All servers use ipchains to limit what packets are allowed in and out of them. Most of these filter rules duplicate those on the perimeter router and PIX, and are chiefly there as fail-safe measures.

When set up correctly, server hosts run pretty much without intervention. Cronjobs roll the logfiles over, move logs into directories to be downloaded by the CSC back onto the LAN, and send alerts on unusual events. Any other connection will trigger alerts on the IDS systems.

The logs need to be monitored for unusual activity – especially the Web server error logs, as they are the most probable entry point for any attacker. Web reporting tools such as Analog can not only generate Web analysis reports, but also can report on error conditions such as failed HTTP login attempts.

Below are the core network applications installed on the Linux boxes. Not all are implemented on every box:

Application	**Non-root Account**	**Chroot Jail**
Qmail (sendmail replacement)	Yes	No (needs to be used by everything)
ssmtp (sendmail replacement for chrooted areas)	Yes	Yes

MySQL (SQL server)	Yes	Yes
Apache (Web server)	Yes	Yes
xntp (Network Time Protocol)	No	No
syslog	No	N/a (but the '-a' option allows you to monitor other chroot'ed apps /dev/log files too)
OpenSSH	No	N/a (wouldn't be very useful for root access otherwise)
djbdns (replaces BIND)	Yes	Yes
xinetd	Yes	Yes (optional)
vsftpd (FTP daemon)	Yes	Yes (configurable per user)
stunnel	Yes	N/a
in.tftpd	Yes	Yes
Squid	Yes	Yes
SOCKS	Yes	Yes
arpwatch	Yes	Yes

1.3.7 Accounts

The servers used in the Internet networks effectively only have one 'active' account – namely 'root'. All other accounts (used to separate and protect by function) can only be used via cronjobs, startup scripts, etc - i.e. they don't have useful passwords to 'crack'. Having the same root password on all these hosts would certainly be sensible from a day-to-day administration point-of-view, but of course opens up the possibility of total compromise if just one system is compromised. The solution chosen here is to have 20 character, randomly generated, separate passwords for each system. They are changed every month, and are printed out and kept securely onsite. Any loss of such printouts of course necessitates the immediate generation and changeover to new passwords throughout. Other options included interfacing with the SecurID authentication server via pam_radius, but it was decided that the Security Architecture should be independent of any LAN services. Such an option would also require opening up ports through to the LAN from all Internet networks. Discussion of the authentication system required for remote VPN access will be covered in the RemoteNet section.

1.4 Description of each network

1.4.1 Perimeter network

Internet traffic enters and leaves the GIAC Enterprises network through this point. Due to the importance

of their Internet link, GIAC Enterprises have a fully redundant link via dual-carriers, and their 'perimeter router' actually consists of two Cisco 7420 running VRSP. As the two routers act as one unit, they are referred to as one from here on.

The job of the perimeter router in a firewall is primarily to get rid of the 'noise'. It should act as both the Egress (i.e. stopping non GIAC Enterprises network addresses being used to transmit to the Internet – 'Good Neighbour' philosophy) and Ingress filter (i.e. stopping private network address packets and stopping packets from GIAC Enterprises entering via the Internet) for the site. It should ensure that only application protocols supported/wanted enter and leave the network. All other packets are dropped (not rejected). The perimeter router also NATs packets from the LAN to the Internet.

In the real world, the perimeter router will drop a large amount of traffic. Misconfigured/broken PCs, OSes, routers, and networks, as well as hackers, Trojans, and crackers will be responsible for this. As such, any logging of such bogus traffic needs to be minimized, while dangerous traffic needs to be noted. Without such screening put in place, the staff monitoring the logs will quickly become overwhelmed with the data, will become bored with it, and will start failing to do their job.

Behind the perimeter router is a Cisco PIX 515 firewall. One interface leads to the Service subnet, another to the WebNet subnet, and another to the LAN edge router. This firewall allows the Service subnet to send and receive DNS, SMTP, and NTP queries from everywhere, and allows the Web subnet to receive http and https queries and send/receive ISAKMP (UDP [protocol 17] port 500) and IPSec (protocol 50) packets. The PIX uses Stateful Packet Inspection to try to ensure that the packets it is screening are part of legitimate traffic. The Web subnet is allowed to connect to the ServiceNET DNS, NTP and SMTP servers only. The CSC is allowed to connect through all the networks on the SSH port.

GIAC Enterprises Computer Use Policy allows users to access NNTP, Web, FTP, CVS, SSH and Telnet services. All user access to the Internet is via proxies. So Internet access is only allowed from the LAN proxy servers (NATed). The LAN proxy servers control just what TCP ports are available to end-users. This 'open access' (i.e. allowing the perimeter router to NAT everything from the LAN proxies) is done primarily to minimize the reasons to alter the router configuration. Access is defined, controlled and logged by the LAN proxy servers. Using proxy servers has other advantages over simple NAT firewalls — the logs are better and control is more fine-grained. Any issues regarding staff circumventing the Security Policy are better dealt with by Human Resources than IS. Such issues are more people problems than technical problems.

1.4.2 ServiceNET

The Service Network contains SMTP, DNS and NTP servers, along with SSC and IDS systems. There are two DNS servers here, and there are no off-site secondaries. That eliminates the need for opening TCP port 53 for zone transfers entirely, reducing one popular area of compromise (e.g. Snort currently lists 10 different TCP-based DNS exploits, and 0 UDP).

Another way of potentially limiting damage, is to use variants of primary services, such as BIND. Historically, ISC's BIND has been affected on several occasions by security holes. Using a different DNS server (such as djbdns) – as long as it does the job – has the added benefit of simply being different. A new security hole in BIND is extremely unlikely to affect a different product, and any holes ever found in those alternatives have the one saving grace of being 'less used' – so probability-wise, security staff are more likely to have a longer window in which to fix the problem before being caught out by it.

As far as the lack of off-site secondaries goes, having dual DNS servers in the ServiceNET is almost as

robust as having them off-site. The only way both are going to be down is if the actual Internet is down – at which point having an operating off-site secondary becomes almost without use anyway.

1.4.3 WebNet

The Web subnet contains Web servers (along with a SSC and IDS system) and a Cisco 3600 VPN router. The Web servers all run under chrooted directories as non-root users, which should severely limit the impact of any break-in should it occur. They all use ipchains to limit what packets are allowed in and out of them. Most of these rules duplicate those on the perimeter router and PIX, and are mainly there as fail-safe measures. As everywhere, the CSC is allowed to make SSH connections to these hosts. The Web servers are allowed to respond to Web queries, connect via DNS, NTP and SMTP to the ServiceNET servers, and are allowed to initiate connections to the SQL databases. Any other connection triggers an alert on the IDS systems.

1.4.4 SQLNet

The hosts in this subnet (private network address space) run MySQL within a chrooted, non-root environment. Only the CSC has SSH access to these servers, as well as SQL admin access. The SQL servers have their default routes set to the VPN router. The Partners also require access. Each has a separate DB account, which is only allowed from the network range they register their IPSec connection with. MySQL logs all such connections, giving a strong audit trail. Each Partner 'owns' a database in which they can add, delete or change data. Separate batch processes merges the Partner data into new tables that are then accessed by the Web servers. This separation of data gives Partners good access without compromising security on a large scale. As usual, the SSC and IDS systems are also present, and alert on events that are not expected.

This private subnet was explicitly chosen as yet another mechanism to limit the impact of a compromise. The GIAC network easily accesses this subnet via their normal default routes, but Internet hosts will not be able to even route packets through to it.

1.4.5 RemoteNET

GIAC Enterprises' LAN (private network address space) runs a mixture of Linux and Windows 2000 workstations, with Linux and Sun servers. GIAC Enterprises' staff uses a variety of network applications, and new ones are introduced on a regular basis. Remote access to the LAN therefore needs to permit all application protocols. In contrast, partners and suppliers only require very specific access to GIAC Enterprises, in order to be able to upload, download, or manage their material.

These two groups have different needs and security implications. Consequently:

- Partners and suppliers access the SQLNet subnet via tunnelling over the Internet to a VPN router (see **VPN Network** section)
- Remote staff use second VPN gateway to access the LAN.

For GIAC staff access, a Cisco 3030 VPN gateway (previously known as: Altiga) is used. These routers do nothing but VPN work. As far as ports/protocols go, only the ISAKMP (UDP 500) port and the ESP protocol (IPSEC protocol 50) are available.

GIAC Enterprises decided that all remote access back to the LAN should undergo strong authentication before being allowed. This solution should have a well-established API so that many different products could be interfaced with it, and should contain centralised accounting for audit purposes. As such, a

token-based authentication system was implemented using Security Dynamics SecurID cards, and the primary API chosen was RADIUS. Choosing such an open and well-established standard as RADIUS allows GIAC Enterprises to move to other authentication systems (with RADIUS support obviously) in the future if need be with little effort or alteration to key equipment. Another advantage is that there is a separation of authentication systems between the LAN system and the RemoteNet system. This means even if a hacker had access to all the usernames and passwords in use on the LAN, they would be useless as she wouldn't be able to access the LAN to use them.

The 'Computer Use Policy' therefore states that all off-site access to the LAN must be via a IS-approved method (in terms of technologies used), and must use SecurID cards for authentication (the satellite sites are an exception to this rule and use RSA certificates instead for their LAN-to-LAN tunnels). There was an existing Linux server used for dialup access, this has been moved onto the same switch as the Staff VPN router, and reconfigured to use a PAM RADIUS authentication module for authentication and accounting - thus allowing it to interface with SecurID cards.

The CSC acts as the RADIUS accounting server, the Security syslog server and the Central IDS MySQL server. All systems chosen in a firewall/security capacity must support syslog event generation and (eventually) log back to the CSC.

As this network area is potentially quite dangerous (staff using their unprotected, virus-ridden home computers connecting to the LAN), Policy states that machines used to connect to the GIAC Enterprises network must run a MIS-supplied virus checker.

Finally an IDS system is installed in this area to monitor all traffic from external sites—whether they are satellite LANs, dialup users or 'road warrior' VPN clients. All VPN packets terminate before this switch, so the IDS can deal with 'cleartext' traffic – not encrypted traffic. Snort was chosen as the IDS system for this network as it is easy to understand and configure, and compares (performance/feature-wise) extremely favourably with commercial IDS systems.

1.4.6 Partner/Supplier VPN network

All three categories of external users (Partners, Suppliers and Customers) require secure access to GIAC Enterprises resources in order to do their business. However, there are two classes of access.

Customers and Suppliers: These users simply need secure Web access (https). For customers this is to download and pay for their fortune sayings. For suppliers, this is to upload their latest batch of cookie sayings. Suppliers are given a Web form to fill in for each new fortune saying that they upload. This form ensures the data they enter is correctly formatted and inserted into the backend network SQL servers.

Partners: These users need comprehensive access to GIAC Enterprises backend SQL servers in order to download fortune sayings in bulk. A Web interface would not suffice. As such GIAC Enterprises has stipulated that each partner must use a Cisco 1700 or better router running IOS 12.1(3) with 3DES IPSec for VPN access to the SQLNet. GIAC themselves have a Cisco 3600 running IOS 12.1(3) with 3DES IPSec. Interoperability issues with the Cisco VPN 3000 series (the better product) and consideration to the Partners' budgets/expertise are the main driving factors for this choice.

Each partner has their own SQL server accounts and this limits what actions they can achieve. The GIAC VPN router uses extended ACL lists to limit what services/ports on the Backend Network can be reached by the partners. Again, a Snort IDS system is installed on the backend network segment to ensure

compliance. In fact, as the type of traffic seen on this network segment is so tightly bounded and controlled, any occurrence of any other type of traffic must be an indication of either a compromise or misconfiguration—both of which are priority one events that trigger pager alerts etc.

1.1.7 IDS network

The IDS network consists of Linux systems running on Pentium-III processors with 256MB RAM, Ultra-Wide SCSI-3 HDD and 2x100Mb/s Ethernet interfaces.

One interface runs without an IP address and is plugged into the monitor port of the switch carrying a particular subnet's traffic, and the other has a 192.168.2.* address, and links into the IDS network. No other boxes are allowed on the network other than IDS boxes, and only the CSC plugs back into the LAN.

Snort on these systems reports locally via MySQL and syslog back into the CSC server. Back on the CSC, real-time monitoring of syslog is achieved via swatch, which in turn triggers email and pager alerts as appropriate. The MySQL data is then replicated (native MySQL feature) back onto the CSC MySQL server.

Ipchains is used to ensure that only syslog, MySQL and OpenSSH traffic is acceptable on the IDS network interfaces, and simply block all outgoing traffic on the monitoring interfaces - as all they want to do is listen to traffic there.

It should be noted that no IDS system is installed in front of the initial perimeter router. This does mean that a significant number of events blocked by that router never end up in the IDS database. This decision was made deliberately to limit the number of reported IDS events. It is one thing to have an IDS system - it's quite another to have one going off every 2 seconds due to some script kiddie having a sniff. My design lets the perimeter router do its job in filtering the traffic, and allows the IDS systems to detect 'real' events.

1.1.8 LAN

GIAC Enterprises has decided on an application proxy-based approach to Internet access. Although the perimeter router runs NAT, only the LAN proxy servers are permitted to use it. On them, only authenticated Internet access is allowed, so the Squid and SOCKS servers used internally both require authentication in order to operate. Users connect to Internet resources via them, and they connect directly to the Internet from there.

Email is treated separately. There are SMTP servers in the ServiceNET to handle incoming email, so they are also used for outgoing email. For load reasons, the ServiceNET SMTP servers just do simple checks to limit incoming Spam, that is, RBL checks and sanity checking Email addresses. The larger LAN SMTP servers do other processes such as virus scanning and archiving of Email. The SMTP servers run Qmail 1.03, with Qmail-Scanner (with Sophos and Trend scanners) for virus protection.

Assignment 2: Security Policy (25 Points)

Based on the security architecture that you defined in Assignment 1, provide a security policy for AT LEAST the following three components:

- *Border Router*
- *Primary Firewall*
- *VPN*

You may also wish to include one or more internal firewalls used to implement defence in depth or to separate business functions.

By 'security policy' we mean the specific ACLs, firewall ruleset, IPSec policy, etc. (as appropriate) for the specific component used in your architecture. For each component, be sure to consider internal business operations, customers, suppliers and partners. Keep in mind you are an E-Business with customers, suppliers, and partners - you MAY NOT simply block everything!

(Special note VPNs: since IPSec VPNs are still a bit flaky when it comes to implementation, that component will be graded more loosely than the border router and primary firewall. However, be sure to define whether split-horizon is implemented, key exchange parameters, the choice of AH or ESP and why. PPP-based VPNs are also fully acceptable as long as they are well defined.)

For each security policy, write a tutorial on how to implement each ACL, rule, or policy measure on your specific component. Please use screen shots, network traffic traces, firewall log information, and/or URLs to find further information as appropriate. Be certain to include the following:

- *The service or protocol addressed by the ACL or rule, and the reason these services might be considered a vulnerability.*
- *Any relevant information about the behaviour of the service or protocol on the network.*
- *The syntax of the ACL, filter, rule, etc.*
- *A description of each of the parts of the filter.*
- *An explanation of how to apply the filter.*

If the filter is order-dependent, list any rules that should precede and/or follow this filter, and why this order is important. (Note: instead of explaining order dependencies for each individual rule, you may wish to create a separate section of your practical that describes the order in which ALL of the rules should be applied, and why.)

Explain how to test the ACL/filter/rule. Be certain to point out any tips, tricks, or "gotchas".

2.1 Perimeter Border Router: Cisco

2.1.1 Policy and tutorial

This 'edge' router is the first line of defence. As such, some aspects are secured more than normal:

- No telnet/SSH login options will be enabled — the console will be the only method by which this router can be managed and altered. The console cable can be plugged into the normal UTP wiring, and linked back to the CSC. This can be done over quite a distance, but obviously it means the CSC must be physically in the same general location as the firewall subnets.

- No routing protocols are used in any of the firewall subnets. As there are only a few of them, static routing tables are easy enough to manage, and it removes the possibility of routing

protocol-based attacks.

As GIAC Enterprises wants to be able to monitor their link usage, SNMP is enabled even on this router, but it is bound to the internal interface only. In a similar way, any Cisco service (e.g. TFTP and Syslog clients) that is required, and can be bound to an interface, is explicitly bound to the internal interface.

This Cisco runs IOS 12.1(3) IP/FW. This IOS extends the normal IOS to support Context-Based Access-Control (CBAC), also known as 'Stateful Filtering'. This provides major advantages over simple TCP packet filters in that CBAC allows protected networks to be immunized against SYN-attacks and the like. As these rogue packets aren't part of a standard three-way handshake, CBAC never allows them through. CBAC support for UDP/ICMP is much cruder — this is an inherent problem due to protocol design.

Once configured correctly, this router shouldn't need to be touched for months on end. It should be very low maintenance.

The following three practical tips need to be observed:

1. Ciscos have a habit of deleting configuration lines that are the default. If you hand-write a configuration, explicitly mentioning all these things that the Cisco should and shouldn't do, you may find that the running config has some lines missing. This can have a detrimental effect when there is a major IOS upgrade to a Cisco version with different default settings. A feature that was turned on by default may suddenly be turned off. The lack of an explicit rule can mean the difference between a secure system and an insecure one. Always test after upgrades!

2. By default, Ciscos' access-lists 'deny' calls cause an ICMP 'host unreachable' (ICMP code 1) to be sent to the client (also known as a 'reject'). Other more sophisticated firewalls give you other options such as RESETs for TCP packets, as well as dropping the packet (not sending anything back). To 'drop' the packet with a Cisco, set 'no ip unreachables' on each interface you wish this to apply to. Unfortunately it's an all-or-nothing thing. If you want to drop some packets, but reject others, then you will have to let such packets through the Cisco, and reject them via reject rules on later devices (e.g. ipchains/iptables, Checkpoint firewalls).

3. Use the 'log-input' command instead of just 'log' when logging packets that are denied by Cisco ACLs. 'log-input' includes the MAC address of the packet being dropped, whereas log doesn't. When the packet is coming from your own networks, the MAC address can be very useful for figuring out who is responsible for it (i.e. differentiates routers from hosts).

For the rest of the assignment, any references to entities by the logical names listed in Table 1 should be read as being IP addresses and/or subnets.

Table 1 Entity logical names used in this assignment

Name	**Referred to as**
SSC host	ssc_host
CSC host	csc_host
Partner VPN router	partner_vpn

ServiceNET network	service_net
WebNet network	web_net
SQLNet network	sql_net
GIAC LAN	giac_lan
GIAC LAN proxy servers	giac_proxies

The following rules need to be implemented:

1. CSC_host needs to be able to read SNMP variables
2. Perimeter router should log its SNMP traps and syslog messages to the ServiceNET SSC
3. Internet needs to access SMTP and DNS servers in ServiceNET
4. Internet needs to access Web servers in WebNet
5. Internet needs to access IPSec router in WebNet
6. ServiceNET needs to access SMTP, DNS and NTP servers on Internet
7. ServiceNET needs to access SMTP servers on the LAN
8. GIAC LAN proxy servers need to have NAT'ed full Internet access
9. GIAC LAN servers need access to all Internet subnets.
10. CBAC should be used to implement stateful packet filtering

Point 7 is often overlooked in security policies. Generally a firewall environment is designed such that the hosts in the firewalled area never have access to the LAN, for obvious reasons. However, when it comes to Email, this rule is often broken. A more secure approach would be to have all Email received by the ServiceNET SMTP servers to be delivered locally onto their hard disks. Then the internal mail servers could use tools like fetchmail to download that Email via POP or IMAP (thus they instigate the connection), and reprocess it to the final destinations. This is certainly more secure than allowing ServiceNET hosts to connect to the LAN, but now Email becomes batch-driven and delays of a minute or more would be added to delivery time. Email is such an essential service these days that people expect it to be as responsive as possible. GIAC Enterprises has decided that an exception is made for Email, thus point 7 remains.

2.1.2 Perimeter Router Config

```
!Turn off silly small services like ECHO, http
 no udp-small-servers
 no tcp-small-servers
 no ip direct-broadcast
 no ip bootp server
 no ip http server
 !
 !Syslog sees informational and worse - all via Fe0
 logging source-interface FastEthernet0
 logging ssc_host  informational
```

```
 !
!Stop all source-routed packets
 no ip source-route
 !
 ! ensures your config loads always occur off the internal Ethernet int
 ip tftp source-interface FastEthernet0
 !
 ! NAT translation rules
 ! Do static NAT - i.e. map the internal GIAC proxy server
 ! to a valid Internet address so that all Internet applications
 ! it could ever want to use have the highest possibility of working
 ip nat inside source static giac_proxy internet_address_of_proxy
 !
 !CBAC ruleset
 !
 !set some timeouts
 ip inspect dns-timeout 10
 !
 !Specifies how many half-open TCP sessions with the same host address
 !can exist at a time, before the software starts deleting half-open
sessions
 ip inspect tcp max-incomplete host 100 block-time 5
 !
 !GIAC LAN to Internet rules
 !Handle odd apps like TFTP and FTP
 ip inspect name giac2internet ftp timeout 3600
 ip inspect name giac2internet realaudio timeout 3600
 ip inspect name giac2internet tftp timeout 10
 !
 !CBAC known to "get it wrong" with SMTP and HTTP.
 !Will just let the TCP default do for them...
 ip inspect name giac2internet tcp timeout 3600
 ip inspect name giac2internet udp timeout 10
 !
 !Internet to Service Networks rules
 !
 ip inspect name internet2service tcp timeout 3600
 ip inspect name internet2service udp timeout 10
 !
 !
 !SNMP info
 snmp-server community jmfey7ndy236nchsd765t4bfc4 RO 21
 snmp-server trap-source FastEthernet0
 snmp-server location GIAC Enterprises - Internet Perimeter Router
 snmp-server contact network_engineering@giacenterprises.com
 snmp-server enable traps snmp authentication linkdown linkup coldstart
 snmp-server enable traps config
 snmp-server enable traps syslog
 snmp-server enable traps rtr
 snmp-server host ssc_host
 !
 interface Serial0
  description Internet Link
  no ip redirects
  no ip directed-broadcast
  no ip proxy-arp
  no ip broadcast-address
  no ip route-cache
```

```
 no cdp enable
 ip access-group 101 in
 ip accounting access-violations
 ip inspect internet2service in
! Drop rejected packets - that'll slow hackers down a bit
 no ip unreachables
! Define this as the outside of the NAT environment
 ip nat outside
!
!
interface FastEthernet0
 description GIAC Perimeter subnet
 ip access-group 102 in
 no ip directed-broadcast
 no ip proxy-arp
 ip accounting access-violations
 ip inspect giac2internet in
 no ip route-cache
 no cdp enable
! Don't drop rejected packets - we want our users to know as
! soon as possible when they're doing something wrong
 ip unreachables
! Define this as the inside of the NAT environment
 ip nat inside
!
!Always delete access list as first action. That way you always know
!what order rules will be run in.
no access-list 21
access-list 21 permit   csc_host
access-list 21 deny   any log
!
no access-list 101
access-list 101 remark Incoming Internet to Perimeter
access-list 101 deny ip giac_lan any
access-list 101 deny ip service_net any
access-list 101 deny ip web_net any
access-list 101 deny ip sql_net any
!As we are not sending unreachables, we'll have to let
!ident in, and reject it at the host end to stop long
!timeouts from occurring
access-list 101 permit tcp any service_net 113
access-list 101 permit tcp any service_net 25
access-list 101 permit udp any service_net 53
access-list 101 permit tcp any web_net 80
access-list 101 permit tcp any web_net 443
!Allow IPSec
access-list 101 permit udp any 500 partner_vpn 500
access-list 101 permit 50 any partner_vpn
!
access-list 101 permit udp any service_net 123
!CBAC dynamically generated rules kick in here...
!Block and log everything else
access-list 101 deny ip any any log-input
!
no access-list 102
access-list 102 remark Outgoing GIAC packets to the Internet
!Block all protocol types you don't want the GIAC LAN to
!ever initiate
```

```
access-list 102 deny   tcp any any eq 111 log-input
access-list 102 deny   udp any any eq 111 log-input
access-list 102 deny   tcp any any eq 2049 log-input
access-list 102 deny   udp any any eq 2049 log-input
access-list 102 deny   tcp any any eq 745 log-input
access-list 102 deny   udp any any eq 747 log-input
!Block these too - but don't log them as they're just noise
access-list 102 deny   tcp any any eq 137
access-list 102 deny   tcp any any eq 138
access-list 102 deny   tcp any any eq 139
access-list 102 deny   udp any any eq 137
access-list 102 deny   udp any any eq 138
access-list 102 deny   udp any any eq 139
access-list 102 deny   udp any any eq 443
!
access-list 102 permit tcp giac_proxies any
access-list 102 permit udp giac_proxies any
access-list 102 permit tcp service_net any 25
access-list 102 permit udp service_net any 123
access-list 102 permit udp service_net any 53
access-list 102 permit udp 500 partner_vpn any 500
access-list 102 permit 50 partner_vpn any
!Block and log everything else
access-list 102 deny ip any any log-input
!
line con 0
 exec-timeout 0 0
 transport input none
line aux 0
line vty 0 4
 access-class 12 in
 password 7 $1$imdst8m345nuehfjdhf
 login
 escape-character 3
!
end
```

2.2 PIX Firewall

2.2.1 Policy and tutorial

The PIX Firewall is the primary firewall. It connects the Internet in various filtered ways to the ServiceNET and WebNet networks.

The following rules need to be implemented:

1. The PIX is managed directly via SSH from the CSC and can also read and write its configuration via TFTP to the CSC.
2. Internet needs to be able to access ServiceNET hosts via SMTP, DNS (UDP only as GIAC runs its own secondaries—so XFERs aren't required) and NTP.
3. All Internet subnet hosts need SMTP, DNS and NTP access to the ServiceNET.
4. Internet needs to be able to access WebNet hosts via HTTP and HTTPS.
5. Partners and Suppliers need to be able to VPN via IPSec to the VPN router in the WebNet, from

where they gain access to the SQLNet.

6. The GIAC LAN proxy servers need fairly open access to the Internet.

Stateful packet filtering should be turned on. Don't use the SMTP and HTTP-specific options. These are notorious for breaking valid traffic, so rely on well-maintained host security to stop this being a concern.

2.2.2 PIX Firewall Config

```
PIX Version 5.2(5)
!Note: a packet from a higher "security" interface to a lower
!"security" interface is "allow". A packet from a lower
!"security" interface to a higher defaults to "deny".
nameif ethernet0 outside security0
nameif ethernet1 web_net security20
nameif ethernet1 service_net security40
nameif ethernet2 sql_net security50
nameif ethernet3 lan security100
!
enable password XxXxXxXxXx encrypted
passwd XxXxXxXxXx encrypted
hostname pix-fw
!fixup is stateful inspection. e.g. For ftp it will "merge"
!the FTP control and data connections into one object WRT ACL
!comparisons.
fixup protocol ftp 21
fixup protocol h323 1720
!
names
pager lines 24
logging on
no logging timestamp
no logging standby
no logging console
no logging monitor
logging buffered debugging
logging trap debugging
logging history debugging
logging facility 20
logging queue 512
logging host service_net ssc_host
ip address outside w.w.w.w
ip address service_net x.x.x.x
ip address sql_net y.y.y.y
ip address lan z.z.z.z
ip audit info action alarm
ip audit attack action alarm
arp timeout 14400
timeout xlate 3:00:00
snmp-server host service_net ssc_host
snmp-server location GIAC Enterprises - Internet Perimeter Router
snmp-server contact network_engineering@giacenterprises.com
snmp-server community kdjsf8932dksnfd9fdnmfdnf
snmp-server enable traps
floodguard enable
ssh csc_host lan
ssh timeout 5
!
access-group acl_internet in interface outside
```

```
access-group acl_service in interface service_net
access-group acl_web in interface web_net
access-group acl_lan in interface lan
!
access-list acl_internet permit tcp any web_net eq 80
access-list acl_internet permit tcp any web_net eq 443
access-list acl_internet permit tcp any service_net eq 25
access-list acl_internet permit udp any service_net eq 121
access-list acl_internet permit udp any service_net eq 53
access-list acl_internet permit udp host perimter-rtr service_net eq 514
access-list acl_internet permit udp any web_net eq 500
access-list acl_internet permit 50 any host partner_vpn
!
access-list acl_service permit tcp service_net any eq 25
access-list acl_service permit tcp service_net any eq 53
access-list acl_service permit udp service_net any eq 53
access-list acl_service permit udp service_net any eq 121
access-list acl_service permit udp host perimter-rtr any eq 514
access-list acl_service permit udp host pix-fw any eq 514
!
access-list acl_web permit tcp web_net service_net  eq 25
access-list acl_web permit udp web_net service_net  eq 53
access-list acl_web permit udp web_net service_net  eq 121
access-list acl_web permit udp host partner_vpn eq 500 any eq 500
access-list acl_web permit 50 host partner_vpn any
!
access-list acl_lan permit tcp host csc_host any
access-list acl_lan permit udp host csc_host any
access-list acl_lan permit tcp giac_proxies any
access-list acl_lan permit udp giac_proxies any
```

2.3 Cisco VPN 3000 (Altiga)

2.3.1 Policy and tutorial

The Altiga is the VPN router used to connect remote GIAC Enterprise staff back to the corporate LAN. It supports both LAN-to-LAN and remote client (also known as 'road warrior') connections. The Altiga supports IPSec, PPTP and L2TP for LAN-to-LAN (authentication methods include pre-shared keys and PKI), and an array of authentication options for the proprietary remote client support (authentication methods include RADIUS, NT Domain, SecurID and an internal auth database).

The following rules need to be implemented:

1. The Altiga is managed via https and telnet/SSL only.
2. Only IPSec will be used for all VPNs.
3. LAN-to-LAN tunnels will be via pre-shared keys.
4. Remote client access will require RADIUS authentication off backend SecurID server. RADIUS will also be used for accounting .
5. Remote clients will be assigned a LAN address out of a 'pool' assigned to the Altiga
6. Idle and maximum connection-time timeout limit of 1 hour and 8 hours respectively are set for remote access. LAN-to-LAN tunnels have no such limits of course.

2.3.2 Cisco VPN 3000 Config

Unlike the previous two components, the Altiga is primarily configured via its Web interface. **Figure 2** to **Figure 8** contain screen snaps to show how it is configured.

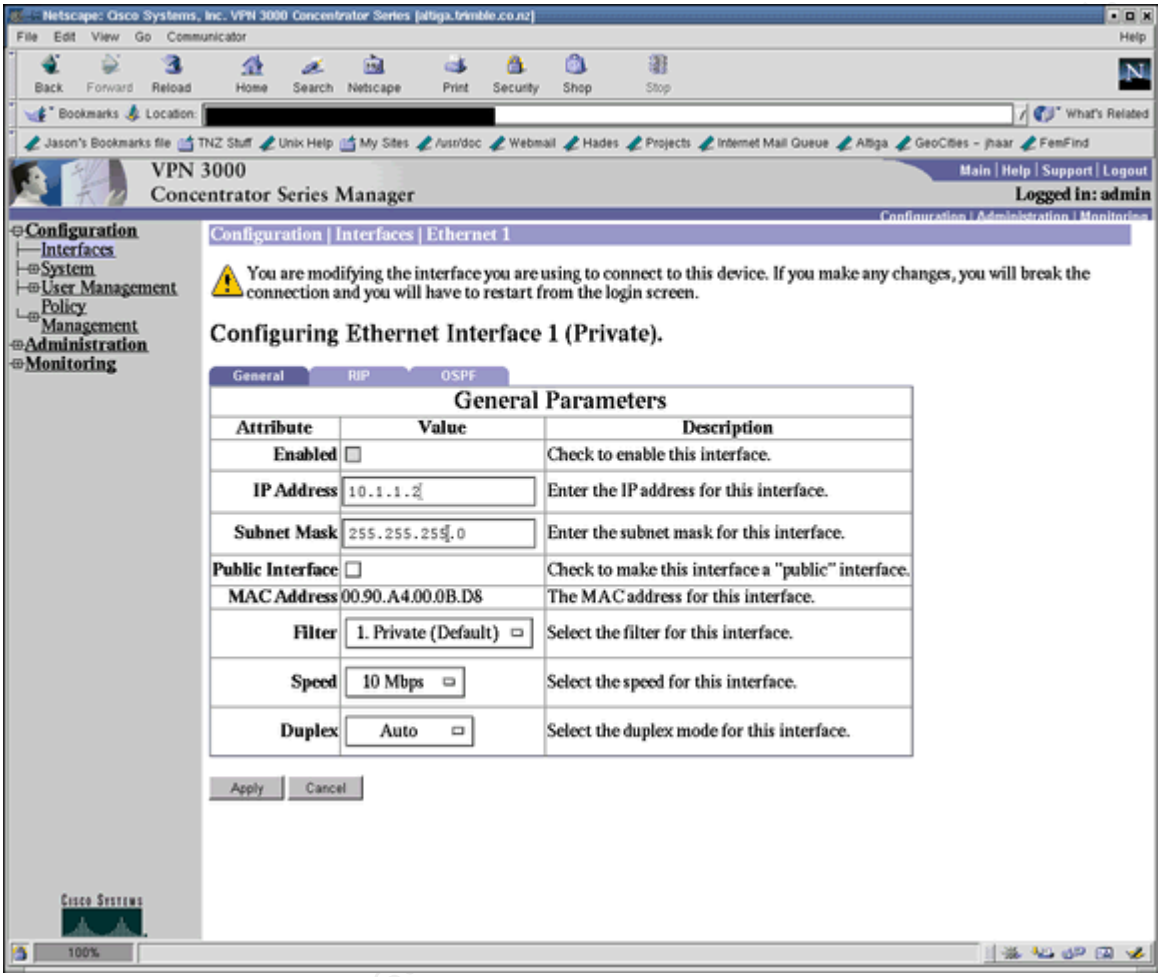


Figure 2 Screen snap showing how to configure the LAN interface of the VPN device. The Altiga makes some pretty sane assumptions: all Altiga services - such as it's telnet and Web interface - are only available on its private interface - not its public.

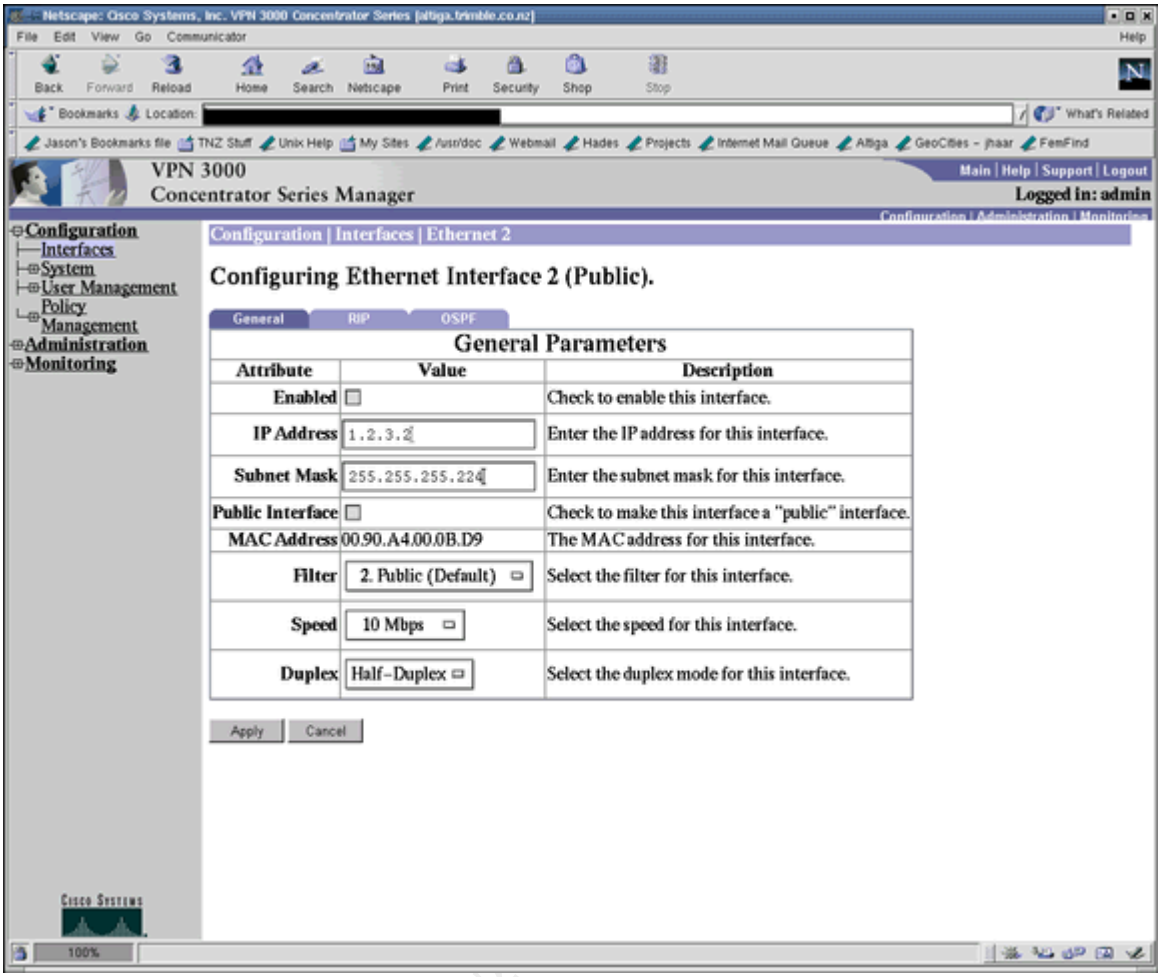


Figure 3 Screen snap showing how to configure the public (i.e. Internet) interface of the VPN device.

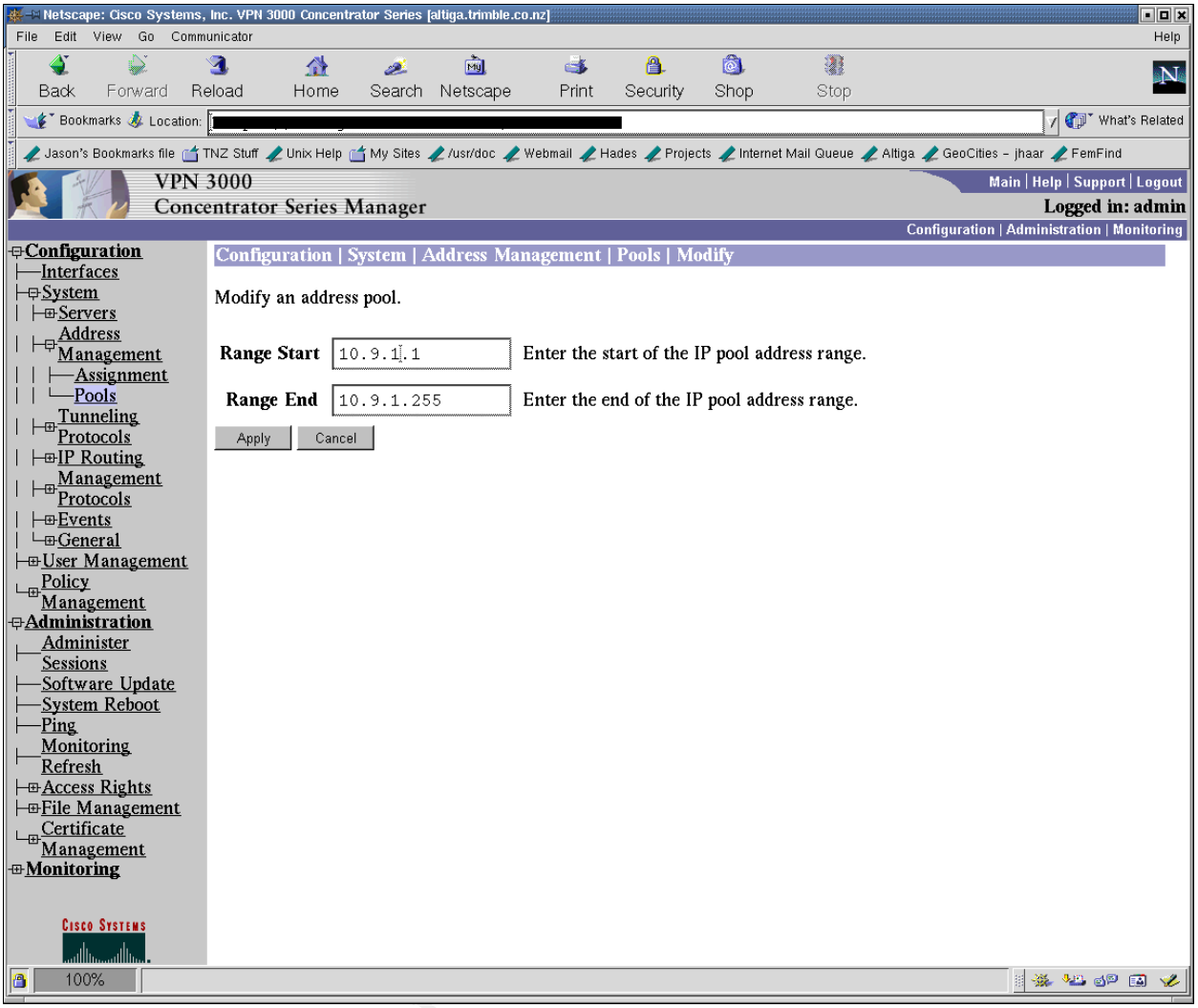


Figure 4 Screen snap showing how to configure the range of IP addresses the Altiga will assign to any remote clients (i.e. 'road warrior' support).

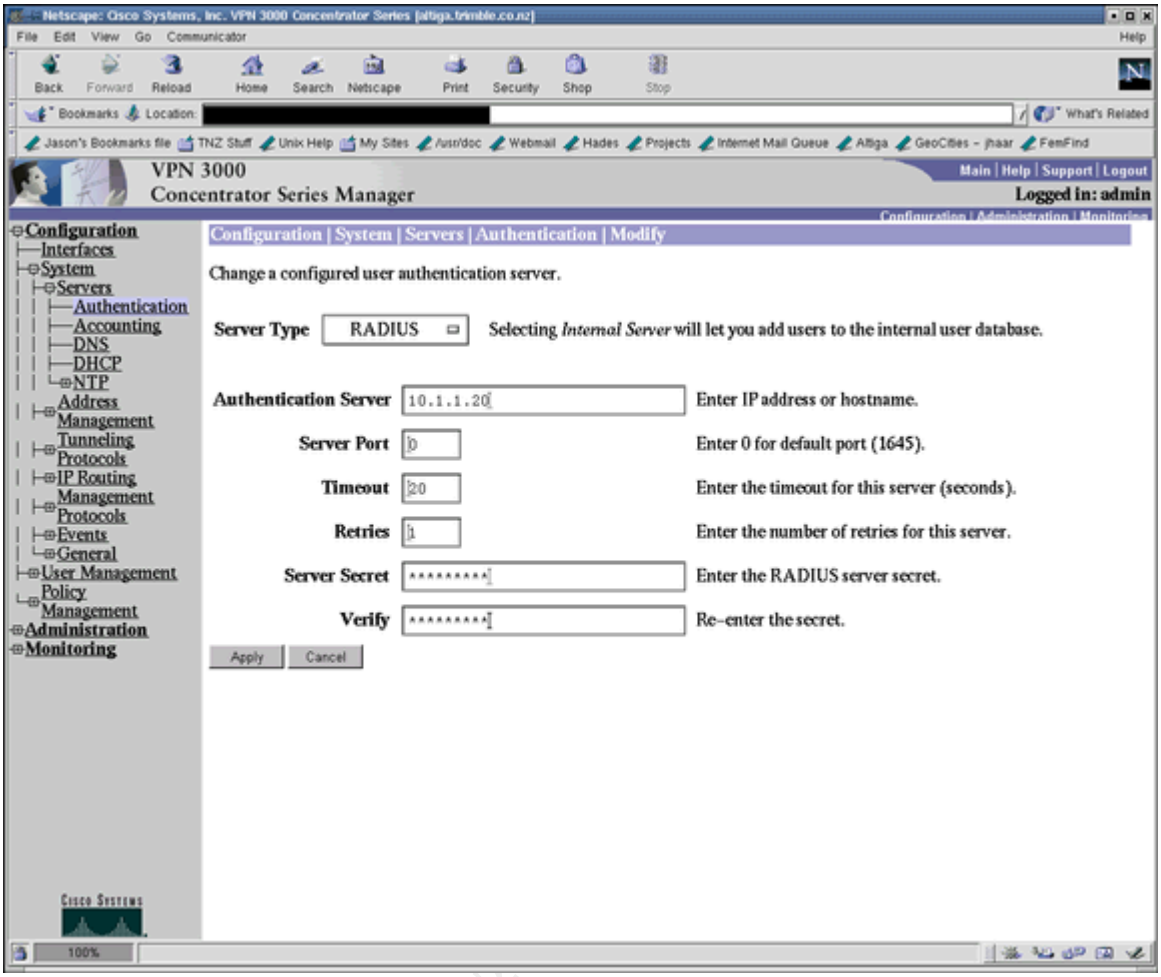


Figure 5 Screen snap showing how to configure what authentication method should be used for authenticating remote clients. Having several is possible as different 'profiles' can be configured to use different authentication methods. Here RADIUS is chosen and it points to GAIC Enterprises SecurID server.

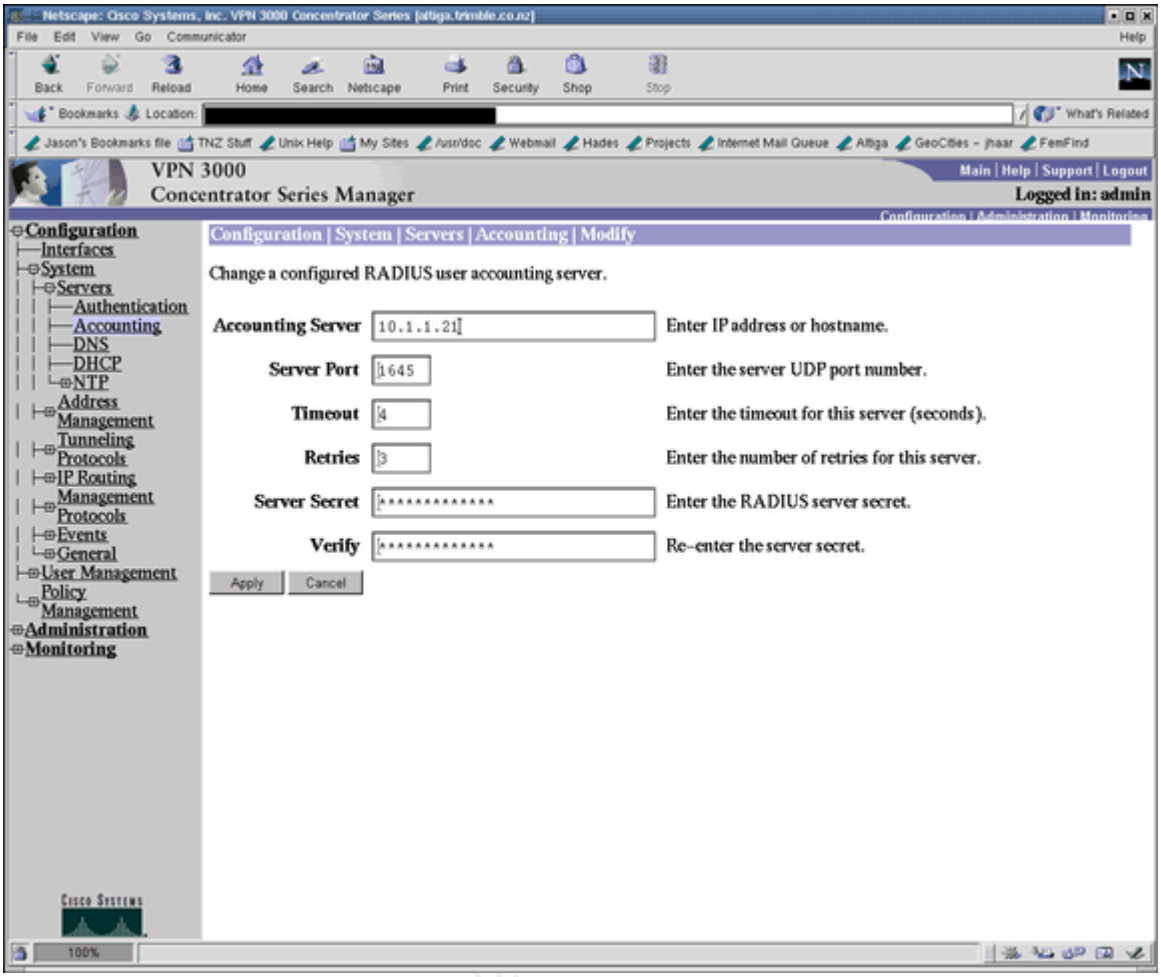


Figure 6 Screen snap showing how to configure RADIUS accounting. This gives in-depth details of how the Altiga has been used by remote clients, including who is using it, when they used it, for how long, and how much traffic was generated.

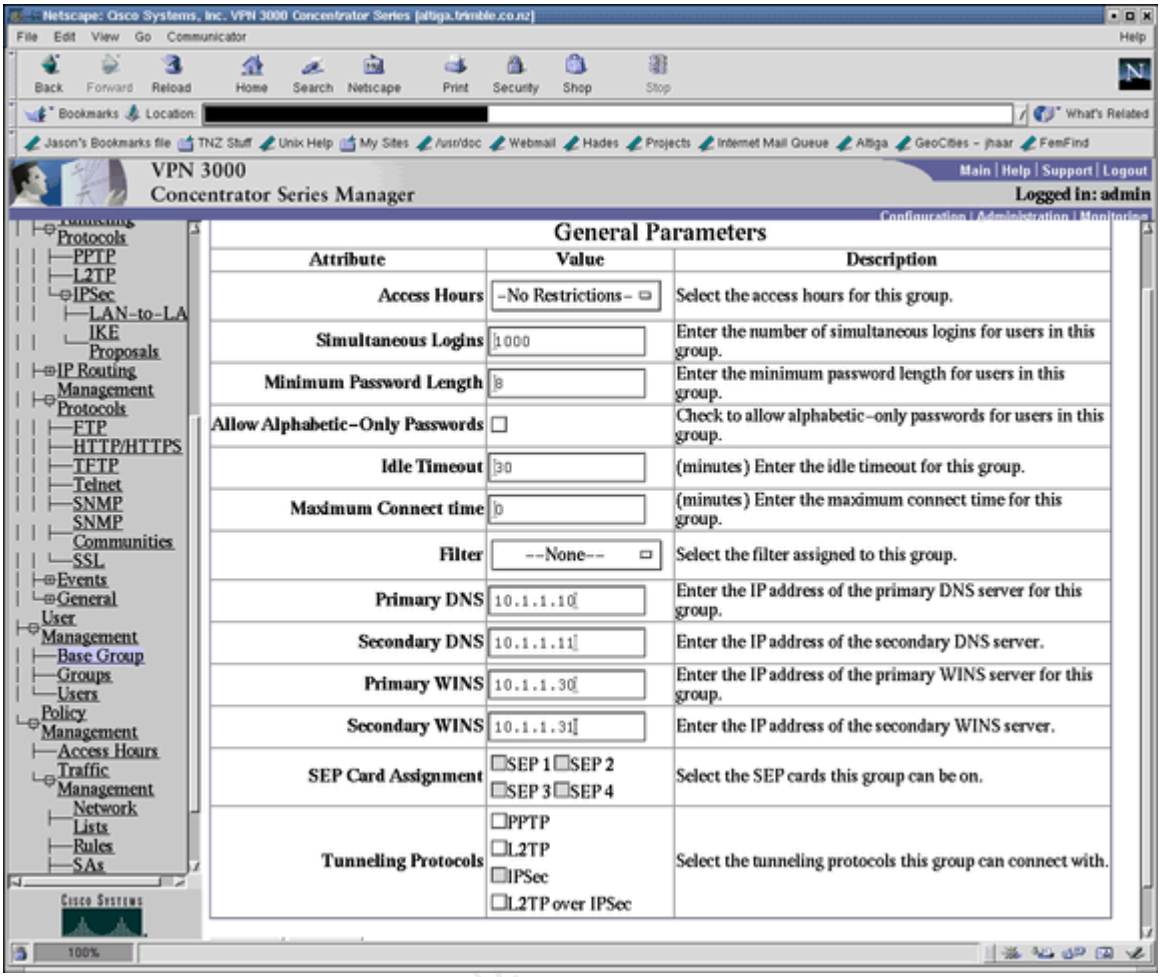


Figure 7 Screen snap showing the default settings associated with this remote client profile. Similarly to PPP, you can set the DNS and WINS server addresses on the client, how long the VPN session is valid for, and what VPN protocols are supported.

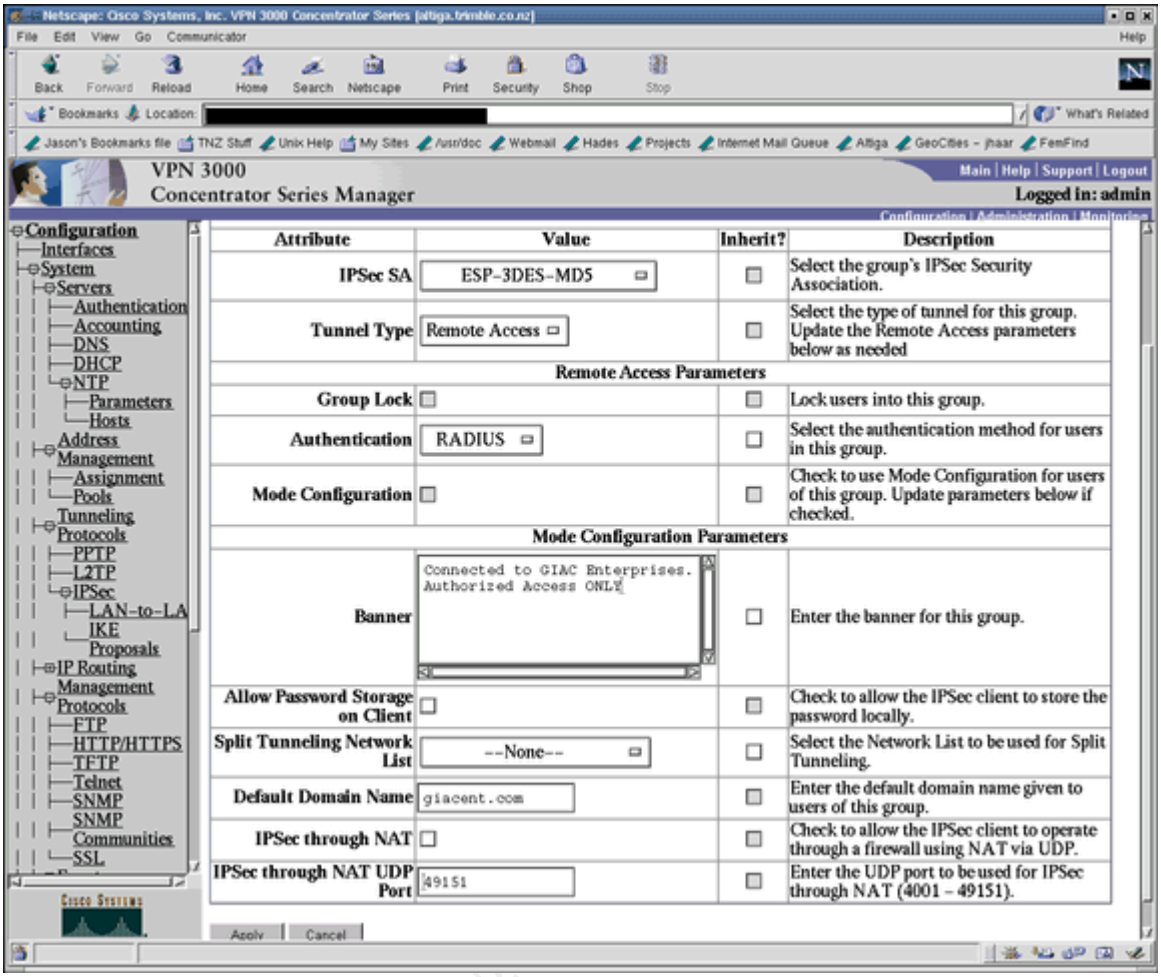


Figure 8 Screen snap showing more settings. Here you can set what quality encryption is supported, along with pop-ups that will appear on the client after they have successfully authenticated. Here is where you choose which authentication method (configured earlier) is wanted for this profile.

2.4 IPCHAINS for Server Hosts

2.4.1 Policy and Tutorial

Linux systems based on the 2.2 kernel natively come with IP firewall support (ipchains). This technology can be compared with the Cisco 'access-list' command set in terms of coverage – although ipchains allows finer tuning down to the IP header level. The newer 2.4 kernel comes with iptables – which has been compared favourably with the PIX and Checkpoint as far as it's Stateful Filtering support goes – a feature that ipchains generally lacks. As iptables is still relatively new, and there are other firewall layers above the Linux hosts, they will use the older (but more tested) ipchains for packet filtering.

The following rc script shows how a Linux system can be 'hardened' at boot time so that it will resist a network attack should an attacker somehow bypass the firewall systems.

2.4.2 IPCHAINS Config

```
#/etc/rc.d/init.d/firewall
#
```

```
# For Linux 2.4.2 Web server in WEB_NET
#

MYADDR=`/sbin/ifconfig eth0|grep 'inet addr:'|sed -e 's/^.*inet
addr://g'|cut -d' ' -f1`
SSC_HOST="XXX.XXX.XXX.2"
CSC_HOST="AAA.AAA.AAA.2"
SERVICE_NET="BBB.BBB.BBB.0/22"
SQL_NET="CCC.CCC.CCC.0/22"

# Set kernel settings to block many IP related problems:
sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=1
sysctl -w net.ipv4.conf.default.accept_source_route=1
sysctl -w net.ipv4.conf.default.accept_redirects=0
sysctl -w net.ipv4.conf.default.forwarding=0
sysctl -w net.ipv4.conf.all.send_redirects=0
sysctl -w net.ipv4.conf.all.rp_filter=1
sysctl -w net.ipv4.conf.all.accept_redirects=0
sysctl -w net.ipv4.conf.all.forwarding=0
sysctl -w net.ipv4.tcp_ecn=0
sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=1
sysctl -w net.ipv4.tcp_syncookies=1
#
# By default REJECT all services except forwarding, which should be denied
# (i.e. no notification to baddie that this isn't allowed)

ipchains -A input  DENY
ipchains -A output -DENY

#This kernel doesn't support IP forwarding, but still turn  -F to DENY
#just in case a bad kernel is put on this system somehow...
ipchains -A forward  DENY

# Flush all commands
ipchains -A input -F
ipchains -A output -F
ipchains -A forward -F

#Always allow the firewall to connect to it's own addresses
ipchains -A input -s 127.0.0.1 -d 127.0.0.1 -i lo -j ACCEPT
ipchains -A output -s 127.0.0.1 -d 127.0.0.1 -i lo -j ACCEPT
#Drop and Log any other localhost packets
ipchains -A input -s 127.0.0.1 -d 127.0.0.1 -j DENY -l

#Allow host to connect to itself
ipchains -A input -s $MYADDR -d $MYADDR -i eth0 -j ACCEPT
ipchains -A output -s $MYADDR -d $MYADDR -i eth0 -j ACCEPT
#Drop and Log any other MYADDR packets (spoofed)
ipchains -A input -s  $MYADDR  -j DENY -l

#Allow everything from CSC_HOST
ipchains -A input -s $CSC_HOST -d $MYADDR -i eth0 -j ACCEPT
ipchains -A output -d $CSC_HOST -s $MYADDR -i eth0 -j ACCEPT

#Allow host to talk to SSC host
ipchains -A output -s $MYADDR -d $SSC_HOST syslog -p UDP -i eth0 -j ACCEPT
ipchains -A output -s $MYADDR -d $SSC_HOST snmptrap -p UDP -i eth0 -j
ACCEPT
```

```
ipchains -A input -d $MYADDR -s $SSC_HOST syslog -p UDP -i eth0 -j ACCEPT
ipchains -A input -d $MYADDR -s $SSC_HOST snmptrap -p UDP -i eth0 -j ACCEPT
ipchains -A output -s $MYADDR -d $SSC_HOST -p ICMP -i eth0 -j ACCEPT
ipchains -A input -d $MYADDR -s $SSC_HOST -p ICMP -i eth0 -j ACCEPT

#DROP ident connections - on the Service_Net it's better to REJECT
#them as many mail servers attempt ident lookups. But on a Web network
#that "rule" can be ignored as no Web client does ident lookups.
#Anything that does, I want to know about...
ipchains -A input -d $MYADDR ident -i eth0 -j DROP -l

#Drop all ICMP except allowed types
#But don't log as there'll be too many
for icmptype in source-quench destination-unreachable time-exceeded
do
  ipchains -A input -d $MYADDR --icmp-type $icmptype -p ICMP   -i eth0 -j
ACCEPT
done
ipchains -A input -p ICMP -i eth0 -j DENY

#Allow Web connections from everywhere
# (note lack of stateful packet inspection in ipchains)
# (iptable ROCKS!)
ipchains -A input  -d $MYADDR http -p TCP -i eth0 -j ACCEPT
ipchains -A input  -d $MYADDR https -p TCP -i eth0 -j ACCEPT
ipchains -A output -s $MYADDR http -p TCP -i eth0 -j ACCEPT !-y
ipchains -A output -s $MYADDR https -p TCP -i eth0 -j ACCEPT !-y

#Allow host to send Email, do DNS and NTP queries
ipchains -A output -s $MYADDR -d $SERVICE_NET smtp -p TCP -i eth0 -j ACCEPT
ipchains -A output -s $MYADDR -d $SERVICE_NET dns -p UDP -i eth0 -j ACCEPT
ipchains -A output -s $MYADDR -d $SERVICE_NET ntp -p UDP -i eth0 -j ACCEPT
ipchains -A input -d $MYADDR -s $SERVICE_NET smtp -p TCP -i eth0 -j ACCEPT
!-y
ipchains -A input -d $MYADDR -s $SERVICE_NET dns -p UDP -i eth0 -j ACCEPT
ipchains -A input -d $MYADDR -s $SERVICE_NET ntp -p UDP -i eth0 -j ACCEPT

#Allow host to talk to MySQL servers
ipchains -A output -s $MYADDR -d SQL_NET 3306 -p TCP -i eth0 -j ACCEPT
ipchains -A input -d $MYADDR -s SQL_NET 3306 -p TCP -i eth0 -j ACCEPT !-y

#Done.
#Now block and log anything else
#
ipchains -A output -j DENY -l
ipchains -A input -j DENY -l
#Block and don't log forwarded packets. If you're hanging off a hub, this
#can lead to you logging EVERY packet sent by any other host!!!
ipchains -A forward -j DENY
```

Assignment 3: Audit Your Security Architecture (25 Points)

You have been assigned to provide technical support for a comprehensive information systems audit for GIAC Enterprises. You are required to audit the Primary Firewall described in Assignments 1 and 2. Your assignment is to:

- *Plan the assessment. Describe the technical approach you recommend to assess your perimeter. Be certain to include considerations such as what shift or day you would do the assessment. Estimate costs and level of effort. Identify risks and considerations.*
- *Implement the assessment. Validate that the Primary Firewall is actually implementing the security policy. Be certain to state exactly how you do this, including the tools and commands used. Include screen shots in your report if possible.*
- *Conduct a perimeter analysis. Based on your assessment (and referring to data from your assessment), analyse the perimeter defence and make recommendations for improvements or alternate architectures. Diagrams are strongly recommended for this part of the assignment.*

Note: DO NOT simply submit the output of nmap or a similar tool here. It is fine to use any assessment tool you choose, but annotate the output.

3.1 Assessment plan

An objective audit is more likely to arise from an external team than from an internal team, since the latter is likely to comprise the same people who implemented the security architecture in the first place. However the cost and disruption arising from frequent external audits is not justifiable for GIAC Enterprises. Here, auditing should be carried out by both the internal staff and by an external teams from a reputable company. Frequent internal audits are likely to focus only on areas already known to be strong, but will nevertheless still be 90 per cent thorough. Less frequent (every year or so) external audits can provide that extra 10 per cent coverage, and give the internal security staff confidence in what they're doing (assuming external audits discover nothing exceptional).

Any penetration attempt has the potential to uncover unknown network problems. These problems could impact the performance of services, or even cause system failure. The audit team therefore needs to have sign-off from senior management that their audit is authorized and operates with 'all care, but no responsibility'. If the systems involved are kept up-to-date, such problems are unlikely.

The audit should be carried out during normal business hours, as any unexpected events can best be dealt with all staff being available. Also, if any of the tests overly tax the capabilities of a system, that test can be immediately cancelled, therefore removing the cause of the DoS, and can be logged as a successful DoS attack.

Network security audits are by their nature methodical and repetitive, suggesting automation (at least where they are internally lead). Such 'scripted' audits ensure consistency and can be the network equivalent of Tripwire — they demonstrate the systems are responding as they should, and that nothing has changed/been added since the last run.

The GIAC Enterprises security architecture consists of four component types that need to be audited:

1. Physical Security: A fairly obvious one, but if physical security of protected devices is compromised, then the devices are compromised. At the very minimum, DoS can occur (e.g. a cleaner turned the firewall off to vacuum the floor), and at worse, total compromise of data. All

components of the security architecture need to be in a physically secure environment. GIAC Enterprises needs to have a locked machine-room with limited and monitored access. e.g. swipe cards.

2. Network Devices: In most ways these are very similar to hosts, and need to be audited in a similar fashion. However, they play a pivotal role in the security architecture, as they tend to join together components of the network. Compromise of a firewall usually provides the perpetrator with access to several parts of the network, something to be avoided at all costs. Network devices need to be as inaccessible as possible. As mentioned earlier, switches need to have all management options disabled so that no-one else has any chance of commandeering them. Routers and firewalls similarly should either not be accessible remotely (i.e. console only), or should be accessible from only one address (the Central Security Console in our case). Thankfully, normally firewalls and routers can be configured so that packets terminating on them are rejected, as they normally do not provide any network services themselves.

3. Service computers: Auditors should check that the OSes of the Service computers' are up-to-date (without known security holes). Systems should be minimal in order to minimize exposure to security holes, therefore they should contain only software they actually use. Application security is even more important. The vast majority of security holes within a particular architecture will be within applications, not OS or network devices. Monitoring mailing lists/newsgroups and web sites for any issues with these applications will be part of the daily tasks of those involved in the maintenance and upkeep of the GIAC Enterprises security architecture.

4. Penetration scenarios: These would be instigated from the Internet and from within each internal network. Staff involved should not be told of the more in-depth penetration attempts, so that their response to such attacks can be measured. This is similar to testing fire alarms: there is no point having a wonderful system in place if everyone ignores it.

In any audit, it is a good idea to have a checklist. This ensures consistency in the analysis, and allows for the migration of responsibilities between staff. Obviously the tests need to be altered as new attack methods and risks become known. A checklist should contain at least the following:

- Check physical security. Follow all cables to ensure only known equipment is plugged into the firewall networks.
- Check OS versions of hosts and network devices against current known best. Justify why you are not running known best if that is the case. Check version of all network applications against current known best
- Consider how up-to-date you feel you are with the components of the firewall networks.
- Are you following newsgroups/mailing-lists/Web sites on a daily basis? Do you feel well informed?
- Conduct penetration attempt from Internet host. This should result in numerous security log messages, but no alerts.
- Conduct penetration attempt from hosts internal to each firewall subnet. This should result in numerous security log messages, and plenty of alerts, as it is indicative of a compromise. Testing of alerting systems should be proven by the above two tests.

3.2 Assessment implementation

3.2.1 Internet Penetration Attempt

For an Internet penetration attempt, security auditors should use a typical dialup or DSL ISP account. This reflects the standard type of connection a real intruder would use, and ensures that the tests aren't coming from an internal address, which would change the way the perimeter environment would respond. Auditors should try to map the GIAC network from an attacker's perspective. They should see how much information can be gleaned from public sources, then proceed with portscans to map what is actually available, and finish with application vulnerability scans.

Auditors should assume that an attacker knows the name of the GIAC network and can therefore use DNS lookups, whois, and Web searches to discover information about the network. Such lookup tools are, from the attacker's point of view, zero-risk, because they can be done without a single packet entering or leaving the GIAC network. An attacker may learn where GIAC is located physically (issues: dumpster diving, physical breakin attempts), as well as more network-related information. The following whois and nslookup output demonstrates the breadth of the official information about GIAC Enterprises that is easily discoverable.

```
$ fwhois giacenterprises.com@whois.networksolutions.com
      Registrant:
      GIAC Enterprises
      1411 Willow Drive
      New Tripoli, PA 18066
     US
      Domain Name: GIACENTERPRISES.COM
      Administrative Contact:
      Jeremy Welling jwelling@AOL.COM
      1411 Willow Drive
      New Tripoli, PA 18066
      610-555-1362
      Technical Contact, Billing Contact:
      Adam Mendelssohn admendel@AOL.COM
      1411 Willow Drive
      New Tripoli, PA 18066
      610-555-1348
      Record last updated on 16-Aug-2000.
      Record expires on 05-Aug-2002.
      Record created on 16-Aug-2000.
      Database last updated on 19-Feb-2001 10:23:07 EST.
      Domain servers in listed order:
      DNS1.GIACENTERPRISES.COM NNN.NNN.NNN.11
      DNS2.GIACENTERPRISES.COM NNN.NNN.NNN.12

      $ nslookup -type=any giacenterprises.com.
      Server:  localhost
      Address:  127.0.0.1
      Non-authoritative answer:
      giacenterprises.com    nameserver = ns1.giacenterprises.com
      giacenterprises.com    nameserver = ns2.giacenterprises.com
      giacenterprises.com    preference = 10, mail exchanger =
mail.giacenterprises.com
      giacenterprises.com
              origin = giacenterprises.com
              mail addr = hostmaster.giacenterprises.com
              serial = 200009210
              refresh = 7200 (2H)
              retry   = 3600 (1H)
              expire  = 1728000 (2w6d)
              minimum ttl = 7200 (2H)
```

```
      mail.giacenterprises.com     internet address = NNN.NNN.NNN.13
      mail.giacenterprises.com     internet address = NNN.NNN.NNN.14
      dns1.giacenterprises.com     internet address = NNN.NNN.NNN.11
      dns2.giacenterprises.com     internet address = NNN.NNN.NNN.12
```

Note that the phone numbers of the contacts returned via whois probably indicate that a PABX/PBX is in use (issue: wardialing to discover modems).

Email and News messages from GIAC Enterprises employees may show evidence of server names and IP addresses in the headers. If available, other sites' web server logs may show evidence of GIAC Enterprises internal network, for example, whether their Internet access is proxy-based or NATed.

Email will also show attackers what MUAs the employees use, which will usually allow them to deduce the OSes in use on the GIAC network. Finding a known address and then tracerouting address ranges around it, will tell an attacker what GIAC's network address space is — and what it isn't — by seeing when the traceroute starts going down other paths.

Search engines will enable an attacker to look up '@giacenterprises.com'. This will only return documents containing GIAC email addresses, typically from employees to public mailing lists etc. Further searching on that employee's Email address plus the string 'x-mailer' will only return matches containing full transcriptions of mail messages, including all headers. From the X-Mailer header the MUA can be discerned, and the Received headers indicate the path of SMTP servers a message has to take to leave the GIAC network.

The following is an example of the sort of email message that can be found via a search engine.

```
Received: from exchange.giacenterprises.com (10.2.2.5)
        by av.giacenterprises.com with SMTP; 3 Nov 2000 09:50:38 -0000
 Received: from xx.yy.zz ([1.2.3.4]) by
      exchange.giacenterprises.com with SMTP (Microsoft Exchange Internet
Mail Service
      Version 5.5.2650.21)
              id WQAZ7TQA; Fri, 10 Nov 2000 00:08:42 -0600
 Received: (qmail 6679 invoked by uid 1002); 3 Nov 2000 09:50:38 -0000
 Mailing-List: contact qmail-help@list.cr.yp.to; run by ezmlm
 Precedence: bulk
 Delivered-To: mailing list qmail@list.cr.yp.to
 Received: (qmail 19461 invoked from network); 3 Nov 2000 09:50:37 -0000
 Received: from massive.43545.net (HELO 43545.net)
      (qmailr@127.0.0.1)
        by xx.yy.zz with SMTP; 3 Nov 2000 09:50:37 -0000
Received: (qmail 15612 invoked by uid 1000); 3 Nov 2000 09:48:59 -0000
Date: Fri, 3 Nov 2000 10:48:59 +0100
From: A Name <peter@454545.net>
To: qmail <qmail@list.cr.yp.to>
 Subject: Re: Yahoo delivery failure - short test and proposal
 Message-ID: <20001103104859.L5048@454545.net>
 Mail-Followup-To: qmail <qmail@list.cr.yp.to>
 References: <32434.m3d7gevmqo.fsf@dffdfdsf.com>
      <20001102085716.K5048@fgfgf.net> <3A015C41.6442BFFA@fgfg.com>
      <fgfg.m3aebitv5m.fsf@fgfgf.com>
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Disposition: inline
User-Agent: Mutt/1.2.5i
```

The example shows that the message was sent from a Unix box (MUA=mutt, MTA=qmail), through a mailing-list server to a Microsoft Exchange server. Armed with this information, along with other hostnames gained via DNS lookups, any attacker can learn the IP addresses to the following GIAC Enterprises hosts:

- DNS servers (external ones only)
- SMTP servers (internal and external)
- Web servers (external only)
- IP address of Internet provider's router

Until this point, the above techniques will trigger no alerts with GIAC Enterprises, so are completely safe for an attacker. From now on the attacker can, and should, be noticed.

The attacker may attempt to increase the number of hosts to scan later by trying to dump the external DNS via a zone transfer (AXFER), for example by using the nslookup tool:

```
(
echo "server ns.giacenterprises.com"
echo "ls giacenterprises.com"
) | nslookup
```

The zone transfer should fail, as GIAC Enterprises' primary and secondary DNS servers are housed by GIAC Enterprises itself (there are no offsite secondaries), and the perimeter router will not allow TCP DNS packets through. TCP DNS AXFERs (TCP port 53) are fairly rare as accidental events, so should cause an alert, as the example syslog messages from the Cisco shows.

```
Apr 16 21:22:56 perim-fw 116927: 28w4d: %SEC-6-IPACCESSLOGP: list 106
denied tcp a.b.c.d(1247) (Serial0.1 DLCI 51) -> w.x.y.z(53), 1 packet
Apr 16 21:22:59 perim-fw 116928: 28w4d: %SEC-6-IPACCESSLOGP: list 106
denied tcp a.b.c.d(1248) (Serial0.1 DLCI 51) -> w.x.y.z(53), 1 packet
```

The attacker then portscans known addresses (or estimated network range) with a tool such as nmap (nmap -O -P0 addr1 addr3). The scan is via the standard TCP 3-way handshake, not SYN scan (nmap -sS). Only very old or broken firewalls allow SYN scans through without logging, whereas most modern firewalls and IDS systems alert on detecting SYN scans. This means using the standard method will trigger fewer alerts than 'stealth mode'. However the hosts being scanned may log them as normal connections. Any scan should return only the ports allowed by the GIAC security architecture, and all other ports should be denied by the perimeter router. Given the daily volume of simple portscans, they will be ignored by GIAC staff. Ignored doesn't mean forgotten. Portscans are still logged, and could be useful in a forensic capacity.

The following is an example of an nmap scan of the GIAC Enterprises Internet mail server. The '-O' option to nmap allows nmap to attempt to discover the OS in use.

```
 [/root]# nmap -O -P0 -F mail.giacenterprises.com
Starting nmap V. 2.54BETA7 ( www.insecure.org/nmap/ )
Interesting ports on mail.giacenterprises.com (NNN.NNN.NNN.13):
(The 1068 ports scanned but not shown below are in state: filtered)
Port       State       Service
25/tcp     open        smtp
80/tcp     closed      http
443/tcp    closed      https
```

```
1723/tcp   closed      pptp
TCP Sequence Prediction:
Class=random positive increments
Difficulty=3247184 (Good luck!)
Remote operating system guess: Linux 2.1.122 - 2.2.16

Nmap run completed -- 1 IP address (1 host up) scanned in 20 seconds
```

The three 'filtered' entries are not real. I have placed them there to show how nmap can differentiate between TCP ports that are 'closed' (no service installed on that port on the host being scanned, i.e. scan generated a TCP/IP RESET from the host), and those that are 'filtered' (scan caused a 'ICMP unreachable' or no response, typically from a 'filtering' router/firewall).

Below is an example of how the Snort IDS would notice this nmap scan. The perimeter router would block all bar the TCP ports 25,80,443 and 1723 – so only packets on those port numbers got to be 'seen' by Snort. By analysing the characteristics of the packets themselves, Snort is able to make an educated guess that these packets were generated by nmap.

```
Apr 16 21:46:23 csc snort: <eth1> SCAN nmap TCP: a.b.c.d:61909 ->
w.x.y.z:80
Apr 16 21:47:13 csc snort: spp_portscan: portscan status from a.b.c.d: 4
connections across 1 hosts: TCP(4), UDP(0) STEALTH
Apr 16 21:47:13 csc snort: <eth1> SCAN nmap fingerprint attempt:
a.b.c.d:59577 -> w.x.y.z:25
Apr 16 21:47:13 csc snort: <eth1> SCAN nmap TCP: a.b.c.d:59578 ->
w.x.y.z:25
Apr 16 21:47:13 csc snort: <eth1> SCAN nmap TCP: a.b.c.d:59580 ->
w.x.y.z:443
```

The attacker's TCP scan reveals a firewall, so typically they would not attempt a UDP portscan. This is because it can be assumed that anyone filtering TCP will also be filtering UDP. UDP services don't 'handshake', so it is impossible to distinguish between an open UDP port and one blocked-and-dropped (i.e. no ICMP unreachable generated). This feature of UDP makes portscans extremely lengthy and unreliable. Only direct scanning for applications is worth doing.

The following is an example of scanning localhost for an operational DNS server:

```
[/root]# nmap -P0 -p 53 -sU localhost

Starting nmap V. 2.54BETA7 ( www.insecure.org/nmap/ )
Interesting ports on localhost (127.0.0.1):
Port       State      Service
53/udp     open       domain

Nmap run completed -- 1 IP address (1 host up) scanned in 12 seconds
```

The following is what tcpdump showed really happened.

```
/root]# tcpdump -i lo -n
Kernel filter, protocol ALL, datagram packet socket
tcpdump: listening on lo
08:34:52.052298 > 127.0.0.1.56188 > 127.0.0.1.domain: 0 [0q] (0)
08:34:52.052298 < 127.0.0.1.56188 > 127.0.0.1.domain: 0 [0q] (0)
08:34:58.062778 > 127.0.0.1.56189 > 127.0.0.1.domain: 0 [0q] (0)
08:34:58.062778 < 127.0.0.1.56189 > 127.0.0.1.domain: 0 [0q] (0)
```

As can be seen, the DNS server does not actually send any traffic back to nmap. This is because UDP

packets are generated by the application itself. As nmap doesn't send valid DNS packets, the DNS server doesn't respond.

Here's an example of scanning localhost on a closed UDP port

```
root]# nmap -P0 -p 54 -sU localhost

Starting nmap V. 2.54BETA7 ( www.insecure.org/nmap/ )
The 1 scanned port on localhost (127.0.0.1) is: closed
Nmap run completed -- 1 IP address (1 host up) scanned in 1 second
```

As before, the following is what tcpdump saw

```
08:35:11.011287 > 127.0.0.1.36342 > 127.0.0.1.54: udp 0
08:35:11.011287 < 127.0.0.1.36342 > 127.0.0.1.54: udp 0
08:35:11.011443 > 127.0.0.1 > 127.0.0.1: icmp: 127.0.0.1 udp port 54
unreachable (DF) [tos 0xc0]
08:35:11.011443 < 127.0.0.1 > 127.0.0.1: icmp: 127.0.0.1 udp port 54
unreachable (DF) [tos 0xc0]
```

In this case it is the OS that informs nmap that the port number is unreachable. Oddly enough, UDP port scans assume ports are up when they get no response, and down when they do! This is exactly the opposite of how TCP scans operate.

Although tools such as nmap are invaluable at mapping networks, they are definitely manual tools. To script scans, auditors will find the tool ndiff useful. NDIFF is a 'wrapper' around nmap that allows for benchmark scans to be compared against scans done at a later date and the differences to be shown. It reports not only what hosts have been added/removed between scans, but also changes in services. It is extremely good for showing how a current audit compares to what was last recorded. It should not only be used for audits, but should be installed on all Security Consoles and run regularly from there. There is direct HTML output support so that the entire process can be run out of cron and outputted as web pages. Below is an example of this.

New Hosts
Missing Hosts
Changed Hosts

New Hosts
[previously un-observed hosts found in the more recent scan
-- all interesting ports shown]

1.2.3.9

Port

Protocol

Service Name

State

25

tcp

smtp

open

53

udp

dns

open

Figure 9: Example of ndiff HTML output

3.2.2 Vulnerability Scanner

To gain information about what applications are running on each open port, auditors should use Nessus, 'a free, powerful, up-to-date and easy to use remote security scanner'.Sara or Saint are other alternatives, but I've personally found less false positives with Nessus. In fact, the previous nmap portscans can be dropped in favour of just using Nessus, as it uses nmap to scan for open ports anyway. Features of Nessus include:

- Can detect services running on non-standard port numbers (e.g. sendmail running on port 45 instead of 25);
- Detects security holes in applications via its continually updated security vulnerability database;
- Client-server architecture. Nessus server can be managed from a client-install on another host;
- Optimized tests. If no FTP server is found, Nessus will disable all other FTP-related tests, decreasing scan time immensely;
- Can be run in batch mode.

Typically auditors should run Nessus manually first, with all the options appropriate for the GIAC network. Upon exiting, Nessus 'remembers' those settings for next time and subsequently can be run in command-line mode (could be from a cronjob too):

```
nessus --batch-mode -T html_graph localhost 1421 username hosts-to-scan.txt
nessus-report.html
```

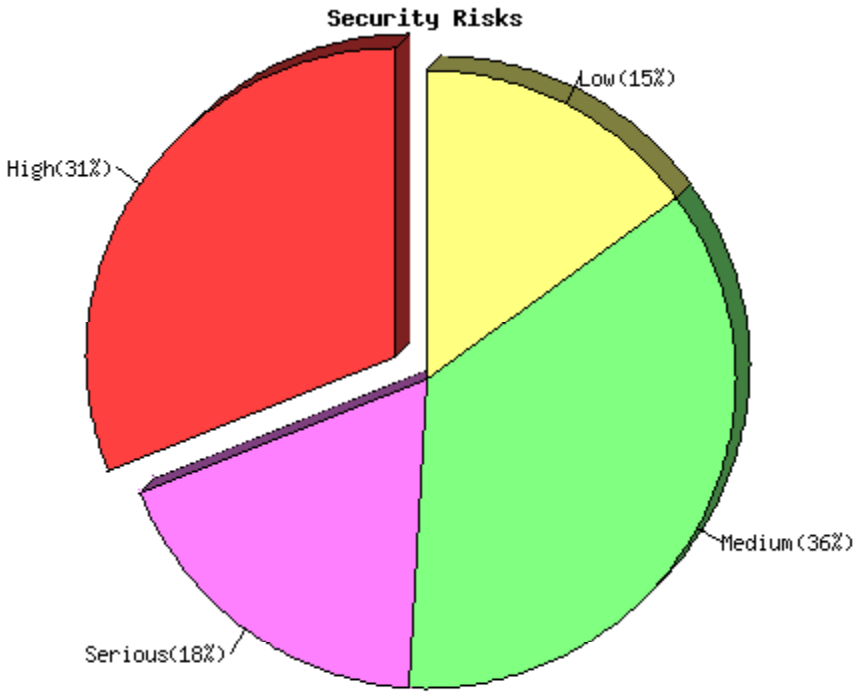
This will scan the hosts listed in the file **hosts-to-scan.txt** and will generate a report in **nessus-report.html**. The report lists all the ports that are open, what service is associated with each one, and what possible vulnerabilities are present. As the Nessus vulnerability database is updated regularly, it can even be used to ensure the GIAC general firewall maintenance program is working and up-to-date.

It must be noted that all such scanners are notorious for generating false positives. After the GIAC audit is completed, all entries should be compared against the GIAC systems to see if the claimed security hole is actually present. As an example, Nessus has always misreported a Qmail SMTP server as potentially having a whole list of Sendmail problems, none of which actually affect Qmail.

What follows is an example of a Nessus scan of a Linux host. It has been edited for brevity, but shows the great range of detail reported.

www.giacenterprises.com

Repartition of the level of the security problems :



[Back to the index]

List of open ports :

www (80/tcp) (Security hole found)

https (443/tcp) (Security hole found)

Vulnerability found on port www (80/tcp)

Requesting an URL with '?wp-cs-dump' at its end
makes some Netscape servers dump the listing of the page
directory, thus showing potentially sensitive files.

Risk factor : Medium/High.

Solution : disable the 'web publishing' feature of your server
CVE : CAN-2000-0236

[back to the list of ports]

Vulnerability found on port www (80/tcp)

Requesting an URL with '?PageServices' at its end
makes some Netscape servers dump the listing of the page
directory, thus showing potentially sensitive files.

Figure 10: Nessus batch-mode HTML output

Nessus contains detailed explanations of the faults it find, along with solutions (very useful). Another useful resource for learning details of exploits is the CVE database (Common Vulnerabilities and Exploits) <URL: http://cve.mitre.org/>, which is home to a vast database of known attack signatures. The database not only contains details of attacks, but also references the fixes as they become available. The database is OS—and vendor—independent and everyone working in network security should be aware of its existence.

3.2.3 Protected Network Audits

The next task is to audit each Internet subnet. This enables auditors to emulate the situation of an attacker gaining remote access to some portion of the GIAC Enterprises network. The audit should demonstrate how the security policy design minimizes the impact of such an event, how the attacker is limited in extending their attack, and how quickly an attacker would be noticed.

Each Internet subnet consists of a Subnet Security Console (SSC), a (hidden) IDS, and hosts appropriate to the particular subnet. e.g. Web servers for the WebNet network, SMTP and DNS servers for the ServiceNET and MySQL servers for the SQLNet network.

On each host, auditors should use ps, netstat and lsof to report on what software is running, and on what ports. What follows is an example of how the Apache web server (typically installed as **/usr/sbin/httpd**) looks to the ps and lsof command. As expected, it binds to TCP ports 80 and 443.

```
[/tmp]# ps auxww |grep /usr/sbin/httpd|grep -v grep
root      2267  0.0  2.6  7584 3352 ?   S    Apr11   0:00 /usr/sbin/httpd
www      29498  0.0  2.7  7668 3428 ?   S    Apr15   0:00 /usr/sbin/httpd
www      29499  0.0  2.7  7668 3428 ?   S    Apr15   0:00 /usr/sbin/httpd
www      29500  0.0  2.7  7668 3428 ?   S    Apr15   0:00 /usr/sbin/httpd
www      29501  0.0  2.7  7668 3428 ?   S    Apr15   0:00 /usr/sbin/httpd
www      29502  0.0  2.7  7668 3428 ?   S    Apr15   0:00 /usr/sbin/httpd
[/tmp]# lsof -i |grep httpd
httpd      2267 root   16u  IPv4   9412       TCP *:https (LISTEN)
httpd      2267 root   17u  IPv4   9413       TCP *:www (LISTEN)
httpd     29498 root   16u  IPv4   9412       TCP *:https (LISTEN)
httpd     29498 root   17u  IPv4   9413       TCP *:www (LISTEN)
httpd     29499 root   16u  IPv4   9412       TCP *:https (LISTEN)
httpd     29499 root   17u  IPv4   9413       TCP *:www (LISTEN)
httpd     29500 root   16u  IPv4   9412       TCP *:https (LISTEN)
httpd     29500 root   17u  IPv4   9413       TCP *:www (LISTEN)
```

Auditors should also check the cronjobs to see what automated jobs are in place. These should be compared against the last audit to see if anything has changed. Anything different should be accounted for. On top of this, a file auditor like Tripwire should be installed to ensure the integrity of the OS and applications.

After the host audits, the Subnet Security Console (SSC) should be used to initiate scans against each of the other hosts in that particular subnet. This will provide confirmation that the services installed match with those viewable from the network. Any unexpected mismatch may indicate a system has been compromised and trojans installed that 'hide' themselves from standard system apps. It should be noted that the OpenSSH port on all hosts is only accessible from the CSC, so such scans should report it as

been filtered, but the system would show sshd running on port 22.

```
]# ps aux|grep sshd
root       736  0.0  0.3  2332  504 ?        S    08:15   0:00 sshd
root      3323  0.0  0.4  1364  528 pts/1    S    09:58   0:00 grep sshd
[/root]# netstat -an|grep :22
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
[/root]# telnet localhost 22
Trying 127.0.0.1...
telnet: Unable to connect to remote host: Connection timed out
```

The timeout above demonstrates that the system (via ipchains) is DENYing packets—not rejecting them (via RESETs or ICMP unreachables). This complies with the security policy in that it minimises the information available to attackers.

There is a good reason for auditors to use the SSC as their scanning host. SSC's only run OpenSSH, arpwatch, snmptrapd, and syslog servers. The SNMP and Syslog services are only accessible from addresses on the same subnet, and the OpenSSH service can only be accessed from the Central Security Console. This is enforced at both an application level (if possible) and via ipchains. There are no ports on SSC's that are accessible directly from the Internet, so an attacker should have an infinitesimal chance of breaking into them directly. As such these hosts may be classified as being the strongest hosts, that is, the most resistant to attack.

Finally, auditors should attempt to access services in other subnets, as well as the GIAC Enterprise LAN itself. The security policy states that all hosts in the Internet subnets need to be able to access DNS, NTP and SMTP servers in the ServiceNET subnet, that the Web servers need to be able to access the MySQL service in the SQLNet subnet, and that the ServiceNET SMTP servers can access the LAN SMTP servers. Other than that, no other access is permitted.

What follows is an example of scanning a Web server from a ServiceNET SMTP server:

```
/root]# nmap -P0  www

Starting nmap V. 2.54BETA7 ( www.insecure.org/nmap/ )
Interesting ports on www.giacenterprises.com (w.x.y.z):
(The 1519 ports scanned but not shown below are in state: filtered)
Port       State       Service
80/tcp     open        http
98/tcp     open        linuxconf
443/tcp     open         https

Nmap run completed -- 1 IP address (1 host up) scanned in 3 second
```

The results show how the security policy is working. All unsupported ports cause the packets to be dropped. The IDS servers can be used to see that the firewall is dropping the packets instead of ipchain rules. Auditors should simply bring up tcpdump on the IDSs in the WebNet and ServiceNET networks, and do the scan. If the firewall is blocking the packets, the ServiceNET IDS will see the packets going out, and no reply occurring, whereas the WebNet IDS won't see even one packet. Any other result implies a problem.

3.3 Analysis

After the audit is complete, an assessment needs to be done to see if any problems were found or

whether any improvements can be made. The Nessus scan reported the following:

Vulnerability Discovered	**Corrective Action**
CVE-2000-0236 (Netscape CGI issue)	False positive: running Apache
CVE-1999-0269 (Netscape Pageservices)	False positive: running Apache
Detected Web server and OS used	Add 'ServerTokens ProductOnly' to Apache config.
Ndiff shows linuxconf network interface installed on Web server	Mistake during install of new Web server. Network interface to linuxconf removed.

Postscript

Since completing this assignment, an enhancement to the security architecture has occurred to me. Internet access from the GIAC Enterprises LAN flows outwards via NAT from the perimeter router. This leads to NAT being enabled in the perimeter network, and has some security implications regarding any holes found in the NAT code in the future (holes have been found before in Cisco's NAT implementation). A better approach would be to use the second Internet link dedicated to GIAC remote staff for outgoing Internet access as well as incoming. The primary advantage is that this link is unknown to the world. Potential attackers know of GIAC Enterprises perimeter network because everyone needs access for the Web servers, etc. The VPN-only Internet link is via a different link altogether, and no-one except GIAC staff need know about it. This 'security through obscurity' approach should not be relied on as a security measure, but it would certainly not do any harm.

Assignment 4: Design Under Fire (25 Points)

The purpose of this exercise is to help you think about threats to your network and therefore develop a more robust design. Keep in mind that the next certification group will be attacking your architecture!

Select a network design from any previously posted GCFW practical (http://www.sans.org/giactc/gcfw.htm) and paste the graphic into your submission. Be certain to list the URL of the practical you are using. Design the following three attacks against the architecture:

An attack against the firewall itself. Research vulnerabilities that have been found for the type of firewall chosen for the design. Choose an attack and explain the results of running that attack against the firewall.

A denial of service attack. Subject the design to a theoretical attack from 50 compromised cable modem/DSL systems using TCP SYN, UDP, or ICMP floods. Describe the countermeasures that can be put into place to mitigate the attack that you chose.

An attack plan to compromise an internal system through the perimeter system. Select a target, explain your reasons for choosing that target, and describe the process to compromise the target.

Note: this is the second time this assignment has been used. The first time, a number of students came up with magical "hand-waving" attacks. You must supply documentation (preferably a URL) for any vulnerability you use in your attack, and the exploit code that you use to accomplish the attack. The purpose of this exercise is for the student to clearly demonstrate they understand that firewall and perimeter systems are not magic "silver bullets" immune to all attacks.

4.1 The Defendant

I have chosen to 'attack' the network design by Keith Gardiner (submitted Sept 2000).

<URL: http://www.sans.org/y2k/practical/Keith_Gardiner_gcfw.doc>

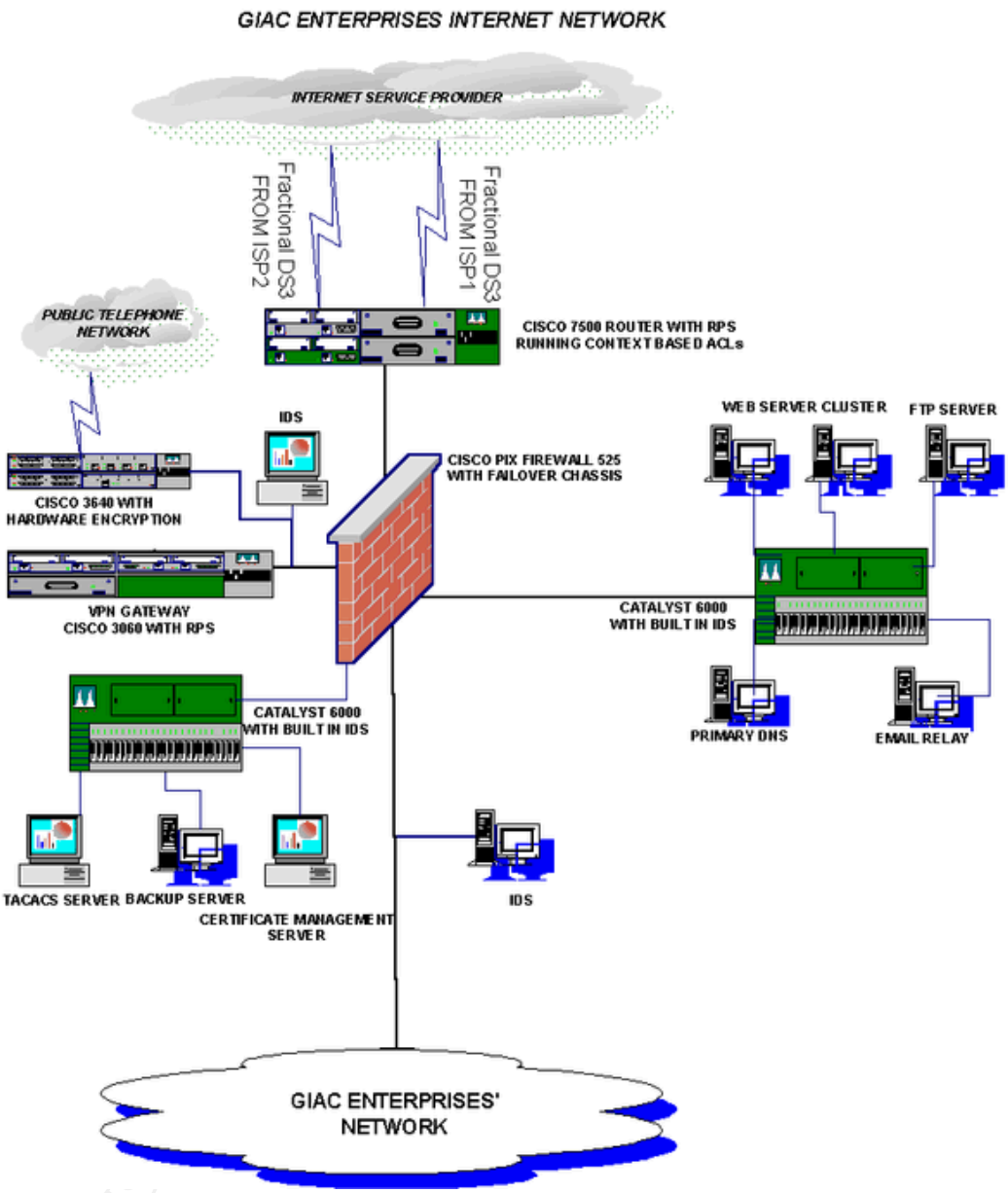


Figure 11 Keith Gardiner's Network Design

4.2 Attack on Firewall

Keith's perimeter area consists of a Cisco 7500 perimeter router joining to a PIX firewall. Version numbers aren't given for these devices, so we'll have to assume they are the most established versions used at the time of writing (Sept 2000).

There isn't much detail given about the perimeter router, but it is said that the perimeter router allows everything by default, and simply blocks certain packet types as defined by the Visa Top 10 Security List, as well as spoofed addresses and most ICMP. No mention is made of filtering packets to the router

itself, nor what services were enabled/disabled on the router. It will be assumed that the bootp and telnet interface were disabled, but that the http and SSH services were enabled instead. SSH would indeed provide a 'secure channel' over which to manage the border router, but has no advantages over telnet with respect to securing the router itself. No mention is made of logging from the border router either, so that will be assumed to be non-existent. As such, two possibilities for compromising the border router present themselves:

- Brute force the passwords via SSH (no logging, so no risk); and
- Brute force SNMP community string.

Searching on the Web for search strings such as 'cisco IOS vulnerability' returns a few matches post Sep-2000, but all are Denial of Service type weaknesses, not actual compromises. The perimeter router doesn't directly protect any networks, so breaking into it brings little gain. The PIX makes a more useful target, as it is the device that controls access to different networks via different interfaces (one which is directly connected to the LAN).

As would be expected, finding security holes in something being sold explicitly as a firewall is even harder than finding such holes in a router. Searching sites such as www.securityfocus.com (home of the great BugTraq mailing-list) and via www.google.com brought back only three, and they were all DoS problems, not actual exploits. In the case of this PIX, even a flaw that caused packets to be allowed through some ruleset they shouldn't be would have been useful, but none could be found that were recent enough.

4.3 Denial of Service Attack

A task typically easier to accomplish than compromising a firewall is to make that firewall and/or the networks it is protecting effectively non-responsive. This can be done by either flooding the site with so much traffic that 'real' traffic cannot be responded to in a timely fashion, or exploiting known holes in the devices used such that they either crash or become non-responsive themselves. The 'trick' of course is to do this by using as few packets as possible. An attacker with a 10Mb/s Internet link can always DoS a network with only a T1 (1.5Mb/s) – the trick is to do it with only a 56Kb/s modem.

First off, from previous searches of the Internet for actual compromises, the following possible DoS flaws were found with the Cisco perimeter router and Cisco PIX firewall:

Denial of Service	Reference
Cisco "?/" DoS vulerabilty (need enable passwd)	http://www.securityfocus.com/frames/?content=/vdb/%3Fid%3D1838
Cisco "http://<router-ip>/%%" DoS	http://www.securityfocus.com/frames/?content=/vdb/%3Fid%3D1154
Cisco Secure PIX Firewall TCP Reset DoS vulnerability	http://www.securiteam.com/securitynews/Cisco_Secure_PIX_Firewall_TCP_Reset_DoS_vulnerability.html
PIX 5.1 DoS Vulnerability	http://www.securityfocus.com/templates/archive.pike?list=1&mid=174577

From the above examples of DoS attacks available, the "%%" attack is probably the one most likely to succeed. The "?/" attack requires the enable password to be applicable, and the 'PIX Firewall 5.1 DoS Vulnerability' requires AAA authentication to be used and enabled on the external interface in order to be exploitable, which probably isn't the case. The 'TCP Reset' DoS is more interesting, as this flaw

allows for the internal hosts TCP sessions to be RESET by a remote attacker. However, this involves the remote attacker knowing the source and destination IP addresses – along with their port numbers – in order to spoof an appropriate TCP RESET. Such an attack is highly unlikely to be of much use for disrupting most TCP-based services as a brute force attack (flooding the network with TCP RESETS where the port numbers are just cycled through) would probably take longer to find a match than the average WWW connection would last. That leaves the 'http://<router-ip>/%%" attack. As no logging is done, the existence and location of the attacker won't even be found until such logging is enabled. The following attack has the potential of being effective for some time.

```
while `sleep 10`
do
      wget 'http://a.b.c.d/%%'
done
```

This simple script uses the wget HTTP tool to 'tickle' the Cisco bug every 10 seconds. This means that within 10 seconds of rebooting, the Cisco will crash again. A very effective DoS.

The more publicised DoS attacks seen on the Internet today are the 'flood' methods. If an army of PCs can be focused together as one entity, then almost any site will not be able to withstand them. Tools such as SubSeven (for Windows) and the latest batch of BIND security holes such as li0n (for Linux) can be used to create an army of Internet hosts under one attackers command. Then a tool such as Tribal Flood Network (TFN2K) can be used from those machines to launch a DoS against the network. TFN has the advantage over other DoS tools such as Trinoo in that it supports other protocols for DoS attacks besides UDP, such as ICMP. TCP isn't an option for these DoS tools as they all try to 'hide' their own addresses by generating fake source IP addresses. As TCP requires a three-way handshake in order to create a valid session, having fake addresses isn't a workable option. ICMP and UDP are stateless protocols and as such are much more useful in DoS attacks. As ICMP is blocked by the perimeter router, UDP DNS packets will be used instead. They will be allowed through the perimeter Cisco router, the Cisco PIX, and also by ipchains on the DMZ DNS servers. All three device classes have the possibility of being rendered inoperative by such an attack. Even if the router and firewall withstand the volume assault, bogus data may render the DNS servers inoperative – effectively taking the network off-air as no-one will be able to resolve DNS information anymore.

4.4 Attack Internal Systems

The hosts directly accessible from the Internet are:

- SMTP servers (Microsoft Exchange 5.5)
- DNS servers (Solaris running BIND 8.2.2)
- Web servers (Microsoft IIS)
- FTP servers (Microsoft IIS)
- VPN gateway

Breaking into the LAN via compromising either the VPN or dialup servers would be the most devastating compromise. No mention is made of dialup access in Keith's paper, but wardialing the phone address range (as discovered via WHOIS) searching for such devices (official or not) could be the simplest and most effective way of accessing the LAN.

VPN gateways are even more difficult to break into than firewalls, due to them typically ignoring all packets except IPSec. Any attack needs to be instigated via IPSec, which in this case requires both PKI certificates and some form of external authentication via RADIUS. Breaking through all that would then allow access only to a Microsoft Terminal Server—not even the LAN. This won't be even looked into as an option without having access to a (presumably stolen) pre-configured 'road warrior' laptop.

As is normally the case, the weakest link in most networks are the application servers. In this case the SMTP, DNS and Web servers. There are several remotely exploitable BIND holes affecting BIND 8.2 ('tsig bug' and 'ntx bug') which this site runs. However, finding one that works against their Solaris platform is a lot harder. All such buffer overflow attacks are architecture-dependent, and exploits for non-Intel systems are a lot rarer. The version of Microsoft Exchange being used may be susceptible to the following DoS attack:

DoS Description	**Reference**
MIME charset ="" vulnerability	http://www.securityfocus.com/vdb/?id=1869

This still isn't of much use for gaining access. Strangely enough, it seems that a lot of DoS attacks against Microsoft services look like buffer overruns, but the hacker community doesn't appear to convert those into system exploits like their Unix counterparts do – perhaps a growth area we can all look forward to…

This leaves the Microsoft IIS Web servers. These have the distinction of being one of the most exploited applications available to date (listed in SANS 'Top 10 Security Threats' - http://www.sans.org/topten.htm). In fact it would be fairer to say that most security holes will be found in Web servers today as they are the one server application where inexperienced staff tend to write applications. As almost every page contains 'server-side' extensions (e.g. ASP, mod_php/mod_perl), it is fair to say that these 'pages' are actually applications. Expecting the 'webmasters' of every organization to be able to write secure code for their own Web pages appears to be too much to ask for, especially when that is layered on top of Web interfaces such as IIS that give such open access to almost all aspects of the underlying OS.

Whenever attempting to 'crack' a Web server which is within a firewalled environment, it would be prudent to make use of the HTTPS interface if one is available. Doing so totally removes the opportunity for the attack to be even noticed by any IDS system. An IDS system cannot do signature matching on packets when every packet is encrypted. Tools such as Nessus and whisker can be used to discover any IIS vulnerabilities present, and then the exploit can be downloaded from the Web for actual execution. As the exploits and whisker itself don't normally support HTTPS directly themselves, their efforts should be tunnelled through the likes of sslproxy, which allows a system to be set up whereby a local HTTP connection is made to a localhost port, and sslproxy then converts that into a SSL tunnel to the remote HTTPS server.

The following use of sslproxy shows how it can be used to pipe standard HTTP calls on localhost port 4001 through to a remote HTTPS Web server.

```
sslproxy -l 4001  -R www.giacenterprises.com -r 443
```

An attacker can take extra precautions to minimize the opportunity of being noticed by the system they are attacking. Simply using local IP filtering (e.g. ipchains under Linux) can stop the attacker box from sending anything to the victim network besides HTTPS packets. Then running large general-purpose

tools such as Nessus to discover any Web-related holes becomes almost guaranteed to slip past any IDS system. It cannot be stressed enough that normal Web logs *must* be monitored by security staff.

Through the Nessus scan it is discovered that the remote IIS server suffers from the UNICODE vulnerability reported in November 2000. IIS fails to check UNICODE versions of strings the same way it checks ASCII strings, and as such allows attackers to gain access to local resources. Using tools such as unicodexecute.pl or iis-zang, it quickly becomes apparent the system is vulnerable:

```
[/root]$ ./unicodexecute.pl www.giacenterprises.com:80 'dir '
Sensepost.exe found - Executing [dir ] on a.b.c.s:80
HTTP/1.1 200 OK
Server: Microsoft-IIS/4.0
Date: Sun, 15 Apr 2001 23:06:16 GMT
Content-Type: application/octet-stream
Volume in drive C has no label.
Volume Serial Number is D86F-6805

 Directory of C:\Inetpub\scripts

04/16/01  10:55a        <DIR>          .
04/16/01  10:55a        <DIR>          ..
11/18/99  10:04a               208,144 sensepost.exe
               3 File(s)        208,144 bytes
                         1,007,892,480 bytes free
[root]$
```

A truly committed attacker would then attempt to discover security holes in these Web sites manually, looking for holes introduced by the staff that wrote them. However, that falls out of the range of this assignment, as too much would have to be assumed. Nonetheless, the security holes found by Nessus and whisker are merely examples of the same general problem. In their case it's just that these 'Web pages' were written once and then used by the general Internet community instead of being developed internally. Common programming flaws such as lack of variables checking can lead to compromise with anyone's work.

Bibliography

Extreme Switches – http://www.extremenetworks.com/

MRTG – http:// ee-staff.ethz.ch/~oetiker/webtools/mrtg/mrtg.html

Snort IDS system – http://www.snort.org/

ACID Snort Analyser – http:// www.cert.org/kb/aircert/

MySQL SQL Database server – http://www.mysql.com/

Logsnorter (inserts firewall logs into Snort database) - http://www.snort.org/Files/logsnorter.tar.gz

Linux – http://www.linux.org/

PIX Cisco Firewall – http://www.cisco.com/

Cisco Routers – http://www.cisco.com/

Ettercap – allows sniffing on switched networks - http://ettercap.sourceforge.net/

Qmail SMTP MTA – http://www.qmail.org/

Apache Web server – http://www.apache.org/

XNTP Time Server – http://www.ntp.org/

OpenSSH Secure Shell – http://www.openssh.com/

djbdns Secure DNS Server – http://www.djbdns.org/

xinetd INETD server – http://www.xinetd.org/

vsftpd 'Very Secure' FTP server - ftp://ferret.lmh.ox.ac.uk/pub/linux/

stunnel SSL TCP/IP wrapper – http://www.stunnel.org/

Squid Web Proxy server – http://www.squid.org/

SOCKS Server - http://www.inet.no/dante/

Sun Microsystems - http://www.sun.com/

Cisco VPN 3000 series – http://www.cisco.com/

RSA SecurID - http://www.rsasecurity.com/

Qmail-Scanner Anti-Virus SMTP harness – http://qmail-scanner.sourceforge.net/

nmap port scanner – http://www.insecure.org/nmap/

ndiff Nmap 'diff' tool - http://www.vinecorp.com/ndiff/

Nessus Security Scanner – http://www.nessus.org/

Sara Security Scanner - http://www-arc.com/sara/

Brutus HTTP Password Cracker – http://www.hoobie.net/brutus/

Analog Web Log Analyzer - http://www.statslab.cam.ac.uk/~sret1/analog/

Tribal Flood Network -http://mixter.warrior2k.com/

SubSeven - http://www.sans.org/newlook/resources/IDFAQ/subseven.htm

Whisker Web vulnerability scanner http://www.wiretrip.net/rfp/

Whisker + SSL support - http://www.nightbirdfr.com/outils/whisker-1.4+SSL.tar.gz

Sslproxy Reverse SSL proxy - http://www.obdev.at/products/ssl-proxy/index.html

Unicodexecute.pl and iis-zang - http://darknet.hq.alert.sk/exploits/daemon/iis/