Global Information Assurance Certification Paper

Guarding the Fortune

SANS GIAC Certified Firewall Analyst
Practical Assignment Version 2.0

By

Tim Lewis

29 December 2003

Assignment 1 – Security Architecture for GIAC Enterprises

Abstract

GIAC Enterprises has been producing fortune cookies and fortune cookie sayings for more than 10 years. Over the past several years a shortage in cookie making supplies and an increase in the cost of unionized labor (cookie line production) have forced GIAC to re-evaluate their business plan. They have decided to sell off their cookie making division and focus on the resale of their industry renowned cookie fortunes. The cookie fortune division has always maintained high profit margins. GIAC's board of directors believes that GIAC's survival depends upon the growth of this division.

With the decision to take the company in this direction, GIAC has been transformed into an e-commerce business and as a result have a greater need for a more stringent network security architecture. As a business that now relies on the Internet as their primary mode of business, it has become critical that their data and the systems responsible for distributing that data are protected.

GIAC has commissioned me to design their network and security architecture to meet these new demands. GIAC is not a large company. Worldwide they have less than 100 employees. Financial means are not unlimited either. While we will be designing GIAC a new security architecture, it will be important for us to consider this along with the business requirements.

The objective of this document will be to provide GIAC with:

1. A Security Architecture based on GIAC's technical and business requirements.
2. The security policy for the primary firewall and Internet border routers in this architecture, and additionally a tutorial on how to implement the security policy for the Internet Border router.
3. A validation of the Primary Firewall Security Policy.
4. An example scenario of potential attacks on an alternative Security Architecture.

Summary of Access requirements

We spent considerable time discussing and developing an understanding of how GIAC's business operations need to take place. Some of these operations currently take place in an existing architecture. Other operations need to be implemented in order for GIAC to stay competitive.

As the outcome of discussions centered on GIAC's business requirements, we have identified six classes of access requirements.

In general our access policy will be to implicitly deny all communication and allow only what is required. Far too often we have run across scenarios where inbound (from the Internet) network traffic was denied, but all traffic originating from a companies internal network was allowed out (to the Internet), or even allowed to other networks within the company (for example a screened subnet/dmz network). Obviously, no business functions can take place with all communication denied, thus in the following sections we will discuss what communication is required for GIAC's business to operate. We will discuss which services, protocols and applications are used when GIAC conducts business.

Customers (Companies or individuals that purchase bulk online fortunes)

GIAC customers use GIAC's proprietary e-Cookie™ system. e-Cookie™ is a web based application that allows customers to tailor custom made orders of fortunes. In the current architecture, e-Cookie™ runs over http. GIAC will be implementing SSL Certificates and require customers to access e-Cookie™ via HTTPS. The implementation of SSL will not be part of this project, other than to be included in our security policies. For compatibility we will continue to allow http for the initial requests, thereafter the requestor will be redirected to use https. Currently, due to performance limitation of the existing system, e-Cookie™ orders are batched and processed by the database server during off-hours. The order (in the format of an .XLS or .CSV file) is then attached to an e-mail message and sent to the customer. Loyal customers have been threatening to take their business elsewhere if GIAC does not start to provide real-time order fulfillment. The use of HTTP in the clear has resulted in stolen fortunes and the use of e-mail for order delivery has often resulted in non-delivered messages. Customers are unhappy.

GIAC's fortunes are so popular that it lead to the development of the *GIAC Inside*™ Database. This is a database of stores, restaurants, and online retailers that carry *GIAC Inside*™ premier branded cookies. The database is updated monthly. Rather than establish complex database links to customer databases, part of GIAC's premier customer contractual agreement stipulates that they will supply a delimited file of *GIAC Inside*™ customers each month. This file will be transferred to GIAC via FTP. To keep the transaction secure and confidential, Premier customers will connect through a site-to-site VPN for this transfer.

As an added perk to being a Premier customer, they not only have access to the e-Cookie™ system, they will also have access to the Fortune-in-Sales™ Sales Management system. FIS is an industry renowned analytical business application developed by GIAC which has shown demonstrable results for GIAC's sales revenue since it was launched. Premier customers will access FIS

over the VPN connection as well. Where not governed by export regulations, this link will be 3DES encrypted. Otherwise DES encryption will be used.

Suppliers (Companies that supply GIAC Enterprises with their fortune cookie sayings)

GIAC's suppliers are a unique group. The suppliers are usually individuals or small think tanks of Wise Persons. These are the people that think up, discover or articulate 99% of the world's fortunes. GIAC would often recruit their suppliers by placing advertisements in the back of magazines such as Reader's Digest and Popular Science. Initially these suppliers would mail their fortunes to GIAC and GIAC's data entry department would enter them into GIAC's dbFortune™ Database System. However, with the rise in popularity and widespread use of the Internet, GIAC's suppliers have increasingly sought an easier method for transferring their files. Freelance (individual) Wise Persons want to be able to enter their fortunes individually into GIAC's system via a web based application. GIAC's competitors currently have such systems in place. The Think-Tank organizations have already changed their way of doing business and provide secure downloads (using HTTPS and SSL 128 Bit) from their websites. There is nothing preventing GIAC from receiving their fortunes this way, but a review of the operations revealed that they were still paying a premium to have these fortunes mailed to them. Moving forward, GIAC procurement specialists will require HTTPS to the Internet such that they may download from the Think-Tank Suppliers. GIAC freelance suppliers will require HTTPS access to 'suppliers.giac.com' such that they can enter their fortunes online. There are still some countries were Internet access is not readily available and so GIAC will continue to maintain their Data Entry Department to handle mail-in suppliers.

Partners (International companies that translate and resell fortunes)

GIAC's partners provide a value-added service that allows GIAC to deliver their product to countries in which there is a language barrier. In order to do this with precision, partners need the ability to query the dbFortune™ database, with which they subsequently provide translation services. The dbFortune™ database server is a Sybase Server and the database listens on TCP port 6500 Partners also need to collaborate with GIAC's research staff so that quality assurance is maintained across cultural and language barriers. Lotus Domino Server was originally installed to run a custom collaboration application and database. Both GIAC's research staff and partners need access to the Domino Server. Both of these requirements necessitate the need for a VPN connection between GIAC and its partners to ensure data confidentiality.

GIAC Enterprises Employees located on GIAC Enterprise's internal Network.

GIAC Enterprises is a small organization with one immediate location. The company is divided into departmental based workgroups. Each of these workgroups may or may not have their own systems (Servers) and may or may not require a certain level of access to GIAC's production systems (located on the Screened VPN and Screened Public subnets), and all of them require some level of Internet access. There is also a common subnet containing systems that all internal staff may need access to. These systems include general File and Print Services, Anti-Virus distribution Server, E-mail Servers, and Network Services Servers (DHCP, DNS, and WINS). GIAC's internal workgroup and common servers are all Microsoft Windows Based Systems running a mixture of Windows NT 4.0 and Windows 2000 Server. GIAC's email platform is Lotus Domino Server Release 6.x All internal employees access the Domino server using the Notes Client using the Lotus Notes specific transport which runs on TCP Port 1352. An Anti-Virus distribution server is provided, using Network Associates (NAI) ePolicy Server for Mcafee Anti-Virus software. All internal systems are configured to query the NAI ePolicy server for updates via HTTP. The Internal Mail server is Lotus Domino and will be configured to route mail to the external mail server (which is also Domino 6.x) via TCP Port 1352. For the sake of this discussion, any internal or workgroup related systems will be assumed to be secured and configured properly, including the maintenance of patches and fixes.

A Cisco 2948G-L3 Layer 3 Switch is used for GIAC's internal, inter-VLAN routing. This switch is configured with a basic set of ACL's to provide a level of isolation between VLANS. This device helps us to adhere to the principals of defense-in-depth by allowing us to provide a basic level of filtering where a full featured firewall might not be economically feasible to do so. The primary concern in performing this isolation of VLANS in not the to deter the determined criminal or even the mischievous employee (though that is a concern relating to the Finance Department). The primary concern is due to the number of computer worms and viruses that have been seen over the past 2-3 years which, upon infecting a system, usually a Windows system, begin propagating by evaluating the local IP address of the infected system and begin network scans based on that IP subnet. Isolating the various workgroup through basic ACL based filtering can often help prevent the spread of infections throughout the entire organization. While keeping systems patched and up to date is an even better prevention method, let's be realistic: GIAC's IT staff is already wearing several hats and is likely stretched pretty thing – to put is simply, it may not be realistic to update and patch every desktop and server in the amount of time between the public disclosure of a vulnerability and the time someone determines to exploit it. It may not be realistic at least until it happens at GIAC a few times! While there are inherent security flaws in standard VLAN configurations [1], configuring our internal switched network (including the layer 3 switch) to guard against such weakness, was not considered a priority at the time.

The following are the departments / groups located at the GIAC facility.

Research Department

The Research Department is tasked with verifying the fortunes which GIAC sells. They make sure fortunes are properly credited to the sage from which they originate. This department also engages in much collaboration with similarly tasked departments in GIAC partners. At some point in the past they purchased a custom built collaboration application built with Lotus Notes. There is a Domino Server in GIAC's Screened VPN network which hosts this application. For the sake of this engagement, it is assumed that the Domino servers within GIAC's organization are configured and secure properly.

Procurement Department

Procurement is responsible for procuring fortunes from GIAC's suppliers. This department downloads bulk fortunes from the aforementioned Think-Tank organizations. They also interface with GIAC's independent suppliers and coordinate the submission of those fortunes, which are subsequently delivered to the Data Entry department.

Data Entry Department

The Data Entry Department's primary responsibility is to enter process and enter fortunes into GIAC's dbFortune Database system. Quality and Product Assurance functions also fall under their responsibilities.

Information Services and Technology Department

This department loosely consists of developers, database administrators (DBA's), Systems Administrators, and IT Support Staff. Because GIAC is a relatively small company, many of these functions and roles often overlap. This department needs the most access to resources overall and yet requires more security for their local and workgroup access due to the sensitive nature of source code and their inherent levels of access to other systems.

Finance Department

The Finance department is responsible for all financial related company business including accounting, billing and payroll. Obviously their workgroup systems require a secure separation from the rest of the company

Executive Administration and Sales Organizations

These groups include GIAC's sales force and executive management and corporate administration.

With the exception of the Finance Department, each of these groups needs access to the systems on both screened subnets.

Each group also requires outbound HTTP, HTTPS, and FTP access to the Internet.

GIAC Enterprises mobile sales force and teleworkers.

GIAC's mobile sales force and teleworkers all need inbound access to email, and the systems on the screened VPN subnet. Since our VPN gateway is going to be Checkpoint Firewall-1/VPN-1, we will provide access for these users via Checkpoint's SecureClient remote access VPN software. SecureClient is not free, unlike SecuRemote, but GIAC decided that the built-in firewall inherent in SecureClient was a worthwhile investment. They will be able administer the security policy of these desktops from their Checkpoint management server. Checkpoint's VPN client also provides for relatively easy administration since all network topology information is automatically downloaded by the client one initially configured. This will allow use to make changes and expand GIAC's network (should future demand drive such changes), and not have to revisit all mobile workers to alter configurations. We will divide VPN access into various different groups based on access requirements.

Currently we will only define two VPN User groups, Sales and IT Support.

- All VPN Users require access to the General Purpose subnet where e-mail and other services reside.
- Sales Force VPN Users will additionally require access to the 'Fortune-in-Sales' Management system (on the Screened VPN subnet).
- IT Support needs access to all networks, for both the standard services (http, https, database), but also for remote access and control of systems for support purposes.
- Should we decided to add other functional groups, such as VPN Finance Users, for instance, we will easily be able to allow them to only the Finance systems.

The general public

The general public has access to www.giac.com AKA e-fortune.giac.com. This public information web site includes press releases, investor information and contact information. On the web site can also be found, links to customer web sites (fortune cookie producers.), the *Fortune of the Day*™ e-mail subscription service, and a database of where you can find *GIAC Inside* branded fortune Cookies. E-fortune.giac.com will link to the dbfortune.giac.com database server to provide this data.

To summarize these access groups and what protocols and services they require, consider the following tables.

Summary of Access Requirements

Source	Destination	Service
Customers	e-Cookie	https, http
Premier Customers	Ftp (via VPN)	ftp (via VPN)
Premier Customers	Fortune-in-Sales DB (via VPN)	https
Suppliers	Suppliers.giac.com	https
Partners	DMZ Domino Notes Replication	1352 tcp
Partners	DbFortune.giac.com	6500 (custom Sql Port)
Internal Employees	Corp. DMZ	1352, 443,
		6500, RDP, VNC, https, ftp
GIAC Mobile	Any	Require VPN (email, https, ftp, db)
General Public	www, email	HTTP, SMTP
GIAC Procurement Specialists	Suppliers	Https
e-fortune.giac.com	Dbfortune	6500
e-cookie.giac.com	Dbfortune	6500
Internal Mail system	MAIL Relay	1352
Internal DNS System	Hosted DNS	Domain-Udp,
Internal DNS System	Provider DNS Any DNS	Domain-udp, forwarding
VPN Clients	Internal networks	Various (1352, http, https, db, domain-udp,ftp, nbt, rdp, VNC)

GIAC will use a third party DNS provider to handle its public DNS. DNS is often exploited, given our relatively small infrastructure it is one component where we believe it is best left to a company that specializes in that service. We came to this determination after evaluating numerous factors including technical resources, exposure, and cost.

Defining access requirements

After determining what the business requirements were, we can now move on to defining the Security Architecture. For the sake of clarity it must be understood that this document focuses on the Network Architecture. This is a subset of a larger project for GIAC that is beyond the scope of this document. GIAC will be implementing new servers for their WWW and Database operations. We will define where those servers are placed and how the access requirements

are implemented but we will not focus on the configuration of those systems. GIAC has employed competent system and database administrators to handle that part of the project. For the sake of this document, it is assumed that those systems will be installed according the recommended security guidelines for the given operating systems and will be kept current with patches and service packs.

While evaluating the need and options for this new network security architecture, GIAC has put different weights of importance on factors such as confidentiality, data integrity, redundancy, and availability.

Architecture Overview

We have chosen to implement the architecture as follows. Details on the relevant components will follow. A single Cisco 3640 Series Router will connect GIAC to the public Internet through two separate ISP connections. This router will run Border Gateway Routing Protocol (BGP) so that the ISP connections are redundant. We will receive the customer routing tables from each provider. Since each provider is comparable in size this will allow for an ad hoc level of load sharing. Our experience has been that provider outages, especially when using smaller carriers, would often be our largest source of downtime. BGP should help mitigate this risk. In the case of using a single router, GIAC opted to purchase 7x24x4 Onsite maintenance, or potentially keep a spare router available. There are an abundance of Cisco 3640 series routers available on the refurbished market so a spare may prove less expensive then the 7x24x4 maintenance, however initially GIAC will not have someone on staff to handle replacing the router and configuring it should it fail. We will be providing them with a tutorial on how to configure the router based upon our security architecture and configuration, so that may be less of an issue in the future. One thing that needs to be clarified: GIAC is not Amazon.com. While extended downtime would negatively affect operations, there is a reasonable amount of acceptable downtime. Much of the decisions regarding availability and redundancy are made in lieu of this factor. Provider downtime is often out of our control. We'll assume for the sake of this exercise that each provider's circuit terminates at a separate POP and uses a separate LEC for the local loop. For other components, however, GIAC intends to use a combination of cold spares and equipment contracts to achieve their desired level of availability.

GIAC's Internet LAN (referred to as the DMZ) connects to a single Checkpoint Firewall-1/VPN-1 gateway which will be running Checkpoint NG FP3. The firewall will be running on an HP Kayak workstation which runs Red Hat Linux 7.3. The underlying OS will be secured and hardened in accordance with guidelines in Checkpoint's Linux Minimum configuration reference [2], and additionally using general hardening procedures such as removing unnecessary services, as detailed in [3]. Using an existing component (the HP Kayak), and an open-source operating system will help provide a cost effective platform for the firewall software. Well use the Checkpoint Express product since our needs will

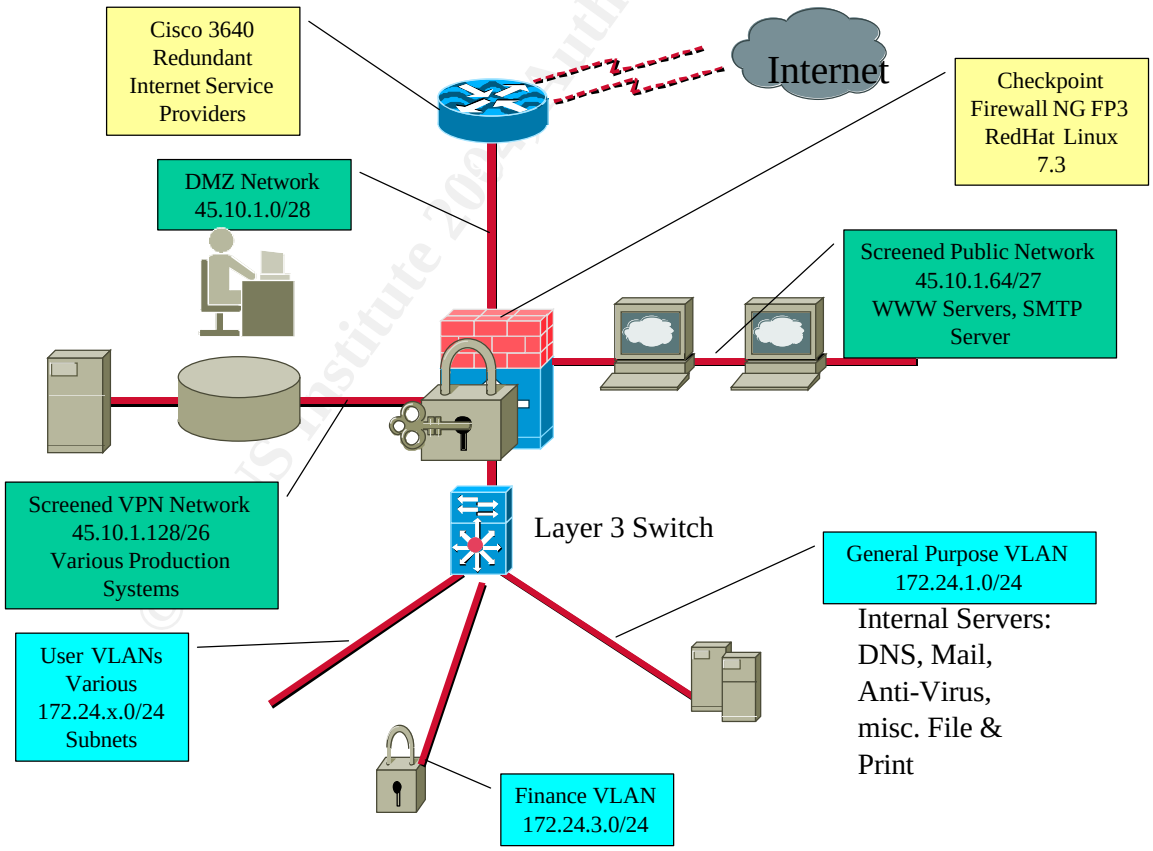
include the firewall modules, VPN module, and management server. GIAC will only require a 100 User license since they do not have that many nodes combined between their internal network and two screened subnets.

There are two publicly addressed screened subnets connected to the firewall: The screened VPN subnet and the screened Public Subnet. We will be referring to these subnets as 'Screened subnets' instead of DMZ so that we conform to the standard nomenclature [4].

The fourth connection to the firewall will be for GIAC's internal networks. The Internal networks will use private RFC 1918 based addresses and will be sub netted and segmented by a Cisco 2948G-L3 switch. There will be several distinct subnets for various departments and resources.

We will not detail the configuration of the 2948G other than to identify the subnets and the basic access restrictions. Our primary focus is to provide a general level of separation between the Finance Department VLAN and other VLANS in the company. This will be achieved through Cisco standard Access Control Lists.

Diagram of Architecture



Border Routers

For our Internet Border Router we chose a single Cisco 3640 Series Modular Access Router. The 3640 was chosen because it is robust enough to handle our ACL filtering and our BGP router process, while supporting redundant T1 connections to two separate Internet Service Providers. (Broadwing and Cable & Wireless).

The router has 128 MB DRAM, 16 MB Flash a single NM-1FE2W with 2 WIC-1DSU-T1. We've chosen to use Cisco IOS IP Feature Set Version 12.2.19a. At the time of this writing that was the most recent stable version for our release train.

The router is particularly cost effective for GIAC because it meets their immediate needs. There are large numbers of these routers available on the refurbished market. GIAC will also purchase Cisco's 7x24 Onsite Smartnet with 4 Hour response time. In arriving at these decisions we had to weigh numerous options including cost, the need for redundancy, life cycle of the equipment and performance needs. The reason for GIAC to receive Internet service through two providers is related to experience based perception that telecom providers are not always responsive unless we pay premiums for SLA offerings.

This device provides connectivity to the worldwide Internet via redundant Internet Service Providers. This connectivity is critical to GIAC's business. Since they sold of the manufacturing part of the business, all of GIAC's revenue is tied to being able to communicate over the Internet.

This device is GIAC's first line of defense in the security architecture. At this placement in the architecture, we will use this device to filter out most of our incoming traffic. Obviously, if we know that don't allow any MS-RPC (TCP Port 135) traffic in through our firewall, why even let it get to the firewall if our router can sufficiently filter it? This methodology will also help protect us in the event of user error. What if we accidentally left our firewall software in a stopped state and the underlying OS was not properly configured? Well, this would indeed be bad, but the router might prevent anyone from detecting any exposed services on the firewall if we are filtering them at the router. Since Checkpoint firewalls have quite a number of default properties, many of which are notoriously left 'on', using the router as our first line of defense could also help protect us from information gathering attempts if we accidentally or through ignorance overlook some of these settings. Using the Internet router as our first line of defense will also allow us to focus our firewall on the applications that it needs to support and the systems it needs to protect. Additionally we are not tied to a single vendor for our security architecture.

To summarize, we chose the Cisco 3640 for a variety of technical and budgetary reasons. Cisco provides a rich set of robust features. However, the principle of Defense in Depth allows us to balance performance vs. security because we are using multiple layers of security. If our Internet router was also our firewall, then our choices about which feature sets to use, and which options

to enable would be drastically different. In our scenario, however, using our Internet router as a basic packet filter will allow us to screen out a large portion of unwanted traffic in an efficient manner. It will then be the task of the firewall to handle more complex traffic.

Primary Internet Firewall and VPN Gateway

For our primary (and only) Internet firewall we chose to use Checkpoint Firewall-1/VPN-1 software running on Red Hat Linux 7.3. For the firewall hardware we are going to be re-using one of GIAC's HP Kayak Workstations. We will install an Adaptec Quad-Port Fast Ethernet adapter. This adapter is supported in Red Hat using the 'starfire.o' loadable module. We will configure the OS according to Checkpoint's Linux Minimum configuration document [2]. We decided to use Linux as opposed to Windows or another platform altogether (an appliance based platform like Nokia for instance) out of budgetary concerns. Linux was free and would provide a robust platform using our existing HP Kayak.

The version of Checkpoint is NG FP3 with Hotfix-2. Beginning with Checkpoint NG, updates were provided in the form of Feature Packs (FP1, FP2, etc). FP3 contains improvements to various areas including IKE interoperability, which is a factor we need to consider because GIAC will have VPN links to various partners and customers. Additionally FP3 begins to provide support for SSL based (clientless) VPNs which GIAC intends to make use of as the technology matures.

The firewall component plays a very important part in GIAC's security architecture because it is responsible for protecting both GIAC's publicly accessible screened subnet (web server, email server), and, more importantly GIAC's screened VPN subnet which contains most of the systems critical to GIAC's business.

The VPN Gateway component, while combined on the same system as the firewall, plays an important role in that much of GIAC's business operations take place with partners and customers through VPN connections. VPN is used so that we can continue to use well known applications such as FTP in a more secure fashion. While it is true that we don't necessarily trust our customers and partners, we tend to trust them a little more then any unknown entities that may be accessing use over the Internet. So the VPN does provide an added layer of security in addition to protecting the confidentiality of our data. We know exactly who needs to access the systems on GIAC's screened VPN subnet; therefore there is no need for them to be publicly accessible. In contrast, we cannot know who will need to access our public web server, or send our employees e-mail, so we have to allow all access (to only the necessary services, of course) to those systems on GIAC's screened public Subnet.

We decided to combine our firewall and VPN Gateway into a single system. This is something that Checkpoint does very well, allowing us to administer the security policy for both functions together. While some would argue that combining these functions together does not adhere to Defense-in-

Depth, we believe that it is a good choice for a small organization such as GIAC. Once we install this system, it is going to be GIAC's IT Support Staff's day to day responsibility to maintain these systems. Having both functions on the same system is allows us to view the entire policy in a single comprehensive view. Checkpoint provides a nice GUI for managing their firewall / VPN gateways. While this is often something frowned upon by security professionals, it is something that can be quite beneficial for a company like GIAC who needs to train their IT support staff to administer the firewall. It's going to require less of a learning curve to train a Windows system administrator to configure Checkpoint's firewall, then to administer an Iptables firewall, or a PIX firewall, for example. The degree of accuracy is going to be better with the familiar interface.

Internal Layer – 3 Switch

This section is only meant to detail the general access restrictions established between internal subnets. We'll be using a Layer 3 Switch (a Cisco 2948G-L3) to separate our internal VLANS. Primarily we want to filter access to the Finance VLAN from else where on the internal network.

This Layer 3 switch and its configuration further promote our use of Defense –in-depth by providing us with a reasonable level of access separation between GIAC's various functional departments. It is not intended to overcome various VLAN based attacks [1].

IP Addressing Scheme

For GIAC's IP Addressing Scheme, we choose to use an arbitrary routable public address for the purposes of this document. In real life, GIAC's ISP would provide such an address range. Globally routable IP addresses are necessary to conduct any sort of business using the Internet. Without them, it goes without saying, nobody could get to us. Nobody could respond to our requests. Communication would cease.

Note: We choose this subnet somewhat arbitrarily. According to IANA the entire 45/8 network is registered to the Interop Show Network [5]. wonder how much of that is in use?

We asked GIAC's primary ISP to issue us a single class C subnet. Many ISPs now require justification when requesting this many addresses (254 useable). While we only need a handful of public addresses, our justification was that one of GIAC's business requirements is to have fully redundant Internet connections to two separate ISPs. In order to facilitate this through BGP, a full class C subnet (a /24 bit netmask) is required, because, most ISP's will not advertise anything smaller than a /24 outside of their networks.

Additionally, we had to request an AS (Autonomous System) number from ARIN [6], in order for use to be able to multi-home to two providers using BGP.

Our justification for obtaining an AS number was that we are a multi-homed site. For the purposes of this document, however, we will be using a private AS number (64512) which are typically reserved for ISPs to provide to their customers. We'll assume that it is not private in our examples.
The following information describes the issued public, routable subnet, and how we have further divided it for GIAC's purposes.

Public – Routable IP Addressing

Subnet	Mask bits	Purpose
45.10.1.0	24	Base subnet
45.10.1.0	28	GIAC DMZ – Internet LAN
45.10.1.16	28	Reserved for expansion
45.10.1.32	27	Reserved for expansion
45.10.1.64	27	Public Screened Subnet
45.10.1.96	27	Reserved for expansion
45.10.1.128	26	VPN Screened Subnet
45.10.1.192	26	Reserved for expansion

We have chosen to implement our screened Public and screened VPN subnets as publicly addressed networks. There are several reasons for this. For one, it will mean less NAT related rules on our firewall. Another good reason is that it will eliminate any IP address conflicts when establishing our VPN links to GIAC's customers and partners. Should Partner X be using the same RFC 1918 networks in their internal network, it won't matter to us because there will be is no overlap in encryption domains. Actually, this is not entirely true. Since our remote VPN users will want to access their workgroup systems, our 172.24.x subnet will have to be part of our encryption domain. This is something GIAC will need to be aware of then negotiating connection details with partners and customers.

The following details GIAC's Private IP Addressing scheme

Private RFC 1918 Internal Addressing

172.24.0.0	20	Base Subnet
172.24.1.0	24	General Purpose VLAN
172.24.2.0	24	Research VLAN
172.24.3.0	24	Finance VLAN
172.24.4.0	24	Sales – Executive VLAN
172.24.5.0	24	Data Entry VLAN
172.24.6.0	24	Procurement VLAN
172.24.7.0	24	IS & T VLAN
172.24.8.0	21	Reserved for expansion

To conserve public address space, GIAC will use RFC 1918 addressing for their internal networks [7]. This will allow us plenty of room for future expansion. It will also eliminate the need for company–wide re-addressing should we decide to change ISPs, since most public addressing is no longer portable.

For the purpose of this document, our ISP serial connections will be using the following subnets: 45.127.254.8/30 and 45.254.250.32/30. Once again, these are fictitious addresses. Any resemblance to existing known networks is strictly coincidental.

Summary of Architecture

We chose this architecture based on a variety of business requirements, technical preferences, and budgetary factors. The architecture provides Defense-in-Depth by leveraging several levels of security present in the various components. We believe our architecture to be a balance between security and usability. It is a simple design, largely dictated by budgetary factors. There are countless other security devices and features which we could have included in our design such as Intrusion detection systems, authentication systems, proxy servers, etc. However, given GIACs size, and the demands on GIACs staff, (who will ultimately be tasked with maintaining the day to day operations of the network), additional complexity would have been a hindrance at this time. The chosen architecture will give them a chance to succeed, with a reasonable amount of training, and a continued proactive business attitude towards the importance of network security systems. The design should be simple enough that it allows for expansion and/or the addition of other mechanisms when the need arises.

Assignment 2 – Security Policy and Tutorial

Internet Border Router Security Policy

Cisco provides a very feature rich operating system on their routers (Cisco IOS). We've decided to use Cisco's IP Feature Set on our Internet Border routers due to cost and because we have not chosen to implement some of these advanced features at this time. These include CBAC and TCP Intercept. Reverse path forwarding check can't be used because our routing may not be symmetric due to our use of multiple ISPs.

The following are now defaults in version 12.2 of the Cisco IOS. Even if you enter them into the configuration, they don't display when doing a "show running-config"

- *no ip directed-broadcast - most commonly used in a "smurf" DOS attack.*
- *no service tcp small - echo, chargen, daytime and discard. We don't need these and they can be used for malicious purposes.*
- *no service udp small*
- *No ip finger - Useful for obtaining information on a node. We don't need this.*

Configuration of GIAC's Internet Border Router

!

```
version 12.2
```

IOS Revision, automatically generated

```
Service timestamps debug datetime localtime
```

Put a timestamp on debug entries.

```
Service timestamps log datetime localtime
```

Put a timestamp on logging messages
Both of these are very useful when attempting to correlate events between multiple systems.

```
service password-encryption
```

This encrypts passwords as stored in the config file. This is useful for 'looking over your shoulder' situations. According to Cisco, it "provides a minimum of protection for configured passwords" [8]

!

```
hostname giacigw1
```

Our chosen hostname; Not particularly significant.

```
!
logging buffered 16384 debugging
```

Increase the logging buffer to actually hold more. We've got the memory to handle it.

```
enable secret 5 $1$vIbf$xOwgptJ0PAarBbbIz9qD21
```

Don't use old 'enable password', as it is known weak encryption and easy to crack using readily available tools [9]

```
!
no ip source-route
```

Good practice to disable this by default. This is best explained by Cisco: "The IP protocol supports source routing options that allow the sender of an IP datagram to control the route that datagram will take toward its ultimate destination, and generally the route that any reply will take. These options are rarely used for legitimate purposes in real networks!" [8]

```
interface FastEthernet0/0
 ip address 45.10.1.1 255.255.255.248
 ip access-group fe00_in in
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 duplex auto
 speed auto
 no cdp enable
```

ip access-group fe00_in in applies our Extended IP Access-List, fe00_in, to the inbound direction of the interface. We are using duplex & speed auto because our interface is connected to an unmanaged hub. We're we connected to a managed switch, we'd likely want to force these to their desired settings. *No ip redirects* is turned off because we have no functional requirement for ip redirects. A malicious entity could use redirects to cause problems on our network, as an ip redirect instructs a node to use a specific router as its path to a particular destination. [8] *no ip unreachables* disables icmp 'unreachable' messages, which could be useful for discovering information about our networks.

No ip proxy-arp disables ip proxy-arp, which has been used by attackers to cause a resource to be consumed responding to arp requests, thus cause a Denial of Service scenario. [10]
No cdp enable We don't need CDP (Cisco Discovery Protocol), and because it can be used to glean all manner of useful information about neighbor devices, we should turn it off by default. Note that we'll also disable it global using the *no cdp run* command.

!

```
interface Serial0/0
 description Broadwing Internet
 ip address 45.127.254.10 255.255.255.252
 ip access-group SerialIn in
 ip access-group SerialOut out
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 no cdp enable
```

On our serial interfaces (to our ISPs) we'll be implementing inbound and outbound ACLs, *SerialIn in* and *SerialOut out*. It's also very helpful for troubleshooting purposes to put a description of the link (such as the provider). Any half-wit attacker can figure this out anyway.

!

```
interface Serial0/1
 description Cable & Wireless Internet
 ip address 45.254.250.34 255.255.255.252
 ip access-group SerialIn in
 ip access-group SerialOut out
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 no cdp enable
```

!

```
router bgp 64512
 bgp log-neighbor-changes
 network 45.10.1.0 mask 255.255.255.0
 neighbor 45.127.254.9 remote-as 6395
 neighbor 45.127.254.9 password 7 050303032D43
 neighbor 45.127.254.9 filter-list 86 in
 neighbor 45.127.254.9 filter-list 80 out
 neighbor 45.254.250.33 password 7 02A87FA8233
 neighbor 45.254.250.33 remote-as 3561
 neighbor 45.254.250.33 filter-list 85 in
 neighbor 45.254.250.33 filter-list 80 out
```

It's important to coordinate with our ISPs when possible and provide a password for our BGP peers in addition to filtering BGP traffic with our ACLs. This will help mitigate potential attacks on our BGP routing processes. We also want to filter incoming routes, in this case, to include only are providers customer routes.

```
!
ip classless
```

Enabled because we are further subnetting our /24 subnet.

```
no ip http server
```

My personal opinion is that the Cisco IOS HTTP interface is simply horrible to use. The CLI is much easier. That aside, it is also highly exploitable, with numerous exploit scripts readily available, and should be disabled. [11] [12]

```
ip as-path access-list 80 permit ^$
ip as-path access-list 85 permit ^3561_[0-9]*$
ip as-path access-list 86 permit ^6395_[0-9]*$
```

This is where we define our filters for sending and receiving BGP routing updates. These are what routes we want to advertise (ip as-path access-list 80), which is currently set to distribute any route that we've defined with our *network x.x.x.x* command. (Currently just 45.10.1.0 netmask 255.255.255.0). ip as-path access-list 85 & 86 are to filter our incoming routing advertisements from our providers. Following the "...permit" statement is Cisco regular expression syntax. [13]. In this case we only want to receive each provider's directly connected customer routes. The provider will likely also filter out what they send us based on what we've asked for, but it's in our best interest to filter the inbound updates as well.

```
!
ip access-list extended SerialIn
 deny   ip 127.0.0.0 0.255.255.255 any log-input
 deny   ip 224.0.0.0 15.255.255.255 any log-input
 deny   ip 10.0.0.0 0.255.255.255 any log-input
 deny   ip 192.168.0.0 0.0.255.255 any log-input
 deny   ip host 0.0.0.0 any log-input
 permit tcp any 45.10.1.64 0.0.0.63 eq www reflect inbound
 permit tcp any 45.10.1.64 0.0.0.63 eq 443 reflect inbound
 permit tcp any 45.10.1.64 0.0.0.63 eq smtp reflect inbound
 permit udp any 45.10.1.0 0.0.0.15 eq isakmp reflect inbound
 permit esp any 45.10.1.0 0.0.0.15 reflect inbound
 permit tcp any 45.10.1.0 0.0.0.15 eq 264 reflect inbound
 permit udp any 45.10.1.0 0.0.0.15 eq 2746 reflect inbound
permit tcp any 45.10.1.0 0.0.0.15 eq 18231 reflect inbound
```

```
permit udp any 45.10.1.0 0.0.0.15 eq 18234 reflect inbound
permit tcp host 45.254.250.33 host 45.254.250.34 eq bgp reflect inbound
permit tcp host 45.127.254.9 host 45.127.254.10 eq bgp reflect inbound

evaluate outbound
deny   ip any any log-input
deny   icmp any any log-input
deny   53 any any log-input fragments
deny   55 any any log-input fragments
deny   77 any any log-input fragments  (reference Cisco / CERT Bulletin)
deny   pim any any log-input fragments
```

This defines the Access Control List which will be applied to the inbound direction of our serial Interfaces.
The first 5 entries deny any hosts with a source address that is either an RFC 1918 address, illegal (host 0.0.0.0), or multicast (224.x.x.x), since we are not using any multicast services over the Internet. These entries will help prevent spoofing and other questionable traffic. We use log-input to log any occurrences and which interface they arrived on. By periodically reviewing our logs this could potentially alert us to suspicious or problematic situations. These rules must go fist because our very next rules (entries 6 through 8), which we anticipate the most matches on, specify *any* as the source addresses. This will allow Internet access to our Screened Public network for those services that we allow (www.443, smtp). We use reflexive access lists for these rules to reduce the complexity of our ACLs overall. Reflexive ACLs replace and improve upon the formerly used 'established' keyword, by creating a temporary 'session table'. It essentially provides stateful tracking. We for each reflexive acl (*inbound* in SerialIn), we'll need a corresponding *Evaluate* statement in our outbound ACL.
Statements 9 through 14 define the protocols and ports we need to allow through to our VPN Gateway. This will support both site-to-site VPNs and our Checkpoint SecureClient users.
Entries 15 & 16 will allow BGP traffic from each of our providers.
Entry 17 is the *Evaluate* rule necessary to support the reflexive ACL *outbound,* defined in our outbound SerialOut extended IP Access-List.
Entries 18 & 19 deny any IP and ICMP traffic not previously explicitly allowed.
Entries 20 through 23 deny several other protocols related to a specific Cisco security vulnerability which could potentially result in a DoS situation in which " A device receiving these specifically crafted IPv4 packets will force the inbound interface to stop processing traffic". See the Cert Advisory CA-2003-15 for more information [15] We definitely want to use *log-input* in case we encounter any of these.

Because the entries in our ACLs are processed sequentially, we've order our rules such that those we anticipate matching most frequently are as close to the beginning of the ACL as we can get them, without compromising the functionality of the ACL. We never want to see traffic from any of the hosts and networks listed in entries number 1 through 5. Even though most of our traffic is going to be to our screened public network or our VPN gateway, if we placed those rules first, someone could still send http, https, smtp or various VPN-related packets to our systems with invalid addresses. Because protocols 53, 55, 77, and PIM are not IP or TCP or UDP, we can safely place them last yet remain effective.

```
Ip access-list extended SerialOut
 permit udp 45.10.1.0 0.0.0.15 any eq isakmp reflect outbound
permit esp 45.10.1.0 0.0.0.15 any reflect outbound
 permit ip 45.10.1.0 0.0.0.255 any reflect outbound
 permit icmp 45.10.1.0 0.0.0.255 any reflect outbound
 permit tcp host 45.254.250.34 host 45.254.250.33 eq bgp reflect outbound
 permit tcp host 45.127.254.10 host 45.127.254.9 eq bgp reflect outbound
 evaluate inbound
 deny   ip any any log-input
 deny   icmp any any log-input
 deny   53 any any log-input fragments
 deny   55 any any log-input fragments
 deny   77 any any log-input fragments
 deny   pim any any log-input fragments
```

The basic reason which applies to our inbound Serial ACL also applies to our outbound serial ACL. We have less entries because we are not going to be a specific. Our firewall will be filtering on a more detailed level for our outgoing traffic. Therefore we allow any IP, ICMP, and ESP out. We have included a permit rule for ISAKMP in case we need to tune any reflexive ACL timeout properties for this service. Notice that we are only allow traffic from our subnet.

Next we have the *evaluate* statement which applies to the reflexive ACLs used in our SerialIn ACL.

Finally we deny any IP and ICMP that is not from our subnet, as well as protocols 53,55,77, and PIM. Our inbound ACL for our Ethernet interface will take care of all of that as well, but it doesn't impact us negatively to have these on our outbound serial ACL as well. (For instance if we dropped the ACL on our Ethernet Interface for troubleshooting and forgot to re-apply it.)

```
ip access-list extended fe00_in
 permit ip 45.10.1.0 0.0.0.255 any
 permit icmp 45.10.1.0 0.0.0.255 any
 deny   ip any any log-input
```

```
deny   53 any any log-input fragments
deny   55 any any log-input fragments
deny   77 any any log-input fragments
deny   pim any any log-input fragments
```

After our firewall, this will be our next layer of filtering for all traffic leaving our network through our router. We will permit anything from our subnet and deny everything else.

```
logging facility local5
logging source-interface FastEthernet0/0
logging 45.10.1.4
```

GIAC will be centrally logging to a syslog server. Specifying *local5* as our logging facility will allow us to configure our syslog.conf file to put all messages from our router into a dedicated log file.

```
no cdp run
```

This disables CDP globally. As previously mentioned, we have no functional need for CDP.

!

```
line con 0
 password 7 09444B05150A
line aux 0
 password 7 151A0E000825
line vty 0 4
 password 7 0507030C35595C0C
 login
 transport input telnet
```

Don't forget to assign passwords to the console and auxiliary ports on the router. This prevents someone from hooking a readily available Cisco console cable into our router and side stepping whatever security measures we may have configured on the router. We also define a password, and specify our transport input on our VTY lines. These are used for telnet into the router. Cisco routers support a variety of input transports including rlogin, telnet, DEC mop, and others depending on the IOS feature set. It is best to explicitly define the allowed transport on our VTYs.

!

```
ntp server 128.118.46.3 version 2 source FastEthernet0/0
ntp server 138.39.7.20 version 2 source FastEthernet0/0 prefer
```

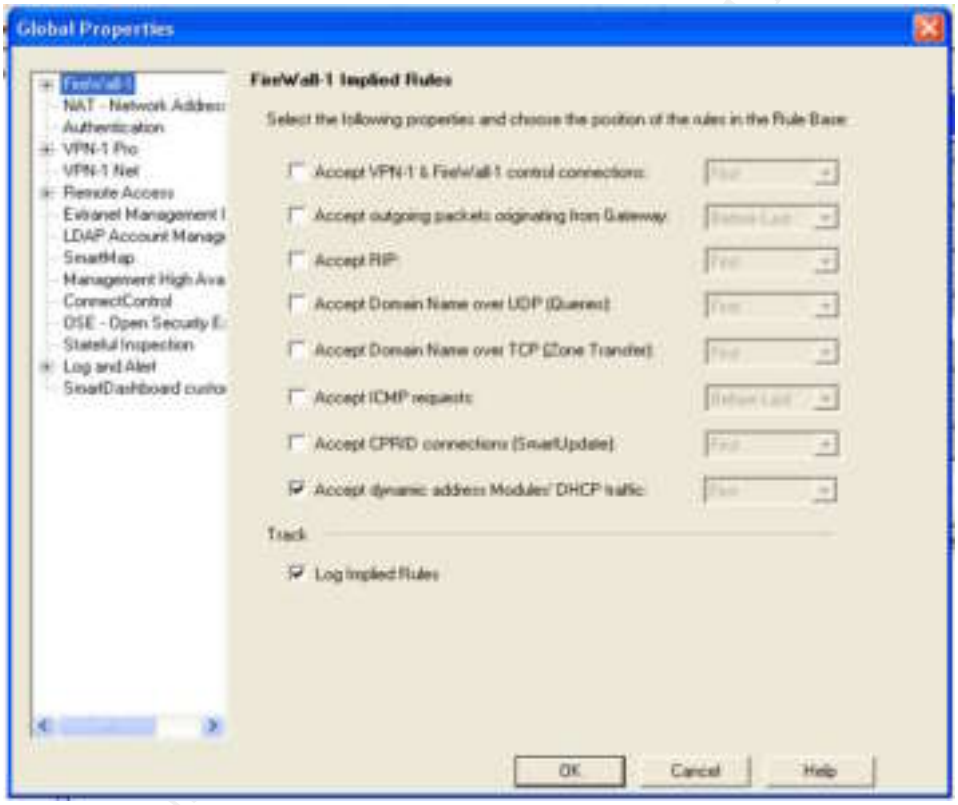
Here we define which NTP servers we are going to use. Despite the fact that NTP can often be exploited, proper time synchronization is important to our logging operations, and is essential when correlating events from between numerous devices, such as routers, switch, firewall, and Intrusion systems. We have also specified to use our FastEthernet interface as the source interface from which the requests come.

end

Internet Firewalls

First we will define what our Global Properties are going to be for our Firewall-1 installation. Since our component is both a firewall and a VPN gateway, some of these properties apply to the firewall components while others apply to the VPN Gateway components. Firewall-1's default implied rules leave a lot of services enabled. While this makes it easy to get going out of the box, it also leaves a lot of possibilities for reconnaissance and also targets for attacks. Thus we've gone and changed the default properties as follows.

Firewall-1 Implied Rules



[de-selected] Accept VPN-1 & Firewall-1 control connections – Our rulebase will include specific rules for allowing the firewall administrators and VPN clients and remote nodes to connect. Use caution as you can deny yourself access by de-selecting this.

[de-selected] Accept Outgoing packets originating from the gateway – Another rule we will define as needed in rulebase. We wouldn't want the firewall to be used to access the Internet, or our internal network should the firewall itself become compromised. It should be noted however, if someone should compromise our firewall, it would not be difficult for them to modify our rulebase.

[de-selected] Accept RIP – Deselected because we do not use RIP. RIP is a dynamic routing protocol.

[de-selected] Accept Domain Name over UDP(Queries) – We'll define this in our rule base. Generally we only want to allow our DNS server to forward queries. This will help prevent various applications from tunneling out using port 53. An example of this would be AOL Instant Messenger. While AOL IM will still be able use any other allowed port to get out (80,443), it will be easier for us to regulate in the future, should we choose to do so, by taking other measures such as installing an application proxy server on the network.

[de-selected] Accept Domain Name over TCP – Another case where we'll define more granular access in our rulebase. GIAC's public, root-level domain will be hosted by a third party DNS provider. GIAC's internal DNS will be configured with a secondary zone for the giac.com domain, simply to provide local resolution. Therefore the internal DNS server will require domain-tcp to the third-party DNS servers.

[de-selected] Accept ICMP requests. ICMP can be very useful for gathering information; therefore it's best not to allow access by default. We'll create rules where we want to handle ICMP traffic for any reason.

[de-selected] Accept CPRID Connections. TCP port 18208 is used for the Checkpoint Remote Installation protocol and is utilized by SecureUpdate, which is a tool for managing remote software installations of Checkpoint Software. We only have one installation and will not be using SecureUpdate.

[selected] Accept Dynamic Access modules DHCP traffic - We left this selected because we plan to use DAIP at some of our partner locations. Selecting this will create an implied rule, though it will only apply to DAIP modules. See the following link for more information on using DAIP modules.
http://support.checkpoint.com/kb/docs/public/sofaware/pdf/DAIP_Support.pdf
This will only apply if we are using our management server to manage the DAIP modules. GIAC potentially wants to do this in the future, but it will require licensing upgrades to the existing firewall.

[selected] Log Implied Rules - We log it all. Unless we were seriously deficient on disk space, there's very little reason not to log everything. Even if it is predictable, "harmless" traffic, having a log of it can often be extremely useful for troubleshooting.

Firewall-1 – Security Servers

We are not going to use any security servers. If we were, here we would define welcome banners, and various other properties.

Firewall-1 – VOIP

Here we would define specific configuration options for supporting Voice over IP through the Checkpoint firewall. GIAC currently has no VOIP plans.

NAT – Network Address Translation Properties

Automatic NAT Rules - Allow bi-directional NAT

This affects how the firewall processes address translation rules when using automatic NAT. (NAT is selected on a network object). This is best described in the Checkpoint Helpfile:

> *"Allow bi-directional NAT - If more than one automatic NAT rule matches a connection, then both rules are matched.*
> *When NAT is defined for a network object, an automatic NAT rule is generated which performs the required translation. If there are two such objects and one is the source of a connection and the other the destination, then without bi-directional NAT, only one of these objects will be translated, because only one of the automatically generated NAT rules will be applied. With Bi-directional NAT, both automatic NAT rules are applied, and both objects will be translated.*
> *The operation of bi-directional NAT can be tracked using the SmartView Tracker" [16]*

Automatic NAT Rules - Translate Destination on Client Side

This is selected by default. Were we to maintain compatibility with earlier versions we could deselect this property. This represents a change in the order in which packets are processed which occurred between Firewall-1 4.x and NG versions. In NG, a packet coming into a gateway that matches a Static Destination mode NAT rule is translated on the client side of the gateway. In prior versions, the translation took place on the server side of the gateway. This meant, for instance, that we had to have additional routing configuration for each destination NAT rule to ensure that the incoming packet would be sent out the correct interface after being translated. It would also further complicate Anti-Spoofing configuration.

Automatic NAT Rules - Automatic ARP configuration

> In prior versions of Firewall-1, an arp entry had to be manually created on the firewall for each NAT'd object. This property removes that requirement.

Manual NAT Rules – Translate destination on client side.

> Same property as described, however in this case it will apply by default to any manual NAT rules we might create.

IP Pool NAT

This is a very useful feature of Firewall-1, especially in organizations that have multiple Internet gateways. This is a mechanism which will prevent asymmetrical routing in such scenarios, by modifying inbound SecuRemote/SecureClient traffic such that it gets translated with an address that will be routed back to the specific entry point gateway. Since our network is not that complex and we don't have any requirements to allow connections to our VPN clients (which would get screened by the firewall), we won't enable this Since NAT adds some overhead, it's generally best not to NAT where NAT is not needed.

Non-Unique Address Ranges

These properties apply to how SmartMap treats certain address ranges. Since we are not going to be using SmartMap we will leave this set to the default.

Authentication Properties

We will leave everything on the Authentication page set to the defaults. This will allow for a balance between security and usability. Connections for rlogin and telnet will be terminated after 3 failed attempts. 3 attempts will also apply to session and client authentication.

VPN-1 Pro Properties

Setting this to Traditional or Simplified mode per new Security Policy will allow us to choose which type of policy we want to create. Generally speaking, our experience is more with traditional mode configurations and so we are going to choose that for our architecture. Simplified mode may be designed to make configuring the firewall easier if you are not familiar with it, but we are comfortable using traditional mode.

We'll leave much of the following sections set to defaults. Checkpoint provides many features and many of these properties apply to the various nuances of these features.

VPN-1 Pro – Early Versions Compatibility

We are not going to be managing any earlier versions so we'll simply leave these properties set to their defaults.

VPN-1 Pro Advanced Properties

This section contains various properties concerning Multiple Entry Point configuration (when an organization has more then one VPN-1 Gateway).

Enable decryption on accept – applies to packets that are accepted but don't explicitly match an encryption rule. For the most part this won't apply to us as we are going to explicitly define our encryption rules. It may be prudent to disable this.

Remote Access – VPN-1 SecuRemote/SecureClient
Remote Access – VPN Basic
Remote Access – VPN – Advanced

Here is where we would set our default encryption properties. We could also choose to force these setting for all users as a default. We are not going to do that because it will allow use more flexibility when dealing with interoperability issues. It may be beneficial to require different encryption properties for various users, particularly if some of our users are in countries affected by encryption laws and other legal issues.

Additional Sections (left at defaults, not discussed herein)
Remote Access – Certificates
Remote Access – Secure Configuration Verification
Remote Access – Early Versions Compatibility
Extranet Management Interface
LDAP Account Management

SmartMap

We are not going to be using Smartmap.

Management High Availability

A nice feature for larger shops, we will not be using a high availability management station. Our management information will be backed up daily.

ConnectControl

For load balancing and availability configurations. We do not currently have such features implemented.

OSE – Open Security Extensions

Default properties when defining OSE devices. OSE Stands for Open Security Extension and is a mechanism that allows us to manage the ACLs on our Cisco or Nortel Routers. This requires a separate license from Checkpoint. GIAC will not be using this functionality.

Stateful Inspection Properties
Log and Alert Properties
Log and Alert – Alert Commands
SmartDashboard customization

Various features affecting the look and feel of the SmartDashboard management interface. Not relevant to our security policy and configuration, will leave as defaults.

Rulebase Configuration

Before we detail our rulebase, it will be helpful to review the objects we have created.

Firewall / VPN Gateway Object

General Properties

Name:tautog
Ip address: 45.10.1.2
Version: NG Feature Pack 3
[checked] Firewall-1
[checked] VPN-1 Pro
[checked] Primary Management Station
[checked] SVN Foundation
[checked] Log Server

Topology

Topology is important to define properly because this is where we will define anti-spoofing configuration.

Name	IP Address	Network Mask	IP Addresses behind interface
eth0	172.24.1.1	255.255.255.0	GIAC_PrivateNets
eth1	45.10.1.2	255.255.255.240	External
eth2	45.10.1.65	255.255.255.192	This Network
eth3	45.10.1.129	255.255.255.192	This Network

NAT

We are not utilizing automatic NAT for our gateway object.

VPN

Since we are not using “Simplified Mode” for our VPN configuration, we’ll need to define our encryption properties within the Traditional Mode Configuration dialog. We’ll define our object to support 3DES, AES-356,

and DES for Key Exchange Encryption, MD5 or SHA1 for data integrity support. We'll also select 'Exportable for SecuRemote/SecureClient. Within this dialog is where we'll define any Pre-Shared Secrets, for example, to support any site to site VPNs with our partners and /or Premier Customers.

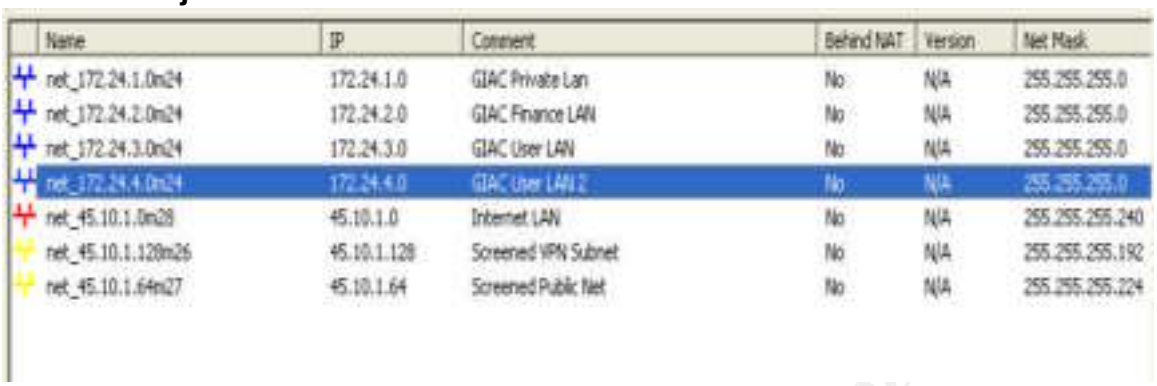
Well generally be accepting the defaults for the remaining properties in the object configuration. One important addition is under Authentication, we'll check VPN-1 & Firewall-1 Password. Currently users will be authenticated to the Checkpoint user database. GIAC will be able to change this mechanism in the future should the need arise.

Node Objects

Name	IP	Comment	Behind NAT	Version	Net Mask
av.inside.giac.com	172.24.1.51	NAI ePolicy Anti-Virus Distribution Server & NT...	No	N/A	
dbFortune.giac.com	45.10.1.140	GIACs Sybase Database System	No	N/A	
eCookie.giac.com	45.10.1.72	Secure Server for Customer Ordering	No	N/A	
fis.giac.com	45.10.1.142	Fortune In Sales Sales Management System	No	N/A	
ftp.giac.com	45.10.1.141	FTP Server for delivering data from customers ...	No	N/A	
fwadmin1	172.24.1.100		No	N/A	
giacgw1	45.10.1.1	Internet Router	No	N/A	
logger.giac.com	45.10.1.4	Syslog Server	Yes	N/A	
mail.inside.giac.com	172.24.1.52	Lotus Domino 6.x Mail Server	No	N/A	
nat_hide_PrivateNets	45.10.1.6	Hide 172.24.x.x Nets	No	N/A	
ns1.dnsprovider.com	128.10.11.250		No	N/A	
ns1.inside.giac.com	172.24.1.50	Internal DNS Server	No	N/A	
ns2.dnsprovider.com	38.100.161.15		No	N/A	
smtp.giac.com	45.10.1.71	GIAC SMTP Relay (Domino) Notes Replication t...	No	N/A	
suppliers.giac.com	45.10.1.73	Supplier Web Server	No	N/A	
teamwork.giac.com	45.10.1.143	Lotus Domino Server for Collaboration with Par...	No	N/A	
www.giac.com	45.10.1.70	AKA e-fortune.giac.com	No	N/A	

Our 'node' objects define the hosts and host IP Addresses which are relevant in our security policy. These hosts represent systems on our Screened VPN subnet, Screened Public subnet, our Internet DMZ, and our Private Networks.

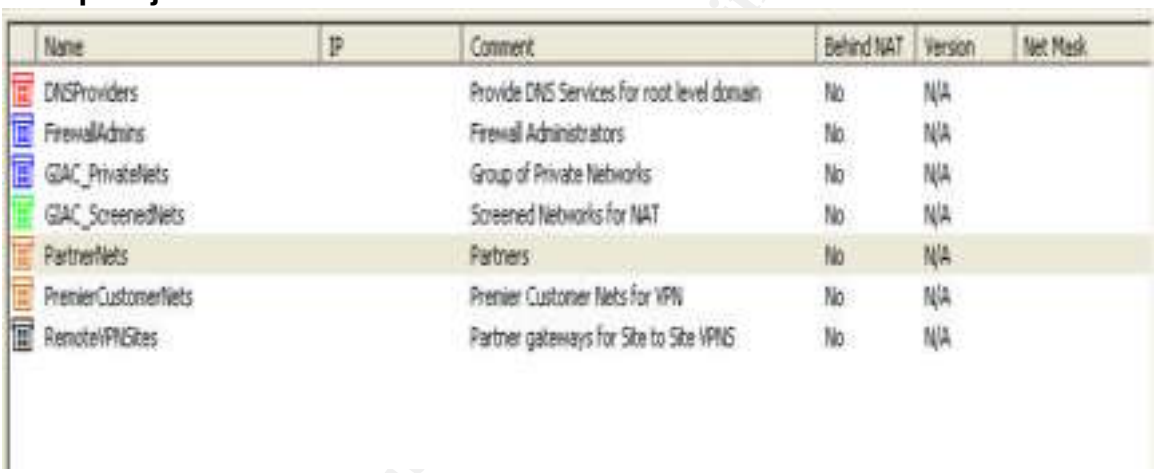
Network Objects



Name	IP	Comment	Behind NAT	Version	Net Mask
net_172.24.1.0m24	172.24.1.0	GIAC Private Lan	No	N/A	255.255.255.0
net_172.24.2.0m24	172.24.2.0	GIAC Finance LAN	No	N/A	255.255.255.0
net_172.24.3.0m24	172.24.3.0	GIAC User LAN	No	N/A	255.255.255.0
net_172.24.4.0m24	172.24.4.0	GIAC User LAN 2	No	N/A	255.255.255.0
net_45.10.1.0m28	45.10.1.0	Internet LAN	No	N/A	255.255.255.240
net_45.10.1.128m26	45.10.1.128	Screened VPN Subnet	No	N/A	255.255.255.192
net_45.10.1.64m27	45.10.1.64	Screened Public Net	No	N/A	255.255.255.224

Network Objects, like some host objects will become members of groups in some cases.

Group Objects



Name	IP	Comment	Behind NAT	Version	Net Mask
DNSProviders		Provide DNS Services for root level domain	No	N/A	
FirewallAdmins		Firewall Administrators	No	N/A	
GIAC_PrivateNets		Group of Private Networks	No	N/A	
GIAC_ScreenedNets		Screened Networks for NAT	No	N/A	
PartnerNets		Partners	No	N/A	
PremierCustomerNets		Premier Customer Nets for VPN	No	N/A	
RemoteVPNSites		Partner gateways for Site to Site VPNS	No	N/A	

We will often make use of group objects to keep the rulebase from getting cluttered and to facilitate an easier understanding of the rule base.

Custom Services

Additionally we have defined the following custom services:

GIAC_Sybase	TCP Port 6500	The TCP Port which our Sybase Server listens
MS-TermServ	TCP Port 3389	Remote Desktop Access
VNC	TCP Port 5900	Remote Access

Firewall-1 User Groups

We also created several User Groups, including

Patchers	Users allowed to patch systems (run WindowsUpdate, etc.)
VPN_ITSupport	VPN Users with any reason to telework (developers, dba's it support)
VPN_Sales	***Sales and remote Sales Users***

We'll use these groups in our encryption and client authentication rules.

SecuRemote DNS Server Object

We've also created a Server object of type Securemote DNS. We named this object GIAC_InternalDNS, with a host value of ns1.inside.giac.com. This object will respond to queries for *.giac.com and *.inside.giac.com. The object is used by Securemote/SecureClient to encrypt DNS queries for those domain suffixes. This will allow our VPN clients to transparently use GIACs internal DNS server and solves problems associated with "Split DNS" configurations, where an organization uses separate internal (or private) and public DNS configurations.

Rulesbases

Security – GIAC_PrimaryPolicy

Rule No.	Source	Destination	Service	Action	Track
1	FirewallAdmins	tautog giacigw1	ssh CPMI icmp-requests telnet	accept	log
2	GIAC_PrivateNets	any	http https ftp	accept	log
3	any	eCookie.giac.com suppliers.giac.com www.giac.com	http https	accept	log
4	any smtp.giac.com	smtp.giac.com any	smtp	accept	log
5	GIAC_PrivateNets	net_45.10.1.128m26 net_45.10.1.64m27	lotus GIAC_Sybase MS-Termserv VNC	accept	log
6	smtp.giac.com mail.inside.giac.com	mail.inside.giac.com smtp.giac.com	lotus	accept	log
7	GIAC_ScreenedNets	av.inside.giac.com	http	accept	log
8	GIAC_ScreenedNets	ns1.inside.giac.com	domain-udp	accept	log
9	ns1.inside.giac.com	DNSProviders	domain-udp domain-tcp	accept	log
10	eCookie.giac.com www.giac.com	dbFortune.giac.com	GIAC_Sybase	accept	log
11	giacigw1	logger.giac.com	syslog	accept	log
12	RemoteVPNSites tautog	tautog RemoteVPNSites	IPSEC	accept	log
13	any	tautog	FW1_topo FW1_pslogon tunnel_test IPSEC	accept	log
14	Patchers@GIAC_ScreenedNets	any	http https ftp	client Auth	log
15	VPN_ITSupport@any	GIAC_PrivateNets GIAC_ScreenedNets	domain-udp lotus https http ftp GIAC_Sybase VNC MS-TermServ	Client Encrypt	log
16	VPN_Sales@any	mail.inside.giac.com ns1.inside.giac.com fis.giac.com	lotus domain-udp https http	Client Encrypt	log
17	PartnerNets	teamwork.giac.com dbfortune.giac.com	GIAC_Sybase lotus	encrypt	log
18	PremierCustomerNets	fis.giac.com ftp.giac.com	ftp https http	encrypt	log

Descriptions of rules for Security – GIAC_PrimaryPolicy

This is our security policy for GIACs primary firewall and VPN Gateway. For each rule we will explain what function the rule provides for and why it is important.

1. This allows a pre-defined group of Firewall Administrators with direct access to the firewall (tautog). A word on the name, it's arbitrary. We didn't want to name our firewall something obvious like 'GIACfw' or 'gatekeeper', so we choose an arbitrary name. This does not actually matter in the context of Checkpoint's rulebase and object definitions, but concerning DNS names we didn't want to provide any clues. It's important that we limited access to the firewall by individual hosts to prevent just any "internal" host from being able to attempt to gain access to the firewall. Because our firewall is using Linux as the underlying operating system, we want to allow Ssh (Secure Shell – TCP Port 22) access to the firewall to perform any OS related functions such as updating packages or rebooting the system. Our firewall is also our management server, so by allowing CPMI (Checkpoint Management Interface – TCP Port 18190) our firewall administrators can management the firewall using the Checkpoint Smart Clients. We also want a limited number of systems to be able to ping the firewall. You'll notice that 'giacigw1' is also a destination in the rule. 'Giacigw1' is our Internet router. Since it is running the standard 'IP Feature Set' of the Cisco IOS, we need to allow telnet for remote administration. This doesn't create a problem with Tautog as our telnet daemon has been turned of intentionally on the firewall.

2. This allows the group of GIAC_PrivateNets basic web and ftp access to the Internet.

3. Rule number three allows HTTP and HTTPS access to eCookie.giac.com, www.giac.com (AKA e-fortune.giac.com), and suppliers.giac.com for any host on the Internet. These systems are on our Screened Public network.

4. Rule number four allows GIAC Enterprises to send and receive Internet mail with the rest of the world. Like rule 3, smtp.giac.com is on our Screened Public subnet.

5. This rule allows GIAC's Private networks to access GIAC's Screened VPN subnet and Screened Public subnets for Lotus Notes, which allows GIAC internal employees to connect with teamwork.giac.com for collaborative work with trusted partners. It also allows a Lotus Notes client connection to smtp.giac.com (also a Domino 6.0 server). This rule also allows internal users to access dbfortune.giac.com on TCP Port 6500 for database connections. Microsoft Terminal Services / Remote Desktop connections are allowed on TCP port 3389 and VNC connections on TCP port 5900. Remote Desktop is used by GIAC employees to administer

Windows 2000 servers, and VNC is used for remote control of Windows NT4.0. There are a mixture of such servers on GIACs screened subnets

6. This allows bi-directional Lotus Notes traffic on TCP Port 1352, between the mail relay (smtp.giac.com) and the internal mail server (mail.inside.giac.com) Because both servers are Lotus Domino servers we can use Notes mail routing to transfer mail to/from smtp.giac.com.

7. This rule allows those servers on the screened subnets to access av.inside.giac.com, which is an NAI ePolicy server, to receive the latest anti-virus updates.

8. Rule number 8 allows screened subnets to access ns1.inside.giac.com, our internal DNS server, for DNS queries on udp port 53. All GIAC systems (with the exception of the Internet router) use this server for DNS resolution.

9. This rule allows our internal DNS Server to forward queries to our providers DNS Servers. It also allows ns1.inside.giac.com to do zone transfers from our providers DNS servers. GIACs DNS configuration is as follows: Publicly addressed systems (those systems on the screened subnets) all resolve to names in the root level domain, GIAC.COM. DNS for GIAC.COM is hosted by a third party DNS provider. This was a business decision by GIAC to outsource Internet DNS resolution. Since GIAC is also using RFC 1918 addressing internal to their network, we needed to set up internal DNS as well. All internal systems are part of the INSIDE.GIAC.COM sub-domain, for which NS1.INSIDE.GIAC.COM is the primary. NS1 acts as a private secondary for GIAC.COM, strictly for GIAC employees.

10. This rule allows both publicly accessible web servers to query the dbfortune database server on TCP port 6500. There are certain applications on those servers that need to get information from dbfortune.

11. This allows our Internet router to send logging information to logger.giac.com, which is an internal linux server, accessible from the outside only by giacigw1, via an automatically created static NAT rule.

12. This rule allows GIACs partners and premier customers to establish site to site VPNs with GIAC, using IPSEC protocols. Primarily IKE on UDP 500 and protocol 51, ESP.

13. This rule allows any potential SecuRemote/SecureClients to connect to GIACs VPN gateway with the necessary protocols to do a topology download (FW1_topo – TCP port 264), log into the policy server (FW1_pslogon – TCP Port 18207, SecureClient tunnel test (UDP –

18234), and finally IPSEC protocols to establish the encrypted tunnel. Since our SecuRemote/SecureClients are typically coming from dynamic IP addresses, we can do nothing but allow these connections from ANY.

14. This rule allows for IT Support staff to open a temporary connection from any Screened Subnet Host to the Internet, via the Client Authentication method, for purposes of patch updating via HTTP, HTTPS, and FTP. The session will time out after 30 minutes idle and is limited to 15 concurrent connections. To initiate this session, the user first makes a connection from a web browser to the firewall on port 900, then logs in using their Checkpoint Username / Password. Once successfully authenticated, the user will have internet access to be able to download patches, etc. It's important that these systems don't have outbound Internet access so that they can't be used if compromised. This rule allows a temporary, authorized session to bypass that policy.

15. This is a remote access VPN encryption rule. This uses the 'Client Encrypt' Action. This will allow Securemote/SecureClient users who are in the VPN_ITSupport group to access GIACs Private Networks and Screened public network via VPN. Generally VPN_ITSupport includes all users that need more than just email access. We'll allow them to query the internal DNS to resolve *.giac.com names (including *.inside.giac.com hosts). We've done this through the use of the previously defined Securemote DNS server object. Securemote/Secureclient has additionally been instructed through the remote access properties for our gateway to encrypt DNS queries. All other DNS queries will continue to be resolved by the particular users provider supplied DNS servers. SecureClient simply intercepts requests destined for the giac.com domain name. We'll additionally allow http, https, lotus (for e-mail and Lotus Domino administration), ftp, database access via the GIAC_Sybase service, VNC for remote control of NT4.0 hosts, MS-TermServ for remote control administration of Windows 2000 & 2003 Servers, and Windows XP Professional workstations (used by most developers and Dba's).

16. This rule is similar to rule 15, however it is meant for those users who generally need little more than access to email through the VPN. In this case we'll call them VPN_Sales, and they'll need access to GIAC's internal mail servers and additionally, the 'Fortune-in-Sales' Sales Management System (fis.giac.com). Encrypted DNS also applies for this rule.

17. GIAC's partners, as previously defined, need to be able to access teamwork.giac.com (Lotus Notes) and dbfortune.giac.com (GIAC_Sybase). This rules uses the 'Encrypt' action, and is meant for those partners which are configured in a site to site VPN with GIAC, via

their defined networks. Their networks are defined in the PartnerNets group.

18. Similar to rule 17, this is for those Premier Customers who need to ftp their customer data to GIACs FTP Server (ftp.giac.com) and as an added benefit per their contractual agreements, have access to the industry renowned 'Fortune-in-Sales' Sales Management System.

19. This is our 'Drop All' rule. If a connection attempt makes it all the way through the rulebase with out matching, drop it and log it.

Rule Ordering

Generally speaking, since rule processing is sequential, it is best to put our most frequently hit rules closer to the top of the rule base. This may not always be possible or practical. In our rulebase we've put our firewall access rule first, followed by what we anticipate to be our most frequently used rules: Outbound Internet browsing, inbound Internet access to Public services, followed by intra-network communication between GIACs various subnets. Lastly we place our VPN rules together. This is primarily for a consistent logical ordering.

Address Translation – GIAC_PrimaryPolicy

Original Packet				Translated Packet		
Rule No.	Source	Destination	Service	Source	Destination	Service
1	logger.giac.com	any	any	logger.giac.com (valid address)	equals original	equals original
2	any	logger.giac.com (valid address)	any	equals original	logger.giac.com	equals original
3	GIAC_PrivateNets	GIAC_ScreenedNets	any	equals original	equals original	equals original
4	GIAC_PrivateNets	any	any	nat_hide_PrivateNets	equals original	equals original

The address translation rulebase is where any NAT configuration is defined. Because we have decided to implement legally addressed, screened subnets, GIAC's translation is very basic.

1. Rule number 1 and 2 are automatically created rules because we defined the host object 'logger.giac.com' with automatic NAT properties. This will allow our Internet router to send logging information to our internal syslog server. The security policy for this is defined in the Security rulebase, rule 11. Rule 1 translates any traffic originating from logger.giac.com to the public address.

2. Rule 2 translates any traffic destined to the 'valid address' of logger.giac.com, into the private address.

3. Rule 3 is necessary so that any traffic originating from GIAC_PrivateNets and destined for GIAC_ScreenedNets, will not be NAT'ed.

4. Rule 4 is for any traffic leaving the GIAC network that didn't match on rule 3. This rule is necessary for any outbound Internet communication and is known as our Hide Nat rule, as it hides many addresses inside to a single address outside. Rules 1 & 2 where Static NAT rules. (A one to one relationship). Hide NAT is a many to one relationship.

Rule Ordering

Rule ordering for the address translation policy is primarily based on functionality. In this case, rules 1 & 2 are automatically created. Rule 3 is required before rule 4 simply because rule 4 states ANY as the destination, in which case traffic from GIAC_PrivateNets would be NAT'd including when going to the screened subnets. While this would technically still work, it doesn't really make sense to NAT where it isn't necessary.

Desktop Security – GIAC_PrimaryPolicy

Inbound Rules					
Rule No.	Source	Desktop	Service	Action	Track
1	any	all users@any	any	block	log
Outbound Rules					
Rule No.	Desktop	Destination	Service	Action	Track
1	all users@any	any	any	Accept	none

These rules apply strictly to our SecureClient clients, who will download this policy from our policy server. One of the reasons for choosing a SecureClient license is that it comes with desktop firewall capabilities. These rules apply to that functionality. We have a relatively small user base, but one of the reasons for choosing the Checkpoint product was due to plans for expansion and employing more remote workers in the future. Our policy is to allow any traffic outbound from the client (though we will certainly restrict traffic coming through the GIAC VPN gateway – albeit restricted on the gateway itself). We will also deny any inbound traffic. By allowing all outbound traffic we hope to encourage remote users to leave SecureClient running at all times, thus being protected in the inbound direction.

Tutorial

As part of this project, GIAC has contracted us to write a tutorial on how to implement the security policy on the Internet border router. This will be part of the 'knowledge transfer' process that GIAC hopes to achieve with select members of the IT Support staff.

Configuring your Cisco Router from scratch.

For this exercise, you will need the following equipment:

- Your Cisco Router
- Cisco supplied DB9 Adapter and RJ45 cable.
- A Windows based PC with the 'HyperTerm' application installed.
- The router configuration from earlier in this chapter.

When you first receive your router, whether new or refurbished, it should be un-configured. The router will come with a serial console cable and DB9 adapter. Most likely, you will need to connect the DB9 adapter to an available serial port on your computer. The serial console cable is an RJ45 to RJ45 flat 4-pair cable, one end of which will connect into the DB9 adapter, the other into your Cisco router 'Console' port.

For this tutorial, we will be using the default 'Hyperterm' application that is provided with every Windows OS.

With our cables connected and Hyperterm configured and running, we will power on the router for the first time. When the router begins to boot, it will recognize that it is not configured and automatically launch into a 'Setup' configuration dialog wizard.

We are not very fond of 'wizard'-like configuration methods, so we will use ctrl+c to break out of the dialog.

At this point we should see the router> prompt.
We will enter 'enable' and the default blank password. Then we will enter 'conf t' to enter terminal configuration mode.

The Cisco IOS allows us to truncate commands at the point which they become unique. Hence we can enter 'Conf t' in leui of 'configure terminal', but not 'con term' as there exists the command 'connect' as well.

Router(config)# hostname giacigw1 <enter>
This set's the name of our router, and changing the prompt immediately upon hitting enter.

Giacigw1(config)#enable secret h0Zz8I%$a <enter>

Let's pretend that this is our password.

Giacigw1(config)#service timestamps debug datetime localtime <enter>
Giacigw1(config)#service timestamps log datetime localtime <enter>
Giacigw1(config)#service password-encryption<enter>

This changes our passwords to hashes in the config file. Useful to prevent unauthorized glances.

Giacigw1(config)#logging buffered 16384<enter>
Giacigw1(config)#no ip source-route<enter>
Giacigw1(config)#no cdp run<enter>

Disable Cisco Discovery Protocol in global configuration.

Next we configure our Interfaces.

Giacigw1(config)#interface fasteth0/0<enter>

Notice that our prompt changes when we are interface configuration mode.

Giacigw1(config-if)#ip address 45.10.1.1 255.255.255.240<enter>
Giacigw1(config-if)#ip access-group fe00_in in<enter>

This extended IP access list has yet to be defined, but we can still assigned it as our interface inbound filter.

Giacigw1(config-if)#no ip redirects<enter>
Giacigw1(config-if)#no ip unreachables<enter>
Giacigw1(config-if)#no ip proxy-arp<enter>
Giacigw1(config-if)#no cdp enable<enter>

Giacigw1(config-if)#interface serial0/0<enter>
Giacigw1(config-if)#desc Broadwing Internet<enter>
Giacigw1(config-if)#ip address 45.127.254.10 255.255.255.252<enter>
Giacigw1(config-if)#ip access-group SerialIn in<enter>
Giacigw1(config-if)#ip access-group SerialOut out<enter>

Add the following as well:
No ip redirects
No ip unreachables
No ip proxy-arp
No cdp enable

Giacigw1(config-if)#interface serial0/1<enter>
Giacigw1(config-if)#desc Cable & Wireless Internet <enter>
Giacigw1(config-if)#ip address 45.254.250.34 255.255.255.252<enter>
Add the following as well:
Ip access-group SerialIn in
Ip access-group SerialOut out
No ip redirects
No ip unreachables
No ip proxy-arp

No cdp-enable

Next we configure our BGP routing process. This assumes that we have already coordinated with our ISPs regarding the details of this configuration.

Giacigw1(config-if)#router bgp 64512<enter>
Giacigw1(config-router)#bgp log-neighbor changes<enter>
Giacigw1(config-router)#network 45.10.1.0 mask 255.255.255.0<enter>
We specify a mask for the network we intend on advertising via BGP otherwise the router OS defaults to a Class A mask and we end up advertising 45.0.0.0/8.
Giacigw1(config-router)#Neighbor 45.127.254.9 remote-as 6395<enter>
Giacigw1(config-router)#Neighbor 45.127.254.9 password h3lL0z3Nv2<enter>
Giacigw1(config-router)#Neighbor 45.127.254.9 filter-list 86 in<enter>
This defines which routes we will accept from our BGP neighbor. We have yet to define this list, but can still assign it as our filter.
Giacigw1(config-router)#Neighbor 45.127.254.9 filter-list 80 out<enter>
This defines which routes we will advertise to our neighbor.

Next we configure our BGP process for our second ISP.
Giacigw1(config-router)#Neighbor 45.254.250.33 remote-as 3561<enter>
Giacigw1(config-router)#Neighbor 45.254.250.33 password ff998sb2K9<enter>
Giacigw1(config-router)#Neighbor 45.254.250.33 filter-list 85 in<enter>
Giacigw1(config-router)#Neighbor 45.254.250.33 filter-list 80 out<enter>
Giacigw1(config-router)#end<enter>
This takes us back to global config mode.
Next we'll configure our various ACLs and filters.

These are specialized filters for our BGP Configuration.

Giacigw1(config)#ip as-path access-list 80 permit ^$<enter>
Giacigw1(config)#ip as-path access-list 85 permit ^3561_[0-9]$<enter>*
Giacigw1(config)#ip as-path access-list 85 permit ^6395_[0-9]$<enter>*

Giacigw1(config)#ip access-list extended SerialIn<enter>

Names Access Control Lists are easier to work with because they can be descriptive. This ACL applies to the 'inbound serial interface'. In some cases named ACLs are even required. (For example, when using reflexive access-lists). Note also that named access –lists are case-sensitive. SerialIn is a different ACL than Serialin.

Giacigw1(config)#

Note: It's typically not very practical to enter ACLs in manually. For one reason, rules are added to the end of the ACL. So, for example, if you mistakenly omitted line 15 of 21 in your ACL you could not simply go back and insert it, you would actually have to re-type lines 1-14 first, type line 15, then 16-21 again. This is where a TFTP server comes in handy. The TFTP Server needs to be secure so that there is no unauthorized access to the router config files. Then you would make changes to your ACLs by editing the configuration file and then saving it to the appropriate directory on the TFPT Server. On the router, you negate the existing ACLs, then copy the configuration file from the tftp server.

For example:

Giacigw1#write mem<enter>
First, save any changes we've made.
Giacigw1#copy running-conf tftp<enter>
Address or name of remote host [ ]? 45.10.1.4 <enter>
This is our defined tftp/syslog server.
Destination filename [giacigw1-confg]? <enter>
Enter accepts the default file name, which the router has determined for us based on the router hostname.
At this point the copy will take place.

Then we edit the file in our favorite text editor, saving it when we are done. It is assumed that we made changes to our ACLs.

Now on our router, we negate the existing ACLs. Remember, the ACLs are still assigned to our interfaces, so all we need do is:
Giacigw1#conf t<enter>
Giacigw1(config)#no ip access-list extended SerialIn <enter>
Repeat as necessary for other named ACLs.

Giacigw1(config)#exit <enter>

Next we'll copy our newly modified configuration file from the TFTP Server to the router.
Giacigw1#copy tftp running-conf <enter>
We'll copy into our running-config, which means the changes will take place immediately. Note that we are actually merging our configuration files in this case.
Giacigw1#
Address or name of remote host [ ]? 45.10.1.4 <enter>
Source filename [ ] ? giacigw1-confg <enter>
Destination filename [running-config] <enter>
The file will be copied into running-config.

We can use this method for most of the configuration, however it is commonly used when modifying ACLs.

Next we'll continue with our the manual configuration of our router.
Giacigw1(config)#ip access-list extended SerialIn<enter>

Enter the ACL entries as they are listed in our router security policy. Example:

Giacigw1(config)# deny ip 127.0.0.0 0.255.255.255 any log-input<*enter*>
Giacigw1(config)# deny ip 224.0.0.0 15.255.255.255 any log-input<*enter*>
Giacigw1(config)# deny ip 10.0.0.0 0.255.255.255 any log-input<*enter*>
Giacigw1(config)# deny ip 192.168.0.0 0.0.255.255 any log-input<*enter*>
Giacigw1(config)# deny ip host 0.0.0.0 any log-input<*enter*>
Giacigw1(config)# permit tcp any 45.10.1.64 0.0.0.63 eq www reflect inbound<*enter*>
Giacigw1(config)# permit tcp any 45.10.1.64 0.0.0.63 eq 443 reflect inbound<*enter*>

...and so forth. Repeat this for each defined ACL.

A note on bit-masking differences in the Cisco IOS.

It is often confusing when you first start working with Cisco Access Control lists because you may be familiar with assigning IP addresses to interfaces using the follow format:

Giacigw1(config-if)#ip address 10.100.1.1 255.255.255.0 <enter>

Which means, "10.100.1.1 is my ip address" and "255.255.255.0" is my netmask.

However, if you create an ACL such as:
Deny ip any 172.20.1.0 255.255.255.0
What really happens is, any packet destined for the host 172.20.1.0 is denied, but a packet destined for 172.20.1.1-255 doesn't match this rule. The proper way to write this ACL would be:
Deny ip any 172.20.1.0 0.0.0.255
The last octet is considered a wildcard and so it will match on any value, so long as the first three octets match the source, 172.20.1.

Giacigw1(config)#logging facility local5 <enter>
Giacigw1(config)#logging source-interface FastEthernet0/0 <enter>
Giacigw1(config)#logging 45.10.1.4<enter>

This configures logging to our syslog server (45.10.1.4). Specifying the logging facility local5 will allow use to log our messages to a custom log file. We'll define this on our syslog server using the syslog.conf configuration file. We also specify source-interface FastEthernet0/0 so that we can configure our

firewall security policy to only accept syslog messages from a specific address (that of fa0/0 on our router).

Next we configure our NTP settings.

Giacigw1(config)#ntp server 128.118.46.3 version 2 source FastEthernet0/0<enter>
Giacigw1(config)#ntp server 138.39.7.20 version 2 source FastEthernet0/0 prefer<enter>

When you are all done entering in the security policy, exit and save the configuration.

Giacigw1(config)#exit <enter>
Giacigw1# wr me <enter>

Wr me is short for *write memory*, recall what we discussed about not having to enter complete commands.

Now that the configuration is saved, it is prudent that we reboot our router so that we may observe the boot process through the console connection. Since this is our initial configuration of the router, there should be no production issues in doing this.

Giacigw1#reload <enter>
Proceed with reload? [confirm]

Type y and the router will reload (reboot). Now keep your eyes open for any potential problems. Generally these will be in the form of syntax errors (if we've loaded the configuration from a tftp server usually)

Assignment 3 - Validation of GIAC's Primary Firewall Policy

The validation process need not be overly complex. This is not a vulnerability assessment. We have a certain expectation of our firewall based on the policy we have designed. The validation process should either confirm our expectations or reveal area's over looked.

For our validation we will use a port scanning tool, NMAP, since it is free and widely used and trusted.

Additionally we will be using a packet crafting tool formerly called Rafale X, which I believe may now be called Engage Packet builder. We'll be using this tool to test our anti-spoofing configuration primarily.

We will also be using the Checkpoint Firewall log viewer for our analysis. Since our policy is to log everything, we'll be generating a lot of log entries.

We will scan the firewall from 7 perspectives.

1. From the Internet to GIAC's DMZ and Screened Subnets.
2. Internal to Screened Subnets and DMZ.
3. Screened Public Network to Internet.
4. Screened VPN Network to Internet.
5. Screened VPN & Public Networks to Internal.
6. Screened Public Net to Screened VPN Net
7. Screened VPN Net to Screened Public

Because we often allow only a single IP to access another IP for a particular service, our validater will ideally have to assume the IP address of the source system. This may not be practical in a production environment. It should be sufficient to show through the firewall logs that the appropriate system was allowed through were necessary, and all others were denied.

We will also test the anti-spoofing configuration of the firewall (using Rafale X). We'll be assuming various knowledge of the firewall policy. For instance, we know that we allow internal systems to access systems on the screened Subnets via Microsoft Terminal Services (TCP 3389) and VNC (TCP 5900).

We don't want to disrupt GIAC's business operations while performing our validation. While we don't anticipate impacting performance much during our validation, we are concerned about the particular hosts that we may be scanning. Some vendor's have had issues with their IP stacks in regards to port scanning. On particular versions of Compaq Tru64 Unix, for example, a port scan could possibly hang the inetd service, thus denying new telnet connections to the system. Therefore we have coordinated with GIAC's technical staff and

determined that our work is best performed after 9PM on a Saturday evening. We anticipate that it will take us roughly 16 man hours to position our systems and perform our various scans, and another 8 hours worth of analysis and report preparation. The validation should give GIAC some degree of confidence in the chosen architecture.

While we anticipate the risks to be minimal, we will coordinate with GIAC's technical staff to make sure they are available during the time we'll be conducting our scans. Since we do not intimately know all of GIAC's systems there's also a risk that we'll have a negative performance impact on some network element or host system. We have explained these risks to the customer. We believe that having the knowledge that the firewall policy is correctly implemented is justification for any risks that we may encounter. This is specific to GIAC's situation. To another customer, with different business requirements, these risks might be unacceptable, it which case we would potentially need to alter our approach to something that would invariably take more time. (For instance, if we spread our scans out over a significantly longer period of time.)

Analysis of Scans

1. From the Internet to GIAC's DMZ and Screened Subnets.

   We expect to see certain traffic allowed through the firewall to certain hosts on our Screened Public Networks (http, https, and smtp). We expect to see VPN related traffic to our Firewall (DMZ). We don't expect to see anything allowed through to our Screened VPN Network.

   Before performing this scan, we will drop any filters on our routers. This will give a truer representation of what the firewall itself is doing for us. This will also facilitate spoofing as our routers play a role in our anti-spoofing configuration. Don't forget to re-assign the access-groups after done with the Internet to GIAC part of the validation.

```
In CLI config mode, enter the follow commands:
Interface fasteth0/0
        No ip access-group in
        No ip access-group out
Interface serial0/0
        No ip access-group in
        No ip access-group out
Interface serial0/1
        No ip access-group in
        No ip access-group out
```

Anti-Spoofing Validation

With our validation laptop configured as a random Internet host (38.161.210.150), we used the Rafale X packet crafter to send spoofed packets, pretending to come from 172.24.1.102, to our web server, 45.10.1.70, using TCP Port 3389, since we know that we allow Internal hosts to use port 3389 to connect to the Screened Public network. The result, as seen in the Checkpoint Log, is that our spoofing attempt was detected and dropped. Additionally, as the last entry shows, we attempted to send a packet with the ACK bit set, from our Internet host and the Checkpoint correctly determine it to be out of state.

```
"89091" "24Dec2003" "1:50:28" "VPN-1 & FireWall-1" "eth1" "tautog.giac.com"
"Log" "Drop" "3389" "172.24.1.102" "45.10.1.70" "tcp" "" "1027" "" "message_info:
Address spoofing; "
"89097" "24Dec2003" "1:53:28" "VPN-1 & FireWall-1" "eth1" "tautog.giac.com"
"Log" "Drop" "3389" "172.24.1.102" "45.10.1.70" "tcp" "" "1027" "" "message_info:
Address spoofing; "
"89098" "24Dec2003" "1:55:13" "VPN-1 & FireWall-1" "eth1" "tautog.giac.com"
"Log" "Drop" "http" "38.161.210.150" "45.10.1.70" "tcp" "" "1027" "" "th_flags: 10;
message_info: TCP packet out of state; "
```

2. Internal to Screened Subnets and DMZ.

When we first performed our scan, as follows, we had to specify –P0 so as not to attempt to ping hosts first. Without this, we would never even get to the actual connect scan. We attempted first to do a standard –sT TCP Connect Scan.

```
C:\Program Files\NMapWin\bin>nmap -sT -P0 -T 5 45.10.1.64/27

Starting nmap V. 3.00 ( www.insecure.org/nmap )
Skipping host   (45.10.1.64) due to host timeout
Skipping host   (45.10.1.65) due to host timeout
Skipping host   (45.10.1.66) due to host timeout
Skipping host   (45.10.1.67) due to host timeout
Skipping host   (45.10.1.68) due to host timeout
Skipping host   (45.10.1.69) due to host timeout
Skipping host   (45.10.1.70) due to host timeout
Skipping host   (45.10.1.71) due to host timeout
Skipping host   (45.10.1.72) due to host timeout
Skipping host   (45.10.1.73) due to host timeout
Skipping host   (45.10.1.74) due to host timeout
Skipping host   (45.10.1.75) due to host timeout
Skipping host   (45.10.1.76) due to host timeout
^C
C:\Program Files\NMapWin\bin>
```

I believe that we received the host timeout due to the equipment we used for our validation lab. Our Screened Public network contained a single host configured with the appropriate IP addresses as aliases. This host was a 486 DX2-66 running FreeBSD 4.7. Apache 2.0 was configured to Listen on the applicable addresses and ports. Since we are using our firewall as little more than a glorified stateful packet filter, this would seem an acceptable 'lab' scenario as it will provide for the basic TCP connectivity that we've defined in our policy, and it gives us something on the other side to respond to our scans. When we look at our firewall log entries while scanning, we see that our scans are being appropriately dropped.

C:\Program Files\NmapWin\bin>nmap –n –p "1-" –P0 45.10.1.71/32
(-n prevents nmap from trying to do name resolution, which is highly beneficial in our ad hoc lab environment)
(-p "1-" indicates to scan ports 1 and up (1 – 65535))

Sample selection of Checkpoint Firewall Log (exported into a text file)

```
"16569" "24Dec2003" "0:07:24" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "62735" "172.24.1.102" "45.10.1.71" "tcp" "20" "45387" "" ""
"16570" "24Dec2003" "0:07:24" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "65343" "172.24.1.102" "45.10.1.71" "tcp" "20" "45387" "" ""
"16571" "24Dec2003" "0:07:24" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "54348" "172.24.1.102" "45.10.1.71" "tcp" "20" "45387" "" ""
"16572" "24Dec2003" "0:07:24" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "8652" "172.24.1.102" "45.10.1.71" "tcp" "20" "45387" "" ""
"16573" "24Dec2003" "0:07:24" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "45056" "172.24.1.102" "45.10.1.71" "tcp" "20" "45387" "" ""
"16574" "24Dec2003" "0:07:24" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "17338" "172.24.1.102" "45.10.1.71" "tcp" "20" "45387" "" ""
"16575" "24Dec2003" "0:07:24" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "45706" "172.24.1.102" "45.10.1.71" "tcp" "20" "45387" "" ""
"16576" "24Dec2003" "0:07:24" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "57327" "172.24.1.102" "45.10.1.71" "tcp" "20" "45387" "" ""
"16577" "24Dec2003" "0:07:24" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "6729" "172.24.1.102" "45.10.1.71" "tcp" "20" "45387" "" ""
"16578" "24Dec2003" "0:07:24" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "22706" "172.24.1.102" "45.10.1.71" "tcp" "20" "45386" "" ""
"16579" "24Dec2003" "0:07:24" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "34413" "172.24.1.102" "45.10.1.71" "tcp" "20" "45386" "" ""
"16580" "24Dec2003" "0:07:30" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "22706" "172.24.1.102" "45.10.1.71" "tcp" "20" "45387" "" ""
"16581" "24Dec2003" "0:07:30" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "34413" "172.24.1.102" "45.10.1.71" "tcp" "20" "45387" "" ""
"16582" "24Dec2003" "0:07:30" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "43128" "172.24.1.102" "45.10.1.71" "tcp" "20" "45386" "" ""
```

```
"16583" "24Dec2003" "0:07:30" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "42574" "172.24.1.102" "45.10.1.71" "tcp" "20" "45386" "" ""
"16584" "24Dec2003" "0:07:30" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "43139" "172.24.1.102" "45.10.1.71" "tcp" "20" "45386" "" ""
"16585" "24Dec2003" "0:07:30" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "31561" "172.24.1.102" "45.10.1.71" "tcp" "20" "45386" "" ""
"16586" "24Dec2003" "0:07:30" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "31810" "172.24.1.102" "45.10.1.71" "tcp" "20" "45386" "" ""
"16587" "24Dec2003" "0:07:30" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "38531" "172.24.1.102" "45.10.1.71" "tcp" "20" "45386" "" ""
"16588" "24Dec2003" "0:07:30" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "13432" "172.24.1.102" "45.10.1.71" "tcp" "20" "45386" "" ""
"16589" "24Dec2003" "0:07:30" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "35538" "172.24.1.102" "45.10.1.71" "tcp" "20" "45386" "" ""
"16590" "24Dec2003" "0:07:30" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "43521" "172.24.1.102" "45.10.1.71" "tcp" "20" "45386" "" ""
"16591" "24Dec2003" "0:07:30" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "54963" "172.24.1.102" "45.10.1.71" "tcp" "20" "45386" "" ""
"16592" "24Dec2003" "0:07:30" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "31750" "172.24.1.102" "45.10.1.71" "tcp" "20" "45386" "" ""
"16593" "24Dec2003" "0:07:30" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "21785" "172.24.1.102" "45.10.1.71" "tcp" "20" "45386" "" ""
"16594" "24Dec2003" "0:07:30" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "42499" "172.24.1.102" "45.10.1.71" "tcp" "20" "45386" "" ""
"16595" "24Dec2003" "0:07:36" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "43128" "172.24.1.102" "45.10.1.71" "tcp" "20" "45387" "" ""
"16596" "24Dec2003" "0:07:36" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "42574" "172.24.1.102" "45.10.1.71" "tcp" "20" "45387" "" ""
"16597" "24Dec2003" "0:07:36" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "43139" "172.24.1.102" "45.10.1.71" "tcp" "20" "45387" "" ""
"16598" "24Dec2003" "0:07:36" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "31561" "172.24.1.102" "45.10.1.71" "tcp" "20" "45387" "" ""
"16599" "24Dec2003" "0:07:36" "VPN-1 & FireWall-1" "eth0" "45.10.1.2" "Log"
"Drop" "31810" "172.24.1.102" "45.10.1.71" "tcp" "20" "45387" "" ""
```

We know that the only defined objects on our Screened Public Network have the following IP Addresses:

45.10.1.64	Network Address
45.10.1.65	firewall eth2
45.10.1.70	www.giac.com AKA e-fortune.giac.com
45.10.1.71	smtp.giac.com
45.10.1.72	eCookie.giac.com
45.10.1.73	suppliers.giac.com
45.10.1.95	Broadcast Address

Because we are doing a validation as opposed to a vulnerability assessment or an audit, we are not going to waste our time scanning any systems that do not exist (45.10.1.64/27 for example)

We performed several scans:
1. TCP Connect scan for all ports.
2. –sU UDP Scan
3. –F scan (only ports listed in nmap-services file)

An example of –F scan:

```
C:\Program Files\NMapWin\bin>nmap -n -F -P0 45.10.1.70/32

Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on  (45.10.1.70):
(The 1144 ports scanned but not shown below are in state: filtered)
Port       State       Service
21/tcp     closed      ftp
80/tcp     open        http
443/tcp    open        https
1352/tcp   closed      lotusnotes
3389/tcp   closed      ms-term-serv
5900/tcp   closed      vnc

Nmap run completed -- 1 IP address (1 host up) scanned in 226 seconds
```

This scan gives an accurate representation of our firewall policy. Because host 45.10.1.70 is NOT listening on TCP Port 21 (or 1352, 3389, and 5900), NMAP reports those ports as closed, but not filtered. On the wire we see the host issuing a TCP Reset, which is the expected behavior, as seen in the following tcpdump output of the scan:

```
23:16:49.022771 IP 172.24.1.102.45996 > 45.10.1.70.21: S 1672240388:1672240388(0 ) win 3072
23:16:49.025287 arp who-has 172.24.1.102 tell 172.24.1.1 23:16:49.025327 arp reply 172.24.1.102 is-at
0:0:86:44:8c:a2
23:16:49.025634 IP 45.10.1.70.21 > 172.24.1.102.45996: R 0:0(0) ack 1672240389 w in 0
```

This behavior does, however, provide information that we may not wish to provide, and can be used to facilitate OS fingerprinting and detection. To mitigate this, we could add further granularity to rulebase, but because this is traffic from our Internal private networks to our screened Subnets, we will assume a certain level of trust. We wouldn't want someone on an un-trusted network such as the Internet to be receiving these resets.

3. Screened Public Network to Internet.

The only allowed outbound traffic from the Screened Public Network is SMTP from 45.10.1.71, which our firewall logs confirm. Because our validater has a different IP address, SMTP is not be allowed out for the validater.

4. Screened VPN Network to Internet.
   No traffic is allowed from the Screened VPN Network to the Internet, accept when using Client Authentication for members of the VPN "Patchers" Group. Firewall logs confirm this.

5. Screened VPN & Public Networks to Internal.
   The only allowed traffic should be HTTP to av.inside.giac.com for Anti-Virus Updates and DNS Queries (UDP 53) to ns1.inside.giac.com. This too is confirmed by our firewall logs.

6. Screened Public Net to Screened VPN Net
   The only allowed traffic from Screened Public to Screened VPN should be TCP Port 6500 from www.giac.com and ecookie.giac.com. The firewall logs confirm this.
7. Screened VPN Net to Screened Public
   No traffic should be allowed from Screened VPN to Screened Public Network. Again, our firewall logs confirm this.

Results

Because we disabled or modified most of the Checkpoint default global properties, our validation scans were predictable. This is by no means an indication that our firewall is 'in-penetrable'. It is but one component in a broader security architecture. As previously mentioned, our firewall is basically a glorified, stateful packet filter. It will not prevent our Public web servers from being infected by self-propagating worms such as Code-Red or Nimda, nor from malicious individuals using similar HTTP exploits. (Or exploits surrounding any of our allowed services and particular Operating Systems/Application Software) It is assumed that GIAC is faithful updating security patches. However, we could make improvements on our architecture by implementing certain devices that handle application layer filtering. This could be a proxy server, for instance, or even configuring Checkpoint's various security servers present with their firewall software. Another alternative for GIAC would be to implement an Intrusion Detection systems with session reset capabilities. Such a device could be configured to detect application layer anomalies and issue reset packets to disrupt potentially malicious connections. This functionality should be used with extreme care because it is, for all practical purposes, still in its infancy, and could result in disrupting all manner of legitimate traffic if the particular device were not finely tuned. There are technical and financial ramifications to such alternatives including increased expertise required for GIAC, as these systems will require continuous maintenance to be effective.

Assignment 4 Design under fire

The purpose of this assignment will be to analyze another architecture "from the perspective of a malicious attacker", by researching specific components and vulnerabilities. The exercise should facilitate a further understanding of Defense-in-Depth principals by examining the architecture in detail and looking for potential weaknesses.

We chose Andrew Lemick's GCFW Practical for our design under fire assessment. I wanted to learn about some of the pros and cons of various other firewalls and security architectures since I am primarily familiar with Checkpoint. I have limited experience with PIX and Netfilter, but not enough to feel comfortable designing an architecture with them. Since the 'best' firewall is usually a matter of opinion, it's best for security analysts to be versed in firewall technology in addition to any particular vendor solution.

Andrew Lemick's practical can be found at:

http://www.giac.org/practical/GCFW/Andrew_Lemick_GCFW.pdf

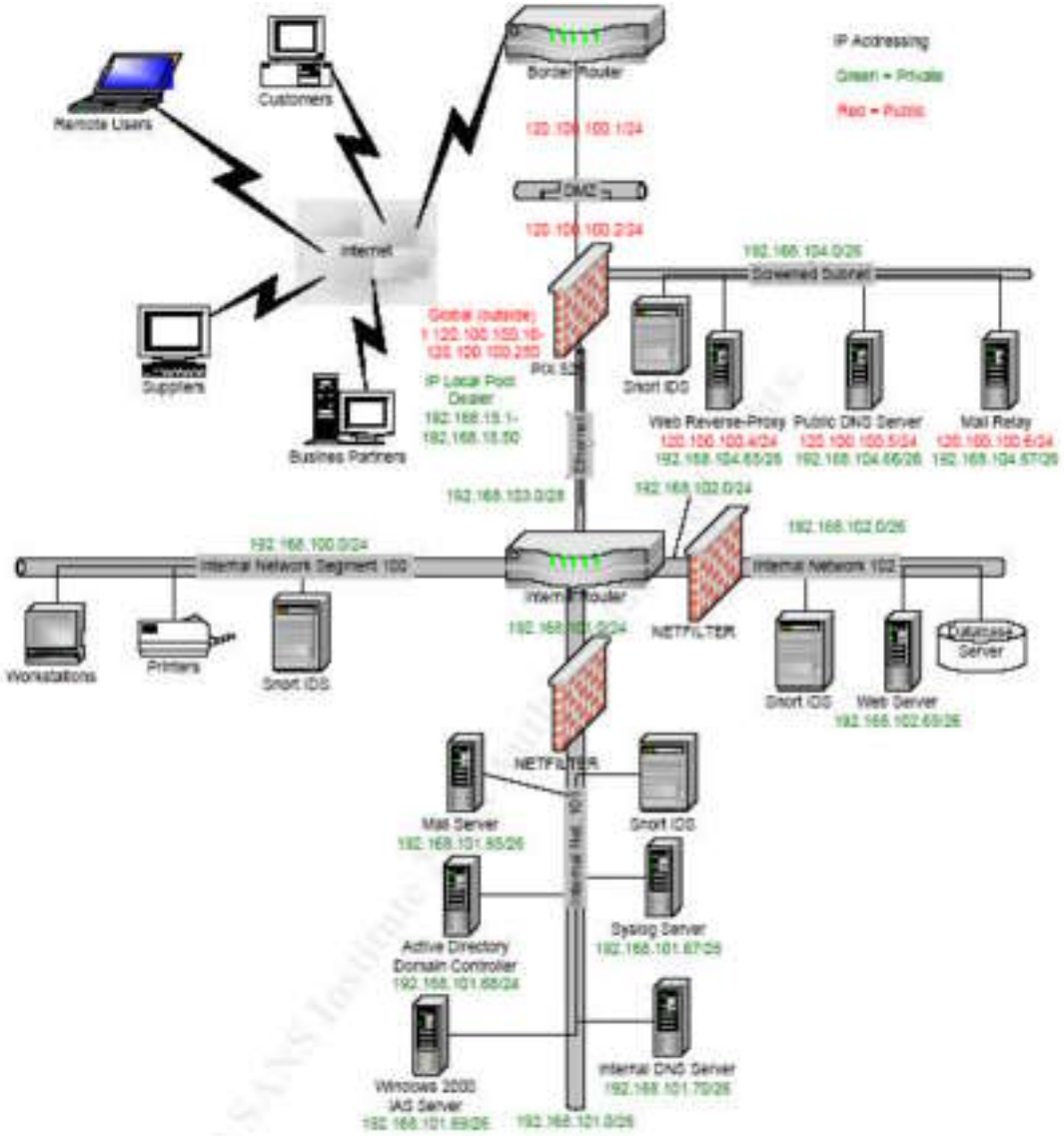
For this assignment, we need to design and detail the following:

1. An attack against the firewall itself.

2. A distributed Denial-of-Service Attack against some component in the architecture.

3. An attack plan to compromise an internal system.

Graphic from Andrew Lemick's GCFW Practical



Andrew uses a Cisco PIX 525 as the primary external firewall for his design. The PIX is running firmware version 6.3
The border router in this configuration is a Cisco 3620 and is running IOS c3620-jo3s56I-mz.121-18.

An attack against the firewall itself

We've found several potential vulnerabilities in Cisco PIX firmware 6.3

1. OpenSSL ASN.1 Parsing Vulnerabilities
http://www.securityfocus.com/bid/8732
"Multiple vulnerabilities were reported in the ASN.1 parsing code in OpenSSL. These issues could be exploited to cause a denial of service or to execute arbitrary code." [18]

2. Cisco PIX ICMP Echo Request Network Address Translation Pool Exhaustion Vulnerability
http://www.securityfocus.com/bid/8754 [19]

This reported vulnerability involves exposing PIX global IP address pools to ICMP traffic. It is reported that this may cause a DoS condition.

- We do not currently know what types of ICMP traffic are involved.
- Does the security architecture in question contain provisions, which might allow us to execute an attack based on this vulnerability?

The firewall configuration indicates that global pools are configured for the outside and screened interfaces:

Global (outside) 1 120.100.100.10-120.100.100.250 netmask 255.255.255.0
Global (screened) 1 192.168.104.70-192.168.104.75 netmask 255.255.255.192

We also know that the border router potentially allows some ICMP through:

Access-list 101 permit tcp any 120.100.100.0 0.255.255.255 established

This allows some ICMP packets through to the same subnet that our 'Global (Screened) ' Pool is defined on.

Some other potential vulnerabilities:

3. http://www.cisco.com/warp/public/707/cisco-sa-20031215-pix.shtml
To directly quote the Cisco advisory,
"The Cisco PIX firewall crashes and reloads while processing a received SNMPv3 message when **snmp-server host <if_name> <ip_addr>** or **snmp-server host <if_name> <ip_addr> poll** is configured on the Cisco PIX firewall. This happens even though the Cisco PIX firewall does not support SNMPv3.

A Cisco PIX firewall configured to only generate and send traps using the **snmp-server host <if_name> <ip_addr> trap** command is not vulnerable."[17]

- Is Andrew's architecture susceptible to this vulnerability? What measures are in place to mitigate? Neither the PIX nor the border router are configured to use SNMP or allow such traffic through, so it is doubtful that this is something we could exploit.

A distributed denial of service attack:

Compromising 50 Cable/DSL systems could be accomplished relatively easy by sending a Trojan via email with a misleading subject. We would need to harvest email addresses from potentially unwitting sources, for instance, various online 'Home Improvement Forums', 'Parenting' Forums, etc. There are many out there for the picking in which posters do not obscure their email addresses. We would send them an email with a misleading subject, perhaps masquerading as another user of the forum with a subject related to the specific forum. This will fool a certain percentage of users into executing the attachment. The attachment could be designed to indeed show something of content related but also install our Trojan behind the scenes.

- Some percentage of users could be exploited if they are running vulnerable versions of Outlook Express for example.

We'll assume that the Cable Modem hosts are using a personal firewall. However, they will still allow outbound HTTP access. As such we will configure our trojan to 'phone home' via HTTP and maintain a low-key but persistent outbound connection to our master host, similar to the way technologies such as GoToMyPC.com work.

We'll be judicious in our selection of harvested email addresses so that we choose a population segment that is less likely to notice our Trojan. We'll have a limited amount of time to exploit the systems and orchestrate our attack before our Trojan is discovered and detected by anti-virus software and other detection utilities. We are relying on our general selection of targets to help increase our chance of being undetected in the wild, however this is not foolproof. There are certainly people using the Home Improvement forum that my have the background to detect our activity, but the probability is less likely than if we targeted users on an Information Security forum, for example.

Our selected Trojan should allow some level of shell type access to our compromised hosts. For our attack, we'll attempt to use something called The WC Remote Administration Tool (v1.2b). Again, to be found on www.packetstormsecurity.com There are many choices. Our experience is limited with all of them so any might do.

From there we'll want to download an application to use to attack our target. www.packetstormsecurity.com is one site that contains many, many examples of such applications. For our purposes we'll use something called Syn Flooder Version 1.6, by "meto". This application runs on Windows XP. Here's the helpfile to give us an example of how easy it would be to use this application. [21]

Syn Flooder Version 1.6
==========

Syn floods fills backlog of the destination service's socket. This causes the service not to accept any more clients.

Note: This program is only tested on Windows XP.

I. USAGE
==========

syn.exe <victim> [options]

Options:
-S: Spoof host (0 is random (default))
-p: Comma separated list of dest ports (0 is random (default))
-s: Comma separated list of src ports (0 is random (default))
-n: Num of packets (0 is continuous (default))
-d: Delay (in ms) (default 0)

Example I: syn 217.155.32.170 -p 21,23,80,110

On this attack:

- Victim: 217.155.32.170
- Source IP: Random
- Destination ports: 21 + 23 + 80 + 110
- Source ports: Random
- Count: Continuous
- Delay: 0 ms (no delay between packets)

Example II: syn 217.155.32.170 -p 21,23,80,110 -s 42,63 -n 1

On this attack:

- Victim: 217.155.32.170
- Source IP: Random
- Destination ports: 21 + 23 + 80 + 110
- Source ports: Random
- Count: 1 * Please note that 1 count will transmit the syn packets from any source port to any destination port. This means 8 packets will be transmited with a 1 count on this attack.
- Delay: 0 ms (no delay between packets)

Once all of our compromised systems are prepared, and it is assumed that we've prepared them programmatically (because that is a lot of work otherwise!), we commence our attack.

We've chosen to attack the Reverse-Proxy Web Server (120.100.100.4) because, being a proxy server, it is already doing it's share of work. Knocking it offline will also disrupt access to the web server, which should have a negative impact on GIAC's business operations.

Though automated through our Trojan, the command we'll use to attack would look something like this:

```
syn 120.100.100.4 -p 80,443
```

This should send a continuous flood of packets with random source addresses. A syn flood works by beginning the three-way handshake of a tcp connection but never completing it (because the source address is spoofed). This uses up resources while the server waits to time-out the connection request. Assuming we have 50 compromised systems, each one sending 25 SYN packets a second, that's 75,000 SYN request over the course of a minute. Depending on the resources of the Reverse-Proxy Server, this could well be more than enough to know the system offline.

We would probably want to limit our attack. If it was sporadic, in other words, over almost as soon as it started, we might be able to wreak havoc on the organization for a longer period of time before they could figure out what was happening and take measure to mitigate it.

How could we prevent or mitigate this attack?

The Cisco IOS provides a feature known as TCP Intercept. "The TCP intercept feature helps prevent SYN-flooding attacks by intercepting and validating TCP connection requests." [10] TCP Intercept is an example of one such method that could be used to mitigate this type of attack.

- TCP Intercept is a feature that can operate in two different modes, Intercept Mode and Watch Mode. We could run TCP Intercept on the border router. It's a fairly configurable feature so it can be custom tailored to our environment to balance the need for protection and performance.
- The PIX firewall can also act as a TCP Intercept device.

An attack plan to compromise an internal system:

We choose the Internal Mail Server (192.168.101.65) as our selected target. This server is running Lotus Domino Version 6 on the Solaris 9 Operating System. This version of Lotus Domino, specifically the Lotus Domino Web Server, is reportedly vulnerable to "a buffer overflow via non-existent "h_SetReturnURL" parameter. [20]

Is this something we could exploit? We are making some broad assumptions. The author does not specifically say whether or not the Lotus Domino Web Server is running on the Internal Mail Server or not. It may be running by default. We have no way of knowing based on the documentation. It may be best to assume it is not running, in which case our attack would fail.

Another potential opportunity to exploit involves the fact that the PIX firewall is configured to allow SSH traffic from the border router to the administrator's workstation IP address.

Access-list acl-out permit tcp host 120.100.100.1 host 192.168.100.1 eq 22.

Since the administrator's PC in question is an internal system, could we potentially exploit it? SSH has had its share of vulnerabilities and exploits available. But in order to be able to compromise the administrator system, we first have to gain access to the border router.

The border router allows established connections on the inbound ACL.

Access-list 101 permit tcp any 120.100.100.0 0.255.255.255 established

We could experiment with specially crafted TCP packets in which the SYN bit and the ACK bit are set. We haven't discovered any known exploits for this behavior for this particular router / IOS version, but that doesn't mean there isn't one.

One thing to consider is that the aforementioned ACL is mis-configured. The wildcard mask should read 0.0.0.255, instead, in effect, it says, "allow any host to 120.any.any.any established." As long as the first octet matches (and the ACK bit is set) the packet should pass. There's not mention of what our serial interface addresses are to our ISP's. If these fell somewhere within the 120.x.x.x address space there might exist some opportunity for exploitation. Other wise this ACL entry is closely followed by:

Access-list 101 deny tcp any any range 21 23 log

Which limits our ability to exploit the router for any FTP, SSH or telnet vulnerabilities. The entry preceding that allows certain types of ICMP which have not been previously denied (redirect, echo, and mask-request have been denied). This might present an opportunity for passing some time of ICMP based traffic.

We should further investigate whether or not there is a way to exploit the ACL mechanism in such a manner as to by pass it (and subsequently exploit a more useful service such as SSH) If we can gain access to this router, it opens up a whole new world of possibilities based on that one entry which allows SSH into the Internal network.

References

[1] Chris Hoffmann, VLAN Security in the LAN and MAN Environment
http://www.giac.org/practical/GSEC/Chris_Hoffmann_GSEC.pdf

[2] Checkpoint linux minimum configs:
http://support.checkpoint.com/kb/docs/public/os/linux/pdf/linux_minimal_ng_fp2.pdf

[3] Lance Spitzner's Whitepapers
http://www.spitzner.net/

[4] Northcut, Stephen, Zeltser, Lenny, Winters, Scott, Frederick, Karen, Ritchey, Ronald. Inside Network Perimeter Security, Indianapolis: New Riders Publishing. 2003

[5]INTERNET PROTOCOL V4 ADDRESS SPACE
http://www.iana.org/assignments/ipv4-address-space

[6] AS Numbers Templates
http://www.arin.net/library/index.html#templates

[7] RFC 1918 - Address Allocation for Private Internets
http://www.faqs.org/rfcs/rfc1918.html

[8] Improving Security on Cisco Routers
http://www.cisco.com/warp/public/707/21.html

[9] Mark Wolfgang, Exploiting Cisco Routers: Part 2
http://www.securityfocus.com/infocus/1749

[10] Peter Davis & Associates, Configuring Cisco Denial of Service Security Features, Part 1
http://www.pdaconsulting.com/dospart1.htm

[11] Mark Wolfgang, Exploiting Cisco Routers (Part One)
http://www.securityfocus.com/infocus/1734

[12] Cisco IOS HTTP Configuration Arbitrary Administrative Access Vulnerability
http://www.securityfocus.com/bid/2936

[13] Cisco Regular Expression Syntax
http://www.cisco.com/univercd/cc/td/doc/product/atm/c8540/12_0/13_19/cmd_ref/appc.htm

[14] Peter Davis & Associates, Configuring Cisco Reflexive Access Lists
http://www.pdaconsulting.com/reflex.htm

[15] CERT® Advisory CA-2003-15 Cisco IOS Interface Blocked by IPv4 Packet
http://www.cert.org/advisories/CA-2003-15.html

[16] Checkpoint Policy Editor Helpfile
No link available.

[17] Cisco Security Advisory: Cisco PIX Vulnerabilities
http://www.cisco.com/warp/public/707/cisco-sa-20031215-pix.shtml

[18] OpenSSL ASN.1 Parsing Vulnerabilities
http://www.securityfocus.com/bid/8732

[19] Cisco PIX ICMP Echo Request Network Address Translation Pool Exhaustion Vulnerability
http://www.securityfocus.com/bid/8754

[20] Vulnerability Note VU#772817
http://www.kb.cert.org/vuls/id/772817

[21] Syn Flooder Home Page
http://metin.adnanmenderes.net.