Global Information Assurance Certification Paper

Securing a Windows Network with OpenBSD and third party software

GIAC Security Essentials Practical Assignment 1.4b Option 2

Mark Keating

February 27, 2004

Abstract:

Microsoft's (MS) Windows Operating System software has repeatedly proved to be less than optimal with regards to security, especially over the last two years. Due to continued vulnerabilities in MS software and the addition of network services, a network connected to the Internet was audited for an acceptable level of risk. The audit revealed several areas of unacceptable risk that needed immediate remediation and other areas that needed improvement over the longer term.

A plan was developed to reduce the risk of the network and to the data that resides on the systems. Over the period of several months during 2003 the network was migrated from a variety of Microsoft OSes and software to a mixture of OpenBSD, third party software and current versions of Windows. An examination of the network after the implementation of OpenBSD and other software proved to increase the overall security of the system in the perimeter, the services provided to internal and external clients, data integrity, system/data availability, and auditing. An informal security policy was put into place to ensure that the system did not lapse into an unsatisfactory state in the future.

Introduction:

Microsoft's (MS) Windows Operating System (OS) software has been criticized for being insecure (most intensely for the last 2 years). The opinions have ranged from off-hand remarks on Usenet to a high-profile white paper[1] published in September 2003 by six highly-respected members of the computer security field. One of the authors was dismissed by his company[2], @Stake, who counts Microsoft as one of its clients. Admittedly, the organization that published this paper is funded by a large number of Microsoft competitors, but the paper is not without merit (There was an interesting discussion of this paper and it's implications on the "firewall-wizards" mailing list[3] following the release of the paper). Microsoft has responded in a variety of ways, from launching the "Trustworthy Computing" program, to blaming system administrators for their lack of vigilance in applying patches, to admitting they had lost revenue due to security concerns of their customers[4].

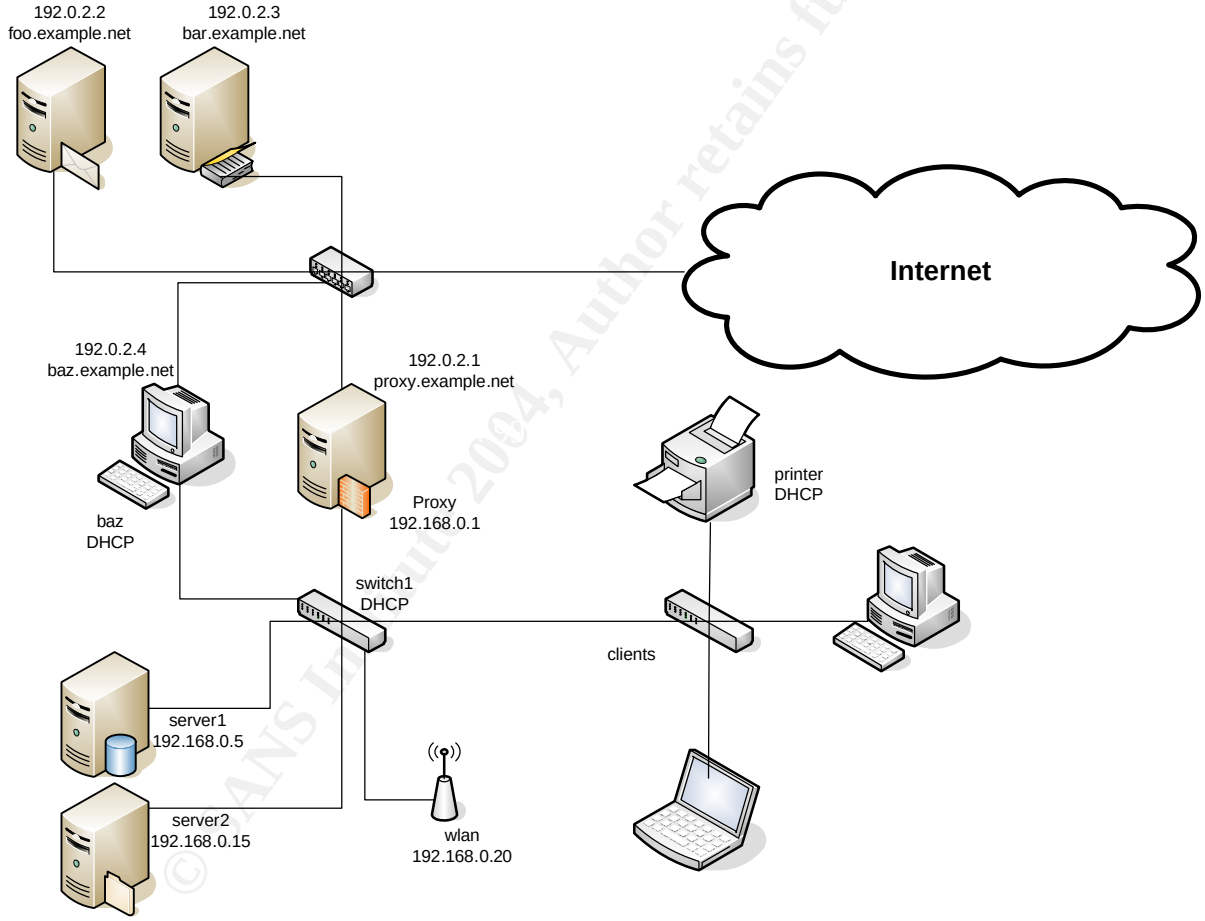
The decision on who gets what share of blame is left as an exercise for the reader. The result of the increasing number of patches for MS software, the necessity of loading anti-virus software, keeping virus signatures current and monitoring of network health has increased the burden on administrators who maintain Microsoft based systems. While a lot of these steps were prudent 10 years ago, when the Internet was a much friendlier place, it is now an obligation of the system/network administrator to maintain an ever-vigilant stance with regards to security if one is to be a responsible participant on the Internet. The nature of recent viruses/worms (Sapphire, MSBlast) wreaks havoc on the networks to which the infected hosts are attached. If a network is infected with Sapphire, it can consume 998MBits of a Gigabit Ethernet segment[5]. An anonymous company (100000+ employees) experienced 3 days of lost employee productivity due to the Sapphire worm. It is okay to leave the front door open on your house, but it is an entirely different matter to leave the front door open and let the 8 dogs with a history of biting people loose in the neighborhood.

A vendor neutral issue that affects almost every Internet connected network is SPAM[6]. According to one vendor of anti-SPAM software, sometime between August and September of 2003, SPAM became the majority of email traversing the Internet[7]. Spammers are now using trojaned (defined here as systems running programs unbeknownst to the end user) systems to send 1000s of messages a minute from unsuspecting home user PCs. There are also open proxies available on the Internet that can be utilized to anonymously send SPAM.

Throughout this paper the actual IP addresses have been replaced with addresses from the 192.0.2.0/24 block defined in RFC 3330[8] for use in documentation. Similarly, the domain names have been replaced with example.net, example.com or example.org as defined in RFC 2606[9] for use in documentation.

Previous environment:

The network described is a small network in the US. The network is connected to the Internet via a local Internet Service Provider (ISP) and xDSL. There are 5 static IP addresses assigned by the ISP. The ISP has also delegated via CNAMEs on their DNS server the reverse-mapping of the static IP addresses. The network provides services for a small business, a startup non-profit organization and several individuals. The network included 3 systems running either MS Internet Information Server (IIS) 4 or 5 (foo, bar and baz) at various patch levels. One system was providing DNS internally (server2) and there was an external DNS server (bar.example.net) for Internet clients. An off-the-shelf Windows based SMTP[10] package running on a Windows NT 4 server (foo.example.net) was providing mail for three of the domains hosted by the network. Below is a diagram of the network:



The state of the security on the network and its associated systems at the time of the initial investigation was poor. There was some thought given to security, indicated by the presence of an MS Proxy 2.0 firewall/application level gateway (proxy.example.net) and anti-virus software present on some of the systems. This was the extent of implementing a secure network. Some of the more obvious security issues:

- Perimeter security - There was a high level of risk of intrusion due to several non-secured entry points into the network. The least secure of these entry points was

the wireless network. There was a wireless access point (wlan) that had the default SNMP[11] community string for management/configuration. The access point was configured as an open system with the default SSID (Service Set Identifier) without the Wired Equivalent Privacy (WEP) protocol enabled. Even with WEP enabled there would be issues with the WEP key being cracked[12]. This was an open door to anyone in close proximity to the residence with a wireless network protocol analyzer. An attacker could easily pick an unused IP address on the network and utilize a free Internet connection and as a hidden system on the internal LAN. There would be no barriers to the attacker attempting to exploit a vulnerability on one of the internal machines to gain administrative access on the system. The state of the wireless LAN compromised the integrity of the data on all systems on the internal LAN, because there was the possibility that someone had gained entry to the network and tampered with systems and/or data on the systems. The dual homed system (baz.example.net) not running any packet filtering (firewall) or application level gateway (ALG) software was another weak point in the perimeter. If an attacker could exploit a vulnerability on the system, they could gain access to the internal network with ease.

- SPAM – The amount of SPAM coming through the mail server was estimated to be about 25% of mail traffic. The biggest issues here were lost productivity and viruses/Trojan programs. Some of the users were utilizing the filtering capabilities in their mail clients (MUAs) to minimize their exposure to SPAM. This was effective on about 75% of the SPAM, but the users still had to deal with false-positive issues. The issue of possible viruses/Trojan programs was addressed only on those clients running anti-virus software (~50%). The final issue with SPAM is the lost bandwidth on the Internet link and the lost system resources (disk space, CPU time, etc.) internally.
- Data availability – The data on the internal network was spread across several different machines with little thought into an organized structure or data availability. There was one system in the role of the main file server (server1), but there was a significant amount of data residing on the client machines and the dual-homed Windows 2000 system (baz). There was a tape drive attached to the main file server, but there was not a centralized backup strategy. One of the laptops had lost a hard drive in January 2003 and all the user data was lost.
- Auditing/logging – There was no way to reliably track what activity was taking place on the network or systems. The dual-homed Windows 2000 (baz) system running IIS 5 did have the IIS logs and there was evidence of attempted exploits of Code Red and other IIS vulnerabilities. Luckily, the system had the appropriate patches for IIS in place. The Windows auditing feature was not enabled on any of the systems. There was no centralized logging in place, so each system would need to be visited to review the logs. There may have been a penetration of the network at some point or there may not have been any compromises, but there was no way to state it definitively from the logs on any of the systems. There was no monitoring of any network traffic.
- Authentication – All of the systems except one had passwords on the Administrator account. There was no centralized administration of the accounts,

so there were duplicate accounts on several of the machines. Most of the passwords were simple and would have been discovered by a password auditing program running dictionary attacks. Most of the network equipment (switch, print server) had the default passwords.
- Physical security – Several of the systems did not have automatic logouts or screen locking enabled. Most of the systems and network equipment was located in a single room. One of the systems (server1) was on a UPS.

There were several factors that introduced the desire to improve the overall security of the system. The small business owner was sensitive to losing opportunities due to lost email or unplanned downtime. The founder of the non-profit organization had similar concerns for potential donors as well as the beneficiaries of the organization losing the ability to utilize the website or send the organization email. The burden in keeping the systems patched and up to date with anti-virus definitions was not looked upon fondly and was often neglected. The up-tick in media attention to computer security issues had raised the consciousness of the principals to take action. It was judged that the level of risk being assumed by the principals involved with the network was unacceptable and changes needed to be made to the system.

Securing the network:

The first action to take place was to meet with the stakeholders to establish a plan that was agreeable to everyone. The goal was to define a set of objectives relating to the desired operational level of the network by the end of 2003. The finalized objectives were:

- Take immediate action to correct the highest risk issues.
- Continue to utilize IIS and MS SQL to allow existing web based applications to continue to function.
- Continue to utilize Windows in the internal environment for file and print sharing.
- Provide Anti-Virus support for all systems.
- Provide a method to reduce the amount of SPAM coming into the domains hosted by the network.
- Minimize unscheduled downtime of the small business and non-profit organization websites and SMTP server.
- Provide a method of ensuring the confidentiality, integrity and availability of the data on the network.
- Deploy a secure perimeter for the network.
- Minimize disruption to users during changes to the network. Minimize changes in user environment after the new system was implemented.
- Centralize as many of the administrative functions as possible.
- Create a flexible security policy to maintain a secure system into the future.

After examining the requirements and examining the alternatives, it was decided to utilize OpenBSD on the perimeter and external systems. One of the deciding factors in this decision was reading Blair Heiserman's GSEC paper "Setting Up and Securing a

Small Network with OpenBSD"[13]. Protecting a Windows network with OpenBSD seemed like a logical extension of the original body of work. A factor that lent credence to the OS choice was the announcement from SANS that "OpenBSD is the winner of the 2003 Information Security Leadership Award for effective security testing of an operating system"[14]. Some of the stated goals of the OpenBSD development team include[15]:

- Pay attention to security problems and fix them before anyone else does. (Try to be the #1 most secure operating system)
- Greater integration of cryptographic software. This means IPsec, IPv6, key engines, Kerberos, free-AFS, and other forms of strong crypto or crypto-using systems. OpenBSD is developed and released from Canada and due to Canadian law it is legal to export crypto to the world. (As researched by a Canadian individual and as documented in the Export Control list of Canada). OpenBSD developers are doing active research and development on IPsec.

The motto of the development team is "Secure by default"[16].

OpenBSD can be loaded on an older system with common components and run perfectly well. It is a compact OS that will not consume a lot of disk space or CPU cycles during typical operation. OpenBSD runs on a number of different architectures and there is a list of supported hardware[17]. It does not support symmetric multi-processing (SMP), therefore single processor systems are ideal.

For the internal systems, it was decided to continue using Windows on both the desktop and server systems. This allowed the current administrators to continue to work on the platform they are most proficient with and would have little effect on users. The server systems would run Windows 2003 server and the clients would run Windows XP. The current Windows SMTP server would be retained until Microsoft Exchange is installed at a later date. Further changes to the Windows environment are discussed later in the paper.

Immediate action taken:

The existing MS Proxy server was upgraded to Windows 2000 Service Pack 4 with current hot-fixes. The dual-homed system (baz) was disconnected from the Internet and configured to use the proxy server for Internet connectivity. The FTP service being provided by the system was decommissioned and the web pages being hosted on the system were migrated to the external web server (bar.example.net). The Windows NT 4 system running the SMTP service (foo.example.net) was upgraded to Windows 2003 and the SMTP software was upgraded to the current revision. The external DNS and web server (bar.example.net) was updated to Service Pack 4 and current revisions of hot fixes. On all of the above systems:

- a commercial anti-virus package was installed and set to scan on a weekly basis and update definitions from the vendor's site once a day.
- All unknown user accounts were disabled

- All passwords that did not meet complexity requirements were changed.
- All un-needed services were disabled, NetBIOS and File & Print sharing were unbound from the NICs in the systems
- Logging and auditing were enabled on all systems and all services.

Strong passwords were assigned to all network switches, the print server and the wireless access point. Unused ports on the switches were disabled. MAC ACLs were placed on the wireless access point and SSID broadcasting was disabled.

It was felt that there was a sufficient baseline of security to proceed with replacing the external systems with OpenBSD based systems:

Install Secure OS on perimeter systems:

A general installation methodology was developed to load OpenBSD on all systems. The systems generally were installed with a 512MB partition for the root of the file system, a 512MB swap partition, a 1GB partition for the /tmp directory, an 8GB partition for the /usr and /var directories and the remaining space for /home. Larger partition sizes are used for /var because most of these systems will be logging to /var. The larger size of /usr is used to hold the source files for updating the system and the /ports tree for installing applications on the system. The ports system allows compilation and installation of software on systems in an elegant, reliable fashion[18]. The system can be installed from an OpenBSD CDROM or from a boot floppy created from an image that can be downloaded from ftp.openbsd.org. The installation document available online[19] outlines the installation process quite clearly. The installation process consists of partitioning the disk(s), creating the file system(s), network configuration and installing the OS from the desired media.

The system reboots after the install program completes. There are several configuration changes that were made to all systems (most changes are described in man afterboot(8)[20]):

1. Configure /etc/ssh/sshd_config.

Make the following changes:

```
# only allow version 2 of the protocol
Protocol 2
# no remote root login
PermitRootLogin no
# no empty passwords
PermitEmptyPasswords no
# authpf support
ClientAliveInterval 15
ClientAliveCountMax 3
# to allow tunneling of protocols in ssh
GatewayPorts yes
# point to a legal disclaimer
Banner /etc/ssh_banner
```

2. Configure inetd.conf.

Make the following changes:

```
# disable comsat
#127.0.0.1:comsat dgram  udp     wait    root    /usr/libexec/comsat     comsat
#[::1]:comsat    dgram   udp6    wait    root    /usr/libexec/comsat     comsat
```

3. Add user accounts for admins. (man adduser(8)[21] for details)

4. Upload public-key to admin home directories to allow public key authentication from Windows based ssh client

5. Configure sudo to allow appropriate permissions for admins. (man visudo(8)[22], man sudoers(5)[23] for details).

6. Change /etc/motd to contain appropriate legal disclaimer. Copy to /etc/ssh_banner

7. Put admins' email address in /root/.forward and ~/username/.forward:

```
#.forward to redirect mail
admin@example.net
admin2@example.net
user@example.net
```

8. Clear the console when a user logs out:

Make the following changes in /etc/gettytab:

```
P|Pc|Pc console:\
        :np:sp#9600:\
        :cl=\E[H\E[2J:
```

9. Update to the –stable[24] branch of OpenBSD. The basic process is outlined below. Greater detail is in the FAQ[25]. This ensures we are at the latest patch level for the OS.

```
# export CVSROOT=anoncvs@anoncvs.ca.openbsd.org:/cvs
# export CVS_RSH=/usr/bin/ssh
# cd /usr
# cvs -q get -r OPENBSD_3_3 -P src
# rm -rf /usr/obj/*
# cd /usr/src
# make obj
# cd /usr/src/etc && env DESTDIR=/ make distrib-dirs
# cd /usr/src/sys/arch/`machine`/conf
# config GENERIC
# cd ../compile/GENERIC
# make clean && make depend && make
# cp /bsd /bsd.old && cp bsd /bsd
# chown root:wheel /bsd (if you compiled as someone else)
(reboot)
# cd /usr/src
# make build
```

10. enable time synchronization via ntp:

- Change ‘rdate_flags=NO’ to ‘rdate_flags=192.0.2.2’ in /etc/rc.conf to allow time synchronization at system startup

- Add the following line to cron (crontab –e) to enable time synchronization with the system running ntpd. This was added to all systems running OpenBSD.

```
# daily time sync with ntp server
0       0       *       *       *       /usr/sbin/rdate -nca 192.0.2.2 | logger -t NTP
```

After a baseline install procedure had been instituted for the OS, the services and applications that were going to be run on each system were installed. Each of the particular services or applications has been described below:

DNS service:
DNS is hosted by the network for 4 different domains. It was decided to migrate DNS hosting from a Windows system to an OpenBSD system directly connected to the Internet. The system installation was installed using the procedure outlined above. OpenBSD 3.3 ships with BIND 9.2.2[26]. The OpenBSD development team audits the BIND code and makes changes that it feels BIND needs[27]:

> “Please also note that we have disabled a lot of those fancy features such as DNSSEC (since it is not ready yet), multi-threading (stability) and lwresd (noone uses it). On the other hand we've added stuff such as default chroot, default setuid and support for a device-less chroot.”

The Secure BIND template[28] was utilized to implement the DNS infrastructure. Several features of BIND were utilized to improve the security of the DNS infrastructure. Views were setup to allow increased functionality for clients in our network but to disallow those functions for foreign systems. To accomplish this, set up ACLs and views in the named.conf:

```
acl "examplenet" {
    // trusted systems
    192.0.2.u;
    192.0.2.v;
    192.0.2.w;
    192.0.2.x;
    192.0.2.y;
    192.0.2.z;

};

view "internal" {
    match-clients { examplenet; };
    recursion yes;
.
.
.

view "external" {
    match-clients { any; };
    recursion no;
```

.
.
.

We can then allow access to certain functions (IE: recursion) within BIND to the systems in the ACL that override the functions listed globally in named.conf.

As pointed out in the Secure BIND template:

```
The ordering of our views is very important. The named daemon accepts the first match.
Because our external view permits all clients, our internal clients also match this view.
For this reason we place our internal view first (permitting only our approved internal
hosts) and our external view second (permitting all comers).
```

Another feature utilized from the Secure BIND template was to create a 'bogon' ACL that prevented the DNS server from answering to queries from IP addresses that should not be routed on the Internet. These addresses include the RFC1918[29] addresses as well as IP address blocks that are reserved by IANA for future IP address allocations. Rob Thomas, the maintainer of the Secure BIND template also maintains a list of Bogon addresses that can be utilized to populate the ACL. This can be useful in preventing DOS attacks against a DNS server.

Some of the issues that commonly arise when running BIND from a chroot jail is that BIND cannot access files outside of what is defined as the top of the jail (/var/named, in this case). If access to other programs is needed, they must be copied into the jail. An excellent discussion of the implications of running BIND in a chroot jail can be found in Building and configuring BIND9 by Steve Friedl[30]. This document was of great assistance when configuring the correct permissions on all of the BIND files.

As the DNS system is handling the reverse DNS entries for the IP addresses assigned to the system, chapter 9 in DNS & BIND, 4th Edition[31] was reviewed to ensure the correct configuration for hosting reverse mappings for a subnet a non-octet boundary.

An example of the named.conf utilized by this network is included as Appendix A.

NTP:

Log analysis and system analysis in general become much more difficult (approaching impossible) when the timestamps on log entries or network traces are not accurate across all systems being reviewed. To enable reliable and consistent time on all systems, it was decided to utilize the ntp protocol. A good overview of the ntp protocol is available at http://www.ntp.org. A list of publicly available servers that provide ntp service is available[32]. If unsure whether to use stratum-1 or stratum-2 servers, users are advised to read the document in its entirety.

The ntp daemon is enabled by default on OpenBSD via rc.conf, if installed on the system. NTP was installed via the ports system:

```
$cd /usr/ports/net/ntp
```

```
$sudo make
$sudo make install
```

The following configuration file was used for ntpd:

```
$ cat etc/ntp.conf
# example.net ntp.conf
# created 20030802

server tick.uh.edu prefer
server now.okstate.edu
server navobs1.wustl.edu
server tick.usno.navy.mil
server tock.usno.navy.mil

restrict default ignore
restrict 129.7.1.66 noquery nomodify notrap nopeer
restrict 139.78.100.163 noquery nomodify notrap nopeer
restrict 128.252.19.1 noquery nomodify notrap nopeer
restrict 192.5.41.40 noquery nomodify notrap nopeer
restrict 192.5.41.41 noquery nomodify notrap nopeer
restrict 127.0.0.1 nomodify

restrict 192.0.2.3 mask 255.255.255.0 nomodify nopeer

driftfile /etc/ntp.drift
```

The 'server' lines indicate ntp servers that we are using to sync time. The restrict lines place certain restrictions on the ntp stratum-1 servers. The final restrict line limits queries to our subnet. The driftfile line lists a file used to track the time drift on the system.

Firewall:

The MS Proxy system was replaced with an OpenBSD 3.3 system running pf (Packet Filter), the OpenBSD TCP/IP firewall service. Pf provides TCP/IP packet filtering, network address translation (NAT), port redirection (reverse-NAT), packet normalization, load balancing incoming and outgoing connections and a sophisticated queuing system utilizing class based queuing (CBQ) or priority queuing (PRIQ) that support Random Early Detection (RED) and Explicit Congestion Notification (ECN)[33]

Pf is built into the OpenBSD distribution. To enable pf, change the 'pf=NO' to 'pf=YES' in /etc/rc.conf and remove the '#' from the '#net.inet.ip.forwarding=1' line in /etc/sysctl.conf and reboot the system.

The system running pf on this network has three interfaces: an Internet 10BaseT connection, an internal LAN connected at 100BaseTx and a PCI 802.11b wireless adapter. The wireless adapter functions as an access point for UNIX based wireless clients at the site. From the manpage for the wi driver[34]:

```
Cards based on the Intersil chipsets also have a host-based access point mode which allows
the card to act as an access point (base station).
```

The card is set to operate in access point mode at boot via the hostname.wi0 (man hostname.if[35]) configuration file:

```
inet 192.168.2.1 255.255.255.0 NONE
!wicontrol \$if -e 0 -t 3 -n unixwireless -s firewall -p 6 -a 2 -f 3 -P 0
```

The first line is passed to ifconfig and sets the IP address of the wi0 adapter to 192.168.2.1. The second line configures the wireless card to disable wep (-e 1), set the card to Auto Rate Select (High) (-t 3), set the network name to 'unixwireless' (-n unixwireless), set the station name to 'firewall' (-s firewall), set the port type to hostap mode (-p 6), set the access point density to medium (-a 2), set the frequency to 2422Mhz (-f 3) and disable power management (-P 0).

The configuration file for pf is /etc/pf.conf[36] and the configuration for the network is:

```
# Macros: define common values, so they can be referenced and changed easily.
ext_if="fxp0" # Internet connected interface
int_if="fxp1" # internal wired LAN
wi_if="wi0"   # wireless lan for OpenBSD systems
int_nets="{192.168.0.0/24, 192.168.1.0/24, 192.168.2.0/24}"
ext_addr="192.0.2.1"
wi_net="192.168.2.0/24"
log_hosts="{192.0.2.1/32, 192.0.2.2/32, 192.0.2.3/32, 192.0.2.4/32}"

table <examplenet> { 192.0.2.1, 192.0.2.2, 192.0.2.3, 192.0.2.4, \
       192.0.2.5 }

bogons = "{0.0.0.0/7, 2.0.0.0/8, 5.0.0.0/8, 7.0.0.0/8, 10.0.0.0/8, 23.0.0.0/8, \
              27.0.0.0/8, 31.0.0.0/8, 36.0.0.0/7, 39.0.0.0/8, 41.0.0.0/8, 42.0.0.0/8, \
              49.0.0.0/8, 50.0.0.0/8, 58.0.0.0/7, 70.0.0.0/7, 72.0.0.0/5, 83.0.0.0/8, \
              84.0.0.0/6, 88.0.0.0/5, 96.0.0.0/3, 169.254.0.0/16, 172.16.0.0/12, \
              173.0.0.0/8, 174.0.0.0/7, 176.0.0.0/5, 184.0.0.0/6, 189.0.0.0/8, \
              190.0.0.0/8, 192.0.2.0/24, 192.168.0.0/16, 197.0.0.0/8, 198.18.0.0/15, \
              223.0.0.0/8, 224.0.0.0/3, 255.255.255.255/32}"

# log dropped packets and scrub all packets
set loginterface $ext_if
scrub in all

# NAT internal traffic and handle FTP out
no nat on $ext_if from $wi_net to $int_net
no nat on $ext_if from $int_net to $wi_net
nat on $ext_if from any to any -> $ext_if
rdr on $int_if proto tcp from any to any port ftp -> 127.0.0.1 port 8021
rdr on $ext_if proto tcp from any to $ext_if port 80 -> 192.168.0.2 port 80

# support for authpf
nat-anchor authpf
rdr-anchor authpf
binat-anchor authpf


# Filtering: the implicit first two rules are
pass in log all
pass out log all

# Allow loopback and block bogons
pass quick on lo0 all
block drop in  quick on $ext_if from $bogons to any
block drop out quick on $ext_if from any to $bogons

# block all incoming packets but allow ssh, pass all outgoing tcp and udp
```

```
# connections and keep state, logging blocked packets.
block in log all
pass  in  on $ext_if proto tcp from any to $ext_if port 22 keep state
pass  in  on $ext_if proto tcp from any to 192.168.0.2 port 80 keep state
pass  in  on $ext_if proto tcp from $log_hosts to $ext_if port 5140 keep state
pass  out on $ext_if proto { tcp, udp, icmp } all keep state
pass  in  on $wi_if  proto { tcp, udp, icmp } all keep state
pass  in  on $int_if proto { tcp, udp, icmp } all keep state
pass  out on $wi_if  proto { tcp, udp, icmp } all keep state
pass  out on $int_if proto { tcp, udp, icmp } all keep state

# pass incoming ports for ftp-proxy
pass in on $ext_if inet proto tcp from any to $ext_if user proxy keep state

# allow pptp through
pass in on $int_if  proto gre keep state
pass in on $int_if proto tcp from any to any port 1723 keep state
pass in on $wi_if proto gre keep state
pass in on $wi_if proto tcp from any to any port 1723 keep state
pass out on $ext_if proto gre keep state
pass out on $ext_if proto tcp from any to any port 1723 keep state

# authpf support
anchor authpf
```

Due to the requirement of continuing to utilize the existing Microsoft IIS Active Server Pages (ASP) based applications, we had to perform port forwarding to allow outside systems to reach the internal IIS server. This was accomplished using port redirection. The line below redirects port 80 connection requests to the external interface (defined as the web server address in DNS) to the internal web server at 192.168.0.2:

```
rdr on $ext_if proto tcp from any to $ext_if port 80 -> 192.168.0.2 port
```

The following line is needed to allow the http traffic to pass through the external interface.

```
pass  in  on $ext_if proto tcp from any to 192.168.0.2 port 80 keep state
```

Some of the internal systems were using the Windows PPTP[37] to create a VPN connection to other networks. This required some changes to the pf system. The section above titled 'allow pptp through' was added to allow the PPTP traffic through the firewall. From the OpenBSD FAQ[38] "Because of a conflict between the In-Kernel gre(4) support and pptp, you will need to re-compile your kernel, removing support for gre(4)." This was accomplished by using the procedure for compiling a kernel outlined above. The only change to the above procedure was to edit the /usr/src/sys/arch/i386/conf/GENERIC file and put a '#' in front of the '`pseudo-device gre 1 # GRE encapsulation interface`' line. After rebooting the system, running the 'ifconfig –a' command showed no gre* devices present and the PPTP client functioned correctly.

httpd system:

Two of the websites hosted on these systems were serving static content. It was decided to move these systems to an external host. OpenBSD ships with the Apache web server, version 1.3.27. It is confined to a chroot jail[39]. Since the websites are

static, the pages were edited to reflect the file system layout of OpenBSD and copied to the /var/www/*sitename* directory. The different domain names were handled by using name based virtual host support in Apache[40]. The following was added to the httpd.conf file:

```
NameVirtualHost *

    <VirtualHost *>
    ServerName www.example.net
    DocumentRoot /www/example_net
    </VirtualHost>

    <VirtualHost *>
    ServerName www.example.org
    DocumentRoot /www/example_org
    </VirtualHost>
```

User accounts were added to the web server for the content authors/admins of the respective websites. The only difference in adding the users was that their home directories were set to be the root directory of the respective website located under the /var/www directory. This was needed to allow content to be added and removed from the respective websites by the authors while Apache was operating in the chroot jail.

smtp gateway system:

The method chosen to reduce the amount of SPAM being received in the user mailboxes was to setup a mail gateway/"smart host" that would identify SPAM before it reached the end user mailboxes. After researching several alternatives, the method described in Fairly-Secure Anti-SPAM Gateway Using OpenBSD, Postfix, Amavisd-new, SpamAssassin, Razor and DCC[41] was used. It seemed to fit with the overall philosophy of the OpenBSD community by running in a chroot environment. As the mail system on the internal network will be upgraded to MS Exchange in the future, this particular solution seemed ideal. As the amount of mail currently being processed by the Windows SMTP server was ~1000 messages a day, a desktop PC system was felt to be more than adequate to handle the mail system.

The installation method described above was used to install and configure the base OS. The general install method was followed and the DNS records were changed to set the new SMTP server as the primary mail system:

```
example.net.    IN    MX    10  baz.example.net.
example.net.    IN    MX    20  foo.example.net.
```

The preference value was set to 10 for the new Anti-SPAM gateway (baz) and the Windows SMTP server (foo) preference was set to 20. This will cause SMTP systems contacting the DNS server for the domain to be directed to baz.example.net as the mail host. If baz.example.net is unavailable, they will be directed to foo.example.net. After a couple of days the backup MX record (foo) was removed. An examination of the

headers of SPAM mails delivered to end-users showed that the gateway was being bypassed by SPAM senders in favor of the old mail server (foo).

The results of implementing the SMTP gateway were impressive. The amount of SPAM coming into the domains served by the SMTP system was about 25%. Users reported that they were receiving one or two SPAM messages a day versus the 20-40 that were normal before the implementation of the new gateway. Some fine tuning of the SpamAssassin configuration was needed to achieve the desired results. In the /etc/amavisd.conf file, the following values were changed:

```
$sa_tag_level_deflt  = 0;
```

This added the SpamAssassin score to every email that passed through the gateway and was useful in debugging.

```
$sa_tag2_level_deflt = 3.5;
```

This was changed from the default value of 5.0 (testing indicated that 5.0 was too conservative).

Securing the internal network:

Centralize administration:

Administration of the internal systems was distributed in the original environment. It was decided to centralize all the functions of the internal network. All of the data was removed from server2. Windows 2003 was installed on the server2. The system was promoted to a domain controller and DNS Start of Authority (SOA) for the example.corp namespace. It was decided to use a different name for the internal namespace to minimize the chance of internal names leaking to the Internet. The domain was changed to the Windows 2003 functional level. An Enterprise Certificate Authority was installed and configured on server2. DHCP was installed and started. Accounts were created for all of the users, service/application accounts, administrators and client computers. A login script was instituted to map the appropriate resources to each client and install anti-virus software if it is not present on the client machine. A new domain-wide group policy was created and implemented. Some of the group policy settings include:

- Disabling any un-needed services
- Set a standard policy for password settings
- Enable auditing on all systems
- Account lockout settings
- Appropriate user rights assignment
- Standard event log settings
- Enable system locking after 10 minutes of inactivity
- Redirect ‘My Documents’ folder to users home directory

After running a full virus scan against the entire system, two known good backups were taken of server1 using the attached tape drive and the Windows backup program. All of the existing partitions on the disks were removed. Windows 2003 was installed and the system was promoted to a domain controller. DNS was installed on the system. IIS6 was installed with only ASP enabled (no cgi execution or .Net framework installed). MS SQL 2000 SP3 was installed and configured to run under a domain account with rights limited to what is need for SQL to operate correctly. Third-party backup software was installed and configured with it's service account configured similarly to SQL. The file system on the server was configured with user home directories, directories for the developers of the different applications and administrative files. Appropriate NTFS permissions were applied to each directory. The data was then restored to the appropriate directories. The backup program was configured to perform weekly full backups with daily differential backups and to notify of events via SMTP.

The MS Baseline Security Analyzer was run against both systems and any suggested hot-fixes or system changes were made to the respective systems. A freeware utility GFI Languard System Integrity Monitor[42] that functions as file integrity utility similar to Tripwire was loaded on the servers and configured to forward reports via SMTP.

All of the critical systems are located in a secured room and connected to UPSes.

All of the clients that did not have Windows XP installed were upgraded, with the exception of two older systems that were replaced by the end of the year. All systems were patched to current hot-fix levels and anti-virus software was installed. All users were advised to keep their files in their home directory.

Wireless security:

As there are two different wireless networks in this configuration, there were two different solutions used for securing the Unix and Windows wireless networks respectively. The Unix wireless network can support Windows VPN clients as well as UNIX clients. The Windows implementation only supports Windows XP clients.

- authpf/isakmpd for UNIX:

The authorization of users with the authpf (man authpf[43]) and encryption of the transmissions using isakmpd (man isakmpd[44]) was utilized to provide secure authentication and encryption of the data for the UNIX wireless network. This was implemented using Securing 802.11 with OpenBSD[45] as a guide. The high-level overview of how this works is:

- create a Certificate Authority (CA) on the firewall with the wireless interface
- create and sign a certificate for the CA
- create and sign a certificate and create a key for the wireless gateway (firewall)

- create and sign x.509 certificates ,create a key and create a pkcs12 file for the users
- enable isakmpd in the /etc/rc.conf file
- create the /etc/isakmpd/isakmpd.conf file
- create the /etc/isakmpd/isakmpd.policy file
- create the authpf.conf, authpf.allow, authpf.rules, authpf.message and authpf.problem files in /etc/authpf/
- import the pkcs12 file in the appropriate VPN client software

The client opens an ssh session to the gateway and all traffic that is allowed by the firewall rules will traverse the firewall while the ssh session is active. Rules are dynamically added and removed from the pf.conf file when the ssh session is started and ended. Currently there is a only a single user of the Unix based wireless LAN.

- RADIUS/PEAP for Windows:

Securing the Windows based wireless LAN was achieved using the Step-by-Step Guide for Setting Up Secure Wireless Access in a Test Lab[46]. An overview of the setup:

- install an Enterprise root CA on server2.example.corp (this was done during the initial installation of the system).
- grant dial-up access to all the client computers and users
- create a domain group for wireless users and add the user and computer accounts
- install Internet Authentication Service(IAS) on server2.example.corp
- request a certificate for the IAS system
- setup the wireless access point as a RADIUS client on the IAS system
- create a wireless access policy on the IAS system
- configure the access point for RADIUS authentication
- configure the clients to connect to the wireless LAN

The authentication method chosen was PEAP-MS-CHAP v2, as this did not require client side certificates to authenticate. The system still uses WEP for encrypting the data, but the risk of compromising the system is minimized by dynamic updating of the WEP key to the clients by the IAS server.

Creation of the Windows Enterprise CA had the advantage of preparing for the installation of Exchange and Outlook Web Access (OWA) in the future.

Anti-virus:

The internal systems all had a commercial off-the-shelf anti-virus program installed. The systems are all managed via a central Windows system on the internal network. Virus definitions are updated every four hours. The definitions are pulled by the centralized anti-virus server from the vendor. These updated definitions are then

pushed out to all of the clients. The anti-virus client software is configured to perform a full system scan on the clients once a week. The client software is also configured to scan any file written to disk on all of the disks on the system and to scan all incoming and outgoing emails. Client policy is defined on the management console and pushed out to the client systems. All clients that log on to the Windows domain have anti-virus software installed via the login script if the anti-virus software is not detected.

Centralize logging:

Logging of activity on all systems was accomplished via a centralized syslog (man syslog) server. This centralized server was an internal Windows system that was running the Kiwi syslog daemon[47]. The decision was made to log to a Windows based system for a few reasons:

- the admins of the system were more comfortable with a Windows based system
- the Kiwi syslog daemon was able to log to an MS SQL server
- the administrator and SQL Database Administrator (DBA) would be able to generate pertinent views of the logs

The use the Kiwi syslog daemon was decided upon after reviewing the excellent SANS GSEC Case Study: Using Syslog in a Microsoft & Cisco Environment by Dan Rathbun[48]. The same client software described by Dan Rathburn[49] was used on the internal Windows systems. All of the network devices natively supported syslog and were configured to log to the Kiwi server. The major hurdle after Kiwi syslog was configured was to get the log information from the external hosts into the network with a minimal loss of security. This was solved by using stunnel[50] and syslog-ng[51], as described by Eric "Loki" Hines[52]. This article is targeted towards the Snort intrusion detection system, but the principle is similar. Syslog-ng has the advantage of using a reliable TCP connection, opposed to the UDP protocol used by the standard syslog that ships with OpenBSD. We are able to encrypt the log entries by using stunnel, which utilizes SSL to encrypt the data passing through the tunnel which the log entries are transmitted. The external systems are configured to log to both the local file system and to forward entries to the firewall. This is configured in the syslog-ng.conf file on the external systems:

```
source gateway {
        unix-dgram("/dev/log");
        internal();
};

destination localhost {
        file("/var/log/syslog-ng.all");
};

destination stunnel {
        tcp("localhost" port(5141));
};

log {
        source(gateway); destination(localhost);
```

```
        source(gateway); destination(stunnel);
};

options {
        keep_hostname(yes);
};
```

On the firewall, we have an instance of stunnel listening for log messages from the external systems. The firewall then stores a copy of the log messages locally via syslog-ng and also forwards a copy to the internal system running the Kiwi syslog service. The configuration used by syslog-ng on the firewall:

```
source shell {
        unix-dgram("/dev/log");
        internal();
        tcp(ip(localhost) port(514) max-connections(4));
};

destination localhost {
        file("/var/log/syslog-ng.all");
};

destination logsrvr {
        tcp("192.168.0.2" port(1468));
};

log {
        source(shell); destination(localhost);
        source(shell); destination(logsrvr);
};

options {
        keep_hostname(yes);
        use_dns(yes);
        chain_hostnames(no);
};
```

Syslog-ng and stunnel were built using the ports system described above. Syslog-ng built from the ports without any problems, but there was additional configuration required to get stunnel working as described in the Eric Hines paper. Using the standard 'sudo make && sudo make install' was used to get stunnel 4.04 installed. In order to get the 'make cert' option described in the paper to function correctly, the following procedure was used:

1. Download stunnel source (stunnel-4.04.tar.gz) from http://www.stunnel.org. The source is needed to have the 'make cert' option function correctly.
2. Extract the archive in /usr/ports/security/stunnel
3. Run the following command in /usr/ports/security/stunnel/stunnel-4.04: './configure --with-tcp-wrappers --with-pem-dir={$SYSCONFDIR}/ssl –with-random=/dev/arandom --with-ssl=/usr --localstatedir=/var'
4. Change to the /usr/ports/security/stunnel/stunnel-4.04/tools directory.
5. Edit the Makefile in that directory.
6. Change the 'openssl = $(ssldir)/bin/openssl' to 'openssl = $(ssldir)/sbin/openssl'
7. Change to the /usr/ports/security/stunnel/stunnel-4.04 directory.
8. Execute sudo make cert

Another change in stunnel was the use of a configuration file instead of the command line options used in the past. The configuration file used on the external systems:

```
client = yes

# Service-level configuration
[syslog-ng]
accept  = 5141
connect = 192.0.2.1:5140
```

The configuration used on the firewall:

```
cert = /etc/ssl/stunnel.pem

# Service-level configuration
[syslog-ng]
accept  = 5140
connect = 514
```

IDS:

One of the last additions to the system was a Network Intrusion Detection System (NIDS). One of the advantages of having a hub connecting all of the Internet facing systems to the WAN uplink was that all of the traffic destined for the Internet systems and the firewall would be visible to a system connected to the hub. The administrator that would be monitoring the NIDS system was not familiar with Unix based systems and preferred to have a Windows system utilized. The initial plan to run Snort[53] on OpenBSD was modified to utilize the EagleX IDS package[54] on a Windows XP system. EagleX uses Snort, Apache server, PHP[55], MySQL[56] and ACID[57]. Snort logs to the MySQL database, ACID runs under Apache and pulls data from the MySQL database and uses PHP to display a console application for Snort alerts. EagleX requires that WinPCAP[58] be installed on the system:

> WinPcap is an architecture for packet capture and network analysis for the Win32 platforms. It includes a kernel-level packet filter, a low-level dynamic link library (packet.dll), and a high-level and system-independent library (wpcap.dll, based on libpcap version 0.6.2).

Considering all the effort that had gone into putting a buffer between the Windows hosts and the Internet, it seemed rather illogical to connect a NIDS to the hub. The most acceptable compromise was to connect the XP system to the hub with the transmit pair on the Ethernet cable severed and no IP address bound to the Internet facing network adapter. The system was connected to the internal LAN with a static IP address to allow the administrator to view the ACID application on port 8877. The NIDS functioned quite well and some of the more interesting statistics gathered over three months from the system are:

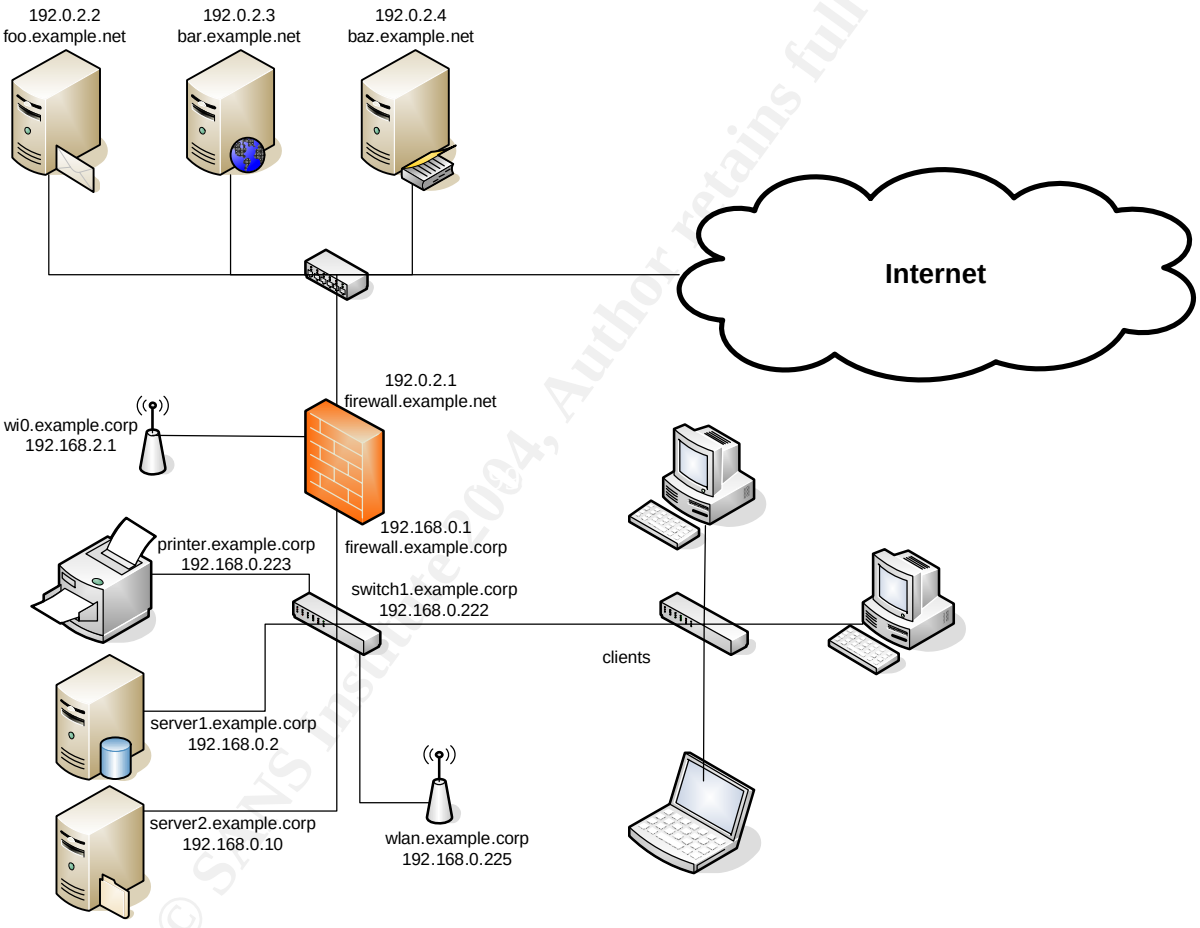
- Ping packets to public IP addresses from the Internet (Nachi/Welchia): 136574 packets from 2784 distinct hosts.
- Sapphire Worm propagation attempts: 7357 from 4058 distinct hosts

- Scans looking for a proxy on port 8080: 588 from 46 distinct hosts

These intrusion attempts are blocked by the firewall or not relevant to the Internet facing systems and they are not getting into the internal network. It is planned to implement a Snort system on the internal network in the near future to correlate external and internal network traffic. The analysis will then be utilized to fine tune the pf.conf file filter rules.

Current network configuration:

The current topology of the network and the services/functionality of each of the systems:



192.0.2.1 – firewall.example.net: OpenBSD 3.3 –stable, pf, authpf, CA, syslog-ng, stunnel server, isakpmd, ntp

192.0.2.2 – foo.example.net: Windows 2003, commercial SMTP software, GFI System Integrity monitor

192.0.2.3 – bar.example.net: OpenBSD 3.3 –stable, BIND, Apache, ntpd, syslog-ng, stunnel client

192.0.2.4 – baz.example.net: OpenBSD 3.3 –stable, Postfix, SpamAssassin, Amavisd-new, Razor, DCC, syslog-ng, stunnel client, ntp

192.168.0.2 – server1.example.corp: Windows 2003, file and print server, Windows DC, DNS, backup software, GFI System Integrity monitor

192.168.0.10 – server1.example.corp: Windows 2003, IAS server, CA, Windows DC, DNS, GFI System Integrity monitor

Policy and go forward issues:

After the changes had been made to the network, an informal policy was defined for the system. The policy was instituted to ensure that the system maintained an acceptable level of security and stability. The administrators of the network have several tasks that are performed on a regular basis:

- Run Windows Update against all Windows servers to maintain current patch levels on a weekly basis
- Review the logging information in the SQL database for any suspicious activity on a weekly basis. Continue to work with the SQL DBA to develop scripts to filter un-needed events from the log reports from the database.
- Review the anti-virus console to ensure the updates on the definitions are taking place and identify any problem areas with viruses on a weekly basis.
- Run a vulnerability scanner (ala nessus[59]) on a regular basis against the hosts on the Internet and the MS Baseline Security Analyzer against the internal hosts on a weekly basis.

The administrators subscribed to several public mailing lists that would notify them of system vulnerabilities or threats to their OpenBSD(security mailing list) or Windows (Microsoft Security Notification Service) systems[60, 61, 62]. They are receiving daily, weekly and monthly reports from each of the OpenBSD systems reporting on the health of the systems (man daily[63]). They are also receiving reports whenever critical system files are changed or permissions on those files change (man security)[64] on the OpenBSD systems.

On a monthly basis (or whenever there is a bug announced), update the OpenBSD system to the latest version of the –stable branch of the OS as detailed above.

Conclusion:

Completing all of the changes described above has resulted in a more secure system on many different levels. The prevailing attitude among many companies and individuals today seems to be "We have our firewall in place, so we are safe from the bad guys on the Internet." While there was an improvement in the security of the firewall at the Internet chokepoint, the security of the entire perimeter of the network was improved, including the wireless network and removing the dual-homed host. There was also improvement in the internal LAN via the deployment of Active directory to implement a centralized point of administration. The centralized administration allowed for proper ACLs to be placed on resources, which provides for a method of controlling access to network resources to only those who need access. There is now an audit trail if there is a question of unauthorized access or network abuse. The integrity of the data residing on the network is at a much higher level due to decreased risk of outsider access, a formalized backup policy and the institution of a network-wide anti-virus package. The users of the network have had minimal disruption during the changes and they can continue to utilize the applications and services with which they are most familiar. The reduction of unsolicited email entering the system has increased user productivity and reduced the risk of undesired attachments entering the system. The small business owner and the non-profit organization are much more confident that the services provided to them by this network are more robust and reliable. The administrators of the network are able to maintain the level of security and the overall health of the system with a reduced level of effort and a defined level of responsibility through the policy.

Given all the changes, there is still room for improvement. The deployment of a DMZ between the Internet gateway and the internal LAN was discussed during the planning phase and deferred until the deployment of MS Exchange sometime in 2004. It is planned to move a dedicated Windows IIS server to the DMZ and provide MS SQL connectivity to this dedicated system via an encrypted tunnel. It is also planned to provide some sort of content scanning of email and http traffic during the Exchange deployment, either through a dedicated scanning package or combining the scanning package with a proxy server.

The analysis of log files needs to be improved and this will most likely entail developing a custom solution that will consolidate the logging information from the SQL database and the various reports being generated by the OpenBSD systems, the anti-virus software, the Snort alerts and the RADIUS information being logged by the MS wireless network. There is also preliminary work being done with monitoring various system parameters with open-source systems management software.

References:

1 - Geer, Daniel, et al. "CyberInsecurity: The Cost of Monopoly". September 24, 2003. URL: http://www.ccianet.org/papers/cyberinsecurity.pdf (February 15, 2004)

2 - Borland, John. "Microsoft critic dismissed by @Stake". September 25, 2003. URL: http://news.com.com/2100-1009-5082649.html (February 15, 2004)

3 - Ranum, Marcus. "[fw-wiz] Personal Firewall Day?". October 5, 2003. URL: http://honor.trusecure.com/pipermail/firewall-wizards/2003-October/015447.html (February 15, 2004)

4 - Friend, Ina. "Security woes hit Microsoft balance sheet". October 23, 2003. URL: http://news.com.com/2100-7355-5096001.html (February 15, 2004)

5 - Kirkwood, Grant. "Re: Level3 routing issues?". January 25, 2003. URL: http://www.merit.edu/mail.archives/nanog/2003-01/msg00625.html (February 15, 2004)

6 - Spamhaus. "The Definition of Spam". URL: http://www.spamhaus.org/definition.html (February 15, 2004)

7 - Brightmail Inc. "Spam statistics". URL: http://www.brightmail.com/spamstats.html (February 15, 2004)

8 - Internet Assigned Numbers Authority (IANA), "Special Use IPv4 Addresses". September 2002. URL: ftp://ftp.rfc-editor.org/in-notes/rfc3330.txt (February 15, 2004)

9 - Eastlake, D., Panitz, A. "Reserved Top Level DNS Names". June 1999. URL: ftp://ftp.rfc-editor.org/in-notes/rfc2606.txt (February 15, 2004)

10 - Klensin, J. ed. "Simple Mail Transfer Protocol". April 2001. URL: ftp://ftp.rfc-editor.org/in-notes/rfc2821.txt (February 15, 2004)

11 - Case, J et al. "A Simple Network Management Protocol (SNMP)". May 1990. URL: ftp://ftp.rfc-editor.org/in-notes/rfc1157.txt (February 15, 2004)

12 - Stubblefield, Adam, John Ioannidis and Aviel D. Rubin. "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP". February 2002. URL: http://www.isoc.org/isoc/conferences/ndss/02/proceedings/papers/stubbl.pdf (February 15, 2004)

13 - Heiserman, Blair. “Setting Up and Securing a Small Network with OpenBSD”. January 27, 2003. URL: http://www.sans.org/rr/papers/index.php?id=935 (February 15, 2004)

14 - SANS. “Users Recognize Leadership in Operating System and Network Security”. July 22, 2003. URL: http://www.sans.org/press/ISLA.php (February 15, 2004)

15 - OpenBSD. “Project Goals”. 1.63. August 4, 2003. URL: http://www.openbsd.org/goals.html (February 15, 2004)

16 - OpenBSD. “Security”. 1.272. February 9,2004. URL: http://www.openbsd.org/security.html (February 15, 2004)

17 - OpenBSD. “Platforms”. 1.61. February 7, 2004. URL: http://www.openbsd.org/plat.html (February 15, 2004)

18 - OpenBSD. “The Ports & Packages collection”. 1.76. February 13, 2004. URL: http://www.openbsd.org/ports.html (February 15, 2004)

19 - OpenBSD. “4 - OpenBSD 3.4 Installation Guide” 1.163. February 10,2004. http://www.openbsd.org/faq/faq4.html (February 15, 2004)

20 - OpenBSD. “Manual Pages: afterboot.” URL: http://www.openbsd.org/cgi-bin/man.cgi?query=afterboot&sektion=8&apropos=0&manpath=OpenBSD+3.3&arch=i386&format=ascii (February 15, 2004)

21 - OpenBSD. “Manual Pages: adduser.” URL: http://www.openbsd.org/cgi-bin/man.cgi?query=adduser&sektion=8&apropos=0&manpath=OpenBSD+3.3&arch=i386&format=ascii (February 15, 2004)

22 - OpenBSD. “Manual Pages: visudo.” URL: http://www.openbsd.org/cgi-bin/man.cgi?query=visudo&sektion=8&apropos=0&manpath=OpenBSD+3.3&arch=i386&format=ascii (February 15, 2004)

23 - OpenBSD. “Manual Pages: sudoers.” URL: http://www.openbsd.org/cgi-bin/man.cgi?query=sudoers&apropos=0&sektion=5&manpath=OpenBSD+3.3&arch=i386&format=ascii (February 15, 2004)

24 - OpenBSD. “Patch Branches”. 1.10. June 18, 2002. URL: http://www.openbsd.org/stable.html (February 15, 2004)

25 - OpenBSD. “MINI-FAQ: Upgrading OpenBSD”. 1.188. February 2, 2004. URL: http://www.openbsd.org/faq/upgrade-minifaq.html (February 15, 2004)

26 - Internet Software Consortium(ISC). "ISC BIND". URL: http://www.isc.org/products/BIND/ (February 15, 2004)

27 - Shlyter, Jakob. "No Subject Given". January 21, 2003. URL: http://www.deadly.org/article.php3?sid=20030121022208 (February 15, 2004)

28 - Thomas, Rob. "Secure BIND Template Version 4.2 15 JAN 2004". 4.2 January 15, 2004. URL: http://www.cymru.com/Documents/secure-bind-template.html (February 15, 2004)

29 - Rehkter, Yakov, et al. "Address Allocation for Private Internets". February 1996. URL: ftp://ftp.rfc-editor.org/in-notes/rfc1918.txt (February 15, 2004)

30 - Friedl, Steve. "Building and configuring BIND 9". URL: http://www.unixwiz.net/techtips/bind9-chroot.html (February 15, 2004)

31 - Albitz, Paul and Cricket Liu. DNS and BIND, 4th Edition. Sebastopol, O'Reilly, 2001

32 - Mills, David. "Public NTP Time Servers". URL: http://www.eecis.udel.edu/~mills/ntp/servers.html (February 15, 2004)

33 - OpenBSD. "PF: The OpenBSD Packet Filter". 1.14. January 1, 2004. URL: http://www.openbsd.org/faq/pf/index.html (February 15, 2004)

34 - OpenBSD. "Manual Pages: wi." URL: http://www.openbsd.org/cgi-bin/man.cgi?query=wi&apropos=0&sektion=4&manpath=OpenBSD+3.3&arch=i386&format=ascii (February 15, 2004)

35 - OpenBSD. "Manual Pages: hostname.if." URL: http://www.openbsd.org/cgi-bin/man.cgi?query=hostname.if&apropos=0&sektion=5&manpath=OpenBSD+3.3&arch=i386&format=ascii (February 15, 2004)

36 - OpenBSD. "Manual Pages: pf.conf." URL: http://www.openbsd.org/cgi-bin/man.cgi?query=pf.conf&apropos=0&sektion=5&manpath=OpenBSD+3.3&arch=i386&format=ascii (February 15, 2004)

37 - Microsoft. "PPTP-based remote access VPN." Windows 2000 Server Documentation. February 28, 2000. URL: http://www.microsoft.com/windows2000/en/server/help/default.asp?url=/windows2000/en/server/help/sag_RASS_scen_pptp_rc.htm (February 15, 2004)

38 - OpenBSD. "6 – Networking". 1.184. February 6, 2004. URL: http://www.openbsd.org/faq/faq6.html#PPTP (February 15, 2004)

39 - OpenBSD. "10 – System management". 1.96. January 1, 2004. URL: http://www.openbsd.org/faq/faq10.html#httpdchroot (February 15, 2004)

40 – Apache. “Name-based Virtual Host Support”. 1.3. URL: http://httpd.apache.org/docs/vhosts/name-based.html (February 15, 2004)

41 - Vintinner, Scott. “Fairly-Secure Anti-SPAM Gateway Using OpenBSD, Postfix, Amavisd-new, SpamAssassin, Razor and DCC.” January 13, 2004. URL: http://www.flakshack.com/anti-spam/ (February 15, 2004)

42 – GFI. “GFI LANguard System Integrity Monitor”. URL: http://www.gfi.com/lansim/ (February 15, 2004)

43 – OpenBSD. “Manual Pages: authpf.” URL: http://www.openbsd.org/cgi-bin/man.cgi?query=authpf&apropos=0&sektion=8&manpath=OpenBSD+3.3&arch=i386&format=ascii (February 15, 2004)

44 - OpenBSD. “Manual Pages: isakmpd.” URL: http://www.openbsd.org/cgi-bin/man.cgi?query=isakmpd&apropos=0&sektion=8&manpath=OpenBSD+3.3&arch=i386&format=ascii (February 15, 2004)

45 - Turkulainen, Jarkko. “Securing 802.11 with OpenBSD.” January 5, 2003 URL: http://www.klake.org/~jt/tips/80211.html (February 15, 2004)

46 - Microsoft. “Step-by-Step Guide for Setting Up Secure Wireless Access in a Test Lab”. 1.0. November 20, 2003 URL: http://www.microsoft.com/downloads/details.aspx?familyid=0f7fa9a2-e113-415b-b2a9-b6a3d64c48f5&displaylang=en (February 15, 2004)

47 – Kiwi Enterprises. “Kiwi Syslog Daemon”. URL: http://www.kiwisyslog.com/info_syslog.htm (February 15, 2004)

48 - Rathburn, Dan. “Case Study: Using Syslog in a Microsoft & Cisco Environment” June 27, 2003 URL: http://www.sans.org/rr/papers/index.php?id=1100 (February 15, 2004)

49 – InterSect Alliance. “Snare Agent for Windows”. URL: http://www.intersectalliance.com/projects/SnareWindows/index.html (February 15, 2004)

50 – Hatch, Brian. “Stunnel -- Universal SSL Wrapper”. February 14, 2004 http://www.stunnel.org (February 15, 2004)

51 – syslog-ng. “syslog-ng”. URL: http://www.balabit.com/products/syslog_ng/ (February 15, 2004)

52 - Hines, Eric “Loki”. “Flying Pigs: Snorting Next Generation Secure Remote Log Servers over TCP.” July 5, 2002. http://www.securityfocus.com/guest/13283 (February 15, 2004)

53 – Snort. “Snort - The Open Source Network Intrusion Detection System“. February 10, 2004. URL: http://www.snort.org (February 15, 2004)

54 – Engage Security. **“**Eagle X- Preconfigured Intrusion detection system”. URL: http://www.engagesecurity.com/products/eaglex/ (February 15, 2004)

55 – php. “PHP:Hypertext preprocessor” February 12, 2004 URL: http://www.php.net (February 15, 2004)

56 – MySQL. “MySQL: The World's Most Popular Open Source Database” February 13, 2004 URL: http://www.mysql.com (February 15, 2004)

57 – Danyliw, Roman. “AIR-CERT: Analysis Console for Intrusion Databases (ACID)” URL: http://www.cert.org/kb/acid (February 15, 2004)

58 – WinPcap. “WinPcap: the Free Packet Capture Architecture for Windows”. February 4, 2004. URL: http://winpcap.polito.it/ (February 15, 2004)

59 – Nessus. “Nessus”. URL: http://www.nessus.org (February 15, 2004)

60 – Microsoft. “Microsoft Security Notification Service”. URL: http://register.microsoft.com/subscription/subscribeme.asp?ID=135 (February 15, 2004)

61 – Symantec. “Symantec Security alert Emails”. URL: http://nct.symantecstore.com/virusalert/ (February 15, 2004)

62 – OpenBSD. “Mailing Lists”. 1.75. February 7, 2004. URL: http://www.openbsd.org/mail.html (February 15, 2004)

63 - OpenBSD. “Manual Pages: daily”. 1.3 September 18, 2003 URL: http://www.openbsd.org/cgi-bin/man.cgi?query=daily&apropos=0&sektion=8&manpath=OpenBSD+3.3&arch=i386&format=ascii (February 15, 2004)

64 - OpenBSD. “Manual Pages: security”. 1.7 September 25, 2003 URL: http://www.openbsd.org/cgi-bin/man.cgi?query=security&apropos=0&sektion=8&manpath=OpenBSD+3.3&arch=i386&format=ascii (February 15, 2004)

Appendix A:

Named.conf:

```
options {
       version "";   // remove this to allow version queries
       directory    "/etc";
       pid-file "named.pid";
       statistics-file "named.stats";
       dump-file "named.dump";
       listen-on    { any; };
       listen-on-v6 { any; };
       allow-query  { any; };
       recursion     no;
};

acl "xfer" {
    // Allow transfers to our systems
    192.0.2.1;
    192.0.2.2;
};

acl "examplenet" {
    // trusted systems
    192.0.2.1;
    192.0.2.2;
    192.0.2.3;
    192.0.2.4;
    192.0.2.5;

};

acl "bogon" {
    // Filter out the bogon networks.  These are networks
    // listed by IANA as test, RFC1918, Multicast, experi-
    // mental, etc.  If you see DNS queries or updates with
    // a source address within these networks, this is likely
    // of malicious origin. CAUTION: If you are using RFC1918
    // netblocks on your network, remove those netblocks from
    // this list of blackhole ACLs!
    0.0.0.0/8;
    1.0.0.0/8;
    2.0.0.0/8;
    5.0.0.0/8;
    7.0.0.0/8;
    10.0.0.0/8;
    23.0.0.0/8;
    27.0.0.0/8;
    31.0.0.0/8;
    36.0.0.0/8;
    37.0.0.0/8;
    39.0.0.0/8;
    41.0.0.0/8;
    42.0.0.0/8;
    49.0.0.0/8;
    50.0.0.0/8;
    58.0.0.0/8;
    59.0.0.0/8;
    70.0.0.0/8;
```

```
71.0.0.0/8;
72.0.0.0/8;
73.0.0.0/8;
74.0.0.0/8;
75.0.0.0/8;
76.0.0.0/8;
77.0.0.0/8;
78.0.0.0/8;
79.0.0.0/8;
83.0.0.0/8;
84.0.0.0/8;
85.0.0.0/8;
86.0.0.0/8;
87.0.0.0/8;
88.0.0.0/8;
89.0.0.0/8;
90.0.0.0/8;
91.0.0.0/8;
92.0.0.0/8;
93.0.0.0/8;
94.0.0.0/8;
95.0.0.0/8;
96.0.0.0/8;
97.0.0.0/8;
98.0.0.0/8;
99.0.0.0/8;
100.0.0.0/8;
101.0.0.0/8;
102.0.0.0/8;
103.0.0.0/8;
104.0.0.0/8;
105.0.0.0/8;
106.0.0.0/8;
107.0.0.0/8;
108.0.0.0/8;
109.0.0.0/8;
110.0.0.0/8;
111.0.0.0/8;
112.0.0.0/8;
113.0.0.0/8;
114.0.0.0/8;
115.0.0.0/8;
116.0.0.0/8;
117.0.0.0/8;
118.0.0.0/8;
119.0.0.0/8;
120.0.0.0/8;
121.0.0.0/8;
122.0.0.0/8;
123.0.0.0/8;
124.0.0.0/8;
125.0.0.0/8;
126.0.0.0/8;
127.0.0.0/8;
169.254.0.0/16;
172.16.0.0/12;
173.0.0.0/8;
174.0.0.0/8;
175.0.0.0/8;
176.0.0.0/8;
177.0.0.0/8;
178.0.0.0/8;
179.0.0.0/8;
180.0.0.0/8;
181.0.0.0/8;
182.0.0.0/8;
183.0.0.0/8;
```

```
    184.0.0.0/8;
    185.0.0.0/8;
    186.0.0.0/8;
    187.0.0.0/8;
    189.0.0.0/8;
    190.0.0.0/8;
    192.0.2.0/24;
    192.168.0.0/16;
    197.0.0.0/8;
    223.0.0.0/8;
    224.0.0.0/3;
};


logging {

        channel "default_syslog" {
        // Send most of the named messages to syslog.
        syslog local2;
        severity debug;

        };

        channel audit_log {
                // Send the security related messages to a separate file.
                file "/named.log";
                severity debug;
                print-time yes;

        };

        category default { default_syslog; };
        category general { default_syslog; };
        category security { audit_log; default_syslog; };
        category config { default_syslog; };
        category resolver { audit_log; };
        category xfer-in { audit_log; };
        category xfer-out { audit_log; };
        category notify { audit_log; };
        category client { audit_log; };
        category network { audit_log; };
        category update { audit_log; };
        category queries { audit_log; };
        category lame-servers { audit_log; };

};


view "internal" {
    match-clients { examplenet; };
    recursion yes;

     // Standard zones
    //
    zone "." {
        type hint;
        file "/standard/root.hint";
    };

    zone "localhost" {
        type master;
        file "/standard/localhost";
       allow-transfer { localhost; };
    };

    zone "127.in-addr.arpa" {
        type master;
```

```
        file "/standard/loopback";
        allow-transfer { localhost; };
    };

    zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa" {
        type master;
        file "/standard/loopback6.arpa";
        allow-transfer { localhost; };
    };

    zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.int" {
        type master;
        file "/standard/loopback6.int";
        allow-transfer { localhost; };
    };

    // Master zones

    zone "example.net" {
        type master;
        file "/master/example.net";
        allow-transfer { xfer; };
    };

    zone "example.com" {
       type master;
       file "/master/example.com";
       allow-transfer { xfer; };
    };

    zone "example.org" {
       type master;
       file "/master/example.org";
       allow-transfer { xfer; };
    };

    zone "1.2.0.192.in-addr.arpa" {
       type master;
       file "/master/db.192.0.2.1";
       allow-transfer { xfer; };
    };

    zone "2.2.0.192.in-addr.arpa" {
       type master;
       file "/master/db.192.0.2.2";
       allow-transfer { xfer; };
    };

    zone "3.2.0.192.in-addr.arpa" {
       type master;
       file "/master/db.192.0.2.3";
       allow-transfer { xfer; };
    };

    zone "4.2.0.192.in-addr.arpa" {
       type master;
       file "/master/db.192.0.2.4";
       allow-transfer { xfer; };
    };

    zone "5.2.0.192.in-addr.arpa" {
       type master;
       file "/master/db.192.0.2.5";
       allow-transfer { xfer; };
    };

};
```

```
view "external" {
    match-clients { any; };
    recursion no;

     // Standard zones
    //
    zone "." {
        type hint;
        file "/standard/root.hint";
    };

    zone "localhost" {
        type master;
        file "/standard/localhost";
       allow-transfer { localhost; };
    };

    zone "127.in-addr.arpa" {
        type master;
        file "/standard/loopback";
        allow-transfer { localhost; };
    };

    zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.arpa" {
        type master;
        file "/standard/loopback6.arpa";
        allow-transfer { localhost; };
    };

    zone "0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.ip6.int" {
        type master;
        file "/standard/loopback6.int";
        allow-transfer { localhost; };
    };

    // Master zones

    zone "example.net" {
        type master;
        file "/master/example.net";
        allow-transfer { xfer; };
    };

    zone "example.com" {
       type master;
       file "/master/example.com";
       allow-transfer { xfer; };
    };

    zone "example.org" {
       type master;
       file "/master/example.org";
       allow-transfer { xfer; };
    };

    zone "1.2.0.192.in-addr.arpa" {
       type master;
       file "/master/db.192.0.2.1";
       allow-transfer { xfer; };
    };

    zone "2.2.0.192.in-addr.arpa" {
       type master;
       file "/master/db.192.0.2.2";
       allow-transfer { xfer; };
    };
```

```
    zone "3.2.0.192.in-addr.arpa" {
       type master;
       file "/master/db.192.0.2.3";
       allow-transfer { xfer; };
    };

    zone "4.2.0.192.in-addr.arpa" {
       type master;
       file "/master/db.192.0.2.4";
       allow-transfer { xfer; };
    };

    zone "5.2.0.192.in-addr.arpa" {
       type master;
       file "/master/db.192.0.2.5";
       allow-transfer { xfer; };
    };

};
```