Global Information Assurance Certification Paper

Creating an OpenBSD 3.3 Webserver with Multiple Apache Instances

SANS GCUX Assignment v1.9
Secure Unix Server Step-by-Step
by Christopher Kruslicky

Table of Contents

Executive Summary

This document outlines a specific webserver installation using OpenBSD 3.3 as the operating system. The resulting webserver utilizes the chroot environment provided by OpenBSD. In addition traditional webserver "virtual hostnames" are segregated into separate chroot environments for added resilience in the event of a compromise or untrusted CGI author. The resulting machine provides a single webserver to the outside world by using the proxy support that is part of Apache, without requiring additional hardware purchases. An added benefit is that since each virtual server is segregated from the others, migrating to multiple servers as needed can be more straightforward due to the upfront planning.

Introduction

Project Goals

The OpenBSD team has made an OS with consideration for security in the default install beyond what most would consider the industry standard. One feature is an Apache 1.3.27 Webserver that is chroot'd into /var/www. The goal of this project is to create multiple webservers within a single system that are not only protected by the standard unix file permissions, but also from other CGI authors by the segregation of Apache instances.

One of the challenges with such a setup is manageability. There has been discussions on mailing lists [misc@openbsd.org, http://www.openbsd.org/mail.html] regarding the chroot feature and the additional steps required to enable CGI in this environment. Having multiple, segregated Apache instances will add to the complexity of the system. Multiple chroot'd servers is not a useful product if it becomes unmanageable.

The resulting system will be useful to those building a webserver that can be compartmentalized into distinct areas of content. These content domains need not be completely dissociated from one another, they can merely be portions of active content that each has unique permissions to a backend database. The resulting server can also be useful in a SOHO situation where multiple separate internet domains are served from a single machine due to resource limitations. Likewise, it can be a useful first iteration intranet webserver, where resource needs may be gauged during actual use and additional investments made based on those metrics.

Available Means to the End

It turns out that the authors of Apache had considered the need to protect not only the system from (potentially abusive) CGI authors, but also the content owned by others on the same system. The solution is called suEXEC and was introduced in Apache 1.2 [http://httpd.apache.org/docs/suexec.html]. This may be the preferable solution for a service provider who chooses to allow hundreds of users to execute CGI programs with their own userid, not only from a manageability standpoint, but also because it requires little extra resources of the underlying system.

One of the nice things about suEXEC is that it is written by the Apache team and support can be compiled from the normal Apache package, it is not an additional

piece of software. The default OpenBSD installation does appear to included suEXEC support for those that wish to use it.
There are third party alternatives also, each with their own spin on the same idea. CGIWrap is such product [http://cgiwrap.unixtools.org/faq.html]. The idea is the same as with suEXEC, a setuid root binary is installed, and all CGIs are called through it. That binary then performs some sanity checks on the environment, and sets the effective userid of a single httpd process to that of the CGI owner before executing the actual CGI. Another similar solution is sbox. Information comparing each of these three methods is provided in PDF form at www.extropia.com.
[http://www.extropia.com/presentations/birznieks/pdf/cgi_dev_security.pdf]
The setuid method does seem compelling. Each user has their own CGIs in their own home directory, owned by themselves. The unix file permissions are straightforward, each user can protect their content using their own userid. One drawback is that in most cases URL rewrites are necessary to make the use of these setuid binaries invisible to end users.
While useful for a system with a large number of untrusted users, the setuid solutions are in opposition to the goals of this project. The OpenBSD distribution offers a chroot Apache installation which should be utilized, rather than undoing the chroot by giving the webserver setuid root permissions on each CGI call.
The solution offered in this paper is multiple chroot environments. It will be more useful on purpose-built webservers with several rather than several hundred domains of content due to the additional system users, file system maintenance, and potential resource exhaustion due to the additional processes running on a single system.
To achieve the goals outlined above multiple instances of Apache are started at system startup. Each instance has a different userid, each in its own chroot environment. What's more, the userid of each Apache instance should not be the same as the CGI author who maintains that content. This adds a layer of defense against a hijacked CGI session, but also imposes additional complexity on the file permissions needed within each chroot environment, and management of additional system users.
As always, there are tradeoffs between security and convenience. In this paper a webserver is configured choosing security over convenience, while allowing a functional dynamic content webserver.

Why Multiple, Segregated Webservers

Where multiple Apache virtualhosts may have been used before, on this server each compartmentalized Apache instance will run as a unique system user, unable to access the document root of other virtual servers. This not only protects the system from a poorly written CGI, but also limits damage to other portions of the purpose-built webserver should a vulnerability be exposed in a single CGI.
And since a primary goal is to segregate virtual hostnames from one another, most of the work will have already been done in separating content should additional machines be needed to handle increased demand on the system.

Permissions and File System Considerations

Each Apache instance will run as a unique system user. And, each of the instances will be maintained by people whose usernames are unique from the webservers. Ideally, each instance of a webserver will be maintained by a single person, and a unique login will be used by that person for that purpose. This allows the most granular control over file permissions, limiting the damage possible even from a compromised system account. Additionally, the content developer accounts will not have access to the apache configuration files, preventing accidental or purposeful changes to the environment.

In order to provide the necessary support files for CGI use, files traditionally outside the chroot must be available to each Apache instance. There is more than one way to accommodate this need.

In this paper Perl CGI is supported. No matter how those files are provided, the straightforward result will be to have a file structure that mimics the default layout within the /usr partition, in each webserver chroot. A major difference is that only the files from within the traditional /usr that are needed for Perl are available to each instance of Apache.

The most granular method from the standpoint of least-privileges is to copy only the files required for a given instance into each chroot environment. For those that have the time to research which particular files are needed within each chroot this may be the preferred solution. While not a showstopper, this method would add somewhat to the storage requirements. Most likely this method would add more complexity than the others presented below, due to the proliferation of unique files spread across the chroot environments. While the benefit of adding support for certain modules only where needed is compelling, for this project the increased management overhead outweighs the benefits. Also, following the steps in this paper should provide enough example for those brave enough to try this method to have some success by extending the concept to each chroot.

From both a security and management standpoint, a null filesystem is probably ideal. Using mount_null it should be possible to create a single, unified directory structure containing CGI related support files, then mount that structure read-only within each chroot. [http://www.openbsd.org/cgi-bin/man.cgi?query=mount_null&apropos=0&sektion=0&manpath=OpenBSD+Current&arch=i386&format=html] This is a win from a manageability standpoint as only a single set of files is maintained, and each webserver requires a single, easily scripted system call to make that structure available within the chroot environment. Unfortunately, it is not possible to use this option with an official OpenBSD release.

OpenBSD 3.2 had support for null fs within the kernel, however it was buggy. Testing has shown that for the most part it works, but unmounting a null mountpoint caused either a kernel panic or a complete system lockup, depending on how it had been used. Support for null fs had been removed from the OpenBSD 3.3 kernel for the official release, so it is not even an option. There are indications that it has been fixed in the current OpenBSD snapshots as of May 14, 2003, judging from this line in the daily changelog:

"Re-enable NULLFS, UMAPFS and UNION in the GENERIC kernel."

[http://openbsd.org/plus.html] OpenBSD Current is typically not used for production servers as the snapshots are intermediate versions of the entire OS during ongoing development. Perhaps with the release of OpenBSD 3.4 this will be the supported method of choice. In the meantime, Appendix A provides an brief overview of this method.
For the purposes of this project, a compromise is needed that provides a single point of configuration the way null fs would, yet provides the necessary files in its absence. The resulting method used for OpenBSD 3.3 is to populate a directory in /var/www/usr with the files needed to support Perl CGI, then hardlink to that directory within each chroot. Using hardlinks requires the actual files to reside on the same file system, it also means that each instance of the webserver will be accessing the actual files more directly. The support files cannot be mounted read-only for example. However, it does provide a more straightforward management plan than copying the files individually, and through the use of chflags, the support files themselves can be made immutable.
[http://www.openbsd.org/cgi-bin/man.cgi?query=chflags&apropos=0&sektion=0&manpath=OpenBSD+Current&arch=i386&format=html]

Webserver Specifications

Hardware Description

The machine being configured is a 1GHz Athlon (Thunderbird) system with a single 256MB non-DDR DIMM. The hard disk is a 30GB, 7200 RPM IDE disk, the CD-ROM a Sony CD-R unit capable of 2x CD writing and 4X reading. A generic floppy disk is installed but not in use. An additional fastethernet interface is added, it is a generic RealTek capable of 100Mbps full-duplex. The motherboard uses AMIBIOS v1.21.06 which is used to disable the USB ports. A dmesg is included in Appendix B.

Operating System Description

The operating system being installed is OpenBSD 3.3. This is an official release installed from a CD set purchased at http://openbsd.org/orders.html.

Risk Analysis

Physical Security

Ideally the server would sit in a room with access control enough to prevent unauthorized persons from even knowing the machine exists. In reality, it may be that it sits in the corner of a home office. But as resources spent securing the physical machine may indicate the worth of the data on it, and vice versa, this will probably be sufficient to prevent unauthorized use. As an example, if the machine were to be placed in a marketing booth as a demo unit making the rounds at conventions, often left unattended, it should definitely not contain sensitive data.
The role of this machine is such that the sensitive data it contains is perhaps a login to a database. If the information on the database server warrants hiring a security guard to watch over the machine, then the webserver should be placed in the same rack monitored by the same security.

Presumably if the machine were stolen it would be noticed. In that case the database logins would be changed immediately. As none of the system logins should be the same on the webserver and machines with such sensitive data, there should not be a high risk of compromised administrative accounts. The firewalls should prevent the database login from being used remotely, although some information regarding trusted networks would be leaked in the form of the webserver PF configuration.
Note that by default OpenBSD denies everyone but root the ability to mount a CD, even from the console.
Boot Options
No passwords are used to prevent system boot as the system is expected to come up automatically. This BIOS offers the password choices of 'always' and 'setup', setup is chosen. The password is not the same as any system accounts on the server.
Console Access
For the purpose of this paper, the webserver does have a monitor and keyboard connected. In a datacenter, console access should be configured via authenticated serial connection, preferably with an alarm if the cable is disconnected. In such a case it is possible to restrict which method(s) of console access are possible using /etc/ttys.
Backup Media
Physical access to the backup media should be similar to that of the machine itself. Again, the worth of the information will determine the method used. In a small office environment it may be enough to create a backup tarball, use blowfish to encrypt that file, and write the file to CD which is ultimately stored in a safe deposit box. This has the added benefit of an off-site backup. Once a tarball is created with the data to be backed up, the following command can be used to encrypt the file:
$ openssl enc -blowfish < file.tgz > file_tgz.bf

To restore the backup, the encrypted file can be expanded directly from the CD using a command similar to:
$ openssl enc -d -blowfish < /mnt/cdrom/file_tgz.bf | tar -zxf -

Network Security
It is often joked that the only secure computer is disconnected from the internet, stored in an underground bunker with an armed guard, and no power source. Or something similar to that.
However, it would not make a very useful webserver in such a state. Luckily OpenBSD 3.3 comes with a pre-installed stateful firewall that, on modern hardware, imposes very little overhead. It is used to prevent access to services other than those intended: HTTP for the webserver, and SSH from a few select workstations for management purposes. Likewise, in the event of unauthorized access, outbound connections are restricted to allow Syslogging to a remote server only. This makes it much more difficult for an intruder to bring software onto the system to elevate their privileges.

Having one or two TCP ports open to a potential abuser is still cause for concern. It is absolutely necessary that any publicly known vulnerabilities with either Apache or OpenSSH be patched as soon as possible. Sooner even. If that's not possible for some reason, then firewall off the port until the system can be patched in person.

System Installation

Preparation

Installation Media

Each of the supported installation methods is fairly straightforward, but purchasing an official OpenBSD CD set does save time as well as support the project. That is the method that will be used throughout this paper, although FTP and other options are available. There has been more demand for more recent releases, which translates into longer delays before the product is shipped. The OpenBSD 3.3 release took almost a month to arrive from the date of the order. It is certainly possible to order the CD yet install via FTP, but the CD does come with instructions in the sleeve, very useful for a first time OpenBSD install.

Disk Requirements

A stock OpenBSD 3.3 installation could fit onto a 300MB disk, although that would leave little room for additional software, not to mention content. Using disk partitions is strongly encouraged both to protect partitions from one another, and to follow the principle of least privileges with the mount options. However, since a duplicate development system should be setup with distribution sources (for patching) and disk space for additional software. To accommodate these needs, the server will follow the recommendations in the CD sleeve, giving plenty of room to all partitions above those recommendations since disk space isn't really at a premium.

Network Parameters

Obviously network settings like IP address, subnet mask, and default gateway will be needed. DNS servers, hostnames, and domain names will be needed as well, better to have those ready ahead of time. There will be two interfaces on the server, one for the public side web server, the other for internal management and logging, this will need to be taken into account as well. Typically the management side will use private (RFC 1918) addresses on a separate segment. In the ideal case this would not be part of the internal network nor would the management segment be directly available via the internet.

Server Requirements

Server requirements can be specified in terms of webserver performance using a variety of metrics: connection latency, concurrent successful connections, and average throughput could all be considered. In this case the server hardware was available and tested for acceptability, rather than going through a formal cost analysis.

Maintenance Considerations

We are assuming a webserver that is physically accessible to the administrators within a reasonable amount of time. Note this could be accomplished by providing remote console access via serial cable for most tasks. This console

access is important because single-user mode will be needed to update certain binaries on the system after it is hardened. The development server, an identical platform, will be the one that is patched regularly, with software updates being copied to the production server in binary form.
OpenBSD is released twice each year, the plan will be to format the development server, install the new release, and restore the data from backup. Thus the development server becomes the new production server. It should only be necessary to do this once per year unless there is compelling reason each release. Also the production server would then obviously be formatted and setup as the development server.

Additional Software

Although the CGI language choices will be kept to a minimum for security and management simplification, there may still be the need for additional libraries/modules. Also there are some packages that will be added to monitor the system. Some consideration will be needed for the update of the CGIs and addition of Perl modules. Although CVS would be a good solution to provide CGI updates within an organization, in this case it may be preferable to maintain the separation established by the multiple chroot environments, and maintain a strict policy of SCP/SFTP for file transfers.

Initial OpenBSD 3.3 Installation

Disk Layout

In anticipation of the website content comprising the bulk of data on the system, **/var/www** is setup as the largest partition. Assume that a separate development server will be setup for compiling and patching the OS, and developing and testing the website.
This implies the **/usr** partition itself will not need to store ports, packages, or source code. There is also no compelling reason to include the X Windows software at this point. This base install requires less than 200MB for **/usr**, however we still make it larger in case the disk space is needed later.
The **/home** directory is created last as per the CD sleeve instructions, the size is a rough estimate as it is simply what is left over after the other partitions are created.
Given the parameters above, these partition sizes were chosen:
a] /root - 1G
b] swap - 1G
d] /tmp - 700M
f] /var - 2G
g] /var/www - 20G
h] /usr - 2G
i] /home - ~1.2G

Initial Network Setup

After the disk formatting completes, network configuration is straightforward with the installation script. During this phase there is no network access, however it is easier to use the installer script, even though manual configuration will likely be necessary at some point in the lifetime of the server.

The install script places hostnames, including localhost, into **/etc/hosts**. Even though DNS will allow many hostname aliases, using the hosts file means the server does not rely on DNS for its own interface names. First the hostname is chosen to reflect the purpose of this machine, www.
IP address and DNS configuration is not shown as it will likely not be portable anyway. The domain name is shown, for this purpose it is an IANA reserved domain name, example.net.
Note that for each interface, there is the option to use the DHCP client for automated configuration. This is not the recommended method on production servers however, any interfaces should be statically configured.
Below is a summary of relevant settings for this server:
hostname: www
external i/f: rl0
domain name: example.net

Package Installation
Next on the list of installation CD steps is the package chooser. As a production webserver, we can uninclude the games packages. There's no reason to have them on a server not meant as a shell server. The only command needed to remove the games packages is:
-games33.tgz

The rest of the packages are mostly necessary. Although it could be argued that having the manpages on the system is a potential security risk, the same manpages are available online for the world to see. Having the manpages installed takes little diskspace compared with the benefit of having the information readily available to the legitimate users and administrators.
The compiler components are also removed. Again, this is done by using a minus in front of the fileset name:
-comp33.tgz
Although one would expect someone capable of compromising the system to have the resources to build their own software, there is the potential in preventing accidental security risks caused by legitimate users by removing the compiler components. If it is known that some software will be compiled from source before deploying the server then it may be useful to do so on the server itself, removing gcc and other binaries manually later. In this case the package system will be used to install binaries directly from the official OpenBSD 3.3 CD set. So the compiler is not needed.
TimeZone Setting
There's only one step after the OS files are installed, setting the timezone. This particular host is being configured on the East Coast of the U.S.A., we choose the proper timezone which will take into account daylight savings time:
EST5EDT

Administrative Account Creation

To boot the server issue the *halt* command and wait for the disk to finish synching, then reset machine. A feature of this particular BIOS is that by default there is an F-key option to change the boot disk, if this were not the case the installation CD would need to be ejected before issuing the soft-reset.
Traditionally there is a minor quirk to the OpenBSD installation that results in the root password having a slightly weaker hashing in the shadow (/etc/master.passwd) file. This may still be the case as the installation instructions in the CD sleeve state that the root password should be changed after rebooting to the disk. This has stemmed from a different number of rounds being specified on the boot disks than in the default OS install. So, still having only console access, login as *root* and use **passwd** to change the *root* password.
The next step is to create an account to use for the role of administering the system. Again, this isn't a shell server, so even the administrator does not need to login often (we hope). For this reason I choose an account name that isn't my usual login, nor my email address name, nor an easily guessed name like *admin*. For this server I choose *charlie* because I know I'll remember it based on the "Full Name" setting for *root* in **/etc/passwd**, yet I hope it's not too obvious. This is the user that belongs in the *wheel* and *operator* groups. The **adduser** command provides a prompt-driven method of adding users. The main settings in this case are the username, *charlie*, the standard shell, **/bin/sh**, and when prompted whether to add the user to additional groups, invite charlie into the *wheel* and *operator* groups.
Test that the system can be administered without needing to login as *root*: logout, login as *charlie*, and **su** to *root*.
System Warnings and Banners
The Department of Energy in the U.S. has issued an advisory on the use of login banners to aid in prosecuting intruders and enforce internal policies. [http://www.ciac.org/ciac/bulletins/j-043.shtml] The advisory outlines in more detail why login banners are important, and gives examples for use not only for interactive logins to the operating system, but also using JavaScript popups and various other mediums. Any policy notifications issued via the web would necessarily be specific to the content owners and is beyond the scope of this paper. If banners via webpage are desired, the CIAC advisory is a good place to start.
OpenSSH includes an option to display a banner before login via the daemon configuration file in /etc/ssh/sshd_config [OpenSSH v3.6]. To keep things simple the banner is configured in /etc/banner. This is a seperate consideration from the message of the day issued after login via /etc/motd on the system. Taken together the two system warnings can make a defense of ignorance very difficult indeed. Appendix C shows the login warnings issued by this system when logging in via SSH. These files can be copied from another machine once network access is granted, however they will show as a change in the file integrity database created while offline.

OpenBSD 3.3 Setup

Replace the installation CD, and mount it. You may need to create a mount point for it using mkdir -p /mnt/cdrom, before mounting the drive with something like

mount_cd9660 /dev/cd0c /mnt/cdrom. This will be necessary to install some of the included packages on the CD.

System Initialization Configuration

The /etc/inetd.conf file however does have a few network services enabled by default, they are: ident, comsat, daytime, and time. Since this server is purpose-built as a webserver, there is no need of comsat or pop3 services, likewise kerberos is not going to be used, so really /etc/inetd.conf is not needed at all. The file can be completely removed.

The initial startup scripts are minimal in the default OpenBSD install. Settings are kept via environment variables in /etc/rc.conf. The vast majority of those are set to "NO" to prevent services from starting. A notable exception is the inet daemon. Since /etc/inetd.conf is not going to be used to offer network services anyway, change that line to read

inetd=NO

and save /etc/rc.conf.

/etc/login.conf configures the process priority, umask, and resource limits for all processes spawned by init [man init]. Add :coredumpsize=0: to both the default and daemon classes, also add it to the staff class if any users are added to that group. The default group also gets a :filesize=512MB: just in case someone "accidentally" could fill the disk. Also, the default class gets :passwordtries=0: so that passwords must always meet the minimum requirements (6 characters) - default behavior is to verify a weak password 3 times then accept it anyway. The resulting file is shown in Appendix D.

sshd_config

Uncomment port 22 and Protocol 2 to make the intention explicit. ListenAddress should be set to the IP address on the management network rather than all available addresses on the system. PermitRootLogin gets set to no. The banner is in /etc/banner. The full configuration file is included in Appendix E.

fstab

The file /etc/fstab contains a line for each mount point to be added at system boot. Since this server is only expected to be used as a webserver, and since /var/www is mounted separately, mount options can be used to restrict most of the system. The distribution file already uses the nodev and nosuid options on most of the partitions created. To tighten things a little more, add the noexec option to /home, and /var. This is most usable for now, but prior to being placed in production some will wish to mount /usr as read-only. Likewise, it would be preferable to mount /tmp with noexec in production, however for now, that cannot be done until additional software packages are installed.

There is also an option on FFS filesystems to use soft dependencies. These can significantly speedup file metadata writes, to use this option add softdep to all mount points. The resulting fstab file is shown in Appendix F, just remember for now /usr is not mounted read-only, and the noexec option is not yet implemented on /tmp.

Note that there is a directory /var/tmp which happens to be used alot internally to OpenBSD. Most notably, vi.recover files and package maintenance use that directory. It is removed and replaced with a symlink to /tmp.

```
root@www# cd /var/
root@www# rm -rf tmp/
root@www# ln -s /tmp/ tmp
```

sysctl.conf

The use of commented configuration options is roughly opposite of sshd_config, comments are not the default, but are the values people typically want to change. The changes made here affect the TCP/IP stack to disable redirects, both generated and received, and reduce the errors generated from 100 to 2 per second. The TTL value is also changed from the default of 64 to 57. By itself, this does not make a huge difference, but it could help slightly to distract OS fingerprinting attempts. That change is not expected to cause reachability problems as most hosts on the internet can be reached in less hops than that anyway. In a study by Fred Baker it was found that the average hop-count between two internet hosts in the sample space was just under 16. [Method 2 at http://www.nlanr.net/NA/Learn/wingspan.html] ICMP timestamp replies are also disabled. Some more values are explicitly set to their defaults, making it easier to tweak the config file later.

Additional Loopback Interface Aliases

Each virtual webserver will require its own internal IP address. This is accomplished via the loopback interface, lo1. The default loopback, lo0, is not disturbed this way. In this case, IP addresses from an RFP 1918 address space is chosen, and will not be routed to this host:

```
root@www# cat /etc/hostname.lo1
inet 172.16.0.1 255.255.255.0
inet alias 172.16.0.2 255.255.255.255
inet alias 172.16.0.3 255.255.255.255
```

The first entry, 172.16.0.1 is setup with a large subnet mask so that the internal routing table can find any IP in the range via this interface. The additional aliases are needed so the system considers those IP addresses internal, and Apache can bind to them individually.

Host-Based Firewall Configuration

The standard configuration file for the included stateful firewall, PF, is the file /etc/pf.conf. Before connecting the freshly installed system to the network, configure PF to allow SSH access from one internal workstation. To activate the change, modify /etc/rc.conf so that pf=YES. Since this host will not be acting as a router, there is no NAT configuration and the only configuration needed is actual packet filter settings. To make management of the file easier later the interface name, management IP, Webserver IP, and management host are all setup as macros (or tables). The actual filter configuration becomes simpler to read and maintain. The end result is shown in Appendix G, it includes configuration for access to the webserver as well as restrictions on outbound connections. Since this server is not for shell access it is prefered to prevent its use as such.

Perl Modules

For this particular server the base install is very close to what is desired. It is wished that no external software be added, however depending on use some may be required. Perl Modules for MySQL access are included on the CD for example. The OpenBSD 3.3 packages can be found here:
/mnt/cdrom/3.3/packages/i386
(Additional packages are also available via FTP from OpenBSD.org:
ftp://ftp.openbsd.org/pub/OpenBSD/3.3/packages/i386/)
To add these using the pkg_add command, become root with su, change directories to the package location shown above, then use pkg_add <package name>.

Further System Hardening

George Shaffer wrote a very good set of documents on hardening an OpenBSD server. [http://BSD.GeodSoft.com/howto/harden/] It is recommended reading, worth the time to digest many of the ideas presented whether they are implemented or not.

The base files in /bin and /sbin are good candidates for using chflags to set the immutable flag. These files should not need to change and should be depended upon. The files in /usr/bin and /usr/sbin are more likely to need patching at some point, but it should be worthwhile to patch a development server and apply those patches to the production server during a time of planned maintenance. This would be necessary when copying the files into place as the system immutable flag makes it so even the root user cannot delete/modify a given file. Here, all the files in each of those directories is set to system immutable except /usr/bin:

```
root@www# chflags schg /bin/*
root@www# chflags schg /sbin/*
root@www# chflags schg /usr/sbin/*
```

There will be error messages such as:
chflags: /bin/test: Operation not permitted

This is normal, it is a result of some files being hardlinks of other files. Once the original file is set system immutable, the same file cannot be set again when it is referenced by another name.

This server will not need some of the binaries installed by default, and may have many of the configuration file permissions tightened without adverse affects. It can be difficult to maintain a list of files to remove when upgrading to another OpenBSD release, and most of the unwanted services are to be firewalled off in both directions anyway. For these reasons it is assumed someone capable of utilizing a program like uucp will also be capable of adding the file if removed. While a file integrity report may show such a change, it may already be too late in such a case. The decision is made that removing the unneeded binaries can be done but will not be outlined in this paper. Likewise, many of the configuration files in /etc can safely be changed to remove the world-readable bit, for the purpose of this server it would not provide enough of a benefit. The one exception is the system initialization files: /etc/rc*. Change these to be writeable by root only, and readable by the wheel group:

root@www# chmod 640 /etc/rc*
Host-Based File Integrity Configuration
In case it hadn't been done yet, the steps are repeated here to install a package. In this case, installation of aide v0.7 for use as a file integrity checker.
root@www# mkdir -p /mnt/cdrom
root@www# mount_cd9660 /dev/cd0c /mnt/cdrom/
root@www# cd /mnt/cdrom/3.3/packages/i386/
root@www# pkg_add -v aide-0.7.tgz
The example /etc/aide.conf configuration file included in Appendix H will be used on the server in production, for now though the default configuration file is kept in place, and copied to /etc/aide.conf.dist. Then the file /etc/aide.conf is modified to also look in /var/www/usr. The actual integrity database build will be done later, but for now it is enough to know to add the following line at the end of the /etc/aide.conf file:
/var/www/usr R

While Appendix H shows a more complicated configuration file, this is configuration is used as a baseline, pre-networked snapshot.
cdrecord Installation
System backups will be made using the cdrecord program, available as a package under OpenBSD. Assuming the CDROM is still mounted on /mnt/cdrom as shown above, the following will install the necessary software:
root@www# cd /mnt/cdrom/3.3/packages/i386/
root@www# pkg_add -v cdrtools-2.0.tgz

User Accounts
Some standard naming conventions are needed to make maintenance easier. For the webserver userids, the names will all begin with www, followed by a hyphen, then a short, preferably 3 letter designation for the hostname where it logically resides. For example, the webserver answering as www.example.net would have the user and group names www-wen. Likewise, the content maintainer logins will begin with usr, hyphen, and the same suffix. So the same website would be maintained by someone with the login usr-wen. The webserver account is created first, and the user is invited into the webserver group. This allows files created by the user to be read by the webserver without going through too many hoops.
Later a maintenance script will be used to create the httpd.conf files for each virtual webserver. It will not create the users and groups however. This seems like a simple enough step that a checklist and manual configuration is actually preferred, so that each new instance configuration is manually checked at some point.
Here are the manual steps required for the proxy server which handles all requests initially, it is unique in that it does not server webpages, and does not have a content maintainer. Also, it does not follow the convention above, this is to differentiate it slightly as an account name, and because it does not necessarily fit into any of the domains being served.

```
root@www# groupadd www-proxy
root@www# useradd -g www-proxy -d /var/empty/ -s /sbin/nologin -c "HTTPd
Proxy Account" www-proxy
```

And this example shows a standard user setup for the virtual hostname login.example.net. The first two steps setup an account for the Apache process, without specifying a valid shell nor password. The last step uses the interactive **adduser** command to setup an account for the person who will maintain login.example.net web content:

```
root@www# groupadd www-len
root@www# useradd -g www-len -d /var/empty/ -s /sbin/nologin -c "HTTPd
process - login.example.net" www-len

root@www# adduser
Use option ``-silent'' if you don't want to see all warnings and questions.

Reading /etc/shells
Check /etc/master.passwd
Check /etc/group

Ok, let's go.
Don't worry about mistakes. I will give you the chance later to correct any input.
Enter username [a-z0-9_-]: usr-len
Enter full name []: Maintainer - login.example.net
Enter shell csh ksh nologin sh [sh]:
Uid [1004]:
Login group usr-len [usr-len]:
Login group is ``usr-len''. Invite usr-len into other groups: guest no
[no]: www-len
Enter password []:
Enter password again []:

Name:     usr-len
Password: ****
Fullname: Maintainer - login.example.net
Uid:      1004
Gid:      1004 (usr-len)
Groups:   usr-len www-len
HOME:     /home/usr-len
Shell:    /bin/sh
OK? (y/n) [y]:
Added user ``usr-len''
Copy files from /etc/skel to /home/usr-len
Add another user? (y/n) [y]: n
Goodbye!
```

Verify the accounts were created properly:
root@www# grep len /etc/group
www-len:*:1003:usr-len
usr-len:*:1004:

root@www# grep len /etc/master.passwd
www-len:*:1003:1003::0:0:HTTPd process -
login.example.net:/var/empty/:/sbin/nologin
usr-
len:$2a$07$ro806/kX5ufbIVI2qo5MBudHbEt5GagJoi7U1.LAtkI8NK.5nAO1m:100
4:1004::0:0:Maintainer - login.example.net:/home/usr-len:/bin/sh

It is shown in the master.passwd file that the HTTPd process account www-len does not have a valid password, and the shell is set to /sbin/nologin as expected.

Segregated File Structure

As with user accounts, directory structures will need some consistency to make things more intuitive for anyone administering the system. The choice is made that each chroot should exist within a directory of the same name as a virtual hostname, within /var/www. For example, the webserver answering to login.example.net would be chrooted within /var/www/login.example.net. There are two exceptions however. First, the proxy server will sit within /var/www/proxy to signify its difference of purpose. Also, there will be an unused (as a server) directory structure setup within /var/www/template, used as the template for all the other virtual servers being created.
The directory structure in all cases is based on the default installation within /var/www. So start by moving this structure for further work.
root@www# cd /tmp
root@www# mkdir -p var_www
root@www# cd var_www
root@www# mv /var/www/* .
root@www# rm cgi-bin/*

The Proxy File Structure
The directory /var/www should now be empty. Copy files from /tmp/var_www into a newly created directory called proxy:
root@www# cd /var/www
root@www# mkdir proxy
root@www# cd proxy
root@www# cp -PR /tmp/var_www/* .

The next step is to configure the proxy server to only handle proxy requests, and serve no webpages itself. The resulting configuration file is shown in its entirety in Appendix I. For now, the main issue will be making it easier to test it is operating properly later. So, remove the Apache manual from htdocs, and add a single file that will make it obvious if the proxy is serving its own webpages:
root@www# cd /var/www/proxy/htdocs/

root@www# rm -rf *
root@www# echo "<html><body><h3>PROXY SERVER</h3></body></html>" > index.html

The Template File Structure
Finally, configure a template structure to be copied each time a new virtual server is added. The full text of the configuration file is in Appendix J. The basic gist of it is to place the token _TEMPLATE_ throughout the config so that a script or even vi can be used to search and replace on it. For example, the hostname will become _TEMPLATE_, and when creating the httpd.conf file for login.example.net, it will be replaced by that text.
Following the steps above to copy from /tmp/var_www/* into /var/www/template, it will contain the same files as the original distribution (minus the files in cgi-bin.) The difference from the proxy is that the Apache manual is left in place, so that it can be used for testing purposes beyond a single webpage, to ensure self-referencing works properly. Also, each instance will behave roughly as a new Apache install normally would.
root@www# cd /var/www
root@www# mkdir template
root@www# cd template
root@www# cp -PR /tmp/var_www/* .

CGI Support
Directory Structure
A structure is created within /var/www/usr to provide the files needed to run Perl in this environment. Essentially, this is the same step that would be done to use Perl as an interpreter with the default install in /var/www. Although no webserver will be chroot'd within /var/www (only below that point) that is still used so that hardlinks can be created within each chroot.
It is important to note that the entire contents of /usr are not copied, and very important not to include any setuid binaries. There are ways for the root user to break out of a chroot. Each webserver instance is running as an unprivileged user and should not have the means to elevate those privileges. Also, the files in this directory will remain world-readable, but owned by root. It would be possible to create a webserver group to which all www-fqhn users would belong. However, that seems likely to cause more maintenance headaches later than it is worth.
root@www# cd /var/www
root@www# mkdir usr
root@www# chown root:daemon usr
root@www# chmod 755 usr

Populate Binaries
It is possible to find out which files are needed for mod_perl by using ldd. The command will find libraries that a binary depends upon. Instead, on a development server with a similar setup, more of a trial and error method was

used. By carefully inspecting the error_log output by Apache while trying to run a simple CGI, the necessary libraries were found fairly quickly. Obviously there needs to be a file usr/bin/perl, and to simplify matters everything from /usr/libdata/perl5 was included. The resulting file structure can be recreated as follows:

```
root@www# cd /var/www/usr
root@www# mkdir bin
root@www# cp /usr/bin/perl bin/

root@www# mkdir lib
root@www# cd lib/
root@www# cp -R /usr/lib/apache .
root@www# cp /usr/lib/libc.* .
root@www# cp /usr/lib/libm.* .
root@www# cp /usr/lib/libperl* .
root@www# cp /usr/lib/libutil.* .

root@www# cd /var/www/usr/
root@www# mkdir libdata
root@www# cd libdata
root@www# cp -R /usr/libdata/perl5/ .

root@www# cd /var/www/usr
root@www# mkdir libexec
root@www# cd libexec
root@www# cp /usr/libexec/ld.so .

root@www# cd /var/www/usr
root@www# mkdir -p local/libdata/perl5/site_perl/
root@www# cd local/libdata/perl5/site_perl/
root@www# cp -R /usr/local/libdata/perl5/site_perl/* .
```

This last step, copying the files from /usr/local/libdata/perl5/site_perl will copy any additional modules installed using pkg_add. It is also within the Perl include path, making the site_perl directory a good place to add customer modules later. If there are any customer modules that will be used on all webservers, which can be copied via CD, now is a good time to add them as well.

Mirror Directories

When it comes time to create additional virtual webservers, the find utility will be used, along with xargs, to create a directory structure within the chroot that is identical to /var/www/usr. There is a tradeoff to be made with regards to security and ease of maintenance. Once a set of chroots is created and stable (no more expected to be added) the immutable flag can be set across /var/www/usr files. However, it is not possible to add additional hard links to those files once that is done. This implies that no more chroot environments can be filled using hardlinks.

The decision can be weighed on an individual server basis, perhaps making the option of populating each chroot with a copy from the original /usr more appealing, as each can be individually set immutable. For the purposes of this paper, the first approach is taken, implying that additional chroot environments can only be populated by rebooting into single-user mode. The benefit of this approach is that while any patches applied to existing perl components will also require single-user mode access, there will only be one directory that needs to be updated for the patches to be applied across all virtual webservers.
Note that this should be alleviated in a future OpenBSD release by the use of null fs, assuming support will be reintroduced to the generic kernel. It is possible that recompiling the kernel in OpenBSD 3.3 can be done to accomplish its use now, it is not supported and will cause at a minimum an fsck every time the system is rebooted (even intentionally).

Starting Each Apache Instance at System Boot

The last step to the basic configuration is to modify /etc/rc.local and /etc/rc.conf so that starting the additional Apache services is automatic. The config file rc.conf is where environment variables are set for use the /etc/rc and /etc/rc.local scripts. The default httpd_flags variable will remain set to NO, and a comment added that it shouldn't be used. Technically, by removing the supporting configuration file from /var/www it should fail to load at boot anyway.

/etc/rc.conf

This is how the appropriate section of /etc/rc.conf should look:

```
# use -u to disable chroot, see httpd(8)
httpd_flags=NO          # for normal use: "" (or "-DSSL" after reading ssl(8))

### use NO above normally, set wwwdomains to NO to disable all webserver
# space-seperated list of chroot directories to start services
wwwdomains="proxy"
```

Later, when additional domains are added, the wwwdomains variable will be set to a space-delimited set of names, for example:

```
wwwdomains="proxy example.net login.example.net"
```

/etc/rc.local

The rc.local file is executed after the usual /etc/rc script. Added to the end of that file is a for loop which starts an Apache instance for each domain listed above:

```
# Custom Apache Processes
if [ "X$wwwdomains" != X"NO" -a -x /usr/sbin/httpd ]; then
        for conffile in $wwwdomains; do
                echo -n "HTTPD: starting $conffile ..."
                /usr/sbin/httpd -f /var/www/$conffile/conf/httpd.conf
                echo "done."
        done
fi
```

This will provide a nice informative message as each process is started during system boot. Unfortunately, there's no simple way to test for success of /usr/sbin/httpd, but any actual error is echoed during boot up and normally remains visible on the console after the login prompt is printed.

Base Server System Snapshot

Now that the system is sufficiently configured it is almost ready to be connected with the internet. First, a baseline snapshot is taken and burnt onto CD-R for backup purposes, to make rolling out a duplicate server easier, and to store the file integrity database on a read-only medium.

Create File Integrity Database

As root run aide with the --init flag to create a brand new database. Also explicitly point aide at the configuration file in /etc/aide.conf. Optionally time the program run to get an idea of how much load this may put on the system once in production:

```
root@www# time /usr/local/bin/aide --init --config=/etc/aide.conf
  33.23s real     9.56s user     2.33s system

root@www# mv /var/db/aide.db.new /var/db/aide.db
```

Also, move the file into /var/db/aide.db to indicate it is the baseline. On the hard drive it can be the original database used to compare the current system, using the --check flag to aide.

Create A Snapshot CD

As much as possible the hope is to create a new system by using the OpenBSD install CD for the basic boot options and disk partitioning, then extract the rest of the system using the Snapshot CD. The idea is to do everything as above when installing a second system, but choose none of the filesets, rather escape to a shell with "!" and untar from this CD manually.

To create the CD, create a file listing and use tar to create a snapshot from that listing. The file listing at a minimum will need to have the /tmp directory removed. Then burn that tarball onto a CD-R (preferably not CD-RW).

```
root@www# cd /
root@www# ls -1 > /tmp/filelist.txt
root@www# vi /tmp/filelist.txt
root@www# cat /tmp/filelist.txt
altroot
bin
boot
bsd
dev
etc
home
mnt
root
sbin
stand
```

```
sys
usr
var

root@www# mkdir -p /tmp/cdr
root@www# tar -zcvf /tmp/cdr/site33.tgz -I /tmp/filelist.txt
root@www# cd /tmp
root@www# /usr/local/bin/mkisofs -R -o cd.iso cdr/
```

Once the disk image is ready, a blank CD-R is inserted into the CD writer. It may be necessary to umount /mnt/cdrom before removing the OpenBSD CD if that hasn't been done yet.

```
root@www# /usr/local/bin/cdrecord -v dev=/dev/rcd0c speed=2 -data cd.iso
```

The benefit of using the name site33.tgz is it will be a valid file set to the OpenBSD installation script. It should be possible to create a boot floppy, and use this system backup when it comes time to install files from the CD. Also, it may be necessary to change where the tarball is created, and where the ISO disk image is created, depending on the size of various disk partitions. The resulting disk image is less than 200MB. It should be relatively easy to fit the files from a production server, minus any very large webcontent directories, onto another CD for a second, production system snapshot.

Connecting to the Network

At this point the server can be rebooted, and the network cable plugged in while the server comes up. Test that it is possible to SSH into the machine from the management workstation while console access is still relatively painless.

Maintenance Plan

Adding an Apache Instance

There's only a few steps to add another virtual webserver with its own hostname. The first is to add the user accounts as shown above for login.example.net. Next, add another alias to /etc/hostname.lo1 and issue the same command as root on the command line. Thirdly copy the template directory within /var/www to a new name, using **cp -RP** to copy recursively and preserve file permissions. Modify the conf/httpd.conf file, changing any occurrence of _TEMPLATE_ to the new hostname, and each CHANGEME to an appropriate value: for example, the user and group names added in the first step, and the IP address added in the second step here. Finally, add a virtualhost directive in /var/www/proxy/conf/httpd.conf allowing access to that hostname via the IP chosen in the second step. The configuration file included in Appendices I and K should help illustrate the results.

Note that while one webserver instance will not have access to the files of another, it is still the best practice to make sensitive webcontent owned by the maintainer of that content, and accessible to the webserver using the appropriate group and read-only permissions.

Backups

It is commonly accepted that regular data backups are absolutely necessary. While it should not have a huge impact on system performance to make a system backup from time to time, realistically, only a few configuration files should be necessary. The reason is that web content should be developed on a seperate, nonproduction server, with its own revision control and backup plan. In fact, even the internal (lo1 interface) IP addresses should be the same on the development server, so the vast majority of configuration files would be consistent between the servers. The main differences would be the configuration file for the proxy and network configuration files, which change infrequently. Although no automated plan is described here, the procedure for modifying the production server configuration should include backing up those files, preferably to the development server in a CVS repository. Recovering from a complete hardware failure should consist of a base OS install from the official OpenBSD 3.3 CD, restoring files from the site33.tgz file stored on CD, and overwriting the configuration and content files in /var/www from the development server.

Patching

The production server should only be patched after testing the patch on the development server. This means patches are compiled on the development server, and the resulting files copied to the production machine. It may be necessary to move the files into the administrative home directory, and boot into single-user mode to apply the patches from the console, due to system immutable flags.

It should be noted that (hopefully) most patches to come from the OpenBSD team will not be necessary on the purpose-built webserver. Some software simply won't be used on the system, and other, local-only vulnerabilities may be more easily worked-around by removing the offending files if they are unneeded in production. Also, patches from the OpenBSD team come in source code form. This means the development server does need to have the source tree on it and the development tools contained in the comp33.tgz fileset. Both are available on the official OpenBSD CD.

OpenBSD patches are listed on the errata webpage: http://openbsd.org/errata.html - almost two months since OpenBSD 3.3 was released there are no problems identified. A new release is available every six months, so it may very well happen that a system is not patched in a significant way before it is upgraded to a newer operating system. Although there was a flurry of patches for OpenSSH and Apache in the past year, since these are the only two network services available from the outside of this server, hopefully those are the last vulnerabilities found for awhile with those pieces of software.

File Integrity Checking

Note that some configuration file checking is done nightly as part of the default OpenBSD install. The Daily Insecurity report should be emailed to a human for inspection. The simplest way to be sure all system emails are read is to setup a forward file for root (and the administrative account).

File integrity checking is automated via root's crontab. At a minimum aide should be run with the --check option nightly. For this server a script is called from cron which performs an aide --update, and maintains a short history of the results. It

is also responsible for mailing a human in the event of a change. Once the change is verified the new database can be copied into place manually, to stop the alerts from recurring. This is what is added to root's crontab entry:

```
### FI
30      4       *       *       *       /root/bin/fi_01.sh
### reboot emailer
@reboot mail -s "`hostname` rebooted, up at `date`" admin@example.net
```

The integrity checking script is run nightly at 4:30 am, and an email is sent to the administrator whenever the server reboots for any reason. See Appendix H for the aide.conf being used, as there are considerations in it for the way the /root/bin/fi_01.sh script works. The script is shown here:

```
root@www# cat /root/bin/fi_01.sh
#!/bin/sh

if [ ! -z `/usr/bin/diff /var/db/aide.db /root/etc/aide.db` ]
then
  echo "!!! aide.db doesn't match backup !!!"
fi

[ -f /var/db/aide.db.03 ]  && /bin/mv /var/db/aide.db.03  /var/db/aide.db.04
[ -f /var/db/aide.db.02 ]  && /bin/mv /var/db/aide.db.02  /var/db/aide.db.03
[ -f /var/db/aide.db.01 ]  && /bin/mv /var/db/aide.db.01  /var/db/aide.db.02

/usr/local/bin/aide --update --config=/etc/aide.conf | /usr/bin/mail -s "[AIDE report]
host: www" admin@example.net

[ -f /var/db/aide.db.new ] && /bin/chmod 640 /var/db/aide.db.new
[ -f /var/db/aide.db.new ] && /bin/mv /var/db/aide.db.new /var/db/aide.db.01

[ ! -z /var/db/aide.db.01 ] || /bin/echo "no new db created?" | /usr/bin/mail -s
"[AIDE] host: www" admin@example.net
```

It is a simple script to maintain a short history of aide databases, and email an administrator whenever there is a difference noted between nightly runs. Whenever intentional changes are made to the filesystem, an alert is still generated but the database is updated with the following:

```
root@www# cat /var/db/aide.db.01 > /var/db/aide.db
root@www# cat /var/db/aide.db.01 > /root/etc/aide.db
```

Thus the newly updated database is now compared with the filesystem each night, with the intentional changes integrated into it.

Monitoring Logs

It would be preferable that syslog be configured to log to another, dedicated syslog host. That host should not be reachable from the internet, and should contain any monitoring scripts. Logwatch is one package which can be

configured on either host to monitor the system logs for unusual events. [http://www.logwatch.org] Regardless, if the logs are only kept on the webserver they can and will very likely be removed by any intruder wishing to hide their tracks.
An example of a more specific, custom script is included in Appendix L. It is a short perl script that checks /var/log/authlog for any failed logins, or occurrences of a user becoming root with su. It then emails an administrator a report for the host highlighting the events.

System upgrades

As with system patching, system upgrades should be done on a separate server then rolled into production. Specifying a development server upfront has the benefit of a standing backup server in case of hardware failure; preferably two identical machines are purchased at the same time. Another option is to upgrade the hardware each time the Operating System is upgraded. In any case, a non-production machine is formatted and receives a fresh install from an official OpenBSD CD. The system snapshot created above will not be usable, the new operating system will be installed from CD. However, the same steps applied to this release will be mostly unchanged, and a CVS repository (or last minute CD backup of the development server) will contain the unique content for the server. It should be noted that only the two previous OpenBSD releases are supported. This means that security patches will be available for the latest two releases, older systems need to be upgraded to be brought up to par. This means that even if the upgrade plan does not include new hardware purchases each year, the development server should be upgraded and put into production once a year. Part of that plan should also be to put the "old" production server on the internal network, and upgraded as well, becoming the new development server.

Public Notices

It is necessary to be aware of any patches available for the operating system the webserver is running. And with a public Internet server, it is mandatory to be aware of any publicly known vulnerabilities in the software being used. Since Apache is part of the OpenBSD distribution, and the OpenBSD team develops OpenSSH, it is possible to fill the minimum requirement by subscribing to the OpenBSD security-announce mailing list. Subscription instructions are at http://openbsd.org/mail.html. The announce mailing list is also a low volume list, mainly used to announce new releases becoming available and old releases being end of lifed.

Server Verification

Network Scanning

A portscan using nmap from within the internal network should only find the SSH and HTTP ports open. These are both available intentionally. An external host performing the same scan should only be able to connect with the HTTP port. This is the case in part because all unnecessary services were turned off, but helped in big part by the use of PF. Using PF to drop all packets that are not incoming TCP connections on port 80 (and port 22 connections from the internal network) means that a portscan simply gets no reply from the rest of the ports. It

even takes more time to perform a full portscan due to the scanning host waiting for timeouts. And finally, if /var/log/pflog is analyzed regularly, such scans will be evident even amongst the noise of variously infected hosts continually trying to share their infections.

To verify the intention, install and run nmap from another machine on the internal network. Nmap can be installed as a package under OpenBSD, or installed from source available at http://www.insecure.org. Then also attempt to SSH (or telnet to port 22) from another host outside the network. Here is a sample of such a portscan:

```
root@cryptic# /usr/local/bin/nmap -vv -sT -O -P0 192.168.64.66

Starting nmap V. 2.54BETA33 ( www.insecure.org/nmap/ )
Host www.example.net (192.168.64.66) appears to be up ... good.
Initiating Connect() Scan against www.example.net (192.168.64.66)
Adding open port 80/tcp
Adding open port 22/tcp
The Connect() Scan took 750 seconds to scan 1554 ports.
Warning:  OS detection will be MUCH less reliable because we did not find at
least 1 open and 1 closed TCP port
For OSScan assuming that port 22 is open and port 31294 is closed and neither
are firewalled
Interesting ports on www.example.net (192.168.64.66):
(The 1552 ports scanned but not shown below are in state: filtered)
Port       State       Service
22/tcp     open        ssh
80/tcp     open        http
Remote operating system guess: OpenBSD 3.0 SPARC with pf "scrub in all"
feature
OS Fingerprint:
TSeq(Class=TR%IPID=RD%TS=2HZ)
T1(Resp=Y%DF=Y%W=403D%ACK=S++%Flags=AS%Ops=MNWNNT)
T2(Resp=N)
T3(Resp=N)
T4(Resp=N)
T5(Resp=N)
T6(Resp=N)
T7(Resp=N)
PU(Resp=N)

Uptime 450.017 days (since Thu Apr  4 18:46:41 2002)
TCP Sequence Prediction: Class=truly random
                         Difficulty=9999999 (Good luck!)
TCP ISN Seq. Numbers: 7A053726 4C4757EC 280D73E0 A024F9A
IPID Sequence Generation: Randomized

Nmap run completed -- 1 IP address (1 host up) scanned in 756 seconds
```

Note that although this isn't the newest nmap, it did have a hard time guessing the exact parameters of the operating system (such as the architecture). Also, the uptime is much less than reported:
root@www# uptime
8:17PM up 4 hrs, 1 user, load averages: 0.06, 0.09, 0.08

Valid Interactive Sessions
It may sound silly to verify that an interactive session works, but it can cause unnecessary downtime if the administrative accounts configured above had never been tested, and turned out not to work! Armed with the root password access can still be had via the console so all is not lost. It should also be verified that it is possible to login as a web content maintainer, but not using the login of an Apache process.
It is also advised that the protection offered by the chroot be tested via Apache process. This can be done with a short Perl script whose sole purpose is to find configuration files and anything that shouldn't be available from this webserver.
www-len@www$ cat badcgi.cgi

```
#!/usr/bin/perl

use warnings;
use strict;

use CGI;

my $cgi=new CGI;
my $cwd="/";
my @listing;

print $cgi->header;
print $cgi->start_html;

opendir( DIR, "$cwd" ) || die "cannot open dir, $cwd, $!";
while( my $item = readdir(DIR) )
  print <<EOT;
    $item<br />
EOT

close( DIR );

print $cgi->end_html;
```

When called via URL, the script would be run as the Apache user and printout a directory listing as that user. By modifying the $cwd definition directory listings as the webserver are generated. It turns out that setting it to "/" outputs the contents of the chroot directory as expected. Presumably, on a non-chroot

webserver it would show the root directory of the entire system, although it was not tested that way. It was tested using a valid account by calling it from the command line, and in fact the result was a directory listing of the system root.

Webserver Functionality

A straightforward test - load a web browser on a workstation away from the server and see what come up! After configuring DNS locally to hand requests for example.net to the webserver, it did indeed bring up the Apache manual (and not the "PROXY SERVER" test page. Several links were clicked upon to make sure the self-referencing URLs were working properly.

File Integrity Reports

The way the integrity checker script (/root/bin/fi_01.sh) is written, no report is generated when no changes are made. This is intentional, but it is easy to test whether things are running as expected. For example, changing an account password generates a report as expected:

AIDE found differences between database and filesystem!
Start timestamp: 2003-06-25 21:02:31
Summary:
Total number of files=20032,added files=0,removed files=0,changed files=3

Changed files:
changed:/etc
changed:/etc/master.passwd
changed:/etc/spwd.db
Detailed information about changes:

File: /etc
Mtime: old = 2003-06-22 22:15:03, new = 2003-06-25 20:57:21
Ctime: old = 2003-06-22 22:15:03, new = 2003-06-25 20:57:21

File: /etc/master.passwd
Mtime: old = 2003-04-05 12:59:56, new = 2003-06-25 20:57:21
Ctime: old = 2003-04-05 12:59:56, new = 2003-06-25 20:57:21
Inode: old = 39505 , new = 39856
SHA1: old = KGBnhqe07vX9hdvnCyrXadZDrRs= , new =
nvOaaMOCiE1EsmpCAyqR0ddBLGs=
CRC32: old = e24r8Q== , new = efvnIQ==

File: /etc/spwd.db
Mtime: old = 2003-04-05 12:59:56, new = 2003-06-25 20:57:21
Ctime: old = 2003-04-05 12:59:56, new = 2003-06-25 20:57:21
Inode: old = 39860 , new = 39857
SHA1: old = OOFU+DCvZV55YCEJDKjAZDvmQF4= , new =
p9C/lUQEBgyOQKl6krH/7XX7lFo=
CRC32: old = Oh5tXA== , new = lJF8jw==

End timestamp: 2003-06-25 21:05:35

Password Policy
The file /etc/login.conf was changed so that password policy is maintained regardless of how many times a user tries the same password. In other words, the system does not give up and allow a poor password choice after 3 warnings as is the default. This is verified simply by typing **passwd** as the administrator and using a password that consists solely of lower-case letters. Even though the test password was 9 characters long, the same warning came up over and over until I as the user gave up with <ctrl-C> leaving the better password in place. The same warning came up 10 times in a row, satisfying the intent.
New password:
Please don't use an all-lower case password.
Unusual capitalization, control characters or digits are suggested.

Note that on another, default OpenBSD install, the same password was verified and accepted after 3 warnings like the one shown above.

File System Testing
To test the settings in /etc/fstab and the immutable settings on /bin, a cronjob is setup which attempts to touch /bin/test, and then execute a script in /home/charlie. The script is simply a command to echo the text "why it run?". The output of root's cron is then automatically emailed to an administrator. By running this script once each week it is verified that the mount options were not reversed by someone else. These are the relevant lines in root's crontab:
03 03 * * 1 touch /bin/test
03 04 * * 1 /home/charlie/why.sh

The results are emailed automatically, but can be shown using the following results from the command line, run as root:
root@www# touch /bin/test
touch: /bin/test: Operation not permitted

root@www# /home/charlie/why.sh
sh: /home/charlie/why.sh: Permission denied
root@www# ls -al /home/charlie/why.sh
-rwxr-xr-x 1 root charlie 29 Jun 28 21:14 /home/charlie/why.sh

Performance Evaluation
When testing the website from a workstation on the 100 Mbps internally connected network, there was no noticeable lag. It may be that under heavier load it does not perform as well as a single Apache instance on the same server, due to the overhead of proxying each request, but this was not observed in practice using the timeframe human reaction time as a guide.

References

1. Birznieks, Gunther, “Wrapping CGI Scripts”
URL: http://www.extropia.com/presentations/birznieks/pdf/cgi_dev_security.pdf

2. OpenBSD Team, “Manual Pages”
URL: http://www.openbsd.org/cgi-bin/man.cgi

3. U.S. Department of Energy, “Creating Login Banners”
URL: http://www.ciac.org/ciac/bulletins/j-043.shtml

4. kc@nlanr.net, “Assessing Average Hop Count of a Wide Area Internet Packet”
URL: http://www.nlanr.net/NA/Learn/wingspan.html

5. Shaffer, George, “Hardening OpenBSD Internet Servers”
URL: http://BSD.GeodSoft.com/howto/harden

Appendices

Appendix A – mount_null usage

```
root@www# cd /var/www/example.net/
root@www# mkdir –p usr
root@www# mount_null /var/www/usr usr
```

Appendix B – dmesg

```
OpenBSD 3.3 (GENERIC) #44: Sat Mar 29 13:22:05 MST 2003
    deraadt@i386.openbsd.org:/usr/src/sys/arch/i386/compile/GENERIC
cpu0: AMD Athlon Model 4 (Thunderbird) ("AuthenticAMD" 686-class) 1 GHz
cpu0:
FPU,V86,DE,PSE,TSC,MSR,PAE,MCE,CX8,SYS,MTRR,PGE,MCA,CMOV,PAT,PSE36,MMX,
FXSR
real mem  = 267956224 (261676K)
avail mem = 242634752 (236948K)
using 3296 buffers containing 13500416 bytes (13184K) of memory
mainbus0 (root)
bios0 at mainbus0: AT/286+(00) BIOS, date 09/20/01, BIOS32 rev. 0 @
0xfdae0
apm0 at bios0: Power Management spec V1.2
apm0: AC on, no battery
pcibios0 at bios0: rev. 2.1 @ 0xf0000/0x10000
pcibios0: PCI IRQ Routing Table rev. 1.0 @ 0xf7810/160 (8 entries)
pcibios0: PCI Interrupt Router at 000:02:0 ("SIS 85C503 PCI System I/O
Chipset" rev 0x00)
pcibios0: PCI bus #1 is the last bus
bios0: ROM list: 0xc0000/0xa800 0xca800/0x8000
pci0 at mainbus0 bus 0: configuration mode 1 (no bios)
pchb0 at pci0 dev 0 function 0 "SIS 735 Host-PCI" rev 0x01
ppb0 at pci0 dev 1 function 0 "SIS 86C201 Host-AGP" rev 0x00
pci1 at ppb0 bus 1
pcib0 at pci0 dev 2 function 0 "SIS 85C503 PCI System I/O Chipset" rev
0x00
pciide0 at pci0 dev 2 function 5 "SIS 5513 EIDE" rev 0xd0: DMA, channel
0 wired to compatibility, channel 1 wired to compatibility
wd0 at pciide0 channel 0 drive 0: <ST330620A>
wd0: 16-sector PIO, LBA, 28629MB, 16383 cyl, 16 head, 63 sec, 58633344
sectors
wd1 at pciide0 channel 0 drive 1: <ST32132A>
wd1: 16-sector PIO, LBA, 2015MB, 4095 cyl, 16 head, 63 sec, 4127760
sectors
wd0(pciide0:0:0): using PIO mode 4, Ultra-DMA mode 2
wd1(pciide0:0:1): using PIO mode 4, DMA mode 2
atapiscsi0 at pciide0 channel 1 drive 0
scsibus0 at atapiscsi0: 2 targets
cd0 at scsibus0 targ 0 lun 0: <SONY, CD-R CDU928E, 1.1e> SCSI0 5/cdrom
removable
cd0(pciide0:1:0): using PIO mode 3, DMA mode 1
sis0 at pci0 dev 3 function 0 "SIS 900 10/100BaseTX" rev 0x90: irq 11
address 00:07:95:0a:70:91
ukphy0 at sis0 phy 1: Generic IEEE 802.3u media interface
ukphy0: OUI 0x000020, model 0x0020, rev. 1
```

```
vga1 at pci0 dev 9 function 0 "Nvidia Vanta" rev 0x15
wsdisplay0 at vga1: console (80x25, vt100 emulation)
wsdisplay0: screen 1-5 added (80x25, vt100 emulation)
rl0 at pci0 dev 11 function 0 "Realtek 8139" rev 0x10: irq 12 address
00:50:fc:56:f9:a1
rlphy0 at rl0 phy 0: RTL internal phy
emu0 at pci0 dev 17 function 0 "Creative Labs SoundBlaster Live" rev
0x08: irq 12
ac97: codec id 0x83847608 (SigmaTel STAC9708/11)
ac97: codec features 18 bit DAC, 18 bit ADC, SigmaTel 3D
audio0 at emu0
"Creative Labs SoundBlaster Live Digital Input" rev 0x08 at pci0 dev 17
function 1 not configured
isa0 at pcib0
isadma0 at isa0
pckbc0 at isa0 port 0x60/5
pckbd0 at pckbc0 (kbd slot)
pckbc0: using irq 1 for kbd slot
wskbd0 at pckbd0: console keyboard, using wsdisplay0
pcppi0 at isa0 port 0x61
midi0 at pcppi0: <PC speaker>
sysbeep0 at pcppi0
lpt0 at isa0 port 0x378/4 irq 7
npx0 at isa0 port 0xf0/16: using exception 16
pccom0 at isa0 port 0x3f8/8 irq 4: ns16550a, 16 byte fifo
pccom1 at isa0 port 0x2f8/8 irq 3: ns16550a, 16 byte fifo
fdc0 at isa0 port 0x3f0/6 irq 6 drq 2
fd0 at fdc0 drive 0: 1.44MB 80 cyl, 2 head, 18 sec
biomask c040 netmask d840 ttymask d8c2
pctr: user-level cycle counter enabled
mtrr: Pentium Pro MTRR support
dkcsum: wd0 matched BIOS disk 80
dkcsum: wd1 matched BIOS disk 81
root on wd0a
rootdev=0x0 rrootdev=0x300 rawdev=0x302
```

Appendix C - /etc/banner

```
$ cat /etc/banner

THIS SYSTEM IS FOR THE USE OF AUTHORIZED USERS ONLY.

       Individuals using this computer system without authority, or in
       excess of their authority, are subject to having all of their
       activities on this system monitored and recorded by system
       personnel. UNAUTHORIZED access to this system will be tracked
       and logged.  IF YOU HAVE ACCESSED THIS SYSTEM WITHOUT PROPER
       AUTHORITY - DISCONNECT NOW.

       In the course of monitoring individuals improperly using this
       system, or in the course of system maintenance, the activities
       of authorized users may also be monitored.

       Anyone using this system expressly consents to such monitoring
       and is advised that if such monitoring reveals possible
       evidence of criminal activity, system personnel may provide the
       evidence of such monitoring to law enforcement officials.
```

Appendix D - /etc/login.conf

```
root@www# cat /etc/login.conf
# Default authentication methods
auth-defaults:auth=passwd,skey:

# Default authentication methods for ftp
auth-ftp-defaults:auth-ftp=passwd:

#
# The default values
# To alter the default authentication types change the line:
#       :tc=auth-defaults:\
# to be read something like: (enables passwd, "myauth", and activ)
#       :auth=passwd,myauth,activ:\
# Any value changed in the daemon class should be reset in default
# class.
#
default:\
        :path=/usr/bin /bin /usr/sbin /sbin /usr/X11R6/bin
/usr/local/bin:\
        :umask=022:\
        :datasize-max=256M:\
        :datasize-cur=64M:\
        :maxproc-max=128:\
        :maxproc-cur=64:\
        :openfiles-cur=64:\
        :stacksize-cur=4M:\
        :localcipher=blowfish,6:\
        :ypcipher=old:\
        :tc=auth-defaults:\
        :tc=auth-ftp-defaults:\
        :login-backoff=1:\
        :login-tries=3:\
        :passwordtries=0:\
        :filesize=512MB:\
        :requirehome=true:\
        :coredumpsize=0:

#
# Settings used by /etc/rc and root
# This must be set properly for daemons started as root by inetd as
well.
# Be sure reset these values back to system defaults in the default
class!
#
daemon:\
        :ignorenologin:\
        :datasize=infinity:\
        :maxproc=infinity:\
        :openfiles-cur=128:\
        :stacksize-cur=8M:\
        :localcipher=blowfish,8:\
        :tc=default:\
        :coredumpsize=0:
```

```
#
# Staff have fewer restrictions and can login even when nologins are
set.
#
staff:\
        :datasize-cur=64M:\
        :datasize-max=infinity:\
        :maxproc-max=256:\
        :maxproc-cur=128:\
        :ignorenologin:\
        :requirehome@:\
        :tc=default:
```

Appendix E - /etc/ssh/sshd_config

```
root@www# cat /etc/ssh/sshd_config
#       $OpenBSD: sshd_config,v 1.59 2002/09/25 11:17:16 markus Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options change a
# default value.

Port 22
Protocol 2
#ListenAddress 0.0.0.0
#ListenAddress ::
ListenAddress 192.168.64.66

# HostKey for protocol version 1
#HostKey /etc/ssh/ssh_host_key
# HostKeys for protocol version 2
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_dsa_key

# Lifetime and size of ephemeral version 1 server key
#KeyRegenerationInterval 3600
#ServerKeyBits 1024

# Logging
#obsoletes QuietMode and FascistLogging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 120
PermitRootLogin no
#StrictModes yes

#RSAAuthentication yes
#PubkeyAuthentication yes
#AuthorizedKeysFile     .ssh/authorized_keys
```

```
# rhosts authentication should not be used
#RhostsAuthentication no
# Don't read the user's ~/.rhosts and ~/.shosts files
#IgnoreRhosts yes
# For this to work you will also need host keys in
/etc/ssh/ssh_known_hosts
#RhostsRSAAuthentication no
# similar for protocol version 2
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# RhostsRSAAuthentication and HostbasedAuthentication
#IgnoreUserKnownHosts no

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes

#AFSTokenPassing no

# Kerberos TGT Passing only works with the AFS kaserver
#KerberosTgtPassing no

#X11Forwarding no
#X11DisplayOffset 10
#X11UseLocalhost yes
#PrintMotd yes
#PrintLastLog yes
#KeepAlive yes
#UseLogin no
#UsePrivilegeSeparation yes
#PermitUserEnvironment no
#Compression yes

#MaxStartups 10
# no default banner path
Banner /etc/banner
#VerifyReverseMapping no

# override default of no subsystems
Subsystem       sftp    /usr/libexec/sftp-server
```

Appendix F - /etc/fstab

```
root@www# cat /etc/fstab
/dev/wd0a / ffs rw,softdep 1 1
/dev/wd0i /home ffs rw,nodev,nosuid,noexec,softdep 1 2
/dev/wd0d /tmp ffs rw,nodev,nosuid,noexec,softdep 1 2
```

```
/dev/wd0h /usr ffs rw,nodev,rdonly,softdep 1 2
/dev/wd0f /var ffs rw,nodev,nosuid,noexec,softdep 1 2
/dev/wd0g /var/www ffs rw,nodev,nosuid,softdep 1 2
```

Appendix G - /etc/pf.conf

```
root@www# cat /etc/pf.conf
ext_if="rl0"    # replace with actual external interface name i.e., dc0
tcp_flags="flags S/SAFR"
loghost="192.168.64.3"
table <mgr> { 192.168.64.0/24, 10.123.1.6/32 }

scrub in all

pass in quick on lo0 all
# don't slow down the proxy to apache instance traffic
pass in quick on lo1 all

block in log all

pass  in  on $ext_if proto tcp from <mgr> to $ext_if port 22 keep state
$tcp_flags
pass in on $ext_if proto tcp from any to $ext_if port 80 keep state
$tcp_flags

pass out on $ext_if proto udp from any to $loghost port 514 keep state
```

Appendix H - /etc/aide.conf

```
#
# AIDE 0.7
#
# example configuration file

@@ifndef TOPDIR
@@define TOPDIR /
@@endif

@@ifdef DEBUG
@@define DEBUG ison
@@undef NOT_DEBUG
@@else
@@define NOT_DEBUG true
@@undef DEBUG
@@endif

@@ifhost korppi
@@define KORPPI yes
@@endif

@@ifnhost ftp
@@define BUMMER true
@@endif

# The location of the database to be read.
#database=file:aide.db
database=file:/var/db/aide.db
```

```
# The location of the database to be written.
#database_out=sql:host:port:database:login_name:passwd:table
database_out=file:/var/db/aide.db.new

# Whether to gzip the output to database
# gzip_dbout=no

#verbose=5
verbose=20

#report_url=stdout
#other possibilities
#report_url=stderr
#NOT IMPLEMENTED report_url=mailto:root@foo.com
#report_url=file:/tmp/some_file.txt
#NOT IMPLEMENTED report_url=syslog:LOG_AUTH
report_url=stdout

#p:     permissions
#i:     inode
#n:     number of links
#u:     user
#g:     group
#s:     size
#b:     block count
#m:     mtime
#a:     atime
#c:     ctime
#S:     check for growing size
#md5:   md5 checksum
#sha1:  sha1 checksum
#rmd160:     rmd160 checksum
#tiger:     tiger checksum
#R:     p+i+n+u+g+s+m+c+md5
#L:     p+i+n+u+g
#E:     Empty group
#>:     Growing logfile p+u+g+i+n+S
#The following are available if you have mhash support enabled.
#haval:         haval checksum
#gost:          gost checksum
#crc32:         crc32 checksum

# Rule definition
All=R+a+sha1+rmd160+tiger

# Personal definitions
Strict=p+i+n+u+g+s+b+c+m+sha1+crc32
Normal=p+i+n+u+g+s+c+m+md5
Aidedb=p+i+n+u+g+b
Device=i+n+b
Exists=p+u+g


# ignore_list is a special rule definition
# the attributes listed in it are not displayed in the
# final report
```

```
# Attributes that can be used to verify that aide in intact
# by people that have downloaded it from the web.
# Let's be paranoid
#Norm=s+n+b+md5+sha1+rmd160+tiger

/$                      Normal
=/bsd$                  Strict
=/boot$                 Strict

/bin                    Strict

/dev                    Normal
=/dev/bpf0$             Device
=/dev/null$             Device
=/dev/pty[pqr][0-9a-f]$ Device+p+u+g
=/dev/tty$              Device+p+u+g
=/dev/tty[pqr][0-9a-f]$ Device

/etc                    Normal
=/etc/aide.conf$        Strict
=/etc/daily$            Strict
=/etc/group$            Strict
=/etc/master.passwd$    Strict
=/etc/monthly$          Strict
=/etc/passwd$           Strict
=/etc/resolv.conf$      Exists
=/etc/rc                Strict
=/etc/spwd.db$          Strict
=/etc/ssh               Strict
=/etc/ssl               Strict
=/etc/sudoers$          Strict
=/etc/systrace          Strict
=/etc/weekly            Strict

=/home$                 Normal

/root                   Normal
=/root/etc/aide.db$     Aidedb

/sbin                   Strict

!/sys$

/tmp                    Exists

/usr                    Normal
/usr/bin                Strict
/usr/lib                Strict
/usr/libexec            Strict
/usr/local/bin          Strict
/usr/local/lib          Strict
/usr/local/libexec      Strict
=/usr/local/man         Exists
/usr/local/sbin         Strict
!/usr/local/share
!/usr/obj/
```

```
!/usr/ports/
/usr/sbin               Strict
!/usr/share
!/usr/src

/var                    Normal
=/var/backups/          Exists
=/var/cron$             Exists
=/var/cron/log          Exists
=/var/db                Exists
=/var/db/aide.db$       Aidedb
!/var/db/aide.db.0[0-4]$
=/var/empty             Strict
=/var/log$              Exists
=/var/log/              Exists
=/var/msgs/             Exists
=/var/mail/             Exists+i+c
/var/run                Exists
!/var/spool/clientmqueue
!/var/spool/mqueue
=/var/tmp               Exists
!/var/tmp/vi.recover
/var/www/example.net/cgi-bin        Strict
/var/www/login.example.net/cgi-bin  Strict
/var/www/proxy/conf/    Strict
/var/www/example.net/conf/          Strict
/var/www/login.example.net/conf/    Strict
/var/www/logs/          Exists
```

Appendix I – /var/www/proxy/conf/httpd.conf

```
root@www# cat /var/www/proxy/conf/httpd.conf
#       $Id$
#

### Section 1: Global Environment
ServerType standalone

# Do NOT add a slash at the end of the directory path.
ServerRoot "/var/www/proxy"

#LockFile logs/accept.lock
PidFile logs/httpd.pid
ScoreBoardFile logs/apache_runtime_status

Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

MinSpareServers 5
MaxSpareServers 10
StartServers 5
MaxClients 150
MaxRequestsPerChild 0
```

```
BindAddress 192.168.64.66

# Dynamic Shared Object (DSO) Support
# Note: The order is which modules are loaded is important.  Don't
change
# the order below without expert advice.
LoadModule proxy_module /usr/lib/apache/modules/libproxy.so

#ExtendedStatus On



### Section 2: 'Main' server configuration
Port 80

##  SSL Support
<IfDefine SSL>
Listen 80
Listen 443
</IfDefine>

# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
User www-proxy
Group www-proxy

ServerAdmin webmaster@example.net
ServerName proxy.example.net
DocumentRoot "/var/www/proxy/htdocs"

# First, we configure the "default" to be a very restrictive set of
# permissions.
<Directory />
    Options FollowSymLinks
    AllowOverride None

    Order deny,allow
    Deny from all
</Directory>


#CacheNegotiatedDocs
UseCanonicalName On

TypesConfig conf/mime.types
DefaultType text/plain

<IfModule mod_mime_magic.c>
    MIMEMagicFile conf/magic
</IfModule>

HostnameLookups Off
ErrorLog logs/error_log
LogLevel warn

LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
Agent}i\"" combined
```

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

#CustomLog logs/access_log common
#CustomLog logs/referer_log referer
#CustomLog logs/agent_log agent
CustomLog logs/access_log combined

ServerSignature Off



###
# Proxy Server directives. Uncomment the following lines to
# enable the proxy server:
#
<IfModule mod_proxy.c>
ProxyRequests On

<Directory proxy:*>
    Order deny,allow
    Deny from all
#    Allow from .your_domain.com
</Directory>

<Directory proxy:http://example.net/>
   Order deny,allow
   Allow from all
</Directory>

<Directory proxy:http://www.example.net/>
   Order deny,allow
   Allow from all
</Directory>


# Enable/disable the handling of HTTP/1.1 "Via:" headers.
# ("Full" adds the server version; "Block" removes all outgoing Via:
headers)
# Set to one of: Off | On | Full | Block
ProxyVia On

#
# To enable the cache as well, edit and uncomment the following lines:
# (no cacheing without CacheRoot)
#
#CacheRoot "/var/www/proxy/cache"
#CacheSize 5
#CacheGcInterval 4
#CacheMaxExpire 24
#CacheLastModifiedFactor 0.1
#CacheDefaultExpire 1
#NoCache a_domain.com another_domain.edu joes.garage_sale.com

</IfModule>
# End of proxy directives.
```

```
###


# IndexIgnore is a set of filenames which directory indexing should
ignore
# and not include in the listing.  Shell-style wildcarding is
permitted.
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t

#
# AddEncoding allows you to have certain browsers (Mosaic/X 2.1+)
uncompress
# information on the fly. Note: Not all browsers support this.
# Despite the name similarity, the following Add* directives have
nothing
# to do with the FancyIndexing customization directives above.
#
AddEncoding x-compress Z
AddEncoding x-gzip gz


#
# Customizable error response (Apache style)
#  these come in three flavors
#
#    1) plain text
#ErrorDocument 500 "The server made a boo boo.
#  n.b.  the (") marks it as text, it does not get output
#
#    2) local redirects
#ErrorDocument 404 /missing.html
#  to redirect to local URL /missing.html
#ErrorDocument 404 /cgi-bin/missing_handler.pl
#  N.B.: You can redirect to a script or a document using server-side-
includes.
#
#    3) external redirects
#ErrorDocument 402 http://some.other_server.com/subscription_info.html
#  N.B.: Many of the environment variables associated with the original
#  request will *not* be available to such a script.

# Built-in Broken Browser Tweaks
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-
1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0



### Section 3: Virtual Hosts
# If you want to use name-based virtual hosts you need to define at
# least one IP address (and port number) for them.
#NameVirtualHost 12.34.56.78:80
#NameVirtualHost 12.34.56.78
```

```
NameVirtualHost 192.168.64.66

<VirtualHost 192.168.64.66>
   ServerName example.net
   ProxyPass / http://172.16.0.2/
   ProxyPassReverse / http://172.16.0.2/
#   CustomLog logs/172.16.0.2.access_log combined

   <Location />
      Order allow,deny
      Allow from all
   </Location>
</VirtualHost>


##  SSL Global Context
##  All SSL configuration in this context applies both to
##  the main server and all SSL-enabled virtual hosts.

#   Some MIME-types for downloading Certificates and CRLs
<IfDefine SSL>
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl
</IfDefine>


<IfModule mod_ssl.c>
SSLPassPhraseDialog  builtin
SSLSessionCache         dbm:logs/ssl_scache
SSLSessionCacheTimeout  300
SSLMutex  sem

#   Pseudo Random Number Generator (PRNG):
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
#SSLRandomSeed startup file:/dev/random  512
#SSLRandomSeed startup file:/dev/urandom 512
#SSLRandomSeed connect file:/dev/random  512
#SSLRandomSeed connect file:/dev/urandom 512
SSLRandomSeed startup file:/dev/arandom  512

#   Logging:
SSLLog      logs/ssl_engine_log
SSLLogLevel info

</IfModule>
```

Appendix J - /var/www/template/conf/httpd.conf

```
root@www# cat /var/www/template/conf/httpd.conf
#       $Id$
#
# Based upon the NCSA server configuration files originally by Rob
McCool.
#
```

```
### Section 1: Global Environment
ServerType standalone

# Do NOT add a slash at the end of the directory path.
ServerRoot "/var/www/_TEMPLATE_"

#LockFile logs/accept.lock

PidFile logs/httpd.pid

ScoreBoardFile logs/apache_runtime_status


# Timeout: The number of seconds before receives and sends time out.
Timeout 300

# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
KeepAlive On

# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited
amount.
# We recommend you leave this number high, for maximum performance.
MaxKeepAliveRequests 100

# KeepAliveTimeout: Number of seconds to wait for the next request from
the
# same client on the same connection.
KeepAliveTimeout 15

# Server-pool size regulation.  Rather than making you guess how many
# server processes you need, Apache dynamically adapts to the load it
# sees --- that is, it tries to maintain enough server processes to
# handle the current load, plus a few spare servers to handle transient
# load spikes (e.g., multiple simultaneous requests from a single
# Netscape browser).
MinSpareServers 5
MaxSpareServers 10
StartServers 5
MaxClients 150

# MaxRequestsPerChild: the number of requests each child process is
# allowed to process before the child dies.  The child will exit so
# as to avoid problems after prolonged use when Apache (and maybe the
# libraries it uses) leak memory or other resources.  On most systems,
this
# isn't really needed, but a few (such as Solaris) do have notable
leaks
# in the libraries.
MaxRequestsPerChild 100

BindAddress 172.16.0.CHANGEME



### Section 2: 'Main' server configuration
```

```
Port 80
User www-CHANGEME
Group www-CHANGEME
ServerAdmin webmaster@example.net
#ServerName new.host.name
DocumentRoot "/var/www/_TEMPLATE_/htdocs"

# First, we configure the "default" to be a very restrictive set of
# permissions.
<Directory />
    Options None
    AllowOverride None

    Order deny,allow
    Deny from all
</Directory>

# This should be changed to whatever you set DocumentRoot to.
<Directory "/var/www/_TEMPLATE_/htdocs">
#
# This may also be "None", "All", or any combination of "Indexes",
# "Includes", "FollowSymLinks", "ExecCGI", or "MultiViews".
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
    Options None

#
# This controls which options the .htaccess files in directories can
# override. Can also be "All", or any combination of "Options",
"FileInfo",
# "AuthConfig", and "Limit"
#
    AllowOverride None

#
# Controls who can get stuff from this server.
#
    <Limit GET PUT>
        Order allow,deny
        Allow from all
    </Limit>

    <Limit OPTIONS PROPFIND PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE
LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>



UserDir disabled


DirectoryIndex index.html
```

```
AccessFileName .htaccess
<Files .htaccess>
    Order allow,deny
    Deny from all
</Files>

#CacheNegotiatedDocs
UseCanonicalName On
TypesConfig conf/mime.types
DefaultType text/plain

<IfModule mod_mime_magic.c>
    MIMEMagicFile conf/magic
</IfModule>

HostnameLookups Off
ErrorLog logs/error_log

# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn

# The following directives define some format nicknames for use with
# a CustomLog directive (see below).
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# The location and format of the access logfile (Common Logfile
Format).
# If you do not define any access logfiles within a <VirtualHost>
# container, they will be logged here.  Contrariwise, if you *do*
# define per-<VirtualHost> access logfiles, transactions will be
# logged therein and *not* in this file.
#CustomLog logs/access_log common
#CustomLog logs/referer_log referer
#CustomLog logs/agent_log agent
#
# If you prefer a single logfile with access, agent, and referer
information
# (Combined Logfile Format) you can use the following directive.
CustomLog logs/access_log combined

ServerSignature Off

# Note that if you include a trailing / on fakename then the server
will
# require it to be present in the URL.  So "/icons" isn't aliased in
this
# example, only "/icons/"..
Alias /icons/ "/var/www/example.net/icons/"

<Directory "/var/www/icons">
    Options Indexes MultiViews
```

```
    AllowOverride None
    <Limit GET>
        Order allow,deny
        Allow from all
    </Limit>
    <Limit PUT OPTIONS PROPFIND PUT DELETE PATCH PROPPATCH MKCOL COPY
MOVE LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>

<Directory "/var/www/_TEMPLATE_/htdocs/manual">
    Options MultiViews
    AllowOverride None
    <Limit GET>
        Order allow,deny
        Allow from all
    </Limit>
    <Limit PUT OPTIONS PROPFIND PUT DELETE PATCH PROPPATCH MKCOL COPY
MOVE LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>


# ScriptAlias: This controls which directories contain server scripts.
# ScriptAliases are essentially the same as Aliases, except that
# documents in the realname directory are treated as applications and
# run by the server when requested rather than as documents sent to the
client.
# The same rules about trailing "/" apply to ScriptAlias directives as
to
# Alias.
ScriptAlias /cgi-bin/ "/var/www/_TEMPLATE_/cgi-bin/"

# "/var/www/cgi-bin" should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
<Directory "/var/www/_TEMPLATE_/cgi-bin">
    AllowOverride None
    Options None
    <Limit GET PUT>
        Order allow,deny
        Allow from all
    </Limit>
    <Limit OPTIONS PROPFIND PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE
LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>


# Redirect allows you to tell clients about documents which used to
exist in
```

```
# your server's namespace, but do not anymore. This allows you to tell
the
# clients where to look for the relocated document.
# Format: Redirect old-URI new-URL
#

# FancyIndexing is whether you want fancy directory indexing or
standard
IndexOptions FancyIndexing

# AddIcon* directives tell the server which icon to show for different
# files or filename extensions.  These are only displayed for
# FancyIndexed directories.
#
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*

AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core

AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^

# DefaultIcon is which icon to show for files which do not have an icon
# explicitly set.
DefaultIcon /icons/unknown.gif

# AddDescription allows you to place a short description after a file
in
# server-generated indexes.  These are only displayed for FancyIndexed
# directories.
# Format: AddDescription "description" filename
#AddDescription "GZIP compressed document" .gz
#AddDescription "tar archive" .tar
#AddDescription "GZIP compressed tar archive" .tgz

# ReadmeName is the name of the README file the server will look for by
```

```
# default, and append to directory listings.
#
# HeaderName is the name of a file which should be prepended to
# directory indexes.
#
# The server will first look for name.html and include it if found.
# If name.html doesn't exist, the server will then look for name.txt
# and include it as plaintext if found.
#
ReadmeName README
HeaderName HEADER

#
# IndexIgnore is a set of filenames which directory indexing should
ignore
# and not include in the listing.  Shell-style wildcarding is
permitted.
#
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t

#
# AddEncoding allows you to have certain browsers (Mosaic/X 2.1+)
uncompress
# information on the fly. Note: Not all browsers support this.
# Despite the name similarity, the following Add* directives have
nothing
# to do with the FancyIndexing customization directives above.
#
AddEncoding x-compress Z
AddEncoding x-gzip gz

#
# AddLanguage allows you to specify the language of a document. You can
# then use content negotiation to give a browser a file in a language
# it can understand.  Note that the suffix does not have to be the same
# as the language keyword --- those with documents in Polish (whose
# net-standard language code is pl) may wish to use "AddLanguage pl
.po"
# to avoid the ambiguity with the common suffix for perl scripts.
#
AddLanguage en .en
AddLanguage fr .fr
AddLanguage de .de
AddLanguage da .da
AddLanguage el .el
AddLanguage it .it

#
# LanguagePriority allows you to give precedence to some languages
# in case of a tie during content negotiation.
# Just list the languages in decreasing order of preference.
#
LanguagePriority en fr de

#
# AddType allows you to tweak mime.types without actually editing it,
or to
```

```
# make certain files to be certain types.
#
# For example, the PHP module (not part of the Apache distribution)
# will typically use:
#
#AddType application/x-httpd-php .php

#
# AddHandler allows you to map certain file extensions to "handlers",
# actions unrelated to filetype. These can be either built into the
server
# or added with the Action command (see below)
#
# If you want to use server side includes, or CGI outside
# ScriptAliased directories, uncomment the following lines.
#
# To use CGI scripts:
#
#AddHandler cgi-script .cgi

#
# To use server-parsed HTML files
#
#AddType text/html .shtml
#AddHandler server-parsed .shtml

#
# Uncomment the following line to enable Apache's send-asis HTTP file
# feature
#
#AddHandler send-as-is asis

#
# If you wish to use server-parsed imagemap files, use
#
#AddHandler imap-file map

#
# To enable type maps, you might want to use
#
#AddHandler type-map var

#
# Action lets you define media types that will execute a script
whenever
# a matching file is called. This eliminates the need for repeated URL
# pathnames for oft-used CGI file processors.
# Format: Action media/type /cgi-script/location
# Format: Action handler-name /cgi-script/location
#

#
# MetaDir: specifies the name of the directory in which Apache can find
# meta information files. These files contain additional HTTP headers
# to include when sending the document
#
#MetaDir .web
```

```
#
# MetaSuffix: specifies the file name suffix for the file containing
the
# meta information.
#
#MetaSuffix .meta

#
# Customizable error response (Apache style)
#  these come in three flavors
#
#    1) plain text
#ErrorDocument 500 "The server made a boo boo.
#  n.b.  the (") marks it as text, it does not get output
#
#    2) local redirects
#ErrorDocument 404 /missing.html
#  to redirect to local URL /missing.html
#ErrorDocument 404 /cgi-bin/missing_handler.pl
#  N.B.: You can redirect to a script or a document using server-side-
includes.
#
#    3) external redirects
#ErrorDocument 402 http://some.other_server.com/subscription_info.html
#  N.B.: Many of the environment variables associated with the original
#  request will *not* be available to such a script.

#
# The following directives modify normal HTTP response behavior.
# The first directive disables keepalive for Netscape 2.x and browsers
that
# spoof it. There are known problems with these browser
implementations.
# The second directive is for Microsoft Internet Explorer 4.0b2
# which has a broken HTTP/1.1 implementation and does not properly
# support keepalive when it is used on 301 or 302 (redirect) responses.
#
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-
1.0

#
# The following directive disables HTTP/1.1 responses to browsers which
# are in violation of the HTTP/1.0 spec by not being able to grok a
# basic 1.1 response.
#
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0

#
# There have been reports of people trying to abuse an old bug from
pre-1.1
# days.  This bug involved a CGI script distributed as a part of
Apache.
```

```
# By uncommenting these lines you can redirect these attacks to a
logging
# script on phf.apache.org.  Or, you can record them yourself, using
the script
# support/phf_abuse_log.cgi.
#
#<Location /cgi-bin/phf*>
#    Deny from all
#    ErrorDocument 403 http://phf.apache.org/phf_abuse_log.cgi
#</Location>



### Section 3: Virtual Hosts
#
# VirtualHost: If you want to maintain multiple domains/hostnames on
your
# machine you can setup VirtualHost containers for them.
# Please see the documentation at
<URL:http://www.apache.org/docs/vhosts/>
# for further details before you try to setup virtual hosts.
# You may use the command line option '-S' to verify your virtual host
# configuration.

# If you want to use name-based virtual hosts you need to define at
# least one IP address (and port number) for them.
#NameVirtualHost 12.34.56.78:80
#NameVirtualHost 12.34.56.78

# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
#
#<VirtualHost ip.address.of.host.some_domain.com>
#    ServerAdmin webmaster@host.some_domain.com
#    DocumentRoot /www/docs/host.some_domain.com
#    ServerName host.some_domain.com
#    ErrorLog logs/host.some_domain.com-error_log
#    CustomLog logs/host.some_domain.com-access_log common
#</VirtualHost>

#<VirtualHost _default_:*>
#</VirtualHost>
```

Appendix K - /var/www/example.net/httpd.conf

```
root@www# cat /var/www/example.net/conf/httpd.conf
#       $Id$
#
# Based upon the NCSA server configuration files originally by Rob
McCool.
#

### Section 1: Global Environment
ServerType standalone

# Do NOT add a slash at the end of the directory path.
```

```
ServerRoot "/var/www/example.net"

#LockFile logs/accept.lock

PidFile logs/httpd.pid

ScoreBoardFile logs/apache_runtime_status


# Timeout: The number of seconds before receives and sends time out.
Timeout 300

# KeepAlive: Whether or not to allow persistent connections (more than
# one request per connection). Set to "Off" to deactivate.
KeepAlive On

# MaxKeepAliveRequests: The maximum number of requests to allow
# during a persistent connection. Set to 0 to allow an unlimited
amount.
# We recommend you leave this number high, for maximum performance.
MaxKeepAliveRequests 100

# KeepAliveTimeout: Number of seconds to wait for the next request from
the
# same client on the same connection.
KeepAliveTimeout 15

# Server-pool size regulation.  Rather than making you guess how many
# server processes you need, Apache dynamically adapts to the load it
# sees --- that is, it tries to maintain enough server processes to
# handle the current load, plus a few spare servers to handle transient
# load spikes (e.g., multiple simultaneous requests from a single
# Netscape browser).
MinSpareServers 5
MaxSpareServers 10
StartServers 5
MaxClients 150

# MaxRequestsPerChild: the number of requests each child process is
# allowed to process before the child dies.  The child will exit so
# as to avoid problems after prolonged use when Apache (and maybe the
# libraries it uses) leak memory or other resources.  On most systems,
this
# isn't really needed, but a few (such as Solaris) do have notable
leaks
# in the libraries.
MaxRequestsPerChild 100

BindAddress 172.16.0.2



### Section 2: 'Main' server configuration
Port 80
User www-en
Group www-en
ServerAdmin admin@www.example.net
```

```
#ServerName new.host.name
DocumentRoot "/var/www/example.net/htdocs"

# First, we configure the "default" to be a very restrictive set of
# permissions.
<Directory />
    Options None
    AllowOverride None

    Order deny,allow
    Deny from all
</Directory>

# This should be changed to whatever you set DocumentRoot to.
<Directory "/var/www/example.net/htdocs">
#
# This may also be "None", "All", or any combination of "Indexes",
# "Includes", "FollowSymLinks", "ExecCGI", or "MultiViews".
#
# Note that "MultiViews" must be named *explicitly* --- "Options All"
# doesn't give it to you.
#
    Options None

#
# This controls which options the .htaccess files in directories can
# override. Can also be "All", or any combination of "Options",
"FileInfo",
# "AuthConfig", and "Limit"
#
    AllowOverride None

#
# Controls who can get stuff from this server.
#
    <Limit GET PUT>
        Order allow,deny
        Allow from all
    </Limit>

    <Limit OPTIONS PROPFIND PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE
LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>



UserDir disabled


DirectoryIndex index.html
AccessFileName .htaccess
<Files .htaccess>
    Order allow,deny
    Deny from all
```

```
</Files>

#CacheNegotiatedDocs
UseCanonicalName On
TypesConfig conf/mime.types
DefaultType text/plain

<IfModule mod_mime_magic.c>
    MIMEMagicFile conf/magic
</IfModule>

HostnameLookups Off
ErrorLog logs/error_log

# LogLevel: Control the number of messages logged to the error_log.
# Possible values include: debug, info, notice, warn, error, crit,
# alert, emerg.
LogLevel warn

# The following directives define some format nicknames for use with
# a CustomLog directive (see below).
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-
Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# The location and format of the access logfile (Common Logfile
Format).
# If you do not define any access logfiles within a <VirtualHost>
# container, they will be logged here.  Contrariwise, if you *do*
# define per-<VirtualHost> access logfiles, transactions will be
# logged therein and *not* in this file.
#CustomLog logs/access_log common
#CustomLog logs/referer_log referer
#CustomLog logs/agent_log agent
#
# If you prefer a single logfile with access, agent, and referer
information
# (Combined Logfile Format) you can use the following directive.
CustomLog logs/access_log combined

ServerSignature Off

# Note that if you include a trailing / on fakename then the server
will
# require it to be present in the URL.  So "/icons" isn't aliased in
this
# example, only "/icons/"..
Alias /icons/ "/var/www/example.net/icons/"

<Directory "/var/www/icons">
    Options Indexes MultiViews
    AllowOverride None
    <Limit GET>
        Order allow,deny
        Allow from all
```

```
    </Limit>
    <Limit PUT OPTIONS PROPFIND PUT DELETE PATCH PROPPATCH MKCOL COPY
MOVE LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>

<Directory "/var/www/example.net/htdocs/manual">
    Options MultiViews
    AllowOverride None
    <Limit GET>
        Order allow,deny
        Allow from all
    </Limit>
    <Limit PUT OPTIONS PROPFIND PUT DELETE PATCH PROPPATCH MKCOL COPY
MOVE LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>


# ScriptAlias: This controls which directories contain server scripts.
# ScriptAliases are essentially the same as Aliases, except that
# documents in the realname directory are treated as applications and
# run by the server when requested rather than as documents sent to the
client.
# The same rules about trailing "/" apply to ScriptAlias directives as
to
# Alias.
ScriptAlias /cgi-bin/ "/var/www/example.net/cgi-bin/"

# "/var/www/cgi-bin" should be changed to whatever your ScriptAliased
# CGI directory exists, if you have that configured.
<Directory "/var/www/example.net/cgi-bin">
    AllowOverride None
    Options None
    <Limit GET PUT>
        Order allow,deny
        Allow from all
    </Limit>
    <Limit OPTIONS PROPFIND PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE
LOCK UNLOCK>
        Order deny,allow
        Deny from all
    </Limit>
</Directory>


# Redirect allows you to tell clients about documents which used to
exist in
# your server's namespace, but do not anymore. This allows you to tell
the
# clients where to look for the relocated document.
# Format: Redirect old-URI new-URL
#
```

```
# FancyIndexing is whether you want fancy directory indexing or
standard
IndexOptions FancyIndexing

# AddIcon* directives tell the server which icon to show for different
# files or filename extensions.  These are only displayed for
# FancyIndexed directories.
#
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip

AddIconByType (TXT,/icons/text.gif) text/*
AddIconByType (IMG,/icons/image2.gif) image/*
AddIconByType (SND,/icons/sound2.gif) audio/*
AddIconByType (VID,/icons/movie.gif) video/*

AddIcon /icons/binary.gif .bin .exe
AddIcon /icons/binhex.gif .hqx
AddIcon /icons/tar.gif .tar
AddIcon /icons/world2.gif .wrl .wrl.gz .vrml .vrm .iv
AddIcon /icons/compressed.gif .Z .z .tgz .gz .zip
AddIcon /icons/a.gif .ps .ai .eps
AddIcon /icons/layout.gif .html .shtml .htm .pdf
AddIcon /icons/text.gif .txt
AddIcon /icons/c.gif .c
AddIcon /icons/p.gif .pl .py
AddIcon /icons/f.gif .for
AddIcon /icons/dvi.gif .dvi
AddIcon /icons/uuencoded.gif .uu
AddIcon /icons/script.gif .conf .sh .shar .csh .ksh .tcl
AddIcon /icons/tex.gif .tex
AddIcon /icons/bomb.gif core

AddIcon /icons/back.gif ..
AddIcon /icons/hand.right.gif README
AddIcon /icons/folder.gif ^^DIRECTORY^^
AddIcon /icons/blank.gif ^^BLANKICON^^

# DefaultIcon is which icon to show for files which do not have an icon
# explicitly set.
DefaultIcon /icons/unknown.gif

# AddDescription allows you to place a short description after a file
in
# server-generated indexes.  These are only displayed for FancyIndexed
# directories.
# Format: AddDescription "description" filename
#AddDescription "GZIP compressed document" .gz
#AddDescription "tar archive" .tar
#AddDescription "GZIP compressed tar archive" .tgz

# ReadmeName is the name of the README file the server will look for by
# default, and append to directory listings.
#
# HeaderName is the name of a file which should be prepended to
# directory indexes.
#
```

```
# The server will first look for name.html and include it if found.
# If name.html doesn't exist, the server will then look for name.txt
# and include it as plaintext if found.
#
ReadmeName README
HeaderName HEADER

#
# IndexIgnore is a set of filenames which directory indexing should
ignore
# and not include in the listing.  Shell-style wildcarding is
permitted.
#
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t

#
# AddEncoding allows you to have certain browsers (Mosaic/X 2.1+)
uncompress
# information on the fly. Note: Not all browsers support this.
# Despite the name similarity, the following Add* directives have
nothing
# to do with the FancyIndexing customization directives above.
#
AddEncoding x-compress Z
AddEncoding x-gzip gz

#
# AddLanguage allows you to specify the language of a document. You can
# then use content negotiation to give a browser a file in a language
# it can understand.  Note that the suffix does not have to be the same
# as the language keyword --- those with documents in Polish (whose
# net-standard language code is pl) may wish to use "AddLanguage pl
.po"
# to avoid the ambiguity with the common suffix for perl scripts.
#
AddLanguage en .en
AddLanguage fr .fr
AddLanguage de .de
AddLanguage da .da
AddLanguage el .el
AddLanguage it .it

#
# LanguagePriority allows you to give precedence to some languages
# in case of a tie during content negotiation.
# Just list the languages in decreasing order of preference.
#
LanguagePriority en fr de

#
# AddType allows you to tweak mime.types without actually editing it,
or to
# make certain files to be certain types.
#
# For example, the PHP module (not part of the Apache distribution)
# will typically use:
#
```

```
#AddType application/x-httpd-php .php

#
# AddHandler allows you to map certain file extensions to "handlers",
# actions unrelated to filetype. These can be either built into the
server
# or added with the Action command (see below)
#
# If you want to use server side includes, or CGI outside
# ScriptAliased directories, uncomment the following lines.
#
# To use CGI scripts:
#
#AddHandler cgi-script .cgi

#
# To use server-parsed HTML files
#
#AddType text/html .shtml
#AddHandler server-parsed .shtml

#
# Uncomment the following line to enable Apache's send-asis HTTP file
# feature
#
#AddHandler send-as-is asis

#
# If you wish to use server-parsed imagemap files, use
#
#AddHandler imap-file map

#
# To enable type maps, you might want to use
#
#AddHandler type-map var

#
# Action lets you define media types that will execute a script
whenever
# a matching file is called. This eliminates the need for repeated URL
# pathnames for oft-used CGI file processors.
# Format: Action media/type /cgi-script/location
# Format: Action handler-name /cgi-script/location
#

#
# MetaDir: specifies the name of the directory in which Apache can find
# meta information files. These files contain additional HTTP headers
# to include when sending the document
#
#MetaDir .web

#
# MetaSuffix: specifies the file name suffix for the file containing
the
# meta information.
```

```
#
#MetaSuffix .meta

#
# Customizable error response (Apache style)
#  these come in three flavors
#
#    1) plain text
#ErrorDocument 500 "The server made a boo boo.
#  n.b.  the (") marks it as text, it does not get output
#
#    2) local redirects
#ErrorDocument 404 /missing.html
#  to redirect to local URL /missing.html
#ErrorDocument 404 /cgi-bin/missing_handler.pl
#  N.B.: You can redirect to a script or a document using server-side-
includes.
#
#    3) external redirects
#ErrorDocument 402 http://some.other_server.com/subscription_info.html
#  N.B.: Many of the environment variables associated with the original
#  request will *not* be available to such a script.

#
# The following directives modify normal HTTP response behavior.
# The first directive disables keepalive for Netscape 2.x and browsers
that
# spoof it. There are known problems with these browser
implementations.
# The second directive is for Microsoft Internet Explorer 4.0b2
# which has a broken HTTP/1.1 implementation and does not properly
# support keepalive when it is used on 301 or 302 (redirect) responses.
#
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-
1.0

#
# The following directive disables HTTP/1.1 responses to browsers which
# are in violation of the HTTP/1.0 spec by not being able to grok a
# basic 1.1 response.
#
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0

#
# There have been reports of people trying to abuse an old bug from
pre-1.1
# days.  This bug involved a CGI script distributed as a part of
Apache.
# By uncommenting these lines you can redirect these attacks to a
logging
# script on phf.apache.org.  Or, you can record them yourself, using
the script
# support/phf_abuse_log.cgi.
#
```

```
#<Location /cgi-bin/phf*>
#    Deny from all
#    ErrorDocument 403 http://phf.apache.org/phf_abuse_log.cgi
#</Location>



### Section 3: Virtual Hosts
#
# VirtualHost: If you want to maintain multiple domains/hostnames on
your
# machine you can setup VirtualHost containers for them.
# Please see the documentation at
<URL:http://www.apache.org/docs/vhosts/>
# for further details before you try to setup virtual hosts.
# You may use the command line option '-S' to verify your virtual host
# configuration.

# If you want to use name-based virtual hosts you need to define at
# least one IP address (and port number) for them.
#NameVirtualHost 12.34.56.78:80
#NameVirtualHost 12.34.56.78

# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
#
#<VirtualHost ip.address.of.host.some_domain.com>
#    ServerAdmin webmaster@host.some_domain.com
#    DocumentRoot /www/docs/host.some_domain.com
#    ServerName host.some_domain.com
#    ErrorLog logs/host.some_domain.com-error_log
#    CustomLog logs/host.some_domain.com-access_log common
#</VirtualHost>

#<VirtualHost _default_:*>
#</VirtualHost>
```

Appendix L – authlog_watch.pl

```
#!/usr/bin/perl

###
# $Log: authlog_watch.pl,v $
# Revision 1.1  2003/05/04 16:17:47  christopher
# Initial revision
#
###


use strict;
use warnings;


###
# use `date +"%b %e"` to get today's date in authlog format
# and others to make grep patterns parameters
```

```
###
my $hostname = `hostname`;
chomp $hostname;
my $today_str = `date +"%b %e"`;
chomp $today_str;
my $su_str = " su:";
my $failed_str = ": Failed";


###
# store raw authlog for today in array for reporting
# plus some additional lines of note in seperate array
###
my @rawlog;
my @sulog;
my @failedlog;


###
# need some flags to determine if/how to report
###
my $su_cnt = int 0;
my $failed_cnt = int 0;


###
# need some globals to store resulting report
###
my $subject = "[authlog_watch] ";


###
# open up authlog and process
###
open( IN, "</var/log/authlog" ) || die "cannot open authlog, $!";
{
        while( <IN> ) {
                if( $_ =~ /^$today_str/ ) {
                        chomp $_;
                        push @rawlog, $_;

                        ### look for 'su' usage
                        if( $_ =~ /$su_str/ ) {
                                push @sulog, $_;
                                $su_cnt++;
                        }

                        ### look for failed logins
                        if( $_ =~ /$failed_str/ ) {
                                push @failedlog, $_;
                                $failed_cnt++;
                        }

                } else {
                        next;
                }
        }
```

```
}
close( IN );


###
# include summary in subject
###
if( $su_cnt > 0 ) {
        $subject .= "su: $su_cnt";
        if( $failed_cnt > 0 ) {
                $subject .= ", fail: $failed_cnt ";
        } else {
                $subject .= " ";
        }

} elsif( $failed_cnt > 0 ) {
        $subject .= "fail: $failed_cnt ";

} else {
        $subject .= "nothing to report ";
}


###
# for now test, in future, only when needed
###
if( 1 ) {
        ###
        # write report to sendmail
        ###
        open( OUT, "|/usr/sbin/sendmail -t" ) || die "cannot output to
sendmail, $!";
        print OUT "From: \"authlog_watch\@$hostname\"
<admin\@example.net>\n";
        print OUT "To: \"Admin\" <admin\@example.net>\n";
        print OUT "Subject: $subject - $hostname - $today_str\n\n";

        my $line;

        ### su's
        if( $su_cnt > 0 ) {
              print OUT "su successes:\n";
              print OUT "------------\n";
              foreach $line (@sulog) {
                      print OUT "$line\n";
              }
              print OUT "\n\n";
        }

        ### failed's
        if( $failed_cnt > 0 ) {
              print OUT "failed logins:\n";
              print OUT "-------------\n";
              foreach $line (@failedlog) {
                      print OUT "$line\n";
              }
              print OUT "\n\n";
```

```
        }

        ### include full daily log at end
        print OUT "today's log:\n";
        print OUT "-----------\n";
        foreach $line (@rawlog) {
                print OUT "$line\n";
        }
        print OUT "\n\n";
        print OUT "===========\n";
        print OUT "report end.\n";
        close( OUT );
}
```