Global Information Assurance Certification Paper

Deploying and Securing a FreeBSD-Based Squid Proxy Server

Step-by-Step

Neil Belden
GCUX, Version 2.0, Option 1 – Securing Unix Step by Step
Submitted December 29, 2003

TABLE OF CONTENTS

Neil Belden
GCUX, Version 2.0, Option 1 – Securing Unix Step by Step
Submitted December 29, 2003
Deploying and Securing a FreeBSD-Based Squid Proxy Server – Step-by-Step

Abstract/Summary:

This paper outlines steps to install and secure a FreeBSD-based Squid proxy server for implementation in a small business network. Among the topics addressed are FreeBSD 5.1 OS installation and configuration, Squid 2.5.STABLE4 proxy server installation and configuration, content filtering with Dansguardian 2.6.1-3, Squid proxy log analysis with Sarg 1.4.1 and network security architecture issues. System hardening will also be detailed step-by-step. The system will be protected locally by IPFW and file integrity will be monitored by Tripwire 1.3.1. The system will also be located in a DMZ to isolate it from the internal network and protect it from outside attack. Additional firewall rules will prevent internal users from accessing the Internet directly – forcing the use of the proxy.

Description of the System:

The primary system described herein is a FreeBSD-based server which functions as a proxy server and content filter for the corporate network of a small business (Great Lakes Consultants). Great Lakes desires to isolate its internal network from direct connections to the Internet. The implementation of a new "acceptable use" policy has triggered the need for a content filter to restrict access to certain Internet content and prevent the download of insecure files and executables. The proxy server (squid.great.lakes.local) will relay all http, https and browser-based ftp requests from internal hosts. Internal hosts will be restricted from making direct requests to the Internet – only the proxy will be permitted to make outbound http and https connections. A high-level diagram of the network appears in Figure 1.

Squid and Dansguardian both act as proxy servers in the final configuration of the server. Dansguardian listens for client requests on TCP port 8080. If the request is not banned based on IP, URL or other filter, Dansguardian hands the request to Squid via the loopback on port 3128. Squid gets the requested data (caches it for others and logs the event) and sends it back to Dansguardian for content inspection. If the content is acceptable, Dansguardian passes the data to the original requesting client. If the content is unacceptable, Dansguardian generates a web page on Apache and serves that to the client and also logs the event. This arrangement is depicted in Figure 2.

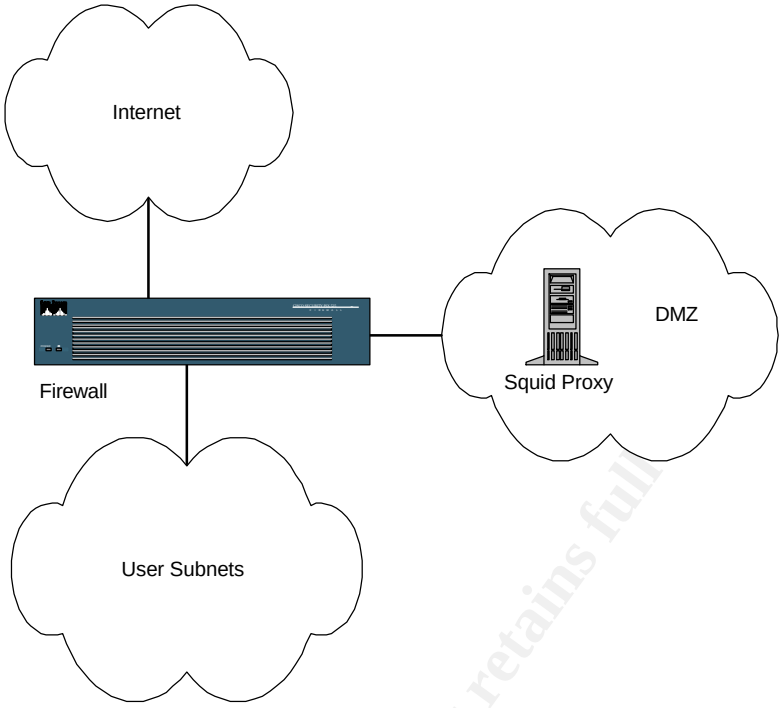

Figure 1

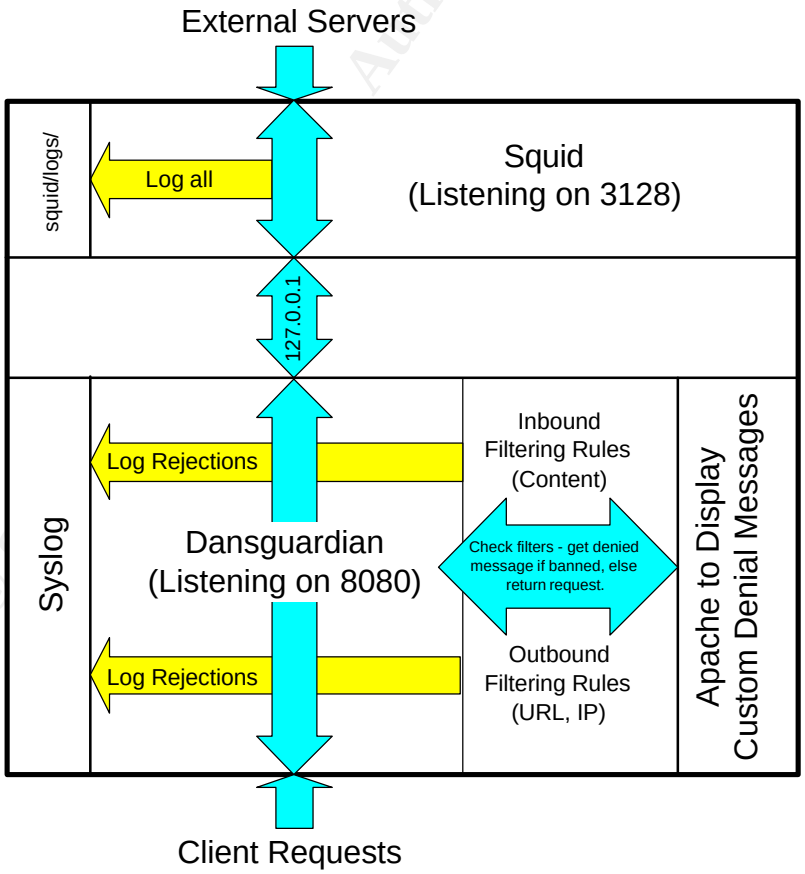

Figure 2

Risk Analysis of the System:

The proxy server will be located in the network's DMZ. Internal hosts will be required to utilize the proxy to contact external servers. No direct connections to the Internet will be allowed from the internal network.

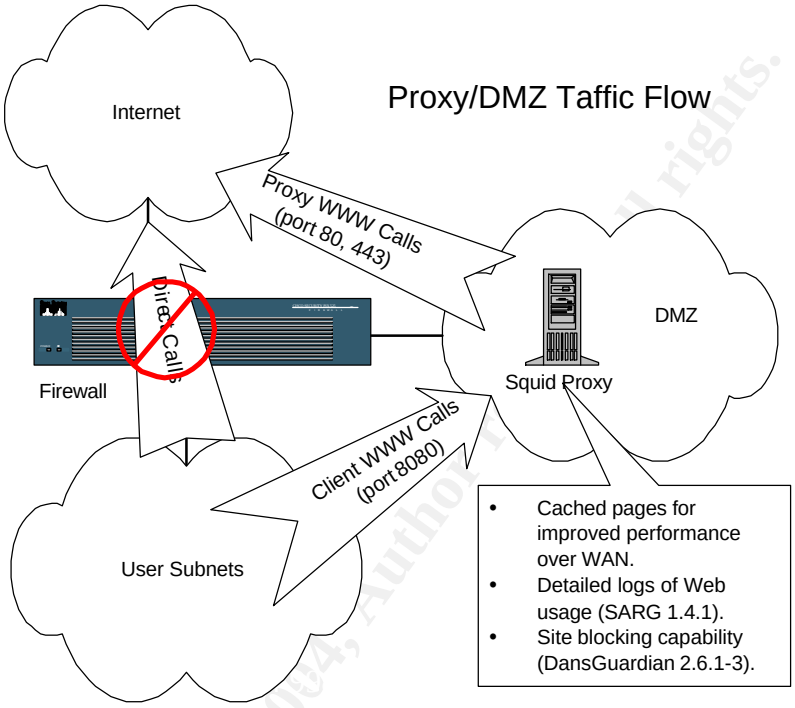

Figure 3

The proxy server will be permitted to communicate to the internal network over UDP 514 (syslog); TCP/UDP 123 (ntp) and TCP/UDP 161/162 (snmp/snmp trap).[1] It will be allowed to initiate DNS requests anywhere over UDP 53 (DNS). Additionally, certain internal IPs will be permitted to connect to the server on TCP port 22 for administration and port 80 to view the Sarg logs published in the /usr/local/www/data-dist/ directory.

Primary threats to this system include buffer overflow attacks, denial of service exploits[2] and, since Apache will be installed, attacks against the httpd service[3].

[1] Only syslog configuration is addressed in this paper. However, it will be highly desirable to synchronize this server with the organization's central NTP server so that all logs share a common time reference. Additionally, it will be advantageous to monitor the server's status via SNMP. Squid has its own SNMP features that must be enabled at compile time. See Pearson, Oskar. Squid: A Users' Guide. Sandton, South Africa: Qualica Technologies, 2003. URL http://squid-docs.sourceforge.net/latest/zip-files/ (29 DEC 2003).

[2] Some versions of Squid are vulnerable to denial of service and buffer overflow attacks. See http://www.sfu.ca/~siegert/linux-security/msg00094.html for examples of such exploits against versions

To mitigate these threats, only established traffic will be allowed to reach the proxy server from the Internet. This will be enforced by both the network firewall and IPFW rules on the host itself. Despite these protections, attacks against proxy servers are common[4] so the server will be located in a DMZ to isolate it from the internal network. The firewall rulesets will be tested to insure no unauthorized access is permitted. Ultimately, this proxy server will be a classic bastion host. It will be hardened, monitored with Tripwire and will have no special privileges on the internal network. No unnecessary services will be installed or running. Moreover, the proxy server stores no valuable data apart from its logs. The proxy's log files will be mirrored to a separate internal syslog server for long-term retention.

Secondary threats include physical attacks/disasters and hardware failures. The system will be protected in a climate-controlled, secure data center to deter most physical threats. Periodic full backups of the system will be stored off-site. Spare hardware will be available to rebuild the system from backup in the event of physical loss. Periodic restores from backup will test the validity of the backup process. Since this system was built from an inexpensive old workstation, a hot spare could easily be pre-configured and ready to cutover if needed.

Security Configuration Methodology:

We will assess and scan the system at four stages of the server build process. Each assessment is set forth in the sections "Vulnerability Assessment – Part I, II, III and IV", below. First, Netstat, Nmap and Nessus will be run against the base install of the OS before any additional applications are installed and before any services are disabled, patched or protected by a host-based firewall. This will provide a baseline for comparison as applications are added and services are brought up or down. Once the baseline is established, we'll install the necessary applications. The second vulnerability assessment will be conducted after Squid is installed to see if any new services appear unprotected or present opportunities for exploit. The third assessment will be conducted after installation of Dansguardian and Apache to see if those applications create vulnerabilities. Finally, the fourth assessment will be conducted after enabling IPFW. IPWF is enabled last to make sure no vulnerabilities are hidden by the firewall rules.

Depending on the results of each assessment, services will be patched, disabled, or protected by firewall rules as appropriate. The preference is to uninstall or disable all services that are unnecessary, patch and harden all services that are necessary and then restrict access to those services to only those hosts

---

prior to 2.4.STABLE4. A quick search of Bugzilla at http://www.squid-cache.org/bugs/index.cgi revealed 683 resolved bugs and 60 new reported bugs currently open for version 2.5.

[3] A search for "Apache" at www.cert.org reveals numerous vulnerabilities in the product's history.

[4] Hackers commonly scan for unprotected proxies. If they can be accessed, they provide a convenient means to conceal other activities or may provide access into the organization's internal network. See port 3128 activity history and user comments at http://isc.incidents.org/port_details.html?port=3128. (29 DEC 2003).

necessary through IPFW rules. The IPFW ruleset will have a permit by exception, deny by default posture.

Finally, Tripwire will be installed and a Tripwire database baseline will be created.

Step-by-Step Installation and Configuration:

Hardware Configuration:

The following hardware was chosen for this server:

Compaq DeskPro END 450+
256MB RAM
9.1 GB SCSI HD
P-III/450 MHz Processor
100 Mpbs NIC
APCI Motherboard

This old desktop system is perfect for the job as a proxy server. It has been outfitted with additional RAM at 256MB and a fast 9.1 GB SCSI HD for retrieving cache information and writing log data to disk. The P-III processor is not the fastest available but is more than sufficient for the job. When choosing hardware, performance will hinge on the following components, listed in order of importance:

- Disk random seek time
- Amount of system memory
- Sustained disk throughput
- CPU power[5]

FreeBSD 5.1 Installation[6]:

Installing FreeBSD is not always easy. For those new to FreeBSD, stick with the "standard" installation option. The "custom" installation is unintuitive and the menus and options seem to be circular.[7] Do yourself a favor and take the

[5] Pearson, Oskar. Squid: A Users' Guide. Sandton, South Africa: Qualica Technologies, 2003. URL http://squid-docs.sourceforge.net/latest/zip-files/ (29 DEC 2003). 2 – 5.

[6] For a general guide on installation and configuration of FreeBSD, see Stokely, Murray, and Clayton, Nik, eds. FreeBSD Handbook, Second Edition. Concord: FreeBSD Mall, 2002. Frequent updates available at http://www.freebsd.org/doc/handbook (29 DEC 2003) and in various downloadable formats at ftp://ftp.freebsd.org/pub/FreeBSD/doc/handbook/ (29 DEC 2003).

[7] Should you elect the custom path, be aware that the root user will have no password, no distributions will be installed and interfaces will have to be configured manually. Also, there is never a real obvious end to the custom installation process. The trick is to exit out of the process yourself once you have navigated all the menus, otherwise you'll find yourself reinstalling the same things and answering the same questions over and over again. Luckily, if you missed anything the first time, you can always go back into the setup program with the /usr/sbin/sysinstall utility or by booting with the CD again.

standard option as outlined below. This is not a default install of numerous unwanted packages as you might get in a standard installation with another OS. FreeBSD installs and enables only what you tell it to.

1. Boot the system with the first installation CD for FreeBSD 5.1[8]. When the "Welcome to FreeBSD!" message appears, selection option 2, "Boot FreeBSD with ACPI enabled" since this will allow proper detection of all devices on the APCI motherboard we're using.
2. At the sysinstall Main Menu, select "Standard" to begin a standard installation.
3. Select "OK" to proceed to the FDISK Partition Editor. Delete any existing partitions by highlighting them and typing "d" and the type "a" to use the entire disk for this installation. Next, highlight the new FreeBSD partition and type "s" to make it active. Type "q" to finish. Note that our SCSI disk was labeled da0.
4. At the next screen, select "Standard" which will install a standard MBR. We do not need a Boot Manager since this will not be a dual-boot system.
5. Select "OK" to proceed to the Disklabel Editor. Type "a" to set up the default sizes. The default sizes are nearly ideal for this server although our proxy applications (Dansguardian in particular) create extensive volumes of log data. It makes sense to shrink the size of /usr a GB or so and increase the size of /var by that same amount[9]. The new partition scheme is as follows:

Part	Mount	Size	Newfs	Part
da0s1a	/	256MB	UFS2	Y
da0s1b	swap	490MB	SWAP	
da0s1d	/var	1000MB	UFS2+S	Y
da0s1e	/tmp	256MB	UFS2+S	Y
da0s1f	/usr	6000MB	UFS2+S	Y

6. Type "q" when satisfied with the partitions. At the next menu, choose the distribution set. Select "Kern-Developer" because we don't want all the binaries but may need to recompile the kernel at some point. If you want (or can't live without) X on this system, select the X-Kern-Developer distribution instead. We will not install it because it comes with many

[8] CD ISOs can be downloaded from ftp://ftp.freebsd.org/pub/FreeBSD/releases/i386/ISO-IMAGES/ (29 DEC 2003). FreeBSD 5.X is officially a "CURRENT" release of the OS – not a "STABLE" release. For those uncomfortable with a leading edge OS, FreeBSD 4.8-RELEASE may be more appropriate. 5.X was chosen because it has acpi support. Additionally, 5.X was chosen for this project in the name of academic exploration and long-term usefulness. A 5.X STABLE release is on the near horizon.

[9] A test build of this server with all applications installed revealed that less than 2GB is needed in the /usr partition. For a good discussion on creating partition sizes tailored to a server's role, see Jang, Michael. RHCE Red Hat Certified Engineer Linux Study Guide (Exam RH302), Third Edition. Berkeley: McGraw-Hill, 2002. 160 – 164.

unwanted dependencies. The more moving parts, the less secure the system will be.

7. When prompted to install the FreeBSD ports collection, select “Yes”. Select the defaults for the XFree86 components if you elected to install X.

8. Exit the distributions menu. Select CD/DVD as the installation media.

9. The next screen gives one last chance to abort before formatting the disk. Select OK and the installation begins. Go to virtual terminal 2 [alt+F2] if you want to see details of the files being copied to the hard drive.

10. Once installation is complete, select “OK” for additional configuration options.

11. Select “Yes” to configure the NIC. Select the NIC (in this case fxp0), select no for IPv6 configuration (sorry), no for DHCP and then provide the network information requested for a static address. Our server will be called “squid” with a domain of “great.lakes.local”. Its gateway is 192.168.1.1 and the name server as assigned by the ISP. The IPv4 address is 192.168.1.9 with a mask of 255.255.255.0. Select “Yes” to bring up the interface now so long as you are on an isolated test network. Otherwise, wait until the device is hardened before enabling the interface.

12. Select No when asked to act as a gateway. Select No to inetd and all it’s services. Select No to anonymous FTP. Select No to NFS server and client. Select No to accept a “moderate” default security profile (we will fine tune IPFW later). Customize the system console settings if necessary (alternate keyboard, screen font, etc.). Set the local time zone as appropriate. Select No to enable Linux binary capability. Select “Yes” to confirm existence of the mouse.

13. If you elected to install X, select “No” to configure the X server at this time (it can be done later from a much better utility invoked within sysinstall). Select “No” when asked to browse the package collection (this too can be done later). Select “Yes” to add an additional user so that you do not have to log in as root. Next, set the password for root. Choose a password that meets the complexity requirements you will enforce on this system.

14. Select “No” when asked to visit the general configuration menu for a chance to set any last options. Then Exit the sysinstall menu. Allow the system to reboot and remove any installation media from the drives.[10]

[10] Optional X Configuration (if installed): Configure X by launching /usr/sbin/sysinstall. Select Configure; Xfree86; sf86config (shell-script based). Select Auto for the mouse protocol; “y” to emulate 3 buttons; and /dev/sysmouse as the device. Select the appropriate keyboard; country; and layout. Select “n” to additional XKB options. For the monitor, enter your specific horizontal

The first thing to change is the login banner or message of the day (motd) to replace the "Welcome to BSD" message that could be interpreted as an invitation to uninvited users. Edit /etc/motd in vi or another editor and change it to say something to the effect of:

THIS IS A PRIVATE COMPUTER SYSTEM AND UNAUTHORIZED USE OR ACCESS IS STRICTLY PROHIBITED. USE OF THIS COMPUTER SYSTEM, AUTHORIZED OR UNAUTHORIZED, CONSTITUTES CONSENT TO MONITORING. DURING MONITORING, INFORMATION MAY BE EXAMINED, RECORDED, COPIED, AND USED FOR AUTHORIZED PURPOSES. ALL INFORMATION, INCLUDING PERSONAL INFORMATION, PLACED ON OR SENT OVER THIS SYSTEM MAY BE MONITORED. UNAUTHORIZED USE OF THIS COMPUTER SYSTEM MAY SUBJECT YOU TO CRIMINAL PROSECUTION. EVIDENCE OF UNAUTHORIZED USE COLLECTED DURING MONITORING MAY BE USED FOR ADMINISTRATIVE, CRIMINAL, CIVIL OR OTHER ADVERSE ACTION. USE OF THIS SYSTEM CONSTITUTES CONSENT TO MONITORING FOR ALL LAWFUL PURPOSES.[11]

Vulnerability Assessment – Part I:

As set forth in the Security Configuration Methodology section above, we will assess the vulnerabilities and lockdown services as necessary at four intervals during the build process. This first assessment takes a look at the base install of the OS without any additional applications installed or IPFW rules in place.

Netstat: "netstat –an" reveals that TCP ports 587 (Submission), 25 (SMTP) and 22 (SSH) are listening and that UDP 514 (Syslog) is also up. Nmap confirmed these as the only open ports as well.[12] Next, after running nessus-update-plugins, a baseline Nessus scan is performed. Our results look good and confirm the OpenBSD reputation that it is locked down out of the box. However, Nessus did alert us to the following items that need to be addressed:

---

sync and vertical sync ranges if known (otherwise choose a conservative setting). Name it "monitor0". Select "y" to look at the card database and find the exact video card make and model. Name it "card0". Also select a range of video modes for each screen depth and select "n" to virtual screens for each. Finally, select the preferred depth and write the XF86Config file. Select one of the available Window Managers as preferred. Rerun sysconfig and the sf86config utility and fiddle with the options until successful.

[11] This banner is based on the standard DoD warning banners that emphasize a "consent to monitoring". See, i.e., Army Regulation AR 380-53 at http://www.usapa.army.mil/USAPA_PUB_search_p.asp. Other warnings may be applicable to your environment. Check with your legal department for guidance.

[12] It was interesting to see the console messages displayed by FreeBSD during the tcp scan. It repeatedly announced that it was limiting the closed port RST response to 200 packets/sec. During the UDP scan it similarly stated it was limiting the icmp unreachable response. A nice feature that demonstrates the resilience of the FreeBSD TCP/IP stack.

Nessus: Nessus reported the Sendmail version as 8.12.9 based on the banner. This version is subject to a CERT Advisory (CA-2003-25 Buffer Overflow). To eliminate this vulnerability, we'll update Sendmail following the instructions at ftp://ftp.freebsd.org/pub/FreeBSD/CERT/advisories/FreeBSD-SA-03:13.sendmail.asc. Also, Sendmail will be configured in a localhost-only listening mode by placing the following line in /usr/rc.conf:

sendmail_enable="NO"

When netstat is run again, port 25 is listening only on the loopback 127.0.0.1. Sendmail's related Submission service (port 587) also goes away[13].

Nessus also highlighted the fact that sshd is accepting connections with versions 1.33 and/or 1.5 of the protocol. While we could force sshd to only accept version 2 connections, this is not a high risk vulnerability in our environment since ssh traffic will not be allowed to traverse untrusted networks. To mitigate, IPFW will be configured to allow ssh connections from the internal trusted host IP space only.

Finally, Nessus determined that the server responds to ICMP timestamp requests. This vulnerability will also be mitigated by IPFW rulesets blocking all ICMP requests completely.

Before enabling IPFW, it makes sense to install Squid and the other applications first. That way we can elimate IPFW as the cause of errors when we're just trying to get the applications to work properly. Before installing any applications, update the ports directory as detailed in the next section. That way, near-current versions of the applications will be installed.[14]

Updating the system:

We will be using the FreeBSD ports to install Squid, Sarg, Dansguardian and any other applications we need. Before installing any applications, we'll want to update the ports on the local machine to make sure we get the latest or near-

[13] See man rc.sendmail. To completely disable all Sendmail daemons, place the following lines in /usr/rc.conf:

sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"

[14] While the FreeBSD ports system is a convenient way to install applications (and their dependencies), it does not make sense to install outdated or unpatched versions of software. If the ports directory does not have the latest version (or has a vulnerable version), don't use it. Alternatively, check other port repositories such as http://www.freshports.org. Otherwise, get the latest source code or binaries from the vendor site and compile/install it as you would on any *nix OS. See i.e., Shah, Steve. Linux Administration: A Beginner's Guide, Second Edition. Berkeley: Osborne/McGraw Hill, 2001.

latest ports. CVSup is the utility for updating the /usr/ports tree. CVSup itself can be installed as follows:

cd /usr/ports/net/cvsup-without-gui[15]
make install

This installs cvpasswd, cvsup and cvsupd. Before running CVSup, you must customize a "supfile" to tell CVSup what you want to update and where you want to get the updates. Since we want the latest ports, open /usr/share/examples/cvsup/ports-supfile in a text editor. Edit the following line with the path to a CVSup mirror site:

Change:
*default host=CHANGE_THIS.FreeBSD.org
to read:
*default host=cvsup9.us.FreeBSD.org[16]

From the command line, run "/usr/local/bin/cvsup –L 2 /usr/share/examples/cvsup/ports-supfile" (–L 2 tells it to be very verbose). CVSup updates all the ports in the /etc/ports directory including the indexes.

Now that we have the latest ports, some essential utilities will be installed. First, we'll install the open file lister lsof 4.69.1:

Cd /etc/ports/sysutils/lsof
Make install

Next, install tcplist-2.2.1, a TCP connection lister from the /etc/ports/sysutils/tcplist. Finally, install sudo-1.6.7.5 from /etc/ports/security/sudo. Tcplist provides a nice listing of open connections and sudo allows execution of root commands by non-root users. This is a good security practice and essential when doing remote administration over ssh since root logins are prohibited.

Installation of Squid (Proxy Server):

Squid is the proxy server that will relay all http and https client requests. The latest stable version, 2.5.STABLE4 will be installed.[17] Subsequently, all client browsers must be configured to use the proxy at 192.168.1.9 on port 8080. However, Squid will be configured to listen on its default port of 3128. This is because Dansguardian listens on port 8080 for the client requests and then passes them on to Squid on port 3128. To test Squid prior to installing

[15] A gui version of CVSup is in /usr/ports/net/cvsup.
[16] A list of CVSup sites is listed at the FreeBSD site at: http://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/cvsup.html (29 DEC 2003). Cvsup9.us.FreeBSD.org was chosen at random.
[17] Available at http://www.squid-cache.org/Versions/v2/2.5/ (29 DEC 2003).

DanGuardian, some clients will need to be set to proxy on port 3128 temporarily. Ultimately, 3128 and 80 will be blocked on the server and at the firewall so that no client can bypass the proxy server on 8080.

To install Squid, change to the /usr/ports/www/squid directory and enter "make install". Squid configuration can be fairly tricky. The Squid Configuration Manual found at http://squid.visolve.com/squid24s1/squid24s1.pdf (29 DEC 2003) provides a wealth of information on the variables in the squid.conf file but is not a good beginner's guide to get things started. The user's guide at http://squid-docs.sourceforge.net/latest/zip-files/ (29 DEC 2003) is an excellent reference as well and provides more of the how-to steps.[18] If you're anxious to just get the thing started, the best quick-start reference is Hack Proofing Linux[19]. Critical files for Squid are as follows:

Start/Stop script:	/usr/local/etc/rc.d/squid.sh
Configuration File:	/usr/local/etc/squid/squid.conf
Executable:	/usr/local/sbin/squid
Cache Directory:	/usr/local/squid/cache
Log Directory:	/usr/local/squid/logs

To get squid up and running, the following "tags" must be set in the squid.conf file:

http_port:	The port Squid will listen on (default is TCP 3128). You can also specify the interface if multiple exist.
cache_dir:	The location, size and structure of the cache directory.
acl:	Keyword for statements that define names for hosts for use in an http_access statement.
http_access:	Keyword for statements that allow or deny access to the hosts defined by an "acl" statement.

For the squid.great.lakes.local server, the default http_port of 3128 will be used. The cache_dir tag will be changed to "cache_dir ufs /usr/local/squid/cache 500 16 256" to increase the size of the cache directory from 100 to 500 MB. The following acl and http_access statements were added to the section of the configuration file below the statement "# INSERT YOUR OWN RULE(S) HERE":

```
acl dmz_hosts src 192.168.1.0/255.255.255.0
acl internal_hosts src 192.168.2.0/255.255.255.0
http_access allow dmz_hosts
http_access allow internal_hosts
http_access allow localhost
```

[18] Pearson, Oskar. Squid: A Users' Guide. Sandton, South Africa: Qualica Technologies, 2003. URL http://squid-docs.sourceforge.net/latest/zip-files/ (29 DEC 2003).

[19] Stanger, James, Lane, Patrick and Danielyan, Edgar. Hack Proofing Linux: A Guide to Open Source Security. Rockland: Syngress, 2001.

Localhost is defined by default in the squid.conf as 127.0.0.1. Localhost access to Squid is needed so that Dansguardian can connect over the loopback. In fact, once Dansguardian is running, the http_access statements created for "dmz_hosts" and "internal_hosts" will no longer be needed and should be removed. Those hosts will connect to the server over 8080 to Dansguardian and Dansguardian will connect to Squid for them. We need to rely on Dansguardian's IP filters and IPFW to limit access to the proxy at that point. This presents an interesting problem for the Squid logs as well. Once Dansguardian is installed, the logs will no longer show the IPs of the clients connecting to the proxy. Every connection will appear to come from the loopback. The Dansguardian logs are the only remaining means to track user activity and those logs will only show attempts to reach banned sites.[20]

You should also set the Squid process owner. At boot time, Squid starts as the root owner and then switches to the user specified in the "cache_effective_user" tag. The default user is "nobody" but that should be changed to a new user called "squid" with a primary group of "squid". The new group and user will need to be created and the squid user should be given a home directory of /usr/local/squid and a shell of /sbin/nologin. To set the process owner in the squid.conf file, uncomment the line "# cache_effective_user nobody" and change it to read "cache_effective_user squid". Add a new line below that to read: "cache_effective_group squid".[21]

Create the squid group with the following command: "pw groupadd squid". Next create the squid user with the adduser utility. Change ownership and permissions for the squid log and cache directories as follows:

chown squid /usr/local/squid/cache
chown squid /usr/local/squid/logs
chmod 770 /usr/local/squid/cache
chmod 770 /usr/local/squid/logs

Finally, create the cache subdirectories with the command "squid –z".[22]

---

[20] This is actually a fortuitous outcome. Indeed, we are not interested in tracking the activities of specific users to non-banned sites. Squid actually has a configuration setting to mask the source host to create this very anonymity for users (Tag "client_netmask" in squid.conf). See ViSolve.com. "Squid Configuration Manual." 15 MAY 2002. URL: http://squid.visolve.com/squid24s1/squid24s1.pdf (29 DEC 2003). 22. We only want to collect statistics regarding daily usage, most popular sites, cache performance and the like. These statistics may become useful in identifying sites to add to the banned lists but the ability to shadow every employee is not the goal. If employees were not allowed to use the Internet, we could block it at the firewall altogether. Since they are allowed, where they go is their business within the reasonable limits set by the acceptable use policy.

[21] Pearson, Oskar. Squid: A Users' Guide. Sandton, South Africa: Qualica Technologies, 2003. URL http://squid-docs.sourceforge.net/latest/zip-files/ (29 DEC 2003). 8; 22.

[22] The above configuration may seem unobvious but error messages generated by Squid would have led a sys admin down the same path. If the cache_effective_user had not been changed and no permissions to the cache and log directories had been modified, Squid generates the error: "Cannot open

Hopefully, the above is successful and Squid runs without error when started ("/usr/local/etc/rc.d/squid.sh start"). Verify that it comes up cleanly when the system boots. You can test Squid immediately from the command line with the squidclient utility (try "squidclient –v http://www.google.com). Next, run a few tests from hosts in the subnets defined by dmz_hosts or internal_hosts. Host Web browsers must be configured to use a proxy at 192.168.1.9 on port 3128. Take a look at the output from netstat –an when a client is connecting to an external Web site. You should see the client connection to the proxy on port 3128 and a corresponding proxy connection to the external server on port 80. Try secure sites and ftp sites and observe that https and ftp connections are also handled by Squid. Make sure that all activity is being properly logged in /usr/local/squid/logs/access.log.

Vulnerability Assessment – Part II:

As set forth in the Security Configuration Methodology section above, the vulnerabilities of the server will be assessed at four intervals during the build process. This second assessment takes a look at the base install with just Squid installed prior to installation of Dansguardian and Apache and before IPFW is enabled.

Netstat: As part of the continuing process to lock down the server, as expected, netstat shows the server is now listening on TCP 3128. Unexpectedly, UDP ports 3130 and 49155 are also now up and that demands investigation. To determine the process owner, run "lsof –i".[23] It returns the following information regarding the new ports:

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE/OFF	NODE	NAME
squid	461	squid	4u	IPv4	0xc27cc63c	0t0	UDP	*:49155
squid	461	squid	12u	IPv4	0xc27cc804	0t0	UDP	*:3130

These processes belong to Squid and their purpose might be tracked down in the squid.conf file. A "cat /usr/local/etc/squid/squid.conf | grep 3130" shows that 3130 is the default ICP (Internet Cache Protocol) port. Unfortunately, no such information is available on what UDP 49155 is doing. Since we are not communicating with any neighbor caches, ICP is not needed and can be disabled by setting the icp_port tag equal to "0" in squid.conf. Since we don't

---

/usr/local/squid/logs/access.log for writing – parent directory must be writeable by the user 'nobody' which is the cache_effective_user set in squid.conf." Because the user nobody may be shared by other services, it made sense to create a special user and group specifically for the Squid process. Also, had the cache subdirectories not been created, another error would have suggested running "squid –z" as root.

[23] For a good introduction to LSOF, see Nooning, Thomas. "Track Network Connections with LSOF on Linux." TechRepublic. 10 SEP 2002. URL http://techrepublic.com.com/5100-6261-1049412.html (29 DEC 2003).

know what port 49155 (both squid.conf and a search of the Internet turned up no clues) we will protect that port with IPFW.

Installation of SARG:

Sarg (the Squid Analysis Report Generator) is a simple tool that creates html pages to display data regarding Squid usage.[24] To install, perform the following steps:

```
cd /usr/ports/www/sarg
make install
```

Installation creates the following critical files:

Configuration File:	/usr/local/etc/sarg/sarg.conf
Executable:	/usr/local/bin/sarg

Several edits should be made to /usr/local/etc/sarg/sarg.conf. First, modify the access_log file path to "/usr/local/squid/logs/access.log" which is where Squid creates its logs by default. Second, modify the output_dir file path to "/usr/local/etc/sarg/logs" or some other location. The default location is a non-existent path.[25]

Once it's installed, merely run "sarg" and it will generate its html pages. Sarg can be run daily as a cron job. You'll want to purge the old logs from /usr/local/squid/logs/access.log after each sarg run as well.

Installation of Dansguardian (Web Content Filter):

Dansguardian is a web content filter that works along side Squid. It behaves as sort of a second proxy on the server as discussed in the section above "Description of the System" and depicted in Figure 2. Dansguardian rules are configured in multiple text lists that allow filtering based on URL, IP address, phrases, file extensions and other characteristics. Note that Dansguardian requires installation of Apache. Fortunately, the Apache service does not have to be visible to the outside world or even the clients. Dansguardian makes internal calls to Apache over the loopback and then serves the generated pages back to the clients over their original TCP 8080 connection. We will firewall port 80 to protect Apache from external exploits.

[24] Sarg is available at http://web.onda.com.br/orso/sarg.html (29 DEC 2003).

[25] If you elect to install Dansguardian and its dependency Apache, you can specify the log directory as "/usr/local/www/data-dist/sarg" (this will allow the reports to be viewed in a browser at http://localhost/squid/index.html). You must create the sarg sub-directory and should chmod it to 644 so only root can write to it and others can only read from it. To further restrict access, create a special group for users that are allowed to view these logs and restrict permissions to the log directory accordingly.

Source download for this application is restricted so you have to go to http://dansguardian.org to get it.[26] Download Dansguardian-2.6.1-3.source.tar.gz into /usr/ports/distfiles. Then cd to /usr/ports/dansguardian and "make install".

Installation creates the following critical files:

Start/Stop script:	/usr/local/etc/rc.d/start-dg.sh
Configuration Files:	/usr/local/etc/dansguardian/dansguardian.conf
Filter definition files:	/usr/local/etc/dansguardian/*list
Executable:	/usr/local/sbin/dansguardian
Log Directory:	/var/log/dansguardian.log
Template HTML file:	/usr/local/etc/dansguardian/template.html

Key Apache files are created as follows:

Start/Stop script:	/usr/local/etc/rc.d/apache.sh-dist
Executable:	/usr/local/sbin/httpd
Default directory:	/usr/local/www/data-dist

Dansguardian configures itself to automatically start on boot. It also launches Apache. Apache appears to be used solely for the purpose of generating the various "access denied" messages to users who request a banned site. Dansguardian expects that Squid will be listening on 3128 (not 8080). Dansguardian then listens on 8080 and hands permissible requests to Squid over the loopback (127.0.0.1) on port 3128. Clients should be configured to connect to the proxy on port 8080. Make sure /usr/local/etc/squid/squid.conf permits connections from the loopback address.

A quick test reveals that Dansguardian is working properly. A client request to www.whitehouse.gov returns successfully while www.whitehouse.com (a known adult site) is blocked:

[26] Dansguardian is free for non-commercial use. There is a nominal fee for commercial use ranging from $89 for small installations and $330 for enterprise installations. See http://dansguardian.org (29 DEC 2003) for details.

The format of the "access denied" message can be customized by editing the /usr/local/etc/dansguardian/template.html file. Numerous policy settings can be made by modifying the filter lists. Base configurations such as logging behavior and format are set in /usr/local/etc/dansguardian/dansguardian.conf. Actual content filter settings are configured in the various filter files and allow for filtering based on phrases, IP address, URL and file extension. Additionally, updated filter lists can be obtained from the Dansguardian site at http://blacklist.dansguardian.org/. There's even an automated script to get regular updates from the site.

The amount of filtering you want to do is limited only by your imagination. The real question becomes one of policy. Is it permissible for your users to check their personal Hotmail accounts on company time? Can they pay their bills online at www.paytrust.com? What if the sites they visit actually encourage them to stay in the office rather than leave to run errands offsite? Clearly, some institutional soul-searching is required when developing an acceptable use policy. Not many will argue that it is unreasonable to filter porn from the workplace and there is clear justification for filtering potentially dangerous content (.exe or .vbs files for example). However, it is easy to go overboard. If you really have a problem with people spending their day on eBay, you might need better managers – not stricter content filtering.

Since the Dansguardian logs can grow quite large, you should configure newsyslog to periodically archive the dansguardian.log. This requires adding an entry for /var/log/dansguardian.log to the newsyslog.conf file and specifying a frequency or log file size to trigger archiving. You should also specify the number of archive files to keep. The following entry is added for the Dansguardian logs:

Logfilename	[owner:group]	mode	count	size	when	flags
/var/log/dansguardian.log		600	7	10000	@T00	J

This will archive the log in a bzip2 compressed file (flag "J") every night at midnight (@T00) or when the current log reaches 10000 KB (9.76 MB). Seven previous copies will be saved with the oldest named dansguardian.log.6 and the latest named dansguardian.log.0. By setting the mode to 600, the files will be readable by root only. Newsyslog itself is configured to run every hour by default in /etc/crontab. See newsyslog(8) and cron(8) man pages for details.

We are also sending all logs to the central syslog server (superior.great.lakes.local) at 192.168.2.6. To configure this, add the following line to /etc/syslog.conf:

```
*.*                                     @superior.great.lakes.local
```

We'll need to remember to add a permit statement to the IPFW rules to allow UDP 514 traffic outbound to the Syslog server.

Vulnerability Assessment – Part III:

As set forth in the Security Configuration Methodology section above, the vulnerabilities of the server will be assessed at four intervals during the build process. This third assessment takes a look at the base install with all additional applications installed but prior to enabling IPFW.

Netstat: As expected, after installation of Dansguardian and Apache, netstat now shows additional listeners on TCP ports 80 (httpd) and 8080 (Dansguardian). We will block external requests to port 80 from all but a few administrator host addresses in the next section. These are hosts that we want to allow to view the sarg reports via http. However, before the firewall is enabled, another Nessus scan is run to check for any new vulnerabilities introduced by the installed applications.

Nessus: For this Nessus scan, enable all the plug-ins, including the "dangerous" ones to see whether any of the DOS and application killer attacks are successful.

Nessus did not take down the server or any of the services so we have an initial success. It did indicate that the server was susceptible to a "WWW infinite request" attack on 8080. It states, however, that it was unable to crash the server so this may be a false positive. To test, it seems like a good idea to run the attack again and keep and eye on the processor usage at the same time. Run "top" and then launch the attack again. The "Infinite HTTP Request" attack is listed under the DOS category. No change in processor utilization is observed at all. Running all the DOS attacks against the server again produces only brief fluctuations in the CPU usage with Dansguardian jumping to the top of the list on several occasions. Nessus reports no findings this time. Running all tests again results in no finding as well. Perhaps this can be written off as a false positive but we will make sure that port 8080 is blocked by IPFW just to be sure.

Additionally, Dansguardian sets a limit of 120 processes it can create to handle incoming connections. That number could be lowered in /usr/local/etc/dansguardian/dansguardian.conf by changing the “maxchildren =” value if denial of service attacks were actually able to down the server.

Nessus also claimed that Squid was accessible from anywhere although that appears to be a false positive as well. The acl statements only allow connections from our two trusted subnets and the loopback.

The final configuration of the server enables the following listeners:

Active Internet connections (including servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	(state)
tcp4	0	0	*.8080	*.*	LISTEN
tcp4	0	0	*.3128	*.*	LISTEN
tcp4	0	0	*.80	*.*	LISTEN
tcp4	0	0	127.0.0.1.25	*.*	LISTEN
tcp4	0	0	*.22	*.*	LISTEN
tcp6	0	0	*.22	*.*	LISTEN
udp4	0	0	*.49155	*.*	
udp4	0	0	*.514	*.*	
udp6	0	0	*.514	*.*	

As discussed, the only traffic we want to allow inbound is 8080 from the dmz (192.168.1.0/24) and the internal network (192.168.2.0/24) and port 80 and 22 from select administrator hosts in the internal network. Established traffic will be permitted as well. Outbound, we would want to allow UDP port 514 traffic to an internal syslog server. IPFW configuration is discussed in the next section.

Firewall Setup and Configuration – IPFW:

Although we installed the kernel with firewall support, it is not enabled by default. Attempting to show the current firewall rules produces the following message:

```
SQUID# ipfw show
Ipfw: getsockopt(IP_FW_GET): Protocol not available
```

What’s happening here is that IPFW is disabled by default in /etc/defaults/rc.conf. This file should not be edited (as the comments in the file itself will inform you). Instead, you should edit /etc/rc.conf to override the defaults in /etc/defaults/rc.conf: Insert the following statements:

```
firewall_enable=”YES”
firewall_script=”/etc/rc.firewall”
firewall_type=”client”
```

firewall_quiet="NO"[27]
firewall_logging="YES"[28]
log_in_vain="YES"[29]

Logging can be enabled by compiling the kernel with the option IPFIREWALL_VERBOSE or with the firewall_logging="YES" entry shown above in rc.conf.[30]

As stated in /etc/rc.firewall, options for firewall_type are:

open	- will allow anyone in
client	- will try to protect just this machine
simple	- will try to protect a whole network
closed	- totally disables IP services except via lo0 interface
UNKNOWN	- disables the loading of firewall rules.
filename	- will load the rules in the given filename (full path required)

For our chosen firewall_type of "client", entries for net, mask and ip in /etc/rc.firewall need to be customized with the actual values. For our proxy server, the following values were edited under the section [Cc][Ll][Ii][Ee][Nn][Tt] (read as "client"):

```
# set these to your network and netmask and ip
internal="192.168.2.0"
intmask="255.255.255.0"
dmz="192.168.1.0"
dmzmask="255.255.255.0"
serverip="192.168.1.9"
```

Additional edits, deletions and insertions were made to the default rules provided in /etc/rc.firewall so that it now reads as follows:

---

[27] This option is nice to have enabled while building the system. It displays the firewall rules as they are applied during the boot process and provides reassuring feedback that everything is working properly at first. Before deploying the system in production however, set the option to "YES" so that non-privileged users cannot see the ruleset during boot up.

[28] Although not addressed in this paper, it will be important to configure the server to synchronize with the organization's central NTP server. The logs collected on this machine will need to be relative to the logs collected on all other servers in the enterprise including firewall appliances and IDS sensors.

[29] This option tells IPFW to log connection attempts even when there is no listener on the port in question. This may take up significantly more log space but it can provide useful information regarding any reconnaissance scans directed at the server. It is also valuable to see what traffic is being allowed through the network firewall that's protecting the DMZ.

[30] Stokely, Murray, and Clayton, Nik, eds. FreeBSD Handbook, Second Edition. Concord: FreeBSD Mall, 2002. Frequent updates available at http://www.freebsd.org/doc/handbook (29 DEC 2003). For additional discussions on IPFW, see http://www.mostgraveconcern.com/freebsd/ipfw.html (29 DEC 2003) and http://www.onlamp.com/pub/a/bsd/2001/04/25/FreeBSD_Basics.html (29 DEC 2003).

```
[Cc][Ll][Ii][Ee][Nn][Tt])
        #############
        # This is a prototype setup that will protect your system somewhat
        # against people from outside your own network.
        #############

        # variables to define internal ("int") network, dmz network and server ip
        int="192.168.2.0"
        intmask="255.255.255.0"
        dmz="192.168.1.0"
        dmzmask="255.255.255.0"
        serverip="192.168.1.9"

        setup_loopback

        # Allow TCP through if setup succeeded
        ${fwcmd} add pass tcp from any to any established

        # Allow IP fragments to pass through
        ${fwcmd} add pass all from any to any frag

        # Allow setup of incoming 8080 from internal or dmz
        ${fwcmd} add pass tcp from ${int}:${intmask} to ${serverip} 8080 setup
        ${fwcmd} add pass tcp from ${dmz}:${dmzmask} to ${serverip} 8080 setup

        # Allow setup of incoming 22 (ssh) from internal only
        ${fwcmd} add pass tcp from ${int}:${intmask} to ${serverip} 22 setup

        # Allow setup of incoming 80 (http) from internal only
        ${fwcmd} add pass tcp from ${int}:${intmask} to ${serverip} 80 setup

        # Allow setup of outgoing TCP connections
        ${fwcmd} add pass tcp from ${serverip} to any setup

        # Deny and log setup of all other TCP connections
        ${fwcmd} add deny log[31] tcp from any to any setup

        # Allow DNS queries out in the world
        ${fwcmd} add pass udp from ${serverip} to any 53 keep-state

        # Allow NTP queries out in the world
        ${fwcmd} add pass udp from ${serverip} to any 123 keep-state
```

[31] It may not be wise to log every dropped packet. A single Nmap scan created almost 10 MB of log data. During initial deployment however, it's helpful to see what traffic is trying to enter and exit the server. It may turn out to be something you want to allow through like DNS or NTP traffic.

```
# Allow Syslog traffic out to Syslog server only
${fwcmd} add pass udp from ${serverip} to 192.168.2.6 514

# Deny and log everything else
${fwcmd} add deny log all from any to any

# Everything else is denied by default, unless the
# IPFIREWALL_DEFAULT_TO_ACCEPT option is set in your kernel
# config file.
;;
```

Once these edits are made to rc.conf and rc.firewall, the system must be restarted to enable IPFW support with the above options. Any subsequent edits to /etc/rc.firewall can be reloaded by running "sh /etc/rc.firewall" from the command line. Note that we only allow proxy requests on port 8080. If we allowed 3128, users could potentially bypass the content filtering provided by Dansguardian. Also, port 80 (http) and port 22 (ssh) are currently allowed from all hosts on the internal network. This should be changed to restrict access to a few administrators' host addresses.

Firewall logs are sent to syslog with the tag "ipfw". To send these to a separate log file, place the following two lines at the end of /etc/syslog.conf:

```
!ipfw
*.*                     /var/log/ipfw.log
```

Be sure to create the new log file before syslog is restarted (touch /var/log/ipfw.log). Then change the file's mode to 600 (chmod 600 /var/log/ipfw.log). Test the firewall with some permitted traffic (i.e., ssh from another host) and also some prohibited traffic (i.e., telnet from another host). Make sure the first connection is successful and the second connection attempt is denied and logged in ipfw.log. As with the Dansguardian logfile, add an entry to /etc/newsyslog.conf for ipfw.log so that it is archived daily:

```
Logfilename          [owner:group]      mode count  size   when  flags
/var/log/ipfw.log                       600  7      1000   @T00  J
```

Vulnerability Assessment – Part IV:

As set forth in the Security Configuration Methodology section above, the vulnerabilities of the server will be assessed at four intervals during the build process. This forth assessment takes a look at the base install with all additional applications installed and IPFW rules in place. Only those services we intend to have open should be visible and no unacceptable vulnerabilities should be detected.

Nmap: Now that the firewall is in place, a final Nmap scan of the system is conducted to make sure no unexpected ports are visible on the server. This should be done on both TCP and UDP on all 65535 ports. The following results were obtained from a host on the internal network where we allowed ports 22 and 80 in addition to port 8080. First, the TCP scan:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Warning: OS detection will be MUCH less reliable because we did not find at
least 1 open and 1 closed TCP port
Interesting ports on squid.great.lakes.local.1.168.192.in-addr.arpa (192.168.1.9):
(The 65532 ports scanned but not shown below are in state: filtered)
Port       State       Service
22/tcp     open        ssh
80/tcp     open        http
8080/tcp   open        http-proxy
Remote operating system guess: AIX 4.3.2.0-4.3.3.0 on an IBM RS/*
Uptime 0.063 days (since Fri Dec 26 02:14:04 2003)
Nmap run completed -- 1 IP address (1 host up) scanned in 3913 seconds
```

The UDP scan results:

```
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Warning: OS detection will be MUCH less reliable because we did not find at
least 1 open and 1 closed TCP port
All 65535 scanned ports on squid.great.lakes.local.1.168.192.in-addr.arpa
(192.168.1.9) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess
Nmap run completed -- 1 IP address (1 host up) scanned in 6943 seconds
```

Nessus: A final Nessus scan also reveals that the original ICMP vulnerabilities are gone because we are not allowing any ICMP traffic to or from the server. The only vulnerability remaining is the previously detected fact that sshd accepts client connections below version 2.0 and that is an acceptable risk. And, yes, those Nmap scans really did take several hours.

Installation of Tripwire:

Tripwire will be installed to monitor file changes. The original Academic Source Release (ASR) version will be installed.[32] Place the Tripwire-1.3.1-1.tar.gz file in /usr/ports/distfiles. Change directories to /usr/ports/security/tripwire-131 and enter “make install”.

Configuration of Tripwire has been covered in numerous references and will not be repeated here. A user manual for the ASR version is available at

[32] The ASR version of Tripwire is available from http://www.tripwire.com/products/tripwire_asr/index.cfm (29 DEC 2003). You need to provide information about yourself to get it.

www.tripwire.com.[33] For versions 2.3.X, Hack Proofing Linux provides an excellent step-by-step.[34] The key configuration element to Tripwire is maintaining the database on write-protected media such as a CD-R. Additionally, a process must be established to update the database after legitimate changes are made to the OS such as patches and new applications.

Backups:

While not the subject of this paper, the importance of verified backups cannot be over-emphasized. Many organizations create backups, but many fail to conduct periodic restores to validate the backup process. This server will be configured with a local tape drive and the dump procedures outlined in Chapter 12 of The FreeBSD Handbook will be followed.[35] This includes storage in a physically separate location. The restore process will be tested monthly.

Keeping The System Current:

The main source for information on vulnerabilities and bugs with the FreeBSD OS is the FreeBSD Security Advisories at http://www.freebsd.org/security/index.html. Advisories applicable to each FreeBSD release are available at ftp://ftp.freebsd.org/pub/FreeBSD/CERT/advisories/. To receive notification of new advisories and other security issues, it's a good idea to subscribe to the FreeBSD-Security-notifications and FreeBSD-security mailing lists at http://lists.freebsd.org/mailman/listinfo/freebsd-security-notifications and http://lists.freebsd.org/mailman/listinfo/freebsd-security respectively. Fixes should be applied immediately after sufficient testing.

The ports collection should be periodically updated with cvsup. As stated previously, from the command line, run "/usr/local/bin/cvsup –L 2 /usr/share/examples/cvsup/ports-supfile" (–L 2 tells it to be very verbose). CVSup updates all the ports in the /etc/ports directory including the indexes.[36]

Application-specific sites should be monitored to track new vulnerabilities and bugs in the main applications. This includes Squid (http://www.squid-cache.org/Advisories/), Dansguardian (http://dansguardian.org/?page=knownbugs) and Apache

[33] "Tripwire Academic Source Release 1.3.1 for Unix User Manual." 30 APR 1999. URL: http://www.tripwire.com/products/tripwire_asr/index.cfm (29 DEC 2003). You need to provide information about yourself to get it.
[34] Stanger, James, Lane, Patrick, and Danielyan, Edgar. Hack Proofing Linux: A Guide to Open Source Security. Rockland: Syngress, 2001.
[35] Stokely, Murray, and Clayton, Nik, eds. FreeBSD Handbook, Second Edition. Concord: FreeBSD Mall, 2002. Chapter 12.
[36] For a good discussion on keeping ports and sources current, see Imamura, Michael. "Keeping FreeBSD Up-to-date." 18 APR 2002. URL: http://www.lugatgt.org/articles/freebsd_update/ (29 DEC 2003).

(http://www.apache.org) and the sites for those products should be checked periodically. Updated blacklists for Dansguardian can be obtained at http://blacklist.dansguardian.org/. It's wise to subscribe to the Carnegie Mellon CERT mailing list as well at http://www.cert.org/contact_cert/certmaillist.html.

Ongoing Maintenance:

This server will be added to the organization's standard processes for ongoing maintenance. It's OS and installed applications will be recorded as configuration management items and will be included in enterprise patch management efforts.

Other regular activities include:

- Change root password and snmp strings (if applicable) every 90 days.
- Review logs daily for root logins, excessive firewall denys and other suspicious activity.
- Review Tripwire reports daily and investigate any unauthorized file changes. Update database to CD monthly.
- Monitor disk space and particularly log file size (increase frequency of log rotation and archival as necessary).
- Conduct periodic vulnerability scans including Nmap (diff output to determine presence of new services) and Nessus as new plugins are released.
- Test ability to restore system from backups at least monthly.
- Review Squid logs for statistics on heavy traffic sites and cache performance; review Dansguardian logs for excessive attempts to violate policy.
- Review network-based firewall logs for evidence of systems not properly configured to use proxy.

Conclusion:

Every detail of FreeBSD system security could not be addressed in this paper. Because the server itself has a security function in the Great Lakes Consultants network, emphasis was placed on building a working proxy with content filtering capability. Have a great time building your own server. I know I did!

List of References:

Imamura, Michael. “Keeping FreeBSD Up-to-date.” 18 APR 2002. URL: http://www.lugatgt.org/articles/freebsd_update/ (29 DEC 2003).

Jang, Michael. RHCE Red Hat Certified Engineer Linux Study Guide (Exam RH302), Third Edition. Berkeley: McGraw-Hill, 2002.

Pearson, Oskar. Squid: A Users’ Guide. Sandton, South Africa: Qualica Technologies, 2003. URL http://squid-docs.sourceforge.net/latest/zip-files/ (29 DEC 2003).

Stanger, James, Lane, Patrick and Danielyan, Edgar. Hack Proofing Linux: A Guide to Open Source Security. Rockland: Syngress, 2001.

Stokely, Murray, and Clayton, Nik, eds. FreeBSD Handbook, Second Edition. Concord: FreeBSD Mall, 2002. Frequent updates available at http://www.freebsd.org/doc/handbook (29 DEC 2003). For additional discussions on IPFW, see http://www.mostgraveconcern.com/freebsd/ipfw.html (29 DEC 2003) and http://www.onlamp.com/lpt/a/791 (29 DEC 2003).

ViSolve.com. “Squid Configuration Manual.” 15 MAY 2002. URL: http://squid.visolve.com/squid24s1/squid24s1.pdf (29 DEC 2003).

“Tripwire Academic Source Release 1.3.1 for Unix User Manual.” 30 APR 1999. URL: http://www.tripwire.com/products/tripwire_asr/index.cfm (29 DEC 2003). You need to provide information about yourself to get it.