Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Step by Step Installation of a Secure Linux Web, DNS and Mail Server

This paper will show how the author configured a Linux based web and e-mail server for a small company. This server is co-located at a local ISP. Because of budget limitations, the company can only locate one physical box at the ISP which limits what security measures that can be installed. The author will seek to explain the choices made. The paper will include instructions on how to build a secure web and email server with an emphasis on two key security areas: 1) Keeping crackers out 2) Detecting any signs of cracke...

Copyright SANS Institute
Author Retains Full Rights

AD

John Holbrook
Step by Step Installation of a Secure Linux Web, DNS and Mail Server
Feb 10, 2004
GIAC GSEC Practical – Version 1.4b, Option 1

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Table of Contents

Abstract....4
Introduction....4
Current Setup....4
Reasons for new install....4
Sudo....5
Security Comparison of Redhat 9.0 and Openna 1.0....7
Default Installed Services....7
Configuration Notes....8
The New Setup....8
Layers of Protection....9
Verifying Integrity of Downloaded Files....9
RPMs....9
Md5sums....10
PGP/GPG Keys....11
A Word About Passwords....12
Openna Linux 1.0 Installation....12
Adding a User....13
OpenSSH Configuration....14
MySQL Installation....16
Securing MySQL....18
BIND Installation....20
Chroot Jailing BIND....23
Qmail Installation....24
Vpopmail Install....32
Apache Installation....34
Mod_security Installation....39
Mod_Dosevasive Installation....40
PHP Installation....41
Giptables Installation....43
Giptables Customization....45
Snort Installation....46
MySQL Snort Configuration....48
ACID Installation....49
ADODB....49
PHPLOT....50
JPGRAPH....50
ACID Installation....50
Authenticated access to the acid pages....52
Time Synchronization....53
AIDE....53
Final Cleanup....56
Chattr of key files....56

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Remove Development RPMs....56
Removal of Downloaded Files....56
Autoupdate 56
Mailing Lists and other sources of information....57
Appendix A BIND Configuration File – named.conf....59
Appendix B – named Initialization Script....61
Appendix C – Apache configuration options....63
Appendix D Apache Initialization File 65
Appendix E – Apache Initialization file....67

© SANS Institute 2004, Author retains full rights

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Abstract

This paper will show how the author configured a Linux based web and e-mail server for a small company. This server is co-located at a local ISP.

Because of budget limitations, the company can only locate one physical box at the ISP which limits what security measures that can be installed. The author will seek to explain the choices made. The paper will include instructions on how to build a secure web and e-mail server with an emphasis on two key security areas:

1) Keeping crackers out
2) Detecting any signs of cracker activity and limiting the changes a cracker can make

This document expects the reader to have a good understanding of installing Linux and the various tools included for text editing, configuration etc.

Introduction

Current Setup

The currently configured server is a Red Hat 7.2 box running several externally available services:

- ➔ Apache 1.3.x Web Server (hosting approximately 10 domains)
- ➔ Bind 9.x
- ➔ qmail
- ➔ Openssh

Reasons for new install

The current server has been in service for approximately 30 months. When it was originally configured the author's knowledge of securing Linux was somewhat limited. Specifically the following items were not installed on the server or configured correctly:

1) Firewall
2) Intrusion Detection System
3) Bind was not configured in a chroot jail

The author has since set up several Linux servers and has standardized on locations for configuration files, etc which make it easier to administer. This wasn't done on the existing server and has caused several problems over the last year or so when updating software.

Another reason for an upgrade is Red Hat has announced the end of life for Red Hat 7.2

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

as of December 31, 2003 and is discontinuing their freely available download distribution in favor of a commercially packaged version.

Their new free version is now called the “Fedora Project” (http://fedora.redhat.com/) but this version is intended for 'bleeding edge' type development, not for a stable, secure web server.

The author looked at several Linux distributions including Mandrake (www.mandrake.com) SuSE (www.suse.com), Debian (www.debian.org) and Openna (www.openna.com). After comparing these distributions, the decision was made to use Openna Linux 1.0 which is available as a free download or can be purchased in a retail package.

Why the author chose Openna Linux:

- ➔ Secure distribution. What isn't needed isn't installed by default. With Red Hat the author usually spends several hours disabling unneeded services and removing unnecessary packages.
- ➔ All software packages for Openna Linux are compiled for the i686 processor which gives us better performance on newer CPUs
- ➔ Prior experience with the creator of Openna Linux – Gerhard Mourani. Gerhard has written several books on securing and optimizing RedHat Linux and Openna Linux which the author has used in the past.

Sudo

Instead of using 'su' (super user) to gain root access Openna uses Sudo.

"Sudo (superuser do) allows a system administrator to give certain users (or groups of users) the ability to run some (or all) commands as root or another user while logging the commands and arguments."[1]

Here's an example of how you can fine tune Sudo. I have a user named “bob” who I want to allow to start and stop Apache and make changes to the Apache configuration files under /etc/httpd. Normally, I would have to give “bob” root access by making him a member of the 'wheel' group, give him the root password, and trust that he does not do anything beyond administering Apache. With sudo here's what I can do:

```
# visudo
```

visudo is the administration tool for the sudo configuration file - /etc/sudoers.

Note: Never directly edit /etc/sudoers. Always use 'visudo'.

This is what my /etc/sudoers file will look like on Openna:

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
# /etc/sudoers: OpenNa, Inc.
# This file MUST be edited with the 'visudo' command as root.

# User alias specification
User_Alias      APACHE_ADMINS = bob

# Cmnd alias specification
Cmnd_Alias     HTTP = /etc/init.d/httpd, /bin/vi /etc/httpd/*

# User privilege specification
# Super-user root can run anything as any user.
root            ALL=(ALL) ALL

# Every users member of the group wheel will be allowed
# to run all commands as super-user root.
%wheel          ALL=(ALL) ALL

# Apache admins may administrate httpd
APACHE_ADMINS  ALL = HTTP
```

Now to test this I secure shell into the server as user 'bob' and do the following:

```
$ sudo /etc/init.d/httpd restart

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these two things:

#1) Respect the privacy of others.
#2) Think before you type.

Password:
Shutting down httpd:                          [ OK  ]
Starting httpd:                               [ OK  ]
```

In /var/log/messages we see the following:

```
Feb  2 13:54:53 server sudo:      bob : TTY=pts/0 ; PWD=/home/bob ; USER=root ;
COMMAND=/etc/init.d/httpd restart
```

Now this is what happens if bob now tries to restart 'sshd' which he is not authorized for:

```
$ sudo /etc/init.d/ssh restart
Sorry, user bob is not allowed to execute '/etc/init.d/ssh restart' as root on
server.domain.com.
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

This unauthorized access is also logged in /var/log/messages:

```
Feb  2 13:59:17 server sudo:      bob : command not allowed ; TTY=pts/0 ;
PWD=/home/bob ; USER=root ; COMMAND=/etc/init.d/ssh restart
```

Sudo is installed by default on Openna Linux and will allow the sysadmin to fine tune access for other users to administer the server. This is unlike plain 'su' which is an all or nothing proposition.

Sudo can be installed on any Linux distribution and would be highly recommended by the author.

Security Comparison of Redhat 9.0 and Openna 1.0

As mentioned earlier, Red Hat Linux installs quite a few services and packages be default which need to be disabled or removed to heighten security.

What follows is a quick security comparison of base installs of RedHat to Openna Linux.

Default Installed Services

Here's a view of the ports open on a base Red Hat 9.0 Server install with the only packages selected being development and ??

```
# netstat -natp
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local AddressForeign Address  State  PID/Program name
tcp       0      0       0.0.0.0:32768        0.0.0.0:*          LISTEN     1572/
tcp       0      0       127.0.0.1:32769      0.0.0.0:*          LISTEN     1702/xinetd
tcp       0      0       0.0.0.0:111          0.0.0.0:*          LISTEN     1553/
tcp       0      0       0.0.0.0:22           0.0.0.0:*          LISTEN     1688/sshd
tcp       0      0       127.0.0.1:631        0.0.0.0:*          LISTEN     1770/cupsd
tcp       0      0       127.0.0.1:25         0.0.0.0:*          LISTEN     1731/
```

Here's what it looks like on a base Openna install:

```
# netstat -natp
Proto Recv-Q Send-Q Local Address Foreign Address  State PID/Program name
tcp       0      0       192.168.0.50:22      0.0.0.0:*          LISTEN     13527/sshd
```

Only 1 port open on Openna vs. 6 on RedHat 9.0.

This is not to say that Red Hat Linux can't be a highly secure distribution. It just takes more time and work to lock down the base install versus doing the same thing with Openna Linux.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Configuration Notes

For this paper I will be using the following IP Addresses:
192.168.0.50 – The new server
192.168.0.5 – Primary DNS for the local ISP
192.168.0.6 – Secondary DNS for the local ISP

Once all testing and installation is completed then these addresses will have to be changed to the correct internet routable addresses.

The new server will be called server.domain.com. The local ISP is isp.net.

I will also be downloading all source files to /usr/tmp unless mentioned otherwise. At the end of the install, and before the server is connected to the internet, all the packages under /usr/tmp will be removed.

The New Setup

A base install of Openna 1.0 will be done on a clone whitebox PC. All core packages will be upgraded in the future via RPMs.

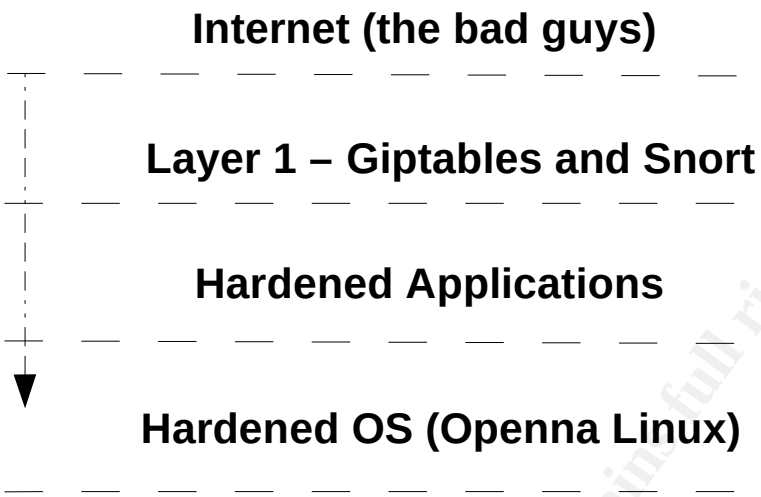
The following packages will be installed from source format instead of RPMs. The reason for source installs is twofold – ease of customization and new versions of software come out in source much quicker than in RPM/binary format. (The web developer for the server also comes out with some very strange requirements for the server which often mean special compile options)

- Apache 2.0.48 with modsecurity and dosevasive modules
- MySQL
- qmail with vpopmail
- Bind
- Giptables firewall scripts

One thing that you will quickly notice whenever you deal with security is that it is always a balancing act. You have to balance security with functionality. Is it possible to make a system 100% secure? Sure. Take the system, put it back in its original box and lock it in Fort Knox. That isn't a realistic option as we'll have high security and no functionality or usability. The decisions on how to install and what to install in this paper are always made with this in mind. However, security is always going to be a higher priority for the author over functionality.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Layers of Protection



With security, one talks about layers of protection. We never want to have only one layer of protection on our systems. We want multiple layers.

On this installation I will be working with what I perceive as three layers:

1) The outer layer which consists of giptables firewall scripts and snort intrusion detection
2) The middle layer which consists of 'hardening' the internet accessible applications to increase security
3) The lowest layer which consists of securing the core operating system, removing unneeded services and software packages.

Verifying Integrity of Downloaded Files

One very important item, from a security perspective, is to ensure is the integrity of any downloaded source files or RPMs. How can you be sure that the file you download has not been changed by a cracker and had a trojan installed on it?

The rest of this paper will assume that each downloaded file's integrity will be verified using one of the following procedures.

RPMs

RPM (RedHat Package Manager supports) pgp keys. Download the key from the website you download RPMs from. For example, the rpm gpg key for openna is available from http://www.openna.com/downloads/RPM-GPG-KEY. Download the file:

The file looks like this (this has been shortened substantially to save on space):

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.2.3 (GNU/Linux)

mQGiBD+7t5sRBADUKbPUIwYUihs1xbPyTCUS7v+TcCFi/uK1uosV/86Ql34Dq06h
9c87HGf6nSDikyUEEC6IIXMKF/dcxcL53L1cgUSf3YJLOSY019cxfxkFyN75jJbm
KIviZtL2D2W9TePODkI0z4ziExCXULAUY/d+JMjjDH376PvIv9ojo9lJ0ic9OohJ
BBgRAgAJBQI/u7eeAhsMAAoJEMYYPar6T2PHamAAn2NuEsVZq1qx+4ZYad4ivWUb
PDX4AJ9ZO+X0Akq6J8oRHu7LEI1kICY94w==
=wYvG
-----END PGP PUBLIC KEY BLOCK-----

As an example. First we import the key

```
#  rpm --import RPM-GPG-KEY
```

Then to verify a signature on a file we can install with

```
# rpm -Uvh rpmfile-x.x.x.rpm
# rpm --checksig autoupdate-5.2.16-1.i686.rpm
autoupdate-5.2.16-1.i686.rpm: (sha1) dsa sha1 md5 gpg OK

# rpm -Uvh autoupdate-5.2.16-1.i686.rpm
Preparing...            ########################################### [100%]
1:autoupdate            ########################################### [100%]
```

If the key wasn't correct we'd see the following:

```
warning: autoupdate-5.2.16-1.i686.rpm: V3 DSA signature: NOKEY, key ID 4b9d15e6
```

Md5sums

The MD5 message-digest algorithm "takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest. The MD5 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA."[2]

Here's an example of verifiying a md5sum on the modsecurity source file
On the modsecurity website we can download mod_security-1.7.4.tar.gz.md5. The file contains the following:

629945812ca7aab4ef2f76ad00172444 *mod_security-1.7.4.tar.gz

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

To verify the integrity of mod_security-1.7.4.tar.gz

```
$ md5sum mod_security-1.7.4.tar.gz
629945812ca7aab4ef2f76ad00172444  mod_security-1.7.4.tar.gz
```

We see that the two numbers match so we can feel comfortable that the file has not been tampered with. Another step to ensure integrity would be to download and verify the md5sum from several mirror sites (if available).

PGP/GPG Keys

As an example, we'll verify the integrity of our Apache web server source files.

Download the Apache PGP Key from http://www.apache.org/dist/httpd/KEYS

Note: On the apache.org website they recommend only downloading the KEYS file from their main ftp server NOT from a mirror.

I downloaded httpd-2.0.48.tar.gz and httpd-2.0.48.tar.gz.asc from one of the Apache mirrors.

First we import the digital key we've received from the website into our keyring.

```
# gpg --import KEYS
```

Now we verify the integrity of the downloaded file.

```
# gpg --verify httpd-2.0.48.tar.gz.asc httpd-2.0.48.tar.gz
```

If you would like further information on using gnupg, an excellent online resource can be found at http://www.gentoo.org/doc/en/gnupg-user.xml.

What if the author of the software being downloaded hasn't offered any way to verify the integrity of her software?

You have a few options. Download the software and hope it hasn't been manipulated. Not a really good idea from a security perspective. The author's recommendation (if you are not a programmer who can analyze the code thoroughly) is to download the software from several mirrors and compare the md5sums. If mirror sites are not available then download the software and hold on to it for several weeks before installing it. Keep a watch out for any notifications of security violations to the site on the various mailing lists from the end of this paper. This isn't the best security recommendation but sometimes its what you have to do.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

A Word About Passwords

Passwords are often the last, and sometimes only, line of defense. In this paper I will be using rather simple passwords (test123,123test and testing). This is only for simplification of the documentation. Never use passwords like these. Always use good passwords.

"What is a good password?[3]

A good password is:

private: it is used and known by one person only
secret: it does not appear in clear text in any file or program or on a piece of paper pinned to the terminal
easily remembered: so there is no need to write it down
not guessable by any program in a reasonable time, for instance less than one week. week."

The best recommendation I can make is this don't use real words. Use the first letters from a sentence that you'll remember (something from a movie for example), change some characters to upper case and swap some numbers for letters.

As an example, let's take a quote from The Matrix. “What is real? How do you define real?” and we'll use the first letter from each word.

wirhdydr

We'll change every odd consonant to uppercase (we'll say y is a consonant).

WirHdYdR

We'll change the i to a 1 and add a space and a non alpha-numeric character.

W1r HdYdR#

Now we have a password that is private, secret, easily remembered and not easily guessable by any program.

Openna Linux 1.0 Installation

I will not go into a lot of details on the installation of the core Openna system. The installation is fairly self explanatory and there is excellent documentation at [4]. I always chose to manually partition the system and here's the new partition table:

dev/hda1 /boot - 50MB

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

/dev/hda5 <swap>
/dev/hda6 / - 1024MB
/dev/hda7 /usr - 2048MB
/dev/hda8 /home - 25600MB
/dev/hda9 /chroot - 512MB
/dev/hda10 /var - 1024MB
/dev/hda11 /tmp - 2048MB

During the install you are asked to enter a Grub (Bootloader password) This password is now required every time the machine is rebooted and must be entered locally from the console. From a security perspective, it is a great idea to have a Grub password. However, reality dictates that we must be able to remotely reboot the server since we do not have 24 hour a day access to the server and problems have a way of occurring in the middle of the night or on weekends. We'll remove the Grub password.

When the system has rebooted completely do the following:

```
# vi /boot/grub/menu.lst
```

You will see a line that reads something like:

```
password –md5 $1$7eJ380$uaA1zbekvQUclLKYpTVpT0
```

Comment it out by adding a '#' to the beginning of the line or you can remove the line completely.

Additional RPMs which are required:

There are several RPMs that are needed for various packages we're going to install. Instead of installing them individually we'll do one big install.

```
# mount /mnt/cdrom
# cd /mnt/cdrom/Openna/RPMS
# rpm -Uvh gnupg-1.2.3-1.i686.rpm autoconf-2.57-1.i686.rpm automake-1.7.8-1.i686.rpm m4-1.4.1-1.i686.rpm libtool-1.5-1.i686.rpm openssl-devel-0.9.7c-1.i686.rpm freetype-2.1.5-1.i686.rpm freetype-devel-2.1.5-1.i686.rpm libjpeg-6b-1.i686.rpm libjpeg-devel-6b-1.i686.rpm libpcap-0.7.2-1.i686.rpm
```

Adding a User

To add a user do the following:

```
# groupadd john
# useradd -g john -c “John Doe” -m -d /home/john -s /bin/bash john
# passwd john
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
Changing password for john
Enter the new password (minimum of 8, maximum of 127 characters)
Please use a combination of upper and lower case letters and numbers.
New password:
Re-enter new password:
Password changed.
```

To allow the user 'john' to sudo to root he needs to be a member of the 'wheel' group:

```
# usermod -G10 john
```

To allow john to log in locally add john to /etc/security/access.conf:

```
# vi +59 /etc/security/access.conf
```

Change
-:ALL EXCEPT root users:ALL
to
-:ALL EXCEPT root john:ALL

OpenSSH Configuration

In the past, sysadmins used unencrypted communication methods such as Telnet, rsync, and FTP to administer the server. This is unacceptable, when easy to use secure alternatives are available.

To enable encrypted communication, we will use OpenSSH which is already installed on our server but will make some changes to the configuration to secure it even more.

What is SSH?

"SSH (Secure Shell) is a program to log into another computer over a network, to execute commands on a remote machine, and to move files from one machine to another. It provides strong authentication and secure communications over insecure channels. It is intended as a replacement for rlogin, rsh, rcp, and rdist."[5]

Instead of using password authentication, public key encryption will be used as an additional layer of security.

The steps to set up public key encryption with OpenSSH are:
1) Create a public/private key pair on our client PC
2) Copy the public key to the server and place it in ~/.ssh
3) Change the configuration on the server to use public key authentication instead of password authentication.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

To generate a 2048 bit DSA key on our client PC:

```
$ ssh-keygen -t dsa -b 2048
Generating public/private dsa key pair.
Enter file in which to save the key (/home/john/.ssh/id_dsa):
Created directory '/home/john/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/john/.ssh/id_dsa.
Your public key has been saved in /home/john/.ssh/id_dsa.pub.
The key fingerprint is:
33:0f:97:4f:d8:f7:a6:44:84:05:a4:64:7a:88:e1:a3 john@client.domain.com
```

For the password use a unique password. You do not want to use the same password as your login or root password.

This produces a pair of keys under /home/john/.ssh/ - id_dsa (private key) id_dsa.pub (public key). The private key stays on the client and must be kept secure. The public key needs to be copied to the server and placed under /home/john/.ssh.

```
$ cd ~/.ssh
$ scp id_dsa.pub john@192.168.0.50:~/
john@192.168.0.50's password:
```

Do not use the same password as your login password.

To configure the public key on the server:

```
$ cd ~
$ mkdir .ssh
$ chmod 700 .ssh
$ cat id_dsa.pub > .ssh/authorized_keys2
$ chmod 600 ~/.ssh/authorized_keys2
```

To change OpenSSH to use DSA key encryption instead of passwords on the server:

```
# vi /etc/ssh/sshd_config
```

Change:
Protocol 1,2
to
Protocol 2

Change:
ListenAddress 0.0.0.0
to

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

ListenAddress 192.168.0.50

Change:
RhostsRSAAuthentication yes
to
RhostsRSAAuthentication no

Change:
RSAAuthentication yes
to
RSAAuthentication no

Change:
PasswordAuthentication yes
to
PasswordAuthentication no

Restart OpenSSH

```
# /etc/init.d/sshd restart
```

Now when you ssh from your client to the server you will enter the password from your DSA key.

```
$ ssh server.domain.com
Enter passphrase for key '/home/john/.ssh/id_dsa':
```

Public key encryption increases the level of security on the server because for somebody to secure shell into the box they must have two things – the password on the DSA key and a copy of the DSA private key.

If you try to secure shell into the server and don't have the correct information you will receive the following:

```
$ ssh server.domain.com
Permission denied (publickey,keyboard-interactive).
```

Another way to protect our OpenSSH traffic would be to block access to the service to specific IP addresses using either tcpwrappers or our Giptables Firewall scripts which we install later. Unfortunately, this is not possible as the users who require access are using cable modem for internet access which do not offer static IP addresses.

MySQL Installation

Download mysql-max-4.0.17-pc-linux-i686.tar.gz from one of the mysql mirrors at

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

http://www.mysql.com.

These instructions can be found in the file called INSTALL-BINARY in the downloaded file.

```
# groupadd -g 49 mysql
# useradd -c "MySQL Server" -d /usr/mysql -g 49 -s /sbin/nologin -u 49 mysql
# mv mysql-max-4.0.17-pc-linux-i686.tar.gz /usr
# cd /usr/
# tar xvzf mysql-max-4.0.17-pc-linux-i686.tar.gz
# ln -sf /usr/mysql-max-4.0.17-pc-linux-i686 /usr/mysql
# cd mysql
# scripts/mysql_install_db
# chown -R root .
# chown -R mysql data
# chgrp -R mysql .
# bin/mysqld_safe --user=mysql &
```

Copy the initialization script to the correct location

```
# cp support-files/mysql.server /etc/init.d/mysqld
# chown 0.0 /etc/init.d/mysqld
# chmod 700 /etc/init.d/mysqld
# vi +47 /etc/init.d/mysqld
```

Change line:
datadir=/usr/local/mysql/data
to
datadir=/usr/mysql/data

```
# vi +51 /etc/init.d/mysqld
```

Change line:
  basedir=/usr/local/mysql
to
  basedir=/usr/mysql

```
# vi +117 /etc/init.d/mysqld
```

Change line:
 if test "$datadir" != "/usr/local/mysql/data"
to
 if test "$datadir" != "/usr/mysql/data"

```
# chkconfig --add mysqld
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
# vi +148 /etc/init.d/mysqld
```

Change line:
$bindir/mysqld_safe --datadir=$datadir --pid-file=$pid_file >/dev/null 2>&1 &
to
$bindir/mysqld_safe --datadir=$datadir --pid-file=$pid_file --user=mysql >/dev/null 2>&1 &

```
# cp /usr/mysql/support-files/my-medium.cnf /etc/my.cnf
```

Under the support-files directory you find several other .cnf files (small, large and huge) that can be used. The my-medium.cnf file is intended for a server which is running other services besides MySQL. You'd want to use huge or large if a pure mysql server.

```
# vi +36 /etc/my.cnf
```

Add the following:
set-variable=local-infile=0

By default MySQL listens on port 3306. We do not need this as we will not be connecting to MySQL from other servers. Stop MySQL from listening on any TCP/IP port:

```
# vi +44 /etc/my.cnf
```

Change
#skip-networking
to
skip-networking

Securing MySQL

A default install of MySQL is somewhat lacking in security. To fix that we are going to do the following:

1) Remove all default users which are installed in MySQL
2) Create a new admin user (sqladmin) instead of using the default name of 'root@localhost'
3) Disable network access to the MySQL port (3306)

Log into mysql

```
# cd /usr/mysql/bin
# ./mysql -u root
```

Delete all users (this is to get rid of any default created accounts) specifically root@localhost and root@hostname.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
mysql> connect mysql;

Connection id:    2
Current database: mysql

mysql> delete from user;
```

At this point in time you have no users in MySQL. That means if you disconnect from MySQL you will not be able to log back in and will have to reinstall MySQL.

In the instructions for MySQL you should be able to insert a plain text password when creating a new user however I was unable to. Instead I generated a 16 bit hexadecimal password to the screen:

```
mysql> select password('test123');

+---------------------+
| password('test123') |
+---------------------+
| 39817a786ddf7333    |
+---------------------+
1 row in set (0.00 sec)
```

Create an admin user (sqladmin) with full privileges and an encrypted password. Remember to use a good password that is unique.

```
mysql> grant all privileges on *.* to sqladmin@localhost identified by password
'39817a786ddf7333' with grant option;
Query OK, 0 rows affected (0.00 sec)

mysql> select host,user,password from user;
+-----------+----------+------------------+
| host      | user     | password         |
+-----------+----------+------------------+
| localhost | sqladmin | 39817a786ddf7333 |
+-----------+----------+------------------+
1 row in set (0.00 sec)
```

To summarize, I deleted the default MySQL users amd created a new admin account..

Restart mysql.

```
# /etc/init.d/mysqld restart
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

BIND Installation

This server will be the master for the domains it hosts. One of the DNS servers at the local ISP (ns1.isp.net) will be the slave.

I am going to install BIND and verify it works, and then install it in a chroot jail for added security.

We will also ensure that only ns1.isp.net will be allowed to do zone transfers.

The following instructions primarily come from [6]. I'm going to modify them slightly because I prefer to have the various configuration files under /etc/named.

Download bind-9.2.3.tar.gz from one of the mirror sites at http://www.isc.org

Create a user and group for Bind:

```
# groupadd -g 25 named > /dev/null 2>&1 || :
# useradd -c "BIND DNS Server" -d /var/named -g 25 -s /bin/false -u 25 named >  /
dev/null 2>&1 || :
# tar xvzf bind-9.2.3.tar.gz
# cd bind-9.2.3
# vi +105 bin/named/include/named/globals.h
```

Change
"/run/named.pid");
to
"/run/named/named.pid");

Change (two lines down)
"/run/lwresd.pid");
to
"/run/named/lwresd.pid");

```
# CFLAGS="-O2 -march=i686 -funroll-loops"; export CFLAGS
# ./configure \
> --prefix=/usr \
> --sysconfdir=/etc \
> --localstatedir=/var \
> --mandir=/usr/share/man \
> --with-libtool \
> --disable-ipv6
# make
# make install
# strip /usr/sbin/named
# mkdir -p /etc/named
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
# mkdir -p /var/run/named
# install -c -m0600 bin/rndc/rndc.conf /etc/
# chown named.named /etc/rndc.conf
# chown named.named /etc/named
# chown named.named /var/run/named/
# /sbin/ldconfig
```

Note: I removed –with-openssl as the ISP does not support SSL.

```
# vi /etc/named.conf
```

See Appendix A for the /etc/named.conf file. The entry for 192.168.0.0/16 under the known fake addresses will have to be uncommented when the server is put into service.

```
# chmod 600 /etc/named.conf
# chown named.named /etc/named.conf
```

Now it time to create the /var/named/db.cache file which is the Root Server Hints File.

```
# dig @a.root-servers.net . ns > db.cache
# mv db.cache /etc/named/
#chmod 644 /etc/named/db.cache
# chown named.named /etc/named/db.cache
```

Create /etc/named/db.localhost

```
# vi /etc/named/db.localhost
```

Add the following:

```
$TTL 86400
@ IN SOA localhost. root.localhost. (
00 ; Serial
10800 ; Refresh after 3 hours
3600 ; Retry after 1 hour
604800 ; Expire after 1 week
86400 ) ; Minimum

IN NS localhost.

localhost IN A 127.0.0.1
```

```
# chmod 644 /etc/named/db.localhost
# chown named.named /etc/named/db.localhost
```

Create /etc/named/0.0.127.in-addr.arpa: The Reverse Mapping File

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
# chmod 644 /etc/named/0.0.127.in-addr.arpa
# chown named.named /etc/named/0.0.127.in-addr.arpa
```

Create the BIND System Configuration File

```
# vi /etc/sysconfig/named
```

Add the following:

```
# This option will run named in a chroot environment.
#ROOTDIR="/chroot/named/"

# These additional options will be passed to named at startup.
# Don't add .t here, use ROOTDIR instead.
#OPTIONS=""
```

Create the named initialization script

```
# vi /etc/init.d/named
```

See Appendix B for a complete listing of /etc/init.d/named

```
# chmod 700 /etc/init.d/named
# chown root.root /etc/init.d/named
# vi /etc/named/db.domain.com
$TTL 4H
@       IN      SOA     domain.com. webmaster.domain.com. (
                 2004013001          ; serial YYYYMMDD##
                 1H              ; Refresh after 3 hours
                 2H              ; Retry after 1 hour
                 1209600S            ; Expire after 1 week
                   1S )              ; Minimum TTL of 1 day

; ***** Nameserver (NS) records. ******************************

domain.com. IN     NS      ns1.domain.com.
;domain.com.        IN      NS      ns2.isp.com.

; ***** Mail Exchange (MX) Records ****************************

       MX      10        mail

; ***** Address (A) Records ***********************************

localhost               A       127.0.0.1
server                  A       192.168.0.50
;
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
; ***** Canonical Name (CNAME) records ************************
;
ns1                     CNAME   server
mail                    CNAME   server
www                     CNAME   server
```

```
# vi /etc/named/db.0.168.192
```

Add the following:

```
$TTL 3h
0.168.192.in-addr.arpa. IN SOA server.domain.com. webmaster.domain.com. (
            2004013001      ; Serial YYYYMMDD##
            3h              ;Refresh after 3 hours
            1h              ; Retry after 1 hour
            1w              ;Expire after 1 week
            1h )            ; Negative caching TTL of 1 hour

;
; Name Servers
;
0.168.192.in-addr.arpa.         IN      NS      server.domain.com.

;
; Addresses point to canonical name
;

50.0.168.192.in-addr.arpa.      IN      PTR     server.domain.com.
```

```
# chmod 644 /etc/named/*
# chown named.named /etc/named/*
```

Now we start Bind and verify that we can resolve names from it.

```
# /etc/init.d/named start
# ping server.domain.com
```

You should see the name resolve to the correct IP Address.

Chroot Jailing BIND

Now to improve the security of Bind we are going to run it in a chroot jailed environment.[7]

What is a chroot jail?

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Application jails, also known as "change root jails" or "*chroot jails,*" are another effective countermeasure. Supported by all Linux and Unix systems, application jails put up a nearly impenetrable barrier between the "jailed" software and the rest of the system. And because a jail is enforced by the operating system and not by an application, it can provide an enormous level of safety. A chroot jail "incarcerates" untrusted applications, and acts like a guard, almost literally, for applications that already have substantial security measures built-in.[8]

```
# /etc/init.d/named stop
# mkdir -p /chroot/named
# cd /chroot/named
# mkdir -p dev etc/named var/run/named
# mknod /chroot/named/dev/null c 1 3
# mknod /chroot/named/dev/random c 1 8
# chmod 666 /chroot/named/dev/null
# chmod 666 /chroot/named/dev/random
# cp /etc/localtime /chroot/named/etc/
# mv /etc/named.conf /chroot/named/etc/
# mv /etc/named/* /chroot/named/etc/named/
# chown -R named.named /chroot/named
```

Now we need to tell BIND to run in the chroot jail.

```
# vi /etc/sysconfig/named
```

Uncomment the line that reads
#ROOTDIR="/chroot/named/"

Restart BIND and verify that it is working.

```
# /etc/init.d/bind restart
```

Qmail Installation

The author chose qmail over other MTAs for the following reasons:

1) Security – Qmail is designed from the ground up to be a secure, drop-in replacement for sendmail. The author of qmail, Daniel J. Bernstein, has an unclaimed $500 bounty offered to the "first person to publish a verifiable security hole in the latest version of qmail: for example, a way for a user to exploit qmail to take over another account"[9] since March 1997.
2) Prior experience with qmail. The author has set up several qmail installations and once they are up and running they take very little maintenance.

As a security comparison, here's the results of a search at http://www.cert.org for security vulnerabilities in different MTAs:

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

sendmail – “38 results found”
postfix – “3 results found”
qmail - “There were no results for *qmail*.”

I will be using the install information from [10] .

Download:

http://www.qmail.ca/netqmail-1.05.tar.gz
http://cr.yp.to/ucspi-tcp/ucspi-tcp-0.88.tar.gz
http://cr.yp.to/daemontools/daemontools-0.76.tar.gz

```
# mkdir -p /package
# mv daemontools-0.75.tar.gz /package
# cd /package
# tar xvzf daemontools-0.76.tar.gz
# chmod 1755 /package
# cd /usr/tmp
# tar xvzf netqmail-1.05.tar.gz
# cd netqmail-1.05
# ./collate.sh
You should see 7 lines of text below.  If you see anything
else, then something might be wrong.
[1] Extracting qmail-1.03...
[2] Patching qmail-1.03 into netqmail-1.05.  Look for errors below:
    24
[4] The previous line should say 24 if you used GNU patch.
[5] Renaming qmail-1.03 to netqmail-1.05...
[6] Continue installing qmail using the instructions found at:
[7] http://www.lifewithqmail.org/lwq.html#installation
```

```
# cd /usr/tmp
# tar xvzf ucscpi-tcp-0.88.tar.gz
# mkdir /var/qmail
# cd /usr/tmp/netqmail-1.05/netqmail-1.05
# cp INSTALL.ids IDS
# vi IDS
```

Remove all entries in IDS not pertinent to a Linux install. I then customized the file to add specific group and user ids. You can use whatever ID #s you'd like. This is just an example. It should look like this:

```
groupadd -g 2108 nofiles
useradd -u 7790 -g nofiles -d /var/qmail/alias -s /bin/false alias
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
useradd -u 7791 -g nofiles -d /var/qmail -s /bin/false qmaild
useradd -u 7792 -g nofiles -d /var/qmail -s /bin/false qmaill
useradd -u 7793 -g nofiles -d /var/qmail -s /bin/false qmailp
groupadd -g 2107 qmail
useradd -u 7794 -g qmail -d /var/qmail -s /bin/false qmailq
useradd -u 7795 -g qmail -d /var/qmail -s /bin/false qmailr
useradd -u 7796 -g qmail -d /var/qmail -s /bin/false qmails
```

```
# chmod 700 IDS
# ./IDS
```

Now we build qmail

```
# make setup check
# ./config
```

You should the following:

```
Your hostname is server.domain.com.
Your host's fully qualified name in DNS is server.domain.com.
Putting server.domain.com into control/me...
Putting domain.com into control/defaultdomain...
Putting domain.com into control/plusdomain...

Checking local IP addresses:
0.0.0.0: PTR lookup failed. I assume this address has no DNS name.
127.0.0.1: Adding localhost to control/locals...
192.168.0.50: Adding server.domain.com to control/locals...

If there are any other domain names that point to you,
you will have to add them to /var/qmail/control/locals.
You don't have to worry about aliases, i.e., domains with CNAME records.

Copying /var/qmail/control/locals to /var/qmail/control/rcpthosts...
Now qmail will refuse to accept SMTP messages except to those hosts.
Make sure to change rcpthosts if you add hosts to locals or virtualdomains!
```

Install ucspi-tcp

```
# cd /usr/tmp/ucspi-tcp-0.88
# patch < /usr/tmp/netqmail-1.05/other-patches/ucspi-tcp-0.88.errno.patch
patching file error.h
# make
# make setup check
```

Install daemontools

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
# cd /package/admin/daemontools-0.76/src
# patch < /usr/tmp/netqmail-1.05/other-patches/daemontools-0.76.errno.patch
patching file error.h
# cd ..
# package/install
```

Daemontools should automatically start now. To verify:

```
# ps waux | grep sv
root     1544  0.0  0.2  2032  992 ?        S    08:47   0:00 /bin/sh /command/svscanboot
root     6447  0.0  0.0  1404  304 ?        S    08:47   0:00 svscan /service
```

Qmail startup script

```
# vi /var/qmail/rc
```

Add the following

```
#!/bin/sh

# Using stdout for logging
# Using control/defaultdelivery from qmail-local to deliver messages by default

exec env - PATH="/var/qmail/bin:$PATH" \
qmail-start "`cat /var/qmail/control/defaultdelivery`"
```

```
# chmod 755 /var/qmail/rc
# mkdir /var/log/qmail
```

Set the default mailbox type

```
# echo ./Maildir > /var/qmail/control/defaultdelivery
```

qmail startup script

Gerhard Mourani has a complete set of configuration files and scripts available at ftp://ftp.openna.com/ConfigFiles-v3.0/floppy-3.0.tgz. I will be using the initialization script from the file.

```
# tar xvzf floppy-3.0.tgz
# cp floppy-3.0/Qmail/etc/init.d/qmail /etc/init.d/
# chmod 700 /etc/init.d/qmail
# chown root.root /etc/init.d/qmail
# chkconfig --add qmail
```

Download http://www.lifewithqmail.org/qmailctl-script-dt70

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
# mv qmailctl-script-dt70 /var/qmail/bin/qmailctl
# chmod 755 /var/qmail/bin/qmailctl
# ln -s /var/qmail/bin/qmailctl /usr/bin
# mkdir -p /var/qmail/supervise/qmail-send/log
# mkdir -p /var/qmail/supervise/qmail-smtpd/log
# mkdir -p /var/qmail/supervise/qmail-pop3d/log
# vi /var/qmail/supervise/qmail-send/run
```

Add the following:

```
#!/bin/sh
exec /var/qmail/rc
```

```
# vi /var/qmail/supervise/qmail-send/log/run
```

Add the following

```
#!/bin/sh

exec /usr/local/bin/setuidgid qmaill /usr/local/bin/multilog t /var/log/qmail
```

```
# vi /var/qmail/supervise/qmail-smtpd/run
```

Add the following:

```
#!/bin/sh

QMAILDUID=`id -u qmaild`
NOFILESGID=`id -g qmaild`
MAXSMTPD=`cat /var/qmail/control/concurrencyincoming`
LOCAL=`head -1 /var/qmail/control/me`

if [ -z "$QMAILDUID" -o -z "$NOFILESGID" -o -z "$MAXSMTPD" -o -z "$LOCAL" ]; then
   echo QMAILDUID, NOFILESGID, MAXSMTPD, or LOCAL is unset in
   echo /var/qmail/supervise/qmail-smtpd/run
   exit 1
fi

if [ ! -f /var/qmail/control/rcpthosts ]; then
   echo "No /var/qmail/control/rcpthosts!"
   echo "Refusing to start SMTP listener because it'll create an open relay"
   exit 1
fi

exec /usr/local/bin/softlimit -m 2000000 \
   /usr/local/bin/tcpserver -v -R -l "$LOCAL" -x /etc/tcp.smtp.cdb -c "$MAXSMTPD" \
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

-u "$QMAILDUID" -g "$NOFILESGID" 0 smtp /var/qmail/bin/qmail-smtpd 2>&1

```
# echo 20 > /var/qmail/control/concurrencyincoming
# chmod 644 /var/qmail/control/concurrencyincoming
# vi /var/qmail/supervise/qmail-smtpd/log/run
```

Add the following:

```
#!/bin/sh
exec /usr/local/bin/setuidgid qmaill /usr/local/bin/multilog t /var/log/qmail/smtpd
```

```
# chmod 755 /var/qmail/supervise/qmail-send/run
# chmod 755 /var/qmail/supervise/qmail-send/log/run
# chmod 755 /var/qmail/supervise/qmail-smtpd/run
# chmod 755 /var/qmail/supervise/qmail-smtpd/log/run
# mkdir -p /var/log/qmail/smtpd
# chown qmaill /var/log/qmail /var/log/qmail/smtpd
# ln -s /var/qmail/supervise/qmail-send /var/qmail/supervise/qmail-smtpd /service
# echo '127.:allow,RELAYCLIENT="" >>/etc/tcp.smtp
# qmailctl cdb
```

Create system aliases:

```
# echo john > /var/qmail/alias/.qmail-root
# echo john > /var/qmail/alias/.qmail-postmaster
# ln -s /var/qmail/.qmail-postmaster /var/qmail/alias/.qmail-mailer-daemon
# chmod 644 /var/qmail/alias/.qmail-root /var/qmail/alias/.qmail-postmaster
```

Start qmail and verify it is working:

```
# qmailctl start
# qmailctl stat
```

You should see:

```
/service/qmail-send: up (pid 31772) 51 seconds
/service/qmail-send/log: up (pid 12997) 51 seconds
/service/qmail-smtpd: up (pid 10926) 51 seconds
/service/qmail-smtpd/log: up (pid 18333) 1 seconds
messages in queue: 0
messages in queue but not yet preprocessed: 0
```

Now we need to test to see that mail is working. Replace the name 'john' here with the name you used when creating the system aliases.

```
# su john
$ /var/qmail/bin/maildirmake $HOME/Maildir
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
$ echo ./Maildir/ > ~/.qmail
# ln -s /var/qmail/bin/sendmail /usr/sbin/sendmail
```

Now we'll do a quick test to see if the system is accepting e-mail

```
# echo To: postmaster@server.domain.com | /var/qmail/bin/qmail-inject
# cd /home/john/Maildir/new
# ls -la
```

You should see a file in the directory like this:

1075665096.22387.server.domain.com

You can take a look at the file and see that it is the e-mail message:

```
# cat 1075665096.22387.server.domain.com

Return-Path: <root@server.domain.com>
Delivered-To: john@server.domain.com
Received: (qmail 27454 invoked by alias); 1 Feb 2004 19:51:36 -0000
Delivered-To: postmaster@server.domain.com
Received: (qmail 3400 invoked by uid 0); 1 Feb 2004 19:51:36 -0000
Date: 1 Feb 2004 19:51:36 -0000
Message-ID: <20040201195136.22873.qmail@server.domain.com>
From: root@server.domain.com
To: postmaster@server.domain.com
```

Now we need to add POP3 access for e-mail.

Download http://cr.yp.to/checkpwd/checkpassword-0.90.tar.gz

```
# tar xvzf checkpassword-0.90.tar.gz
# cd checkpassword-0.90
```

There is a bug in a the error.h files with glibc > 2.2. You must use a patch to fix this issue. Download http://www.qmail.org/moni.csi.hu/pub/glibc-2.3.1/checkpassword-0.90.errno.patch

```
# patch < /usr/tmp/checkpassword-0.90.errno.patch
# make
# make setup check
```

Create /var/qmail/supervise/qmail-pop3d/run

```
# vi /var/qmail/supervise/qmail-pop3d/run
```

Add the following:

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
#!/bin/sh
exec /usr/local/bin/softlimit -m 2000000 \
        /usr/local/bin/tcpserver -v -R -h -l 0 0 110 /var/qmail/bin/qmail-popup \
        server.domain.com /bin/checkpassword /var/qmail/bin/qmail-pop3d Maildir
        2 >&1
```

```
# vi /var/qmail/supervise/qmail-pop3d/log/run
```

Add the following:

```
#!/bin/sh
exec /usr/local/bin/setuidgid qmaill /usr/local/bin/multilog t \
/var/log/qmail/pop3d
```

```
# mkdir /var/log/qmail/pop3d
# chown qmaill /var/log/qmail/pop3d
# chmod 755 /var/qmail/supervise/qmail-pop3d/run
# chmod 755 /var/qmail/supervise/qmail-pop3d/log/run
# ln -s /var/qmail/supervise/qmail-pop3d /service
```

We need to add the pop3d entries to qmailctl

```
# vi /bin/qmailctl
```

Add the following to `qmailctl`'s "start" section:

```
if svok /service/qmail-pop3d ; then
  svc -u /service/qmail-pop3d /service/qmail-pop3d/log
else
  echo qmail-pop3d supervise not running
fi
```

Add the following to `qmailctl`'s "stop" section:

```
echo "  qmail-pop3d"
svc -d /service/qmail-pop3d /service/qmail-pop3d/log
```

Add the following to `qmailctl`'s "stat" section:

```
svstat /service/qmail-pop3d
svstat /service/qmail-pop3d/log
```

Add the following to `qmailctl`'s "pause" section:

```
echo "Pausing qmail-pop3d"
svc -p /service/qmail-pop3d
```

Add the following to `qmailctl`'s "cont" section:

```
echo "Continuing qmail-pop3d"
svc -c /service/qmail-pop3d
```

Add the following to `qmailctl`'s "restart" section:

```
echo "* Restarting qmail-pop3d."
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
svc -t /service/qmail-pop3d /service/qmail-pop3d/log
```

Vpopmail Install

Vpopmail allows us to use virtual domains with qmail. Virtual domains allow us to create e-mail accounts without the account names having to be an actual user on the server.

Following the instructions from [11] with some customization.
Download http://www.inter7.com/vpopmail/vpopmail-5.2.2.tar.gz

```
# tar xvzf vpopmail-5.2.2.tar.gz
# groupadd -g 89 vchkpw
# useradd -c "Vpopmail" -d /home/vpopmail -g 89 -s /bin/false -u 89 vpopmail
```

Quick cleanup of the default bash configuration files

```
# rm -f /home/vpopmail/.b*
# cd vpopmail-5.2.2
# ./configure –prefix=/usr –sysconfdir=/etc ----enable-roaming-users=y –
mandir=/usr/share/man –enable-relay-clear-minutes=60 –enable-tcpserver-
file=/etc/tcp.smtp –enable-clear-passwd=y –enable-tcpserver-file=/etc/tcp.smtp –
enable-default-quota=10000000S,1000C
```

You should see:

```
vpopmail 5.2.2
 Current settings
--------------------------------------

vpopmail directory = /home/vpopmail
             uid = 89
             gid = 89
        ip alias = OFF --enable-ip-alias-domains=n (default)
address extentions = OFF --enable-qmail-ext=n (default)
    roaming users = ON  --enable-roaming-users=y
   tcpserver file = /etc/tcp.smtp
   open_smtp file = /home/vpopmail/etc/open-smtp
      user quota = 15000000S,1000C -enable-hardquota=15000000S,1000C
table optimization = many domains --enable-many-domains=y default
      auth module = cdb default
  system passwords = OFF --enable-passwd=n default
     file locking = ON  --enable-file-locking=y default
        file sync = OFF --enable-file-sync=n default disable vdelivermail fsync
     auth logging = ON  --enable-auth-logging=y default
    mysql logging = OFF --enable-mysql-logging=n default
     clear passwd = ON  --enable-clear-passwd=y (default)
 valias processing = OFF --enable-valias=n
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
      pop syslog = show only failure attempts
                --enable-logging=e default
   default domain =  --enable-default-domain=
        auth inc = -lcdb
        auth lib =
```

```
        # make
        # make install-strip
        # crontab -e
```

Add the following line:

40 * * * * /home/vpopmail/bin/clearopensmtp 2>&1 > /dev/null
Add a virtual domain:

```
        # cd /home/vpopmail/bin
        # ./vadddomain domain.com test123
```

Note that 'test123' is the password for the postmaster account for this virtual domain.

To add more users do the following:

```
        # ./vadduser name@domain.com test123
```

for each user.

```
        # echo 40 > /var/qmail/control/concurrencypop3
        # vi /var/qmail/supervise/qmail-pop3d/run
```

Add the following to the file:

```
#!/bin/sh
MAXPOP3D=`head -1 /var/qmail/control/concurrencypop3`
exec /usr/local/bin/softlimit -m 2000000 \
      /usr/local/bin/tcpserver -v -R -H -l 0 -x /etc/tcp.pop3.cdb -c "$MAXPOP3D" \
      0 110 /var/qmail/bin/qmail-popup server.domain.com /home/vpopmail/bin/vchkpw \
      /var/qmail/bin/qmail-pop3d Maildir 2>&1
```

Lets send a test e-mail

```
        # echo to: joe@domain.com | /var/qmail/bin/qmail-inject
```

You should see an e-mail to joe@domain.com under /
home/vpopmail/domains/domain.com/joe/Maildir/new

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Apache Installation

For the web server, I will install and configure Apache to disable unneeded modules and install a pair of security related modules – mod_security and mod_dosevasive.

Documentation for the Apache Install primarily comes from [12]
Downloaded httpd-2.0.48.tar.gz from one of the Apache mirror sites at http://www.apache.org.

```
# tar xvzf httpd-2.0.48.tar.gz
```

mod_security – Download mod_security-1.7.4.tar.gz from http://www.modsecurity.org/download/.

I am going to install mod_security as a static module in Apache.

```
# tar xvzf mod_security-1.7.4.tar.gz
# cd mod_security-1.7.4/apache2/
# cp mod_security.c ../../httpd-2.0.48/modules/mappers/
```

```
# tar xvzf httpd-2.0.48.tar.gz
```

Create the user that apache will run under (You must never have Apache run as 'root')

```
# groupadd -g 48 www
# useradd -c "Apache Web Server" -d /home/httpd -g 48 -s /sbin/nologin -u 48
www
# cd httpd-2.0.48
# vi +66 support/apxs.in
```

Change
 my $installbuilddir = "@exp_installbuilddir@";
to
 my $installbuilddir = "/usr/lib/httpd/build/";

```
# vi +119 server/mpm/prefork/prefork.c
```

Change
 #define DEFAULT_SERVER_LIMIT 256
to
 #define DEFAULT_SERVER_LIMIT 1024

Configure Apache with the the configuration options in Appendix C. They are too long to put into the main section of this paper.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
# make
# make install
# strip /usr/sbin/httpd
```

The strip command removes symbols from object files. This will make our binary files smaller in size.

```
# chmod 0511 /usr/sbin/httpd
# strip --strip-debug -R .comment /usr/lib/httpd/modules/*.so
# mkdir -p /var/log/httpd
# mkdir -p /var/lib/dav
# rm -rf /var/logs
# mv /home/httpd/build /usr/lib/httpd/build
# rm -f /usr/lib/httpd/build/libtool
# ln -s /usr/bin/libtool /usr/lib/httpd/build/libtool
# ln -s /var/log/httpd /etc/httpd/logs
# ln -s /var/run /etc/httpd/run
# ln -s /usr/lib/httpd/modules /etc/httpd/modules
# ln -s /usr/lib/httpd/build /etc/httpd/build
```

The following is added to fix problem when building PHP4:

```
# mkdir /home/httpd/build
# ln -s /usr/bin/libtool /home/httpd/build/libtool
```

Create the httpd logrotate file:

```
# vi /etc/logrotate.d/httpd
```

Add the following:

```
/var/log/httpd/*_log {
    missingok
    notifempty
    sharedscripts
    postrotate
    /usr/bin/killall -HUP httpd
    endscript
}
```

```
# chmod 640 /etc/logrotate.d/httpd
# chown 0.0 /etc/logrotate.d/httpd
# vi /etc/sysconfig/httpd
```

Add the following:

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
# Uncomment the following line to enable SSL support with Apache.
# Certificate should be already configured into httpd.conf file.
#
#OPTIONS="-DSSL"
```

Create the Apache Initialization script (/etc/init.d/httpd) which is listed in Appendix D.

```
# vi /etc/init.d/httpd
# chmod 700 /etc/init.d/httpd
# chown 0.0 /etc/init.d/httpd
# chkconfig --add httpd
# chkconfig --level 345 httpd on
```

Create the apache configuration file.

```
# vi /etc/httpd/conf/httpd.conf
```

Okay I copied the httpd.conf from the openna website and it worked!

```
# vi +26 /etc/httpd/conf/httpd.conf
```

Change the two Listen lines to the correct IP Address. For example:

Listen 192.168.0.50:80
Listen 192.168.0.50:443

```
# vi +204 /etc/httpd/conf/httpd.conf
```

Change NameVirtualHost to correct IP
Change VirtualHost also to the correct IP

```
# cp /home/httpd/htdocs/index.html.en /home/httpd/htdocs/index.html
```

SETTING UP APACHE WITH TLS/SSL SUPPORT

```
# cp /var/log/boot.log /usr/share/ssl/random1
# cp /var/log/cron /usr/share/ssl/random2
# cp /var/log/dmesg /usr/share/ssl/random3
# cp /var/log/messages /usr/share/ssl/random4
# cp /var/log/secure /usr/share/ssl/random5
# cd /usr/share/ssl
# openssl genrsa -rand random1:random2:random3:random4:random5 -out
www.key 1024
```

Now we will produce a Certificate Signing Request (CSR)

```
# openssl req -new -key www.key -out www.csr
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:CA
State or Province Name (full name) [Some-State]:BC
Locality Name (eg, city) [Some-Locality]:Penticton
Organization Name (eg, company) [Some-Organization Ltd]:Domain Inc.
Organizational Unit Name (eg, section) [Some-Organizational]:Web Hosting
Common Name (eg, YOUR name) [www.domain.com]:www.domain.com
Email Address [admin@domain.com]:root@localhost

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Normally we would send this CSR to a Commercial Certifying Authority, such as Verisign or GeoTrust, however, we already have an fully signed certificate on the existing server. For testing I will self sign this certificate. When this server is installed I will copy over the already signed SSL cert.

```
# openssl genrsa -des3 -out ca.key 1024
Generating RSA private key, 1024 bit long modulus
.............................++++++
.............................................++++++
e is 65537 (0x10001)
Enter pass phrase for ca.key:<testing>
Verifying - Enter pass phrase for ca.key:<testing>

# openssl req -new -x509 -days 365 -key ca.key -out ca.crt
Generating RSA private key, 1024 bit long modulus
.............................++++++
.............................................++++++
e is 65537 (0x10001)
Enter pass phrase for ca.key:
Verifying - Enter pass phrase for ca.key:

# openssl req -new -x509 -days 365 -key ca.key -out ca.crt
Enter pass phrase for ca.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:CA
State or Province Name (full name) [Some-State]:BC
Locality Name (eg, city) [Some-Locality]:SomeCity
Organization Name (eg, company) [Some-Organization Ltd]:Domain Inc
Organizational Unit Name (eg, section) [Some-Organizational]:Cert Division
Common Name (eg, YOUR name) [www.domain.com]:www.domain.com
Email Address [admin@domain.com]:root@domain.com
# mv www.key private/
# mv ca.key private/
# mv ca.crt certs/
# /usr/share/ssl/misc/sign www.csr
CA signing: www.csr -> www.crt:
Using configuration from ca.config
Enter pass phrase for /usr/share/ssl/private/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'CA'
stateOrProvinceName   :PRINTABLE:'BC'
localityName          :PRINTABLE:'SomeCity'
organizationName      :PRINTABLE:'Domain Inc'
organizationalUnitName:PRINTABLE:'Web Hosting'
commonName            :PRINTABLE:'www.domain.com'
emailAddress          :IA5STRING:'root@localhost'
Certificate is to be certified until Jan  7 16:53:26 2005 GMT (365 days)
Sign the certificate? [y/n]:y
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
CA verifying: www.crt <-> CA cert
www.crt: OK
```

```
# mv www.crt certs/
# rm -f www.csr
# chmod 750 /usr/share/ssl/private/
# chmod 400 /usr/share/ssl/certs/ca.crt
# chmod 400 /usr/share/ssl/certs/www.crt
# chmod 400 /usr/share/ssl/private/ca.key
# chmod 400 /usr/share/ssl/private/www.key
```

```
# vi /etc/sysconfig/httpd
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

remove the # from the line that reads:

OPTIONS="-DSSL"

```
# vi /etc/httpd/conf/httpd.conf
```

Uncomment the line:
LoadModule ssl_module modules/mod_ssl.so

```
# vi +110 /etc/httpd/conf/ssl.conf
```

Change
#SSLCertificateFile /etc/httpd/conf/ssl.crt/server-dsa.crt
to
SSLCertificateFile /usr/share/ssl/certs/www.crt

Several lines down change
#SSLCertificateKeyFile /etc/httpd/conf/ssl.key/server-dsa.key
to
SSLCertificateKeyFile /usr/share/ssl/private/www.key

```
# vi +92 /etc/httpd/conf/ssl.conf
```

Change log directories to correct locations:

ErrorLog /var/log/httpd/error_log
TransferLog /var/log/httpd/access_log

```
# vi +243 /etc/httpd/conf/ssl.conf
```

Change line to read:
CustomLog /var/log/httpd/ssl_request_log \

```
# vi +62 /etc/httpd/conf/ssl.conf
```

Change line to read:

SSLMutex

Restart Apache

```
# /etc/init.d/httpd restart
```

If you receive any errors check in /var/log/httpd/error_log

Mod_security Installation

There are several ways to install mod_security. Please check out the latest

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

documentation on their website – http://www.modsecurity.org
Download http://www.modsecurity.org/download/mod_security-1.7.4.tar.gz

```
# tar xvzf mod_security-1.7.4.tar.gz
# cd mod_security-1.7.4/apache2
# /usr/sbin/apxs -cia mod_security.c
```

At the end of the compile:

“[activating module `security' in /etc/httpd/conf/httpd.conf]”

Mod_Dosevasive Installation

What is mod_dosevasive?

"mod_dosevasive is an evasive maneuvers module for Apache to provide evasive action in the event of an HTTP DoS or DDoS attack or brute force attack. It is also designed to be a detection and network management tool, and can be easily configured to talk to ipchains, firewalls, routers, and etcetera. mod_dosevasive presently reports abuses via email and syslog facilities.

Detection is performed by creating an internal dynamic hash table of IP Addresses and URIs, and denying any single IP address from any of the following:

- Requesting the same page more than a few times per second
- Making more than 50 concurrent requests on the same child per second
- Making any requests while temporarily blacklisted (on a blocking list)

This method has worked well in both single-server script attacks as well as distributed attacks, but just like other evasive tools, is only as useful to the point of bandwidth and processor consumption (e.g. the amount of bandwidth and processor required to receive/process/respond to invalid requests), which is why it's a good idea to integrate this with your firewalls and routers for maximum protection.

This module instantiates for each listener individually, and therefore has a built-in cleanup mechanism and scaling capabilities. Because of this per-child design, legitimate requests are never compromised (even from proxies and NAT addresses) but only scripted attacks. Even a user repeatedly clicking on 'reload' should not be affected unless they do it maliciously. mod_dosevasive is fully tweakable through the Apache configuration file, easy to incorporate into your web server, and easy to use." [13]

Download
http://www.nuclearelephant.com/projects/dosevasive/mod_dosevasive.1.8.tar.gz

```
# tar xvzf mod_dosevasive.1.8.tar.gz
# cd /usr/tmp/mod_dosevasive.1.8.tar.gz
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
# /usr/sbin/apxs -cia mod_dosevasive20.c
# vi /etc/httpd/conf/httpd.conf
```

Add the following:

```
# Apache 2.x mod_dosevasive module from
# http://www.nuclearelephant.com/projects/dosevasive/
<IfModule mod_dosevasive20.c>
    DOSHashTableSize    3097
    DOSPageCount        2
    DOSSiteCount        50
    DOSPageInterval     1
    DOSSiteInterval     1
    DOSBlockingPeriod   10
</IfModule>
```

```
# /etc/init.d/httpd restart
```

Run test.pl to test

```
# chmod +x test.pl
# ./test.pl
```

You should see “HTTP/1.1 403 Forbidden” after several lines.

PHP Installation

I will be following the directions from [14]
Download php-4.3.4.tar.gz from one of the mirrors at http://www.php.net/downloads.php

```
# tar xvzf php-4.3.4.tar.gz
# cd php-4.3.4
export CFLAGS="-O2 -march=i686 -funroll-loops -D_REENTRANT -fPIC"
export LIBS="-lttf -lfreetype -ljpeg -lz -lnsl"
export EXTENSION_DIR=/usr/lib/php4
 ./configure \
--prefix=/usr \
--with-layout=GNU \
--with-config-file-path=/etc/httpd \
--enable-force-cgi-redirect \
--with-apxs2=/usr/sbin/apxs \
--with-exec-dir=/usr/bin \
--with-openssl \
--with-zlib \
--with-gd \
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
        --with-zlib-dir=/usr/include \
        --with-ttf \
        --with-png \
        --with-jpeg-dir=/usr \
        --with-png-dir=/usr \
        --with-freetype-dir=/usr \
        --with-pear=/usr/share/pear \
        --with-mysql=/usr/mysql \
        --with-mysql-sock=/tmp/mysql.sock \
        --disable-debug \
        --disable-rpath \
        --disable-posix \
        --enable-pic \
        --enable-discard-path \
        --enable-safe-mode \
        --enable-magic-quotes \
        --enable-bcmath \
        --enable-dio \
        --enable-track-vars \
        --enable-inline-optimization \
        --enable-memory-limit
        # make
        # make install
        # ln -sf /usr/lib/httpd/build/libtool /home/httpd/build/libtool
        # install -m0644 php.ini-dist /etc/httpd/php.ini
        # strip --strip-debug -R .comment /usr/lib/php4/*.so
```

The default php.ini file is set up for development purposes and not a secure web server. The following changes to /etc/httpd/php.ini could cause errors in your PHP code so it is recommended to make a backup copy of the file for troubleshooting purposes.

```
        # cp /etc/httpd/php.ini /etc/httpd/php.ini.bak
        # vi +85 /etc/httpd/php.ini
```

Change
  y2k_compliance = On
to
 y2k_compliance = Off

```
        # vi +117 /etc/httpd/php.ini
```

Change
 zlib.output_compression = Off
to
 zlib.output_compression = On

```
        # vi +154 /etc/httpd/php.ini
```

Change

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

allow_call_time_pass_reference = On
to
allow_call_time_pass_reference = Off

```
# vi +223 /etc/httpd/php.ini
```

Change
expose_php = On
to
expose_php = Off

```
# vi +277 /etc/httpd/php.ini
```

Change
display_errors = On
to
display_errors = Off

```
# vi +287 /etc/httpd/php.ini
```

Change
log_errors = Offs
to
log_errors = On

```
# vi +311 /etc/httpd/php.ini
```

Change
;html_errors = Off
to
html_errors = Off

```
# vi +332 /etc/httpd/php.ini
```

Change
;error_log = syslog
to
error_log = syslog

Add the following line to /etc/httpd/conf/httpd.conf

AddType application/x-httpd-php .php .php4 .php3 .phptml

```
# /etc/init.d/httpd restart
```

Giptables Installation

"GIPTables Firewall is a free set of shell scripts that helps you generate iptables rules for Linux 2.4.x and newer kernels. It is very easy to configure and at present, designed to run on hosts with one or two network cards. It doesn't require you to install any additional

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

components to make it work with your GNU/Linux system. All you need to set-up a very secure firewall for your GNU/Linux machines is iptables and GIPTables Firewall."[15]

The author likes using GIPTables because its very simple to configure and is installed in a very modular way. If a service is added to the server then it is quite simple to add another module to enable access to the new service.

First thing we need to install is netfilter (or iptables as most people refer to it as).

What is netfilter/iptables?

Netfilter and iptables are building blocks of a framework inside the Linux 2.4.x and 2.6.x kernel. This framework enables packet filtering, network addresses [and port] translation (NA[P]T) and other packet mangling. It is the re-designed and heavily improved successor of the previous Linux 2.2.x ipchains and Linux 2.0.x ipfwadm systems.
netfilter is a set of hooks inside the Linux kernel that allows kernel modules to register callback functions with the network stack. A registered callback function is then called back for every packet that traverses the respective hook within the network stack.
iptables is a generic table structure for the definition of rulesets. Each rule within an IP table consists out of a number of classifiers (iptables matches) and one connected action (iptables target).
netfilter, iptables and the connection tracking as well as the NAT subsystem together build the whole framework. [16]

Download http://www.netfilter.org/files/iptables-1.2.9.tar.bz2

```
# bunzip2 iptables-1.2.9.tar.bz2
# tar xvf iptables-1.2.9.tar
# cd iptables-1.2.9
# make BINDIR=/sbin LIBDIR=/usr/lib MANDIR=/usr/share/man
kernel_dir=/usr/src/linux
# make BINDIR=/sbin LIBDIR=/usr/lib MANDIR=/usr/share/man
kernel_dir=/usr/src/linux
```

Giptables Installation Directions from [17]

Download http://www.giptables.org/downloads/giptables-1.1.tar.gz

```
# tar xvzf giptables-1.1.tar.gz
# cd giptables-1.1
# ./install.sh

GIPTables Firewall home directory is /lib/giptables
Usage: /etc/rc.d/init.d/giptables {start|stop|restart|panic}
An installation log file has been created: /tmp/giptables-install-20040121043839.log
GIPTables Fireall v1.1 installation OK!
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Now we need to decide with default giptables configuration file (firewall rules script) that we are going to use. To start off with I am going to use giptables.conf.webserver and customize it for our purposes.

```
# cd /lib/giptables/conf
# cp giptables.conf.webserver giptables.conf.mybox
# ln -sf /lib/giptables/conf/giptables.conf.mybox /etc/giptables.conf
```

Customize /etc/giptables.conf

```
# vi +40 /etc/giptables.conf
```

Change
 INTERFACE0_IPADDR="0.0.0.0"
to
 INTERFACE0_IPADDR="192.168.0.50"

A few lines below that enter your ISP_PRIMARY_DNS_SERVER and ISP_SECONDARY_DNS_SERVER addresses. Enter the correct addresses here.

Because we are setting this up and testing on a private subnet (192.168.0.0/24) we need to enable access from this IP range.

```
# vi +160 /etc/giptables.conf
```

Comment out
 REFUSE_SPOOFING_IPADDR[5]="192.168.0.0/16"
and
 INTERFACE0_IN_REFUSE_SPOOFING[5]="yes"

These lines must be uncommented when the server is put into service online.

Since we will not be running FTP on this server we can remove FTP access from the configuration file

```
# vi +210 /etc/giptables.conf
```

Change
 ACCEPT_FTP="yes"
to
 ACCEPT_FTP="no"

Giptables Customization

If we have some specific IP Addresses that we wish to block from accessing any servers we can add them to to the /etc/rc.d/rc.giptables.blocked and then restart giptables

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
# /etc/init.d/giptables restart
```

We are going to create some customized giptables rules for the following items:

```
# /etc/init.d/giptables start
```

Snort Installation

The following instructions are primarily based on [18] and [19].

Please note that this is not intended as an in-depth review of setting up an Intrusion Detection System on your network. This is intended as an install on how to get Snort up and running. For more information please refer to the Additional Information section at the end of this paper.

Requirements:

Libpcap – Packet Capture library must be installed before installing Snort.

Download http://www.tcpdump.org/release/libpcap-0.8.1.tar.gz

```
# tar xvzf libpcap-0.8.1.tar.gz
# ./configure –-prefix=/usr --mandir=/usr/share/man
# make
# make install
```

Download http://www.snort.org/dl/snort-2.1.0.tar.gz

```
# groupadd -g 70 snort > /dev/null 2>&1 || :[1]
# useradd -c “Snort NIDS” -d /var/log/snort -g 70 -s /bin/false -u 70 snort > /
dev/null 2>&1 || :
# mkdir /etc/snort [2]
# mkdir /var/log/snort
# tar xvzf snort-2.1.0.tar.gz
# cd snort-2.1.0
# CFLAGS=”-O2 -march=i686 -funroll-loops”; export CFLAGS [1]
# ./configure \
# --prefix=/usr \
# --sysconfdir=/etc \
# --localstatedir=/var \
# --mandir=/usr/share/man \
# --with-openssl \
# --with-mysql=/usr/mysql
# make
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
# make install
# mkdir -p /var/log/snort
# chown -R snort.snort /var/log/snort
# install /etc/classification.config /etc/snort/
# install /etc/snort.conf etc/*.rules /etc/snort
# install /usr/tmp/snort-2.1.0/etc/unicode.map /etc/snort/
# install /usr/tmp/snort-2.1.0/etc/reference.config /etc/snort/
# install /usr/tmp/snort-2.1.0/etc/threshold.conf /etc/snort/
# chmod 0644 /etc/snort/*
# strip /usr/bin/snort
# cp /usr/tmp/snort-2.1.0/contrib/S99snort /etc/init.d/snort
# chown 0.0 /etc/init.d/snort
# chmod 700 /etc/init.d/snort
# vi +4 /etc/init.d/snort
```

Add the following lines (with the #):

```
# chkconfig: 2345 40 60
#
# description: Snort is a lightweight intrusion detection tool
```

```
# vi +12 /etc/init.d/snort
```

Change
```
SNORT_PATH=/usr/local/bin
```
to
```
SNORT_PATH=/usr/bin
```
Change (next line)
```
CONFIG=/usr/local/share/snort/snort.conf
```
to
```
CONFIG=/etc/snort/snort.conf
```

```
# vi +19 /etc/init.d/snort
```
Change
```
SNORT_GID=nogroup
```
to
```
SNORT_GID=snort
```

```
# vi +32 /etc/init.d/snort
```
Add '-u snort' just after $SNORT_PATH/snort so the line should look like:
$SNORT_PATH/snort -u snort -c $CONFIG -i $IFACE -g $SNORT_GID $OPTIONS

```
# vi +44 /etc/snort/snort.conf
```
Change
```
# var HOME_NET any
```
to

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

var HOME_NET 192.168.0.0/24

```
# vi +107 /etc/snort/snort.conf
```

Change
var RULE_PATH ../rules
to
var RULE_PATH /etc/snort/rules

MySQL Snort Configuration

We need to create the default snort databases, etc in MySQL. Here's what we do:

Login as sqladmin

```
# ./mysql -u sqladmin -p
```

You could type the mysql password in the above line but that would expose the password to your bash history file.

```
mysql> create database snort;
mysql> connect snort;
mysql> select password('123test');
+---------------------+
| password('123test') |
+---------------------+
| 4f4ef7c05bdb5462    |
+---------------------+
1 row in set (0.00 sec)

mysql> grant CREATE,INSERT,DELETE,UPDATE,SELECT on snort.* to
snort@localhost identified by password '4f4ef7c05bdb5462';
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
```

Above reloads privileges table so you do not need to restart mysql.

```
# /usr/mysql/bin/mysql -usnort -p snort< ./create_mysql
# vi +451 /etc/snort/snort.conf
```

Change
output database: log, mysql, user=root password=test dbname=db host=localhost
to
output database: alert, mysql, user=snort password=123test dbname=snort host=localhost

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Now we need to verify if snort is working correctly and logging to the MySQL database.

From another machine nmap the server

```
# nmap -sF 192.168.0.50
```

On the server:

```
# ./mysql -usnort -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 6 to server version: 4.0.17-max-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use snort
Database changed
mysql> select count(*) from event;
+----------+
| count(*) |
+----------+
|     1667 |
+----------+
1 row in set (0.00 sec)
mysql>
```

What you want to see is a number in under count that is > 0.

ACID Installation

ACID (Analysis Console for Intrusion Databases) is a "PHP-based analysis engine to search and process a database of security events generated by various IDSes, firewalls, and network monitoring tools"[20] I will set this up to allow access to the Snort logs in a much simpler web based interface from a SSL encrypted web page with password authentication required.

The installation documentation I am using for this is from [21]

Requirements

ADODB

Download http://phplens.com/lens/dl/adodb410.tgz

```
# mv adodb410.tgz /home/httpd/htdocs
# cd /home/httpd/htdocs
# tar xvzf adodb410.tgz
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
# rm -f adodb410.tgz
```

PHPLOT

Download http://unc.dl.sourceforge.net/sourceforge/phplot/phplot-4.4.6.tar.gz

```
# mv phplot-4.4.6.tar.gz /home/httpd/htdocs/
# cd /home/httpd/htdocs
# tar xvzf phplot-4.4.6.tar.gz
# rm -f phplot-4.4.6.tar.gz
```

JPGRAPH

Download http://www.aditus.nu/jpgraph/downloads/jpgraph-1.13.tar.gz

```
# mv jpgraph-1.13.tar.gz /home/httpd/htdocs
# cd /home/httpd/htdocs
# tar xvzf jpgraph-1.13.tar.gz
# ln -sf jpgraph-1.13 jpgraph
# rm -f jpgraph-1.13.tar.gz
# cd jpgraph/src
# mv *.php ../
```

ACID Installation

Download http://acidlab.sourceforge.net/acid-0.9.6b23.tar.gz

```
# mv /usr/tmp/acid-0.9.6b23.tar.gz /home/httpd/htdocs/
# cd /home/httpd/htdocs
# tar xvzf acid-0.9.6b23.tar.gz
# rm -f acid-0.9.6b23.tar.gz
# cd acid
# vi acid_conf.php
```

Change (line 12)
 $Dblib_path = “”;
to
 $Dblib_path = “/home/httpd/htdocs/adodb”;

Change (line 32-36)
 $alert_dbname   = "snort_log";
 $alert_host   = "localhost";
 $alert_port   = "";
 $alert_user   = "root";
 $alert_password = "mypassword";
to

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
$alert_dbname   = "snort";
$alert_host     = "localhost";
$alert_port     = "";
$alert_user     = "snort";
$alert_password = "123test";
```

Change lines 39 to 43

```
$archive_dbname   = "snort_archive";
$archive_host     = "localhost";
$archive_port     = "";
$archive_user     = "root";
$archive_password = "mypassword";
```

to

```
$archive_dbname   = "snort";
$archive_host     = "localhost";
$archive_port     = "";
$archive_user     = "snort";
$archive_password = "123test";
```

Change line 69

```
$ChartLib_path = "";
```

to

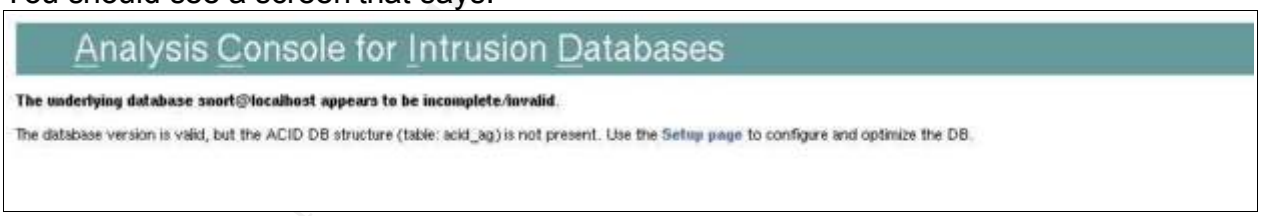
```
$ChartLib_path = "/home/httpd/htdocs/jpgraph";
```

Restart Apache

```
# /etc/init.d/httpd restart
```

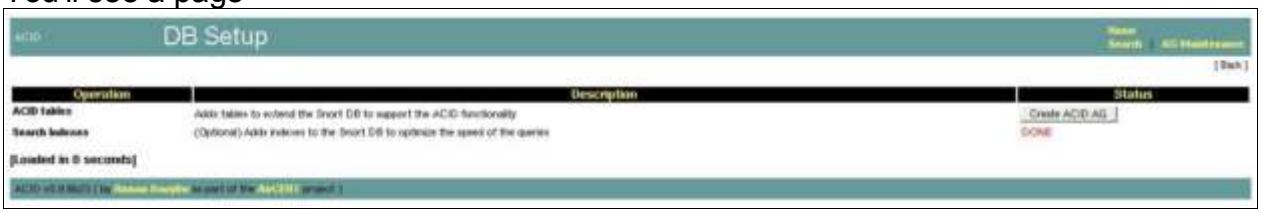
Point your browser to http://192.168.0.50/acid/acid_main.php

You should see a screen that says:



Click on the 'Setup page link'

You'll see a page



© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Click on 'Create ACID AG' button

Open your browser to http://192.168.0.50/acid and you should see a proper ACID page.

Authenticated access to the acid pages

To heighten security we will require a username and password to access the ACID web tools.

```
        # mkdir /home/httpd/htdocs/passwords
        # /usr/sbin/httpasswd -c /home/httpd/htdocs/passwords acid
        # /usr/sbin/htpasswd -c /home/httpd/passwords/passwords acid
        New password:
        Re-type new password:
        Adding password for user acid
# vi /etc/httpd/conf/httpd.conf
```

Add the following:

```
 <Directory "/home/httpd/htdocs/acid">
        AuthType Basic
        AuthName "SnortIDS"
        AuthUserFile /home/httpd/passwords/passwords
        Require user acid
 </Directory>
```

Also uncomment the line that reads:

```
# LoadModule auth_module        modules/mod_auth.so
```

To ensure that there is no eavesdropping on your transmissions through the ACID webpage it is recommended that you only use https://192.168.0.50/acid to access the site.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Time Synchronization

A critical part of maintaining the integrity of your server and being able to have proper valid log files is to ensure that the time on your server is always correct. If you have to contact the authorities regarding an intrusion and try to explain to them that the cracker broke in "sometime between 3:00AM – 4:00AM" because the time is not correct on your server then they're probably not going to be able to help you.

To solve this problem we are going to install ntpd from http://www.ntp.org.

Download ntp-4.2.0.tar.gz from one of the ntp mirrors at http://www.ntp.org/downloads.html

```
# tar xvzf ntp-4.2.0.tar.gz
# cd ntp-4.2.0
# ./configure –prefix=/usr –sysconfdir=/etc –mandir=/usr/share/man
# make
# make check
# make install
# vi /etc/ntp.conf
```

Add the following:[22]

restrict default notrust nomodify ignore
restrict 127.0.0.1
restrict 136.159.2.1 mask 255.255.255.255 nomodify notrap noquery
server 136.159.2.1 prefer
server 127.127.1.0
fudge 127.127.1.0 stratum 10
driftfile /etc/ntp.drift
broadcastdelay 0.008

AIDE

You've installed your complete system and its been running on the internet for six months. Then you start noticing some strange behavior on the server such as log files going missing, directories growing in size and strange connections to the server. Your first assumption is that your system has been cracked but how can you be sure what has been changed?

You need a file integrity checker. For this we will be using AIDE (http://sourceforge.net/projects/aide)

What is AIDE?

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

AIDE (Advanced Intrusion Detection Environment) is a free replacement for Tripwire. It does the same things as the semi-free Tripwire and more.[23]

Requirements

Mhash is required to build AIDE. Unfortunately, aide requires a static build of the mhash libraries so you cannot just install the two mhash and mhash-devel RPMs from the Openna CD-ROM.

Download the source file from one of the mirrors at http://sourceforge.net/projects/mhash/

```
# tar xvzf mhash-0.8.18.tar.gz
# cd mhash-0.8.18
# ./configure –prefix=/usr –sysconfdir=/etc –mandir=/usr/share/man –enable-static=yes
# make
# make check
# make install
```

Download the source file from the project homepage

```
# tar xvzf aide-0.10.tar.gz
# cd aide-0.10
# ./configure –prefix=/usr –sysconfdir=/etc –mandir=/usr/share/man
# make
# make install
```

Now we need to create /etc/aide.conf file[24]. I've customized the file for my particular configuration.

```
# vi /etc/aide.conf
```

Add the following:

```
Rule = p+i+u+g+n+s+md5

/etc p+i+u+g
/sbin Rule
/bin Rule
/etc/httpd/conf Rule
/usr/bin Rule
/usr/local Rule
!/var/spool/.*
!/var/log/.*
!/var/lock/.*
!/var/run/.*
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
!/var/qmail/supervise/qmail-pop3d/.*
!/var/qmail/supervise/qmail-smtpd/.*
!/var/qmail/supervise/qmail-pop3d/.*
!/tmp/.*
```

You may find other directories over time which change quite often which give you false alerts. Add those to the above file as needed.

```
# chmod 600 /etc/aide.conf
```

For more instructions on how to configure AIDE do the following:

```
# man aide.conf
```

Initialize AIDE

```
# aide –init
```

Now you must copy the aide binary (/usr/bin/aide) and the aide database file (/etc/aide.db.new) off the system and put it on a write protected floppy or CD-ROM.

Its also highly recommended that you put a second copy of the media with the files in a secure location offsite from the server.

Here's a quick example of what happens when a file has been changed:

Temporarily we're going to change the permissions of /bin/vi from -r—r—r-- to -rwxrwxrwx and run an aide check to see the alert.

```
# chmod 777 /bin/vi
# cp /etc/aide.db.new /etc/aide.db
# aide –check
```

```
AIDE found differences between database and filesystem!!
Start timestamp: 2004-02-03 19:16:19
Summary:
Total number of files=21661,added files=1,removed files=0,changed files=1

Added files:
added:/etc/aide.db
Changed files:
changed:/bin/vi
Detailed information about changes:

File: /bin/vi
  Permissions: -r--r--r--                    , -rwxrwxrwx
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Change the permissions on /bin/vi back

```
# chmod 444 /bin/vi
```

Final Cleanup

Chattr of key files

We will use the 'chattr' command to change the attributes on some key configuration files to protect them from any changes. We will add the immutable bit to these files with the -i flag.

```
# chattr +i /etc/passwd
# chattr +i /etc/shadow
# chattr +i /etc/group
# chattr +i /etc/gshadow
# chattr +i /etc/resolv.conf
# chattr +i /chroot/named/etc/named.conf
```

Remove Development RPMs

If a cracker breaks into a server, the first thing he is going to do is to install rootkits, trojan packages, etc on the system to maintain his access to the server. To stop him from being able to configure and install these tools we will remove the development RPMs.
rpm -e autoconf automake binutils bison byacc cdecl cpp cracklib-devel db4-devel dev86 file flex gcc gcc-c++ gdbm-devel glibc-devel gmp kernel-headers libelf-devel libstdc++-devel libtool libtool-libs m4 make ncurses-devel pam-devel patch pcre-devel popt-devel pwdb-devel rpm-build rpm-devel utempter-devel

Removal of Downloaded Files

Now that we've completed the installation of all files we will remove all of the source files.

```
# cd /usr/tmp
# rm -rf *
```

Maintenance of the server

Autoupdate

Openna offers a great free package known as autoupdate which, as the name implies, automatically updates your system with the latest RPMs from the Openna FTP site.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

I'll be using the instructions from [25]

Download the autoupdate rpm from
ftp://ftp.openna.com/pub/linux/1.0/UPDATES/RPMS/autoupdate-5.2.16-1.i686.rpm

```
# rpm -Uvh autoupdate-5.2.16-1.i686.rpm
```

If you receive an error about not having perl-DB_File then install it from the Openna CD-ROM.

Make sure that the giptables firewall allows outgoing FTP connections.

```
# vi /etc/giptables.conf
```

Check that ACCEPT_FTP="yes" is enabled and you have enabled "FTP outgoing client request"

```
# vi +41 /etc/autoupdate.d/autoupdate.conf
```

Change
 DoKernel=1
to
 DoKernell=0

Automatically updating the kernel is a dangerous thing to do because if any problems occur they could make the machine unbootable.

Autoupdate will automatically check and install updates now every day at 4:00AM.

Any logs will got to /var/log/messages.

To test that autoupdate is working correctly we can run it manually.

```
# /usr/sbin/autodld
```

Mailing Lists and other sources of information

Openna has several mailing lists that would be worth subscribing to.
http://smtp.openna.com/mailman/listinfo/openna-announce – Openna Announcements
http://smtp.openna.com/mailman/listinfo/openna-linux – Openna Linux Discussions

http://www.us-cert.gov/cas/bulletins/index.html – Cyber Security Bulletins from the US Computer Emergency Readiness Team (CERT)

http://httpd.apache.org/lists.html – Apache Web Server Mailing Lists. There are several

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

listed. At minimum you should subscribe to the "Apache Server Announcements" list.

http://www.giptables.org/support.html#lists – Giptables Mailing Lists.

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Appendix A BIND Configuration File – named.conf

```
// Authorized source addresses.
acl "trusted" {
        localhost;
};
// Known fake source addresses shouldn't be replied to.
acl "bogon" {
        0.0.0.0/8;
        1.0.0.0/8;
        2.0.0.0/8;
        192.0.2.0/24;
        224.0.0.0/3;
        169.254.0.0/16;
        // Enterprise networks may or may not be bogus.
        10.0.0.0/8;
        172.16.0.0/12;
        // 192.168.0.0/16;
};
options {
        directory "/etc/named";
        allow-transfer { 192.168.0.5; };
        allow-query { any; };
        allow-recursion { trusted; };
        blackhole { bogon; };
        tcp-clients 32;
        forwarders { none; };
        version "BIND Baby";
};
logging {
        category lame-servers { null; };
};
// Root server hints
zone "." { type hint; file "db.cache"; };
// Provide a reverse mapping for the loopback address 127.0.0.1/24
zone "localhost" {
        type master;
        file "db.localhost";
        notify no;
};
zone "0.0.127.in-addr.arpa" {
        type master;
        file "0.0.127.in-addr.arpa";
        notify no;
};
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
// We are the master for domain.com
zone "domain.com" {
type master;
file "db.domain.com";
allow-query { any; };
};
// Provide a reverse mapping for domains network 192.168.0.0/24
zone "0.168.192.in-addr.arpa" {
	type master;
	file "0.168.192.in-addr.arpa";
	allow-query { any; };
};
```

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Appendix B – named Initialization Script

```
#!/bin/bash

# This shell script takes care of starting and stopping named.
#
# chkconfig: 2345 55 45
# description: Named (BIND) is a Domain Name Server (DNS) that is used \
# to resolve host names to IP addresses.
#
# processname: named

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/named ] ; then
. /etc/sysconfig/named
fi

# Check that networking is up.
[ "${NETWORKING}" = "no" ] && exit 0

# If Named is not available stop now.
[ -f /usr/sbin/named ] || exit 0
[ -f "${ROOTDIR}"/etc/named.conf ] || exit 0

# Path to the Named binary.
named=/usr/sbin/named

RETVAL=0
prog="Named"

start() {
echo -n $"Starting $prog: "
if [ -n "${ROOTDIR}" -a "x${ROOTDIR}" != "x/" ]; then
OPTIONS="${OPTIONS} -t ${ROOTDIR}"
fi
daemon $named -u named ${OPTIONS}
RETVAL=$?
echo
[ $RETVAL -eq 0 ] && touch /var/lock/subsys/named
return $RETVAL
}

stop() {
echo -n $"Shutting down $prog: "
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
killproc $named
RETVAL=$?
echo
[ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/named
return $RETVAL
}
# See how we were called.
case "$1" in
start)
start
;;
stop)
stop
;;
status)
status $named
RETVAL=$?
;;
restart)
stop
start
RETVAL=$?
;;
condrestart)
if [ -f /var/lock/subsys/named ]; then
stop
start
RETVAL=$?
fi
;;
reload)
/usr/sbin/rndc reload >/dev/null 2>&1 || /usr/bin/killall -HUP $named
return $RETVAL
;;
probe)
/usr/sbin/rndc reload >/dev/null 2>&1 || echo start
return $RETVAL
;;
*)
echo $"Usage: $0 {start|stop|status|restart|condrestart|reload|probe}"
exit 1
esac
exit $RETVAL
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Appendix C – Apache configuration options

```
# export CFLAGS="-O2 -march=i686 -funroll-loops -D_REENTRANT -D_SINGLE_LISTEN_UNSERIALIZED_ACCEPT -fPIC"
        # ./configure --prefix=/etc/httpd \
        > --exec-prefix=/usr \
        > --bindir=/usr/bin \
        > --sbindir=/usr/sbin \
        > --mandir=/usr/share/man \
        > --sysconfdir=/etc/httpd/conf \
        > --includedir=/usr/include/httpd \
        > --libexecdir=/usr/lib/httpd/modules \
        > --datadir=/home/httpd \
        > --localstatedir=/var \
        > --with-mpm=prefork \
        > --enable-access=shared \
        > --enable-actions=shared \
        > --enable-alias=shared \
        > --enable-auth=shared \
        > --enable-auth-dbm=shared \
        > --enable-auth-digest=shared \
        > --enable-autoindex=shared \
        > --enable-cern-meta=shared \
        > --enable-cgi=shared \
        > --enable-cgid=shared \
        > --enable-dav=shared \
        > --enable-dav-fs=shared \
        > --enable-dir=shared \
        > --enable-env=shared \
        > --enable-expires=shared \
        > --enable-file-cache=shared \
        > --enable-headers=shared \
        > --enable-include=shared \
        > --enable-log-config=shared \
        > --enable-mime=shared \
        > --enable-mime-magic=shared \
        > --enable-negotiation=shared \
        > --enable-rewrite=shared \
        > --enable-setenvif=shared \
        > --enable-speling=shared \
        > --enable-ssl=shared \
        > --enable-unique-id=shared \
        > --enable-usertrack=shared \
        > --enable-vhost-alias=shared \
        > --enable-suexec=shared \
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
> --with-suexec-caller=www \
> --with-suexec-docroot=/home/httpd \
> --with-suexec-logfile=/var/log/httpd/suexec.log \
> --with-suexec-bin=/usr/sbin/suexec \
> --with-suexec-uidmin=500 --with-suexec-gidmin=500 \
> --disable-auth-anon \
> --disable-charset-lite \
> --disable-disk-cache \
> --disable-mem-cache \
> --disable-cache \
> --disable-deflate \
> --disable-ext-filter \
> --disable-case-filter \
> --disable-case-filter-in \
> --disable-example \
> --disable-proxy \
> --disable-proxy-connect \
> --disable-proxy-ftp \
> --disable-proxy-http \
> --disable-status \
> --disable-asis \
> --disable-info \
> --disable-imap \
> --disable-userdir \
> --with-z \
> --with-ssl \
> --with-suexec
```

© SANS Institute 2004, Author retains full rights.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Appendix D Apache Initialization File

```
#!/bin/bash

# This shell script takes care of starting and stopping Apache.
#
# chkconfig: 345 85 15
# description: Apache is a World Wide Web server. It is used to serve \
#              HTML files and CGI.
#
# processname: httpd
# config: /etc/httpd/conf/httpd.conf
# pidfile: /var/run/httpd.pid

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source for additional options if we have them.
if [ -f /etc/sysconfig/httpd ] ; then
        . /etc/sysconfig/httpd
fi

# This will prevent initlog from swallowing up a pass-phrase prompt if
# mod_ssl needs a pass-phrase from the user.
INITLOG_ARGS=""

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# If Apache is not available stop now.
[ -f /usr/sbin/httpd ] || exit 0

# Path to the Apache apachectl script and server binary.
apachectl=/usr/sbin/apachectl
httpd=/usr/sbin/httpd

RETVAL=0
prog="httpd"

start() {
        echo -n $"Starting $prog: "
        daemon $httpd $OPTIONS
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && touch /var/lock/subsys/httpd
        return $RETVAL
}

stop() {
        echo -n $"Shutting down $prog: "
        killproc $httpd
        RETVAL=$?
        echo
        [ $RETVAL = 0 ] && rm -f /var/lock/subsys/httpd /var/run/httpd.pid
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  status)
        status $httpd
        RETVAL=$?
        ;;
  restart)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/run/httpd.pid ] ; then
                stop
                start
                RETVAL=$?
        fi
        ;;
  *)
        echo $"Usage: $0 {start|stop|status|restart|condrestart}"
        exit 1
esac
exit $RETVAL
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Appendix E – Apache Initialization file

```
### Section 1: Global Environment
#
ServerTokens OS
ServerRoot "/etc/httpd"
PidFile /var/run/httpd.pid

Timeout 60
KeepAlive Off
MaxKeepAliveRequests 0
KeepAliveTimeout 10

# Prefork MPM
#
<IfModule prefork.c>
StartServers       5
MaxClients         512
ServerLimit        1024
MinSpareServers    5
MaxSpareServers    10
MaxRequestsPerChild 0
</IfModule>

Listen 127.0.0.1:80
Listen 127.0.0.1:443

# Dynamic Shared Object (DSO) Support
#
LoadModule access_module        modules/mod_access.so
#LoadModule auth_module         modules/mod_auth.so
#LoadModule auth_dbm_module     modules/mod_auth_dbm.so
#LoadModule auth_digest_module  modules/mod_auth_digest.so
#LoadModule file_cache_module   modules/mod_file_cache.so
LoadModule include_module       modules/mod_include.so
LoadModule log_config_module    modules/mod_log_config.so
LoadModule env_module           modules/mod_env.so
LoadModule mime_magic_module    modules/mod_mime_magic.so
#LoadModule cern_meta_module    modules/mod_cern_meta.so
#LoadModule expires_module      modules/mod_expires.so
#LoadModule headers_module      modules/mod_headers.so
#LoadModule usertrack_module    modules/mod_usertrack.so
#LoadModule unique_id_module    modules/mod_unique_id.so
LoadModule setenvif_module      modules/mod_setenvif.so
#LoadModule ssl_module          modules/mod_ssl.so
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
LoadModule mime_module          modules/mod_mime.so
#LoadModule dav_module          modules/mod_dav.so
LoadModule autoindex_module     modules/mod_autoindex.so
LoadModule cgi_module           modules/mod_cgi.so
#LoadModule dav_fs_module       modules/mod_dav_fs.so
#LoadModule vhost_alias_module  modules/mod_vhost_alias.so
#LoadModule negotiation_module  modules/mod_negotiation.so
LoadModule dir_module           modules/mod_dir.so
#LoadModule actions_module      modules/mod_actions.so
#LoadModule speling_module      modules/mod_speling.so
LoadModule alias_module         modules/mod_alias.so
LoadModule rewrite_module       modules/mod_rewrite.so
#LoadModule perl_module         modules/mod_perl.so
#LoadModule php4_module         modules/libphp4.so

### Section 2: 'Main' server configuration
#
User www
Group www

ServerAdmin root@localhost
ServerName localhost
UseCanonicalName Off

DocumentRoot "/home/httpd/htdocs"
<Directory />
   Options None
   AllowOverride None
   Order deny,allow
   Deny from all
</Directory>

<Files .pl>
   Options None
   AllowOverride None
   Order deny,allow
   Deny from all
</Files>

<IfModule mod_file_cache.c>
<IfModule mod_include.c>
   Include /etc/httpd/mmap.conf
</IfModule>
</IfModule>

<IfModule mod_dir.c>
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
   DirectoryIndex index.htm index.html index.php default.php index.shtml index.php3
</IfModule>

<IfModule mod_mime.c>
   TypesConfig /etc/httpd/conf/mime.types
   AddEncoding x-compress Z
   AddEncoding x-gzip gz tgz
   AddType application/x-tar .tgz
   AddType application/x-httpd-php .php
   AddType application/x-httpd-php .php3
   AddType application/x-httpd-php .shtml
   AddType application/x-httpd-php-source .phps
</IfModule>

DefaultType text/plain

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

HostnameLookups Off

LogLevel info
ErrorLog /var/log/httpd/error_log
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\"" combined
CustomLog /var/log/httpd/access_log combined

ServerSignature Off

<IfModule mod_alias.c>
Alias /icons/ "/home/httpd/icons/"
<Directory "/home/httpd/icons">
   Options None
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>

ScriptAlias /cgi-bin/  "/home/httpd/cgi-bin/"
 <Directory "/home/httpd/cgi-bin">
   Options None
   AllowOverride None
   Order allow,deny
   Allow from all
 </Directory>
</IfModule>
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
<IfModule mod_autoindex.c>
   IndexOptions FancyIndexing
   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 /icons/unknown.gif
   ReadmeName README.html
   HeaderName HEADER.html
   IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
</IfModule>

ErrorDocument 400 "Server could not understand this request."
ErrorDocument 401 "Server could not verify your access authorization."
ErrorDocument 403 "Access Forbidden -- Go away."
ErrorDocument 404 "Error! The requested page do not exist"
ErrorDocument 405 "Method not allowed for the requested URL."
ErrorDocument 408 "Server closed the network connection."
ErrorDocument 410 "Requested URL no longer available."
ErrorDocument 411 "Requested method requires a valid header."
ErrorDocument 412 "Precondition request failed positive evaluation."
ErrorDocument 413 "Method not allowed for the data transmitted."
ErrorDocument 414 "Requested URL exceeds the capacity limit."
ErrorDocument 415 "Server temporarily unavailable -- Maintenance downtime."
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
ErrorDocument 500 "Server encountered an internal error."
ErrorDocument 501 "Server does not support the action requested."
ErrorDocument 502 "Proxy server received an invalid response."
ErrorDocument 503 "Server temporarily unavailable -- Maintenance downtime."
ErrorDocument 506 "Access not possible."

<IfModule mod_setenvif.c>
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
BrowserMatch "Microsoft Data Access Internet Publishing Provider" redirect-carefully
BrowserMatch "^WebDrive" redirect-carefully
</IfModule>


### Section 3: Virtual Hosts
#
NameVirtualHost 127.0.0.1:80

<VirtualHost 127.0.0.1:80>
ServerAdmin root@localhost
ServerName localhost
DocumentRoot "/home/httpd/htdocs"

<Directory "/home/httpd/htdocs">
   Options Indexes MultiViews
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>

ErrorLog /var/log/httpd/error_log
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\""
TransferLog /var/log/httpd/access_log
</VirtualHost>


## SSL Global Context
#
<IfModule mod_ssl.c>
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl

SSLPassPhraseDialog     builtin
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

```
SSLSessionCache         none
SSLSessionCacheTimeout  300
SSLMutex                sem
SSLRandomSeed startup   file:/dev/urandom 1024
SSLRandomSeed connect   file:/dev/urandom 1024

## SSL Virtual Host Context
#
NameVirtualHost 127.0.0.1:443

<VirtualHost 127.0.0.1:443>
ServerAdmin root@localhost
ServerName localhost
DocumentRoot "/home/httpd/htdocs"

<Directory "/home/httpd/htdocs">
   Options Indexes MultiViews
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>

ErrorLog /var/log/httpd/error_log
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-agent}i\""
TransferLog /var/log/httpd/access_log

SSLEngine on

SSLCipherSuite ALL:!ADH:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
SSLCertificateFile      /usr/share/ssl/certs/www.crt
SSLCertificateKeyFile   /usr/share/ssl/private/www.key
SSLVerifyClient none
SSLVerifyDepth  10

SetEnvIf User-Agent ".*MSIE.*" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0

CustomLog /var/log/httpd/ssl_request_log \
        "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

</VirtualHost>
</IfModule>
```


© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

1 Miller,Todd. “Sudo Main Page.” URL: http://www.courtesan.com/sudo/ (19 Jan 2004)

2 Rivest, R. "RFC 1321 (rfc132) – The MD5 Message-Digest Algorithm." April 1992. URL: http://www.faqs.org/rfcs/rfc1321.html.(15 Jan 2004)

3 Cons,Lionel. “CERN Security Handbook.”http://consult.cern.ch/writeup/security/security_3.html. 12 Dec 1996.

4 Mourani, Gerhard. "The Official OpenNA Linux Installation Guide. "URL: http://www.openna.com/products/os/installation_guide/introduction.php?e=0,13. (10 Jan 2004)

5 Mourani, Gerhard. Securing and Optimizing Linux: The Hacking Solution. Montreal: Open Network Architecture Inc, 2002. 381.

6 Mourani, Gerhard. "How to Build, Install, Secure & Optimize BIND & DNS." URL: http://www.openna.com/documentations/articles/bind/index.php?e=0,1 (12 Jan 2004)

7 Canucklehead. “Da LAN Tech: Bind Security.” 23 Aug 2003. URL: http://www.dalantech.com/boards/showflat-Cat--Number-33017-Main-33017.html (13 Jan 2004)

8 Friedl,Steve.”Go Directly to Jail.” Linux Magazine December 2002. URL:http://www.linux-mag.com/2002-12/chroot_01.html. 20 Jan 2004.

9 Bernstein, D.J. “The qmail security guarantee. URL: http://cr.yp.to/qmail/guarantee.html (30 Jan 2004)

10 Sill, Dave. "Life with qmail." 26 Jan 2004. URL: http://www.lifewithqmail.org/lwq.html (30 Jan 2004)

11 Unknown. "Vpopmail Installation." 07 Mar 2003. URL: http://www.inter7.com/vpopmail/install.txt. (30 Jan 2004)

12 Mourani, Gerhard. "How to Build, Install, Secure & Optimize Apache 2.x." URL: http://www.openna.com/documentations/articles/apache/index.php. (25 Jan 2004)

13 Zdziarski,Jonathan. “Nuclear Elephant: mod_dosevasive.” URL:http://www.nuclearelephant.com/projects/dosevasive/ (25 Jan 2004)

14 Mourani, Gerhard. "How to Build, Install, Secure & Optimize PHP."URL: http://www.openna.com/documentations/articles/php/index.php. (25 Jan 2004)

15 Pascalau, Adrian. “Welcome to Giptables Firewall Homepage.” URL: http://www.giptables.org/. (31 Jan 2004.)

16 Welte, Harald. "netfilter/iptables project homepage." URL: http://www.netfilter.org/index.html (31 Jan 2004)

17 Pascalau, Adrian and Mourani, Gerhard. "GIPTables Firewall Installation Guide." 08 Jun 2002. URL: http://www.giptables.org/installation.html (31 Jan 2004)

18 Mourani, Gerhard. Securing and Optimizing Linux: The Hacking Solution. Montreal: Open Network Architecture Inc, 2002. 454-467.

19 Harper, Patrick. “Snort Install Manual.” Snort, Apache, PHP, MySQL and Acid Install on RH9.0. 09 Nov 2003. URL: http://www.internetsecurityguru.com/documents/snort_acid_rh9.pdf (20 Jan 2004)

20 Danyliw,Roman. “Analysis Console for Intrusion Databases (ACID).” 07 Feb 2004. URL: http://acidlab.sourceforge.net/ (07 Feb 2004).

21 Harper, Patrick. “Snort Install Manual.” Snort, Apache, PHP, MySQL, and ACID Install on RH9.0. Version 5. 09 Nov 2003.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

http://www.internetsecurityguru.com/documents/snort_acid_rh9.pdf (01 Feb 2004)

22 Mourani, Gerhard. Securing and Optimizing Linux: The Hacking Solution. Montreal: Open Network Architecture Inc, 2002. 514.

23 Lehti, Rami. "AIDE – Advanced Intrusion Detection Environment" URL: http://www.cs.tut.fi/~rammer/aide.html (03 Feb 2004)

24 Parker, Chris. "AIDE Integrity Checking" 28 Sep 2000. URL: http://www.linuxsecurity.com/tips/tip-9.html (03 Feb 2004)

25 Mourani, Gerhard. "OpenNA Linux AutoUpdate." URL:http://www.openna.com/downloads/autoupdate.php. (05 Feb 2004)

© SANS Institute 2004, Author retains full rights.

Last Updated: May 21st, 2013

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS Malaysia @ MCMC 2013	Cyberjaya, MY	Jun 03, 2013 - Jun 08, 2013	Live Event
SANS Pen Test Berlin 2013	Berlin, DE	Jun 03, 2013 - Jun 08, 2013	Live Event
Industial Control Systems Security Training - Houston	Houston, TXUS	Jun 10, 2013 - Jun 15, 2013	Live Event
Security Impact of IPv6 Summit 2013	Washington, DCUS	Jun 14, 2013 - Jun 16, 2013	Live Event
SANSFIRE 2013	Washington, DCUS	Jun 14, 2013 - Jun 22, 2013	Live Event
SANS Canberra 2013	Canberra, AU	Jul 01, 2013 - Jul 13, 2013	Live Event
Digital Forensics & Incident Response Summit 2013	Austin, TXUS	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS London Summer 2013	London, GB	Jul 09, 2013 - Jul 16, 2013	Live Event
SANS Rocky Mountain 2013	Denver, COUS	Jul 14, 2013 - Jul 20, 2013	Live Event
SANS Mumbai 2013	Mumbai, IN	Jul 22, 2013 - Jul 27, 2013	Live Event
SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification	Washington, DCUS	Jul 22, 2013 - Jul 26, 2013	Live Event
SANS San Francisco 2013	San Francisco, CAUS	Jul 29, 2013 - Aug 03, 2013	Live Event
SANS SEC 560: Network Penetration Testing @ Bangalore 2013	Bangalore, IN	Aug 05, 2013 - Aug 10, 2013	Live Event
SANS Boston 2013	Boston, MAUS	Aug 05, 2013 - Aug 10, 2013	Live Event
Critical Security Controls Summit	Washington, DCUS	Aug 12, 2013 - Aug 18, 2013	Live Event
Industrial Control Systems Security Training - DC	Washington, DCUS	Aug 12, 2013 - Aug 16, 2013	Live Event
SANS Thailand 2013	Bangkok, TH	Aug 19, 2013 - Aug 31, 2013	Live Event
SANS Virginia Beach 2013	Virginia Beach, VAUS	Aug 19, 2013 - Aug 30, 2013	Live Event
Mobile Device Security Summit 2013	OnlineCAUS	May 30, 2013 - Jun 06, 2013	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced