Global Information Assurance Certification Paper

GIAC Certified Firewall Analyst Practical
Version 1.8

Stanley R. Yachera

03/25/2003

Table Of Contents

Abstract

This paper covers Practical Version 1.8 required for GIAC Certified Firewall Analyst. The paper is composed of the following assignments:

Assignment 1 – Security Architecture – Design of a network security architecture for GIAC Enterprises, an e-business which deals in the online sale of fortune cookie sayings.

Assignment 2 – Security Policy and Tutorial – Provide a security policy for the design in Assignment 1

Assignment 3 –Verify the Firewall Policy – Conduct a technical audit of the primary firewall as described in Assignments 1 and 2

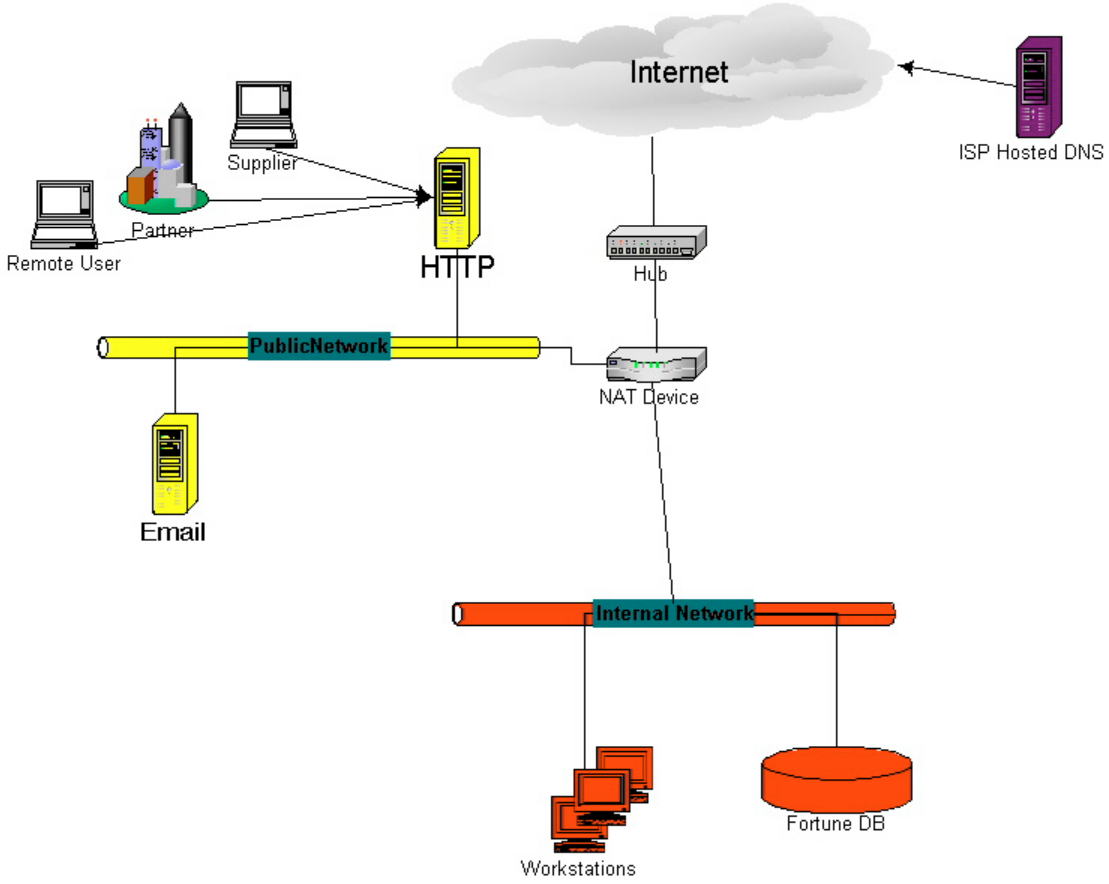
Assignment 4 – Design Under Fire – Select a previous Practical submitted in the past 6 months and design three attacks based on: An attack against the firewall itself, A denial of service attack, and an attack plan to compromise an internal system through the perimeter defense

1. Security Architecture

1.1 Introduction

GIAC Enterprises is a small company that deals in the online sales of online fortunes. GIAC is a very small company, total internal user base approaching 50 employees. A decision was made to undergo a complete restructuring of the Security Architecture of the Enterprise. Since the company is such a small operation, every effort had to be made to layout the architecture using a fiscally feasible model.

The previous regime implemented an architecture consisting of the following:



It can easily be seen the problems with this layout. There are no packet filtering devices anywhere to be found. Suppliers, Partners, and Remote users all connect to the same Web Server to conduct their business. Each of these entities should have separate access requirements, a layered architecture is sorely needed. The HTTP and Email server have Public IP addresses and are completely accessible (and routable) to the outside world. A NAT device was thrown in to give a false sense of security. No host based security was implemented anywhere, and most OS and software installations had little to no hardening. A decision was made to completely scrap this design.

GIAC has many business processes: Customers, Suppliers, Partners, Internal Employees, and Mobile Employees all have different Access Requirements. Utilizing logging, Encryption/VPN, Stateful and Stateless Firewalls, Intrusions Detection Systems, Host Based Security and other security tools we are able to implement a layered design to minimize our security risks.

1.2 Access Requirements:

Customers:

Business Operations:

Customers can browse company information, purchase fortunes, and browse account information online. All general GIAC company information is accessible via a web server for customers, as well as the outside world. All transactions (purchasing and browsing account information) are handled by a secure web server.

What Is Really Going On:

Customers access two web servers on the DMZ network. Both are Redhat 8.0 based servers running apache and SSHv2. Browsing company information happens via an Apache http based server running Apache 2.0.43 and IP Tables 1.2.7 blocking all access except to port 80 for HTTP and 22 for SSH for internal administration. All transactions are made by connecting to a SSL enabled Apache 2.0.43 server (mod_ssl) blocking all access except to port 443 for HTTPS and 22 for SSH for internal administration. Since we are taking secure orders, an encrypted 128 bit SSL connection will be made between the Web Browser and server. We will use Verisign to receive a certificate. Each customer will have a user name and password on our Sales and SubFortune MYSQL databases residing on our Shared DB Network. When purchasing fortunes, or browsing account information, a secure 128 bit SSH connection is made between the HTTPS server and the databases. A custom web application on the HTTPS server uses the MYSQL API to tunnel all traffic through the SSH connection between the HTTPS server and the databases. This allows us to ensure all our database data, our livelihood, is encrypted when traversing different security zones. All activity is logged to a SYLOG server residing within a separate security zone.

Suppliers:

Business Operations:

GIAC has a vast array of suppliers, from all over the country. The amount of suppliers warranted the upload server reside in an accessible area. Suppliers are given a comma delimited text format to adhere to when creating their fortunes. They digitally encrypt the fortunes via PGP and digitally sign them. They then upload them to a secure SSH server residing in our network or email them via a PGP digitally encrypted and signed message.

The messages then go through a rigorous testing period by a trained team of psychics to ensure the value of their message. Only then are the suppliers paid, and the fortunes submitted for production.

What Is Really Going On:

Since GIAC has such a wide array of suppliers, the decision was made to place the server on the DMZ. Suppliers and Customers are both trusted close to the same level. The SSH server is a Red Hat 8.0 based system running SSHv2. The big difference between a Customer and a Supplier is while the supplier user base is extensive, it is limited enough to allow us to restrict by IP and/or domain of their ISP's via the AllowHosts directive in the SSH configuration as well as limiting their access to only SSH through IPTables 1.2.7 to port 22, which is also run on the server. Each supplier is also given a username and password for the server. The other types of access is by internal IT staff SSH access for administration and the SubFortune DB. Every day at 5:00 PM EST. time a 'bot' on the SubFortune DB server creates a SSH connection to the SSH server, and using the MYSQL API pulls the day's fortunes across and puts them in a temporary 'Holding' table. The next day, internal employees then access them, put them through review, and when ready submit them for production in the SubFortune DB. All activity is logged to a SYLOG server residing within a separate security zone.

Partners:

Business Operations:

The company has recently partnered with another company in Belize (apparently the fortune cookie market there is booming). Belize Fortunes handle translating our fortunes into Belizean and reselling them. Partners need direct access to the SubFortune.

What Is Really Going On:

Belize Fortunes need access to the SubFortune DB to download and translate the production fortunes. A decision was made, with out question, we wanted this communication to be completely encrypted to hinder sniffing our valuable fortunes. A decision was also made; we did not want an 'Always On' connection. The partner connects using the Cisco VPN client version 3.6 to our Cisco Pix firewall 515 with PIX IOS 6.2. Due to recent relaxation on encryption export restrictions, we are utilizing a 128 bit 3-DES connection. The VPN client software was installed on their PC, and our profile was loaded. The keys are rotated monthly, the 1st of every month one of our staff members steps them through manually inserting the new keys. For security implications (say a lost laptop) another layer of security was added. A Red Hat 8.0 linux server running FREERADIUS 0.8.1 was implemented to provide another layer of authentication (FREERADIUS is a variant of the Cistron RADIUS server.). After the Partner initiates the VPN tunnel, the PIX then sends an authentication request to the Radius server, at which time the user is prompted for a user name and password. The user names and passwords reside in the Shadowed User DB on the server. Through IP Tables and the hosts.allow file, only the internal interface

on the PIX is allowed to connect to this server. For obvious security issues, our internal IT staff has to physically sit at the machine to add new users. After the tunnel is established the user then fires up a custom application that initiates a SSH tunnel to the database and then utilizes the MYSQL API/Client to pull the files down. All activity is logged to a SYLOG server residing within a separate security zone.

GIAC Internal Employees:

Business Operations:

GIAC employs a small number of internal employees. GIAC employees internally (inside network) access file sharing, an internal exchange server, and MYSQL databases via client applications residing on a Windows 2000 machine running Terminal Services and their desktop. GIAC employee's outbound access includes MYSQL client applications, web browsing, and DNS to access our ISP maintained DNS servers (we have a very small IT staff, we are just selling fortunes you know). In addition to the above requirements, our internal systems administrators (all 2 of us) require outbound telnet to administer routers, outbound SSH to administer servers and the PIX, and outbound ftp to download software, software updates, and most notably software security patches.

What Is Really Going On:

GIAC employs a small number of internal employees. Currently they have outbound Internet access to http and https for messing around on the Internet, I mean doing research (Talks have started about implementing a Squid proxy server with a combination of Calamari and Squid for user based tracking). They also connect to three databases:

The Fortune DB – This database is a Redhat 8.0 machine running MYSQL. It resides directly on the internal network; relying on host based IPTables access control. As the company expands, the thought is to add an additional NIC to internal firewall to better control access. Because speed is a necessity with the amount of processing being done internally, internal user connect directly to the database using the MYSQL client.

The SubFortune DB - This database is a Redhat 8.0 machine running MYSQL. It resides on the shared data base network. This database is a special subset of the SubFortune DB. Because speed is a necessity with the amount of processing being done internally, internal user connect directly to the database using the MYSQL client.

The Sales DB - This database is a Redhat 8.0 machine running MYSQL. It resides on the shared data base network. This database is used for comparing orders against inventory. Because speed is a necessity with the amount of processing being done internally, internal user connect directly to the database using the MYSQL client.

Email services are provided via an internal Microsoft Exchange 2000 server. We wanted the advanced usability of exchange (message collaboration, advanced calendaring..) but did not want the risk associated with having an external

Exchange server. To alleviate these concerns, a bastion host was placed in the DMZ network. This Redhat 8.0 server acts as a bastion host, only SMTP is allowed to flow between the two machines. The external server is running Postfix 2.0 with a combination of Spam Assassin to save some of our precious bandwidth.

GIAC Mobile Employees:

Business Operations:

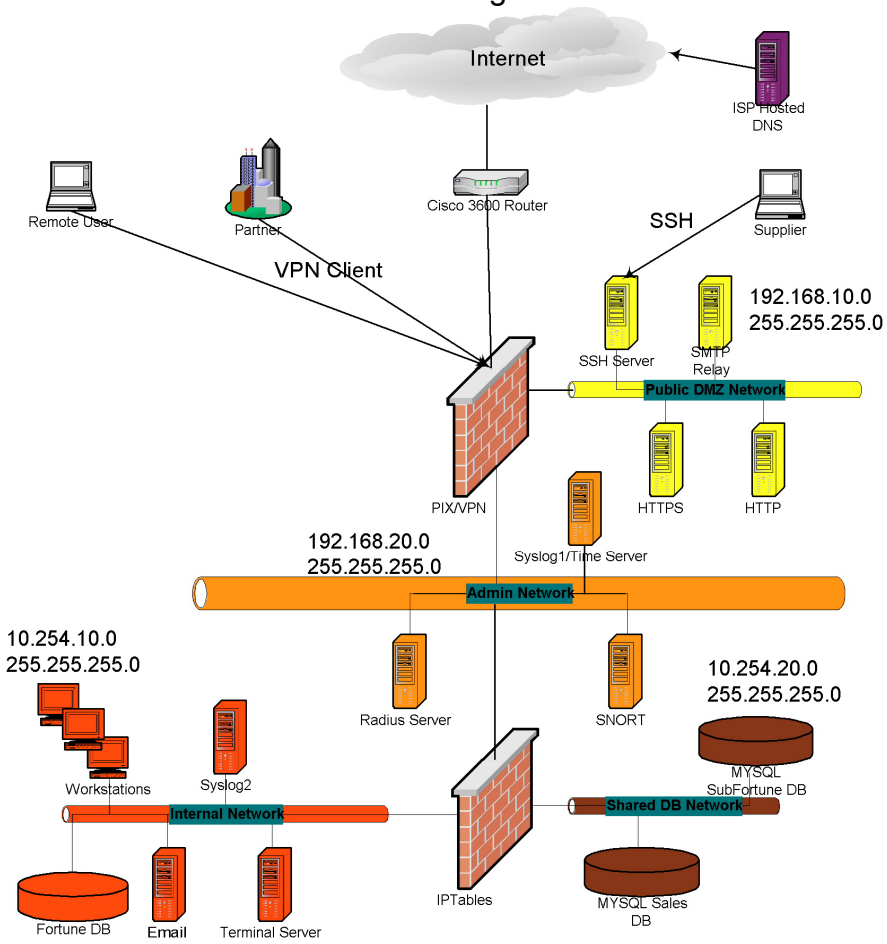
GIAC has a very small number of employees that require access to internal resources via the Internet. These are very high-level management types, which require instant access to all resources they would while in the office. They are provided this access via Cisco's VPN client. After connected they use Microsoft Terminal Services Client to connect to our internal Terminal Server which allows the same type of access.

What Is Really Going On:

This access posed a real challenge to the IT department. We did not feel comfortable giving them complete direct access to all resources. Again, A decision was made, with out question, we wanted this communication to be completely encrypted to hinder sniffing our valuable fortunes. The mobile employee connects using the Cisco VPN client version 3.6 to our Cisco Pix firewall 515 with PIX IOS 6.2. The VPN client software was installed on their PC, and our profile was loaded. The keys are rotated monthly, the 1st of every month one of our staff members steps them through manually inserting the new keys. After the tunnel is initiated, they are prompted, like the Partners, for a user name and password via the Radius Server. After they are connected, due to previous concerns, they are allowed access to only one internal server, a Windows 2000 Terminal Server. This server, once connected, will give them access to any resource they shall need. The server has the Windows 2000 High Encryption Pack installed, and using the Terminal Server Client, they connect via a 128-bit bi-directional encrypted tunnel. All activity is logged to a SYLOG server residing within a separate security zone.

1.3 Network:

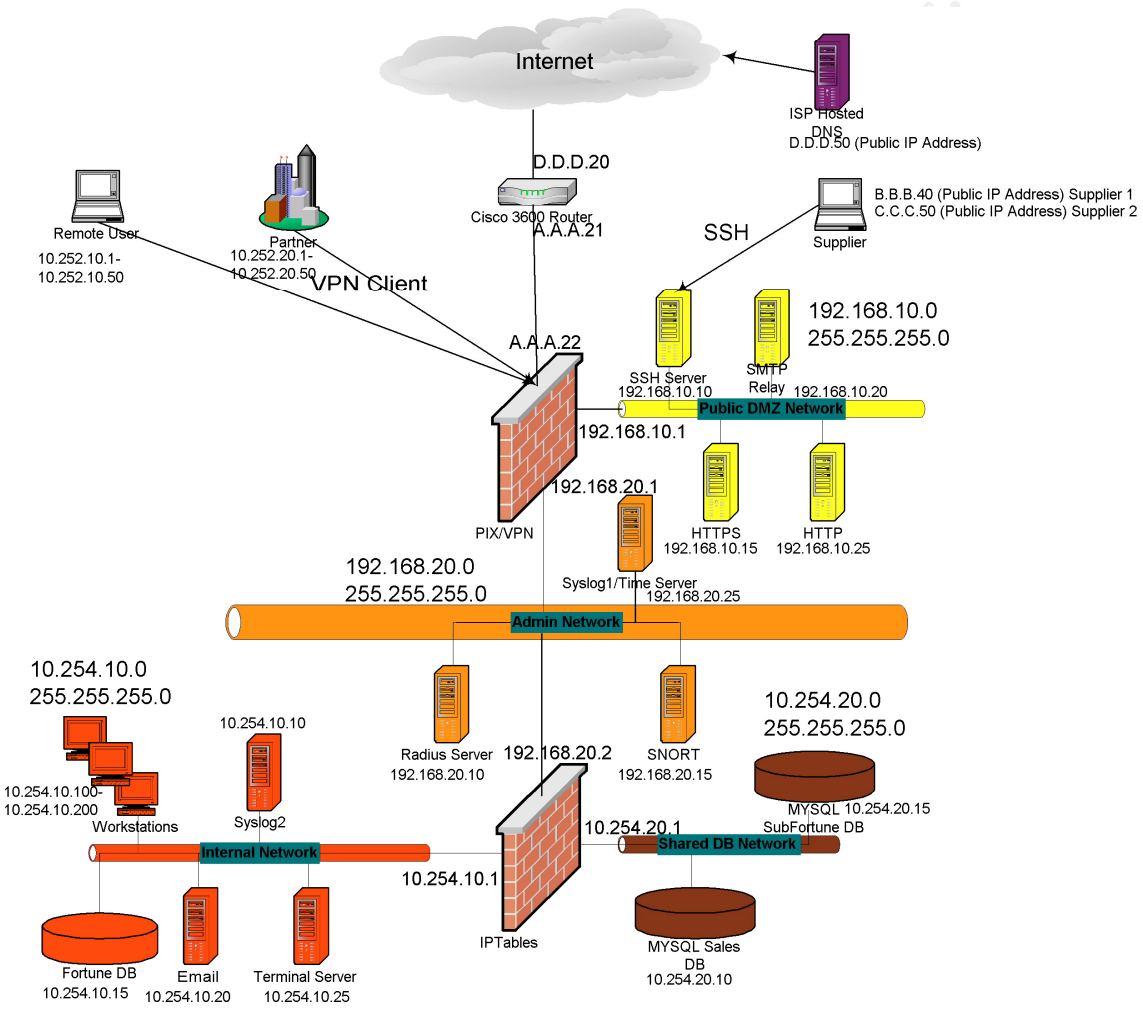
Our network consists of the following:



We have segmented our network into 5 critical areas: The DMZ – location of our public HTTP and HTTPS server, Supplier SSH Server, and The SMTP Relay Server; The Admin Network – Consists of our Radius Authentication Server, The Syslog1 Server, and our SNORT Intrusion Detection System; The SharedDB Network – Contains our Customer Sales Database and Subfortune Database; Internal Network – The Internal Workstations, Syslog2 Server, FortuneDB, Exchange Email Server, and our Windows 2000 Terminal server. Utilizing the PIX 515 and IPTables Firewalls we are able to segment our resources in a manner consistent with their Access Requirements.

1.4 IP Addressing:

All devices are addressed using known, not routable over the Internet addresses, with the exception of the Border Router and the outside interface on the PIX. Every effort was made to provide a high level of granularity to differentiate between separate security zones.

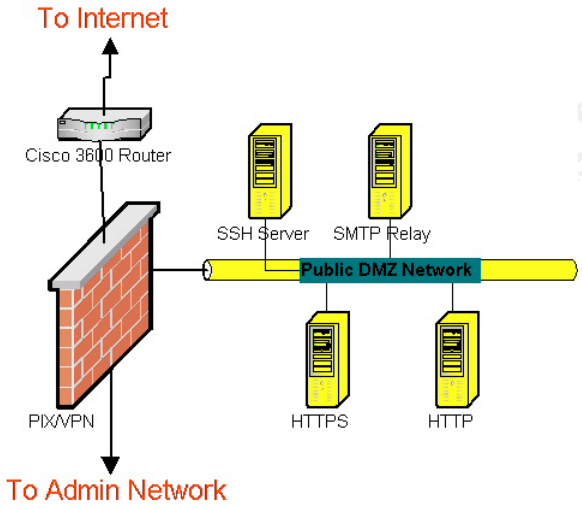


Network	IP Address Scheme	Mask
Admin Network	192.168.20.0	255.255.255.0
Internal Network	10.254.10.0	255.255.255.0
ISP Network	D.D.D.0	-
Partners	10.252.20.1-10.252.20.50	255.255.255.0
PIX – Router connection	A.A.A.x	-
Public DMZ Network	192.168.10.0	255.255.255.0
Remote Users	10.252.10.1-10.252.10.50	255.255.255.0

Shared DB Network	10.254.20.0	255.255.255.0
Suppliers ISP	B.B.B.40, C.C.C.50	-

1.5 Perimeter Device Analysis:

1.5a DMZ Network:



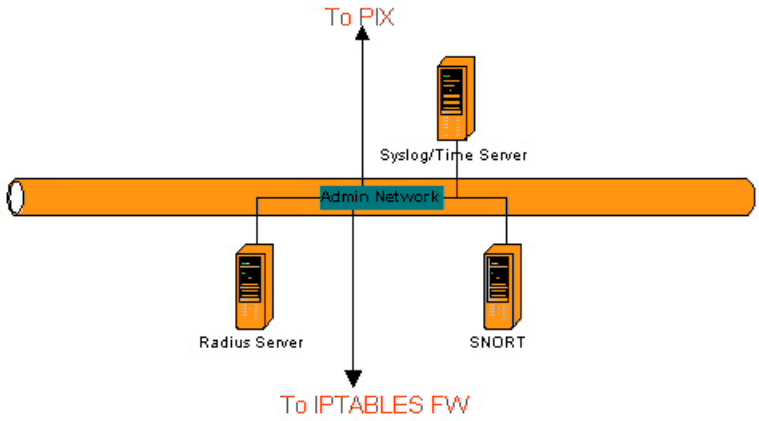
Cisco 3620 MULTISERVICE Platform – IOS Version 12.2. The Cisco 3620 has two slots. Each network module slot accepts a variety of network module interface cards, including LAN and WAN mixed media cards supporting Ethernet, Fast Ethernet, Token Ring, and a variety of WAN technologies. These cards provide the foundation of LAN and WAN connectivity on a single, modular, network module. Additional applications are supported with a series of network module cards offering digital modems, asynchronous and synchronous serial, ISDN PRI, and ISDN BRI interfaces (http://www.cisco.com/en/US/products/hw/routers/ps274/index.html). GIAC has gone with build 12.2 of the IOS. We chose not to go with the Firewall Feature Set, which is very strenuous on the box, and quite pricey. We will use Cisco Access Control Lists to limit access to our network, and give us another layer of defense. This is our first line of defense into our network. All activity is logged to our SYSLOG server residing in the Admin Network.

Cisco PIX 515 Firewall – PIX OS 6.14. Intended for Small-to-Medium Business and Enterprise environments, the Cisco PIX 515E Firewall provides up to 188 Mbps of firewall throughput with the ability to handle as many as 125,000 simultaneous sessions. Certain PIX 515E models includes stateful high-availability capabilities, as well as integrated support for 2,000 IPsec tunnels. The PIX 515E provides a modular chassis with support for up to six 10/100 Fast

Ethernet interfaces (http://www.cisco.com/en/US/products/hw/vpndevc/ps2030/ps4094/index.html). We chose the PIX OS 6.14, which is the latest GD, General Deployment build. This means it has been out in the wild for some duration of time, and thus is thought to be more secure than an Experimental Build. We also obtained the 3-DES software license, to allow 3-DES encrypted connections to our network. The PIX will also play a crucial role in our access control scheme. The FIXUP protocols will allow state-full packet inspection of popular ports. Since it is also hardware based we get improved performance, do not have to rely on the security of an OS under the firewall, and easy router-like software upgrades. All activity is logged to our SYSLOG server residing in the Admin Network.

All other servers in the network have been hardened (Use of security scripts, recent patches, and only running essential services) and explicitly only allow access as defined above and in the following section. All activity is logged to our SYSLOG server residing in the Admin Network.

1.5b Admin Network:



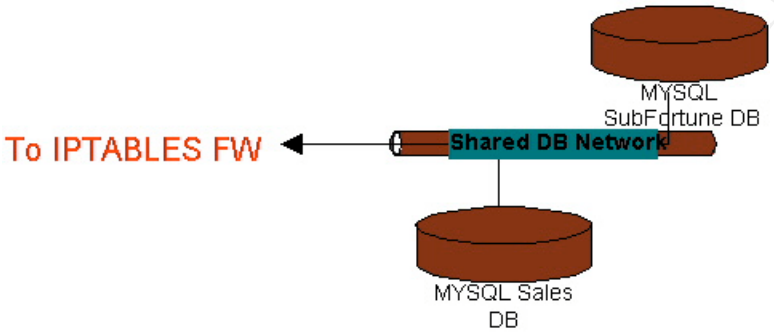
REDHAT 8.0 SYSLOG/Time Server – Redhat 8.0 and Syslog. All servers in the DMZ and the Admin Network log Syslog messages to this server. This allows us to segment a security breach on the DMZ against our logging mechanism. We are also exploring utilizing NTP to synchronize time across the same machines. The machine has also been configured with Log Sentry to mail logs to the internal email server at periodic intervals.

REDHAT 8.0 FREERADIUS Server – Redhat 8.0 and FREERADIUS 0.8.1. This provides another level of security authentication for our VPN clients as well as utilizing ACL pass-to through the Radius authentication (We can control the access control via what logon they utilize). All activity is logged to our SYSLOG server residing in the Admin Network.

REDHAT 8.0 SNORT IDS Server – Redhat 8.0 and SNORT 1.9.0. This allows us to get a real sense of traffic flow between the outside/DMZ and our

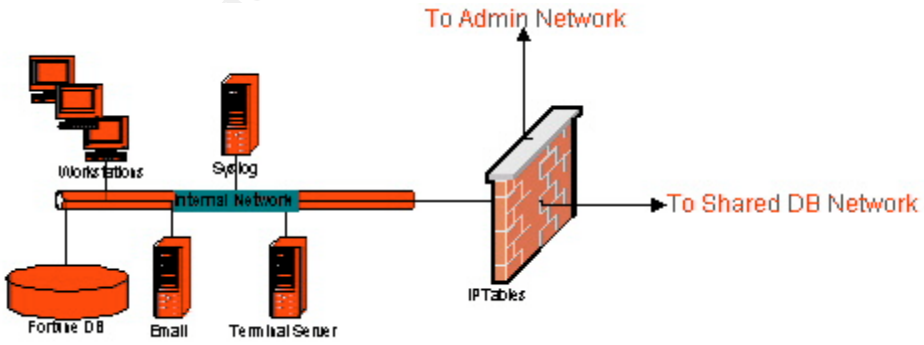
Internal/SharedDB networks. Snort is configured to run in Network intrusion detection mode. Network intrusion detection mode is the most complex and configurable configuration, allowing Snort to analyze network traffic for matches against a user defined rule set and perform several actions based upon what it sees. This allows us to assign custom rules to interact fairly with our environment. The machine has also been configured with Log Sentry to mail ALERTS in the case of major attack signatures. In addition, all activity is logged to our SYSLOG server residing in the Admin Network.

1.5c Shared DB Network:



REDHAT 8.0 MYSQL Servers – Redhat 8.0 and MYSQL 3.23.
The servers in the network have been hardened (Use of security scripts, recent patches, and only running essential services) and explicitly only allow access as defined above and in the following section. All activity is logged to our SYSLOG server residing in the Internal Network, giving us an added layer of logging integrity.

1.5d Internal Network:



REDHAT 8.0 IPTABLES FIREWALL – Redhat version 8.0 with IPTABLES 1.2.7.
This is our last line of defense before relying on host based security internal and on the SharedDB network. We chose to utilize IPTABLES, because it, like the PIX, gives us state-full packet inspection. It also provides for excellent logging, something we needed this close to our important data. It is seen to be a good performer, and we liked the stability of it on top of Linux. It also gives us another vendor independent device, where along with the PIX, a security flaw in one does

not necessarily impose a flaw in the other. We also liked the ability to issue false positives against attack signatures, allowing us to gain valuable time in case of an open hole. The price was also a major selling point. All activity is logged to the SYLOG server residing in the internal network.

Windows 2000 Server Terminal Server – Windows 2000 SVR3 High Encryption Pack with the latest security patches. This is the only server our remote employees have access to.

Windows 2000 Exchange 2000 Server - Windows 2000 SVR3 with the latest security patches. Only internal clients can access this server. All out bound and inbound mail is relayed through our Postfix server residing in our DMZ. The internal Exchange Server is also equipped with Mcafee GroupShield for Exchange to block viruses and messages with other harmful payloads. With the current architecture of an Exchange/Outlook based system this is an absolute necessity.

REDHAT 8.0 SYSLOG Server – Redhat 8.0 and Syslog. All Internal and SharedDB devices log to this server. This was placed in the internal network to allow another layer of protection. The machine has also been configured with Log Sentry to mail logs to the internal email server at periodic intervals.

All other servers in the network have been hardened (Use of security scripts, recent patches, and only running essential services) and explicitly only allow access as defined above and in the following section. All activity is logged to our SYSLOG server residing in the Internal Network, with the exception of the internal clients.

NOTE: All servers in the network have been hardened (Use of security scripts, recent patches, and only running essential services) and explicitly only allow access as defined above and in the following section. Through the use of IPTables we are able to rely not only on Firewall Access Control, but also on host based. All activity is logged to external SYSLOG server where applicable, in case of a compromise we are not guaranteed of log failure/tampering. We were also allowed placement of an IDS system flowing between the outside/DMZ and internal/SharedDB networks to allow us to closely monitor all traffic flow. This design gave us a Defense **IN-DEPTH** design.

1.6 Cost Analysis:

Every attempt was made to keep costs low, we are only selling fortunes after all. Every attempt was made to utilize Open Source software where applicable. The real cost of perimeter software licensing came from the PIX and Router, all other devices rode Open Source, which just gave us just hardware costs, very nominal. No software is completely secure, but we liked the track record of Linux and Open Source and the price could not be beat.

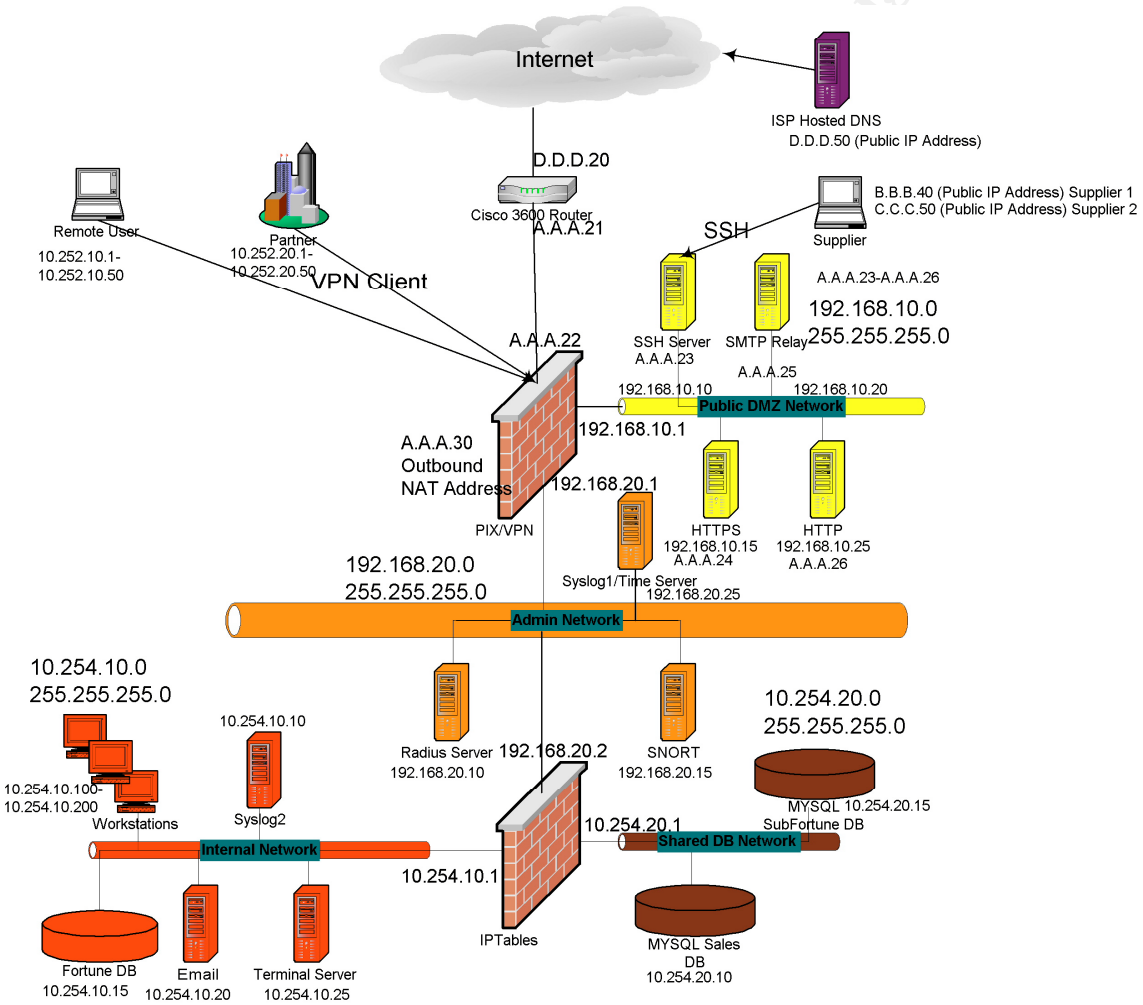
2. Security Policy and Tutorial

2.1 Introduction:

GIAC Enterprises Firewall architecture consists of three main filtering components. The Cisco 3600 Internet router, the Pix 515 device, and our Redhat based IPTABLES Firewall. All three play a crucial role in maintaining our security, the 3600 is our first line of defense, the Pix controls all access to our higher security zones, and the IPTABLES Firewall helps ensure our Database integrity enabling us to segment the SharedDB network in its own security zone. Another important element, which is beyond the scope of this paper, is host based security. A Firewall should never be seen as the solve-everything security device, it is just an additional tool. All servers in the GIAC environment have been hardened and utilize their own host based filtering solutions when available. It is crucial to leverage security throughout the entire network, not just at the perimeter. Layer 2 security is another security tool that is for underutilized. Leveraging IEEE 802.1x authentication, port security, spanning-tree lockdown to prevent DOS attacks, and private VLAN's are all utilized when possible.

2.1a Complete ACL Diagram:

We will implement our Access Policies allowing only what is required, and denying all else. Most of our devices use sequential access list matching, from the top down. Based on this, I like to keep the more specific rules first, the more general rules last. This prevents a general rule being matched before hitting a more specific rule. This helps protect our devices from misconfigurations. We will use the following diagram to implement our policies.



2.2 Cisco 3600 Router:

This again, is our first line of defense. We will utilize Access Control Lists to control our traffic. The lists will be applied as “in filters” as opposed to “out

filters". This will drop the packet immediately at the inbound interface, and will save us some resources. All traffic is automatically dropped unless otherwise defined within the ACL. We have included implicit denies at the end of our lists to enhance readability (and as we all know, sometimes strange things happen with software). We have also included the log command at the end of the list to log to our SYSLOG server in the Admin Network. All commands are accompanied by comments for readability that are as close to legal as possible.

2.2a Access Requirements:

General Inbound Requirements

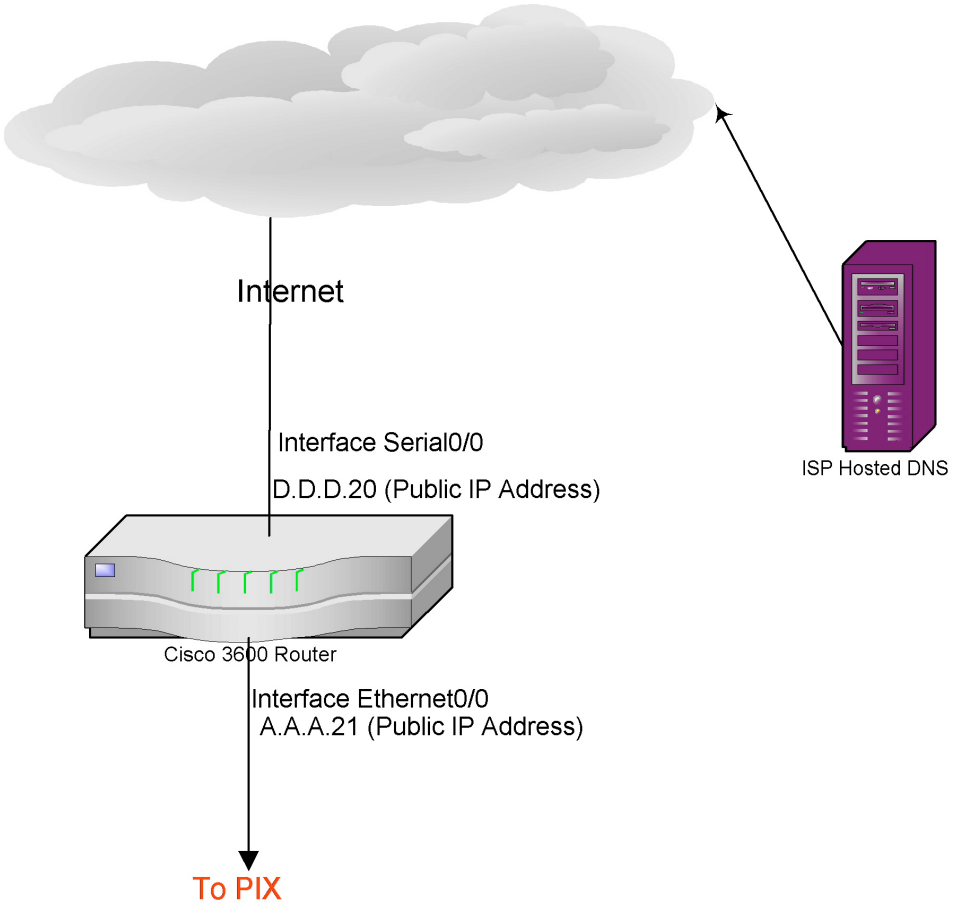
Source	Source Location	Destination	Dest Location	Port(s)	Explanation
Any	Any	SMTP Relay	DMZ Network	25 TCP	Allow SMTP Relay to receive Email
Customers	Any	HTTP Server	DMZ Network	80 TCP	Allow customers to connect to Web Server
Customers	Any	HTTPS Server	DMZ Network	443 TCP	Allow customers to connect to Web Server
Limited "IT" Workstations	Internal Network	Telnet	Cisco 3600 Router	23 TCP	Allow IT to administer Cisco 3600 Router
Partners	Any	PIX	PIX	500 UDP and Protocols 50 and 51	Allow ipsec connections to PIX (IKE uses UDP 500 and IPSEC ESP and AH use Protocols 50 and 51)
Remote Users (Mobile Employees)	Any	PIX	PIX	500 UDP and Protocols 50 and 51	Allow ipsec connections to PIX
Supplier	Suppliers ISP's	SSH Server	DMZ Network	22 TCP	Enable Suppliers to upload Fortunes to SSH Server

General Outbound Requirements

Source	Source Location	Destination	Dest Location	Port(s)	Explanation
Email Server	Internal Network	ANY	ANY	25 TCP	Enable Internal Exchange Server to Send Mail
PIX NAT ADDRESS	PIX	Any HTTP or HTTPS Server	ANY	80 TCP and 443 TCP	Enable Internet Browsing
PIX NAT ADDRESS	Internal Network	SSH Servers	ANY	22 TCP	Allow IT Admin through SSH
PIX NAT ADDRESS	Internal Network	FTP Servers	ANY	20 TCP and 21 TCP	Allow Connection to any FTP Server (Software, Patches etc..)

PIX NAT ADDRESS	PIX	ISP Hosted DNS Server	ISP Hosted DNS Server	53 TCP and 53 UDP	Enable DNS

2.2b Cisco 3600 Diagram:



Inbound ACL for Serial 0/0

```
! First Lets Block and Log some suspicious packets
! No IP address means something is up
access-list 101 deny ip host 0.0.0.0 any log
! LoopBack and Broadcast
access-list 101 deny ip 127.0.0.0 0.255.255.255 any log
access-list 102 deny ip 255.0.0.0 0.255.255.255 any log
! My internal Addresses
access-list 102 deny ip 192.168.0.0 0.0.255.255 any log
access-list 102 deny ip 10.0.0.0 0.255.255.255 any log
```

```
! Other private addresses
access-list 102 deny ip 172.16.0.0 0.15.255.255 any log
! Any of our internet addresses coming in is bad
access-list 102 deny ip A.A.A.0 0.0.0.255 any log
! Block Finger
access-list 101 deny tcp any any eq 79 log
! Block Portmap
access-list 101 deny tcp any any eq 111 log
access-list 101 deny udp any any eq 111 log
! Block FTP and Telnet
access-list 101 deny udp any any eq 20 log
access-list 101 deny tcp any any eq 21 log
access-list 101 deny tcp any any eq 23 log

! Let some meaning full stuff in
! Allow Customers to connect to HTTP and HTTPS servers
access-list 101 permit tcp any host A.A.A.26 eq 80 log
access-list 101 permit tcp any host A.A.A.24 eq 443 log

! Allow Email to SMTP Relay
access-list 101 permit tcp any host A.A.A.25 eq 25 log

! Allow VPN Connections to Outside PIX Interface for Mobile Employees and
Partners
access-list 101 permit udp any host A.A.A.22 eq 500 log
access-list 101 permit 50 any host A.A.A.22 log
access-list 101 permit 51 any host A.A.A.22 log

! Allow Suppliers to connect to SSH Server to upload fortunes
access-list 101 permit tcp host  B.B.B.40 host A.A.A.23 eq 22 log
access-list 101 permit tcp host  C.C.C.50 host A.A.A.23 eq 22 log

! Deny all other traffic
access-list 101 deny ip any any log
```

In addition to the above, another step that is being implemented is blocking known hostile NetBlocks. http://isc.incidents.org/top10.html provides a summarized listing of the top 20 attacking class C subnets over the past three days. They also provide many other useful information, including recent trends in popular port scanning. Whenever appropriate, these NetBlocks should be dropped automatically by your gateway router.

Outbound ACL for Serial 0/0

```
! Allow DNS out to ISP DNS Server for PIX's PAT Address
access-list 102 permit tcp host A.A.A.30 host D.D.D.50 eq 53 log
access-list 102 permit udp host A.A.A.30 host D.D.D.50 eq 53 log

! Allow HTTP and HTTPS out for PIX's PAT Address
access-list 102 permit tcp host A.A.A.30 any eq 80 log
access-list 102 permit tcp host A.A.A.30 any eq 443 log

! Allow outbound Email for PIX's PAT Address
access-list 102 permit tcp host A.A.A.30  host any eq 25 log

! Allow outbound SSH for PIX's PAT Address
access-list 102 permit tcp host A.A.A.30  host any eq 22 log

!Allow Outbound FTP for PIX's PAT Address
access-list 102 permit tcp host A.A.A.30  host any eq 20 log
access-list 102 permit tcp host A.A.A.30  host any eq 21 log

!Allow Outbound Telnet for PIX's PAT Address
access-list 102 permit tcp host A.A.A.30  host any eq 23 log

! Deny all other traffic
access-list 102 deny ip any any log
```

Inbound ACL for Telnet Access

```
! Only allow pat address on PIX to telnet to router
access-list 113 permit ip A.A.A.30
access-list 113 deny ip any any

line vty 0 4
access-list 113 in
```

Additional configuration parameters:

While our intent is to implement the access control itself, the following hardening configurations are also utilized.

```
! Make sure passwords are encrypted
service password-encryption

! Turn off the silly stuff
no service tcp-small-servers
no service udp-small-servers
no service finger
no snmp server
no cdp enable
no ip http server
no ip bootp server
```

```
no ip identd
ntp disable

! Helps avoid spoofing
no ip source-route

! Configure our Login, banner
banner login /
WARNING: All unauthorized all unauthorized access will be prosecuted to the full
extent of the LAW!
/

! Configure to log messages to our SYSLOG1 Server in the Admin Network
! turn off console logging
no logging console
! Set logging level, debugging will give us lots of information
logging trap debugging
! This is where we will log to on our Syslog server
logging facility local 6
! Tell it where it is, the is a static on the PIX
logging A.A.A.27

! Our IP route, ie default to ISP router
ip route 0.0.0.0 0.0.0.0 D.D.D.44
```

2.3 Cisco PIX:

Our PIX is our real workhorse of our architecture. This device, a Cisco PIX 515, allows us to implement an integrated VPN and firewall solution. The PIX is hardware based and allows for 120 Mbs throughput. It is also backed by the Cisco TAC center, which allows for excellent support.
We have also created our access lists only allowing appropriate traffic. All our interfaces are “valued”, and you cannot go from a lower to a higher level without configuring it. All other traffic is dropped by default, and we have manually defined and implicit deny at the end of our lists. Our outbound connections are translated, utilizing PAT (Port Address Translation). We have also defined our fixup protocols, which allows for stateful packet inspection. The PIX fixup protocols maintain state and automatically drop any traffic that does not meet it’s criteria. We utilize 3DES VPN Ipsec encryption for our VPN connections, and utilize an internal Radius server for advanced accounting and access control. All our logging is sent to the SYLOG1 server residing in our Admin Network. This syslog server has been configured with Logsentry, to provide centralized Email Alerting for our security devices. Inbound access to our DMZ and our syslog servers is provided through Static command, which provide for redirection. All remote administration is only enabled from our internal IT staff, utilizing SSH

only. All commands are accompanied by comments for readability that are as close to legal as possible.

Our Statics redirect to the following:

Device	LAN IP	WAN IP
SSH Server	192.168.10.10	A.A.A.23
HTTPS Server	192.168.10.15	A.A.A.24
SMTP Relay Server	192.168.10.20	A.A.A.25
HTTP Server	192.168.10.25	A.A.A.26
SYSLOG1 Server	192.168.20.25	A.A.A.27

2.3a Access Requirements:

General ACL Requirements For PIX FireWall

Inbound

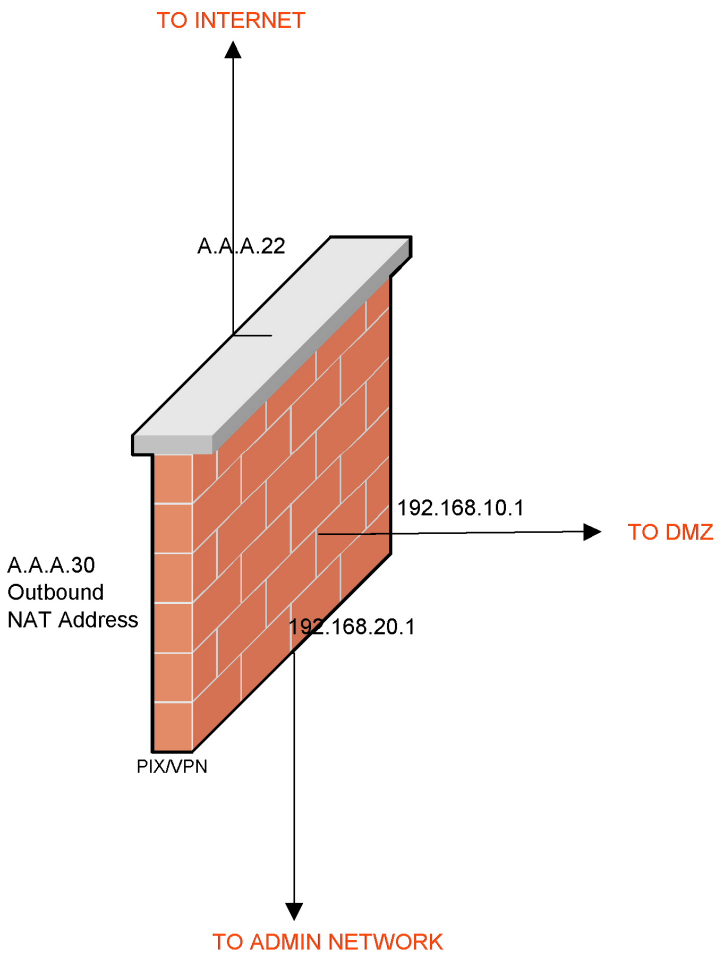
Source	Source Location	Destination	Dest Location	Port(s)	Explanation
ANY	ANY	SMTP Relay	DMZ Network	25 TCP	Allow our relay server to accept inbound email
Customers	ANY	HTTP or HTTPS Server	DMZ Network	80 TCP and 443 TCP	Allow customers to connect to HTTP and HTTPS Server
HTTP Server	DMZ Network	Syslog1 Server	Admin Network	514 UDP	Enable Server to send messages to SYSLOG server
HTTPS Server	DMZ Network	Syslog1 Server	Admin Network	514 UDP	Enable Server to send messages to SYSLOG server
HTTPS Server	DMZ Network	MySQL SubfortuneDB	SharedDB Network	22 TCP	Allow Secure Web Server to connect to MySQL SubfortuneDB
HTTPS Server	DMZ Network	MySQL SalesDB	SharedDB Network	22 TCP	Allow Secure Web Server to connect to MySQL SalesDB
Partners	Partner Network	MySQL SubfortuneDB	SharedDB Network	3306 TCP	Enable Direct Connection to SubfortuneDB
Remote Users (Mobile Employees)	Remote Users Network	Terminal Server	Internal Network	3389 TCP	Allow Mobile Employees to Connect to Terminal Server
SMTP Relay	DMZ Network	Syslog1 Server	Admin Network	515 UDP	Enable Server to send messages to SYSLOG server

SMTP Relay	DMZ Network	Internal Email Server	Internal Network	25 TCP	Enable SMTP Relay to relay to the internal server
SSH Server	DMZ Network	Syslog1 Server	Admin Network	514 UDP	Enable Server to send messages to SYSLOG server
Supplier	Suppliers ISPS's	SSH Server	DMZ Network	22 TCP	Enable Suppliers to upload Fortunes to SSH Server

Outbound

Source	Source Location	Destination	Dest Location	Port(s)	Explanation
Email Server	Internal Network	ANY	ANY	25 TCP	Enable Internal Exchange Server to Send Email
HTTP Server	DMZ Network	ISP Hosted DNS Server	ISP Hosted DNS Server	53 TCP and 53 UDP	Enable DNS
HTTPS Server	DMZ Network	ISP Hosted DNS Server	ISP Hosted DNS Server	53 TCP and 53 UDP	Enable DNS
Internal Network	Internal Network	Any HTTP or HTTPS Server	ANY	80 TCP and 443 TCP	Enable Internet Browsing
Internal Network	Internal Network	ISP Hosted DNS Server	ISP Hosted DNS Server	53 TCP and 53 UDP	Enable DNS
Limited "IT" Workstations	Internal Network	SSH Servers	ANY	22 TCP	Allow IT Admin through SSH
Limited "IT" Workstations	Internal Network	FTP Servers	ANY	20 TCP and 21 TCP	Allow Connection to any FTP Server (Software, Patches etc..)
Limited "IT" Workstations	Internal Network	Telnet	ANY	23 TCP	Allow IT to telnet
MySQL SubfortuneDB	SharedDB Network	SSH Server	DMZ Network	22 TCP	Enable automated 'bot' on SubFortune DB to pull uploaded fortunes in via an encrypted SSH connection
SMTP Relay	DMZ Network	ISP Hosted DNS Server	ISP Hosted DNS Server	53 TCP and 53 UDP	Enable DNS

2.3b Cisco PIX Diagram:



Inbound ACL for Outside Interface

```
! Allow connections to HTTP and HTTPS Servers from anywhere
access-list 102 permit tcp any host A.A.A.26 eq www
access-list 102 permit tcp any host A.A.A.24 eq 443

! Allow our Relay Server to receive inbound Email
access-list 102 permit tcp any host A.A.A.25 eq 25

! Allow suppliers to connect to SSH server
access-list 102 permit tcp host B.B.B.40 host A.A.A.23 eq 22
access-list 102 permit tcp host C.C.C.50 host A.A.A.23 eq 22

!Allow internet router to send SYLOG messages to Syslog1
access-list 102 permit udp host A.A.A.21 host A.A.A.27  eq 514

! Deny all other traffic
access-list 102 deny ip any any
```

access-group 102 in interface outside

Inbound ACL for DMZ Interface

```
! Allow HTTPS Server to SSH to SalesDB and SubfortuneDB
access-list 103 permit tcp host 192.168.10.15 host 10.254.20.10 eq 22
access-list 103 permit tcp host 192.168.10.15 host 10.254.20.15 eq 22

! Allow SMTP Relay Server to relay to internal Email Server
access-list 103 permit tcp host 192.168.10.20 host 10.254.10.20 eq smtp

! Allow servers to send syslog messages to Syslog1
access-list 103 permit udp host 192.168.10.10 host 192.168.20.1 eq 514
access-list 103 permit udp host 192.168.10.15 host 192.168.20.1 eq 514
access-list 103 permit udp host 192.168.10.20 host 192.168.20.1 eq 514
access-list 103 permit udp host 192.168.10.25 host 192.168.20.1 eq 514

! Allow DMZ servers to connect to ISP DNS Server
access-list 103 permit tcp host 192.168.10.15 host D.D.D.50 eq domain
access-list 103 permit tcp host 192.168.10.20 host D.D.D.50 eq domain
access-list 103 permit tcp host 192.168.10.25 host D.D.D.50 eq domain
access-list 103 permit udp host 192.168.10.15 host D.D.D.50 eq domain
access-list 103 permit udp host 192.168.10.20 host D.D.D.50 eq domain
access-list 103 permit udp host 192.168.10.25 host D.D.D.50 eq domain


! Deny all other traffic
access-list 103 deny ip any any

access-group 103 in interface DMZ
```

Outbound ACL for Inside Interface

```
! Allow outbound DNS to ISP DNS Servers from internal network
access-list 104 permit tcp 10.254.10.0 255.255.255.0 host D.D.D.50 eq domain
access-list 104 permit udp 10.254.10.0 255.255.255.0 host D.D.D.50 eq domain

! Allow outbound HTTP and HTTPS from internal network
access-list 104 permit tcp 10.254.10.0 255.255.255.0 any eq www
access-list 104 permit tcp 10.254.10.0 255.255.255.0 any eq 443

! Allow Internal Email Server to send out mail
access-list 104 permit tcp host 10.254.10.20 any eq smtp

! Allow limited IT workstations outbound SSH, FTP, and Telnet
```

```
access-list 104 permit tcp host 10.254.10.70 any eq 22
access-list 104 permit tcp host 10.254.10.71 any eq 22
access-list 104 permit udp host 10.254.10.70 any eq 20
access-list 104 permit udp host 10.254.10.71 any eq 20
access-list 104 permit tcp host 10.254.10.70 any eq 21
access-list 104 permit tcp host 10.254.10.71 any eq 21
access-list 104 permit tcp host 10.254.10.70 any eq 23
access-list 104 permit tcp host 10.254.10.71 any eq 23

! Allow SubfortuneDB to SSH to SSH server to download supplier fortunes
access-list 104 permit tcp host 10.254.20.15 host 192.168.10.20 eq 22

! Deny all other traffic
access-list 104 deny ip any any

access-group 104 in interface inside
```

ACL for controlling NAT/PAT Process

```
access-list 110 permit ip 10.0.0.0 255.0.0.0 10.252.0.0 255.255.0.0

nat (inside) 0 access-list 110
```

ACL for controlling Partners VPN (This utilizes xauth with VPN groups to pull the Acl, cisco-avpair = “acl=120”, when the VPN/Radius Authorization occurs)

```
! Allow partner to connect to SubFortuneDB using MYSQL Client
access-list 120 permit ip 10.252.20.0 255.255.255.0 host 10.254.20.15 eq 3306

! Deny all other traffic
access-list 120 deny ip any any
```

ACL for controlling Remote Users VPN (This utilizes xauth with VPN groups to pull the Acl, cisco-avpair = “acl=121”, when the VPN/Radius Authorization occurs)

```
! Allow remote user to connect to terminal server
access-list 121 permit ip 10.252.10.0 255.255.255.0 host 10.254.10.25 eq 3389

! Deny all other traffic
access-list 121 deny ip any any
```

Additional configuration parameters:

While our intent is to implement the access control itself, the following hardening configurations are also utilized.

```
! The following commands is what tell the PIX to use Stateful filtering for the
specified
! protocols. An important feature for us here is the mailguard, ie fixup protocol 25.
This
! feature will sanitize all of our smtp traffic, only allowing legitimate commands.

fixup protocol ftp 21
fixup protocol domain 53
fixup protocol http 80
fixup protocol h323 1720
fixup protocol rsh 514
fixup protocol rtsp 554
fixup protocol smtp 25
fixup protocol sqlnet 1521
fixup protocol sip 5060
fixup protocol skinny 2000

! This assigns different security levels to our interface
nameif ethernet0 outside security0
nameif ethernet1 inside security100
nameif ethernet2 DMZ security10

! Set up our logging to our SYSLOG server
logging on
logging timestamp
logging buffered debugging
logging trap notifications
logging history notifications
logging facility 19
logging host inside 192.168.20.25

! This is our PAT address for outbound access
global (outside) 1 A.A.A.30

! Helps prevent spoofing
ip verify reverse-path interface outside

! These are our static mappings for our publicly accessible servers
! As A general Rule of thumb, use access list :Higher Security Interface=>Lower
! use static: Lower Security=>Higher
! SSH Server
static (DMZ, outside) A.A.A.23 192.168.10.10 netmask 255.255.255.255 0 0
```

```
! HTTPS Server
static (DMZ, outside) A.A.A.24 192.168.10.15 netmask 255.255.255.255 0 0
! SMTP Relay Server
static (DMZ, outside) A.A.A.25 192.168.10.20 netmask 255.255.255.255 0 0
! HTTP Server
static (DMZ, outside) A.A.A.26 192.168.10.25 netmask 255.255.255.255 0 0
! This allows our Internet Router to forward to our SYSLOG server
static (inside, outside) A.A.A.27 192.168.20.25 netmask 255.255.255.255 0 0

! No SNMP
no snmp-server enable

! Enable Flood Protection
floodguard enable

! Disable Pings
icmp deny any outside
icmp deny any DMZ
icmp deny any inside

! Enable fragmented packet protection
sysopt security fragguard
no sysopt route dnat


! Allow SSH access for remote administration from internal IT only
! Before using ssh you must issue “ca generate rsa key 1024”, and “ ca save all
! to generate an RSA key
ssh 10.254.10.70 255.255.255.255 inside
ssh 10.254.10.71 255.255.255.255 inside
ssh timeout 20

! We need to define some static routing
! Our default to the outside world
route outside 0.0.0.0 0.0.0.0 D.D.D.20 1
! To get to our Internal and Shared DB Networks
route inside 10.254.0.0 255.255.0.0 192.168.20.2 1
```

Note: All VPN configuration is covered in the Cisco Client VPN Tutorial Section

2.4 IPTABLES Firewall:

Our IPTables Firewalls main functionality is to segregate and control traffic between our networks and the SharedDB Network. This is where our live SubfortuneDB is as well as our SalesDB. For a nominal price is provides some great functionality.
We utilize IPTables to provide our firewall capabilities. A large reason we are using IPTables is because of the excellent logging capabilities, we want to be able to log any and all traffic at a moments notice. Iptables provides some of the best logging in the industry. All of our rules, Inbound, Outbound, and Forward (the most pertinent to us, forwarding between interfaces) drop traffic by default unless allowed through one of our rules. We currently are providing for no NAT processes. All traffic is forwarded between interfaces when appropriate. All logging is sent to our internal syslog server, SYSLOG2.

2.4a Access Requirements:

General ACL Requirements For IPTABLES FireWall

Inbound

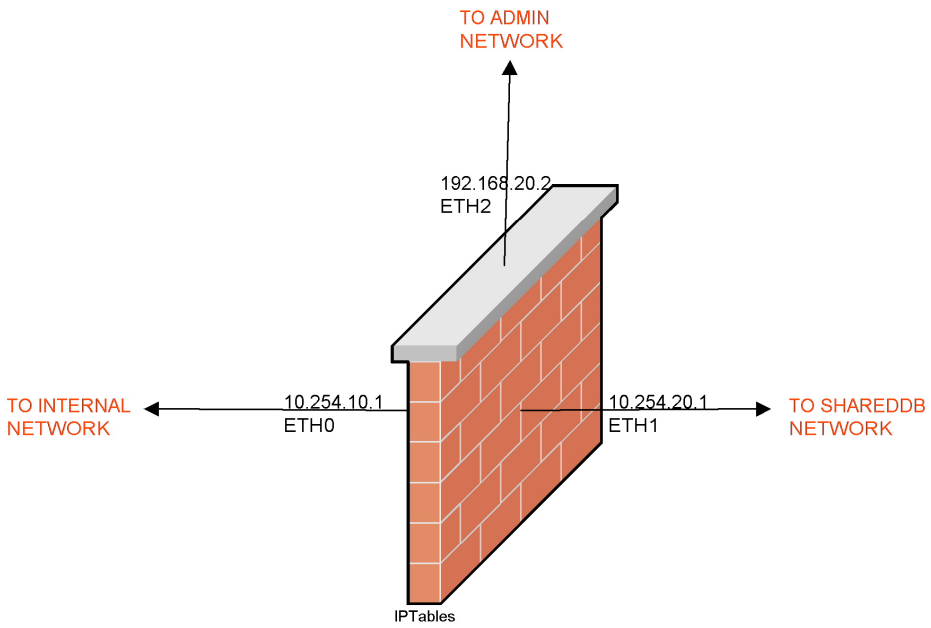
Source	Source Location	Destination	Dest Location	Port(s)	Explanation
HTTPS Server	DMZ Network	MySQL SubfortuneDB	SharedDB Network	22 TCP	Allow Secure Web Server to connect to MySQL SubfortuneDB
HTTPS Server	DMZ Network	MySQL SalesDB	SharedDB Network	22 TCP	Allow Secure Web Server to connect to MySQL SalesDB
MySQL SalesDB	SharedDB Network	Syslog2	Internal Network	514 UDP	Enable DB to send messages to SYSLOG server
MySQL SubfortuneDB	SharedDB Network	Syslog2	Internal Network	514 UDP	Enable DB to send messages to SYSLOG server
Partners	Partner Network	MySQL SubfortuneDB	SharedDB Network	3306 TCP	Enable Direct Connection to SubfortuneDB
Remote Users (Mobile Employees)	Remote Users Network	Terminal Server	Internal Network	3389 TCP	Allow Mobile Employees to Connect to Terminal Server
SMTP Relay	DMZ Network	Internal Email Server	Internal Network	25 TCP	Enable SMTP Relay to relay to the internal server
SYSLOG1 Server	Admin Network	Internal Email Server	Internal Network	25 TCP	Enable Syslog server to email log to Internal Email Server via LogSentry

Outbound

Source	Source Location	Destination	Dest Location	Port(s)	Explanation
Email Server	Internal Network	ANY	ANY	25 TCP	Enable Internal Exchange Server to Send Email

Limited "IT" Workstations	Internal Network	SSH Servers	ANY	22 TCP	Allow IT Admin through SSH
Limited "IT" Workstations	Internal Network	FTP Servers	ANY	20 TCP and 21 TCP	Allow Connection to any FTP Server (Software, Patches etc..)
Limited "IT" Workstations	Internal Network	Telnet	ANY	23 TCP	Allow IT to administer Cisco 3600 Router
MySQL SubfortuneDB	SharedDB Network	SSH Server	DMZ Network	22 TCP	Enable automated 'bot' on SubFortune DB to pull uploaded fortunes in via an encrypted SSH connection
Workstations	Internal Network	MySQL SubfortuneDB	SharedDB Network	3306 TCP	Enable Direct Connection to SubfortuneDB
Workstations	Internal Network	MySQL SalesDB	SharedDB Network	3306 TCP	Enable Direct Connection to SalesDB
Workstations	Internal Network	Any HTTP or HTTPS Server	ANY	80 TCP and 443 TCP	Enable Internet Browsing
Workstations	Internal Network	ISP Hosted DNS Server	ISP Hosted DNS Server	53 TCP and 53 UDP	Enable DNS

2.4b IPTABLES Diagram:




```
#########################################################################
#!/bin/bash
#Stanley R. Yachera 2003
#Firewall.sh script, runs at system startup

# Flush Rule Set
iptables –F
iptables –X
iptables –Z

# Our default policy will always be to DROP
iptables –P INPUT DROP
iptables –P FORWARD DROP
iptables –P OUTPUT DROP


# Lets turn on IP FORWARDING, This allows forwarding between interfaces
/bin/echo “1” > /proc/sys/net/ipv4/ip_forward

# Turn on some excellent security features
# Disable response to PINGS
/bin/echo “1” > /proc/sys/net/ipv4/icmp_echo_ignore_all
# Prevent SYN floods from taking up resources
/bin/echo “1” > /proc/sys/net/ipv4/tcp_syncookies

# Access Control
# Lets set some interface variables for readability
INTERNAL_IFACE = “eth0”
SHAREDDB_IFACE = “eth1”
EXTERNAL_IFACE = “eth2”

# ALLOW SELF ACCESS
iptables –A INPUT –i lo –p all –j ACCEPT
```

```
iptables –A OUTPUT –o lo –p all –j ACCEPT
# ALLOW ESTABLISHED THROUGH
iptables –A INPUT –i $INTERNAL_IFACE –m state –state ESTABLISHED, RELATED –j
ACCEPT
iptables –A OUTPUT –i $EXTERNAL_IFACE –m state –state ESTABLISHED, RELATED –j
ACCEPT
iptables –A FORWARD –i $INTERNAL_IFACE –m state –state ESTABLISHED, RELATED –j
ACCEPT
iptables –A FORWARD –i $SHAREDDB_IFACE  –m state –state ESTABLISHED, RELATED –j
ACCEPT
iptables –A FORWARD –i $EXTERNAL_IFACE –m state –state ESTABLISHED, RELATED –j
ACCEPT

 ################################INPUT RULES#############
# ALLOW INBOUND SSH FOR IT ADMINISTRATION
iptables –A INPUT –p TCP –i $INTERNAL_IFACE –s 10.254.10.70/32
--dport 22 –j ACCEPT
iptables –A INPUT –p TCP –i $INTERNAL_IFACE –s 10.254.10.71/32
--dport 22 –j ACCEPT
######LETS LOG SOME INTERESTING TRAFFIC
# SPOOFING
iptables -A INPUT -p tcp -i $EXTERNAL_IFACE -s 10.0.0.0/8 -j LOG --log-prefix
“SPOOFATTEMPT”
# INVALID TRAFFIC ON OUTSIDE INTERFACE
iptables –A INPUT –m state --state INVALID –i $EXTERNAL_IFACE –j LOG --log-prefix
“INVALIDTRAFFIC”
# SYNFIN SCANS
iptables –A INPUT –p TCP --tcp-flags ALL SYN, FIN –j LOG --log-prefix “SYNFINSCAN”
# FIN SCANS
iptables –A INPUT –p TCP --tcp-flags ACK, FIN FIN –j LOG --log-prefix “FINSCAN”
# ICMP FRAGMENTS
iptables –A INPUT –p ICMP –f –j LOG --log-prefix “ICMPFRAG”
# FINGER PACKETS
iptables –A INPUT –p TCP –d 0/0 --dport 79 –j LOG --log-prefix “FINGER”
# PORT MAP
iptables –A INPUT –p TCP –d 0/0 --dport 111 –j LOG --log-prefix “PORTMAP”
iptables –A INPUT –p UDP –d 0/0 --dport 111 –j LOG --log-prefix “PORTMAP”
# SUB-7 PACKETS
iptables –A INPUT –p UDP –d 0/0 --dport 27374 –j LOG --log-prefix “SUB7”
# NETBUS AND BO2K
iptables –A INPUT –p TCP –d 0/0 --dport 12345:12346 –j LOG --log-prefix “NETBUS”
iptables –A INPUT –p TCP –d 0/0 --dport 54320:54321 –j LOG --log-prefix “BO2K”
iptables –A INPUT –p UDP –d 0/0 --dport 54320:54321 –j LOG --log-prefix “BO2K
######END LETS LOG SOME INTERESTING TRAFFIC
################################END INPUT RULES########

################################OUTPUT RULES#############
# WE DO NOT ALLOW ANYTHING HERE. IF WE NEED OUTBOUND FTP FOR
# UPDATES WE UNCOMMENT THE FOLLOWING TWO LINES
#iptables –A OUTPUT –p TCP --dport 20 –j ACCEPT
#iptables –A OUTPUT –p TCP --dport 21 –j ACCEPT
################################END OUTPUT RULES########

################################FORWARDING RULES#############
######INBOUND ACCESS CONTROL
# EXTERNAL=>INTERNALNETWORK
```

```
# ALLOW REMOTE USERS (VPN) ACCESS TO TERMINAL SERVER
iptables –A FORWARD –p TCP –i $EXTERNAL_IFACE –s 10.252.10.0/24
–o $INTERNAL_IFACE -d  10.254.10.25/32 --dport 3389 –j ACCEPT
# ALLOW SMTP RELAY TO RELAY EMAIL MESSAGES TO INTERNAL EMAIL
# SERVER
iptables –A FORWARD –p TCP –i $EXTERNAL_IFACE –s 192.168.10.20/32
–o $INTERNAL_IFACE -d  10.254.10.20/32 --dport 25 –j ACCEPT
# ALLOW SYLOG1 SERVER TO EMAIL SYSTEM LOGS VIA LOGSENTRY TO
# INTERNAL EMAIL SERVER
iptables –A FORWARD –p TCP –i $EXTERNAL_IFACE –s 192.168.20.25/32
–o $INTERNAL_IFACE -d  10.254.10.20/32 --dport 25 –j ACCEPT


# EXTERNAL=>SHAREDDB
# ALLOW HTTPS SERVER TO SSH (TUNNELS MYSQL TRAFFIC) TO
# SUBFORTUNEDB AND SALESDB FOR CUSTOMER ONLINE TRANSACTIONS
iptables –A FORWARD –p TCP –i $EXTERNAL_IFACE –s 192.168.10.15/32
–o $ SHAREDDB_IFACE -d  10.254.20.10/32 --dport 22 –j ACCEPT
iptables –A FORWARD –p TCP –i $EXTERNAL_IFACE –s 192.168.10.15/32
–o $ SHAREDDB_IFACE -d  10.254.20.15/32 --dport 22 –j ACCEPT
# ALLOW PARTNERS (VPN) TO CONNECT TO SUBFORTUNEDB
iptables –A FORWARD –p TCP –i $EXTERNAL_IFACE –s 10.252.20.0/24
–o $ SHAREDDB_IFACE -d  10.254.20.15/32 --dport 22 –j ACCEPT

# SHAREDDB=>INTERNALNETWORK
# ALLOW SUBFORTUNEDB AND SALES DB TO SEND SYSLOG MESSAGES TO # INTERNAL
SYSLOG2 SERVER
iptables –A FORWARD –p UDP –i $SHAREDDB_IFACE –s 10.254.20.10/32
–o $INTERNAL_IFACE -d  10.254.10.10/32 --dport 514 –j ACCEPT
iptables –A FORWARD –p UDP –i $SHAREDDB_IFACE –s 10.254.20.15/32
–o $INTERNAL_IFACE -d  10.254.10.10/32 --dport 514 –j ACCEPT
######END INBOUND  ACCESS CONTROL

######OUTBOUND ACCESS CONTROL
# INTERNALNETWORK=>EXTERNAL
# ALLOW OUTBOUND HTTP AND HTTPS FROM INTERNAL NETWORK TO
# ANYWHERE
iptables –A FORWARD –p TCP –i $INTERNAL_IFACE –s 10.254.10.0/24
–o $EXTERNAL_IFACE -d  0/0 --dport 80 –j ACCEPT
iptables –A FORWARD –p TCP –i $INTERNAL_IFACE –s 10.254.10.0/24
–o $EXTERNAL_IFACE -d  0/0 --dport 443 –j ACCEPT
# ALLOW OUTBOUND DNS FROM INTERNAL NETWORK TO DNS SERVERS
iptables –A FORWARD –p TCP –i $INTERNAL_IFACE –s 10.254.10.0/24
–o $EXTERNAL_IFACE -d  D.D.D.50/32 --dport 53 –j ACCEPT
iptables –A FORWARD –p UDP –i $INTERNAL_IFACE –s 10.254.10.0/24
–o $EXTERNAL_IFACE -d  D.D.D.50/32 --dport 53 –j ACCEPT
# ALLOW LIMITED IT WORKSTATIONS OUBOUND SSH, FTP, AND TELNET TO
# ANYWHERE
iptables –A FORWARD –p TCP –i $INTERNAL_IFACE –s 10.254.10.70/32
–o $EXTERNAL_IFACE -d  0/0 --dport 22 –j ACCEPT
iptables –A FORWARD –p TCP –i $INTERNAL_IFACE –s 10.254.10.71/32
–o $EXTERNAL_IFACE -d  0/0 --dport 22 –j ACCEPT
iptables –A FORWARD –p TCP –i $INTERNAL_IFACE –s 10.254.10.70/32
–o $EXTERNAL_IFACE -d  0/0 --dport 20 –j ACCEPT
iptables –A FORWARD –p TCP –i $INTERNAL_IFACE –s 10.254.10.71/32
–o $EXTERNAL_IFACE -d  0/0 --dport 20 –j ACCEPT
```

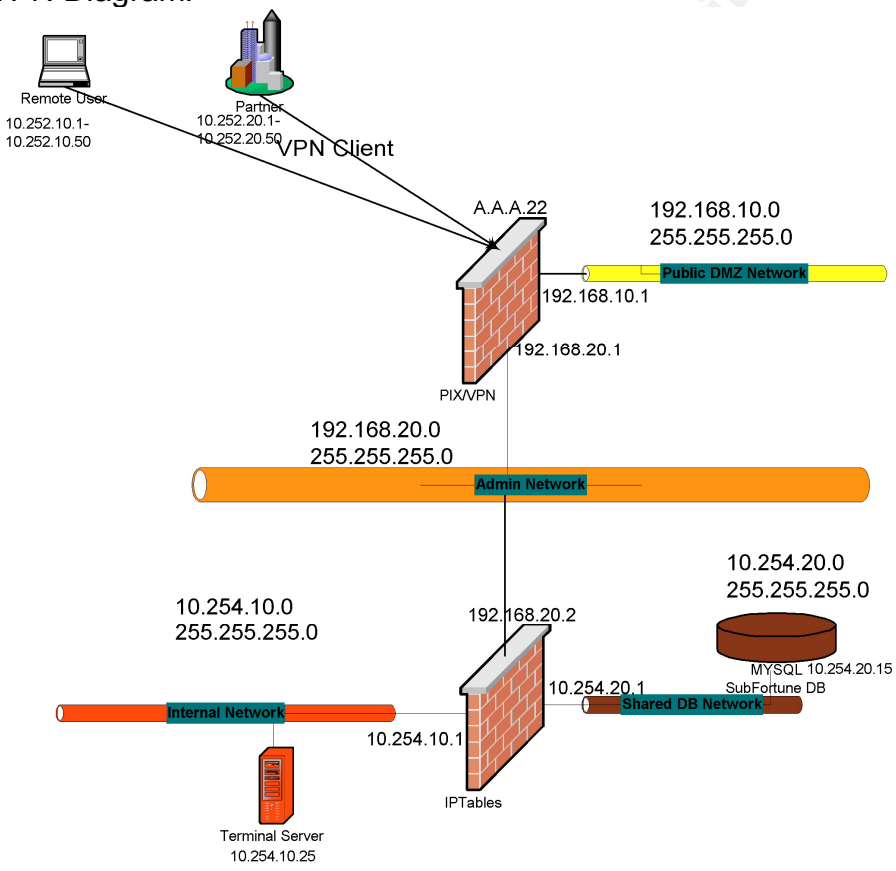
```
iptables –A FORWARD –p TCP –i $INTERNAL_IFACE –s 10.254.10.70/32
–o $EXTERNAL_IFACE -d  0/0 --dport 21 –j ACCEPT
iptables –A FORWARD –p TCP –i $INTERNAL_IFACE –s 10.254.10.71/32
–o $EXTERNAL_IFACE -d  0/0 --dport 21 –j ACCEPT
iptables –A FORWARD –p TCP –i $INTERNAL_IFACE –s 10.254.10.70/32
–o $EXTERNAL_IFACE -d  0/0 --dport 23 –j ACCEPT
iptables –A FORWARD –p TCP –i $INTERNAL_IFACE –s 10.254.10.71/32
–o $EXTERNAL_IFACE -d  0/0 --dport 23 –j ACCEPT
# ALLOW INTERNAL EMAIL SERVER TO SEND MAIL TO ANYWHERE
iptables –A FORWARD –p TCP –i $INTERNAL_IFACE –s 10.254.10.20/32
–o $EXTERNAL_IFACE -d  0/0 --dport 25 –j ACCEPT

# INTERNALNETWORK=>SHAREDDB
# ALLOW INTERNAL WORKSTATIONS TO CONNECT TO THE SALESDB AND
# THE SUBFORTUNEDB IN THE SHAREDDB NETWORK
iptables –A FORWARD –p TCP –i $INTERNAL_IFACE –s 10.254.10.0/24
–o $SHAREDDB_IFACE -d  10.254.20.10/32 --dport 3306 –j ACCEPT
iptables –A FORWARD –p TCP –i $INTERNAL_IFACE –s 10.254.10.0/24
–o $SHAREDDB_IFACE -d  10.254.20.15/32 --dport 3306 –j ACCEPT

# SHAREDDB=> EXTERNAL
# ALLOW SUBFORTUNEDB SSH ACCESS TO SSH SERVER TO PULL IN
# UPLOADED FORTUNES
iptables –A FORWARD –p TCP –i $SHAREDDB_IFACE –s 10.254.20.15/32
–o $EXTERNAL_IFACE -d  192.168.10.10/32 --dport 22 –j ACCEPT
######END OUTBOUND ACCESS CONTROL

######LETS LOG SOME INTERESTING TRAFFIC
# SPOOFING
iptables -A FORWARD -p tcp -i $EXTERNAL_IFACE -s 10.0.0.0/8 -j LOG --log-prefix
“SPOOFATTEMPT
# INVALID TRAFFIC ON OUTSIDE INTERFACE
iptables –A FORWARD –m state --state INVALID –i $EXTERNAL_IFACE –j LOG --log-prefix
“INVALIDTRAFFIC”
# SYNFIN SCANS
iptables –A FORWARD –p TCP --tcp-flags ALL SYN, FIN –j LOG --log-prefix “SYNFINSCAN”
# FIN SCANS
iptables –A FORWARD –p TCP --tcp-flags ACK, FIN FIN –j LOG --log-prefix “FINSCAN”
# ICMP FRAGMENTS
iptables –A FORWARD –p ICMP –f –j LOG --log-prefix “ICMPFRAG”
# FINGER PACKETS
iptables –A FORWARD –p TCP –d 0/0 --dport 79 –j LOG --log-prefix “FINGER”
# SUB-7 PACKETS
iptables –A FORWARD –p UDP –d 0/0 --dport 27374 –j LOG --log-prefix “SUB7”
# NETBUS AND BO2K
iptables –A FORWARD –p TCP –d 0/0 --dport 12345:12346 –j LOG --log-prefix “NETBUS”
iptables –A FORWARD –p TCP –d 0/0 --dport 54320:54321 –j LOG --log-prefix “BO2K”
iptables –A FORWARD –p UDP –d 0/0 --dport 54320:54321 –j LOG --log-prefix “BO2K
######END LETS LOG SOME INTERESTING TRAFFIC

################################END FORWARDING RULES#########

########ROUTING
# We like to do this at the end of our firewall script so we know our rules are in place
# BEFORE we know how to get anywhere…
# Our Default Route, the inside interface on the PIX
```

```
route add default gw 192.168.20.1
# DMZ Network, card ETH 2
route add –net 192.168.10.0 netmask 255.255.0.0 dev $EXTERNAL_IFACE
# VPN for remote users, card ETH 2
route add –net 10.252.10.0 netmask 255.255.255.0 dev $EXTERNAL_IFACE
#VPN for Partners, card ETH 2
route add –net 10.252.20.0 netmask 255.255.255.0 dev $EXTERNAL_IFACE
# Our Inside, SharedDB, and Admin Networks are all directly connected, no need to add
#########END routing

#########CRAZY LOGGING
##This is when we need to see just about everything.
##This should not be turned on at all times due to resource consumption!
##This is utilized when appropriate
#iptables –A INPUT –j LOG –log-prefix “INPUT_DROP: “
#iptables –A FORWARD –j LOG –log-prefix “FORWARD_DROP: “
#iptables –A OUPUT –j LOG –log-prefix “OUTPUT_DROP: “
#########END CRAZY LOGGING

#END Firewall.sh ######################################################
```


2.5 Cisco PIX Client VPN Tutorial:

We chose to implement a Cisco PIX Client VPN Scenario for both our Partner and Remote User connectivity. We liked the hybrid VPN and Firewall solution offered on the PIX 515 platform. This configuration allowed us to configure complete end to end encrypted tunnels utilizing the Cisco VPN client. The VPN client comes with a built in “Stateful” firewall on the client. The client is available for a wide variety of end user OS’s (Windows, Linux, Mac) that enables us to easily deploy the client. We also have implemented a Radius architecture for user accounting and also advanced Access Control (Through use of Reply_Attributes).
We have implemented pre-shared keys, 3-Des encryption, and MD5 hashing. We are also using extended authentication utilizing our Radius Server. Utilizing the extended authentication we are able to control user access down to the user level. The PIX also acts as a DHCP server, and provides an appropriate IP address to the VPN client when connected.

VPN Diagram:



2.5a Cisco PIX and Radius Server Configuration:

1) Open telnet or SSH session with the PIX
2) Lets first enter the access-lists to control the access the clients will have.

```
PIX> en
Password:
PIX# configure terminal
PIX (config) # access-list 120 permit ip 10.252.20.0 255.255.255.0
                host 10.254.20.15 eq 3306
PIX (config) # access-list 120 deny ip any any
PIX (config) # access-list 121 permit ip 10.252.10.0 255.255.255.0
                host 10.254.10.25 eq 3389
PIX (config) # access-list 121 deny ip any any
```

NOTES: Access list 120 allows the partners to only access our SubfortuneDB. Access list 121 allows our mobile employees to only access our Terminal Server. Both of these will be triggered using XAuth with our Radius Server, which we will configure later.

3) Lets add our access-list which will control the NAT process and tell the PIX the proper traffic to put on the IPSEC tunnel.

```
PIX (config) # access-list 110 permit ip 10.0.0.0 255.0.0.0
                10.252.0.0 255.255.0.0
PIX (config) # nat (inside) 0 access-list 110
```

4) Now we will configure our DHCP pools for the clients

```
PIX (config) # ip local pool partnerpool 10.252.20.1-10.252.20.50
PIX (config) # ip local pool remotepool 10.252.10.1-10.252.10.50
```

5) We now have to tell the PIX about our FREERADIUS Server and how to use it.

```
PIX (config) # aaa-server RADIUS protocol radius
PIX (config) # aaa-server login-Radius protocol radius
PIX (config) # aaa-server login-radius (inside) host 192.168.20.10
                mysecretkey timeout 10
```

NOTES: These commands point our PIX to our Radius server. Take note of the mysecretkey. This is the key that the PIX exchanges with the Radius server, this should be as long and cryptic as possible.

6) Lets configure our mappings

```
PIX (config) # sysopt connection permit-ipsec
PIX (config) # crypto ipsec transform-set VPNSet esp-3des esp-md5-
                hmac
PIX (config) # crypto dynamic-map VPNDynMap 1 set transform-set
                VPNSet
PIX (config) # crypto map VPNDynMap 10 ipsec-isakmp dynamic
                VPNDynMap
PIX (config) # crypto map VPNDynMap client authentication login-
                Radius
PIX (config) # crypto map VPNDynMap interface outside
```

NOTES: This enables our VPN mapping. The sysopt permit ipsec is an interesting one. This actually tells the PIX to bypass the normal filtering of the firewall and let the ipsec traffic through. Since we are actually using the Radius server to send ACL mapping info for each of our logins, this suits us. If the command is removed, you can then use your regular ACL's to control your traffic. For whatever reason, this is not documented well on Cisco's web site.

7) Lets configure the isakmp policy

```
PIX (config) # isakmp enable outside
PIX (config) # isakmp identity address
PIX (config) # policy 10 authentication pre-share
```

PIX (config) # policy 10 encryption pre-share
PIX (config) # encryption 3des
PIX (config) # hash md5
PIX (config) # group 2
PIX (config) # lifetime 86400

NOTES: This enables ipsec connections on the outside interface, utilizing 3des with an md5 hashing policy. The lifetime actually tells the Pix and client how often to rotate the keys. We are also utilizing pre-shared keys.

8) Configure our VPN Groups

PIX (config) # vpngroup idle-time 1800
PIX (config) # vpngroup idle-time idle-time 1800
PIX (config) # vpngroup partner address-pool partnerpool
PIX (config) # vpngroup partner dns-server 10.254.10.17
PIX (config) # vpngroup partner default-domain giac.com
PIX (config) # vpngroup partner idle-time 1800
PIX (config) # vpngroup partner password *enterpasshereforgroup*
PIX (config) # vpngroup remote address-pool remotepool
PIX (config) # vpngroup remote dns-server 10.254.10.17
PIX (config) # vpngroup remote default-domain giac.com
PIX (config) # vpngroup remote idle-time 1800
PIX (config) # vpngroup remote password *enterpasshereforgroup*

NOTES: This lets us differentiate security levels between our remote users and clients. Here we define which DHCP pools to use, our password for the client software, and our idle timeouts, which disconnect our users after a set amount of time.

9) That is about it for our PIX Client VPN configuration. Now lets configure Radius server. Dependent on your type of Radius server, your configuration file will be similar to the following:

```
######################## Radius Configuration
# Our Partner VPN Clients
user = partner1 {
        # This is the password user will be prompted for before creating
        # tunnel
        password = clear “passwordpartner1willbeaskedfor”
        # This is how we pass the ACL number back to the PIX for that
        # logon
        Radius=partner {
                reply_attributes= {
                        9,1=”acl=120”
                }
        }
}
user = partner2 {
        # This is the password user will be prompted for before creating
        # tunnel
        password = clear “passwordpartner2willbeaskedfor”
        # This is how we pass the ACL number back to the PIX for that
        # logon
        Radius=partner {
                reply_attributes= {
                        9,1=”acl=120”
                }
        }
}
user = remote {
```

```
        # This is the password user will be prompted for before creating
        # tunnel
        password = clear "passwordremotewillbeaskedfor"
        # This is how we pass the ACL number back to the PIX for that
        # logon
        Radius=remote {
                reply_attributes= {
                        9,1="acl=121"
                }
        }
    }
###############End Radius Configuration
```

2.5b Cisco Client Configuration:

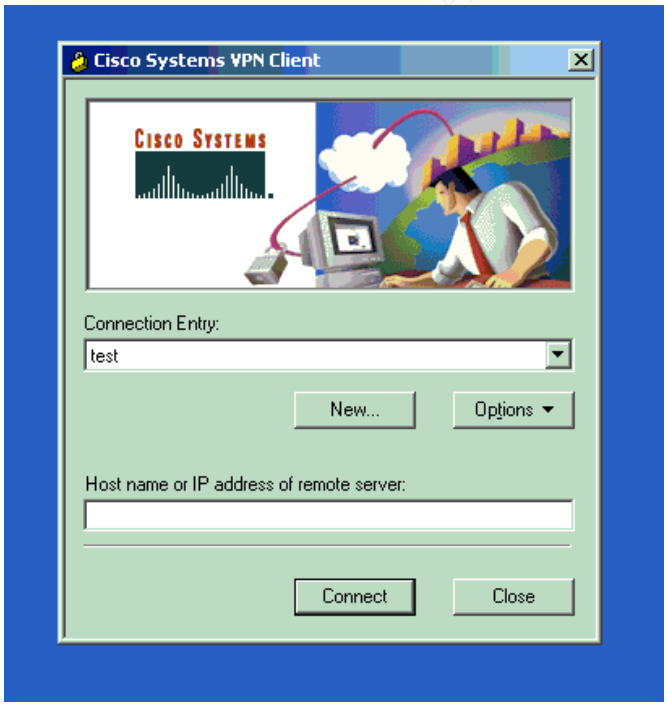
I have included both the manual configuration of the client as well as how large automatic configurations can be accomplished.

Manual Configuration:

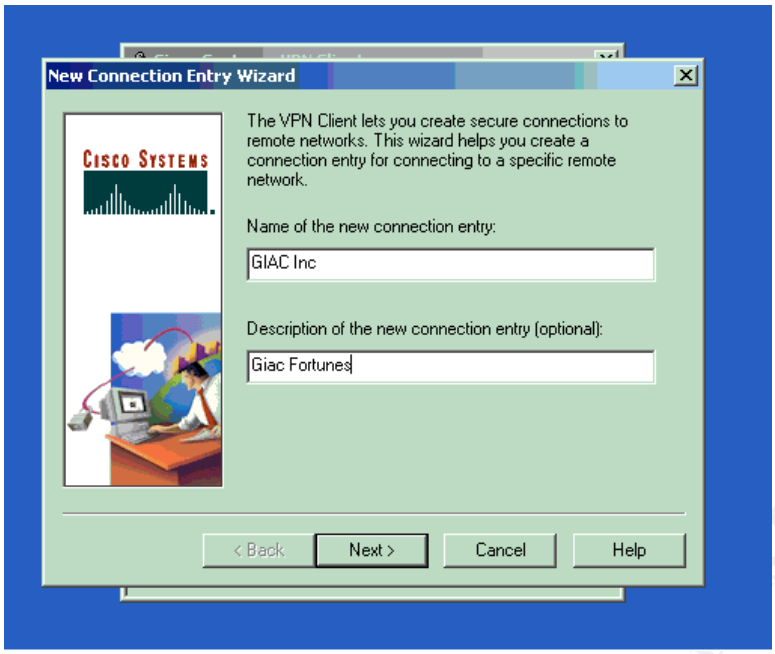
1) Download and install the newest version of the Cisco VPN Client. (http://www.cisco.com/kobayashi/sw-center/vpn/client/)

2) Double click the VPN Dialer. This is where we will configure our connection.

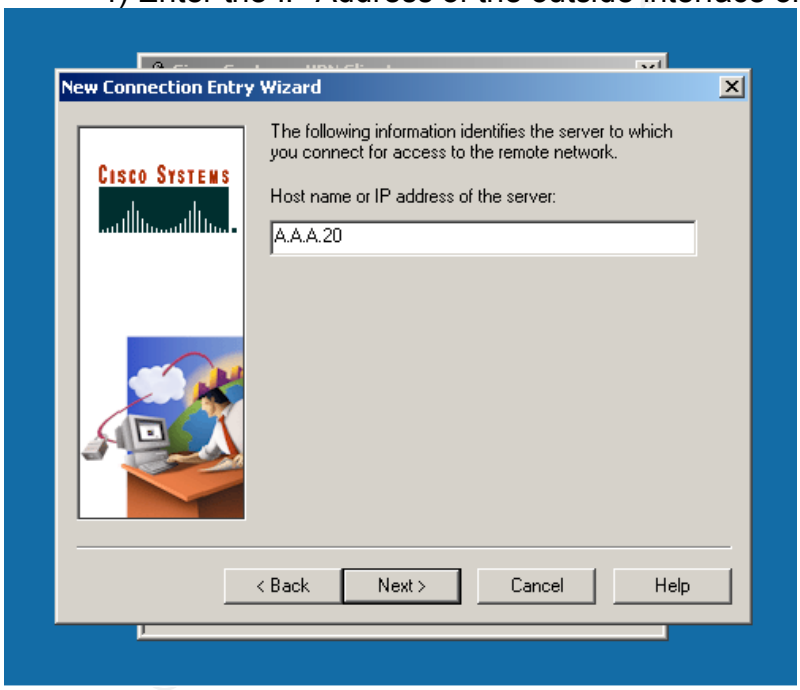
Click New -> Connection



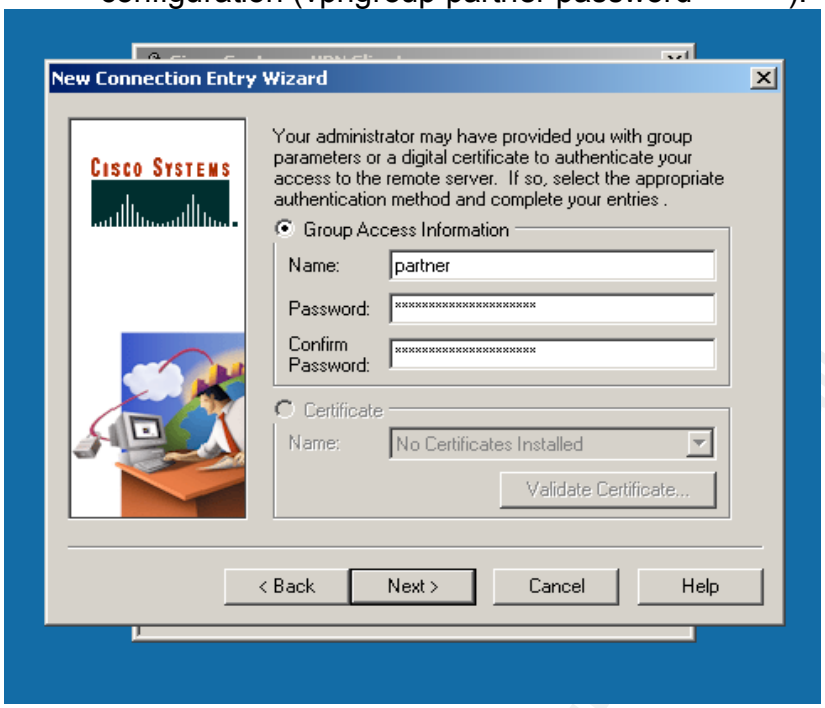
3) Enter the name you would like to refer to the connection as well as a description if desired.



4) Enter the IP Address of the outside interface on the PIX.

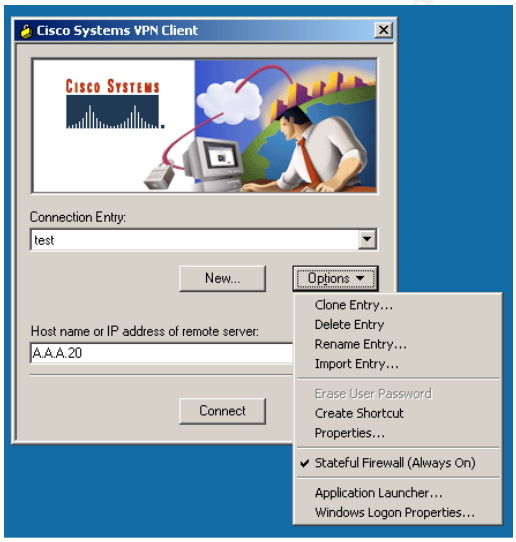


5) Here we enter the group logon information. Name is the name of the VPN group and the Password is the Password entered in the group configuration (vpngroup partner password *******).

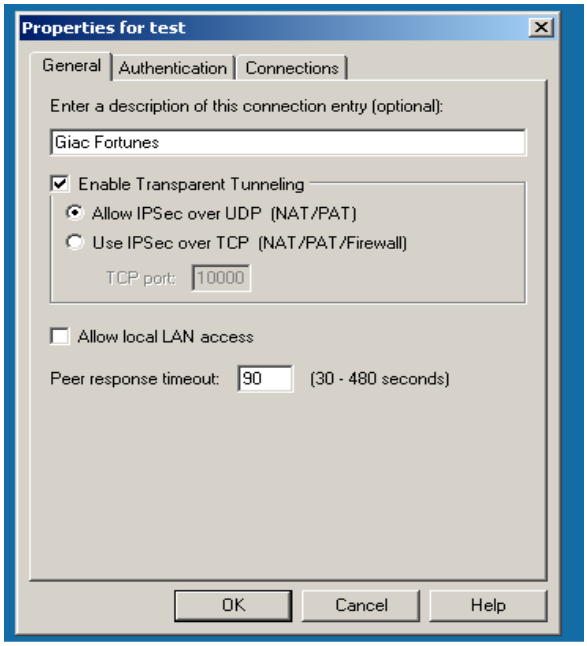


6) Click finish to complete the connection.

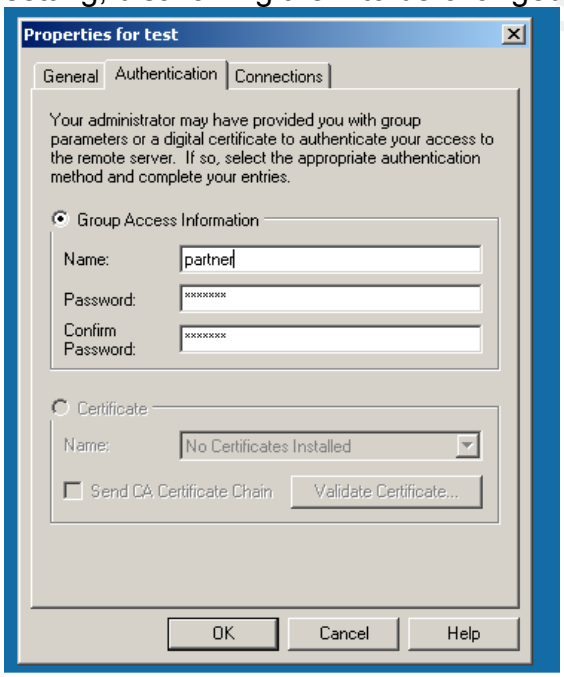
7) Reviewing the Configuration our properties now look like the following:



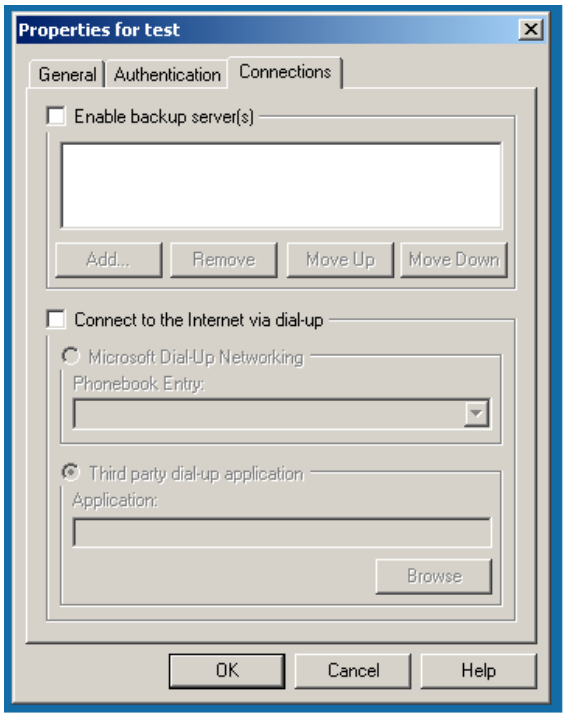
We can see here the Stateful Firewall is on. We can also manipulate the profile, create a shortcut on the desktop, launch a specific application at connection time, or configure logon setting if we wish to authenticate on a Windows Domain.



Our connection description. We are utilizing UDP for our connection, this is the Cisco default. We also have not allowed local LAN access. This disables the ability of the client to connect to local resources, ie servers, printers etc. Pre configuring the client, as we will see later, allows us to lock in most of these setting, disallowing them to be changed from within the client.



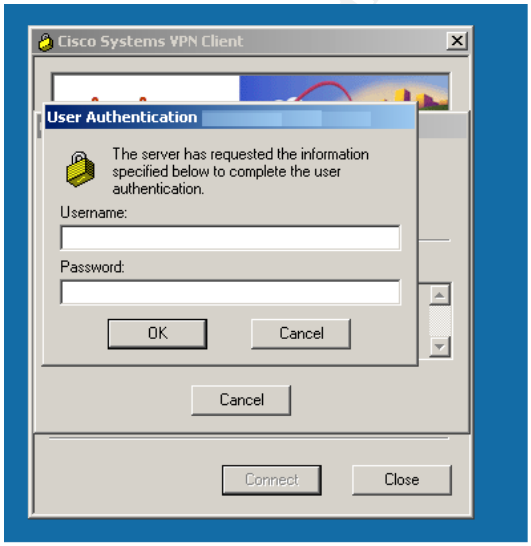
Our VPN Group authentication information. We are currently not utilizing Certificates, but as our user base grows it is a future consideration. This is the username and password entered in the Group configuration of the PIX itself.



And finally our connection information. We can configure things here like a backup VPN server (or in our case a PIX), and the option to utilize a dial-up connection.

8) We are now ready to initiate our connection. Click the DOWN arrow and select the connection we just configured in the dialer. Hit the connect button.

9) You will be prompted for a user name and password. This is the username and password we entered in the configuration file of the Radius server.



10) That's it! Congratulations, you are now connected to the GIAC network.

Automatic Configuration:

The Cisco VPN client also supports large scale rollouts. By defining a vpnclient.ini and the proper profile within the install directory, the client will adhere to that ini through the installation process. This automates the task and disallows improper access. There are many options that can be configured in these files that cannot be accomplished through a manual configuration. A typical installation file and profile may look like the following:

Sample vpnclient.ini file (Global Profile)

This sample file shows what you might see if you open it with a text editor. Note the exclamation point [!] before the parameters. This prevents the user from changing these attributes from within the application.

```
[main]
;We could actually start the client before logging onto the network
; 0 = no
!RunAtLogon=0
;Determines whether the Stateful firewall is on.
;When enabled, the stateful firewall always on feature allows no
;inbound sessions from anywhere, regardless if the tunnel is on or
;not. This protects us from intruders penetrating our network by
;compromising the client. 1 = on
!StatefulFirewall=1
;Turn logging on. This is added in case you want to override logging,
;for performance reasons. 1 = enable
!EnableLog=1
;This is our profile
[partner]
;PCF profile to use
!ConnectionEntry=partner
;These define our logging levels
; 3 = High - all events
[LOG.IKE]
!LogLevel=3
[LOG.CM]
!LogLevel=3
[LOG.DIALER]
!LogLevel=3
[LOG.CVPND]
!LogLevel=3
[LOG.CERT]
!LogLevel=3
[LOG.IPSEC]
!LogLevel=3
[LOG.FIREWALL]
!LogLevel=3
[LOG.CLI]
```

```
!LogLevel=3
```

Sample profile.pcf file (Individual Profile)

This sample file shows what you might see if you open it with a text editor. . Note the exclamation point [!] before the parameters. This prevents the user from changing these attributes from within the application.

```
[main]
;This is what our entry is called
!Description=partner
;Address to connect to
!Host=A.A.A.20
;Authentication method. 1 = PreShared Keys
!AuthType=1
;Group Name
!GroupName=partner
;This is entered the first time. After that the client encrypts it and
;places it in the enc_GroupPwd below
!GroupPwd=
!enc_GroupPwd=55153F4DCF8CADB4B5B511BC1176866FB7D16FDEF55945919BAFCCECD
6FCEF7EBFB42E9B66FE5885BEA46C2855FCA102AA2A85C3F78AB5931CDD8EB05F1915A8
4A7D0DC6A545CCE6
;This tells the client to initiate a dialup networking session when
;trying to connect. All our users are HB so 0 = disable
;Allows secure transmission between the VPN Client and a secure gateway ;through a router
serving as a firewall, which may also be performing
;NAT or PAT. 1= enable
!EnableNat=1
;UDP or TCP connection. 0 = UDP.
!TunnelingMode=0
;Allows the VPN Client to keep sending IKE and ESP
;keepalives for a connection at approximately 20
;second intervals so the port on an ESP-aware
;NAT/Firewall does not close.
!ForceKeepAlives=0
;Timeout if we lose contact with the peer
!PeerTimeout=90
;Enable access to local LAN. 0 = disable.
;This is cause a security problem if we allow this type
;of access, a host could be easily compromised allowing access
;to our network
!EnableLocalLAN=0
```

It can be easily seen how when these profiles are included with the installation, the client install is seamless and no end user configuration is required.
There are many other variables that can be included in these files like, automatic application start with connect, and the ability to seamlessly negotiate a connection on a windows domain. Our requirements just needed TCP/IP connectivity with our network devices, so domain integration was not necessary.

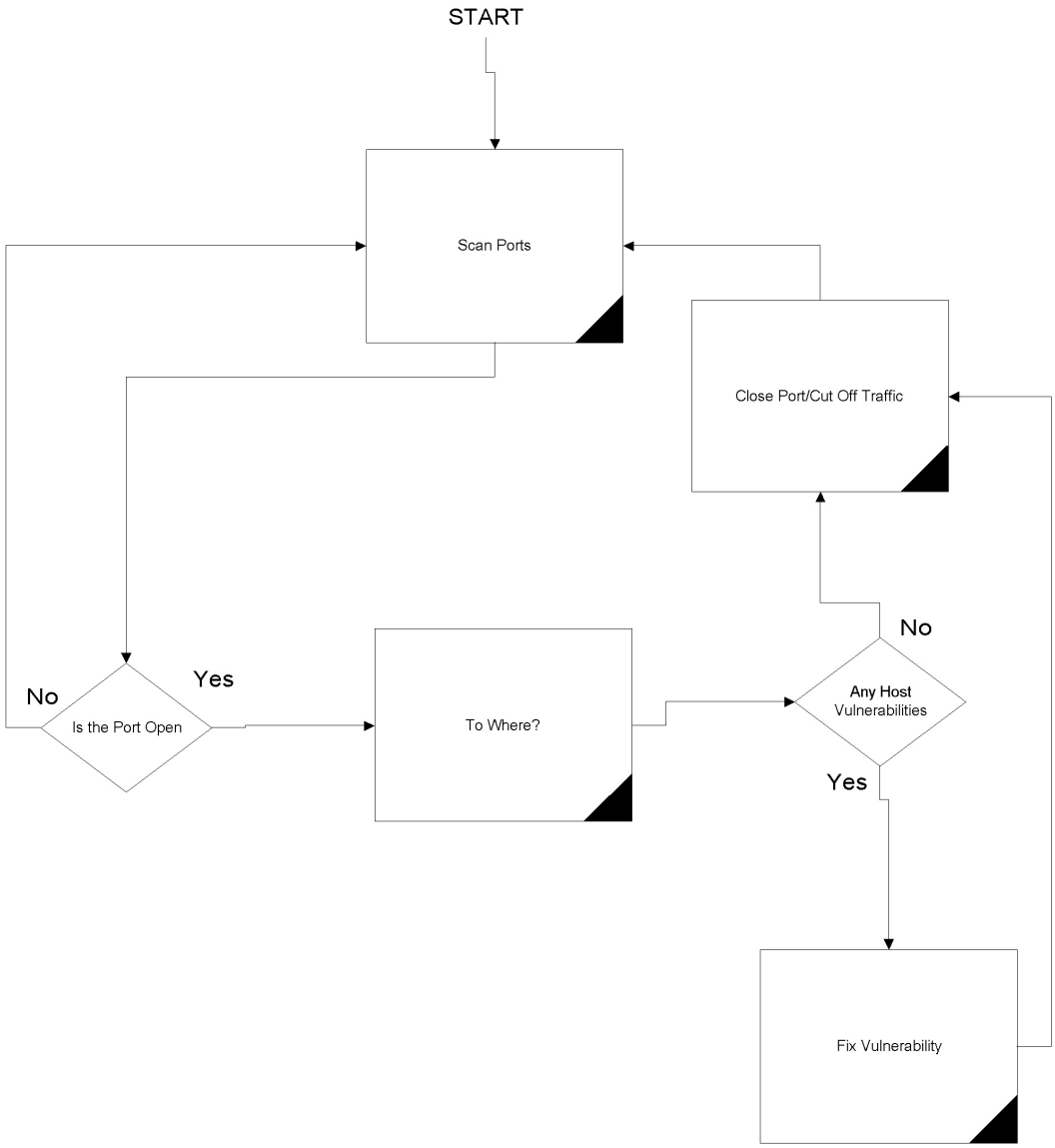
3. Firewall Policy Verification

3.1 Introduction:

The first step in a Firewall Audit is defining what our expectations our. In our GIAC scenario we want to verify that mandatory traffic is able to pass, and bad traffic is dropped. For our audit we will be auditing our PIX 515 firewall, the real workhorse of our architecture.

3.2 Plan the Audit:

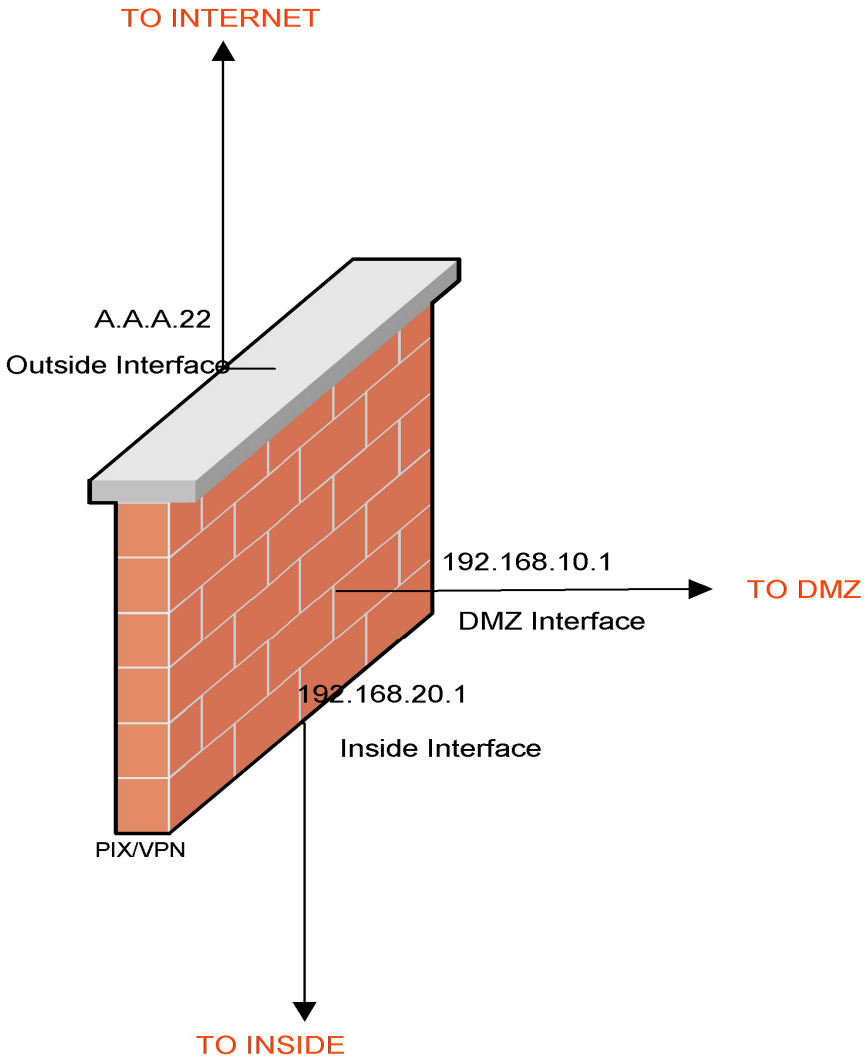
A Typical Firewall Audit scenario would consist of something similar to the following diagram:



For the scope of our Audit we are only concerned with the traffic flow, and not the host vulnerabilities, with the exception being the PIX itself. We want to verify the security of the Firewall itself, and we want to test the rule base, what traffic is allowed through the network and what traffic is dropped.

To perform we must verify the traffic flow through all of the following:

1) Outside (Internet) => DMZ network
2) Outside (Internet) => Inside
3) DMZ => Outside (Internet)
4) DMZ => Inside
5) Inside => Outside (Internet)
6) Inside => DMZ



This will verify traffic flow through all of the interfaces on Cisco PIX. In addition we must verify that the PIX itself is only allowing connection from the proper devices, ie the IT workstations.

The audit will occur on off hours. We have made certain our systems reside is keycard secured areas, as-if an intruder can gain physical access, game over. We also will verify we have full backups of all devices and systems involved in the audit.

Cost: The cost of the audit is nominal. We are conducting the audit off hours, as to not cause any work interruption. We are also utilizing open source resources on a mobile laptop. The real cost is associated with the man power to perform the audit, we are gaining no down time. We are also interesting in performing some of these tests throughout the day to compare results in our dynamic environment.

Tools:
To perform our audit we will be utilizing the NMAP port scanner. This is a very robust scanner. In addition, it is a very good tool to utilize since most of the "bad guys" are also using it. This will allow us to see what they would. It is also very well documented.

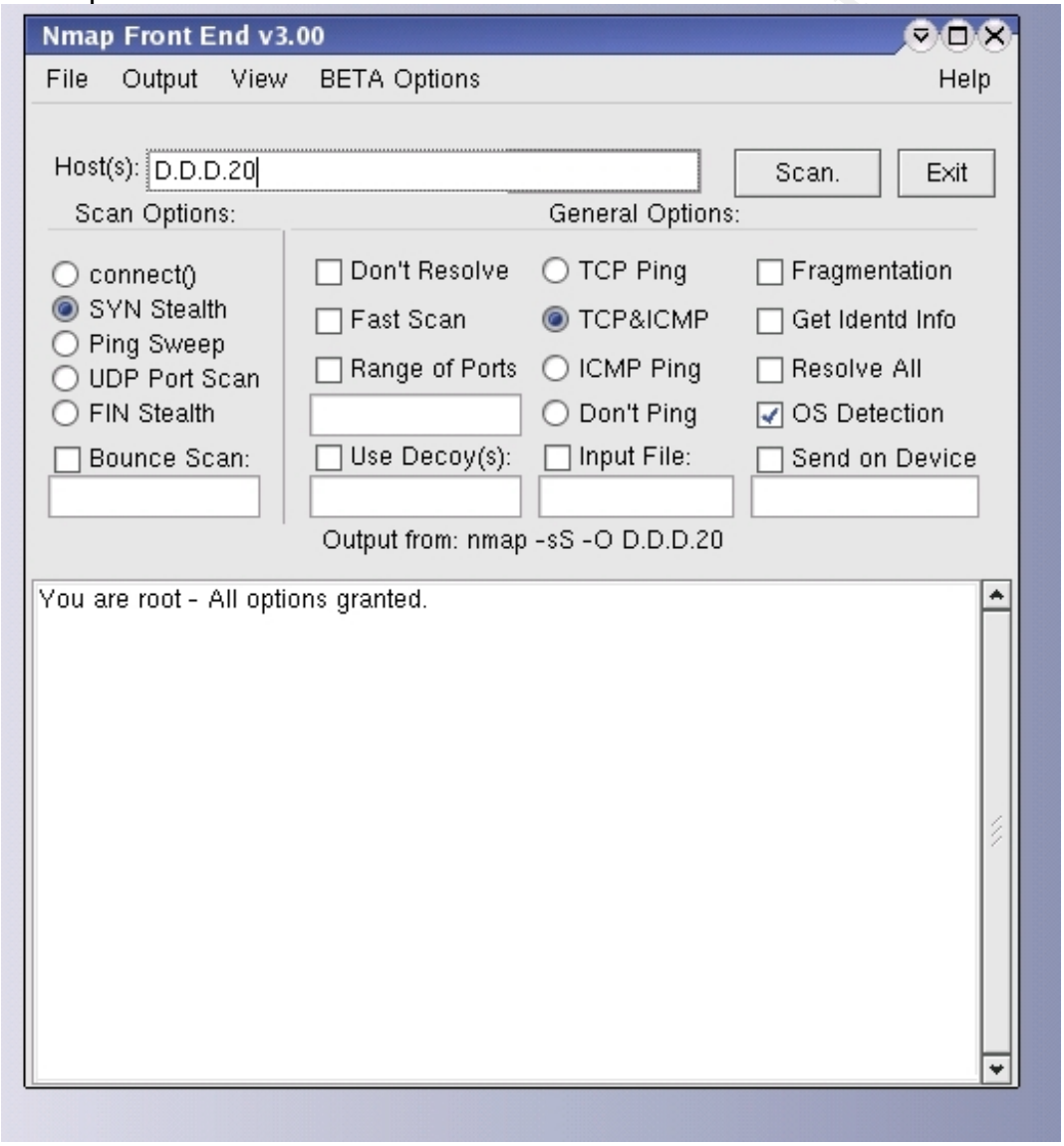
Since we are only concerned with traffic flow, and not vulnerability testing, this tool fits the bill perfectly. We would a tool such as Nesses or Whisker if we were going to go a step further.

An worthy tool that can be used with NMAP is XNMAP. This provides a GUI front-end (GTK+) for the NMAP process, and is very intuitive to use. While not providing all the functionality as the command line usage, it can be used to quickly see the arguments that can be used easily. I have provided both a listing of some of the more useful command line arguments as well as a screen shot of XNMAP.

NMAP Command Line Arguments:
[root@LUPO stany]# nmap --help
Nmap V. 3.00 Usage: nmap [Scan Type(s)] [Options] <host or net list>
Some Common Scan Types ('*' options require root privileges)
* -sS TCP SYN stealth port scan (default if privileged (root))
  -sT TCP connect() port scan (default for unprivileged users)
* -sU UDP port scan
  -sP ping scan (Find any reachable machines)
* -sF,-sX,-sN Stealth FIN, Xmas, or Null scan (experts only)
  -sR/-I RPC/Identd scan (use with other scan types)
Some Common Options (none are required, most can be combined):
* -O Use TCP/IP fingerprinting to guess remote operating system
  -p <range> ports to scan. Example range: '1-1024,1080,6666,31337'
  -F Only scans ports listed in nmap-services
  -v Verbose. Its use is recommended. Use twice for greater effect.
  -P0 Don't ping hosts (needed to scan www.microsoft.com and others)
* -Ddecoy_host1,decoy2[,...] Hide scan using many decoys

-T <Paranoid|Sneaky|Polite|Normal|Aggressive|Insane> General timing policy
-n/-R Never do DNS resolution/Always resolve [default: sometimes resolve]
-oN/-oX/-oG <logfile> Output normal/XML/grepable scan logs to <logfile>
-iL <inputfile> Get targets from file; Use '-' for stdin
* -S <your_IP>/-e <devicename> Specify source address or network interface
--interactive Go into interactive mode (then press h for help)
Example: nmap -v -sS -O www.my.com 192.168.0.0/16 '192.88-90.*.*'
SEE THE MAN PAGE FOR MANY MORE OPTIONS, DESCRIPTIONS, AND EXAMPLES
[root@LUPO stany]#

XNmap Screenshot:



We will be mainly utilizing the –sU and –sS command line arguments, UDP port scan and TCP Syn stealth scan.

We will utilize a NMAP 3.00 on a laptop running Redhat 8.0 as well as our hosts themselves. All security tools we will utilize have been burned on a CD. This ensures they have not been tampered with. One cannot trust their audit results if one does not trust their tools. The laptop will be utilized on the Public Internet, DMZ Network, and well as behind the inside interface. For all our access requirements, we will scan the hosts with NMAP from the hosts requiring access. For the reviewer's sanity, the scans are the same for each of the hosts needing access.

3.3 Conduct the Audit:

To conduct our audit, we will verify the following access requirements are met and not exceeded:

General ACL Requirements For PIX FireWall

Inbound

Source	Source Location	Destination	Dest Location	Port(s)	Explanation
ANY	ANY	SMTP Relay	DMZ Network	25 TCP	Allow our relay server to accept inbound email
Customers	ANY	HTTP or HTTPS Server	DMZ Network	80 TCP and 443 TCP	Allow customers to connect to HTTP and HTTPS Server
HTTP Server	DMZ Network	Syslog1 Server	Admin Network	514 UDP	Enable Server to send messages to SYSLOG server
HTTPS Server	DMZ Network	Syslog1 Server	Admin Network	514 UDP	Enable Server to send messages to SYSLOG server
HTTPS Server	DMZ Network	MySQL SubfortuneDB	SharedDB Network	22 TCP	Allow Secure Web Server to connect to MySQL SubfortuneDB
HTTPS Server	DMZ Network	MySQL SalesDB	SharedDB Network	22 TCP	Allow Secure Web Server to connect to MySQL SalesDB
Partners	Partner Network	MySQL SubfortuneDB	SharedDB Network	3306 TCP	Enable Direct Connection to SubfortuneDB
Remote Users (Mobile Employees)	Remote Users Network	Terminal Server	Internal Network	3389 TCP	Allow Mobile Employees to Connect to Terminal Server

SMTP Relay	DMZ Network	Syslog1 Server	Admin Network	515 UDP	Enable Server to send messages to SYSLOG server
SMTP Relay	DMZ Network	Internal Email Server	Internal Network	25 TCP	Enable SMTP Relay to relay to the internal server
SSH Server	DMZ Network	Syslog1 Server	Admin Network	514 UDP	Enable Server to send messages to SYSLOG server
Supplier	Suppliers ISPS's	SSH Server	DMZ Network	22 TCP	Enable Suppliers to upload Fortunes to SSH Server

Outbound

Source	Source Location	Destination	Dest Location	Port(s)	Explanation
Email Server	Internal Network	ANY	ANY	25 TCP	Enable Internal Exchange Server to Send Email
HTTP Server	DMZ Network	ISP Hosted DNS Server	ISP Hosted DNS Server	53 TCP and 53 UDP	Enable DNS
HTTPS Server	DMZ Network	ISP Hosted DNS Server	ISP Hosted DNS Server	53 TCP and 53 UDP	Enable DNS
Internal Network	Internal Network	Any HTTP or HTTPS Server	ANY	80 TCP and 443 TCP	Enable Internet Browsing
Internal Network	Internal Network	ISP Hosted DNS Server	ISP Hosted DNS Server	53 TCP and 53 UDP	Enable DNS
Limited "IT" Workstations	Internal Network	SSH Servers	ANY	22 TCP	Allow IT Admin through SSH
Limited "IT" Workstations	Internal Network	FTP Servers	ANY	20 TCP and 21 TCP	Allow Connection to any FTP Server (Software, Patches etc..)
Limited "IT" Workstations	Internal Network	Telnet	ANY	23 TCP	Allow IT to telnet
MySQL SubfortuneDB	SharedDB Network	SSH Server	DMZ Network	22 TCP	Enable automated 'bot' on SubFortune DB to pull uploaded fortunes in via an encrypted SSH connection
SMTP Relay	DMZ Network	ISP Hosted DNS Server	ISP Hosted DNS Server	53 TCP and 53 UDP	Enable DNS

1) Outside (Internet) => DMZ network – Laptop placed on Internet as well as behind Suppliers ISP's.

Verify necessary connections from the Public Internet:
Verify customers can connect to web servers (HTTP and HTTPS):
telnet A.A.A.26 80
Trying A.A.A.26…
Connected to A.A.A.26

telnet A.A.A.24 443
Trying A.A.A.24…
Connected to A.A.A.24
Excellent, we are able to connect to the web server and secure web server.

Verify email can be sent to Relay Server:
telnet A.A.A.25 25
Trying A.A.A.25…
Connected to A.A.A.25
Escape character is '^]'.
220 **********
Good sign, we have successfully connected to our Postfix Mail Server.

Verify suppliers can connect to SSH server from their ISPS:
telnet A.A.A.23 22
Trying A.A.A.25…
Connected to A.A.A.25
Escape character is '^]'.
SSH-1.99-OpenSSH_3.1p1
So utilizing our suppliers ISP's we were able to connect.

The other type of access would be for our Partners and Remote Users to be able to successfully connect to our pix and initiate a VPN session.
The simplest way to accomplish this was to connect directly with the client. We had Belize fortunes connect, and they were successful. Also eyeing our Radius logs, the following entry:
Sat Mar 01 22:02:09 : Auth: Login OK : [BelizeFor] (from client 192.168.20.1 port 5)
Verifies that the IPSEC tunnel is functioning properly.

So it appears all of our wanted traffic flow is flowing through. We will now utilize nmap as well as ping to verify we are not allowing unnecessary connections:

For the Pix itself, as well as all DMZ hosts, we will perform the following:

1) Ping the device
2) Perform an NMAP TCP port scan of the device. (SYN Stealth)
3) Perform an NMAP UDP port scan of the device.

PIX
Lets first Ping the outside interface on the pix:
ping –t 20 A.A.A.22
ping A.A.A.22 (A.A.A.22) from A.A.A.154 : 56(84) bytes of data.
From A.A.A.22 icmp_seg=1 Time to live exceeded
From A.A.A.22 icmp_seg=2 Time to live exceeded
From A.A.A.22 icmp_seg=3 Time to live exceeded
From A.A.A.22 icmp_seg=4 Time to live exceeded
Good deal. We are getting no reply.

Now lets see if we can detect our host by scanning the outside interface of the PIX. This is a major advantage for a would be intruder, compromising a known host type could be trivial:
nmap –O A.A.A.22

Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 scanned ports on (A.A.A.22) are :filtered
Too many fingerprints match this host for me to give an accurate OS guess
So it appears our host detection failed.

Now let scan for open ports at that IP address:
UDP
nmap –sU –p 1-65535 A.A.A.22

Starting nmap V. 3.00 (www.insecure.org/nmap/)
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (0 hosts up) scanned in 30 seconds

TCP
nmap –sS –p 1-65535 A.A.A.22
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (0 hosts up) scanned in 30 seconds

So we did not find any open ports. This is surprising considering we have IPSEC port connections being accepted for our Partners and Remote users. Looks like our Firewall is doing a pretty good job of hiding that access.

DMZ Servers
Lets first Ping the ips:
SSH Server
ping –t 20 A.A.A.23
ping A.A.A.23 (A.A.A.23) from A.A.A.154 : 56(84) bytes of data.
From A.A.A.23 icmp_seg=1 Time to live exceeded
From A.A.A.23 icmp_seg=2 Time to live exceeded
From A.A.A.23 icmp_seg=3 Time to live exceeded
From A.A.A.23 icmp_seg=4 Time to live exceeded
HTTPS Server
ping –t 20 A.A.A.24
ping A.A.A.23 (A.A.A.24) from A.A.A.154 : 56(84) bytes of data.
From A.A.A.24 icmp_seg=1 Time to live exceeded
From A.A.A.24 icmp_seg=2 Time to live exceeded
From A.A.A.24 icmp_seg=3 Time to live exceeded
From A.A.A.24 icmp_seg=4 Time to live exceeded
SMTP Relay Server
ping –t 20 A.A.A.25
ping A.A.A.25 (A.A.A.25) from A.A.A.154 : 56(84) bytes of data.
From A.A.A.25 icmp_seg=1 Time to live exceeded
From A.A.A.25 icmp_seg=2 Time to live exceeded
From A.A.A.25 icmp_seg=3 Time to live exceeded
From A.A.A.25 icmp_seg=4 Time to live exceeded
HTTPS Server
ping –t 20 A.A.A.26
ping A.A.A.26 (A.A.A.26) from A.A.A.154 : 56(84) bytes of data.
From A.A.A.26 icmp_seg=1 Time to live exceeded
From A.A.A.26 icmp_seg=2 Time to live exceeded
From A.A.A.26 icmp_seg=3 Time to live exceeded
From A.A.A.26 icmp_seg=4 Time to live exceeded
So no ICMP packets are being allowed through.

Now let scan for open ports at the IP addresses:
SSH Server
UDP
nmap –sU –p 1-65535 A.A.A.23
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (0 hosts up) scanned in 30 seconds.

TCP
nmap –sS –p 1-65535 A.A.A.23
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Interesting ports on (A.A.A.23):

(The 65535 ports scanned but not shown below are in state: filtered)
Port State Service
22/tcp open ssh
Nmap run completed – 1 IP address (1 host up) scanned in 867 seconds.
HTTPS Server
UDP
nmap –sU –p 1-65535 A.A.A.24
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (0 hosts up) scanned in 34 seconds.
TCP
nmap –sS –p 1-65535 A.A.A.24
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Interesting ports on (A.A.A.24):
(The 65535 ports scanned but not shown below are in state: filtered)
Port State Service
443/tcp open https
Nmap run completed – 1 IP address (1 host up) scanned in 897 seconds.
SMTP Relay Server
UDP
nmap –sU –p 1-65535 A.A.A.25
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (0 hosts up) scanned in 34 seconds.
TCP
nmap –sS –p 1-65535 A.A.A.25
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (0 hosts up) scanned in 34 seconds.
HTTP Server
UDP
nmap –sU –p 1-65535 A.A.A.26
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (0 hosts up) scanned in 37 seconds.
TCP
nmap –sS –p 1-65535 A.A.A.26
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Interesting ports on (A.A.A.26):
(The 65535 ports scanned but not shown below are in state: filtered)
Port State Service
80/tcp open http
Nmap run completed – 1 IP address (1 host up) scanned in 669 seconds.

So it appears only the ports we need open are. Our traffic flow from the internet through the Pix has checked out OK. An interesting note is the scan of our SMTP

Relay server. Our filtering firewall is actually making Nmap believe there is nothing there at all.
Lets try a scan just to port 25:
nmap –sS –p 25 A.A.A.25
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (0 hosts up) scanned in 34 seconds
Same result.
Lets send a packet with the ACK flag set.
nmap –sA –p 25 A.A.A.25
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (0 hosts up) scanned in 34 seconds
Same result again. Our mailguard feature seems to be doing a pretty good job of hiding our host.
An important feature that we have seen from above is the suppliers ability to use an encrypted SSH connection to the SSH server. While enabling a secure encrypted connection, it disallows us from using our fixup protocol on the PIX to look for FTP inaccuracies. In this instance we felt it was a bigger risk to have our Fortunes sniffed.

2) Outside (Internet) => Inside

Verify necessary connections from the Public Internet:
Verify Partners can connect to the MySQL SubfortuneDB:
After initiating our Partners VPN connection
telnet 10.254.20.15 3306
Trying 10.254.20.15 …
Connected to 10.254.20.15
Good, we have connected to our database through the 3306 MySQL port.
Verify Remote Users can connect to our Terminal Server:
After initiating our Partners VPN connection
telnet 10.254.10.25 3389
Trying 10.254.10.25 …
Connected to 10.254.10.25
Looks like the proper connectivity exists.
Lets see what happens when our Partners try to connect to our precious SalesDB.
telnet 10.254.20.10
Trying 10.254.20.10 3306 …
Connection Timed Out.
Excellent, looks like the proper connectivity exists. This is one asset we would rather keep to ourselves.

So now that we verified that our required connectivity is there, lets do some penetration testing on the hosts. To accomplish this we will take the following steps:

For each device we will perform the following:

1) Ping the device
2) Perform an NMAP TCP port scan of the device. (SYN Stealth)
3) Perform an NMAP UDP port scan of the device.

Inside Servers

Lets first Ping the ips:

SubfortuneDB

```
ping –t 20 10.254.20.15
ping 10.254.20.15 (10.254.20.15) from 10.252.20.8 : 56(84) bytes of data.
From 10.254.20.15 icmp_seg=1 Time to live exceeded
From 10.254.20.15 icmp_seg=2 Time to live exceeded
From 10.254.20.15 icmp_seg=3 Time to live exceeded
From 10.254.20.15 icmp_seg=4 Time to live exceeded
```

Terminal Server

```
ping –t 20 10.254.10.25
ping 10.254.10.25 (10.254.10.25) from 10.252.10.9 : 56(84) bytes of data.
From 10.254.10.25 icmp_seg=1 Time to live exceeded
From 10.254.10.25 icmp_seg=2 Time to live exceeded
From 10.254.10.25 icmp_seg=3 Time to live exceeded
From 10.254.10.25 icmp_seg=4 Time to live exceeded
```

So as expected we are dropping ICMP as expected.

Now let scan for open ports at the IP addresses:

SubfortuneDB MySQL Database

UDP

```
nmap –sU –p 1-65535 10.254.20.15
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (0 hosts up) scanned in 37 seconds.
```

TCP

```
nmap –sS –p 1-65535 10.254.10.25
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on (10.254.10.25):
(The 65535 ports scanned but not shown below are in state: filtered)
Port       State       Service
3306/tcp open          mysql
Nmap run completed – 1 IP address (1 host up) scanned in 669 seconds.
```

Perfect. As expected the only result we are receiving is from the active MySQL database, just as we expected.

Terminal Server
UDP
nmap –sU –p 1-65535 10.254.10.25
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (0 hosts up) scanned in 566 seconds.
TCP
nmap –sS –p 1-65535 10.254.10.25
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Interesting ports on (10.254.10.25):
(The 65535 ports scanned but not shown below are in state: filtered)
Port State Service
3389/tcp open ms-term-serv
Nmap run completed – 1 IP address (1 host up) scanned in 55 seconds.

Again, no unexpected results.

So we can assume our access requirements for our trusty VPN users are met, but most importantly, not exceeded.

3) DMZ => Outside (Internet)

Verify necessary connections from the DMZ to the Internet:
After mulling over our access requirements to the outside, we realized we did not have any other than DNS.

Verify ISP DNS connectivity for ISP DNS Server:
SMTP Relay server:
telnet D.D.D.50 53
Trying D.D.D.50 …
Connected to D.D.D.50
Good, we have connected to our ISP's DNS Server.
HTTP and HTTPS Servers:
telnet D.D.D.50 53
Trying D.D.D.50 …
Connected to D.D.D.50
telnet D.D.D.50 53
Trying D.D.D.50 …
Connected to D.D.D.50
Good, we have connected to our ISP's DNS Server.

Let's try to connect to our friend DNS Server from our SMTP relay:
telnet F.F.F.13 53
Trying F.F.F.13 …
telnet: connect to address F.F.F.13: Connection refused
So our connection to that DNS server was disallowed. Parsing our PIX log yields:

Mar 05 23:00:12 192.168.20.1 Mar 06 2003 11:00:34 %PIX-4-106023: Deny tcp src DMZ:192.168.10.20/161 dst outside F.F.F.13/162 by access-group “103”
So here we can really see our PIX at work.
Now we will try from our SSH server:
telnet D.D.D.50 53
Trying D.D.D.50 …
telnet: connect to address D.D.D.50 …: Connection refused
Just as we expected. We have no access requirements for DNS on this server, thus the access is not allowed.

We do not want to scan our ISPs, DNS server. There remains legality issues in our contract that prohibits such behavior. We can now take a look at the PIX’s DMZ interface itself.

Lets first Ping the DMZ interface on the pix:
ping –t 20 192.168.10.1
ping 192.168.10.1(192.168.10.1) from 192.168.10.50 : 56(84) bytes of data.
From 192.168.10.1 icmp_seg=1 Time to live exceeded
From 192.168.10.1 icmp_seg=2 Time to live exceeded
From 192.168.10.1 icmp_seg=3 Time to live exceeded
From 192.168.10.1 icmp_seg=4 Time to live exceeded
Good deal. We are getting no reply.

Now lets see if we can detect our host by scanning the outside interface of the PIX. This is a major advantage for a would be intruder, compromising a known host type could be trivial:
nmap –O 192.168.10.1

Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 scanned ports on (192.168.10.1) are :filtered
Too many fingerprints match this host for me to give an accurate OS guess
So it appears our host detection failed.

Now let scan for open ports at that IP address:
UDP
nmap –sU –p 1-65535 192.168.10.1

Starting nmap V. 3.00 (www.insecure.org/nmap/)
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (0 hosts up) scanned in 30 seconds

TCP
nmap –sS –p 1-65535 192.168.10.1
Starting nmap V. 3.00 (www.insecure.org/nmap/)

Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (0 hosts up) scanned in 30 seconds

So we did not find any problems. Just as we expected no open ports. Looking though the firewall logs we see our PIX is hard at work controlling this access. This also taught us while a firewall audit is not only invaluable for querying your security architecture, it is also a good indication that we are providing the access we need. When dealing with complex configurations like ours, an audit can really give a good perspective on whether your requirements are being met.

4) DMZ => Inside (Inside Interface on PIX)

Verify necessary connections from the DMZ :
Verify SMTP Relay can connect to internal Email Server:
telnet
Trying 10.254.10.20…
Connected to 10.254.10.20
Escape character is '^]'.
220 **********
We have successfully connected. Now would be a good time to try out our Mail Guard feature on our PIX. This is a STATEFUL inspection of our SMTP connection.
We will type in a bogus command at the mail prompt:
Escape character is '^]'.
220 **********
rebootserver

OK

Kind of strange huh? The reboot command actually was returned with an OK. This is good and bad. It takes any non SMTP compliant command and intercepts it so it never reaches your mail server, it always returns an OK. This is good because it disallows harmful commands to your server, it is bad because you can OS fingerprint a PIX with this service running pretty easily.

Verify HTTPS server can SSH to our database servers:
SubfortuneDB
telnet 10.254.20.15 22
Trying 10.254.20.15…
Connected to 10.254.20.15
SSH-1.99-OpenSSH_3.1p1
SalesDB
telnet 10.254.20.10 22
Trying 10.254.20.10…
Connected to 10.254.20.10
SSH-1.99-OpenSSH_3.1p1

Connectivity exists there.

Verify all our deemed appropriate servers can connect to our SYSLOG1 server. Since we are concerned with Syslog UDP connections, we verified connectivity utilizing our excellent logs on the client and SYLOG1 server.
Utilizing the logs, we are successfully receiving Syslog messages from the HTTP, HTTPS, SMTP, SSH Server, and from the PIX itself. All messages being sent to the client will have an entry in the /var/log/messages file, as well as any device specific log file we are using.

Lets do some access testing.
For each device we will perform the following:

1) Ping the device
2) Perform an NMAP TCP port scan of the device. (SYN Stealth)
3) Perform an NMAP UDP port scan of the device.

Inside Servers
Lets first Ping the ips from the DMZ:
Syslog1 Server
ping –t 20 192.168.20.25
ping 192.168.20.25 (192.168.20.25) from 192.168.10.56 : 56(84) bytes of data.
From 192.168.20.25 icmp_seg=1 Time to live exceeded
From 192.168.20.25 icmp_seg=2 Time to live exceeded
From 192.168.20.25 icmp_seg=3 Time to live exceeded
From 192.168.20.25 icmp_seg=4 Time to live exceeded
MySQL SalesDB and SubfortuneDB
ping –t 20 10.254.20.10
ping 10.254.20.10 (10.254.20.10) from 192.168.10.56: 56(84) bytes of data.
From 10.254.20.10 icmp_seg=1 Time to live exceeded
From 10.254.20.10 icmp_seg=2 Time to live exceeded
From 10.254.20.10 icmp_seg=3 Time to live exceeded
From 10.254.20.10 icmp_seg=4 Time to live exceeded
ping –t 20 10.254.20.15
ping 10.254.20.15 (10.254.20.15) from 192.168.10.56: 56(84) bytes of data.
From 10.254.20.15 icmp_seg=1 Time to live exceeded
From 10.254.20.15 icmp_seg=2 Time to live exceeded
From 10.254.20.15 icmp_seg=3 Time to live exceeded
From 10.254.20.15 icmp_seg=4 Time to live exceeded
Internal Email Server
ping –t 20 10.254.10.20
ping 10.254.10.20 (10.254.10.20) from 192.168.10.56: 56(84) bytes of data.
From 10.254.10.20 icmp_seg=1 Time to live exceeded
From 10.254.10.20 icmp_seg=2 Time to live exceeded
From 10.254.10.20 icmp_seg=3 Time to live exceeded
From 10.254.10.20 icmp_seg=4 Time to live exceeded

Syslog1 Server (This scan yielded the same result from the HTTP ,HTTPS ,SMTP Relay , and SSH Servers)
UDP nmap –sU –p 1-65535 192.168.20.25
Starting nmap V. 3.00 (www.insecure.org/nmap/)
(The 65535 ports scanned but not shown below are in state: filtered)
Port State Service
515/UDP open syslog
Nmap run completed – 1 IP address (0 hosts up) scanned in 24 seconds.
TCP
nmap –sS –p 1-65535 192.168.20.25
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Interesting ports on (192.168.20.25):
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (1 host up) scanned in 789 seconds.
So our Syslog1 server is allowing the proper access.
MySQL Databases (From HTTPS Server)
SalesDB
UDP
nmap –sU –p 1-65535 10.254.20.10
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Interesting ports on (10.254.20.10):
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (1 host up) scanned in 589 seconds.
(The 65535 ports scanned but not shown below are in state: filtered)
TCP
nmap –sS –p 1-65535 10.254.20.10
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Port State Service
22/TCP open ssh
Nmap run completed – 1 IP address (0 hosts up) scanned in 45 seconds.
SubfortuneDB
UDP
nmap –sU –p 1-65535 10.254.20.15
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Interesting ports on (10.254.20.15):
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (1 host up) scanned in 589 seconds.
(The 65535 ports scanned but not shown below are in state: filtered)
TCP
nmap –sS –p 1-65535 10.254.20.15
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Port State Service
22/TCP open ssh
Nmap run completed – 1 IP address (0 hosts up) scanned in 45 seconds.
Our MySQL databases appear to be allowing the proper access.

Internal Email Server (From SMTP Relay Server)
UDP
nmap –sU –p 1-65535 10.254.10.20
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Interesting ports on (10.254.10.20):
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (1 host up) scanned in 645 seconds.
(The 65535 ports scanned but not shown below are in state: filtered)
TCP
nmap –sS –p 1-65535 10.254.10.20
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Port State Service
25/TCP open smtp
Nmap run completed – 1 IP address (0 hosts up) scanned in 56 seconds.
SMTP connectivity between the Relay and Internal Email server exist.

So all our access from the appropriate servers exists. We have also not found any open resources we should not have.

5) Inside => Outside (Internet)

This is an important one. Much too often, unlimited outbound is utilized. Access to everything rarely really means EVERYTHING.
Verify necessary connections from the Inside to the Internet:
Verify Internal Mail Server can connect to other servers:
telnet mail.myfriend.com
Connected to G.G.G.89
Escape character is '^]'.
220 **********
Excellent. Since we used the full domain name, this also shows we have DNS.
To verify, from another arbitrary workstation we issued:
telnet D.D.D.50 53
Trying D.D.D.50 …
Connected to D.D.D.50
Good, we have connected to our ISP's DNS Server. We are confident our Name Resolution is working correctly.
Verify access to HTTP and HTTPS servers:
telnet www.mygiacproject.com 80
connected to H.H.H.987
telnet www.mygiacproject.com 443
connected to H.H.H.983
Excellent. This once again also shows our DNS connectivity.
Verify IT workstations have connectivity to FTP, SSH, and Telnet servers:
telnet ftp.giacproject.com 23
Connected to G.G.G.987
telnet ssh.giacproject.com 22

Connected to G.G.G.987
telnet tnet.giacproject.com 21
Connected to G.G.G.987
Excellent, all connectivity exists.

Lets do some intrusive testing.

For each device we will perform the following:
1) Ping the device
2) Perform an NMAP TCP port scan of the device. (SYN Stealth)
3)Perform an NMAP UDP port scan of the device

Lets look at the inside interface on the PIX itself

Lets first Ping the inside interface on the pix:
ping –t 20 192.168.20.1
ping 192.168.20.1 (192.168.20.1) from 10.254.10.56 : 56(84) bytes of data.
From 192.168.20.1 icmp_seg=1 Time to live exceeded
From 192.168.20.1 icmp_seg=2 Time to live exceeded
From 192.168.20.1 icmp_seg=3 Time to live exceeded
From 192.168.20.1 icmp_seg=4 Time to live exceeded
Good deal. We are getting no reply. This is our icmp deny any inside command at work.

Now lets see if we can detect our host by scanning the outside interface of the PIX. This is a major advantage for a would be intruder, compromising a known host type could be trivial:
nmap –O 192.168.20.1

Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 scanned ports on (192.168.10.1) are :filtered
Too many fingerprints match this host for me to give an accurate OS guess
Again, no OS match.

Now let scan for open ports at that IP address:
UDP
nmap –sU –p 1-65535 192.168.20.1
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (0 hosts up) scanned in 30 seconds
TCP
nmap –sS –p 1-65535 192.168.20.1
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO

Nmap run completed – 1 IP address (0 hosts up) scanned in 30 seconds
Excellent.

Now something more interesting, we will scan the PIX's pat address. (This is the addresses all of our outgoing traffic is nat'ed to)
UDP
nmap –sU –p 1-65535 A.A.A.30
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Port State Service
53/UDP filtered dns
Nmap run completed – 1 IP address (0 hosts up) scanned in 65 seconds.
TCP
nmap –sS –p 1-65535 A.A.A.30
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Port State Service
80/TCP filtered http
443/TCP open https
53/TCP filtered dns
Nmap run completed – 1 IP address (0 hosts up) scanned in 57 seconds.
Just as we expected. Notice how some are labeled filtered. This is our PIX fixup protocols maintaining state with these protocols. Also notice the SMTP port is not showing up again. This is our PIX hiding it through Stateful inspection.

If we perform the same scan from our IT workstations:
UDP
nmap –sU –p 1-65535 A.A.A.30
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Port State Service
53/UDP filtered dns
Nmap run completed – 1 IP address (0 hosts up) scanned in 65 seconds.
TCP
nmap –sS –p 1-65535 A.A.A.30
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Port State Service
20/TCP open ftp-data
21/TCP filtered ftp
22/TCP open ssh
23/TCP open telnet
80/TCP filtered http
443/TCP open https
53/TCP filtered dns
Nmap run completed – 1 IP address (0 hosts up) scanned in 120 seconds.
The same as the previous scan, except now we have the added access we should have from the IT workstations.

One bit of information that became apparent when looking at this is how we are allowing unlimited outbound HTTP and HTTPS to anywhere. We have little logging ability to actually track the web behavior of our employees. Seems like a proxy server would be a perfect future consideration for the organization. Utilizing Squid, a combination of SARG (a perl based user logging systems for Squid) and Calamari (provides bandwidth based statistics for Squid) we will have a better idea of our users outbound access. Utilizing the caching functionality, we could probably also save some bandwidth.

6) Inside => DMZ

Verify necessary connections from the Inside to the DMZ:
Verify MySQL SubfortuneDB Server can SSH to Suppliers SSH server:
telnet 192.168.10.10 22
Trying 192.168.10.10 …
Connected to 192.168.10.10
SSH-1.99-OpenSSH_3.1p1
We have a connection.

Lets do some intrusive testing.
For each device we will perform the following:

1) Ping the device
2) Perform an NMAP TCP port scan of the device. (SYN Stealth)
3) Perform an NMAP UDP port scan of the device.

ping 192.168.10.10 (192.168.10.10) from 192.168.10.56 : 56(84) bytes of data.
From 192.168.10.10 icmp_seg=1 Time to live exceeded
From 192.168.10.10 icmp_seg=2 Time to live exceeded
From 192.168.10.10 icmp_seg=3 Time to live exceeded
From 192.168.10.10 icmp_seg=4 Time to live exceeded
So pings are not being allowed through.

nmap –O 192.168.10.1

Starting nmap V. 3.00 (www.insecure.org/nmap/)
Warning: OS detection will be MUCH less reliable because we did not find at least 1 open and 1 closed TCP port
All 1601 scanned ports on (192.168.10.1) are :filtered
Too many fingerprints match this host for me to give an accurate OS guess
So it appears our host detection failed.

Lets Scan the server from the SubfortuneDB Server:
SSH Server
UDP
nmap –sU –p 1-65535 192.168.10.10
Starting nmap V. 3.00 (www.insecure.org/nmap/)

Interesting ports on (192.168.10.10):
Note: Host seems down. If it is really up, but blocking our ping probes, try –PO
Nmap run completed – 1 IP address (1 host up) scanned in 589 seconds.
(The 65535 ports scanned but not shown below are in state: filtered)
TCP
nmap –sS –p 1-65535 192.168.10.10
Starting nmap V. 3.00 (www.insecure.org/nmap/)
Port State Service
22/TCP open ssh
Nmap run completed – 1 IP address (0 hosts up) scanned in 45 seconds.
From the database server we are provided the needed access, and nothing more.

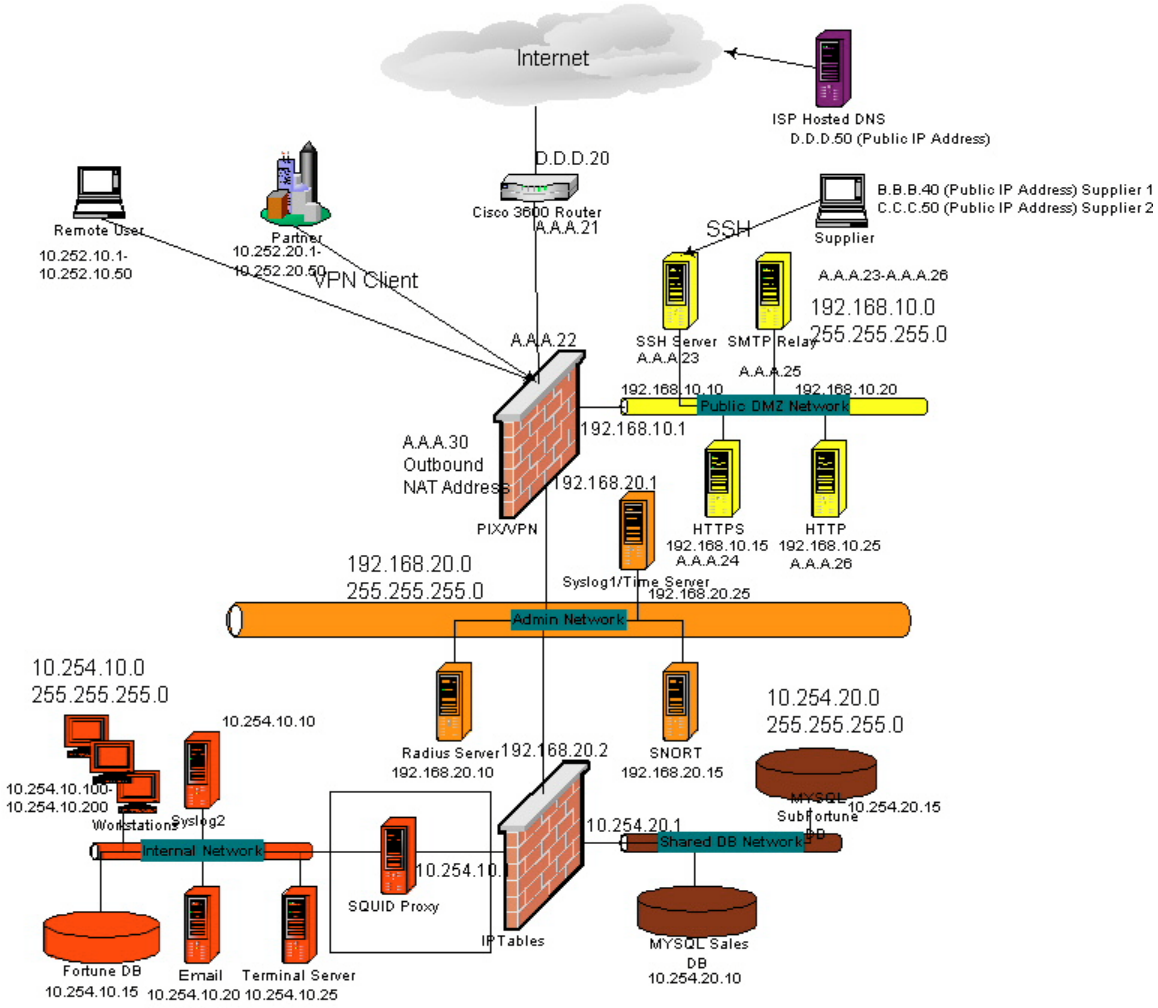
3.4 Evaluation:

Our audit gave us a clear understanding of how the traffic is flowing through our firewall. By no means completely inclusive, we were able to fully see our traffic flow and fix any issues. In a real life scenario, this audit should always be accompanied by a software audit also.

The audit not only helped us verify our security access levels, but also verify our traffic is flowing the way it should. This is an important aspect of what our audit provided, is allows for a clearer understanding of what is the normal operation of our systems. The clearer our understanding of this, the easier it is for us to spot suspicious activity. We originally actually changed our access requirements due to an early audit. This is one of the first steps I performed after the original draft of the Access Requirements. This enables a level of clarity not obvious prior to performing it. We realized we where not allowing outbound DNS for our SMTP Relay, an absolutely critical function. Utilizing the Postfix email server we perform DNS lookups based on Helo, Recipient, and Sender restrictions to help handle SPAM through unrecognized hosts (Open Relays). So the audit actually added functionality to the requirements.

One obvious blunder we found is that we are in no way logging Internet activity. A future consideration is to implement a proxy based system, namely Squid. This will allow us to track our users Internet activity as well as consume some bandwidth through the use of the caching system.

The following diagram shows the placement of such a device:



This will also give us another layer of filtering for our Internal Network.
Another is how we implemented the Supplier connectivity. The Suppliers upload Fortunes to our SSH server utilizing a 128 bit encrypted session. This voids our firewalls ability to Statefuly inspect the FTP data. This can be seen as a fair tradeoff for the ability of lessening the threat of sniffing these fortunes.

Another future consideration is to implement a security zone for our FortuneDB. Currently we are relying on host based security to maintain the integrity of the data.
This is a critical piece of equipment. The addition of a simple network card in our IPTables Firewall, a nominal cost, will allow us to segment this database from the rest of our network.

Another important tool in your auditing your security architecture is log review of every scan, ping, or any other tool utilized. This will give you a good sense of normal traffic flow and help you identify abnormalities once your structure goes live. Any time you can save when viewing live logs and instantly spotting suspicious activity may save your security integrity, and your job.

Audits should always be performed at regular intervals, especially in dynamic, changing environments. A Firewall is only as secure as it's implementation. While not guaranteeing our systems security, it is our last line of defense.

4. Design Under Fire

4.1 Introduction:

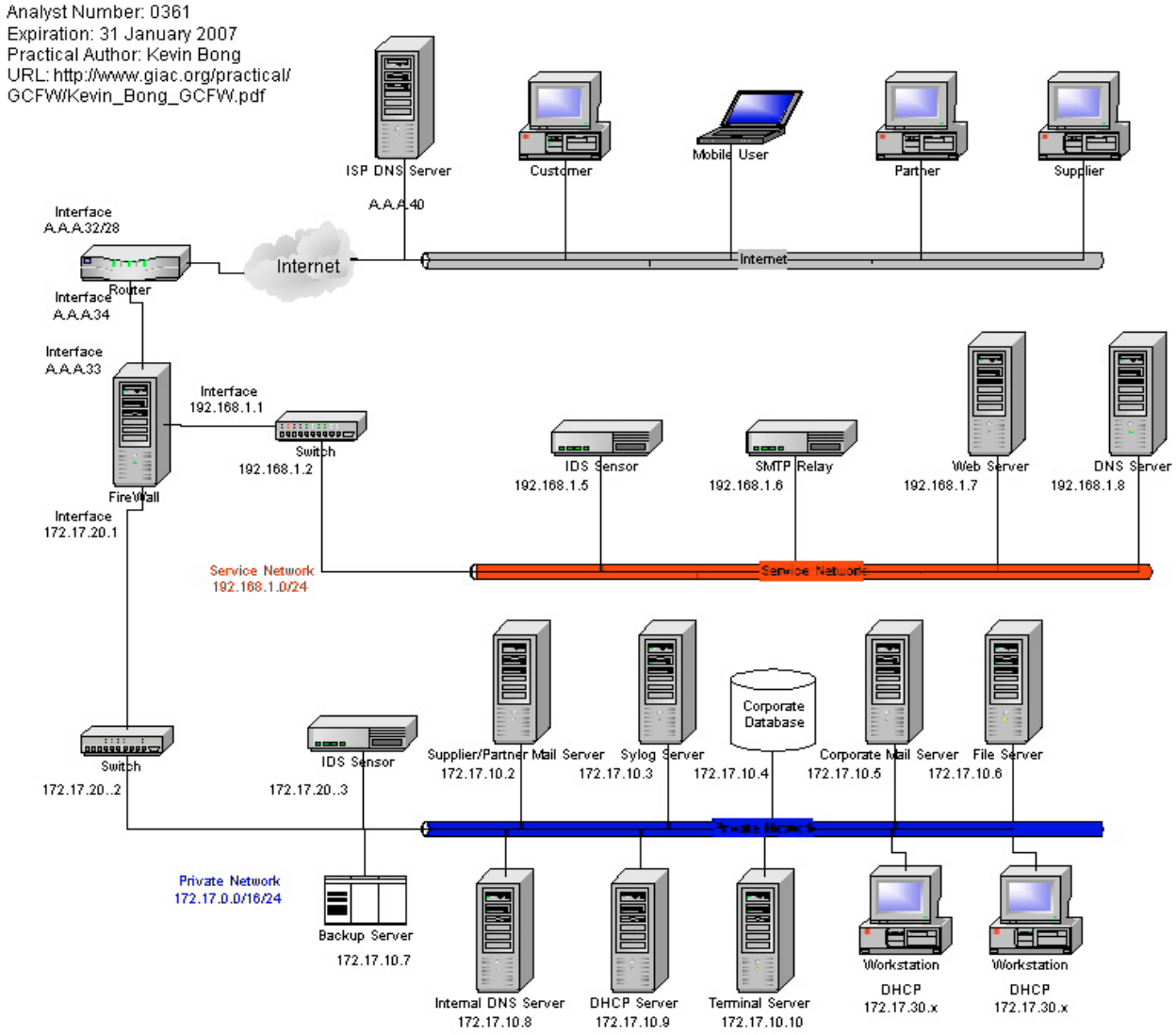
I chose the following Paper.

Analyst Number: 0361
Expiration: 31 January 2007
Practical Author: Kevin Bong
URL: http://www.giac.org/practical/GCFW/Kevin_Bong_GCFW.pdf

I have chosen this paper because at the heart of his security perimeter is a Symantec Enterprise Firewall version 7.0. This firewall, formerly known as the Raptor Firewall, has a pretty steady history of being one of the 'less' vulnerable systems. It has recently been EAL4 certified. The EAL4 approval process consists of the device going through a long and rigorous testing process and conforms to standards sanctioned by the International Standards Organization. I thought it was a perfect opportunity to prove there is no such thing as an invulnerable Firewall or security architecture.

Due to quality issues, I have completely reconstructed the Network Diagram. I have also sanitized Public IP addresses.

Kevin's Network Diagram is the following:



4.2 Attack Against the Firewall Itself:

Kevin's Firewall has the following properties:

OS: Windows 2000 Server SP3
Firewall Software: Symantec Enterprise Firewall Version 7.0
VPN Software: Symantec Enterprise VPN version 7.0
Other Software: Tripwire

IP addressing: DMZ Interface - A.A.A.33, Service Network Interface - 192.168.1.1, Private Network Interface - 172.17.20.1

Vulnerability Assessment:
I surfed over to Bugtraq (http://www.securityfocus.com/cgi-bin/sfonline/vulns.pl) and performed the following search on the Firewall software:
Vendor: Symantec
Title: Enterprise Firewall
Version: 7.0/NT 2000

And received the following results:

Published	Vulnerability	Bugtraq ID
12/13/02	RealAudio Buffer Overflow Vulnerability	6389
10/14/02	HTTP Proxy Denial of Service Vulnerability	5958
09/02/02	Weak TCP Initial Sequence Number Vulnerability	5387
04/16/02	Firewall FTP Bounce Vulnerability	4522
02/20/02	Notify Daemon SNMP Data Loss Vulnerability	4139

I then searched for vulnerabilities in the OS:
Vendor: Microsoft
Title: Windows 2000 Server
Version: SP3

And received the following results:

Published	Vulnerability	Bugtraq ID
03/09/03	Help File .CNT Link Buffer Overflow Vulnerability	7102
03/13/03	PostMessage API Unmasked Password Weakness	7092
01/22/03	Windows Locator Service Buffer Overflow Vulnerability	6666
02/17/03	Riched20.dll Attribute Overflow Vulnerability	6874

Wow, my search returned 14 vulnerabilities for year 2003 alone. I could have filled 3 pages with the results. I have included the 4 most recent.

Finally I searched for Tripwire, which is running on the server (a version was not cited in the paper so I assumed ANY):
Vendor: Tripwire
Title: Tripwire
Version: ANY

And received the following results:

Published	Vulnerability	Bugtraq ID
07/09/01	Insecure Temporary File Symbolic Link Vulnerability	3003
01/04/99	Long Filename Request DoS Vulnerability	1362

Two results were found. I believe we are quickly seeing a pattern developing here that even the most secure platforms are not invulnerable.

Choices, Choices…..For our attack, I wanted to look at a vulnerabilities in the Firewall Software itself. These systems are (usually) designed from the bottom up with security in mind, but usually due to the complexity and dynamic characteristics of software itself, vulnerabilities always show up.
For the Attack against the Firewall itself, we will be looking at:

09/02/02	Weak TCP Initial Sequence Number Vulnerability	5387

Analysis:
I will explain what the vulnerability is, and how one might exploit it.

The vulnerability is related to the Firewalls connection handling. The device generates ISNs (Initial Sequence Numbers) for the IP stack. It has been found that these numbers are generated in a predictable manner.

Predicting these sequence numbers could allow a 3rd part to intercept, or highjack the connection. Connection Highjacking is when an attacker seizes control of a legitimate user's application session, thus enabling the attacker the same access rights as the legitimate user. The legitimate user would then most likely lose connectivity.

The following is a test session performed by Secure Team showing the vulnerability:
http://www.securiteam.com/securitynews/5QP061F80E.html

```
Timeline    Connection                  ISN          Delta
-------------------------------------------------------------
10:33:05    x.x.x.x:1700 -> z.z.z.z:80  2088144436   -
10:33:06    x.x.x.x:1700 -> z.z.z.z:80  2088144436   0
10:33:07    x.x.x.x:1700 -> z.z.z.z:80  2088144436   0
...
10:35:30    x.x.x.x:1700 -> z.z.z.z:80  2088144436   0
10:35:31    x.x.x.x:1700 -> z.z.z.z:80  2088144436   0
10:35:32    x.x.x.x:1700 -> z.z.z.z:80  2088144436   0
...
10:50:43    x.x.x.x:1700 -> z.z.z.z:80  2088144436   0
10:50:44    x.x.x.x:1700 -> z.z.z.z:80  2088144436   0
10:50:45    x.x.x.x:1700 -> z.z.z.z:80  2088144436   0
```

It can easily be seen in this example, in a 20 minute duration, the ISN never changed.

In our scenario, someone could leverage a tool such as Shijack (http://www.securiteam.com/tools/5QP0P0K40M.html) or hunt (http://www.securiteam.com/tools/Hunt__a_new_Hijacking_software.html) to HighJack the connection. Once I have gained control of this connection, I would have the same right as the legitimate user would. This is a good way to get around our bread-and-butter access control. Our firewall now thinks we are a legitimate user, and provides us the same type of access. One obvious leverage here would be to hijack a telnet session and then have telnet access to the gateway router. We would then essentially be able to shut the entire operation down.

This type of attack is often leveraged against web servers, especially ones where transactions are taking place.

Since our network here employs SSL certificates, this type of man-in-the-middle attack is tough to pull off against our secure web server, but not impossible. The real weakness is in the fact that Microsoft Internet Explorer has many CA exploits, mainly it does not check some CA Basic Constraints. So we are not only hurt in a vulnerability by our Firewall, but also the client software utilized by the customer.

A simpler version, which could be pulled off our plain old web server would be:
- Attacker discovers dynamic web application
- Attacker brute forces through a list of sequential URL's.
- When a legitimate URL is found, he pastes it into Web Browser.
- He now has picked up that session.

Again these man-in-the-middle attacks are in no way trivial, but also not impossible.

Recommendations:
Symantec has released a HotFix for this issue
:http://www.symantec.com/techsupp/
This type of attack is really a vulnerability in the TCP/IP stack, our Firewall here just made this type of attack much easier.
Some common ways to reduce your risks are:

- Certificates = good
- Re-authenticate whenever a secure transaction is taking place. It is a small price to pay.
- Always and forever apply the most recent security patches.
- Always check and drop spoofing attempts whenever possible.
- Always remember, no matter how good your security software's track record, vulnerabilities will exist.

4.3 Denial of Service Attack:

For our DOS attack, we arbitrarily decided to pick on the web server. Our systems specifications look like:
OS: Windows 2000 Server SP3
HTTP Software: Microsoft IIS 5.5
Other Software: Tripwire
And our hardware is also relevant in this instance:
Processor: 1.4 GHz PIII
Memory: 512 MB Ram
Not the biggest box, so choking it should not be impossible.

Analysis:
Since we are really not just looking for an exploit here, just a brute force attack with 50 compromised cable modem users, we will utilize a popular tool called tfn2k.tgz, Tribal Flood Network 2000. This is a DDOS (Distributed Denial of Service) tool that can be downloaded at http://packetstormsecurity.nl/distributed.

The authors' description follows:
"Tribe Flood Network 2000. Using distributed client/server functionality, stealth and encryption techniques and a variety of functions, TFN can be used to control any number of remote machines to generate on-demand, anonymous Denial Of Service attacks and remote shell access. The new and improved features in this version include Remote one-way command execution for distributed execution control, Mix attack aimed at weak routers, Targa3 attack aimed at systems with IP stack vulnerabilities, Compatibility to many UNIX systems and Windows NT, spoofed source addresses, strong CAST encryption of all client/server traffic, one-way communication protocol, messaging via random IP protocol, decoy

packets, and extensive documentation. Currently no IDS software will recognize tfn2k."

An excellent technical analysis can be found at:http://packetstormsecurity.nl/distributed/TFN2k_Analysis-1.3.txt

A DOS (Denial of Service) attack is when a host uses some form of communications to yield a host unusable, or performing it's desired task it is normally expected to provide. This is usually accomplished by overwhelming the hosts resources, ie memory, bandwidth, etc. A DDOS (Distributed Denial of Service Attack) is when a number of compromised systems attack a single target or host, therefore causing a DOS on the host system.

Tribe Flood Network 2000 is a DDOS tool. It is composed of the following components:

Client – an application that can be used to initiate attacks by sending commands to other components (see below).
Daemon – a process running on an agent (see below), responsible for receiving and carrying out commands issued by a client.
Master – a host running a client
Agent – a host running a daemon
Target – the victim (a host or network) of a distributed attack

The program is available for both Unix as well as NT. A client is run on the master machine which talks to a daemon running on a client. It can perform a SYN attack, UDP Flood, ICMP Flood, or Smurf Attacks. You can also randomly alternate between the different types.

For our example we are going to look at the SYN attack.

To attack our IIS server we would issue:

tfn –f agents.list –i A.A.A.37 –c 5 –p 80
Syntax:
-f = File containing compromised cable modem users, agents.
-i = Ip address of target
-c = type of attack, in our case 5 = SYN Flood.
-p = Port to flood. This must be specified with the SYN Flood.

We could also attack the outside interface on the firewall itself. I chose the web server due to the fact it is a pretty low end box running Windows 2000 Server with IIS, which are not known for resource consumption.

The next step would be to sit back and see if the system becomes unresponsive. Since the Symantec Enterprise Firewall is configured to "proxy" all http traffic to the Web Server, only allowing valid http requests, this architecture may stall the

attack. The question then becomes if the amount of traffic we are generating will cause a DOS against the firewall itself. There is no sure fire way to stop these types of attacks, a lot of the dependencies are how large a compromised base they come at you with. There are although, a number of preventive tasks you could perform to make yourself a less likely candidate.
Recommendations:
Symantec recommends the following:
Create host network entities for Web servers and the external firewall address. The host entities should use the IP addresses of the requested addresses used in your Redirected Services for HTTP. If you are not using Redirects, then the host entities should be configured using the actual Web server IP addresses. Add these to a Group network entity.
You may also need to include virtual client IP addresses if internal hosts are going out with IP addresses other than external gateway address, as well as real IP addresses for hosts or subnets that are going out transparently, to this Group entity. Otherwise, SYN ACK will be denied for return HTTP packets on outbound requests from inside.
Create 3 individual filters:
a. Allow filter for Universe (entity A) to your Group network entity from step 1(entity B) - for service HTTP (A->B) and HTTP (B->A).
b. Deny filter for Universe (entity A) to Universe (entity B) - for service HTTP (A->B).
c. (6.5x/7.0) Allow filter for filter for Universe (entity A) to Universe (entity B) - for service (A->B) All.
d. (6.0x) Allow filter for filter for Universe (entity A) to Universe (entity B)TCP (A->B) , UDP (A->B), and any IP/ICMP protocols (A-B).

General recommendations to help prevent DOS attacks:
1) Use a firewall that exclusively employs application proxies when feasible.
2) Disallow unnecessary ICMP, TCP, and UDP traffic. Typically only ICMP type 3 (destination unreachable) packets should be allowed.
3) If ICMP cannot be blocked, disallow unsolicited (or all) ICMP_ECHOREPLY packets. Disallow UDP and TCP, except on a specific list of ports.
4) Spoofing can be limited by configuring the firewall to disallow any outgoing packet whose source address does not reside on the protected network.

4.4 Compromise an Internal System:

Again I have chosen the IIS Server. The Partner's of the architecture connect to the IIS server residing in the Service Network which will provide a connection to the internal Supplier/Partner Email server. The IIS server also has access rights to make internal ODBC calls to an internal DB server, also very tantalizing. We will look at one method for gaining the same access rights as a Partner:
Our systems specifications look like:
OS: Windows 2000 Server SP3

HTTP Software: Microsoft IIS 5.5
Other Software: Tripwire
And our hardware is also relevant in this instance:
Processor: 1.4 GHz PIII
Memory: 512 MB Ram

All partners retain a user account on this machine. We will first use an exploit to do a little snooping. IIS utilizes both Basic and NTLM authentication. We could use the following design error to gain a little more information:

Published	Vulnerability	Bugtraq ID
03/05/02	IIS Authentication Method Disclosure	4235

Analysis:
Using a simple Netcat 'HTTP GET' request, with an invalid username and password, we are greeted with an error message that tells quite the story:

GET / HTTP/1.1
Host: A.A.A.37
Authorization: Basic cTFraTk6ZDA5a2xt
If we get back 401 Access Denied, then we know Basic Auth is enabled.
If we get back 200 OK, then we know one of two things: The server does not support Basic Auth, or there is a system account on the server with uid:q1ki9 and password:d09klm

So lets assume we got back the 401 Access Denied. To be absolutely sure we are looking at NTLM Auth we can issue:
GET / HTTP/1.1
Host: iis-server
Authorization: Negotiate TlRMTVNTUAABAAAAB4IAoAAAAAAAAAAAAAAAAAAAAA=
If we get back 401 Access Denied, then the server undoubtedly supports NTLM auth. If a If we get back 200 OK, then the server does not support Integrated Windows authentication.

Some other interesting bits about this attack are:
If Basic Auth is supported, it is possible to gain the internal IP of the server. When a request is made, the clients host HTTP header is used as the realm. This tells the server when it should and shouldn't as for a reauthentication. If the host header field is left blank, the server will use it IP address as the realm. If NAT is employed, it will return something like 192.68.x.x to the client.
If NTLM is employed, it is possible to discover both the netbios name of the server as well as it's domain. It is returned as Base64 encoded text in response to a client Auth request.

Once we know this information, a brute force attack is trivial once we gain access to the SAMS DB.

One method of gaining this would be a slant on the old nimbda directory traversal vulnerability. By executing some commands against the IIS server, it is theoretically possible to get a copy of the SAMS db. This is based on how the permissions are set up on the server. Something like:
GET arg=http://Target IP/a.asp/..%c1%9c../..%c1%9c../winnt/repair/sam

The file /winnt/repair/sam is a backup copy of the Security Accounts Manager database, which contains usernames, account privileges, and security context information for every user allowed to log on to the Web server.

Another choice would be a bit of social engineering:
Our friend Harry is very unhappy at his job at GIAC. He is especially unhappy about his denial for the position of Systems Administrator after serving 10 years in the accounting department. He really likes this computer stuff. Harry works late usually every night, and often is called on to assist with any server issues long after the Senior Admin had gone home. One night the network line accidentally got unplugged from the we server.
Phone-Call:
Senior Admin: “Harry it appears the web server is not responding, could you go take a look at it for me, I am in the middle of dinner.”
Harry: “Well sure. Ok I am in front of it.”
Senior Admin: “What’s it doing? Are there lights on it? Is there a login screen on the monitor?”
Harry: “Yes, It is at a login screen.”
Senior Admin: “That is strange. Let’s try and login, I could always change the password tomorrow. UID: administrator Password:ineedanewjob”
Harry then plugs the network back in.
Harry “ OK, I am in.”
Senior Admin: “Looks like it popped back up, thanks Harry. Please logout for me. Must have been some sort of stalled process….”
Harry: “No problem, always here to help….”
Harry then pops his good old floppy in and proceeds to copy the SAMS database.

We would now utilize a tool such as, BeatLM or L0phtCrack, to perform a brute force password attack on the file.

This will give us all passwords on the web server, including our dear old partners.

Given the architecture of the system, this attack should work. It is a good reminder that application proxying does not render you invulnerable.

Recommendations:
1) When possible, disable Basic and NTLM authentication.
2) When authentication is needed:
    - Set account lockouts

- Rename common user account, ie administrator, whenever possible
- configure the adminscripts to use the hostname, not the IP address

Run the commands:
adsutil set w3svc/UseHostName True
net stop iisadmin /y
net start w3svc
This will cause the IIS server to use the machine's host name instead of its IP address.

3) Just because a vendor did not release a patch, does not mean it does not exist. Microsoft was made aware of this issue, but does not consider it a problem.

4) You can also configure the IIS server to not be able to read the SAMS security DB.
Verify \\HKEY_LOCAL_MACHINE\
HKLM\System\CurrentControlSet\Control\LSA\
RestrictAnonymous is set to 1.

5) Never trust Harry, or anyone else with your security information.

References

@Stake
http://www.atstake.com/research/tools/nc110.tgz - netcat

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

Cisco
http://www.cisco.com – Makers of the Cisco PIX 515 Firewall and 3600 series routers
http://www.cisco.com/en/US/support/index.html - Cisco TAC support
http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/firew_dg.htm - Cisco IOS Firewall Design Guide

Firewall Wizards
http://honor.trusecure.com/mailman/listinfo/firewall-wizards - Firewall Wizards mailing list

Flatline
http://members.1stnetusa.com/wizard/flatline/hack.htm#port – Various Security Tools

FreeRadius Server
http://www.freeRadius.org/ - Open Source Radius Server

Information Security Magazine
http://www.infosecuritymag.com/ - Security Publication

Insecure.org
http://www.insecure.org/nmap - Nmap Port Scanner

Internet Storm Center
http://isc.incidents.org/ - Monitoring global Internet traffic since November 2000.

Microsoft
http://www.microsoft.com – Makers of the Windows series of OS, IIS Webserver, and Exchange Email Server

MySQL
http://www.mysql.com – Open Source SQL Database

Netfilter
http://www.netfilter.org – IPTables Open Source Firewall

OpenRadius
http://www.xs4all.nl/~evbergen/e-advies.html/ - Open Radius Server

OWASP
http://www.owasp.org/asac/auth-session/hijack.shtml - Session HiJacking

OpenSSH
http://www.openssh.org **-** OpenSSH is a FREE version of the SSH protocol

PacketStorm
http://packetstormsecurity.nl/distributed/TFN2k_Analysis.htm - TFN2K – An Analysis

PostFix Mail Server
http://www.postfix.org – Postfix SMTP Relay

Redhat
http://www.redhat.com – Makers of Redhat 8.0 OS

Sans
http://www.sans.org **-** SysAdmin, Audit, Network, Security
http://www.sans.org/rr/threats/understanding_ddos.php - Understanding DDOS Attack, Tools and Free Anti-tools with Recommendation
http://www.sans.org/rr/malicious/trinity.php - "Trinity" Distributed Denial of Service Attack Tool
http://www.sans.org/rr/linux/ssl_enabled.php - Step-By-Step Guide to Configuring an SSL Enabled Web Server

SourceForge
http://sourceforge.net/projects/whisker/ - Whisker Vulnerability Scanner

SecuriTeam
http://www.securiteam.com/securitynews/5QP061F80E.html - Raptor Firewall Weak ISN Vulnerability

Security Focus
http://www.securityfocus.com/archive/1 - Bugtraq
http://www.securityfocus.com/infocus/1674 - IP Spoofing

Symantec
http://www.symantec.com – Maker of the Enterprise Firewall

Tripwire
http://www.tripwire.org/ - Tripwire Integrity Checker