Get Acquainted with Linux Security and Optimization System

A guide for information system, configuration, optimization and network security professionals.

50 Quintin suite 101
St-Laurent H4N 3A5
Quebec Canada
Mail: **gmourani@videotron.ca**
Author: Gerhard Mourani
Version: 1.1
Last Revised: January 11, 2000

Overview

Introduction

Part I Installation-Related Reference

Chapter 1 Introduction to Linux
Chapter 2 Installation of your Linux Server

Part II Security and optimization-Related Reference

Chapter 3 General System Security
Chapter 4 General System Optimization

Part III Kernel-Related Reference

Chapter 5 Configuring and Building Kernels

Part IV Networking-Related Reference

Chapter 6 TCP/IP Network Management
Chapter 7 Networking Firewall

Part V Software's-Related Reference

Chapter 8 Compilers Functionality
Chapter 9 Securities Software
Chapter 10 Servers Software

Part VI Backup-Related reference

Chapter 11 Backup and restore procedures

Part VII Appendixes

Appendix A Tweaks, Tips and Administration Tasks
Appendix B Obtaining Requests foe Comments (RFCs)

Contents

Introduction

When I begin, the first question I ask to my self was how to install a server with Linux and be sure that no one from the outside and the inside can accesses to it without authorization. Then I was wondering if any methods similar to the one on windows exist to improve the computer performance. Next I began a search on the Internet and read several books to get the most information on security and performance for my server. After many years of research and studies, I finally found the answer to my questions. Those answers was found all though different, documents, books, articles, and Internet sites. Then I create a documentation based on my research that can help me through my daily activities. Through the years this documentation got bigger and started to look more like a book and less then just simple scattered notes. I decide to make it public on the Internet so that anyone can take advantage of it. By sharing those information I did my part for the Linux community who has answered to many of my needs in computer with one magic reliable, strong, powerful, fast and free operating system named Linux. I receive a lot of feedback and comments about my documentation, help to improve it and techniques. Also I find that a lot of peoples wants to see it published for it contents, to get advantage out it and see the power of this beautiful Linux system in action.

Overview

This document is tailored as a step-by-step, example driven document instead of a detailed explanation document on each Linux feature. It doesn't go into much debugging aspects since the Linux Documentation Project's (LDP) HOWTOs already cover this.

This document is intended for a technical audience! It's discuss how to install a RedHat Linux Server with all the necessary security and optimization for a high performance Linux specific machine. Since we speak of optimization and configuration options, we will use a source distribution (tar.gz) program the most possible especially for critical server software like Apache, Bind, Samba, Squid, Openssl etc.

Source program will give us a fast upgrade when necessary and a customization, optimization for our specific machines that often we can't have with RPM. I have used many freely available sources to write this documentation, it seems only fair to give the work back to the Linux community. It is focused on the Intel x86 hardware, so if you are looking for PPC, ARM, SPARC, APX, etc., features; you probably won't find what you are looking for.

Minimal installation for this Server require that you recompile the kernel, other programs are specific according to your needs.

These installation instructions assume

You have a CD-ROM drive and the Official Red Hat Linux CD-ROM.
Installations were tested on the Official RedHat Linux 6.1.

You should understand the hardware system on which the operating system will be installed. After examining the hardware, the rest of this document guides you, step-by-step, though the installation process.

PGP Public Key for Gerhard Mourani

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG v1.0.0 (GNU/Linux)
Comment: For info see http://www.gnupg.org

mQGiBDgU8UcRBADiuIKn95nz0qsvjU1GzBxv0AOxJHVTNhFBl6lt+3DzDA0G7UTu
hOhT0aGwVGts3bzjXVbhS44CTfAvvuVYQq7Ic/BHkwlhFvSu/Xv/fGbD3IQy+Gn5

UYzhZegCGwB0KQhGkIwQPus2ONOS5oT3ChZ8L7JlCPBnlOcVBT+hZ3BXUwCg4y4L
Mz5aEe0MPCZ3xkcNE7AE71EEAL4Jf2uVhIRgOfwIpdB1rKVKrDDFxZLx+yZeOZmq
gdwa4m7wV+Rk+c4I1+qBxxkmcUBhTHigx+9kpBDE2J0aEGQezDN+RoqlmdyVFO98
T/znf4ZLIf0upu5aP4kAltJJuFB1AaJyDLesB5xGjfyWz+RhbKOmeqr2zHniOsa8
HcZ/BACKZFBjNElqFUf0niWf822W6IbNf7ASh8pwTgR9PmXcq2qtBBq8uCIpEYcD
wzk+ccl2jt8qt5RB7DXz/r/uG+3YHU+ID4iz6Qm6zl84gYQLDXST2YXZ5BPURo7H
O4nEIJfeHEuUCstE5ROKnblG2U+t5QmxSGbETnK9I/OZrzFwILRDR2VyaGFyZCBN
b3VyYW5pIChPcGVuIE5ldHdvcmsgQXJjaGl0ZWN0dXJlKSA8Z21vdXJhbmlAdmlk
ZW90cm9uLmNhPohVBBMRAgAVBQI4FPFHAwsKAwMVAwIDFgIBAheAAAoJEDPaC2+7
tLqbGcYAnjHIPAsZrRC5qU5OrqdPvvEmICUWAKCdeyWwJ785A58U8Vh1bpxzCVVb
PbkCDQQ4FPI0EAgAy7qa88bVYWIEyAWxJPZRxl8G2GcxgshSu4+5udeP+4PlVAm8
3DUynzlcax4/ikx8Q8MoVR7s6lCLJXCycLENE8xFCJJQ26IxzBjdftGdmvKteVkZ
Kld9PZMzjUsxKzmhZbGEWug6xaav68EIewTw/S0TFtPhXyUKFrYPV6aID7YGatzB
P4hQJfh4Wt3NdP9QznASBze6bPZxR07iEZaUO0AMHeeBKwL6rptEcGuxHPMYc00R
s+SdGTOAa9E/REIiiEike9mXTKKWJYG2e7leDP3SBruM/c7n+DC9ptFAapg1GD9f
Re7LLFqj6EQzZqybPB61B9rB/8ShIrApcNYF4wADBQgAvROi9N0/J5kYvBVb60no
xBUBYtZp4cJO9X1uVdVahCb9XZpbvxhKujaUoWpPCIb0pm8K+J8x0o9HFl9f/JTs
25N/eJwksr63+j8OdCHqxv4z+qQYgc/qvU42ekHlSfMc7vsiAIE1e1liuTBdN9KR
7oSBoaht+dKi16ffxXmMDvQs1YSBR114XXDSzI+xXRuaIISpi75NE6suLLlrksnL
+i/NcLRbCTEv4p1UJGYT4OVnX6quC3CC+U4Drpjf2ohawsXqS7jKUYduZRr9Hbar
/sE0pQ/P0uf+VAspQJgpvBqiDxbIRCDSx8VgDoRL7iayxPDXtFmbPOrUEPdS7qYX
pIhGBBgRAgAGBQI4FPI0AAoJEDPaC2+7tLqbdzQAniStW48nFU6CWkvQTy8fr0lu
ZXmXAKC5bgSLgg1gZAvx61Z20yzM+hwNFQ==
=95nO
-----END PGP PUBLIC KEY BLOCK-----

Part I Installation-Related Reference

In this Part

Introduction to Linux
Installation of your Linux Server

Chapter 1 Introduction to Linux

In this Chapter

What is Linux?
Some good reasons to use Linux
Let's dispel some of the fear, uncertainty, and doubt about Linux

Introduction to Linux

What is Linux?

Linux is an operating system that was initially created as a hobby by a young student, Linus Torvalds, at the University of Helsinki in Finland. Linus had an interest in Minix, a small UNIX system, and decided to develop a system that exceeded the Minix standards. He began his work in 1991 when he released version 0.02 and worked steadily until 1994 when version 1.0 of the Linux Kernel was released. The current full-featured version is 2.2 (released January 25, 1999), and development continues.

Linux is developed under the GNU General Public License and its source code is freely available to everyone. This however, doesn't mean that Linux and it's assorted distributions are free companies and developers may charge money for it as long as the source code remains available. Linux may be used for a wide variety of purposes including networking, software development, and as an end-user platform. Linux is often considered an excellent, low-cost alternative to other more expensive operating systems.

Due to the very nature of Linux's functionality and availability, it has become quite popular worldwide and a vast number of software programmers have taken Linux's source code and adapted it to meet their individual needs. At this time, there are dozens of ongoing projects for porting Linux to various hardware configurations and purposes.

Some good reasons to use Linux

There are no royalty or licensing fees. Although Linus Torvalds holds the Linux trademark, the Linux kernel and much of the accompanying software are distributed under the GNU General Public License. This means you may modify the source code and sell resulting programs for profit, but original authors retain copyright and you must provide the source to your changes.

Although most popular on Intel-based computers, Linux runs on more CPUs and different platforms than any other computer operating system. One of the reasons for this, beside the programming talents of its rabid followers, is that Linux comes with source code to the kernel and is quite portable.

The recent trend of the software and hardware industry is to push consumers to purchase faster computers with ever-increasing amounts of system memory and hard drive storage. Linux doesn't suffer the prevalent bloat of "creeping featurism," and works quite well, even on aging x486-based computers with limited amounts of RAM.

On the rare occasion a program crashes, Linux won't collapse like a house of cards. You can kill the program and continue working with confidence. Linux uses sophisticated, state-of-the-art memory management to control all system processes. You won't lose control and won't have to suffer the indignities of rebooting the system.

If you need a support platform for server operations, Linux has real advantages, especially when compared to the cost of other operating systems, such as Windows 2000. An additional benefit is that Linux is, for practical purposes, immune to the horde of computer viruses that plague other operating systems. Because of the GNU GPL and Open Source, nearly your entire system comes with source code.

Let's dispel some of the fear, uncertainty, and doubt about Linux

It's a toy operating system.
There's a big software company on the West coast of the United States that would love for this to be true, but it's not. Linux is being put to work more and more everyday by Fortune 500 companies, governments, and consumers as a cost-effective computing solution. Just ask IBM, Compaq, Dell, Apple Computer, Burlington Coat Factory, Amtrak, Virginia Power, NASA, and millions of users around the world.

There's no support.
Although touted by unbelievers as an "unsupported" operating system, every Linux distribution comes with more than 12,000 pages of documentation. Commercial Linux distributions such as Red Hat Linux, Caldera, SuSE, and OpenLinux offer initial support for registered users, and small business and corporate accounts can get 24/7 support through a number of commercial support companies. As an Open Source operating system, Linux also comes with full source code. If you have a problem and have the savvy, fix it yourself! There's no six-month wait for a service release, and many serious bugs (such as security flaws) are fixed within hours by the online Linux community.

Chapter 2 Installation of your Linux Server

In this Chapter

Know your Hardware!
Creating the Boot Disk and Booting
Installation Class and Method
Disk Setup
Components to install
Individual Packages Selection
How to use RPM Commands
Starting and Stopping daemon services
Software that must be uninstalled after installation of the server
Software that must be installed after installation of the server
Installed programs on your server
Put some colors on your terminal
Update of the lasted software's

Installation of your Linux Server

Know your Hardware!

Understanding the hardware is essential for a successful installation of RedHat Linux. Therefore, you should take a moment now and familiarize yourself with your hardware. Be prepared to answer the following questions:

1. How many hard drives do you have?
2. What size is each hard drive (3.2GB)?
3. If you have more than one hard drive, which is the primary one?
4. How much RAM do you have?
5. Do you have a SCSI adapter? If so, who made it and what model is it?
6. What type of mouse do you have?
7. How many buttons?
8. If you have a serial mouse, what COM port is it connected to?
9. What is the make and model of your video card? How much video RAM do you have?
10. What kind of monitor do you have (make and model)?
11. Will you be connecting to a network? If so, what will be the following:
    a. Your IP address?
    b. Your netmask?
    c. Your gateway address?
    d. Your domain name server's IP address?
    e. Your domain name?
    f. Your hostname?
    g. Your types of network(s) card(s) (make and model)?

Creating the Boot Disk and Booting

From time to time, you can find that the installation may fail, if this happen, a revised diskette image is required in order for the installation to work properly. In these cases, special images are available via the Red Hat Linux Errata web page to solve the problem.

Since this is a relatively rare occurrence, you will save time if you try to use the standard diskette images first, and then review the Errata only if you experience any problems completing the installation. Before you make the boot disk, insert the Official Red Hat Linux 6.1 CD-ROM Part 1 in your computer. When the program asks for the filename, you enter **boot.img** for the boot disk. To make the floppies under MS-DOS, you need to use these commands (assuming your CD-ROM is drive D: and contain the Official Red Hat Linux 6.1 CD-ROM).

- Open the Command Prompt under Windows: Start | Programs | Command Prompt
  C:\> **d:**
  D:\> **cd \dosutils**
  D:\dosutils> **rawrite**
  Enter disk image source file name: **..\images\boot.img**
  Enter target diskette drive: **a:**
  Please insert a formatted diskette into drive A: and press --ENTER-- :

  D:\dosutils>

The rawrite.exe program asks for the filename of the disk image. Enter **boot.img**. Insert a floppy into drive A. It will then ask for a disk to write to. Enter **a:**. Label the disk Red Hat boot disk.

Red Hat Linux Errata web page: http://www.redhat.com/errata

Since we start the installation directly off the CD-ROM, you have to boot with the boot disk. Insert the boot disk you create into the drive A: on the computer where you wan to install Linux and reboot the computer. At the boot: prompt, press "**Enter**" to continue booting.

- Choose your language
- Choose your keyboard type
- Select your mouse type

Installation Class and Method (Install Type)

RedHat Linux 6.1 includes defines four different classes, or type of installation. They are:

- GNOME Workstation
- KDE Workstation
- Server
- Custom

These classes (GNOME Workstation, KDE Workstation, and Server) give you the option of simplifying the installation process with a lot loss of configuration flexibility that we don't want to have.

For this reason we highly recommend "**Custom**", as this allows you to choose what services are added and how the system is partitioned.

The idea is to load the minimum packages, while maintaining maximum efficiency. The less software that resides on the box, the fewer potential security exploits or holes.

Select "**Custom**" and click Next.

Disk Setup (Disk Druid)

Warning
We highly recommend, therefore, that you make a backup of your current system before proceeding with the disk partitioning.

For performance, stability and security reason you must do something like the following partition listed bellow on your system. We suppose for this partition configuration the fact that you want to setup a Web server with a Proxy Server on your machine.

We will make two special partitions **"/chroot"** and **"/cache"**, **"/chroot"** partition is for DNS server chrooted, Apache server chrooted and other chrooted future programs. The **"/cache"** partition is for our Squid Proxy server. If you are not intended to install Squid Proxy server you don't need to create the **"/cache"** partition but remember that Squid + Apache will improve a lot your machine performance and security.

Putting **"/tmp"** and **"/home"** on separate partitions is pretty much mandatory if users have shell access to the server, splitting these off onto separate partitions also prevents users from filling up

any critical filesystem, putting **"/var"**, and **"/usr"** on separate partitions is also a very good idea. By isolating the **"/var"** partition, you protect your root partition from overfilling.

In our partition configuration we'll reserve 400 MB of disk space for chrooted programs like Apache, DNS and other softwares. This is necessary because Apache DocumentRoot files and other binaries, programs related to Apache will be installed in this partition.

Take a note that the size of the Apache chrooted directory on the chrooted partition is proportional to the size of your "DocumentRoot" files. If you're not intended to install and use Apache on your server, you can reduce the size of this partition to something like 10 MB for DNS server that you always need.

Minimum size of partitions
This is the minimum size in megabyte a partitions of Linux installation may have to function properly. The sizes of partitions listed bellow are really small. This configuration can fit in very old hard disk of 512MB in size that we can found on old 486 computers. I show you this partition just to get an idea only.

/	35MB
/boot	5MB
/chroot	10MB
/home	100MB
/tmp	30MB
/usr	232MB
/var	25MB

Disk Druid
Disk Druid Partitions is a program that partition your hard drive for you. Choose "**Add**" to add new partition, "**Edit**" to edit partition, "**Delete**" to delete partition and "**Reset**" to reset partition to the original state. When adding a new partition, a new window appear on your screen and give you parameters to choose. Different parameters are:

Mount Point: for where you want to mount you new partition.
Size (Megs): for the size of your new partition in megabyte.
Partition Type: Linux native for Linux fs and Swap for Linux Swap Partition.

If you have a SCSI disk the device will be "**/dev/sda"** and if you have an IDE disk it will be "**/dev/hda"**. If you looking for high performance and stability, a SCSI disk is highly recommended.

Linux refers to disk partitions using a combination of letters and numbers. It's uses a naming scheme that is more flexible and conveys more information than the approach used by other operating systems. Here is a summary:

First Two Letters – The first two letters of the partition name indicate the type of device on which the partition resides. You'll normally see either "**hd"** (for IDE disks), or "**sd"** (for SCSI disks).

The Next Letter – This letter indicates which device the partition is on. For example, "**/dev/hda"** (the first IDE hard disk) and "**/dev/hdb"** (the second IDE disk).

Keep this information in mind, it will make things easier to understand when you're setting up the partitions Linux requires.

A swap partition

Swap partition are used to support virtual memory. If your computer has 16 MB of RAM or less, you must create a swap partition. Even if you have more memory, a swap partition is still recommended. The minimum size of your swap partition should be equal to your computer's RAM or 16 MB (whichever is larger). The largest useable swap partition is roughly 1 GB, (since 2.2 kernel, 1 GB swap file are supported) so making a swap partition larger than that will result in wasted space. Note, however, that you can create and use more than one swap partition (although this is usually only necessary for very large server installations).

Try to put your swap partitions near the beginning of your drive. The beginning of the drive is physically located on the outer portion of the cylinder, so the read/write head can cover much more ground per revolution.

Now as an example:
To make the partitions listed bellow on your system (this is the partition we'll need for our server installation); the command will be under Disk Druid:

Add
Mount Point: **/boot** ← our /boot directory.
Size (Megs): **5**
Partition Type: **Linux Native**
Ok

Add
Mount Point: **/usr** ← our /usr directory.
Size (Megs): **1000**
Partition Type: **Linux Native**
Ok

Add
Mount Point: **/home** ← our /home directory.
Size (Megs): **500**
Partition Type: **Linux Native**
Ok

Add
Mount Point: **/chroot** ← our /chroot directory.
Size (Megs): **400**
Partition Type: **Linux Native**
Ok

Add
Mount Point: **/cache** ← our /cache directory.
Size (Megs): **400**
Partition Type: **Linux Native**
Ok

Add
Mount Point: **/var** ← our /var directory.
Size (Megs): **200**
Partition Type: **Linux Native**
Ok

Add
Mount Point: ← our /Swap partition (leave the Mount Point Blank).
Size (Megs): **150**
Partition Type: **Linux Swap**

Ok

Add
Mount Point: **/tmp** ← our /tmp directory.
Size (Megs): **100**
Partition Type: **Linux Native**
Ok

Add
Mount Point: **/** ← our / directory.
Size (Megs): **316**
Partition Type: **Linux Native**
Ok

After the partition of your hard disk have been completed, you must see something like the following information on your screen. Our mount point will look like that:

Mount Point	Device	Requested	Actual	Type
/boot	Sda1	5M	5M	Linux Native
/usr	Sda5	1000M	1000M	Linux Native
/home	Sda6	500M	500M	Linux Native
/chroot	Sda7	400M	400M	Linux Native
/cache	Sda8	400M	400M	Linux Native
/var	Sda9	200M	200M	Linux Native
<Swap>	Sda10	150M	150M	Linux Swap
/tmp	Sda11	100M	100M	Linux Native
/	Sda12	316M	315M	Linux Native

Drive	Geom [C/H/S]	Total (M)	Free (M)	Used (M)	Used (%)
sda	[3079/64/32]	3079M	1M	3078M	99%

Now that you are partitioning and choosing the mount point of your directories, select “Next” to continue. After your partitions are created, the installation program will ask you to choose partitions to format. Choose the partitions you want to initialize, check the **(Check for bad blocks during format)** box, and press “Next”. This formats the partitions and makes them active so Linux can use them.

On the next screen you will see the LILO Configuration where you have the choice to install LILO boot record on:

- Master Boot Record (MBR)
  Or
- First Sector of Boot Partition

Usually if Linux is the only OS on your machine you must choose “Master Boot Record (MBR)”. After you need to configure your LAN and clock. After you finish configuring the clock, you need to give your system a root password and authentication configuration.

For Authentication Configuration don’t forget to select:

- Enable MD5 passwords
- Enable Shadow passwords

Enable NIS doesn’t need to be selected since we are not configuring a NIS service on this server.

Components to Install (Package Group Selection)

After your partitions have been configured and selected for formatting, you are ready to select packages for installation. By default, Linux is a powerful operating system that executes many useful services. However, most of these services are unneeded and pose a potential security risk.

A proper installation of Linux is the first step to a stable, secure system. You first have to choose which system components you want to install. Choose the components, and then you can go through and select or deselect each individual package of each component by selecting (**Select individual packages**) option.

Since we are configuring a Linux Server, we don't need to install a graphical interface (XFree86) on our system (graphical interface on a server mean; less process; less cpu; less memory; security risks and so on). Graphical interface is usually used on workstation only. Select the following packages for installation.

- *Networked Workstation*
- *Network Management Workstation*
- *Utilities*

After selecting the components you wish to install, you may select or deselect packages.

Note: Select the (**Select individual packages)** options (very important) before continuing to have the possibility to select and deselect packages.

Individual Package Selection

The installation program presents a list of the package groups available, select a group to examine.

Component listed bellow must be unselected from the Menu Group for security; optimization and other reason described bellow.

Applications/Archiving:	dump
Applications/File:	git
Applications/Internet:	finger, ftp, fwhois, ncftp, rsh, rsync, talk, telnet
Applications/Publishing:	ghostscript, ghostscript-fonts, mpage, rhs-printfilters
Applications/System:	arpwatch, bind-utils, knfsd-clients, procinfo, rdate, rdist, screen, ucd-snmp-utils
Documentation:	indexhtml
System Environment/Base:	chkfontpath, yp-tools
System Environment/Daemons:	XFree86-xfs, lpr, pidentd, portmap, routed, rusers, rwho, tftp, ucd-snmp, ypbind
System Environment/Libraries:	XFree86-libs, libpng
User Interface/X:	XFree86-75dpi-fonts, urw-fonts

Before we explain each description of programs we want to uninstall, someone can ask why I need to uninstall finger, ftp, fwhois and telnet on the server? First of all we know that those programs by their nature are insecure. Now imagine that cracker have acceded your new server, he can use finger, ftp, fwhois and telnet programs to query or access other node on your network. If those programs are not installed on your new server, he will be compelled to use those

programs from the outside or try to install program on your server in which case you can trace it with toll like Tripwire.

Applications/Archiving:

The dump package contains both dump and restore. Dump examines files in a filesystem, determines which ones need to be backed up, and copies those files to a specified disk, tape or other storage medium. **[Unnecessary, we use other methods]**

Applications/File:

GIT (GNU Interactive Tools) provides an extensible file system browser, an ASCII/hexadecimal file viewer, a process viewer/killer and other related utilities and shell scripts. **[Unnecessary]**

Applications/Internet:

Finger is a utility, which allows users to see information about system users (login name, home directory, name, how long they've been logged in to the system, etc.). **[Security risks]**

The ftp package provides the standard UNIX command-line FTP client. FTP is the file transfer protocol, which is a widely used Internet protocol for transferring files and for archiving files. **[Security risks]**

The fwhois program allows you or your system's users for querying whois databases. **[Security risks]**

Ncftp is an improved FTP client. Ncftp's improvements include support for command line editing, command histories, recursive gets, automatic anonymous logins and more. **[Security risks, unnecessary]**

The rsh package contains a set of programs, which allow users to run commands on remote machines, login to other machines and copy files between machines (rsh, rlogin and rcp). **[Security risks]**

The ntalk package provides client and daemon programs for the Internet talk protocol, which allows you to chat with other users on different systems. **[Security risks]**

Telnet is a popular protocol for logging into remote systems over the Internet. **[Security risks]**

Applications/Publishing:

Ghostscript is a set of software that provides a PostScript(TM) interpreter, a set of C procedures (the Ghostscript library, which implements the graphics capabilities in the PostScript language) and an interpreter for Portable Document Format (PDF) files. **[Unnecessary]**

These fonts can be used by the GhostScript interpreter during text rendering. They are in addition to the shared fonts between GhostScript and X11. **[Unnecessary]**

The mpage utility takes plain text files or PostScript(TM) documents as input, reduces the size of the text, and prints the files on a PostScript printer with several pages on each sheet of paper. **[Unnecessary and not printer installed on the server]**

The rhs-printfilters package contains a set of print filters, which are primarily meant to be used with the Red Hat printtool. **[Unnecessary, no printer installed on the server]**

Applications/System:

The arpwatch package contains arpwatch and arpsnmp. Arpwatch and arpsnmp are both network monitoring tools. Both utilities monitor Ethernet or FDDI network traffic and build databases of Ethernet/IP address pairs, and can report certain changes via email. **[Unnecessary]**

Bind-utils contains a collection of utilities for querying DNS (Domain Name Service) name servers to find out information about Internet hosts. **[We will compile it later on this documentation]**.

The knfsd-clients package contains the showmount program. Showmount queries the mount daemon on a remote host for information about the NFS (Network File System) server on the remote host. **[Security risks]**

The procinfo command gets system data from the /proc directory (the kernel filesystem), formats it and displays it on standard output. You can use procinfo to acquire information about your system from the kernel as it is running. **[Unnecessary, other methods exist]**

The rdate utility retrieves the date and time from another machine on your network, using the protocol described in RFC 868. **[Security risks]**

The rdist program maintains identical copies of files on multiple hosts. If possible, rdist will preserve the owner, group, mode and mtime of files and it can update programs that are executing. **[Security risks]**

The ucd-snmp package contains various utilities for use with the UCD-SNMP network management project. **[Unnecessary, Security risks]**

The screen utility allows you to have multiple logins on just one terminal. Screen is useful for users who telnet into a machine or are connected via a dumb terminal, but want to use more than just one login. **[Unnecessary]**

The ucd-snmp package contains various utilities for use with the UCD-SNMP network management project. **[Unnecessary]**

Documentation:

The indexhtml package contains the HTML page and graphics for a welcome page shown by your Web browser, which you'll see after you've successfully installed Red Hat Linux. **[Unnecessary]**

System Environment/Base:

Chkfontpath is a simple terminal mode program for adding, removing and listing the directories contained in the X font server's path. **[Unnecessary]**

The Network Information Service (NIS) is a system, which provides network information (login names, passwords, home directories, group information) to all of the machines on a network. **[Security risks]**

System Environment/Daemons:

This is a font server for XFree86. You can serve fonts to other X servers remotely with this package, and the remote system will be able to use all fonts installed on the font server, even if they are not installed on the remote computer. **[Unnecessary]**

The lpr package provides the basic system utility for managing printing services. **[Unnecessary and not printer installed on the server]**

The pidentd package contains identd, which implements the RFC1413 identification server. Identd looks up specific TCP/IP connections and returns either the user name or other information about the process that owns the connection. **[Unnecessary, very few things on the net REQUIRE the sender to be running identd, because many machines don't have it and because many people turn it off.]**

The portmapper program is a security tool which prevents theft of NIS (YP), NFS and other sensitive information via the portmapper. A portmapper manages RPC connections, which are used by protocols like NFS and NIS. **[Unnecessary, Security risks]**

The routed routing daemon handles incoming RIP traffic and broadcasts outgoing RIP traffic about network traffic routes, in order to maintain current routing tables. These routing tables are essential for a networked computer, so that it knows where packets need to be sent. **[Unnecessary, Security risks]**

The rusers program allows users to find out who is logged into various machines on the local network. The rusers command produces output similar to who, but for the specified list of hosts or for all machines on the local network. **[Security risks]**

The rwho command displays output similar to the output of the who command (it shows who is logged in) for all machines on the local network running the rwho daemon. **[Security risks]**

The Trivial File Transfer Protocol (TFTP) is normally used only for booting diskless workstations. The tftp package provides the user interface for TFTP, which allows users to transfer files to and from a remote machine. **[Security risks, Unnecessary]**

SNMP (Simple Network Management Protocol) is a protocol used for network management (hence the name). **[Unnecessary, Security risks]**

System Environment/Libraries:

XFree86-libs contain the shared libraries that most X programs need to run properly. These shared libraries are in a separate package in order to reduce the disk space needed to run X applications on a machine without an X server (i.e. over a network). **[Unnecessary]**

The libpng package contains a library of functions for creating and manipulating PNG (Portable Network Graphics) image format files. PNG is a bit-mapped graphics format similar to the GIF format. **[Unnecessary]**

User Interface/X:

XFree86-75dpi-fonts contain the 75 dpi fonts used on most X Window Systems. **[Unnecessary]**

Free versions of the 35 standard PostScript fonts. With newer releases of ghostscript quality versions of the standard 35 Type 1 PostScript fonts are shipped. **[Unnecessary]**

At this point, the installation program will format every partition you selected for formatting. This can take several minutes depending of the speed of your machine. Once all partitions have been formatted, the installation program starts to install packages.

How to use RPM Commands

This section contains an overview of principal modes using with RPM for installing, uninstalling, upgrading, querying, listing, checking and building RPM packages on your Linux system.

- To install a RPM package, use the command:
  [root@deep]# **rpm -ivh foo-1.0-2.i386.rpm**

RPM packages have file names like **foo-1.0-2.i386.rpm**, which includes the package name (**foo**), version (**1.0**), release (**2**), and architecture (**i386**).

- To uninstall a RPM package, use the command:
  [root@deep]# **rpm -e foo**

Notice that we used the package name "**foo**", not the name of the original package file "**foo-1.0-2.i386.rpm**".

- To upgrade a RPM package, use the command:
  [root@deep]# **rpm -Uvh foo-1.0-2.i386.rpm**

RPM automatically uninstall the old version of foo package and install the new one. Always use "rpm -Uvh" to install packages, since it works fine even when there are no previous versions of the package installed.

- To query a RPM package, use the command:
  [root@deep]# **rpm -q foo**

This command will print the package name, version, and release number of installed package foo. Use this command to verify if package are or are not installed on your system.

- To display package information, use the command:
  [root@deep]# **rpm -qi foo**

This command display package information, including name, version, and description of the installed program.

- To list files in package, use the command:
  [root@deep]# **rpm -ql foo**

This command will list all files in installed RPM package.

- To check a RPM signature package, use the command:

[root@deep]# **rpm --checksig foo**

This command checks the PGP signature of package foo to ensure its integrity and origin. PGP configuration information is read from configuration files. Always use this command first before installing new RPM package on your system. Also, GnuPG or Pgp software must be already installed on your system before you can use this command.

- To install a package from source, use the command:
  [root@deep]# **rpm -ivh --rebuild foo.src.rpm**

The above command would configure and compile the "foo" package, producing a binary RPM file in the "/usr/src/redhat/RPMS/i386/" directory. You can then install the package as you normally would.

Starting and stopping daemon services

Init is the program that gets run by the kernel at boot time. It is in charge of starting all the normal processes that need to run at boot time. These include the APACHE daemons, NETWORK daemons, and anything else you want to run when your machine boots.

How does init start and stop services? Each of the scripts is written to accept an argument, which can be “start” and “stop” and are located under “/etc/rc.d/init.d/” directory. You can execute those scripts by hand in fact with a command like:

For example:

- To start the httpd Web Server manually under Linux.
  [root@deep]# **/etc/rc.d/init.d/httpd start**

- To stop the httpd Web Server manually under Linux.
  [root@deep]# **/etc/rc.d/init.d/httpd stop**

Check inside your “/etc/rc.d/init.d/” directory for services available and use command start | stop to work around.

Software that must be uninstalled after installation of the Server

Red Hat Linux install other pre-established program in your system by default and don’t give you the choice to uninstall them during the install setup. For this reason, you must uninstall the following software on your system after the installation of your server:

pump	apmd	isapnptools	redhat-logos
mt-st	kernel-pcmcia-cs	setserial	redhat-release
eject	linuxconf	kudzu	gd
bc	getty_ps	raidtools	pciutils
mailcap	setconsole	gnupg	

Use command RPM like the following to uninstall them.

- The command to uninstall software is:
  [root@deep]# **rpm -e softwarenames**

Where “softwarenames” is the name of the software you want to uninstall e.g. (foo).

Programs like apmd, kudzu, and sendmail are daemons that run as process. It is better to stop those process before uninstalling them from the system.

- To stop those process, use the following commands:
  [root@deep]# **/etc/rc.d/init.d/apmd stop**
  [root@deep]# **/etc/rc.d/init.d/sendmail stop**
  [root@deep]# **/etc/rc.d/init.d/kudzu stop**

Now you can uninstall them safety and all other package, all together like show bellow:

Step 1
Remove the specified packages.
[root@deep]# **rpm -e pump mt-st eject bc mailcap apmd kernel-pcmcia-cs linuxconf getty_ps setconsole isapnptools setserial kudzu raidtools gnupg redhat-logos redhat-release gd pciutils**

Step 2
Remove the linux.conf-installed file manually.
[root@deep]# rm -f /etc/conf.linuxconf-installed

NOTE: This is a configuration file related to linuxconf software that must be removed manually.

Program **hdparm** is needed by IDE disk but not SCSI disk. If you are an IDE disk on your system you must keep this program (hdparm) and if you don’t have an IDE disk you can remove it from your system.

Program **kbdconfig**, **mouseconfig**, and **timeconfig** in order set your keyboard language and type, you mouse type and you time zone. After those configurations have been set, is rare that you need to change them again. So, you can uninstall them and if in the future you must to change your keyboard, your mouse or your time again all you have to do is to install the program with RPM for your CD-ROM.

Program **sendmail**, **procmail** and **mailx** are always needed on your servers for potential messages send to root user by different software services intalled on your machine to syslog messages.

Sendmail is a powerful Mail Transport Agent (MTA) program that send mail from one machine to another. It actually moves your email over networks or the Internet to where you want it to go. Sendmail can be configured from different manners, it can serve as a internal delivery mail to a Mail Hub Server, a stand alone sendmail or can be configured to be a Central Mail Hub Server for all sendmail machines on your network. So depending of what you want to do with sendmail, you must configure it for the specific task. For this reason you must uninstall sendmail and see the section on this book that related to sendmail configuration and installation to fit you needs.

Sendmail does not handle mail delivery itself. Instead, it runs other programs that perform delivery. Procmail is this delivery agent used by Red Hat Linux for all local mail delivery. Also, procmail need to by installed only on the Central Mail Hub Server. So it is not necessary to have procmail program on all you internal computer that run sendmail, since those internal client will forward mail to the Central Mail Hub via “/bin/mail” or sendmail internal program.

The mailx package installs the “/bin/mail” program, which is used to send quick email messages (i.e., without opening up a full-featured mail user agent). Mailx is often used in shell scripts (for example cron job). The Mailx package is also used to read mail on your local terminal. Mailx must be present on your system.

Pump DHCP (Dynamic Host Configuration Protocol) and BOOTP (Boot Protocol) are protocols which allow individual devices on an IP network to get their own network configuration information (IP address, subnetmask, broadcast address, etc.) from network servers. **[Unnecessary]**

The mt-st package contains the mt and st tape drive management programs. Mt (for magnetic tape drives) and st (for SCSI tape devices) can control rewinding, ejecting, skipping files and blocks and more. **[Unnecessary]**

The eject program allows the user to eject removable media (typically CD-ROMs, floppy disks or Iomega Jaz or Zip disks) using software control. **[Unnecessary]**

The bc package includes bc and dc. Bc is an arbitrary precision numeric processing arithmetic language. Dc is an interactive arbitrary precision stack based calculator, which can be used as a text mode calculator. **[Unnecessary]**

The mailcap file is used by the metamail program. Metamail reads the mailcap file to determine how it should display non-text or multimedia material. **[Unnecessary]**

This is an Advanced Power Management daemon and utilities. It can watch your notebook's battery and warn all users when the battery is low. **[Unnecessary]**

Many laptop machines (and some non-laptops) support PCMCIA cards for expansion. Also known as "credit card adapters," PCMCIA cards are small cards for everything from SCSI support to modems. **[Unnecessary]**

Linuxconf is an extremely capable system configuration tool. Linuxconf provides four different interfaces for you to choose from command line, character-cell (like the installation program), an X Window System based GUI and a web-based interface. **[Unnecessary, buggy program]**

The getty_ps package contains the getty and uugetty programs, basic programs for accomplishing the login process on a Red Hat Linux system. Getty and uugetty are used to accept logins on the console or a terminal. **[Unnecessary]**

setconsole is a basic system utility for setting up the /etc/inittab, /dev/systty and /dev/console files to handle a new console. The console can be either the local terminal (i.e., directly attached to the system via a video card) or a serial console. **[Unnecessary]**

The isapnptools package contains utilities for configuring ISA Plug-and-Play (PnP) cards/boards, which are in compliance with the PnP ISA Specification Version 1.0a. **[Unnecessary]**

setserial is a basic system utility for displaying or setting serial port information. Setserial can reveal and allow you to alter the I/O port and IRQ that a particular serial device is using, and more. **[Unnecessary]**

Kudzu is a hardware probing tool run at system boot time to determine what hardware has been added or removed from the system. **[Unnecessary]**

The raidtools package includes the tools you need to set up and maintain a software RAID device (using two or more disk drives in combination for fault tolerance and improved performance) on a Linux system. **[depending if you use Raid or not]**

GnuPG is GNU's tool for secure communication and data storage. It can be used to encrypt data and to create digital signatures. It includes an advanced key management facility and is compliant with the proposed OpenPGP Internet standard as described in RFC2440. **[we will compile it later on our book]**

The redhat-logos package (the "Package") contains files of the Red Hat "Shadow Man" logo and the RPM logo (the "Logos"). **[Unnecessary]**

Red Hat Linux release file. **[Unnecessary]**

Gd is a graphics library for drawing .gif files. Gd allows your code to quickly draw images (lines, arcs, text, multiple colors, cutting and pasting from other images, flood fills) and write out the result as a .gif file. **[Unnecessary]**

This package (pciutils) contains various utilities for inspecting and setting devices connected to the PCI bus. **[we use other methods]**

The kbdconfig utility is a terminal mode program which provides a simple interface for setting the keyboard map for your system. **[Unnecessary]**

Mouseconfig is a text-based mouse configuration tool. Mouseconfig sets up the files and links needed for configuring and using a mouse on a Red Hat Linux system. **[Unnecessary]**

The timeconfig package contains two utilities: timeconfig and setclock. Timeconfig provides a simple text mode tool for configuring the time parameters in /etc/sysconfig/clock and /etc/localtime. **[Unnecessary]**

The procmail program is used by Red Hat Linux for all local mail delivery. In addition to just delivering mail, procmail can be used for automatic filtering, presorting and other mail handling jobs. Procmail is also the basis for the SmartList mailing list processor. **[Only on the Mail Hub Server]**

Software that must be installed after installation of the Server

To be able to compile programs on your server you must install the following RPM's software. This part of the installation is very important and requires that you install all related packages described bellow. Those software are on your Red Hat 6.1 Part 1 CD-ROM under RedHat/RPMS directory and represent the base necessary software needed on Linux to compile programs.

Step 1
First, we mount the CD-ROM drive and move to the RPMS subdirectory of the CD-ROM.

- To mount you CD-ROM drive and move to RPM directory, use the command:
  [root@deep]# **mount /dev/cdrom /mnt/cdrom/**
  [root@deep]# **cd /mnt/cdrom/RedHat/RPMS/**

This is the package that we need to be able to compile program on the Linux system. Remember, this is the minimum package that permits you to compile most of the tarballs program available for Linux. Other compiler packages exist on the Red Hat CD-ROM, so verify with the README file that come with the tarballs program you want to install if you receive an error messages during compilation of the specific software.

autoconf-2.13-5.noarch.rpm
m4-1.4-12.i386.rpm
automake-1.4-5.noarch.rpm
dev86-0.14.9-1.i386.rpm
bison-1.28-1.i386.rpm
byacc-1.9-11.i386.rpm

cdecl-2.5-9.i386.rpm
cpp-1.1.2-24.i386.rpm
cproto-4.6-2.i386.rpm
ctags-3.2-1.i386.rpm
egcs-1.1.2-24.i386.rpm
ElectricFence-2.1-1.i386.rpm
flex-2.5.4a-7.i386.rpm
gdb-4.18-4.i386.rpm
kernel-headers-2.2.12-20.i386.rpm
glibc-devel-2.1.2-11.i386.rpm
make-3.77-6.i386.rpm
patch-2.5-9.i386.rpm

NOTE: It is better to install software describe above all together if you don't want to receive error dependencies message during RPM install.

Step 2
After, we install all the above needed software with one command.

- The RPM command to install all software together is:
  [root@deep]# **rpm -Uvh autoconf-2.13-5.noarch.rpm m4-1.4-12.i386.rpm automake-1.4-5.noarch.rpm dev86-0.14.9-1.i386.rpm bison-1.28-1.i386.rpm byacc-1.9-11.i386.rpm cdecl-2.5-9.i386.rpm cpp-1.1.2-24.i386.rpm cproto-4.6-2.i386.rpm ctags-3.2-1.i386.rpm egcs-1.1.2-24.i386.rpm ElectricFence-2.1-1.i386.rpm flex-2.5.4a-7.i386.rpm gdb-4.18-4.i386.rpm kernel-headers-2.2.12-20.i386.rpm glibc-devel-2.1.2-11.i386.rpm make-3.77-6.i386.rpm patch-2.5-9.i386.rpm**

Step 3
You must exit and re-login for all the change to take effect.

- To exit and re-login, use the command:
  [root@deep]# **exit**

After installation and compilation of all programs that you need on your server, it's a good idea to remove all sharp objects (compilers, etc) describe above unless needed from a system. One of the reasons is if a cracker gain access to your server it couldn't compile or modify binaries programs. Also, this will free a lot space and will help to improve regular scanning of files on your server for integrity checking.

A lot strategies exist for networking systems and the one I want to explain you is from my opinion one of the best for a lot reasons:

First, when you run a server you will give it a special task to accomplish. You will never put all services you want to offer in one machine or you will lost speed (resources available divided by the number of process running on the server), and decrease your security (a lot services running on the same machine, and if cracker access this server, it can attack directly all the other available).

Second, having different server doing different task will simplify the administration, management (you know what task each server are supposed to do, what services to be available, which ports are open to clients access and which one are closes, you know what you are supposed to see in the log files, etc), and give you more control and flexibility on each one (server dedicated for mail, web pages, database, development, backup, etc).

So having for example one server specialized just for the development and test will permit to not be compelled to install compilers program on server each time you want to compile and install

new software on this machine and be obliged after to uninstall compiler, sharp objects. For more information on the subject, please see the chapter 8 “Compiler Functionality” on this book.

Installed programs on your Server

Since we are chosen to custom the installation of our Linux system, this is the list of all installed programs that you must have in your computer after the complete installation of the Linux Server. This list must match exactly the **install.log** file located in your "/tmp" directory or your could run under problem. Don't forget to install all programs listed above "Software that must be installed after installation of the Server" to be able to make compilation on your Server.

Installing setup.
Installing filesystem.
Installing basesystem.
Installing ldconfig.
Installing glibc.
Installing shadow-utils.
Installing mktemp.
Installing termcap.
Installing libtermcap.
Installing bash.
Installing MAKEDEV.
Installing SysVinit.
Installing XFree86-SVGA.
Installing chkconfig.
Installing apmd.
Installing arpwatch.
Installing ncurses.
Installing info.
Installing fileutils.
Installing grep.
Installing ash.
Installing at.
Installing authconfig.
Installing bc.
Installing bdflush.
Installing binutils.
Installing bzip2.
Installing sed.
Installing console-tools.
Installing e2fsprogs.
Installing rmt.
Installing cpio.
Installing cracklib.
Installing cracklib-dicts.
Installing crontabs.
Installing textutils.
Installing dev.
Installing diffutils.
Installing ed.
Installing eject.
Installing etcskel.
Installing file.
Installing findutils.
Installing gawk.
Installing gd.
Installing gdbm.
Installing getty_ps.
Installing glib.
Installing gmp.
Installing gnupg.
Installing gpm.
Installing groff.
Installing gzip.
Installing hdparm.
Installing initscripts.
Installing ipchains.
Installing isapnptools.
Installing kbdconfig.
Installing kernel.
Installing kernel-pcmcia-cs.
Installing kudzu.
Installing ld.so.
Installing less.
Installing libc.
Installing libstdc++.
Installing lilo.
Installing pwdb.
Installing pam.
Installing sh-utils.
Installing redhat-release.
Installing linuxconf.
Installing logrotate.
Installing losetup.
Installing lsof.
Installing mailcap.
Installing mailx.
Installing man.
Installing mingetty.
Installing mkbootdisk.
Installing mkinitrd.
Installing modutils.
Installing mount.
Installing mouseconfig.
Installing mt-st.
Installing ncompress.
Installing net-tools.
Installing netkit-base.
Installing newt.
Installing ntsysv.
Installing passwd.
Installing pciutils.
Installing perl.
Installing procmail.
Installing procps.
Installing psmisc.
Installing pump.
Installing python.
Installing quota.
Installing raidtools.
Installing readline.
Installing redhat-logos.
Installing rootfiles.
Installing rpm.
Installing sash.
Installing sendmail.
Installing setconsole.
Installing setserial.
Installing setuptool.
Installing shapecfg.
Installing slang.
Installing slocate.
Installing stat.
Installing sysklogd.
Installing tar.
Installing tcp_wrappers.
Installing tcpdump.
Installing tcsh.
Installing time.
Installing timeconfig.
Installing timed.
Installing tmpwatch.
Installing traceroute.
Installing utempter.
Installing util-linux.
Installing vim-common.
Installing vim-minimal.
Installing vixie-cron.
Installing which.
Installing zlib.

Put some colors on your terminal

Putting some colors on your terminal can help you to distinguish folders, files, archives, devices, symbolic links and executable file from others. My opinion is that colors help to make less errors and fast navigation on your system.

Edit the **profile** file (vi /etc/profile) and add the following lines:

```
# Enable Colour ls
eval `dircolors /etc/DIR_COLORS -b`
export LS_OPTIONS='-s -F -T 0 --color=yes'
```

Edit the **bashrc** file (vi /etc/bashrc) and add the line:

```
alias ls='ls --color=auto'
```

Then log in and out; after this, the new COLORS-environment variable is set, and your system will recognize that.

Update of the lasted software's

Keep and update all software (especially network software) to the lasted versions, check the errata pages for the Red Hat Linux distribution, available at http://www.redhat.com/corp/support/errata/index.html. The errata pages are perhaps the best resource for fixing 90% of the common problems with Red Hat Linux. In addition, security holes for which a solution exists are generally on the errata page 24 hours after Red Hat has been notified. You should always check there first.

Software's that must be updated at this time for your Red Hat Linux 6.1 server are:

groff-1_15-1_i386.rpm
sysklogd-1_3_31-14_i386.rpm
initscripts-4_70-1_i386.rpm
e2fsprogs-1.17-1.i386.rpm
pam-0_68-10_i386.rpm
Linux kernel 2.2.14 (linux-2_2_14_tar.gz)

NOTE: The Linux kernel is the most important, and always must be updated. See bellow for more information on building a custom kernel for your specific system.

- You can verify if the software is installed on your system before make an update with the following command:
  [root@deep]# **rpm -q <softwarename>**

Where <softwarename> is the name of the software you want to verify like XFree86, telnet, etc.

Part II Security and optimization-Related Reference

In this Part

Chapter 3 General System Security
In this Chapter

Linux General Security

Linux General Security

Overview

A UNIX system is only as secure as the administrator makes it. The more services you add, the more chances of introducing a security hole. Operating systems like SCO and others may actually be more prone to security breaches because they offer more services that are an integral part of how they operate, in order to be more “user friendly”.

Linux itself is very stable and secure, but it in itself is distributed in many flavors. When installing Linux, one should tend to install with the minimum, and then add only the ESSENTIAL items, reducing chances of an “application” of having a security weakness. Linux is the most SECURE if properly implemented.

If a weakness is apparent in the system, there are thousands of volunteers to point it out immediately, along with a fix. In a larger organization, such as some of the commercial products, they have a limited size of team members working on it. It is not always in their best interests to publicize any discoveries too loudly, and sometimes it takes a while before fixes trickle down the pipes into the releases or upgrades. Yes, they soon become available as patches, but most administrators of commercial products tend to use the tools available with the distribution only, with a false sense of comfort in that they have more professionally designed software.

Mistakes can happen in programming at any level, but when you have 10's of thousand of people with the source code available to them, these mistakes are often discovered faster in an open source code environment. Of course, with 10's of thousands of people meddling with the source code, and what? 12 million copies of Linux out there now... there is a much better chance that someone will open a security hole too.

This document will discuss some of the general techniques used to secure your site. The following is for people attaching a computer to the Internet. Generally speaking services such as NFS, Samba, Imap and pop will only need to be accessible to internal users, blocking external access greatly simplifies matters. The following is a list of features that can be used to help prevent attacks from external and internal sources.

1. Basic Information

Let outsiders know as little as possible about your system. A simple finger to a victims system can reveal much about a system; How many users, when the admin is in, when do they work, who THEY are, who uses the system and personal information that might help an attacker guess a users password. You can use a powerful finger daemon and/or tcpd to limit who can connect to your system and how much they will know about your system. Better is to uninstall finger package on your system.

Logs are the only way to know what's going on your Linux system. Of course, this assumes that an attacker has not corrupted your logs, but that aside. Full logging of all connections, can reveille so much about an attacker, and their attempts. If you have problems understanding the logs and what they are saying then get help to understand them. There is nothing wrong with someone helping you with something you don't understand. I get information every-now-and-then on something that I have made a mistake on and that I am told to clarify or fix something. To be arrogant is one thing to be stupid is another. See bellow on this section “28. Physical hard copies of all important logs” for more information on how to have more control on log files.

Limit the number of SUID root programs on your system. SUID root programs are programs when run, run at the level of root (God in the UNIX world). Sometimes this is needed but many times

not. Since SUID root programs can do anything that root can they bear a high level of responsibility in following the golden rules of security programming. Sometimes they do, sometimes they don't and when they don't, users can sometimes get them to do things their not suppose to do. This is where exploits and come in. An exploit is a program or script that will get a SUID root program to do very bad stuff (Give root shells, grab password files, read other people's mail, delete files, ect). See below on this section “31. Bits from root-owned programs” for more information on the subject.

2. BIOS Security, set a boot password

Disallow booting from floppy drives and allow passwords to access some BIOS features. Check your BIOS manual or look at it the next time you boot up. This will improve the security of your system. Disallowing booting from floppy drives will block undesired people trying to boot your Linux system with a special boot disk and allowing passwords to access the BIOS features will protect you from people trying to change BIOS feature like allowing boot from floppy drive or booting the server without prompt password.

3. Security Policy

It is important to point out that you can not implement security if you have not decided what needs to be protected and from whom. You need a security policy, a list of what you consider allowable and what you do not consider allowable upon which to base any decisions regarding security. The policy should also determine your response to security violations. What you should consider when compiling a security policy will depend entirely on your definition of security. The following questions should provide some general guidelines:

- How do you classify confidential or sensitive information?
- Exactly who do you want to guard against?
- Do remote users need access to your system?
- Does the system contain confidential or sensitive information?
- What will the consequences be if this information is leaked to your competitors or other outsiders?
- Will passwords or encryption provide enough protection?
- Do you need access to the Internet?
- How much access do you want to allow to your system from the Internet?
- What action will you take if you discover a breach in your security?

This list is short, and your policy will probably encompass a lot more before it is complete. Perhaps the very first thing you need to assess is the depth of your paranoia. Any security policy must be based on some degree of paranoia; deciding how much you trust people, both inside and outside your organization. The policy must, however, provide a balance between allowing your users reasonable access to the information they require to do their jobs and totally disallowing access to your information. The point where this line is drawn will determine your policy.

4. Password

The starting point of our Linux General Security tour is the password. Many people keep their entire life on a computer and the only thing preventing others from seeing it is the eight-character string called a password. Not something one would call completely reliable. Contrary to popular belief, an uncrackable password does not exist. Given time and resources all passwords can be guessed either by social engineering or by brute force.

Social engineering of server passwords and other access methods is still the easiest and most popular way to gain access to accounts and servers. Most help desk workers have access to many user accounts due poor user security and the general tendency in any environment to place explicit trust in other employees at the same organization – especially those who are there to help solve problems. Several very successful server and client attacks have been documented on many different networks due to ambitious individuals who gained access through lax security methods. Often, something as simple as acting as a superior or executive in a company and yelling at the right person at the right time of the day yields terrific results and, in some cases, total access to internal server resources.

Since password cracking can be a time- and resource consuming art, make it hard for any cracker who has grabbed your password file. Running a password cracker on a weekly basis on your system is a good idea. This helps to find and replace passwords that are easily guessed or weak. Also, a password checking mechanism should be present to reject a weak password when first choosing a password or changing an old one. Character strings that are plain dictionary words, or are all in the same case, or do not contain numbers or special characters should not be accepted as a new password.

I recommend the following rules to make passwords effective:

- They should be at least six characters in length, preferably including at least one numeral or special character.
- They must not be trivial; a trivial password is one that is easy to guess and is usually based on the user's name, family, occupation or some other personal characteristic.
- They should have an aging period, requiring a new password to be chosen within a specific time frame.
- They should be revoked and be required to be reset after a limited number of concurrent incorrect retries.

5. The password length

The minimum acceptable password length by default when you install your Linux system is 5. This mean that when a new user is allowed to have a access on the server, his/her password length will be at minimun 5 mixes of character strings, leter, number, special character ect. This is not enough and must be at less 8. To prevent unconscious people or administrator to be able to enter just 5 character length for the valuable password edit the rather important "/etc/login.defs" file and change the value of 5 length to 8 length.

Edit the login.defs file (vi /etc/login.defs) and change the line that read:

PASS_MIN_LEN 5
To read:
PASS_MIN_LEN 8

The "login.defs" is the configuration file for the login program. You should review or make changes to this file for your particular system. This is where you set other security policy settings (like password expiration defaults or minimum acceptable password length).

6. The root account

The "root" account is the most privileged account on a Unix system. The "root" account has no security restrictions imposed upon it. This means the system assumes you know what you are doing, and will do exactly what you request -- no questions asked. Therefore it is easy, with a mistyped command, to wipe out crucial system files. When using this account it is crucial to be as careful as possible. For security reasons, never log in your server as "root" unless is absolutely necessary for tasks that necessities "root" access. Also if your are not on your server, never sign in and let in as "root". VERY VERY VERY BAD.

7. Encryption

Encryption uses a key, a unique value, as input to an algorithm to make data readable only to someone else who knows the key. This works extremely well while all the hosts involved are under your control in a secure environment, but if a “trusted” host gets compromised, then you are once again at risk. It is not only user IDs and passwords that are at risk here. Most encryption implementations are used to pass confidential or classified information between systems. If one of the systems is compromised then the information may become known or exposed. This will be minimized by an effective security policy, but remains a risk whenever either host containing the encryption key is exposed to the Internet. Use of encryption technologies like OpenSSL, SSH, MD5 can be very helpful, see later in this book for more information on the topic.

8. The “/etc/exports” file

If you are exporting filesystems using NFS, be sure to configure “/etc/exports” file with the most restrictive access possible. This means not using wildcards, not allowing root write access, and mounting read-only wherever possible.

Edit the **exports** file (vi /etc/exports) and add:

For example:
/dir/to/export host1.mydomain.com(ro,root_squash)
/dir/to/export host2.mydomain.com(ro,root_squash)

Where “**/dir/to/export”** is the directory you want to export, **host.mydomain.com** is the machine allowed to log in this directory, <**ro>** mean mounting read-only and <**root_squash>** for not allowing root write access in this directory.

For this change to take effect you will need to run **/usr/sbin/exportfs -a**

NOTE: Please be aware that having an NFS service available on your system can be a security risk. Personally, I don't recommend using it.

9. Disabling console program access

One of the simplest and commonest customizations is to entirely disable all console-equivalent access to programs like shutdown and halt. To do this, run:

[root@deep]# **rm -f /etc/security/console.apps/servicename**

Where **servicename** is the name of the program to which you wish to disable console-equivalent access. Unless you use xdm, however, be careful not to remove the xserver file or no one but root will be able to start the X server. (If you always use xdm to start the X server, root is the only user that needs to start X, in which case you might actually want to remove the xserver file).

As an example:
[root@deep]# **rm -f /etc/security/console.apps/halt**
[root@deep]# **rm -f /etc/security/console.apps/poweroff**
[root@deep]# **rm -f /etc/security/console.apps/reboot**
[root@deep]# **rm -f /etc/security/console.apps/shutdown**
[root@deep]# **rm -f /etc/security/console.apps/xserver** (if removed, root will be the only user able to start X).

Will disable console-equivalent access to programs halt, poweroff, reboot, and shutdown. Once again, the program xserver apply only is you are installed the Xwindow interface on your system.

NOTE: If you are follow our setup installation, the Xwindow interface is not installed in your server and all the files described above will not appear in the "/etc/security" directory, so don't make attention to the above step.

10. Disabling all console access

In order to disable all console access, including program and file access, in the "/etc/pam.d/" directory, comment out all lines that refer to **pam_console.so**. This step is the continuity of the above hack "7. Disabling console program access".

The following script will do the trick automatically for you. As "root" create the **disabling.sh** script file (touch disabling.sh) and add the following lines inside:

```
# !/bin/sh
cd /etc/pam.d
for i in * ; do
sed '/[^#].*pam_console.so/s/^/#/' < $i > foo && mv foo $i
done
```

Make this script executable with the following command and execute it:
[root@deep]# **chmod 700 disabling.sh**
[root@deep]# **./disabling.sh**

This will comment out all lines that refer to "pam_console.so" for all files located under "/etc/pam.d" directory. Once the script has been executed, you can remove it from your system.

11. The "/etc/inetd.conf" file

Inetd, called the "super server", will load a network program based upon a request from the network. The "inetd.conf" file tell inetd which ports to listen to and what server to start for each port. As soon as you put your Linux system on ANY network the first thing to look at is what services you need to offer.

Services that you do not need to offer should be disabled and mush better uninstalled so that you have one less thing to worry about and attackers have one less place to look for a hole. Look at your "/etc/inetd.conf" file and see what services are being offered by your inetd. Disable any that you do not need by commenting them out (# at the beginning of the line), and then sending your inetd process a SIGHUP.

Step 1
Change the permissions on this file to **600**.
[root@deep]# **chmod 600 /etc/inetd.conf**

Step 2
ENSURE that the owner is **root**.

[root@deep]# **stat /etc/inetd.conf**

```
  File: "/etc/inetd.conf"
  Size: 2869        Filetype: Regular File
  Mode: (0600/-rw-------)        Uid: (    0/    root)  Gid: (    0/    root)
Device:  8,6   Inode: 18219     Links: 1
Access: Wed Sep  22  16:24:16   1999(00000.00:10:44)
Modify:  Mon Sep  20  10:22:44   1999(00002.06:12:16)
Change:Mon Sep  20  10:22:44   1999(00002.06:12:16)
```

Step 3
Edit the **inetd.conf** file (vi /etc/inetd.conf) and disable services like:

ftp, telnet, shell, login, exec, talk, ntalk, imap, pop-2, pop-3, finger, auth, etc. unless you plan to use it. If it's turned off it's much less of a risk.

```
# To re-read this file after changes, just do a 'killall -HUP inetd'
#
#echo          stream    tcp     nowait    root        internal
#echo          dgram     udp     wait      root        internal
#discard       stream    tcp     nowait    root        internal
#discard       dgram     udp     wait      root        internal
#daytime       stream    tcp     nowait    root        internal
#daytime       dgram     udp     wait      root        internal
#chargen       stream    tcp     nowait    root        internal
#chargen       dgram     udp     wait      root        internal
#time          stream    tcp     nowait    root        internal
#time          dgram     udp     wait      root        internal
#
# These are standard services.
#
#ftp           stream    tcp     nowait    root        /usr/sbin/tcpd  in.ftpd -l -a
#telnet        stream    tcp     nowait    root        /usr/sbin/tcpd  in.telnetd
#
# Shell, login, exec, comsat and talk are BSD protocols.
#
#shell         stream    tcp     nowait    root        /usr/sbin/tcpd  in.rshd
#login         stream    tcp     nowait    root        /usr/sbin/tcpd  in.rlogind
#exec          stream    tcp     nowait    root        /usr/sbin/tcpd  in.rexecd
#comsat        dgram     udp     wait      root        /usr/sbin/tcpd  in.comsat
#talk          dgram     udp     wait      root        /usr/sbin/tcpd  in.talkd
#ntalk         dgram     udp     wait      root        /usr/sbin/tcpd  in.ntalkd
#dtalk         stream    tcp     wait      nobody      /usr/sbin/tcpd  in.dtalkd
#
# Pop and imap mail services et al
#
#pop-2         stream    tcp     nowait    root        /usr/sbin/tcpd  ipop2d
#pop-3         stream    tcp     nowait    root        /usr/sbin/tcpd  ipop3d
#imap          stream    tcp     nowait    root        /usr/sbin/tcpd  imapd
#
# The Internet UUCP service.
#
#uucp          stream    tcp     nowait    uucp        /usr/sbin/tcpd  /usr/lib/uucp/uucico    -l
#
# Tftp service is provided primarily for booting.  Most sites
# run this only on machines acting as "boot servers." Do not uncomment
# this unless you *need* it.
#
#tftp          dgram     udp     wait      root        /usr/sbin/tcpd  in.tftpd
#bootps        dgram     udp     wait      root        /usr/sbin/tcpd  bootpd
#
```

```
# Finger, systat and netstat give out user information which may be
# valuable to potential "system crackers."  Many sites choose to disable
# some or all of these services to improve security.
#
#finger         stream   tcp     nowait   root     /usr/sbin/tcpd  in.fingerd
#cfinger        stream   tcp     nowait   root     /usr/sbin/tcpd  in.cfingerd
#systat         stream   tcp     nowait   guest    /usr/sbin/tcpd  /bin/ps -auwwx
#netstat        stream   tcp     nowait   guest    /usr/sbin/tcpd  /bin/netstat    -f inet
#
# Authentication
#
#auth           stream   tcp     nowait   nobody   /usr/sbin/in.identd   in.identd -l -e -o
#
# End of inetd.conf
```

NOTE: don't forget to send your inetd process a SIGHUP signal (killall -HUP inetd) after making change to your inetd.conf file.

[root@deep /root]# **killall -HUP inetd**

Step 4
One more security measure you can take to secure the "**inetd.conf"** file is to set it immutable, using the **chattr** command. To set the file immutable simply:

[root@deep]# **chattr +i /etc/inetd.conf**

And this will prevent any changes (accidental or otherwise) to the "inetd.conf" file. A file with the 'i' attribute cannot be modified: it cannot be deleted or renamed, no link can be created to this file and no data can be written to the file. Only the superuser can set or clear this attribute. If you wish to modify the inetd.conf file you will need to unset the immutable flag:

[root@deep]# **chattr -i /etc/inetd.conf**

12. TCP_WRAPPERS

By default Red Hat Linux allows all service requests. Using TCP_WRAPPERS makes securing your servers against outside intrusion is a lot simpler and painless then you would expect. Deny all hosts by putting "ALL: ALL@ALL, PARANOID" in "/etc/hosts.deny" and explicitly list trusted hosts who are allowed to your machine in "/etc/hosts.allow' file is the safest configuration. TCP_WRAPPERS is controlled from two files. The search stops at the first match.

/etc/hosts.allow
/etc/hosts.deny

- Access will be granted when a (daemon, client) pair matches an entry in the /etc/hosts.allow file.
- Otherwise, access will be denied when a (daemon, client) pair matches an entry in the /etc/hosts.deny file.
- Otherwise, access will be granted.

Step 1
Edit the **hosts.deny** file (vi /etc/hosts.deny) and add the following line:

```
Access is denied by default.
# Deny access to everyone.
ALL: ALL@ALL, PARANOID #Matches any host whose name does not match its address, see bellow.
```

Which means all services, all locations, so any service not explicitly allowed is then blocked, unless they are permitted access by entries in the allow file.

NOTE: With the parameter **"PARANOID"**. If you are intended to run telnet or ftp services on your server, don't forget to add client machine name and IP address in your "**/etc/hosts"** file on the server or you can expect to wait several minutes for the DNS lookup to time out, before you get a login: prompt.

Step 2
Edit the **hosts.allow** file (vi /etc/hosts.allow) and add for example, the following line:

The explicitly authorized host are listed in the allow file.

As an example:
sshd: 208.164.186.1 gate.openarch.com

For your client machine: 208.164.186.1 is the IP address and gate.openarch.com the host name of one of your client allowed using sshd.

Step 3
The tcpdchk program, is the tcpd wrapper configuration checker. It examines your tcp wrapper configuration and reports all potential and real problems it can find.

- After your configuration is done, run the program **tcpdchk**.
  [root@deep]# **tcpdchk**

13. The "/etc/aliases" file

The aliases file can easily be used to gain privileged status if it wrongly or carelessly administered. For example, many vendors used to ship systems with a "**decode"** alias in the aliases file.

The intention is to provide an easy way for users to transfer binary files using mail. At the sending site the user converts the binary to ASCII with "**uuencode"**, then mails the result to the "**decode"** alias at the receiving site. That alias pipes the mail message through the "/usr/bin/uuencode" program, which converts the ASCII back into the original binary file. You can imagine the security hole that can happen with this feature turning On in your "aliases" file.

Remove the "**decode**" alias line from your "aliases" file. Similarly, every alias that executes a program that you did not place there yourself and check completely should be questioned and probably removed. For this change to take effect you will need to run:

[root@deep]# **/usr/bin/newaliases**

Edit the **aliases** file (vi /etc/aliases) and remove or comment out the following lines:

```
# Basic system aliases -- these MUST be present.
MAILER-DAEMON:  postmaster
postmaster:     root

# General redirections for pseudo accounts.
bin:            root
daemon:         root
#games:         root  ← remove or comment out.
#ingres:        root  ← remove or comment out.
nobody:         root
#system:        root  ← remove or comment out.
#toor:          root  ← remove or comment out.
```

#uucp: **root** ← remove or comment out.

Well-known aliases.
#manager: **root** ← remove or comment out.
#dumper: **root** ← remove or comment out.
#operator: **root** ← remove or comment out.

trap decode to catch security attacks
#decode: **root**

Person who should get root's mail
#root: marc

Don't forget to run "**/usr/bin/newaliases"** for this change to take effect.

14. Prevent your Sendmail being abused by unauthorized users

The very latest versions of Sendmail (8.9.3) include powerful Anti-Spam features which can help prevent your mail server being abused by unauthorized users. To do that, edit your "/etc/sendmail.cf" file and make a change to the configuration file to block off spammers.

Edit the **sendmail.cf** file (vi /etc/sendmail.cf) and change the line:

O PrivacyOptions=authwarnings
To read:
O PrivacyOptions=authwarnings**,noexpn,novrfy**

The change prevents spammers from using the "EXPN" and "VRFY" commands in sendmail. These commands are too often abused by unethical individuals. See the Sendmail configuration and installation section in this book for more information on this topic.

Edit the **sendmail.cf** file (vi /etc/sendmail.cf) and change the line:

O SmtpGreetingMessage=$j Sendmail $v/$Z; $b
To read:
O SmtpGreetingMessage=$j Sendmail $v/$Z; $b **NO UCE C=xx L=xx**

The change modifies the banner which Sendmail displays upon receiving a connection. You should replace the ``xx'' in the ``C=xx L=xx'' entries with your country and location codes. For example, in my case, I would use ``C=CA L=QC'' for Canada, Quebec. The latter change doesn't actually affect anything, but was recommended by folks in the news.admin.net-abuse.email newsgroup as a legal precaution.

15. Prevent your system from responding to ping request

Preventing your system for responding to ping request can be a big improvement in your network security since no one can ping on your server and receive an answer. The TCP/IP protocol suite has a number of weaknesses that allow an attacker to leverage techniques in the form of covert channels to surreptitiously pass data in otherwise benign packets. Preventing your server from responding to ping request can help to minimize this problem.

An...
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all

... should do the job too and your system won't respond to ping on any interface. You can add this line in your “/etc/rc.d/rc.local” file so the command will be automatically set if your system reboot. Not reponding to pings would at least keep most "hackers" out becaues they would never even know it's there.

To turn it back on, simply
echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_all"

16. Don’t let system issue file to be displayed

If you don't want your systems issue file to be displayed when people log in remotely, you can change the telnet option in your “/etc/inetd.conf” file to look like:

telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd -h

Adding the “-h” flag on the end will cause the daemon to not display any system information and just hit the user with a login: prompt. This hack is only necessary if you’re using Telnet daemon on your server.

17. The “/etc/host.conf” file

Linux uses a resolver library to obtain the IP address corresponding to a host name. The “/etc/host.conf” file specifies how names are resolved. The entries in the “etc/host.conf” file tell the resolver library what services to use, and in what order, to resolve names.

Edit the **host.conf** file (vi /etc/host.conf) and add the following lines:

```
# Lookup names via DNS first then fall back to /etc/hosts.
order bind,hosts
# We have machines with multiple IP addresses.
multi on
# Check for IP address spoofing.
nospoof on
```

The **order** option indicate the order of services. The sample entry specifies that the resolver library should first consult the name server to resolve a name and then check the “/etc/hosts” file. It is recommended to set the resolver library to first check the name server (bind) and then the hosts file (hosts) for better performance and security on all your servers. Of course you must have the DNS/BIND software installed or this configuration will not work.

The **multi** option determines whether a host in the “/etc/hosts” file can have multiple IP addresses (multiple interface ethN). Hosts that have more than one IP address are said to be multiomed, because the presence of multiple IP addresses implies that host has several network interfaces. As an example, a Gateway Server will always have multiple IP address and must have this option set to ON.

The **nospoof** option indicate to take care of not permit spoof on this machine. IP-Spoofing is a security exploit that works by tricking computers in a trust relationship that you are someone that you really aren't. In this type of attack, a machine is set up to “look” like a legitimate server and then issue connections and other types of network activities to legitimate end systems, other servers or large data repository systems. This option must be set ON for all type of server.

18. Routing Protocols

Routing and routing protocols can create several problems. IP source routing, where an IP packet contains details of the path to its intended destination, is dangerous because according to RFC 1122 the destination host must respond along the same path. If an attacker were able to send a source routed packet into your network, then he would be able to intercept the replies and fool your host into thinking it is communicating with a trusted host. I strongly recommend that you disable IP source routing to protect your server from this hole.

To disable IP source routing on your server, type the following command in your terminal:

```
for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
   echo 0 > $f
done
```

Add the above commands to "**/etc/rc.d/rc.local"** file and you'll not have to type it again the next time if you reboot your system. Make a note that the above command will disable Source Routed Packets on all your interfaces (lo, ethN, pppN ect). If you intended to install the IPCHAINS Firewall describe in this book, you are not need to make this command, since its already appear in the Firewall script file.

19. Enable TCP SYN Cookie Protection

A "SYN Attack" is a denial of service (DoS) attack that consumes all the resources on your machine, forcing you to reboot. Denial of service attacks (attacks which incapacitate a server due to high traffic volume or ones that tie-up system resources enough that the server cannot respond to a legitimate connection request from a remote system) are easily achievable from internal resources or external connections via extranets and Internet. In the 2.1 kernel series this config option mearly allows syn cookies, but does not enable them. To enable them, you have to do:

[root@deep]# **echo 1 > /proc/sys/net/ipv4/tcp_syncookies**

Add the above commands to "**/etc/rc.d/rc.local"** file and you'll not have to type it again the next time if you reboot your system. If you intended to install the IPCHAINS Firewall describe in this book, you are not need to make this command, since its already appear in the Firewall script file.

20. Firewalls

Another solution to the security issue is to hide your hosts and internal transmissions from the outside world, and only allow traffic to flow to and from your network via a low risk gateway. Such a gateway is called a firewall, and we will devote a significant chapter of this book to firewalls.

21. The "/etc/services" file

The port numbers on which certain "standard" services are offered are defined in the RFC 1700 "Assigned Numbers". The "/etc/services" file enable server and client programs to convert service names to these numbers (ports), the list is kept on each host and it is stored in the file "/etc/services". Only the "root" user is allowed to make modification in this file and it is rare to edit the "/etc/services" file to make change, since its already contain the more common ones service names to port numbers. To improve security we can immunize this file to prevent unauthorized deletion or addition of services.

- To immunize the "/etc/services" file, use the command:
  [root@deep]# **chattr +i /etc/services**

22. The "/etc/securetty" file

The "/etc/securetty" file allows you to specify which **TTY** devices the "root" user is allowed to login on. The "/etc/securetty" file is read by the login program (usually "/bin/login"). Its format is a list of the **tty** devices names allowed, and on all others **tty** that are commented out or doen't appear in this file, root login is disallowed.

Disable any **tty** that you do not need by commenting them out (# at the beginning of the line).

Edit the **securetty** file (vi /etc/securetty) and comment out the following lines:

tty1
#tty2
#tty3
#tty4
#tty5
#tty6
#tty7
#tty8

Which means root is only allowed to login on tty1. This is my recommendation, allowing "root" to log only on one tty device and use the "su" command to switch to "root" if you need more tty device to log on as "root".

23. Special accounts

DISABLE **ALL default vendor accounts** not necessary shipped with the Operating System. (This should be checked after each upgrade or installation). Linux provides these accounts for various system activities, which you may not need. If you do not need the accounts, remove them. The more accounts you have, the easier it is to access your system.

We assumes you are using the Shadow password suite on your Linux system. If you are not, you should consider doing so, as it helps to tighten up security somewhat. This must already be set if you're follow our Linux installation above and selected under "Authentication Configuration" section the option "Enable Shadow Passwords".

- To delete user on your system, use the command:
  [root@deep]# **userdel username**

- To delete group on your system, use the command:
  [root@deep]# **groupdel username**

Step 1
Type the following commands on your terminal to delete users listed bellow:

[root@deep]# **userdel adm**
[root@deep]# **userdel lp**
[root@deep]# **userdel sync**
[root@deep]# **userdel shutdown**
[root@deep]# **userdel halt**
[root@deep]# **userdel news**
[root@deep]# **userdel uucp**
[root@deep]# **userdel operator**
[root@deep]# **userdel games** (delete this user if you don't use X Window Server).
[root@deep]# **userdel gopher**

[root@deep]# **userdel ftp** (delete this user if you don't use ftp anonymous).

Step 2
Type the following commands on your terminal to delete usersgroups listed bellow:

[root@deep]# **groupdel adm**
[root@deep]# **groupdel lp**
[root@deep]# **groupdel news**
[root@deep]# **groupdel uucp**
[root@deep]# **groupdel games** (delete this group if you don't use X Window Server).
[root@deep]# **groupdel dip**
[root@deep]# **groupdel pppusers**
[root@deep]# **groupdel popusers** (delete this group if you don't use pop server for email).
[root@deep]# **groupdel slipusers**

Step 3
Add the necessary user to the system:

- To add user on your system, use the command:
  [root@deep]# **useradd username**

- To add or change password for user on your system, use the command:
  [root@deep]# **passwd username**

  For example:
  [root@deep]# **useradd admin**
  [root@deep]# **passwd admin**

  The output should look something like this.
  Changing password for user admin
  New UNIX password: somepasswd
  passwd: all authentication tokens updated successfully

Step 4
The immutable bit can be used to prevent accidentally deleting or overwriting a file that must be protected. It also prevents someone from creating a symbolic link to this file, which has been the source of attacks involving deleting "/etc/passwd", "/etc/shadow", "/etc/group" or "/etc/gshadow".

- To set the immutable bit on the passwords and groups files, use the command:

  [root@deep]# **chattr +i /etc/passwd**
  [root@deep]# **chattr +i /etc/shadow**
  [root@deep]# **chattr +i /etc/group**
  [root@deep]# **chattr +i /etc/gshadow**

NOTE: In the future, if you are intended to add or delete user, usergroup on your password, group files, you must unset the immutable bit on all those files or you will not be able to make your changes. Also if you are intended to install a RPM program that will add automatically a new user to the different immunized passwd and group files, then you will receive an error message during the install as so long as you are not unset the immutable bit from those files.

24. Blocking anyone to su to root

If you don't want anyone to su to root or restrict "su" command for certain users then add the following two lines to the top of your config "su" file in "/etc/pam.d/" directory. I highly recommend to limit the person allowed to "su" to root account.

Step 1
Edit the **su** file (vi /etc/pam.d/su) and add the following two lines to the top in the file:

auth sufficient /lib/security/pam_rootok.so debug
auth required /lib/security/pam_wheel.so group=wheel

After adding the two lines above, the "/etc/pam.d/su" file should look like this:

```
#%PAM-1.0
auth            sufficient      /lib/security/pam_rootok.so debug
auth             required       /lib/security/pam_wheel.so group=wheel
auth            required        /lib/security/pam_pwdb.so shadow nullok
account         required        /lib/security/pam_pwdb.so
password        required        /lib/security/pam_cracklib.so
password        required        /lib/security/pam_pwdb.so shadow use_authtok nullok
session         required        /lib/security/pam_pwdb.so
session         optional        /lib/security/pam_xauth.so
```

Which mean only those who are a member of the "**wheel**" group can su to root, it also includes logging. Note that the "wheel" group is a special account on your system that can be used for this purpose. You can not use any group name as you will want to make this hack. This hack combined with which **TTY** devices root is allowed to login on, will improve a lot your security on the system.

Step 2
Now that we are defines the "wheel" group in our "/etc/pam.d/su" file configuration, it is time to add some users allowed to "su" to "root" account. If you want to make as an example the user admin member of the "wheel" group and be able to su to root use the following command:

[root@deep]# **usermod -G10 admin**

Which mean "G" is a list of supplementary groups, which the user is also a member of. "10" are the numerical value of the user's ID "wheel", and "admin" is the user we want to add to "wheel" group. Use the same command above for all users on your system you want to be able to su to "root" account.

25. Resource limits

Set resource limits on all your users so they can't perform denial of service attacks (number of processes, amount of memory, etc). These limits will have to be setup for the user when he or she logs in. For example, limits for all users on your system might look like this.

Step 1
Edit the **limits.conf** file (vi /etc/security/limits.conf) and add or change the lines to read:

```
*       hard    core    0
*       hard    rss     5000
*       hard    nproc   20
```

This says to prohibit the creation of core files "core 0", restrict the number of processes to 20 "nproc 20", and restrict memory usage to 5M "rss 5000" for everyone except the super user "root". All of the above only concern users who have entered through the login prompt on your system. With this kind of quota, you have more control on the processes, core files, and memory usage that users may have on your system. The asterisk "*" mean all users that logs in on the server.

Step 2
You must also edit the "/etc/pam.d/login" file and add the following line to the bottom of the file:

session required /lib/security/pam_limits.so

After adding the line above, the "/etc/pam.d/login" file should look like:

```
#%PAM-1.0
auth        required    /lib/security/pam_securetty.so
auth        required    /lib/security/pam_pwdb.so shadow nullok
auth        required    /lib/security/pam_nologin.so
account     required    /lib/security/pam_pwdb.so
password    required    /lib/security/pam_cracklib.so
password    required    /lib/security/pam_pwdb.so nullok use_authtok md5 shadow
session     required    /lib/security/pam_pwdb.so
session     required    /lib/security/pam_limits.so
#session    optional    /lib/security/pam_console.so
```

26. More control on mounting a file system

You can have more control on mounting a file system like "/home" and "/tmp" with some nifty options like noexec, nodev, and nosuid. This can be setup in the "/etc/fstab" file. The file fstab contains descriptive information about the various file systems. For more information on options that you can set in this file, see the man pages about mount (8).

Edit the **fstab** file (vi /etc/fstab) and change depending of your needs:

```
/dev/sda11      /tmp    ext2    defaults    1 2
/dev/sda6       /home   ext2    defaults    1 2
To read:
/dev/sda11      /tmp    ext2    nosuid,nodev,noexec 1 2
/dev/sda6       /home   ext2    nosuid,nodev 1 2
```

Which mean for *<nodev>* do not interpret character or block special devices on the file system, for *<nosuid>* do not allow set-user-identifier or set-group-identifier bits to take effect and for *<noexec>* do not allow execution of any binaries on the mounted file system.

NOTE: For our example above, the "/dev/sda11" represent our "/tmp" directory on the system, and "/dev/sda6" the "/home" directory. Of course this will be not the same for you, depending of how you have partitioned you hard disk and what kind of disk are installed on your system, IDE (hda, hdb, etc) or SCSI (sda, sdb, etc).

27. Move the binary RPM in a safe place and change its default permission

Once your have installed all software you need on your Linux server with the RPM command, it's a good idea for better security to move it in a safe place like floppy disk or other safe place of your choice. With this method if some one access your server and have the intention to install evil software with RPM command, he shouldn't be able. Of course if in the future you want to install new software via RPM all you have to do is to replace the RPM binary to his original directory again.

- To move RPM binary on the floppy disk, use the command:
  [root@deep]# **mount /dev/fd0 /mnt/floppy/**
  [root@deep]# **mv /bin/rpm /mnt/floppy/**
  [root@deep]# **umount /mnt/floppy**

NOTE: Never uninstall RPM program completely from your system or you will be unable to reinstall it again later since to install RPM or other software you need to have RPM commands available.

On more thing you can do is to change the default permission of “rpm” command from 755 to 700. With this modification, non-root users can’t use the “rpm” program to query, install etc; in case you forget to move it on safe place after installation of new programs.

- To change the default permission of “/bin/rpm”, use the command:
  [root@deep]# **chmod 700 /bin/rpm**

28. Shell logging

To make it easy for you to repeat long commands, the bash shell stores up to 500 old commands in the “~/.bash_history” file (where “~/” is your home directory). Each users that have a account on the system will have this file “.bash_history” in their home directory. Reducing the number of old commands the “.bash_history” files can hold may protect users on the server to enter by mistake their password on the screen in plain text and have their password stored for a long time in the “.bash_history” file.

The HISTFILESIZE and HISTSIZE lines in the “/etc/profile” file determine the size of old commands the “.bash_history” file for all users on your system can hold. For all accounts I would highly recommend setting the HISTFILESIZE and HISTSIZE in “/etc/profile” file to a low value such as **20**.

Edit the **profile** file (vi /etc/profile) and change the lines to:

HISTFILESIZE=**20**
HISTSIZE=**20**

Which mean, the “.bash_history” file in each users home directory can store 20 old commands and no more. Now, if a cracker try to see the “~/.bash_history” file of users on your server to find some password typed by mistake in plain text, he has less change to find one.

29. The “/etc/lilo.conf” file

LILO is a versatile boot loader for Linux. It does not depend on a specific file system, it can boot Linux kernel images from floppy disks and from hard disks and can even act as a "boot manager" for other operating systems.

LILO primarily accesses the following parts of the system: The root file system partition is important for two reasons: first, LILO sometimes has to tell the kernel where to look for it. Second, it is frequently a convenient place for many other items LILO uses, such as the boot sector, the “/boot” directory, and the kernels. The boot sector contains the first part of LILO's boot loader. It loads the much larger second-stage loader. Both loaders are typically stored in the file “/boot/boot.b”. The kernel is loaded and started by the boot loader. Kernels typically reside in the root directory or in “/boot” in our case for Red Hat Linux.

We have seen that LILO is very important in the Linux system and for this reason, we must to protect it the best we can. The most important configuration file of LILO is the “lilo.conf” file and reside under “/etc” directory. Is with this file that we can configure and improve the security of our LILO program and Linux system. Following are tree important options that will improve the security of our valuable LILO program.

- Add: **timeout=00**

This controls how long (in tenths of seconds) LILO waits for user input before booting to the default selection. One of the requirements of C2 security is that this interval be set to 0 (obviously

a dual boot machines blows most security out of the water). It is a good idea to set this to 0 unless the system dual boots something else.

- Add: **restricted**

Relaxes the password protection by requiring a password only if parameters are specified on the command line (e.g. linux single). The option “restricted” can only be used together with the “password” option. Make sure you use this one on each image.

- Add: **password=<password>**

Ask the user for a password when trying to load the Linux system in “single mode”. Passwords are always case-sensitive, also make sure the “/etc/lilo.conf” file is no longer world readable, or any user will be able to read the password. Here is an example of our protected LILO with the “lilo.conf” file.

Step 1
Edit the **lilo.conf** file (vi /etc/lilo.conf) and add or change the tree options above as show:

```
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
prompt
timeout=00 ← change this line to 00.
Default=linux
restricted ← add this line.
password=<password> ← add this line and put your password.
image=/boot/vmlinuz-2.2.12-20
label=linux
initrd=/boot/initrd-2.2.12-10.img
root=/dev/sda6
read-only
```

Step 2
Because the configuration file “/etc/lilo.conf” now, contains unencrypted passwords, it should only be readable for the super-user “root”.

[root@deep]# **chmod 600 /etc/lilo.conf** (will be no longer world readable).

Step 3
Now we must update our configuration file “/etc/lilo.conf” for the change to take effect.

[root@deep]# **/sbin/lilo -v** (to update the lilo.conf file).

Step 4
One more security measure you can take to secure the “lilo.conf” file is to set it immutable, using the **chattr** command.

- To set the file immutable simply, use the command:
  [root@deep]# **chattr +i /etc/lilo.conf**

And this will prevent any changes (accidental or otherwise) to the “lilo.conf” file. If you wish to modify the “lilo.conf” file you will need to unset the immutable flag:

- To unset the immutable flag, use the command:
  [root@deep]# **chattr -i /etc/lilo.conf**

30. Disable the Control-Alt-Delete keyboard shutdown command

Commenting out “#” the line listed bellow in your “/etc/inittab” file will disable the possibility to use Control-Alt-Delete command to shutdown your computer. This is pretty important if you don't have the best physical security on the box.

To do this, edit the **inittab** file (vi /etc/inittab) and change the line:

ca::ctrlaltdel:/sbin/shutdown -t3 -r now
To read:
#ca::ctrlaltdel:/sbin/shutdown -t3 -r now

Now, for the system to understand the change type in the following at a prompt:

[root@deep]# **/sbin/init q**

31. Physical hard copies of all important logs

One of the most security consideration is the integrity of the different log files under “/var/log” directory on your server. If although all the securities we are put in place in our server, a cracker can gain access to it, our last defence are the log files. So it is very important to considered a methodes by which we can be sure of the integrity of our log files.

If you have printer installed in your server or on other server in your network, a good idea would be to have actually physical hard copies of all important logs. This can be easily accomplished by using a continuous feed printer and having syslog sending all logs you seem important out to "/dev/lp0" (the printer device). Cracker can change the files, programs, etc on your server, but can do nothing when you have a real paper that print via the printer a copy of all of your important logs.

As an example:
For loggin of all telnet, mail, boot messages and ssh connections from your server to the printer attached to this server, then you would want to add the following line to "/etc/syslog.conf" file:

Edit the syslog.conf file (vi /etc/syslog.conf) and add at the end of this file the following line:
authpriv.*;mail.*;local7.*;auth.*;daemon.info /dev/lp0

- Now restart your syslog daemon for the change to take effect:
  [root@deep]# **/etc/rc.d/init.d/syslog restart**

As an example:
For loggin of all telnet, mail, boot messages and ssh connections from your server to the printer attached to a remote server in your local network, then you would want to add the following line to "/etc/syslog.conf" file on the remote server.

If you don’t have a printer in your network, you can also copy all the log files to another machine, simply obmit the first step bellow of adding “/dev/lp0” to your “syslog.conf” file on remote and go directly to the “-r” option step on remote. Using the feature of copying all the log files to another machine will give you the possibility to control all syslog messages on one host and will tears down administration needs.

Edit the syslog.conf file (vi /etc/syslog.conf) on the remote server (for example: mail.openarch.com) and add at the end of this file the following line:
authpriv.*;mail.*;local7.*;auth.*;daemon.info /dev/lp0

Since the default configuration of the syslog daemon is to not receive any messages from the network, we must enable on the remote server the facility to receive message from the network. To enable the facility to receive message from the network on the remote server, add the following option "-r" to your syslog daemon script file (only on the remote host):

- Edit the **syslog** daemon (vi +24 /etc/rc.d/init.d/syslog) and change:

daemon syslogd -m 0
To read:
daemon syslogd -r -m 0

- Now restart your syslog daemon on the remote host for the change to take effect:
  [root@mail]# **/etc/rc.d/init.d/syslog restart**

Now, if we have a firewall on the remote server (you are supposed to have), we must add or verify the existance of the following lines:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
      -s $SYSLOG_CLIENT \
      -d $IPADDR 514 -j ACCEPT
```

Where EXTERNAL_INTERFACE="eth0" in the firewall file.
Where IPADDR="208.164.186.2" in the firewall file.
Where SYSLOG_CLIENT="208.164.168.0/24" in the firewall file.

- Now restart your firewall on the remote host for the change to take effect:
  [root@mail]# **/etc/rc.d/init.d/firewall restart**

This firewall rule will allow incoming UDP packet on port 514 (syslog port) on the remote server that come from our internal client to be accepted. For more information on Firewall see the chapter 7 "Networking firewall".

Finally, edit the syslog.conf file (vi /etc/syslog.conf) on the local server, and add at the end of this file the following line:
authpriv.*;mail.*;local7.*;auth.*;daemon.info @mail

Where "mail" is the hostname of the remote server. Now if anyone ever hacks your box and menaces to erase vital system logs, then you still have a hard copy of everything. It should then be fairly simple to trace where they came from and deal with it accordingly.

- Now restart your syslog daemon for the change to take effect:
  [root@deep]# **/etc/rc.d/init.d/syslog restart**

Same as on the remote host, we must add or verify the existence of the following lines in our firewall script file on the local host:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
      -s $IPADDR 514 \
      -d $SYSLOG_SERVER 514 -j ACCEPT
```

Where EXTERNAL_INTERFACE="eth0" in the firewall file.
Where IPADDR="208.164.186.1" in the firewall file.
Where SYSLOG_SERVER="mail.openarch.com" in the firewall file.

- Now restart your firewall for the change to take effect:
  [root@deep]# **/etc/rc.d/init.d/firewall restart**

This firewall rule will allow outgoing UDP packet on port 514 (syslog port) on the local server destined to the remote syslog server to be accepted. For more information on Firewall see the chapter 7 “Networking firewall”.

NOTE: Never use your Gateway Server as a host to control all syslog messages, this is a very bad idea. More options and strategies exist with the sysklogd program, see the man pages about sysklogd (8), syslog(2), and syslog.conf(5) for more information.

32. Fix the permissions under “/etc/rc.d/init.d” directory for script files

Fix the permissions of the scripts files that are responsible to start and stop all your normal processes that need to run at boot time.

[root@deep]# **chmod -R 700 /etc/rc.d/init.d/***

Which means just root is allowed to Read, Write, and Execute scripts files on this directory. I don’t think regular users need to know what inside those script files.

NOTE: If you install a new program or update a program that use the init system V script located under “/etc/rc.d/init.d/” directory, don’t forget to change or verify the permission of this script file again.

33. The “/etc/rc.d/rc.local” file

By default, when you login to a Linux box, it tells you the Linux distribution name, version, kernel version, and the name of the server. This is giving away too much info. We rather just prompt users with a "Login:" prompt.

Step 1
To do this, Edit the "/etc/rc.d/rc.local" file and Place "**#**" in front of the following lines like shown:

```
--
# This will overwrite /etc/issue at every boot.  So, make any changes you
# want to make to /etc/issue here or you will lose them when you reboot.
#echo "" > /etc/issue
#echo "$R" >> /etc/issue
#echo "Kernel $(uname -r) on $a $(uname -m)" >> /etc/issue
#
#cp -f /etc/issue /etc/issue.net
#echo >> /etc/issue
--
```

Step 2
Then, remove the following files “issue.net” and “issue” under “/etc” directory:

[root@deep]# **rm -f /etc/issue**
[root@deep]# **rm -f /etc/issue.net**

NOTE: The “/etc/issue.net” file is the login banner that users will see when they make a networked (i.e. telnet, SSH) connection to your machine. You will find it in the “/etc” directory, along with a similar file called "issue", which is the login banner that gets displayed to local users. It is simply a text file and can be customized to your own tastes, but be aware that if you do change it or remove it like we do, you'll also need to modify the “/etc/rc.d/rc.local” shell script, which re-creates both the "issue" and "issue.net" files every time the system boots.

34. Bits from root-owned programs

All programs and files in your computer with the 's' bits appearing on it mode, have the SUID (-rw**s**r-xr-x) or SGID (-r-xr-**s**r-x) bit enable. Because these programs grant special privileges to the user who is executing them, it is important to remove the 's' bits from root-owned programs that won't require such privilege. This can be accomplished by executing the command **'chmod a-s'** with the name(s) of the offending files as it's arguments.

Such programs include, but aren't limited to:

1.programs you never use.
2.programs that you don't want any non-root users to run.
3.programs you use occasionally, and don't mind having to su (1) to root to run.

We've placed an asterisk (*) next to each program we personally might disable. Remember that your system needs some suid root programs to work properly, so be careful.

- To find all files with the 's' bits from root-owned programs, use the command:
  [root@deep]# **find / -type f \(-perm -04000 -o -perm -02000 \) \-exec ls –lg {} \;**

```
*-rwsr-xr-x  1 root   root      35168  Sep  22  23:35  /usr/bin/chage
*-rwsr-xr-x  1 root   root      36756  Sep  22  23:35  /usr/bin/gpasswd
*-r-xr-sr-x  1 root   tty        6788  Sep   6  18:17  /usr/bin/wall
-rwsr-xr-x   1 root   root      33152  Aug  16  16:35  /usr/bin/at
-rwxr-sr-x   1 root   man       34656  Sep  13  20:26  /usr/bin/man
-r-s--x--x   1 root   root      22312  Sep  25  11:52  /usr/bin/passwd
-rws--x--x   2 root   root     518140  Aug  30  23:12  /usr/bin/suidperl
-rws--x--x   2 root   root     518140  Aug  30  23:12  /usr/bin/sperl5.00503
-rwxr-sr-x   1 root   slocate   24744  Sep  20  10:29  /usr/bin/slocate
*-rws--x--x  1 root   root      14024  Sep   9  01:01  /usr/bin/chfn
*-rws--x--x  1 root   root      13768  Sep   9  01:01  /usr/bin/chsh
*-rws--x--x  1 root   root       5576  Sep   9  01:01  /usr/bin/newgrp
*-rwxr-sr-x  1 root   tty        8328  Sep   9  01:01  /usr/bin/write
-rwsr-xr-x   1 root   root      21816  Sep  10  16:03  /usr/bin/crontab
*-rwsr-xr-x  1 root   root       5896  Nov  23  21:59  /usr/sbin/usernetctl
*-rwsr-xr-x  1 root   bin       16488  Jul   2  10:21  /usr/sbin/traceroute
-rwxr-sr-x   1 root   utmp       6096  Sep  13  20:11  /usr/sbin/utempter
-rwsr-xr-x   1 root   root      14124  Aug  17  22:31  /bin/su
*-rwsr-xr-x  1 root   root      53620  Sep  13  20:26  /bin/mount
*-rwsr-xr-x  1 root   root      26700  Sep  13  20:26  /bin/umount
*-rwsr-xr-x  1 root   root      18228  Sep  10  16:04  /bin/ping
*-rwxr-sr-x  1 root   root       3860  Nov  23  21:59  /sbin/netreport
-r-sr-xr-x   1 root   root      26309  Oct  11  20:48  /sbin/pwdb_chkpwd
```

- To disable the suid bits on selected programs above, type the following commands:
  [root@deep]# **chmod a-s /usr/bin/chage**
  [root@deep]# **chmod a-s /usr/bin/gpasswd**
  [root@deep]# **chmod a-s /usr/bin/wall**
  [root@deep]# **chmod a-s /usr/bin/chfn**
  [root@deep]# **chmod a-s /usr/bin/chsh**
  [root@deep]# **chmod a-s /usr/bin/newgrp**
  [root@deep]# **chmod a-s /usr/bin/write**
  [root@deep]# **chmod a-s /usr/sbin/usernetctl**
  [root@deep]# **chmod a-s /usr/sbin/traceroute**
  [root@deep]# **chmod a-s /bin/mount**
  [root@deep]# **chmod a-s /bin/umount**
  [root@deep]# **chmod a-s /bin/ping**
  [root@deep]# **chmod a-s /sbin/netreport**

If you want to know what those programs do, make a man program-name and read.

As an example:
[root@deep]# **man netreport**

35. Unusual or hidden files

Look everywhere on the system for unusual or hidden files (files that start with a period and are normally not shown by 'ls'), as these can be used to hide tools and information (password cracking programs, password files from other systems, etc.). A common technique on UNIX systems is to put a hidden directory in a user's account with an unusual name, something like '...' or '.. ' (dot dot space) or '..^G' (dot dot control-G). The “find” program can be used to look for hidden files.

As an example:
[root@deep]# **find / -name ".. " -print -xdev**
[root@deep]# **find / -name ".*" -print -xdev | cat -v**

Also, files with names such as '.xx' and '.mail' have been used (that is, files that might appear to be normal).

36. Find all files with the SUID/SGID bit enabled

SUID and SGID files on your system are a potential security risk, and should be monitored closely. Because these programs grant special privileges to the user who is executing them, it is necessary to ensure that insecure programs are not installed.

A favorite trick of crackers is to exploit SUID "root" programs, then leave a SUID program as a backdoor to get in the next time, even if the original hole is plugged. Find all SUID and SGID programs on your system, and keep track of what they are, so you are aware of any changes, which could indicate a potential intruder.

- Use the following command to find all SUID/SGID programs on your system:
  [root@deep]# **find / -type f \(-perm -04000 -o -perm -02000 \) \-exec ls -lg {} \;**

NOTE: See in this book under chapter 9 “Securities Software” for more information about the software sXid that will make the job for you automatically each day and report the results via mail.

37. Find group and World Writable files and directories

Particularly system files, can be a security hole if a cracker gain access to your system and modifies them. Additionally, world-writable directories are dangerous, since they allow a cracker to add or delete files as he wishes. In the normal course of operation, several files will be writable, including some from “/dev”, and symbolic links.

- To locate all world-writable files on your system, use the following command:
  [root@deep]# **find / -type f \(-perm -2 -o -perm -20 \) -exec ls -lg {} \;**
  [root@deep]# **find / -type d \(-perm -2 -o -perm -20 \) -exec ls -ldg {} \;**

NOTE: A file and directory integrity checker like Tripwire software can be used regularly to scan, manage and find modified group or world writable files and directories easily. See in this book under chapter 9 “Securities Software” for more information about Tripwire.

38. Unowned files

Unowned files may also be an indication an intruder has accessed your system. Don't permit any unowned file. If you find unowned file or directory on your system, verify it integrity and if all look fine give it an owner name. Some time you may uninstall a program and get unowned file or directory related to this software, in this case you can remove the file or directory safety.

- To locate files on your system that do not have an owner, use the following command:
  [root@deep]# **find / -nouser -o -nogroup**

NOTE: Once again, files reported under "/dev" directory don't count.

39. Finding ".rhosts" files

The ".rhosts" files should be a part of your regular system administration duties, as these files should not be permitted on your system. Remember that a cracker only needs one insecure account to potentially gain access to your entire network.

- You can locate all ".rhosts" files on your system with the following command:
  [root@deep]# **find /home -name .rhosts**

You can also use a cron job to periodically check for, report the contents of and delete $HOME/.rhosts files. Also, users should be made aware that you regularly perform this type of audit, as directed by policy.

- To use a cron to periodically check and report via mail all ".rhosts" files, do the following:

  Create as "root" the **find_rhosts_files** script file under "/etc/cron.daily" directory (touch /etc/cron.daily/find_rhosts_files) and add the following lines in this script file:

```
#!/bin/sh
/usr/bin/find /home -name .rhosts | (cat <<EOF
This is an automated report of possible existent ".rhosts" files on the server
deep.openarch.com, generated by the find utility command.

New detected ".rhosts" files under the "/home" directory include:
EOF
cat
) | /bin/mail -s "Content of .rhosts file audit report" root
```

Now make this script file executable then change, verify the owner, and group to by "root"
[root@deep]# **chmod 755 /etc/cron.daily/find_rhosts_files**
[root@deep]# **chown 0.0 /etc/cron.daily/find_rhosts_files**

Each day a mail will be send to "root" with a subject:" Content of .rhosts file audit report" containing potential new finding ".rhosts" files.

40. System has been compromised

If you believe that your system has been compromised, contact CERT ® Coordination Center or your representative in FIRST (Forum of Incident Response and Security Teams).

Internet Email: cert@cert.org

CERT Hotline: (+1) 412-268-7090
Facsimile: (+1) 412-268-6989
CERT/CC personnel answer 8:00 a.m. – 8:00 p.m. EST (GMT –5)/EDT (GMT –4)) on working days; they are on call for emergencies during other hours and on weekends and holidays.

Chapter 4 General System Optimization

In this Chapter

Linux General Optimization

Linux General Optimization

Overview

Tuning a network is very much dependent on the implementation of the products in use within the network (both software and hardware). It is impossible to pull together in one volume all the information required to tune a complete network. Also, you will not be able to identify the bottlenecks and holdups that might exist until the network is up and running. Performance tuning is not straight-forward and should be looked upon as a complex task. It requires a great deal of discipline and exactness. Unless appropriate tests are used to identify specific bottlenecks within the system it is difficult to interpret any results. Occasionally performance tuning can be very frustrating and tedious at times especially when after a great deal of analysis the results are still inconclusive. Nevertheless, performance tuning can be very rewarding and provide long term benefits.

1. The "/etc/profile" file

The "/etc/profile" file contains system wide environment stuff and startup programs. All customizations that you will put in this file will apply for the entire environment variable in your system. So putting optimization flags in this file is a good choice. To squeeze the most performance from your x86 programs, you can use full optimization when compiling with the -O9 flag. Many programs contain -O2 in the Makefile. -O9 is the highest level of optimization. It will increase the size of what it produces, but it runs faster.

When compiling, use **-fomit-frame-pointer**. This will use the stack for accessing variables. Unfortunately debugging is almost impossible with this option. You can use the -mcpu=cpu_type and -march=cpu_type switch, this will optimize for the CPU listed to the best of GCC's ability. However, the resulting code will only be runnable on the indicated CPU or higher.

For CPU i686 or PentiumPro, Pentium II, Pentium III

In the "**/etc/profile"** file, put this line for a PentiumPro, Pentium II and III Pro Processor family:

CFLAGS='-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions'

For CPU i586 or Pentium

In the "**/etc/profile"** file, put this line for a Pentium Processor family:

CFLAGS='-O3 -march=pentium -mcpu=pentium -ffast-math -funroll-loops -fomit-frame-pointer -fforce-mem -fforce-addr -malign-double -fno-exceptions'

For CPU i486

In the "**/etc/profile"** file, put this line for a i486 Processor family:

CFLAGS='-O3 -funroll-all-loops -malign-double -mcpu=i486 -march=i486 -fomit-frame-pointer -fno-exceptions'

Now after the choice of your setting (i686, i586, or i486) a bit further down in the "/etc/profile" file, add " **CFLAGS LANG LESSCHARSET**" to the "export" line:

export PATH PS1 HOSTNAME HISTSIZE HISTFILESIZE USER LOGNAME MAIL INPUTRC **CFLAGS LANG LESSCHARSET**

Then log in and out; after this, the new CFLAGS environment variable is set, and Software's and other "configure" tool will recognize that. Pentium (Pro/II/III) optimizations will only work with egcs or pgcc compilers. Egcs compiler is already installed on your Server by default so you don't need to think about.

Benchmark Results
Summaries by Architecture

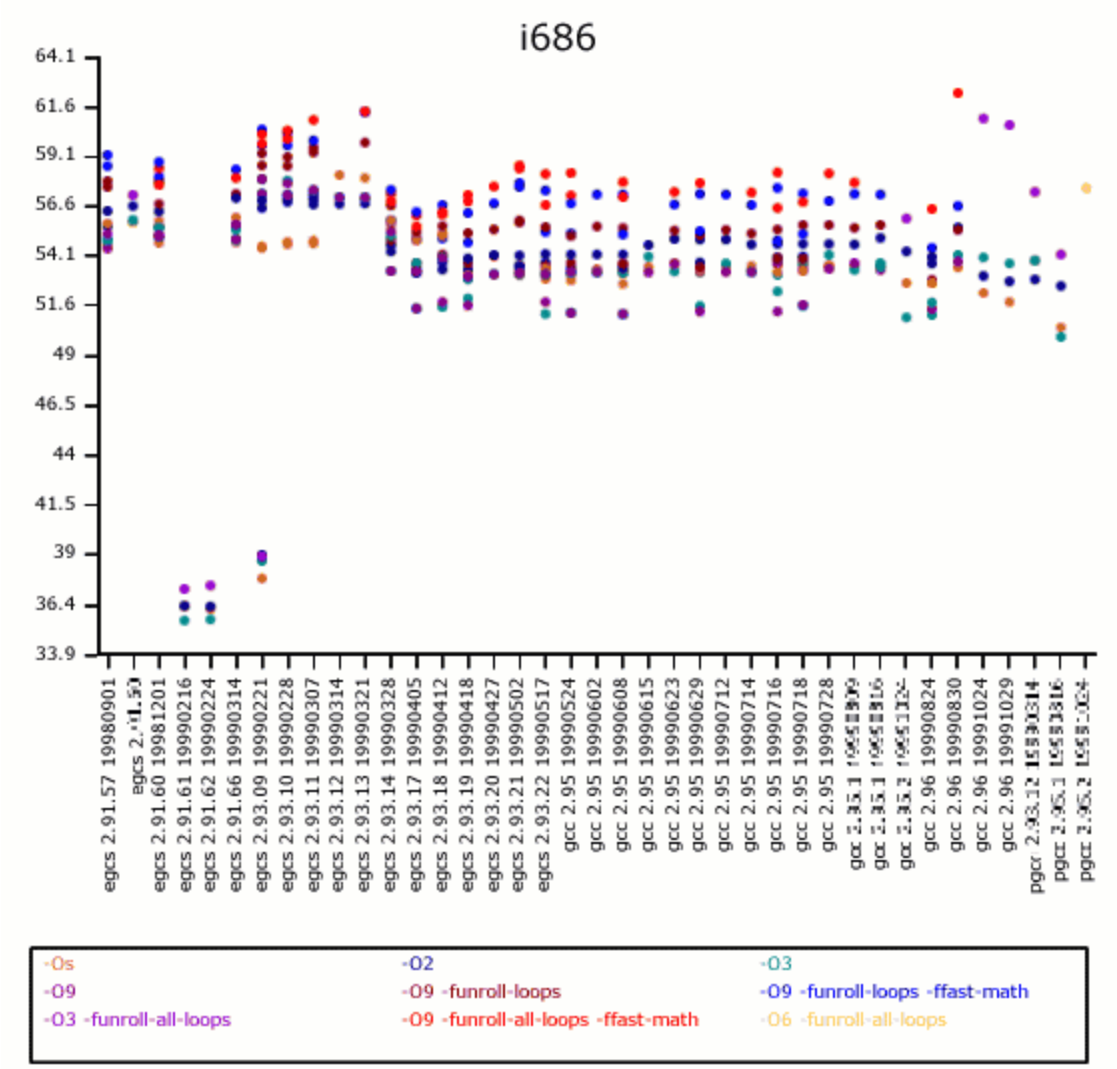
Depending of your processor architecture and the version of your compiler (GCC/EGCS), optimization options may vary. The charts bellow will help you to choose the best compilation flags for your compiler/CPU architecture.

Compiler version installed on your RedHat Linux 6.1 is egcs 2.91.66. But be sure to check it even so before choosing your compiler optimization options.

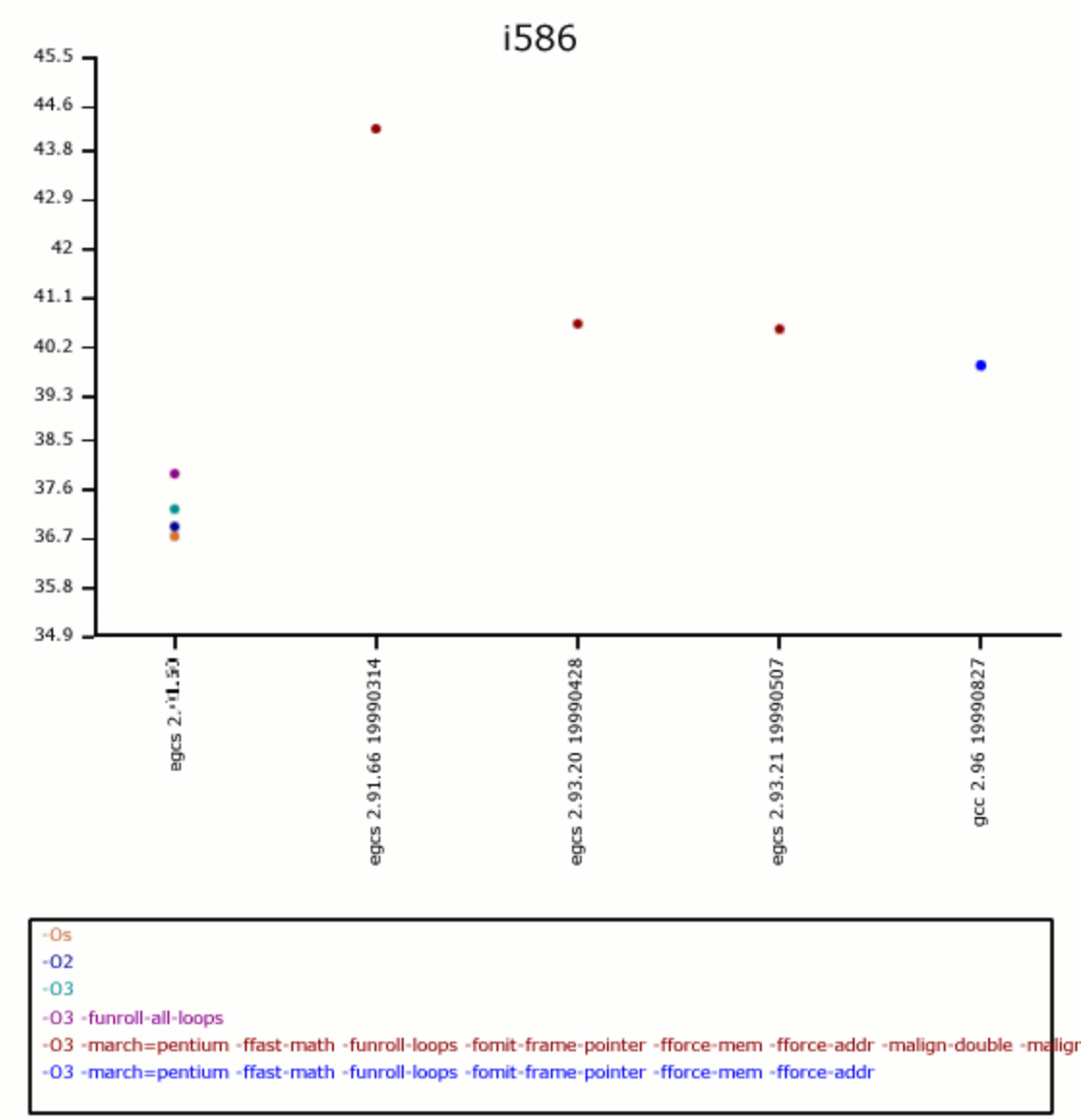
- To verify the compiler version installed on your system, use the command:
  [root@deep]# **egcs --version**

NOTE: All benchmark results and future result can be retrieve from the GCC home page at the following address: http://egcs.cygnus.com/

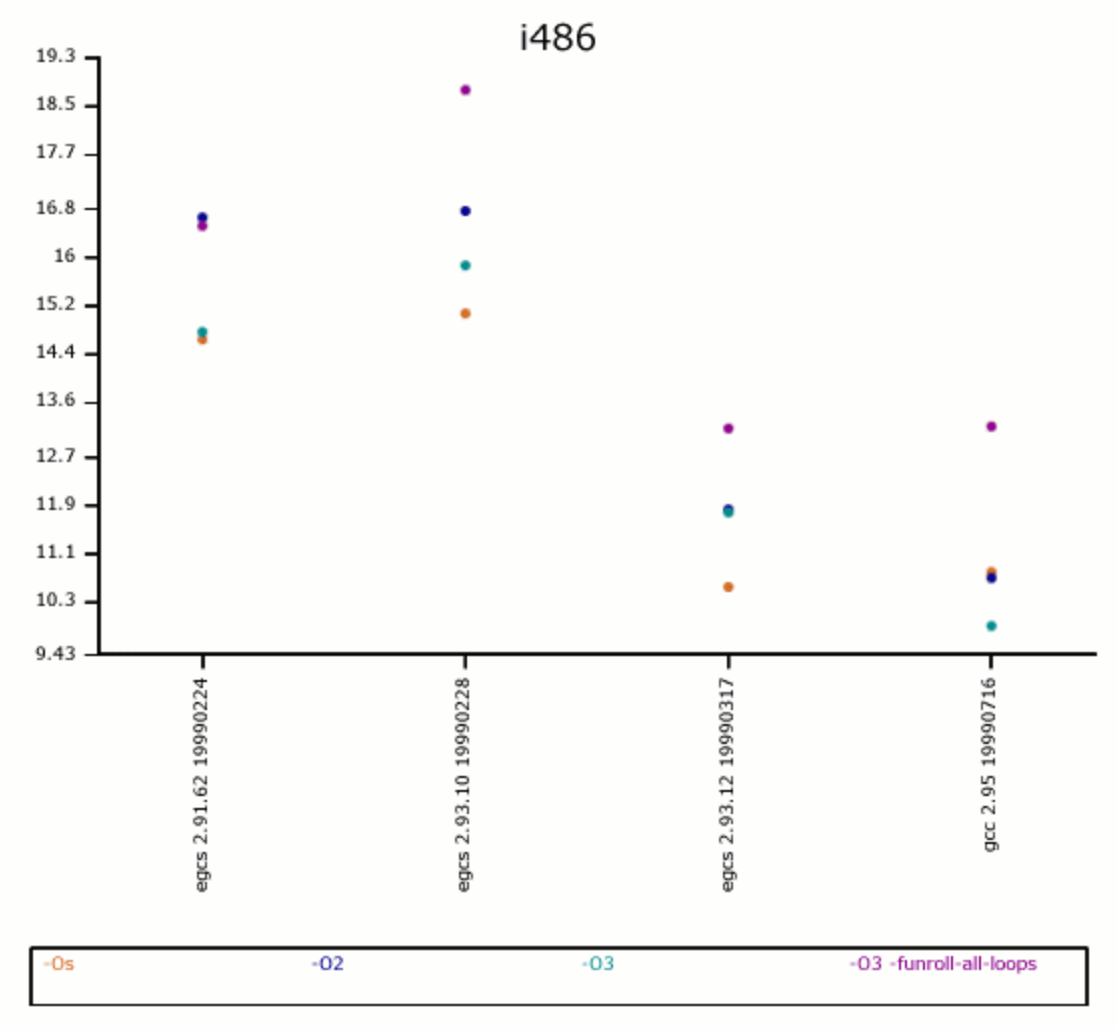
Now as an example :
For a CPU pentium II/III (i686) with compiler version egcs-2.91.66, the best optimization options will be: **CFLAGS='-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions'**



For a CPU pentium (i586) with compiler version egcs-2.91.66, the best optimization options will be: **CFLAGS='-O3 -march=pentium -mcpu=pentium -ffast-math -funroll-loops -fomit-frame-pointer -fforce-mem -fforce-addr -malign-double -fno-exceptions'**



For a CPU i486 with compiler version egcs-2.91.66, the best optimization options will be:
CFLAGS='-O3 -funroll-all-loops -malign-double -mcpu=i486 -march=i486 -fomit-frame-pointer -fno-exceptions'



-funroll-loops
Perform the optimization of loop unrolling. This is only done for loops whose number of iterations can be determined at compile time or run time.

-funroll-all-loops
Perform the optimization of loop unrolling. This is done for all loops and usually makes programs run more slowly.

-ffast-math
This option allows GCC to violate some ANSI or IEEE rules and/or specifications in the interest of optimizing code for speed. For example, it allows the compiler to assume arguments to the sqrt function are non-negative numbers and that no floating-point values are NaNs.

-malign-double

Control whether GCC aligns double, long double, and long long variables on a two-word boundary or a one-word boundary. Aligning double variables on a two-word boundary will produce code that runs somewhat faster on a `Pentium' at the expense of more memory.

-mcpu=cpu_type
Assume the defaults for the machine type cpu type when scheduling instructions. While picking a specific cpu type will schedule things appropriately for that particular chip, the compiler will not generate any code that does not run on the i386 without the `-march=cpu_type' option being used. `i586' is equivalent to `pentium' and `i686' is equivalent to `pentiumpro'. `k6' is the AMD chip as opposed to the Intel ones.

-march=cpu_type
Generate instructions for the machine type cpu type. The choices for cpu type are the same as for `-mcpu'. Moreover, specifying `-march=cpu_type' implies `-mcpu=cpu_type'.

-fforce-mem
Force memory operands to be copied into registers before doing arithmetic on them. This produces better code by making all memory references potential common subexpressions. When they are not common subexpressions, instruction combination should eliminate the separate register-load.

-fforce-addr
Force memory address constants to be copied into registers before doing arithmetic on them. This may produce better code just as `-fforce-mem' may.

-fomit-frame-pointer
Don't keep the frame pointer in a register for functions that don't need one. This avoids the instructions to save, set up and restore frame pointers; it also makes an extra register available in many functions. It also makes debugging impossible on most machines.

NOTE: All future optimization that will describe in this book refer by default for a Pentium II/III CPU family. So, you must if require adjust the compilation flag to your specific CPU processor type.

2. The "bdflush" parameter

This documentation is for the sysctl files in "/proc/sys/vm" and is valid for Linux kernel version 2.2. The files in this directory can be used to tune the operation of the virtual memory (VM) subsystem of the Linux kernel, and one of the files (bdflush) also has a little influence on disk usage.

This file (bdflush) controls the operation of the bdflush kernel daemon. We generally use this command to improve filesystem performance.

echo "100 1200 128 512 15 5000 500 1884 2">/proc/sys/vm/bdflush

Be changing some values from the default and the system seems more responsive, e.g. it waits a little more to write to disk and thus avoids some disk access contention.

Add the above commands to "**/etc/rc.d/rc.local"** file and you'll not have to type it again the next time if you reboot your system.

Look at "/usr/src/linux/Documentation/sysctl/vm.txt" for more information on how to improve kernel parameters related to virtual memory, disk cache, swap, etc.

3. The “ip_local_port_range” parameter

This documentation is for the sysctl files in “/proc/sys/net/ipv4/ip_local_port_range” and is valid for Linux kernel version 2.2. ip_local_port_range - 2 INTEGERS.

Defines the local port range that is used by TCP and UDP to choose the local port. The first number is the first (port), the second the last local port number. For high-usage systems change this to 32768-61000.

echo “32768 61000” > /proc/sys/net/ipv4/ip_local_port_range

Add the above commands to “**/etc/rc.d/rc.local”** file and you’ll not have to type it again the next time if you reboot your system.

4. The “/etc/nsswitch.conf” file

The “/etc/nsswitch.conf” file is used to configure which services are to be used to determine information such as hostnames, password files, and group files. The two last information “password files”, and “group files” in our case are not used since we don’t use NIS service in our server. So we will focus on the “hosts” line in this file.

Edit the **nsswitch.conf** file (vi /etc/nsswitch.conf) and change the “hosts” line to read:

"hosts: dns files"

which mean for programs that want to resolve an address to use dns feature first and after the “/etc/hosts” file if the DNS servers are not available or can’t resolve the address.

Also.. I would recommend to DELETE all instances of NIS from each line of this file UNLESS you *ARE* using NIS! The result must look like this:

passwd: **files**
shadow: **files**
group: **files**
hosts: **dns files**
bootparams: **files**
ethers: **files**
netmasks: **files**
networks: **files**
protocols: **files**
rpc: **files**
services: **files**
automount: **files**
aliases: **files**

5. The “/proc” filesystem

This documentation is for the sysctl files in “/proc/sys/fs” and is valid for Linux kernel version 2.2. The files in this directory can be used to tune and monitor miscellaneous and general things in the operation of the Linux kernel. Since some of the files _can_ be used to screw up your system, it is advisable to read both documentation and source before actually making adjustments.

Increase the value of “**/proc/sys/fs/file-max”** to something reasonable like 256 for every 4M of RAM you have: i.e. for a 128 M machine, set it to 8192 (128/4=32 32*256=8192). As above, also increase the “**/proc/sys/fs/inode-max”** to a value roughly 3 to 4 times (8192*4=32768) the

number of open files. This is because the number of inodes open is at least one per open file, and often much larger than that for large files.

The canonical command to change anything in the "/proc" hierarchy is (as root) **echo "newvalue" >/proc/file/that/you/want/to/change**, so for this item the command line is

echo "8192" >/proc/sys/fs/file-max
echo "32768" >/proc/sys/fs/inode-max

The above technique or modifying the constants in the kernel sources. Not usually the right answer because that will not survive a new kernel source tree. One of the best techniques is to add the above commands to "**/etc/rc.d/rc.local"** file.

[root@deep]# vi /etc/rc.d/rc.local and add at the end of this file the following lines

echo "8192" >/proc/sys/fs/file-max (If you have a 128MB of RAM in your system).
echo "32768" >/proc/sys/fs/inode-max (If you have a 128MB of RAM in your system).

The exact number will vary from the above formula based on what you are actually doing with the machine. A file server or web server need a lot of open files, for instance, but a compute server does not.

Very large memory systems, especially 512 Megabytes or larger, probably should not have more than 50,000 open files and 150,000 open inodes.

The value in **file-max** denotes the maximum number of file handles that the Linux kernel will allocate. When you get a lot of error messages about running out of file handles, you might want to raise this limit. The default value is 4096.

The value in **inode-max** denotes the maximum number of inode handlers. This value should be 3 to 4 times larger than the value in file-max, since stdin, stdout, and network sockets also need an inode struct to handle them. If you regularly run out of inodes, you should increase this value.

6. The "ulimit' parameter

Linux itself has a "Max Processes" per user limit. Add this to your root "**.bashrc"** file or whatever script your particular shell uses:

Edit the **.bashrc** file (vi /root/.bashrc) and add the following line:

ulimit -u unlimited

You must exit and re-login. To verify that you are ready to go, make sure that when you type **ulimit -a** as root, it shows "unlimited" next to **max user processes**.

[root@deep]# **ulimit -a**

```
core file size (blocks)       1000000
data seg size (kbytes)        unlimited
file size (blocks)            unlimited
max memory size (kbytes)      unlimited
stack size (kbytes)           8192
cpu time (seconds)            unlimited
max user processes            unlimited  ← this line.
pipe size (512 bytes)         8
open files                    1024
```

virtual memory (kbytes) 2105343

NOTE: you may also do *ulimit -u unlimited* at the command prompt instead of adding it to the "/root/.bashrc" file, but I always forgot, so I just added it in the "/root/.bashrc" file as a safety net.

7. Increases the system limit on open files

Increases the current process' limit. A process on Red Hat 6.0 with kernel 2.2.5 could open at least 31000 file descriptors this way and a process on kernel 2.2.12 can open at least 90000 file descriptors this way (with appropriate limits). The upper bound seems to be available memory.

Edit the **.bashrc** file (vi /root/.bashrc) and add the following line:

ulimit -n 90000

You must exit and re-login. To verify that you are ready to go, make sure that when you type **ulimit -a** as root, it shows "90000" next to **open files**.

[root@deep]# **ulimit -a**

core file size (blocks) 1000000
data seg size (kbytes) unlimited
file size (blocks) unlimited
max memory size (kbytes) unlimited
stack size (kbytes) 8192
cpu time (seconds) unlimited
max user processes unlimited
pipe size (512 bytes) 8
open files **90000 ← this line.**
virtual memory (kbytes) 2105343

NOTE: In older 2.2 kernels, though, the number of open files per process is still limited to 1024, even with the above changes.

8. The "atime" attribute

In addition to the information about when files were created and last modified, Linux also records when a file was last accessed. This information is not particularly useful, and there is a cost associated with recording it. The ext2 filesystem allows the superuser to mark individual files such that their last access time is not recorded.

This may lead to significant performance improvements when running find, and may also be useful on often accessed frequently changing files such as the contents of "/var/spool/news".

To set the attribute, use:

[root@deep]# **chattr +A filename**

For a whole directory tree, does something like:

[root@deep /root]# **chattr -R +A /var/spool/**
[root@deep /root]# **chattr -R +A /cache/**
[root@deep /root]# **chattr -R +A /home/httpd/ona/**

9. The “noatime” attribute

Linux has a mount option for filesystems call **noatime**. This option can be added to the mount options field in “/etc/fstab”. When a filesystem is mounted with this option, read accesses to the file will no longer result in an update to the atime information associated with the file. The atime info is generally not all that useful, so the lacks of updates to this field are not often relevant.

The importance of the **noatime** setting is that it eliminates the need to make writes to the filesystem for files, which are simply being read. Since writes tend to be somewhat expensive, this can result in measurable performance gains. Note that the wtime information will continue to be updated anytime the file is written to.

Edit the **fstab** file (vi /etc/fstab) and add for example in the line:

E.I: /dev/sda7 /chroot ext2 defaults,**noatime** 1 2

Reboot your system and then test your results with the command:

[root@deep]# **reboot**
[root@deep]# **cat /proc/mounts**

10. TCP/IP Stack specific

Red Hat Linux, out of the box, does NOT optimize the TCP/IP window size. This can make a BIG difference with performance. For more information, check out: RFC 1106 - High Latency WAN links - Section 4.1 and RFC 793 - Transmission Control Protocol.

Edit "/etc/sysconfig/network-scripts/ifup" and around lines 110, 112, 117, 125, and 134, find the lines:

110: "route add -net ${NETWORK} netmask ${NETMASK} ${DEVICE}"
112: "route add -host ${IPADDR} ${DEVICE}"
117: "route add default gw ${GATEWAY} metric 1 ${DEVICE}"
125: "route add default gw ${GATEWAY} ${DEVICE}"
134: "route add default gw $gw ${DEVICE}"

And change them to

110: "route add -net ${NETWORK} netmask ${NETMASK} **window 8192** ${DEVICE}"
112: "route add -host ${IPADDR} **window 8192** ${DEVICE}"
117: "route add default gw ${GATEWAY} **window 8192** metric 1 ${DEVICE}"
125: "route add default gw ${GATEWAY} **window 8192** ${DEVICE}"
134: "route add default gw $gw **window 8192** ${DEVICE}"

11. The swap partition

Try to put your swap partitions near the beginning of your drive. The beginning of the drive is physically located on the outer portion of the cylinder, so the read/write head can cover much more ground per revolution. I typically see partitions placed at the end work 3MB/s slower using hdparm -t.

12. Tuning IDE Hard Disk Performance

Performance increases have been reported on massive disk I/O operations by setting the IDE drivers to use DMA, 32-bit transfers and Multiple sector mode. The kernel seems to use more conservative settings unless told otherwise. The magic command to change the setting of your drive is “hdparm”.

To enable 32-bit I/O over the PCI buses, use the command:
[root@deep]# **/sbin/hdparm -c 1 /dev/hda** (or hdb, hdc etc).

The “hdparm” (8) manpage says that you may need to use -c 3 for some chipsets. All (E)IDE drives still have only a 16-bit connection over the ribbon cable from the interface card.

To enable DMA, use the command:
[root@deep]# **/sbin/hdparm -d 1 /dev/hda** (or hdb, hdc etc).

This may depend on support for your motherboard chipset being compiled into your kernel.

To enable multiword DMA mode 2 transfers, use the command:
[root@deep]# **/sbin/hdparm -d 1 -X34 /dev/hda** (or hdb, hdc etc).

This set the IDE transfer mode for newer (E)IDE/ATA2 drives. (check your hardware manual to see if you have it).

To enable UltraDMA mode2 transfers, use the command:
[root@deep]# **/sbin/hdparm -d 1 -X66 /dev/hda** (or hdb, hdc etc)

You'll need to prepare the chipset for UltraDMA beforehand, also see you man page about “hdparm” for more information. Use this with extreme caution!

To set multiple sector mode I/O, use the command:
[root@deep]# **/sbin/hdparm -m XX /dev/hda** (or hdb, hdc etc)

Where “XX” is the maximum setting supported by your drive. The -i flag can be used to find the maximum setting supported by an installed drive, look for MaxMultSect in the output.

[root@deep]# **/sbin/hdparm -i /dev/hda** (or hdb, hdc etc)

/dev/hda:

```
 Model=Maxtor 7540 AV, FwRev=GA7X4647, SerialNo=L1007YZS
 Config={ HardSect NotMFM HdSw>15uSec Fixed DTR>5Mbs FmtGapReq }
 RawCHS=1046/16/63, TrkSize=0, SectSize=0, ECCbytes=11
 BuffType=3(DualPortCache), BuffSize=32kB, MaxMultSect=8, MultSect=8
 DblWordIO=yes, maxPIO=2(fast), DMA=yes, maxDMA=1(medium)
 CurCHS=523/32/63, CurSects=379584528, LBA=yes, LBA=yes, LBAsects=1054368
 tDMA={min:150,rec:150}, DMA modes: sword0 sword1 *sword2 *mword0
 IORDY=on/off, tPIO={min:240,w/IORDY:180}, PIO modes: mode3
```

Multiple sector mode (aka IDE Block Mode), is a feature of most modern IDE hard drives, permitting the transfer of multiple sectors per I/O interrupt, rather than the usual one sector per interrupt. When this feature is enabled, it typically reduces operating system overhead for disk I/O by 30-50%. On many systems, it also provides increased data throughput of anywhere from 5% to 50%.

You can test the results of your changes by running “hdparm” in performance test mode:
[root@deep]# **/sbin/hdparm -t /dev/hda** (or hdb, hdc etc).

Once you have a set of "hdparm" options, don't forget to put the commands in your "/etc/rc.d/rc.local" file to run it every time you reboot the machine.

Part III Kernel-Related Reference

In this Part

Configuring and Building Kernels

Chapter 5 Configuring and Building Kernels

In this Chapter

Linux Kernel
Making an emergency boot floppy
Securing the kernel
Kernel configuration
Installing the new kernel
Delete program, file and lines related to modules
Making a new rescue floppy
Update your “/dev” entries

Linux Kernel

Overview

The first thing to do next is build a kernel that best suits your system. It's very simple to do but, in any case, refer to the README file in "/usr/src/linux/". When configuring your kernel only compile in code you need and use. Four main reason come into mind; the Kernel will be faster (Less code to run), you will have more memory (Kernel parts are NEVER swapped to the virtual memory), more stable (Ever probed for a non-existent card?), unnecessary parts can be used by an attacker gain access to the machine or other machines. Modules are also slower than support compiled directly in the kernel.

In our configuration and compilation we will build a monolithic kernel. Monolithic kernel mean to only answer **Yes** or **No** to the questions (don't make anything modular) and omit the steps: make_modules and make_modules_install. Also we will patch our new kernel with the buffer overflow protection from kernel patches. Patches for the Linux kernel exist, like Solar Designer's non-executable stack patch, which disallow the execution code on the stack, making a number of buffer overflow attacks harder - and defeating completely a number of current exploits used by "script kiddies" worldwide.

Remember that the steps above to only answer **Yes** or **No** to the questions when configuring your new kernel are requiring only if you're intended to build a monolithic kernel. If you intended to use firewall masquerading function or dial up ppp connection, you cannot build a monolithic kernel, since these function require the build by default of some modules. Build instead a modularized kernel.

A new kernel is very specific to your computer hardware, in the kernel configuration part, I assume the following hardware for my example. Of course you must change them to fit you system component.

1 Pentium II 400 MHz (i686) processor
1 Motherboard SCSI
1 Hard Disk SCSI
1 SCSI Controler Adaptec AIC 7xxx
1 CD-ROM ATAPI IDE
1 Floppy Disk
2 Ethernet Cards Intel EtherExpressPro 10/100
1 Mouse PS/2

These installation instructions assume

Commands are Unix-compatible.
The source path is /usr/src.
Installations were tested on RedHat Linux 6.1 Server.
All steps in the installation will happen in superuser account "root".
Lasted Kernel version number is 2.2.14
Lasted Secure Linux Kernel Patches version number is 2_2_14-ow1

Packages

Kernel Homepage: http://www.kernelnotes.org/
You must be sure to download: linux-2_2_14_tar.gz
Secure Linux Kernel Patches Homepage: http://www.openwall.com/linux/
You must be sure to download: linux-2_2_14-ow1_tar.gz

Making an emergency boot floppy

The first pre-install step is to make an emergency boot floppy (if you haven't made one already). You can simply do this with the mkbootdisk command.

First find out what kernel, you are currently using. Check out your "/etc/lilo.conf" file and see which image was booted from. On my example, I have the following in the file.

```
[root@deep]# cat /etc/lilo.conf
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
image=/boot/vmlinuz-2.2.12-20
        label=linux
        root=/dev/sda6
        initrd=/boot/initrd-2.2.12-20.img
        read-only
```

Now you will need to find the image that you booted from. On a standard install, it will be the one-labeled linux. The above example shows that the machine booted using the "/boot/vmlinuz-2.2.12-20" kernel. Now simply put a formatted 1.44 floppy in your system, and make sure you have logged in as root.

```
[root@deep]# mkbootdisk --device /dev/fd0 2.2.12-20
Insert a disk in /dev/fd0. Any information on the disk will be lost.
Press <Enter> to continue or ^C to abort:
```

Following these guidelines, you will now have a boot floppy with a known working kernel in case of problems with the upgrade. I recommend rebooting the system with the floppy to make sure that the floppy works correctly.

Optimization

Decompress the tarball (tar.gz).

```
[root@deep]# cp linux-version_tar.gz /usr/src/
[root@deep]# cd /usr/src/
[root@deep]# rm -rf linux (This is a symbolic link)
[root@deep]# rm -rf linux-old.version.number (This is your actual directory of kernel header files)
```

NOTE: The steps above of removing the Linux symbolic link (rm -rf linux) and Linux headers directory (linux-old.version.number), are require only if you already have installed a Linux kernel with a tar archive before. If it is a first, fresh install of Linux kernel, then instead uninstall the kernel-headers-version.i386.rpm, kernel-version.i386.rpm package that must be on your system and the directory (/usr/src/linux) will be automatically removed with all related modules files (/lib/modules/2.2.XX).

If a kernel RPM package is installed on your system instead of tar archive, because you have just finished to install your new Linux system, or have using a RPM package before to upgrade you Linux system, then use the following command to uninstall the Linux kernel:

- You can verify if a kernel RPM packages are installed on your system with the following command:
  ```
  [root@deep]# rpm -qa | grep kernel
  kernel-headers-2.2.12-20.i386.rpm
  kernel-2.2.12-20.i386.rpm
  ```

- To uninstall the linux kernel, use the following command:
  [root@deep]# **rpm -e --nodeps kernel-headers kernel**

In the step bellow, we will remove manually the empty "/usr/src/linux-2.2.12" and "/lib/modules/2.2.12" directories after the uninstallation of the kernel RPM (the RPM uninstall program, will not remove completetly those directory). We will untar our new Linux version from the tar archive, change the owner of the new Linux directory created after the decompression to be the super-user "root" and finally remove the Linux tar archive from the system.

[root@deep]# **rm -rf /usr/src/linux-2.2.12/**
[root@deep]# **rm -rf /lib/modules/2.2.12-20/**
[root@deep]# **tar xzpf linux-version_tar.gz**
[root@deep]# **chown -R 0.0 /usr/src/linux/**
[root@deep]# **rm -f linux-version_tar.gz**

Increase the Tasks
To increase the number of tasks allowed (the maximum number of processes per user), you may need to edit the "/usr/src/linux/include/linux/tasks.h" file and change the following parameters.

- Edit the **tasks.h** file (vi +14 /usr/src/linux/include/linux/tasks.h) and change:
  NR_TASKS from 512 to **3072**
  MIN_TASKS_LEFT_FOR_ROOT from 4 to **24**

NOTE: 1) The value in NR_TASKS denotes the maximum number of tasks (processes) handles that the Linux kernel will allocate per users. Increasing this number will allow to handle more connection from client on your server (example an HTTP web server will be able to serve more client connections). 2) Linux is protected to avoid allocating all process slots by normal users. There are reserved MIN_TASKS_LEFT_FOR_ROOT slots for root. So there should be also protection to avoid allocating all memory by normal users.

Optimize the kernel
To optimize the Linux kernel to fit your specific CPU architecture and optimization flags, you may need to edit the "/usr/src/linux/Makefile" file and change the following parameters.

- Edit the **Makefile** file (vi +90 /usr/src/linux/Makefile) and change the line:
  CFLAGS = -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer
  To read:
  CFLAGS = -Wall -Wstrict-prototypes -O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions

- Edit the **Makefile** file (vi +19 /usr/src/linux/Makefile) and change the line:
  HOSTCFLAGS = -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer
  To read:
  HOSTCFLAGS = -Wall -Wstrict-prototypes -O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions

Which turns on an aggressive optimization tricks that may or may not work with all kernels. Please, if the optimization flags above or the one you have chosen for your CPU architecture doesn't work for you, don't try to absolutely force it to work. I don't want to make your system became unstable like Microsoft Window.

Securing the kernel

The secure Linux kernel patches from the Openwall Project are a great way to prevent attacks like Stack Buffer Overflows and other. This patch is a collection of security-related features for the

Linux kernel, all configurable via the new '"Security options" configuration section that will be added to your new Linux kernel.

In addition to the new features, some versions of the patch contain various security fixes. The number of such fixes changes from version to version, as some are becoming obsolete (such as because of the same problem getting fixed with a new kernel release), while other security issues are discovered.

New features of patch version linux-2_2_14-ow1_tar.gz are:
Non-executable user stack area
Restricted links in /tmp
Restricted FIFOs in /tmp
Restricted /proc
Special handling of fd 0, 1, and 2
Enforce RLIMIT_NPROC on execve(2)
Destroy shared memory segments not in use

NOTE: When applying the linux-2_2_14-ow1 patch, new "Security options" section will be added at the end of your kernel configuration. For more information and description of the different features available with this patch, see the README file that come with the source code of the patch.

Applying the patch

Decompress the tarball (tar.gz).

```
[root@deep]# cp linux-2_2_14-ow1_tar.gz /usr/src/
[root@deep]# cd /usr/src/
[root@deep]# tar xzpf linux.2_2_14-ow1_tar.gz
[root@deep]# cd linux-2.2.14-ow1/
[root@deep]# mv linux-2.2.14-ow1.diff /usr/src/
[root@deep]# cd ..
[root@deep]# patch -p0 < linux-2.2.14-ow1.diff
[root@deep]# rm -rf linux-2.2.14-ow1
[root@deep]# rm -f linux-2.2.14-ow1.diff
[root@deep]# rm -f linux-2_2_14-ow1_tar.gz
```

First we copy the program archive to the "/usr/src" directory, then we move to the "/usr/src" directory and decompress the linux-2_2_14ow1_tar.gz archive, we move to the new uncompressed linux patch, move the file linux-2.2.14-ow1.diff containing the patch to the "/usr/src", return to the "/usr/src" and patch our kernel with the file linux-2.2.14-ow1.diff. After, we remove all files related to the patch.

NOTE: All security messages related to the linux-2.2.14-ow1 patch like non-executable stack part should get logged to the log file "/var/log/messages".

The step of patching your new kernel is completed. Now follow the rest of this installation to build the Linux kernel and reboot.

Compilation

Make sure your "/usr/include/asm", "/usr/include/linux", and "/usr/include/scsi" subdirectories are just symlinks to the kernel sources. The "asm", "linux", and "scsi" subdirectories are a soft link to the real include directories needed for this architecture, for example "/usr/src/linux/include/asm-i386" for "asm". If you have a freshly unpacked kernel source tree, you must make symlinks.

- Type the following commands on your terminal:
  [root@deep]# **cd /usr/include/**

```
[root@deep]# rm -rf asm linux scsi
[root@deep]# ln -s /usr/src/linux/include/asm-i386 asm
[root@deep]# ln -s /usr/src/linux/include/linux linux
[root@deep]# ln -s /usr/src/linux/include/scsi scsi
```

This is a very important part of the configuration, we remove the “asm”, “linux”, and “scsi” directories under “/usr/include” then build a new links that point to the same name directories under the new Linux kernel version directory. The “include” directory contains important header files needed by your linux kernel and programs to by able to compile on your system.

Make sure you have no stale .o files and dependencies lying around.

- Type the following commands on your terminal:
  [root@deep]# **cd /usr/src/linux/**
  [root@deep]# **make mrproper**

NOTE: These first two steps above simply clean up any cruft that might have accidentally been left in the source tree by the development team.

You should now have the sources correctly installed. You can configure the Linux kernel in one of three ways. The first method is to use the **make config** command. It provides you with a text-based interface for answering all the configuration options. You are prompted for all the options you need to set up your kernel.

The second method is to use the **make menuconfig** command, which provides all the kernel options in an easy-to-use menu. The third is to use the **make xconfig** command, which provides a full graphical interface to all the kernel options.

For configuration in this chapter, you will use the **make config** command because we are not installed the XFree86 window Interface on our server.

[root@deep]# **cd /usr/src/linux/** (if you are not already in this directory).
[root@deep]# **make config**

kernel configuration

Code maturity level options
Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL) [N]

Processor type and features
Processor family (CONFIG_M386) [Ppro/6x86MX]
Maximum physical Memory (CONFIG_1GB) [1GB]
Math emulation (CONFIG_MATH_EMULATION) [N]
MTRR Memory Type Range Register support (CONFIG_MTRR) [N]
Symmetric multi-processing support (CONFIG_SMP) [Y] **N**

Loadable module support
Enable loadable module support (CONFIG_MODULES) [Y] **N**

General setup
Networking support (CONFIG_NET) [Y]
PCI support (CONFIG_PCI) [Y]
PCI access mode (BIOS, Direct, Any) [Any]
PCI quirks (CONFIG_PCI_QUIRKS) [Y] **N**
Backward-compatible /proc/pci (CONFIG_PCI_OLD_PROC) [Y] **N**
MCA support (CONFIG_MCA) [N]
SGI Visual Workstation support (CONFIG_VISWS) [N]

System V IPC (CONFIG_SYSVIPC) [Y]
BSD Process Accounting (CONFIG_BSD_PROCESS_ACCT) [N]
Sysctl support (CONFIG_SYSCTL) [Y]
Kernel support for a.out binaries (CONFIG_BINFMT_AOUT) [Y]
Kernel support for ELF binaries (CONFIG_BINFMT_ELF) [Y]
Kernel support for MISC binaries (CONFIG_BINFMT_MISC) [Y]
Parallel port support (CONFIG_PARPORT) [N]
Advanced Power Management BIOS supports (CONFIG_APM) [N]

Plug and Play support
Plug and Play support (CONFIG_PNP) [N]

Block devices
Normal PC floppy disk support (CONFIG_BLK_DEV_FD) [Y]
Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support (CONFIG_BLK_DEV_IDE) [Y]
Use old disk-only driver on primary interface (CONFIG_BLK_DEV_HD_IDE) [N]
Include IDE/ATA-2 disk support (CONFIG_BLK_DEV_IDEDISK) [Y]
Include IDE/ATAPI CDROM support (CONFIG_BLK_DEV_IDECD) [Y]
Include IDE/TAPE support (CONFIG_BLK_DEV_IDETAPE) [N]
Include IDE/FLOPPY support (CONFIG_BLK_DEV_IDEFLOPPY) [N]
SCSI emulation support (CONFIG_BLK_DEV_IDESCSI) [N]
CMD640 chipset bugfix/support (CONFIG_BLK_DEV_CMD640) [Y] **N**
RZ1000 chipset bugfix/support (CONFIG_BLK_DEV_RZ10000 [Y] **N**
Generic PCI IDE chipset support (CONFIG_BLK_DEV_IDEPCI) [Y]
Generic PCI bus-master DMA support (CONFIG_BLK_DEV_IDEDMA) [Y]
Boot off-board chipsets first support (CONFIG_BLK_DEV_OFFBOARD) [N]
Use DMA by default when available (CONFIG_IDEDMA_AUTO) [Y]
Other IDE chipset support (CONFIG_IDE_CHIPSETS) [N]
Loopback device support (CONFIG_BLK_DEV_LOOP) [N]
Network block device driver support (CONFIG_BLK_DEV_NBD) [N]
Multiple device driver support (CONFIG_BLK_DEV_MD) [N]
RAM disk support (CONFIG_BLK_DEV_RAM) [N]
XT hard disk support (CONFIG_BLK_DEV_XD) [N]
Mylex DAC960/DAC1100 PCI RAID Controller support (CONFIG_BLK_DEV_DAC960) [N]
Parallel port IDE device support (CONFIG_PARIDE) [N]
Compaq SMART2 support (CONFIG_BLK_CPQ_DA) [N]

Networking options
Packet socket (CONFIG_PACKET) [Y]
Kernel/user netlink socket (CONFIG_NETLINK) [N]
Network firewalls (CONFIG_FIREFALL) [N] **Y**
Socket filtering (CONFIG_FILTER) [N]
Unix domain sockets (CONFIG_UNIX) [Y]
TCP/IP networking (CONFIG_INET) [Y]
IP:Multicasting (CONFIG_IP_MULTICAST) [N]
IP:Advanced router (CONFIG_IP_ADVENCED_ROUTER) [N]
IP:Kernel level autoconfiguration (CONFIG_IP_PNP) [N]
IP:Firewalling (CONFIG_IP_FIREWALL) [N] **Y**
IP:Transparent proxy support (CONFIG_IP_TRANSPARENT_PROXY) [N]
IP:Masquerading (CONFIG_IP_MASQUERADE0 [N]
IP:ICMP masquerading (CONFIG_IP_MASQUERADE_ICMP) [N]
IP:Optimize as router not host (CONFIG_IP_ROUTER) [N]
IP:Tunneling (CONFIG_NET_IPIP) [N]
IP:GRE tunnels over IP (CONFIG_NET_IPGRE) [N]
IP:Aliasing support (CONFIG_IP_ALIAS) [N]
IP:TCP syncookie support (CONFIG_SYN_COOKIES) [N] **Y**
IP:Reverse ARP (CONFIG_INET_RARP) [N]
IP:Allow large windows (CONFIG_SKB_LARGE) [Y]
The IPX protocol (CONFIG_IPX) [N]
AppleTalk DDP (CONFIG_ATALK) [N]

Telephony support
Linux telephony support (CONFIG_PHONE) [N/y/m/?] (NEW)

SCSI support
SCSI support (GONFIG_SCSI) [Y]
SCSI disk support (CONFIG_BLK_DEV_SD) [Y]
SCSI tape support (CONFIG_CHR_DEV_ST) [N]
SCSI CD-ROM support (CONFIG_BLK_DEV_SR) [N]
SCSI generic support (CONFIG_CHR_DEV_SG) [N]
Probe all LUMs on each SCSI device (CONFIG_SCSI_MULTI-LUM) [Y] **N**
Verbose SCSI error reporting (kernel size +=12K) (CONFIG_SCSI_CONSTANTS) [Y] **N**
SCSI logging facility (CONFIG_SCSI_LOGGING) [N]

SCSI low-level drivers
7000FASST SCSI support (CONFIG_SCSI_7000FASST) [N]
ACARD SCSI support (CONFIG_SCSI_ACARD) [N]
Adaptec AHA152X/2825 support (CONFIG_SCSI_AHA152X) [N]
Adaptec AHA1542 support (CONFIG_SCSI_AHA1542) [N]
Adaptec AHA1740 support (CONFIG_SCSI_AHA1740) [N]
Adaptec AIC7xxx support (CONFIG_SCSI_AIC7XXX) [N] **Y**
Enable Tagged Command Queuering (CONFIG_AIC7XXX_TCQ_ON_BY_DEFAULT) [N] **Y**
Maximum number of TCQ commands per device (CONFIG_AIC7XXX_CMDS_PER_DEVICE) [8]
Collect statistics to report in /proc (CONFIG_AIC7XXX_PROC_STATS) [N]
Delay in seconds after SCSI bus reset (CONFIG_AIC7XXX_RESET_DELAY) [5]
IBM ServeRAID support (CONFIG_SCSI_IPS) [N]
AdvanSys SCSI support (CONFIG_SCSI_ADVANSYS) [N]
Always IN2000 SCSI support (CONFIG_SCSI_IN2000) [N]
AM53/79C974 PCI SCSI support (CONFIG_SCSI_AM53C974) [N]
AMI MegaRAID support (CONFIG_SCSI_MEGARAID) [N]
BusLogic SCSI support (CONFIG_SCSI_BUSLOGIC) [N]
DTC3180/3280 SCSI support (CONFIG_SCSI_DTC3280) [N]
EATA ISA/EISA/PCI (DPT and generic EATA/DMA-compliant boards) support (CONFIG_SCSI_EATA) [N]
EATA-DMA (DPT, NEC, AT&T, SNI, AST, Olivetti, Alphatronix) support (CONFIG_SCSI_EATA_DMA) [N]
EATA-PIO (old DPT PM2001, PM2012A) support (CONFIG_SCSI_EATA_PIO) [N]
Future Domain 16xx SCSI/AHA-2920A support (CONFIG_SCSI_FUTURE_DOMAIN) [N]
GDT SCSI Disk Array Controller support (CONFIG_SCSI_GDTH) [N]
Generic NCR5380/53c400 SCSI support (CONFIG_SCSI_GENERIC_NCR5380) [N]
Initio 9100U(W) support (CONFIG_SCSI_INITIO) [N]
Initio INI-A100U2W support (CONFIG_SCSI_INIA100) [N]
NCR53c406a SCSI support (CONFIG_SCSI_NCR53C406A) [N]
symbios 53c416 SCSI support (CONFIG_SCSI_SYM53C416) [N]
Simple 53c710 SCSI support (Compaq, NCR machines) (CONFIG_SCSI_SIM710) [N]
NCR53c7,8xx SCSI support (CONFIG_SCSI_NCR53C7xx) [N]
NCR53C8XX SCSI support (CONFIG_SCSI_NCR53C8XX) [N]
SYM53C8XX SCSI support (CONFIG-SCSI_SYM53C8XX) [Y] **N**
PAS16 SCSI support (CONFIG_SCSI_PAS16) [N]
PCI2000 support (CONFIG_SCSI_PCI2000) [N]
PCI2220i support (CONFIG_SCSI_PCI2220I) [N]
PSI240i support (CONFIG_SCSI_PSI240I) [N]
Qlogic FAS SCSI support (CONFIG_SCSI_QLOGIC_FAS) [N]
Qlogic ISP SCSI support (CONFIG_SCSI_QLOGIC_ISP) [N]
Qlogic ISP FC SCSI support (CONFIG_SCSI_QLOGIC_FC) [N]
Seagate ST-02 and Future Domain TMC-8xx SCSI support (CONFIG_SCSI_SEAGATE) [N]
Tekram DC390(T) and Am53/79C974 SCSI support (CONFIG_SCSI_DC390T) [N]
Trantor T128/T128F/T228 SCSI support (CONFIG_SCSI_T128) [N]
UltraStor 14F/34F support (CONFIG_SCSI_U14_34F) [N]
UltraStor SCSI support (CONFIG_SCSI_ULTRASTOR) [N]

Network device support
Network device support (CONFIG_NETDEVICES) [Y]

ARQnet devices
ARCnet support (CONFIG_ARCNET) [N]
Dummy net driver support (CONFIG_DUMMY) [M] **Y**
EQL (serial line load balancing) support (CONFIG_EQUALIZER) [N]
General Instruments Surfboard 1000 (CONFIG_NET_SB1000) [N]

Ethernet (10 or 100Mbit)
Ethernet (10 or 100Mbit) (CONFIG_NET_ETHERNET) [Y]
3COM cards (CONFIG_NET_VENDOR_3COM) [N]
AMD LANCE and PCnet (AT1500 and NE2100) support (CONFIG_LANCE) [N]
Western Digital/SMC cards (CONFIG_NET_VENDOR_SMC) [N]
Racal-Interlan (Micom) NI cards (CONFIG_NET_VENDOR_RACAL) [N]
Other ISA cards (CONFIG_NET_ISA) [N]
EISA, VLB, PCI and on board controllers (CONDIF_NET_EISA) [Y]
AMD PCnet32 (VLB and PCI) support (CONFIG_PCNET32) [N]
Apricot Xen-II on board Ethernet (CONFIG_APRICOT) [N]
CS89x0 support (CONFIG_CS89x0) [N]
DM9102 PCI Fast Ethernet Adapter support (EXPERIMENTAL) (CONFIG_DM9102) [N]
Generic DECchip & DIGITAL EtherWORKS PCI/EISA (CONFIG_DE4X5) [N]
DECchip Tulip (dc21x4x) PCI support (CONFIG_DEC_ELCP) [N]
Digi Intl. RightSwitch SE-X support (CONFIG_DGRS) [N]
EtherExpressPro/100 support (CONFIG_EEXPRESS_PRO100) [Y]
PCI NE2000 support (CONFIG_NE2K_PCI) [N]
TI ThunderLAN support (CONFIG_TLAN) [N]
VIA Rhine support (CONFIG_VIA_RHINE) [N]
SiS 900/7016 PCI Fast Ethernet Adapter support (CONFIG_SIS900) [N/y/m/?] (NEW)
Pocket and portable adaptors (CONFIG_NET_POCKET) [N]
SysKonnect SK-98xx support (CONFIG_SK98LIN) [N/y/m/?] (NEW)
FDDI driver support (CONFIG_FDDI) [N]
PPP (point-to-point) support (CONFIG_PPP) [N]
SLIP (serial line) support (CONFIG_SLIP) [N]
Wireless LAN (non-hamradio) (CONFIG_NET_RADIO) [N]

Token ring devices
Token Ring driver support (CONFIG_TR) [N]
Fibre Channel driver support (CONFIG_NET_FC) [N]

Wan interfaces
Comtrol Hostess SV-11 support (CONFIG_HOSTESS_SV11) [N]
COSA/SRP sync serial boards support (CONFIG_COSA) [N]
Sealevel Systems 4021 support (CONFIG_SEALEVEL_4021) [N]
MultiGate (COMX) synchronous serial boards support (CONFIG_COMX) [N/y/m/?] (NEW)
Frame relay DLCI support (CONFIG_DLCI) [N]
WAN drivers (CONFIG_WAN_DRIVERS) [N]
SBNI12-xx support (CONFIG_SBNI) [N]

Amateur Radio support
Amateur Radio support (CONFIG_HAMRADIO) [N]

IrDA subsystem support
IrDA subsystem support (CONFIG_IRDA) [N]

ISDN subsystem
ISDN support (CONFIG_ISDN) [N]

Old CD-ROM drivers (not SCSI, not IDE)
Support non-SCSI/IDE/ATAPI CDROM drives (CONFIG_CD_NO_IDESCSI) [N]

Character devices
Virtual terminal (CONFIG_VT) [Y]

Support for console on virtual terminal (CONFIG_VT_CONSOLE) [Y]
Standard/generic (dumb) serial support (CONFIG_SERIAL) [Y]
Support for console on special port (CONFIG_SERIAL_CONSOLE) [N]
Extended dumb serial driver options (CONFIG_SERIAL_EXTENDED) [N]
Non-standard serial port support (CONFIG_SERIAL_NONSTANDAR) [N]
Unix98 PTY support (CONFIG_UNIX98_PTYS) [Y]
Maximum numbers of Unix98 PTYs in use (0-2048) (CONFIG_UNIX98_PTY_COUNT) [256] **128**
Mouse support (Not serial mice) (CONFIG_MOUSE) [Y]

Mice
ATIXL busmouse support (CONFIG_ATIXL_BUSMOUSE) [N]
Logitech busmouse support (CONFIG_BUSMOUSE) [N]
Microsoft busmouse support (CONFIG_MS_BUSMOUSE) [N]
PS/2 mouse (aka “auxiliary device”) support (CONFIG_PSMOUSE) [Y]
C&T 82C710 mouse port support (CONFIG_82c710_MOUSE) [Y] **N**
PC110 digitizer pad support (CONFIG_PC110_PAD) [N]

Joystick support
Joystick support (CONFIG_JOYSTICK) [N]
QIC-02 tape support (CONFIG_QIC02_TAPE) [N]
Watchdog Timer support (CONFIG_WATCHDOG) [N]
/dev/nvram support (CONFIG_NVRAM) [N]
Enhanced Real Time Clock support (CONFIG_RTC) [N]

Video for Linux
Video for Linux (CONFIG_VIDEO_DEV) [N]
Double Talk PC internal speech controller support (CONFIG_DTLK) [N]

Ftape, the floppy tape device driver
Ftape (QIC-80/Travan) support (CONDFIG_FTAPE) [N]

Filesystems
Quota support (CONFIG_QUOTA) [N] **Y**
Kernel automounter support (CONFIG_AUTOFS_FS) [Y] **N**
Amiga FFS filesystem support (CONFIG_AFFS_FS) [N]
Apple Macintosh filesystem support (CONFIG_HFS_FS) [N]
DOS FAT fs support (CONFIG_FAT_FS) [N]
ISO 9660 CDROM filesystem support (CONFIG_ISO9660FS) [Y]
Microsoft Joliet CDROM extensions (CONFIG_JOLIET) [N]
Minix fs support (CONFIG_MINIX_FS) [N]
NTFS filesystem support (CONFIG_NTFS_FS) [N]
OS/2 MPFS filesystem support (CONFIG_HPFS_FS) [N]
/proc filesystem support (CONFIG_PROC_FS) [Y]
/dev/pts filesystem support (CONFIG_DEVPTS_FS) [Y]
ROM filesystem support (CONFIG_ROMFS_FS) [N]
Second extended filesystem (CONFIG_EXT2_FS) [Y]
System V and coherent filesystem support (CONFIG_SYSV_FS) [N]
UFS filesystem support (CONFIG_UFS_FS) [N]

Network File Systems
Coda filesystem support (Advanced Network fs) (CONFIG_CODA_FS) [N]
NFS filesystem support (CONFIG_NFS_FS) [Y] **N**
SMB filesystem support (CONFIG_SMB_FS) [N]
NCP filesystem support (CONFIG_NCP_FS) [N]

Partition Types
BSD disklabel (BSD partition tables) support (CONFIG_BSD_DISKLABEL) [N]
Macintosh partition map support (CONFIG_MAC_PARTION) [N]
SMD disklabel (Sun partition tables) support (CONFIG_SMD_DISKLABEL) [N]
Solaris (x86) partition table support (CONFIG_SOLARIS_X86_PARTITION) [N]

Console drivers
VGA text console (CONFIG_VGA_CONSOLE) [Y]
Video mode selection support (CONFIG_VIDEO_SELECT) [N]

Sound
Sound card support (CONFIG_SOUND) [N]

(Security options will appear only if your are patched your kernel with the Openwall Project patch).
Security options
Non-executable user stack area (CONFIG_SECURE_STACK) [Y]
Autodetect and emulate GCC trampolines (CONFIG_SECURE_STACK_SMART) [Y]
Restricted links in /tmp (CONFIG_SECURE_LINK) [Y]
Restricted FIFOs in /tmp (CONFIG_SECURE_FIFO) [Y]
Restricted /proc (CONFIG_SECURE_PROC) [N] **Y**
Special handling of fd 0, 1, and 2 (CONFIG_SECURE_FD_0_1_2) [Y]
Enforce RLIMIT_NPROC on execve(2) (CONFIG_SECURE_RLIMIT_NPROC) [Y]
Destroy shared memory segments not in use (CONFIG_SECURE_SHM) [N] **Y**

Kernel hacking
Magic SysRq key (CONFIG_MAGIC_SYSRQ) [N]

Now, return to the "/usr/src/linux/" directory (if you are not already on). You need to compile the new kernel. You do so by using the following command:

[root@deep]# **make dep; make clean; make bzImage**

This line contains three commands in one. The first one, **make dep**, actually takes your configuration and builds the corresponding dependency tree. This process determines what gets compiled and what doesn't. The next step, **make clean**, erase all previous traces of a compilation so as to avoid any mistakes in which version of a feature gets tied into the kernel. Finally, **make bzImage** does the full compilation.

After the process is complete, the kernel is compressed and ready to be installed. Before you can install the new kernel, you need to compile the corresponding modules. This is require only if you're say **Yes** to "Enable loadable module support (CONFIG_MODULES)" and are compiled some options above as a module.

- You do so by using the following command:

  [root@deep]# **make modules**
  [root@deep]# **make modules_install**

NOTE: **make modules** and **make modules_install** commands are require only if you're say **Yes** to "Enable loadable module support (CONFIG_MODULES)" in your kernel configuration above.

Installing the new kernel

1. Copy the file "/usr/src/linux/arch/i386/boot/bzImage" from the kernel source tree to "/boot" directory, and give it an appropriate new name.

   [root@deep]# **cp /usr/src/linux/arch/i386/boot/bzImage /boot/vmlinuz-kernel.version.number**

NOTE: An appropriated or recommended new name is something like **vmlinuz-2.2.14**, this is important if you want a new rescue floppy or emergency boot floppy using the **mkbootdisk**

program that require some specific needs like for example: vmlinuz-2.2.14 instead of vmlinuz-2.2.14.a

2. Copy the file "/usr/src/linux/System.map" from the kernel source tree to "/boot" directory, and give it an appropriate new name.

   [root@deep]# **cp /usr/src/linux/System.map /boot/System.map-kernel.version.number**

3. Cd into the "/boot" directory and rebuild the links **vmlinuz** and **System.map** with the following commands:

   [root@deep]# **cd /boot**
   [root@deep]# **ln -fs vmlinuz-kernel.version.number vmlinuz**
   [root@deep]# **ln -fs System.map-kernel.version.number System.map**

We must rebuild the links of "vmlinuz" and "System.map" to point them to the new kernel version installed. Without the new links LILO program will look by default for the old version of your linux kernel.

4. Remove obsolete files under "/boot" directory to make space:

   [root@deep]# **rm -f module-info**
   [root@deep]# **rm -f initrd-2.2.12-20.img**

The "module-info" link point to the old modules directory of your original kernel. Since we are installed a brand new kernel, we don't need to keep this broken link. The "initrd-2.2.12-20" is a file that contains an initial RAM disk image that serves as a system before the disk are available. This file is only available and is installed from the linux setup installation if you system has a SCSI adapter present. If we use a SCSI system, the driver is now incorporated on the Linux kernel since we have built a monolithic kernel. So we can remove this file safety.

5. Create a new Linux kernel directory that will handle all header files related to Linux kernel for future compilation of other programs on your system.

   Recall, we are created three symlinks under the "/usr/include" directory that point to the linux kernel to be able to compile it without receiving error and also be able to compile future programs. The "/usr/include" directory" is where all header files of your Linux system are keeps for reference and dependencies when you compile and install new programs. The asm, linux, and scsi links are used when program require to know some functions in compile time scpecific to the kernel installed on your system. Other header on the "include" directory are called by programs when they must to know specific information, dependencies, etc of your system.

   [root@deep]# **mkdir -p /usr/src/linux-2.2.14/include**
   [root@deep]# **cp -r /usr/src/linux/include/asm-generic /usr/src/linux-2.2.14/include**
   [root@deep]# **cp -r /usr/src/linux/include/asm-i386 /usr/src/linux-2.2.14/include**
   [root@deep]# **cp -r /usr/src/linux/include/linux /usr/src/linux-2.2.14/include**
   [root@deep]# **cp -r /usr/src/linux/include/net /usr/src/linux-2.2.14/include**
   [root@deep]# **cp -r /usr/src/linux/include/video /usr/src/linux-2.2.14/include**
   [root@deep]# **cp -r /usr/src/linux/include/scsi /usr/src/linux-2.2.14/include**
   [root@deep]# **rm -rf /usr/src/linux**
   [root@deep]# **cd /usr/src**
   [root@deep]# **ln -s /usr/src/linux-2.2.14 linux**

First we create a new directory named “linux-2.2.14” based on the version of the kernel we have installed for easy interpretation, then we copy directories asm-generic, asm-i386, linux, net, video, and scsi from “/usr/linux/include” to our new place “/usr/src/linux-2.2.14/include”. After we remove the entire source directory where we are compiled the new kernel and create a new symbolic link named “linux” under “/usr/src” that point to the “/usr/src/linux-2.2.14/include” directory. With these steps, future compiled programs will know where to look for header related to the kernel on your server.

6. Finally, you need to edit the “**/etc/lilo.conf**” file to make your new kernel one of the boot time options:

Step 1
Edit the **lilo.conf** file (vi /etc/lilo.conf) and make the appropriated change on the line “image=/boot/”.

```
[root@deep]# vi /etc/lilo.conf

For example:
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
prompt
timeout=00
restricted
password=somepasswd
image=/boot/vmlinuz-kernel.version.number #(add your new kernel name file here).
        label=linux
        root=/dev/sda6
        read-only
```

NOTE: Don’t forget to remove the line that read “initrd=/boot/initrd-2.2.12-20.img” in the “lilo.conf” file, since this line is not necessary now (monolithic kernel don’t need an initrd file).

Step 2
Now, we update our “lilo.conf” file with the new kernel.

```
[root@deep]# /sbin/lilo -v
LILO version 21, [Copyright 1992-1998 Werner Almesberger

Reading boot sector from /dev/sda
Merging with /boot/boot.b
Boot image: /boot/vmlinuz-2.2.14
Added linux *
/boot/boot.0800 exits – no backup copy made.
Writing boot sector.
```

IMPORTANT NOTE: If you are say **NO** to the option “Unix98 PTY support (CONFIG_UNIX98_PTYS)” in your kernel configuration above, edit your “/etc/fstab” file and remove the line that read:

```
none            /dev/pts        devpts          gid=5,mode=620 0 0
```

Delete program, file and lines related to modules

By default when you install Red Hat Linux for the first time (like we do), the kernel is building as a modularized. This mean that each devices or functions we need are made as a modules and are controlled by Kernel Daemon program named **Kerneld** (now **kmod** with the 2.2x kernel version),

which automatically loads some modules and functions support into memory as it is needed, and uploads it when it's no longer being used.

Kerneld use the **conf.modules** file located in "/etc/conf.modules" to know for example which Ethernet card you have, and if your Ethernet card requires special configuration. Since we are not using any modules in our new compiled kernel, we can remove the "conf.modules" file and uninstall the "modutils" program.

The modutils packages includes the kerneld program for automatic loading of modules under 2.0 kernels and unloading of modules under 2.0 and 2.2 kernels, as well as other module management programs. Examples of loaded and unloaded modules are device drivers and filesystems, as well as some other things.

- To remove the "conf.modules" file, use the command:
  [root@deep]# **rm -f /etc/conf.modules**

- To uninstall the modutils package, use the following command:
  [root@deep]# **rpm -e --nodeps modutils**

One last thing to do is to edit the file **"rc.sysinit"** and comment out all the lines related to **"depmod -a"** by inserting a "**#**" at the beginning of the lines. This is needed since at boot time the system read the rc.sysinit script to find module dependencies in the kernel by default.

Comment out the line 260 in the **rc.sysinit** file (vi +260 /etc/rc.d/rc.sysinit):

```
if [ -x /sbin/depmod -a -n "$USEMODULES" ]; then
To read:
#if [ -x /sbin/depmod -a -n "$USEMODULES" ]; then
```

Comment out the lines 272 to 277 in the **rc.sysinit** file (vi +272 /etc/rc.d/rc.sysinit):

```
   if [ -L /lib/modules/default ]; then
      INITLOG_ARGS= action "Finding module dependencies" depmod -a default
   else
      INITLOG_ARGS= action "Finding module dependencies" depmod -a
   fi
fi
To read:
#    if [ -L /lib/modules/default ]; then
#       INITLOG_ARGS= action "Finding module dependencies" depmod -a default
#    else
#       INITLOG_ARGS= action "Finding module dependencies" depmod -a
#    fi
#fi
```

NOTE: Once again, all this section "Delete program, file and lines related to modules" is require only if you're say **No** to "Enable loadable module support (CONFIG_MODULES)" in your kernel configuration above. The procedure describe above relate to initscripts-4_70-1 package.

Now you must **Reboot** your system and then test your results.
[root@deep]# **reboot**

When the system is rebooted and you are log in, verify the new version of your kernel with the following command:

- To verify the version of your new kernel, use the following command:
  [root@deep]# **uname -a**
  Linux deep.openarch.com 2.2.14 #1 Mon Jan 10 10:40:35 EDT 2000 i686 unknown
  [root@deep]#

Congratulation.

Making a new rescue floppy

If all has gone well, you should have a system with an upgraded kernel. You should now make a new rescue image with this kernel in the case of future emergencies. You should follow the previous instructions, just changing the arguments to **mkbootdisk** to cover the new version of the kernel.

Login as root, and insert a new floppy.

[root@deep]# **mkbootdisk --device /dev/fd0 2.2.14**
Insert a disk in /dev/fd0. Any information on the disk will be lost.
Press <Enter> to continue or ^C to abort:

Important note: The mkbootdisk program run only on modularized kernel. You can't use it on a monolithic kernel, instead use your emergency boot floppy if you have a problem with your system in the future.

What is Rescue Mode

Rescue mode is a term used to describe a method of booting a small Linux environment completely from diskettes. By using rescue mode, it's possible to access the files stored on your system's hard drive, even if you can't actually run Linux from that hard drive.

Making a emergency boot floppy disk

Since it is possible to create only a rescue floppy on modularized kernel, we must find another way to boot our Linux system if the Linux kernel on the hard disk is damaged. After you successfully start Linux and log in as root, you should immediately create a Linux emergency boot floppy disk.

- To create the emergency boot floppy disk, follow these steps:

  1. Insert a floppy disk and format it with the following command:
     [root@deep]# **fdformat /dev/fd0H1440**
     Double-sided, 80 tracks, 18 sec/track. Total capacity 1440 kB.
     Formatting ... done
     Verifying ... done

  2. Copy the file "vmlinuz" from the "/boot" directory to the floppy disk:
     [root@deep]# **cd /boot**
     [root@deep]# **cp vmlinuz /dev/fd0**
     cp: overwrite `/dev/fd0'? **y**

     The "vmlinuz" file happens to be the Linux kernel.

  3. Determine the kernel's root device with the following command:
     [root@deep]# **rdev**
     /dev/sda12 /

     The kernel's root device is the disk partition where the root file system is located. In this case, the root device is "dev/sda12"; the device name may be different on your system.

4. Set the kernel's root device with the following command:
   [root@deep]# **rdev /dev/fd0 /dev/sda12**

To set the kernel's root device, use the device reported by the "rdev" command in the previous step.

5. Mark the root device as read-only with the following command:
   [root@deep]# **rdev -R /dev/fd0 1**

This causes Linux initially to mount the root file system as read-only. By setting the root device as read-only, you avoid several warning and error messages.

6. Now put the boot floppy in the A drive and reboot your system with the following command:
   [root@deep]# **reboot**

Update your "/dev" entries

If you've done a major kernel upgrade, or have added new devices to your system, you're sure to be well of if you update your "/dev" entries. You can do that with the command (as root) **./MAKEDEV update**. You can also use MAKEDEV to create a standard batch of /dev entries or separate ones. See man MAKEDEV.

[root@deep]# **cd /dev**
[root@deep]# **./MAKEDEV update**

NOTE: MAKEDEV is a script that utilizes mknod.

Part IV Networking-Related Reference

In this Part

TCP/IP Network Management
Networking Firewall

Chapter 6 TCP/IP Network Management

In this Chapter

Install more than one Ethernet Card per machine
Files related to networking functionality
Configuring TCP/IP networking manually with the command line
TCP/IP security problem overview

TCP/IP Network Management

Overview

Network management covers a wide variety of topics. In general it includes gathering statistical data and status information about parts of your network, and taking action as necessary to deal with failures and other changes. The most primitive technique for network monitoring is periodic "pinging" of critical hosts. More sophisticated network monitoring requires the ability to get specific status and statistical information from various devices on the network. These should include various sorts of datagram counts, as well as counts of errors of various kinds.

In this chapter we will try to answer fundamental questions about networking devices, files related to networking functionality, essential networking commands and TCP/IP security overview.

Install more than one Ethernet Card per Machine

You might use Linux as a gateway between two Ethernet networks. In that case, you might have two Ethernet cards on your server. To eliminate problems at boot time, the Linux kernel doesn't detect multiple cards automatically. If you happen to have two or more cards, you should specify the parameters of the cards on the "**lilo.conf"** file for a monolithic kernel or on the "**conf.modules"** file for a modularized kernel.

If the driver(s) is/are being used as a loadable module (modularized kernel)

In the case of PCI drivers, the module will typically detect all of the installed cards automatically. For ISA cards, you need to supply the I/O base address of the card so the module knows where to look. This information is stored in the file "/etc/conf.modules".

As an example, consider we have two ISA 3c509 cards, one at I/O 0x300 and one at I/O 0x320.

For ISA cards, edit the **conf.modules** file (vi /etc/conf.modules) and add:

```
alias eth0 3c509
alias eth1 3c509
options 3c509 io=0x300,0x320
```

This says that the 3c509 driver should be loaded for either eth0 or eth1 (alias eth0, eth1) and it should be loaded with the options io=0x300,0x320 so that the drivers knows where to look for the cards. Note that 0x is important – things like 300h as commonly used in the DOS world won't work.

For PCI cards, you typically only need the alias lines to correlate the ethN interfaces with the appropriate driver name, since the I/O base of a PCI card can be safety detected.

For PCI cards, edit the **conf.modules** file (vi /etc/conf.modules) and add:

```
alias eth0 3c509
alias eth1 3c509
```

If the drivers(s) is/are compiled into the kernel (monolithic kernel)

PCI probes will find all related cards automatically. ISA cards will also find all related cards automatically, but in some circumstance ISA cards still need to do the following. This information

is stored in the file "/etc/lilo.conf". The method is to pass boot-time arguments to the kernel, which is usually done by **LILO**.

For ISA cards, edit the **lilo.conf** file (vi /etc/lilo.conf) and add:

append="ether=0,0,eth1"

NOTE: First test your ISA cards without the boot-time arguments in the "lilo.conf" file and if this fail use the boot-time arguments.

In this case eth0 and eth1 will be assigned in the order that the cards are found at boot. Since we have recompiled the kernel, we must use the second method (If the drivers(s) is/are compiled into the kernel) to install our second Ethernet card on the system. Remember this is require only in some circumstance for ISA cards, PCI cards will be find automatically.

Files related to networking functionality

In Linux, the TCP/IP network is configured through several text files you may have to edit in to make networking work. It's very important to know the configurations files related to TCP/IP networking, so that you can edit and configure the files if necessary. Remember that our server doesn't have a Xwindow interface to configure files via graphical interface. The following sections describe the basic TCP/IP configuration files.

The "/etc/HOSTNAME" file

This file stores your system's host name—your system's fully qualified domain name (FQDN), such as deep.openarch.com.

Following is a sample "/etc/HOSTNAME" file:

```
deep.openarch.com
```

The "/etc/sysconfig/network-scripts/ifcfg-ethN" files

Files configurations for each network device you may have or want to add on your system are located in the "/etc/sysconfig/network-scripts/" directory with Red Hat Linux 6.1 and are named **ifcfg-eth0** for the first interface and **ifcfg-eth1** for the second etc.

Following is a sample "/etc/sysconfig/network-scripts/ifcfg-eth0" file:

```
DEVICE=eth0
IPADDR=208.164.186.1
NETMASK=255.255.255.0
NETWORK=208.164.186.0
BROADCAST=208.164.186.255
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
```

If you want to modify your network address manually or add new network on new interface, edit this file (ifcfg-ethN) or create a new one and make the appropriated changes.

DEVICE=***name***, where ***name*** is the name of the physical device.

IPADDR=***addr***, where ***addr*** is the IP address.

NETMASK=***mask***, where ***mask*** if the netmask value.

NETWORK=***addr***, where ***addr*** is the network address.

BROADCAST=***addr***, where ***addr*** is the broadcast address.

ONBOOT=***answer***, where ***answer*** is yes or no (active or inactive at boot time).

BOOTPROTO=***proto***, where ***proto*** is one of the following:

- none No boot-time protocol should be used.
- bootp The bootp protocol should be used.
- dhcp The dhcp protocol should be used.

USERCTL=***answer***, where ***answer*** is one of the following:

- yes Non-root users are allowed to control this device.
- no Non-root users are not allowed to control this device.

The "/etc/resolv.conf" file

This file is another text file used by the resolver—a library that determines the IP address for a host name.

Following is a sample "/etc/resolv.conf" file:

```
search openarch.com
nameserver 208.164.186.1
nameserver 208.164.186.2
```

NOTE: Name servers are queried in the order they appear in the file.

The "/etc/host.conf" file

This file specifies how names are resolved. Linux uses a resolver library to obtain the IP address corresponding to a host name.

Following is a sample "/etc/host.conf" file:

```
# Lookup names via DNS first then fall back to /etc/hosts.
order bind,hosts
# We have machines with multiple addresses.
multi on
# Check for IP address spoofing.
nospoof on
```

The **order** option indicate the order of services. The sample entry specifies that the resolver library should first consult the name server to resolve a name and then check the "/etc/hosts" file.

The **multi** option determines whether a host in the "/etc/hosts" file can have multiple IP addresses (multiple interface ethN). Hosts that have more than one IP address are said to be *multiomed*, because the presence of multiple IP addresses implies that host has several network interfaces.

The **nospoof** option indicate to take care of not permit spoof on this machine. IP-Spoofing is a security exploit that works by tricking computers in a trust relationship that you are someone that you really aren't.

The "/etc/sysconfig/network" file

The "/etc/sysconfig/network" file is used to specify information about the desired network configuration on your server.

Following is a sample "/etc/sysconfig/network" file:

```
NETWORKING=yes
FORWARD_IPV4=yes
HOSTNAME=deep.openarch.com
GATEWAY=0.0.0.0
GATEWAYDEV=
```

The following values may be used:

NETWORKING=***answer***, where ***answer*** is yes or no (configured or not configured).

FORWARD_IPV4=***answer***, where ***answer*** is yes or no (perform IP forwarding not perform IP forwarding).

HOSTNAME=***hostname***, where ***hostname*** is the hostname of your server.

GATEWAY=***gw-ip***, where ***gw-ip*** is the IP address of the network's gateway.

GATEWAYDEV=***gw-dev***, where ***gw-dev*** is the gateway device like eth0.

NOTE: For compatibility with older software, the */etc/HOSTNAME* file should contain the same value as HOSTNAME= ***hostname*** above.

The "/etc/hosts" file

As your machine gets started, it will need to know the mapping of some hostnames to IP addresses before DNS can be referenced. This mapping is kept in the "/etc/hosts" file. In the absence of a name server, any network program on your system consults this file to determine the IP address that corresponds to a host name.

Following is a sample "/etc/hosts" file:

IP Address	Hostname	Alias
127.0.0.1	localhost	gate.openarch.com
208.164.186.1	gate.openarch.com	gate
208.164.186.2	forest.openarch.com	forest
208.164.186.3	deep.openarch.com	deep

The leftmost column is the IP address to be resolved. The next column is that host's name. Any subsequent columns are alias for that host. In the second line, for example, the address 208.164.186.1 if for the host gate.openarch.com. Another name for gate.openarch.com is gate.

After you are finish to configure your networking files, don't forget to restart the network for the change to take effect.

- To restart your network, use the command:
  [root@deep]# **/etc/rc.d/init.d/network restart**

IMPORTANT NOTE: The **tcpd** program is responsible to monitored incoming requests for **telnet**, **ftp** and other services that have a one-to-one mapping onto executable files.

Operation is as follows: whenever a request for service arrives, the inetd daemon is tricked into running the tcpd program instead of the desired server. tcpd logs the request and does some additional checks. When all is well, tcpd runs the appropriate server program and goes away.

tcpd verifies the client host name that is returned by the address->name DNS server by looking at the host name and address that are returned by the name->address DNS server. If any discrepancy is detected, tcpd concludes that it is dealing with a host that pretends to have someone else's host name.

Optionally, tcpd disables source-routing socket options on every connection that it deals with. This will take care of most attacks from hosts that pretend to have an address that belongs to someone else's network. UDP services do not benefit from this protection. Time out problems are often caused by the server trying to resolve the client IP address to a DNS name. Either DNS isn't configured properly on your server or the client machines aren't known to DNS.

If you are intended to run **telnet** or **ftp** services on your server, and aren't using DNS, don't forget to add client machine name and IP in your "**/etc/hosts"** file on the server or you can expect to wait several minutes for the DNS lookup to time out, before you get a login: prompt.

Configuring TCP/IP Networking manually with the command line

Ifconfig is the tool used to set up and configure your network card. You should understand this command in the event you need to configure the network by hand. An important note to take care is when using **ifconfig** to configure your network devices, the settings will not survive a reboot.

- To assigns the eth0 interface the IP-address of 208.164.186.2 use the command:
  [root@deep]# **ifconfig eth0 208.164.186.2 netmask 255.255.255.0**

- To display all interfaces you may have, use the command:
  [root@deep]# **ifconfig**

  The output should look something like this:

```
eth0    Link encap:Ethernet  HWaddr 00:E0:18:90:1B:56
        inet addr:208.164.186.2  Bcast:208.164.186.255  Mask:255.255.255.0
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:1295 errors:0 dropped:0 overruns:0 frame:0
        TX packets:1163 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:100
        Interrupt:11 Base address:0xa800

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        UP LOOPBACK RUNNING  MTU:3924  Metric:1
        RX packets:139 errors:0 dropped:0 overruns:0 frame:0
        TX packets:139 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
```

If ifconfig is invoked without any parameters, it displays all interfaces you configured. An option of -a show the inactive one as well.

- To display all interfaces as well as inactive interfaces you may have, use the command:
  [root@deep]# **ifconfig -a**

The output should look something like this:

```
eth0      Link encap:Ethernet  HWaddr 00:E0:18:90:1B:56
          inet addr:208.164.186.2  Bcast:208.164.186.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1295 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1163 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:11 Base address:0xa800

eth1      Link encap:Ethernet  HWaddr 00:E0:18:90:1B:56
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1295 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1163 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:5 Base address:0xa320

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:139 errors:0 dropped:0 overruns:0 frame:0
          TX packets:139 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

NOTE: When using **ifconfig** to configure your network devices, the settings will not survive a reboot.

- To assign the default gateway for 208.164.186.1 use the command:
  [root@deep]# **route add default gw 208.164.186.1**

In this example, the default route is set up to go to 208.164.186.1, your router.

Verify that you can reach your hosts. Choose a host from your network, for instance 208.164.186.1.

- To verify that you can reach your hosts, use the command:
  [root@deep]# **ping 208.164.186.1**

  The output should look something like this:

```
[root@deep networking]# ping 208.164.186.1
PING 208.164.186.1 (208.164.186.1) from 208.164.186.2 : 56 data bytes
64 bytes from 208.164.186.2: icmp_seq=0 ttl=128 time=1.0 ms
64 bytes from 208.164.186.2: icmp_seq=1 ttl=128 time=1.0 ms
64 bytes from 208.164.186.2: icmp_seq=2 ttl=128 time=1.0 ms
64 bytes from 208.164.186.2: icmp_seq=3 ttl=128 time=1.0 ms

--- 208.164.186.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.0/1.0/1.0 ms
```

You should now display the routing information with the command **route** to see if both hosts have the correct routing entry:

- To display the routing information, use the command:
  [root@deep]# **route -n**

The output should look something like this:

Kernel IP routing table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
208.164.186.2	0.0.0.0	255.255.255.255	UH	0	0	0	eth0
208.164.186.0	208.164.186.2	255.255.255.0	UG	0	0	0	eth0
208.164.186.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0
127.0.0.0	0.0.0.0	255.0.0.0	U	0	0	0	lo

- To check the status of the interfaces quickly, use the netstat -i command, as follows:
  [root@deep]# **netstat -i**

The output should look something like this:

Kernel Interface table

Iface	MTU	Met	RX-OK	RX-ERR	RX-DRP	RX-OVR	TX-OK	TX-ERR	TX-DRP	TX-OVR	Flg
eth0	1500	0	4236	0	0	0	3700	0	0	0	BRU
lo	3924	0	13300	0	0	0	13300	0	0	0	LRU
ppp0	1500	0	14	1	0	0	16	0	0	0	PRU

Another useful netstat option is -t, which shows all active TCP connections. Following is a typical result of netstat -t:

- To shows all active TCP connections, use the command:
  [root@deep]# **netstat -t**

The output should look something like this:

Active Internet connections (w/o servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
Tcp	0	0	deep.openar:netbios-ssn	gate.openarch.com:1045	ESTABLISHED
Tcp	0	0	localhost:1032	localhost:1033	ESTABLISHED
Tcp	0	0	localhost:1033	localhost:1032	ESTABLISHED
Tcp	0	0	localhost:1030	localhost:1034	ESTABLISHED
Tcp	0	0	localhost:1031	localhost:1030	ESTABLISHED
Tcp	0	0	localhost:1028	localhost:1029	ESTABLISHED
Tcp	0	0	localhost:1029	localhost:1028	ESTABLISHED
Tcp	0	0	localhost:1026	localhost:1027	ESTABLISHED
Tcp	0	0	localhost:1027	localhost:1026	ESTABLISHED
Tcp	0	0	localhost:1024	localhost:1025	ESTABLISHED
Tcp	0	0	localhost:1025	localhost:1024	ESTABLISHED

- To shows all active and listen TCP connections, use the command:
  [root@deep]# **netstat -vat**

The output should look something like this:

Active Internet connections (servers and established)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	deep.openarch.co:domain	*:*	LISTEN
tcp	0	0	localhost:domain	*:*	LISTEN
tcp	0	0	deep.openarch.com:ssh	gate.openarch.com:1682	ESTABLISHED
tcp	0	0	*:webcache	*:*	LISTEN
tcp	0	0	deep.openar:netbios-ssn	*:*	LISTEN
tcp	0	0	localhost:netbios-ssn	*:*	LISTEN
tcp	0	0	localhost:1032	localhost:1033	ESTABLISHED
tcp	0	0	localhost:1033	localhost:1032	ESTABLISHED
tcp	0	0	localhost:1030	localhost:1031	ESTABLISHED
tcp	0	0	localhost:1031	localhost:1030	ESTABLISHED
tcp	0	0	localhost:1028	localhost:1029	ESTABLISHED
tcp	0	0	localhost:1029	localhost:1028	ESTABLISHED
tcp	0	0	localhost:1026	localhost:1027	ESTABLISHED

```
tcp        0      0 localhost:1027          localhost:1026          ESTABLISHED
tcp        0      0 localhost:1024          localhost:1025          ESTABLISHED
tcp        0      0 localhost:1025          localhost:1024          ESTABLISHED
tcp        0      0 deep.openarch.com:www   *:*                     LISTEN
tcp        0      0 deep.openarch.com:https *:*                     LISTEN
tcp        0      0 *:389                   *:*                     LISTEN
tcp        0      0 *:ssh                   *:*                     LISTEN
```

- To stop all networks devices manually on your system, use the command:
  [root@deep]# **/etc/rc.d/init.d/network stop**

- To start all networks devices manually on your system, use the command:
  [root@deep]# **/etc/rc.d/init.d/network start**

TCP/IP security problem overview

It is assumed that the reader is familiar with the basic operation of the TCP/IP protocol suite, which includes IP and TCP header field functions and initial connection negotiation. For the uninitiated, a brief description of TCP/IP connection negotiation is given below. The user is strongly encouraged however to research other published literature on the subject.

IP Packets

The term packet refers to an Internet Protocol (IP) network message. It's the name given to a single, discrete message or piece of information that is sent across an Ethernet network. Structurally, a packet contains an information header and a message body containing the data being transferred. The body of the IP packet- it's data- is all or a piece (a fragment) of a higher-level protocol message.

The IP mechanism

Linux supports three IP message types: ICMP, UDP, and TCP. An ICMP (Internet Control Message Protocol) packet is a network-level, IP control and status message. ICMP messages contain information about the communication between the two end-point computers. A UDP (User Datagram Protocol) IP packet carries data between two network-based programs, without any guarantees regarding successful delivery or packet delivery ordering. Sending a UDP packet is akin to sending a postcard to another program. A TCP (Transmission Control Protocol) IP packet carries data between two network-based programs, as well, but the packet header contains additional state information for maintaining an ongoing, reliable connection. Sending a TCP packet is akin to carrying on a phone conversation with another process. Most Internet network services use the TCP communication protocol rather than the UDP communication protocol. In other words, most Internet services are based on the idea of an ongoing connection with two-way communication between a client program and a server program.

IP packet headers

All IP packet headers contain the source and destination IP addresses and the type of IP protocol message (ICMP, UDP or TCP) this packet contains. Beyond this, a packet header contains slightly different fields depending on the protocol type. ICMP packets contain a type field identifying the control or status message, along with a second code field for defining the message more specifically. UDP and TCP packets contain source and destination service port numbers. TCP packets contain additional information about the state of the connection and unique identifiers for each packet.

TCP/IP Security Problem
The TCP/IP protocol suite has a number of weaknesses that allow an attacker to leverage techniques in the form of covert channels to surreptitiously pass data in otherwise benign packets. This section attempts to illustrate these weaknesses in theoretical examples.

Application
A covert channel is described as: "any communication channel that can be exploited by a process to transfer information in a manner that violates the systems security policy. Essentially, it is a method of communication that is not part of an actual computer system design, but can be used to transfer information to users or system processes that normally would not be allowed access to the information.

In the case of TCP/IP, there are a number of methods available whereby covert channels can be established and data can be surreptitiously passed between hosts. These methods can be used in a variety of areas such as the following:

- Bypassing packet filters, network sniffers, and "dirty word" search engines.
- Encapsulating encrypted or non-encrypted information within otherwise normal packets of information for secret transmission through networks that prohibit such activity "TCP/IP Steganography".
- Concealing locations of transmitted data by "bouncing" forged packets with encapsulated information off innocuous Internet sites.

It is important to realize that TCP is a "connection oriented" or "reliable" protocol. Simply put, TCP has certain features that ensure data arrives at the remote host in a usually intact manner. The basic operation of this relies in the initial TCP "three way handshake" which is described in the three steps below.

Step 1
Send a synchronize (SYN) packet and Initial Sequence Number (ISN)

Host A wishes to establish a connection to Host B. Host A sends a solitary packet to Host B with the synchronize bit (SYN) set announcing the new connection and an Initial Sequence Number (ISN) which will allow tracking of packets sent between hosts:

Host A ------ SYN(ISN) ------> Host B

Step 2
Allow remote host to respond with an acknowledgment (ACK)

Host B responds to the request by sending a packet with the synchronize bit set (SYN) and ACK (acknowledgment) bit set in the packet back to the calling host. This packet contains not only the responding clients' own sequence number, but the Initial Sequence Number plus one (ISN+1) to indicate the remote packet was correctly received as part of the acknowledgment and is awaiting the next transmission:

Host A <------ SYN(ISN+1)/ACK ------ Host B

Step 3
Complete the negotiation by sending a final acknowledgment to the remote host.

At this point Host A sends back a final ACK packet and sequence number to indicate successful reception and the connection is complete and data can now flow:

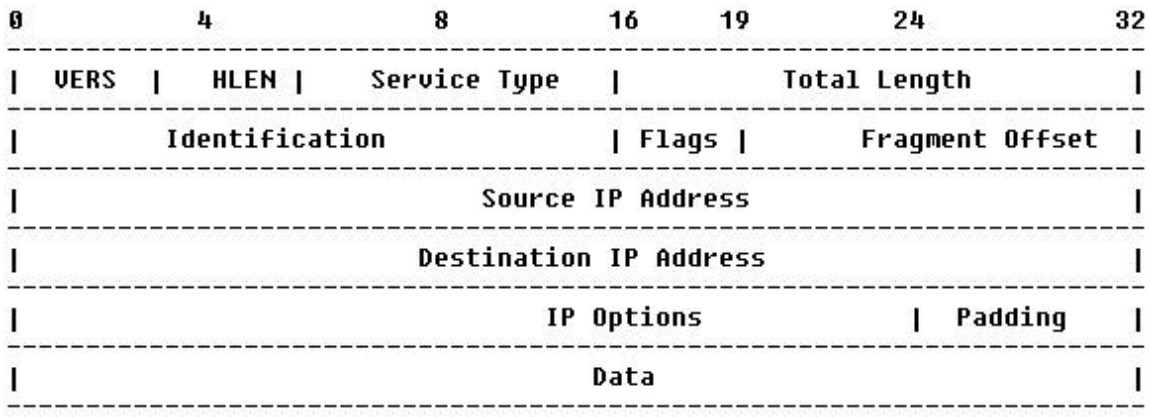
Host A ------ ACK ------> Host B

The entire connection process happens in a matter of milliseconds and both sides independently acknowledge each packet from this point. This handshake method ensures a "reliable" connection between hosts and is why TCP is considered a "connection oriented" protocol. It should be noted that only TCP packets exhibit this negotiation process. This is not so with UDP packets which are considered "unreliable" and do not attempt to correct errors nor negotiate a connection before sending to a remote host.

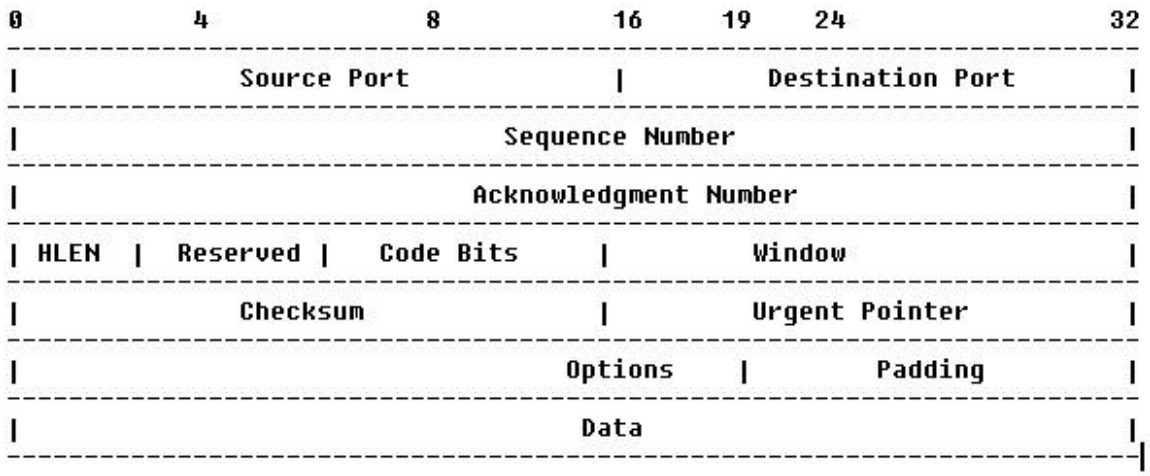
Encoding Information in a TCP/IP Header

The TCP/IP header contains a number of areas where information can be stored and sent to a remote host in a covert manner. Take the following diagrams, which are textual representations of the IP and TCP headers respectively:

IP Header (Numbers represent bits of data from 0 to 32 and the relative position of the fields in the datagram)



TCP Header (Numbers represent bits of data from 0 to 32 and the relative position of the fields in the datagram)



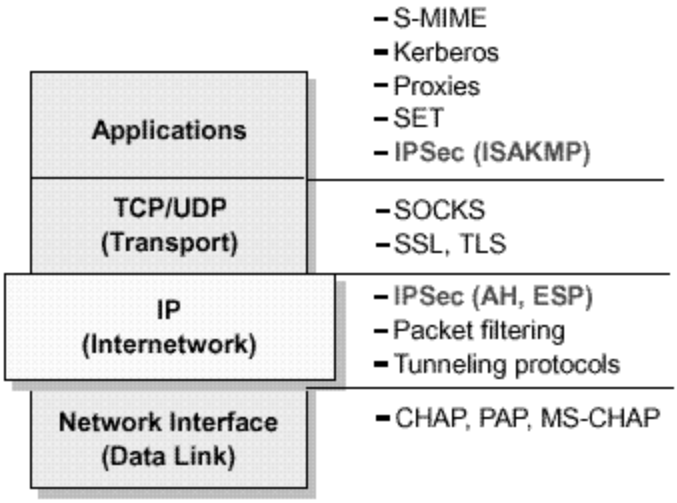
Within each header there are multitudes of areas that are not used for normal transmission or are "optional" fields to be set as needed by the sender of the datagrams. An analysis of the areas of a typical IP header that are either unused or optional reveals many possibilities where data can be stored and transmitted. The basis of the exploitation relies in encoding ASCII values of the range 0-255. Using this method it is possible to pass data between hosts in packets that appear to be initial connection requests, established data streams, or other intermediate steps. These packets can contain no actual data, or can contain data designed to look innocent. These packets can also contain forged source and destination IP addresses as well as forged source and destination ports. This can be useful for tunneling information past some types of packet filters. Additionally, forged packets can be used to initiate an anonymous TCP/IP "bounced packet network" whereby packets between systems can be relayed off legitimate sites to thwart tracking by sniffers and other network monitoring devices.

Implementations of Security Solutions

The following protocols and systems are commonly used to solve and provide various degrees of security services in a computer network.

- IP filtering
- Network Address Translation (NAT)
- IP Security Architecture (IPSec)
- SOCKS
- Secure Sockets Layer (SSL)
- Application proxies
- Firewalls
- Kerberos and other authentication systems (AAA servers)
- Secure Electronic Transactions (SET)

This Graph illustrates where those security solutions fit within the TCP/IP layers:



Security Solutions in the TCP/IP Layers

Summary

By reading this chapter, you learn the following:

- You may have to edit one or more of the following files to configure networking on your Linux system: “/etc/hosts”, “/etc/host.conf”, “/etc/resolv.conf”, “/etc/HOSTNAME”, “/etc/sysconfig/network”, and the scripts in the “/etc/sysconfig/network-scripts” directory.

Chapter 7 Networking Firewall

In this Chapter

Linux IPCHAINS

Overview

Someone can tell why I might want something like a commercial firewall product rather then just using IPchains and restricting certain packets and stuff? What am I losing by using IPchains? Security? Logging?

Now, there is undoubtedly room for debate on this, but IMHO, IPchains is as good and, most of the time better, than commercial firewall packages from a functionality and "support" standpoint. You will probably have more insight into what's going on in your network using IPchains than a commercial solution. That being said, A LOT of corporate types want to tell their shareholders, CEO/CTO/etc. that they have the backing of reputable security Software Company. The firewall could be doing nothing more than passing through all traffic and still the corporate type would be more comfortable than having to rely on the geeky guy in the corner cube who gets grumpy if you turn the light on before noon.

In the end, a lot of companies want to be able to turn around and demand some sort of restitution from a vendor if the network is breached, whether or not they'd actually get anything or even try. All they can typically do with an open source solution is fire the guy that implemented it. At least some of the commercial firewalls are based on Linux or something similar. IF quite probable that IPchains is secure enough for you but not those engaging in serious amounts of high stakes bond trading. (Doing a cost/benefit analysis and asking a lot of pertinent questions is recommended before spending serious money on a $$$$ firewall---otherwise you may end up with something inferior to your IPchains). Quite a few of the NT firewalls are likely to be no better than IPchains and the general consensus on bugtraq and NT bugtraq are that NT is a *far too insecure* a serious firewall.

What is a Network Firewall Security Policy?

An organization's overall security policy must be determined according to security analysis and business needs analysis. Since a firewall relates to network security only, a firewall has little value unless the overall security policy is properly defined. Network firewall security policy defines those services that will be explicitly allowed or denied, how these services will be used and the exceptions to these rules. Every rule in the network firewall security policy should be implemented on a firewall. Generally, a firewall uses one of the following methods.

Everything not specifically permitted is denied

This approach blocks all traffic between two networks except for those services and applications that are permitted. Therefore, each desired service and application should be implemented one by one. No service or application that might be a potential hole on the firewall should be permitted. This is the most secure method, denying services and applications unless explicitly allowed by the administrator. On the other hand, from the point of users, it might be more restrictive and less convenient. This is the method we will use in our Firewall configuration files on this book.

Everything not specifically denied is permitted

This approach allows all traffic between two networks except for those services and applications that are denied. Therefore, each untrusted or potentially harmful service or application should be denied one by one. Although this is a flexible and convenient method for the users, it could potentially cause some serious security problems.

What is Packet Filtering?

Packet Filtering is the type of firewall built into the Linux kernel. A filtering firewall works at the network level. Data is only allowed to leave the system if the firewall rules allow it. As packets arrive they are filtered by their type, source address, destination address, and port information contained in each packet.

Most of the time, packet-filtering is accomplished by using a router that can forward packets according to filtering rules. When a packet arrives at the packet-filtering router, the router extracts certain information from the packet header and makes decisions according to the filter rules as to whether the packet will pass through or be discarded.

The following information can be extracted from the packet header:

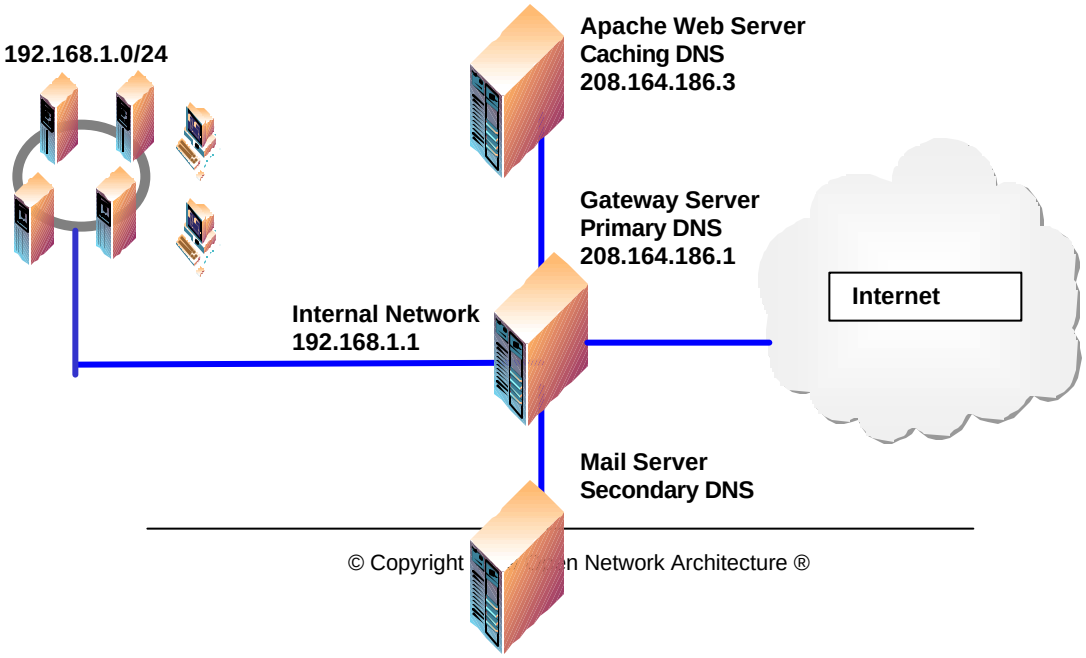
- Source IP address
- Destination IP address
- TCP/UDP source port
- TCP/UDP destination port
- ICMP message type
- Encapsulated protocol information (TCP, UDP, ICMP or IP tunnel)

Because very little data is analyzed and logged filtering firewalls take less CPU and create less latency in your network. There are lots of ways to structure your network to protect your systems using a firewall.

The topology

All servers machines should be configured to block unused ports at least even if there are not a firewall server. This is require for more security. Imagine someone gain access to your firewall gateway server, if your neighborhoods servers are not configured to block unused ports so let get the party. The same is true for local connect. Unauthorized employees can gain access from the inside to your other servers.

In our configuration we will give you tree different examples that can help you to configure your firewall rules depending of the type of the server you want to protect and the emplacement of these servers on your network architecture. The first example firewall rules file will be for a Web Server, the second for a Mail Server and the last for a Gateway Server that act as proxy for the inside Wins, Workstations and Servers machines. See the graph show bellow to get an idea.



208.164.186.2

www.openarch.com **Caching Only DNS** **208.164.186.3**	**deep.openarch.com** **Master DNS Server** **208.164.186.1**	**mail.openarch.com** **Slave DNS Server** **208.164.186.2**
1. Unlimited traffic on the loopback interface allowed 2. ICMP traffic allowed 3. DNS Caching and Client Server on port 53 allowed 4. SSH Server on port 22 allowed 5. HTTP Server on port 80 allowed 6. HTTPS Server on port 443 allowed 7. SMTP Client on port 25 allowed 8. FTP Server on ports 20, 21 allowed 9. Outgoing traceroute request allowed	1. Unlimited traffic on the loopback interface allowed 2. ICMP traffic allowed 3. DNS Server and Client on port 53 allowed 4. SSH Server and Client on port 22 allowed 5. HTTP Server and Client on port 80 allowed 6. HTTPS Server and Client on port 443 allowed 7. WWW-CACHE Client on port 8080 allowed 8. External POP Client on port 110 allowed 9. External NNTP NEWS Client on port 119 allowed 10. SMTP Server and Client on port 25 allowed 11. IMAP Server on port 143 allowed 12. IRC Client on port 6667 allowed 13. ICQ Client on port 4000 allowed 14. FTP Client on port 20, 21 allowed 15. RealAudio / QuickTime Client allowed 16. Outgoing traceroute request allowed	1. Unlimited traffic on the loopback interface allowed 2. ICMP traffic allowed 3. DNS Server and Client on port 53 allowed 4. SSH Server on port 22 allowed 5. SMTP Server and Client on port 25 allowed 6. IMAP Server on port 143 allowed 7. Outgoing traceroute request allowed

The table above shows you the ports I enable on the different servers by default on my firewall scripts file in this book. Depending of what services must be available in the server for the outside, you must configure your firewall script file to allow the traffic on the specified ports. **www.openarch.com** is our Web Server, **mail.openarch.com** is our Mail Hub Server for all the internal network, and **deep.openarch.com** is our Gateway Server for all the example explained in this chapter.

Build a kernel with IPCHAINS Firewall support

The first thing you need to do is ensure that your kernel has been built with Network Firewall support enabled and Firewalling. Remember, all servers machines should be configured to block unused ports at least even if there are not a firewall server. In the 2.2.14 kernel version you need ensure that you have answered **Y** to the following questions:

Networking options:
Network firewalls (CONFIG_FIREFALL) [N] **Y**
IP:Firewalling (CONFIG_IP_FIREWALL) [N] **Y**
IP:TCP syncookie support (CONFIG_SYN_COOKIES) [N] **Y**

NOTE: If you are follow the Linux Kernel section and are recompiled your kernel, the options "Network firewalls, IP:Firewalling, and IP:TCP syncookie support" show above are already set.

IP Masquerading and IP ICMP Masquerading are require only for a Gateway Server.
IP:Masquerading (CONFIG_IP_MASQUERADE) [N] **Y**
IP:ICMP Masquerading (CONFIG_IP_MASQUERADE_ICMP) [N] **Y**

NOTE: Only your **Gateway Server** need to have “IP:Masquerading” and “IP:ICMP Masquerading” kernel option enable. This is require to masquerade your Internal Network for the outside.

Masquerade means that if one of the computers on your local network for which your Linux box acts as a firewall wants to send something to the outside, your box can "masquerade" as that computer, i.e. it forwards the traffic to the intended outside destination, but makes it look like it came from the firewall box itself.

It works both ways: if the outside host replies, the Linux firewall will silently forward the traffic to the corresponding local computer. This way, the computers on your local net are completely invisible to the outside world, even though they can reach the outside and can receive replies. This makes it possible to have the computers on the local network participate on the Internet even if they don't have officially registered IP addresses.

The IP masquerading code will only work if IP forwarding is enabled executing a line like:

echo "1" > /proc/sys/net/ipv4/ip_forward

from a boot time script after the “/proc” filesystem has been mounted. You can add this line in your “/etc/rc.d/rc.local” file so IP forwarding is enable automatically for you even if your server is rebooted.

Edit the **rc.local** file (vi /etc/rc.d/rc.local) and add the line:
echo "1" > /proc/sys/net/ipv4/ip_forward

NOTE: The IP forwarding line above is only require when you answer “Yes” to the kernel option “IP:Masquerading (CONFIG_IP_MASQUERADE) and choose to have a server act as a Gateway and masquerade for your inside network.

If you enable IP Masquerading, then the modules ip_masq_ftp.o (for ftp file transfers), ip_masq_irc.o (for irc chats), ip_masq_quake.o (you guessed it), ip_masq_vdolive.o (for VDOLive video connections), ip_masq_cuseeme.o (for CU-SeeMe broadcasts) and ip_masq_raudio.o (for RealAudio downloads) will automatically be compiled. They are needed to make masquerading for these protocols work.

Also, you’ll need to build a modularized kernel and answer “Yes” to the “Enable loadable module support (CONFIG_MODULES)” option instead of a monolithic kernel to be able to use masquerading functions and modules like ip_masq_ftp.o on your Gateway server (see the Kernel section above for more information).

The basic masquerade code described for "IP: masquerading" above only handles TCP or UDP packets (and ICMP errors for existing connections). IP:ICMP Masquerading adds additional support for masquerading ICMP packets, such as ping or the probes used by the Windows 95 tracer program.

NOTE: Remember, other servers like Web Server and Mail Server (like show above) doesn’t need to have these options enable since there have a real IP address assigned or doesn’t act as a Gateway for the inside network.

Some Points to Consider

If you connect your system to the Internet then you can safely assume that you are potentially at risk. Your gateway to the Internet is your greatest exposure, so we recommend the following:

- The gateway should not run any more applications than are absolutely necessary.
- The gateway should strictly limit the type and number of protocols allowed to flow through it (protocols potentially provide security holes, such as FTP and telnet).
- Any system containing confidential or sensitive information should not be directly accessible from the Internet.

Some explanation of rules used in the firewall script files

The following is an explanation of a few of the rules that will be used in the Firewalling examples below. This is just as a reference, the firewall scripts files are well commented and very easy to modify.

Constants used in the firewall scripts files examples

Constants are used for most values. The most basic constants are:

EXTERNAL_INTERFACE
This is the name of the external network interface to the Internet. It's defined as **eth0** in the examples.

LOCAL_INTERFACE_1
This is the name of the internal network interface to the LAN, if any. It's defined as **eth1** in the examples.

LOOPBACK_INTERFACE
This is the name of the loopback interface. It's defined as **lo** in the examples.

IPADDR
This is the IP address of your external interface. It's either a static IP address registered with InterNIC, or else a dynamically assigned address from your ISP (usually via DHCP).

LOCALNET_1
This is your LAN network address, if any - the entire range of IP addresses used by the machines on your LAN. These may be statically assigned, or you might run a local DHCP server to assign them. In these examples, the range is 192.168.1.0/24, part of the Class C private address range.

ANYWHERE
Anywhere is a label for an address used by ipchains to match any (non-broadcast) address. Both programs provide **any/0** as a label for this address, which is 0.0.0.0/0.

NAMESERVER_1
This is the IP address of your Primary DNS Server from your network or your ISP.

NAMESERVER_2
This is the IP address of your Secondary DNS Server from your network or your ISP.

LOOPBACK
The loopback address range is 127.0.0.0/8. The interface itself is addressed as 127.0.0.1 (in /etc/hosts).

PRIVPORTS
The privileged ports, 0 through 1023, are usually referenced in total.

UNPRIVPORTS
The unprivileged ports, 1024 through 65535, are usually referenced in total. They are addresses dynamically assigned to the client side of a connection.

Default Policy
A firewall has a default policy and a collection of actions to take in response to specific message types. This means that if a given packet has not been selected by any other rule, then the default policy rule will be applied.

NOTE: There are two basic approaches to an IPFW firewall, deny everything by default and explicitly allow selected accesses, or accept everything by default and explicitly deny selected accesses. The deny everything policy is the recommended approach, because it's easier to set up a more secure firewall.

Enabling Local Traffic

Since the default policies are to deny everything, some of this must be undone. Local network services do not go through the external network interface. They go through a special, private interface called the loopback interface. None of your local network programs will work until loopback traffic is allowed.

```
# Unlimited traffic on the loopback interface.
ipchains -A input  -i $LOOPBACK_INTERFACE -j ACCEPT
ipchains -A output -i $LOOPBACK_INTERFACE -j ACCEPT
```

Source Address Filtering

The only means of identification under the Internet Protocol (IP) is the source address in the IP packet header. This fact opens the door to source address spoofing, where the sender replaces its address with either a nonexistent address, or the address of some other site. This can allow unsavory types to break into your system or appear as you while attacking other sites.

```
# Refuse spoofed packets pretending to be from the external address.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $IPADDR -l -j DENY
```

Also, there are at least seven sets of source addresses you should refuse on your external interface in all cases.

These are incoming packets claiming to be from:

- Your external IP address
- Class A private IP addresses
- Class B private IP addresses
- Class C private IP addresses
- Class D multicast addresses
- Class E reserved addresses
- The loopback interface

With the exception of your own IP address, blocking outgoing packets containing these source addresses protects you from possible configuration errors on your part.

The rest of the rules

Other rules used in the firewall scripts files are:

- Accessing a Service from the Outside World
- Offering a Service to the Outside World

- Masquerading the Internal Machines

The firewall scripts files

The tool ipchains allows you to set up firewalls, IP masquerading, etc. Ipchains talks to the kernel and tells it what packets to filter. Therefore all your firewall setup are stored in the kernel, and thus will be lost on reboot.

To avoid this, we recommend using the System V init scripts to make your rules permanent. To do this, create a firewall script file like show bellow in your "/etc/rc.d/init.d/" directory for each servers you have. For sure, each server has different services to offer and need different firewall setup. For this reason, we provide you tree different firewall setting, which you can play, examine and fit your needs. Also I assume that you have a minimum knowledge on how filtering firewall and firewall rules works.

Configuration of the "/etc/rc.d/init.d/firewall" script file for the Web Server

This is the configuration script file for our Web Server machine. This configuration allow, unlimited traffic on the Loopback interface, ICMP, DNS Caching and Client Server (53), SSH Server (22), HTTP Server (80), HTTPS Server (443), SMTP Client (25), FTP Server (20, 21), and OUTGOING TRACEROUTE requests by default.

If you don't want some services listed in the firewall rules files for the Web Server that I make ON by default, comment them out with a "#" at the beginning of the line. If you want some other services that I commented out with a "#", then remove the "#" at the beginning of their lines.

Create the **firewall** script file (touch /etc/rc.d/init.d/firewall) on your Web Server and add:

```
#!/bin/sh
#
# ------------------------------------------------------------------------
# Last modified by Gerhard Mourani:  02-01-2000
# ------------------------------------------------------------------------
# Copyright (C) 1997, 1998, 1999  Robert L. Ziegler
#
# Permission to use, copy, modify, and distribute this software and its
# documentation for educational, research, private and non-profit purposes,
# without fee, and without a written agreement is hereby granted.
# This software is provided as an example and basis for individual firewall
# development.  This software is provided without warranty.
#
# Any material furnished by Robert L. Ziegler is furnished on an
# "as is" basis.  He makes no warranties of any kind, either expressed
# or implied as to any matter including, but not limited to, warranty
# of fitness for a particular purpose, exclusivity or results obtained
# from use of the material.
# ------------------------------------------------------------------------
#
# Invoked from /etc/rc.d/init.d/firewall.
# chkconfig: - 60 95
# description: Starts and stops the IPCHAINS Firewall \
#              used to provide Firewall network services.

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

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

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

# See how we were called.
case "$1" in
  start)
        echo -n "Starting Firewalling Services: "

# Some definitions for easy maintenance.

# ------------------------------------------------------------------------
#  EDIT THESE TO SUIT YOUR SYSTEM AND ISP.

EXTERNAL_INTERFACE="eth0"               # whichever you use
LOOPBACK_INTERFACE="lo"
IPADDR="208.164.186.3"
ANYWHERE="any/0"
NAMESERVER_1="208.164.186.1"            # Your primary name server
NAMESERVER_2="208.164.186.2"            # Your secondary name server

SMTP_SERVER="mail.openarch.com"         # Your Mail Hub Server.
SYSLOG_SERVER="mail.openarch.com"       # Your syslog internal server
SYSLOG_CLIENT="208.164.168.0/24"        # Your syslog internal client

LOOPBACK="127.0.0.0/8"
CLASS_A="10.0.0.0/8"
CLASS_B="172.16.0.0/12"
CLASS_C="192.168.0.0/16"
CLASS_D_MULTICAST="224.0.0.0/4"
CLASS_E_RESERVED_NET="240.0.0.0/5"
BROADCAST_SRC="0.0.0.0"
BROADCAST_DEST="255.255.255.255"
PRIVPORTS="0:1023"
UNPRIVPORTS="1024:65535"

# ------------------------------------------------------------------------

# SSH starts at 1023 and works down to 513 for
# each additional simultaneous incoming connection.
SSH_PORTS="1022:1023"                   # range for SSH privileged ports

# traceroute usually uses -S 32769:65535 -D 33434:33523
TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS="33434:33523"

# ------------------------------------------------------------------------
# Default policy is DENY
# Explicitly accept desired INCOMING & OUTGOING connections

   # Remove all existing rules belonging to this filter
   ipchains -F

   # Set the default policy of the filter to deny.
   ipchains -P input  DENY
   ipchains -P output REJECT
   ipchains -P forward REJECT

# ------------------------------------------------------------------------

   # Enable TCP SYN Cookie Protection
   echo 1 >/proc/sys/net/ipv4/tcp_syncookies
```

```
   # Enable IP spoofing protection
   # turn on Source Address Verification
   for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
      echo 1 > $f
   done

   # Disable ICMP Redirect Acceptance
   for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
      echo 0 > $f
   done

   # Disable Source Routed Packets
   for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
      echo 0 > $f
   done

# ------------------------------------------------------------------------
# LOOPBACK

   # Unlimited traffic on the loopback interface.
   ipchains -A input  -i $LOOPBACK_INTERFACE -j ACCEPT
   ipchains -A output -i $LOOPBACK_INTERFACE -j ACCEPT

# ------------------------------------------------------------------------
# Network Ghouls
# Deny access to jerks

   # /etc/rc.d/rc.firewall.blocked contains a list of
   # ipchains -A input  -i $EXTERNAL_INTERFACE -s address -j DENY
   # rules to block from any access.

   # Refuse any connection from problem sites
   #if [ -f /etc/rc.d/rc.firewall.blocked ]; then
   #    . /etc/rc.d/rc.firewall.blocked
   #fi

# ------------------------------------------------------------------------
# SPOOFING & BAD ADDRESSES
# Refuse spoofed packets.
# Ignore blatantly illegal source addresses.
# Protect yourself from sending to bad addresses.

   # Refuse spoofed packets pretending to be from the external address.
   ipchains -A input  -i $EXTERNAL_INTERFACE -s $IPADDR -j DENY -l

   # Refuse packets claiming to be to or from a Class A private network
   ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_A -j DENY -l
   ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_A -j DENY -l
   ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_A -j REJECT -l
   ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_A -j REJECT -l

   # Refuse packets claiming to be to or from a Class B private network
   ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_B -j DENY -l
   ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_B -j DENY -l
   ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_B -j REJECT -l
   ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_B -j REJECT -l

   # Refuse packets claiming to be to or from a Class C private network
   ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_C -j DENY -l
   ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_C -j DENY -l
   ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_C -j REJECT -l
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_C -j REJECT -l

# Refuse packets claiming to be from the loopback interface
ipchains -A input  -i $EXTERNAL_INTERFACE -s $LOOPBACK -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -s $LOOPBACK -j REJECT -l

# Refuse broadcast address SOURCE packets
ipchains -A input  -i $EXTERNAL_INTERFACE -s $BROADCAST_DEST -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -d $BROADCAST_SRC -j DENY -l

# Refuse Class D multicast addresses (in.h) (NET-3-HOWTO)
# Multicast is illegal as a source address.
# Multicast uses UDP.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_D_MULTICAST -j DENY -l

# Refuse Class E reserved IP  addresses
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_E_RESERVED_NET -j DENY -l

# refuse addresses defined as reserved by the IANA
# 0.*.*.*, 1.*.*.*, 2.*.*.*, 5.*.*.*, 7.*.*.*, 23.*.*.*, 27.*.*.*
# 31.*.*.*, 37.*.*.*, 39.*.*.*, 41.*.*.*, 42.*.*.*, 58-60.*.*.*
# 65-95.*.*.*, 96-126.*.*.*, 197.*.*.*, 201.*.*.* (?), 217-223.*.*.*
ipchains -A input  -i $EXTERNAL_INTERFACE -s 1.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 2.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 5.0.0.0/8  -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 7.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 23.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 27.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 31.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 37.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 39.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 41.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 42.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 58.0.0.0/7 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 60.0.0.0/8 -j DENY -l

#65: 01000001    - /3 includes 64 - need 65-79 spelled out
ipchains -A input  -i $EXTERNAL_INTERFACE -s 65.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 66.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 67.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 68.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 69.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 70.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 71.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 72.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 73.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 74.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 75.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 76.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 77.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 78.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 79.0.0.0/8 -j DENY -l

#80: 01010000   - /4 masks 80-95
ipchains -A input  -i $EXTERNAL_INTERFACE -s 80.0.0.0/4 -j DENY -l

# 96: 01100000    - /4 makses 96-111
ipchains -A input  -i $EXTERNAL_INTERFACE -s 96.0.0.0/4 -j DENY -l

#126: 01111110    - /3 includes 127 - need 112-126 spelled out
ipchains -A input  -i $EXTERNAL_INTERFACE -s 112.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 113.0.0.0/8 -j DENY -l
```

```
ipchains -A input  -i $EXTERNAL_INTERFACE -s 114.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 115.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 116.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 117.0.0.0/8  -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 118.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 119.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 120.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 121.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 122.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 123.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 124.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 125.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 126.0.0.0/8 -j DENY -l

#217: 11011001    - /5 includes 216 - need 217-219 spelled out
ipchains -A input  -i $EXTERNAL_INTERFACE -s 217.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 218.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 219.0.0.0/8 -j DENY -l

#223: 11011111    - /6 masks 220-223
ipchains -A input  -i $EXTERNAL_INTERFACE -s 220.0.0.0/6 -j DENY -l

# ------------------------------------------------------------------------------
# ICMP

#    To prevent denial of service attacks based on ICMP bombs, filter
#    incoming Redirect (5) and outgoing Destination Unreachable (3).
#    Note, however, disabling Destination Unreachable (3) is not
#    advisable, as it is used to negotiate packet fragment size.

# For bi-directional ping.
#     Message Types:  Echo_Reply (0),  Echo_Request (8)
#     To prevent attacks, limit the src addresses to your ISP range.
#
# For outgoing traceroute.
#     Message Types:  INCOMING Dest_Unreachable (3), Time_Exceeded (11)
#     default UDP base: 33434 to base+nhops-1
#
# For incoming traceroute.
#     Message Types:  OUTGOING Dest_Unreachable (3), Time_Exceeded (11)
#     To block this, deny OUTGOING 3 and 11

#  0: echo-reply (pong)
#  3: destination-unreachable, port-unreachable, fragmentation-needed, etc.
#  4: source-quench
#  5: redirect
#  8: echo-request (ping)
# 11: time-exceeded
# 12: parameter-problem

ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 0 -d $IPADDR -j ACCEPT
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 3 -d $IPADDR -j ACCEPT
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 4 -d $IPADDR -j ACCEPT
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 11 -d $IPADDR -j ACCEPT
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 12 -d $IPADDR -j ACCEPT
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s 208.164.186.0/24 8 -d $IPADDR -j ACCEPT
```

```
   ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 0 -d 208.164.186.0/24 -j ACCEPT
   ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 3 -d $ANYWHERE -j ACCEPT
   ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 4 -d $ANYWHERE -j ACCEPT
   ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 8 -d $ANYWHERE -j ACCEPT
   ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 12 -d $ANYWHERE -j ACCEPT
   ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
         -s $IPADDR 11 -d 208.164.186.0/24 -j ACCEPT

# ------------------------------------------------------------------------
   # UDP INCOMING TRACEROUTE
   # traceroute usually uses -S 32769:65535 -D 33434:33523

   ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s 208.164.186.0/24 $TRACEROUTE_SRC_PORTS \
         -d $IPADDR $TRACEROUTE_DEST_PORTS -j ACCEPT -l

   ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $ANYWHERE $TRACEROUTE_SRC_PORTS \
         -d $IPADDR $TRACEROUTE_DEST_PORTS -j DENY -l

# ------------------------------------------------------------------------
   # DNS server
   # ----------

   # DNS forwarding, caching only nameserver (53)
   # -------------------------------------------

   # server to server query or response
   # Caching only name server only requires UDP, not TCP

   ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $NAMESERVER_1 53 \
         -d $IPADDR 53 -j ACCEPT

   ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR 53 \
         -d $NAMESERVER_1 53 -j ACCEPT

   ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $NAMESERVER_2 53 \
         -d $IPADDR 53 -j ACCEPT

   ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR 53 \
         -d $NAMESERVER_2 53 -j ACCEPT

   # DNS client (53)
   # ---------------
   ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $NAMESERVER_1 53 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

   ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR $UNPRIVPORTS \
         -d $NAMESERVER_1 53 -j ACCEPT
```

```
   ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $NAMESERVER_1 53 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

   ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $NAMESERVER_1 53 -j ACCEPT

   ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
         -s $NAMESERVER_2 53 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

   ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR $UNPRIVPORTS \
         -d $NAMESERVER_2 53 -j ACCEPT

   ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $NAMESERVER_2 53 \
         -d $IPADDR $UNPRIVPORTS -j ACCEPT

   ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR $UNPRIVPORTS \
         -d $NAMESERVER_2 53 -j ACCEPT

# ------------------------------------------------------------------------
   # TCP accept only on selected ports
   # --------------------------------
   # ---------------------------------------------------------------

   # SSH server (22)
   # ---------------

   ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $UNPRIVPORTS \
         -d $IPADDR 22 -j ACCEPT

   ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 22 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

   ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $SSH_PORTS \
         -d $IPADDR 22 -j ACCEPT

   ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 22 \
         -d $ANYWHERE $SSH_PORTS -j ACCEPT

   # SSH client (22)
   # ---------------
#    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
#          -s $ANYWHERE 22 \
#          -d $IPADDR $UNPRIVPORTS -j ACCEPT

#    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
#          -s $IPADDR $UNPRIVPORTS \
#          -d $ANYWHERE 22 -j ACCEPT

#    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
#          -s $ANYWHERE 22 \
#          -d $IPADDR $SSH_PORTS -j ACCEPT
```

```
#    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
#            -s $IPADDR $SSH_PORTS \
#            -d $ANYWHERE 22 -j ACCEPT

    # ------------------------------------------------------------------

    # HTTP server (80)
    # ----------------

    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
            -s $ANYWHERE $UNPRIVPORTS \
            -d $IPADDR 80 -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
            -s $IPADDR 80 \
            -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

    # ------------------------------------------------------------------

    # HTTPS server (443)
    # ------------------

    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
            -s $ANYWHERE $UNPRIVPORTS \
            -d $IPADDR 443 -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
            -s $IPADDR 443 \
            -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

    # ------------------------------------------------------------------

    # SYSLOG server (514)
    # -------------------

    # Provides full remote logging. Using  this feature you're able to
    # control all syslog messages on one host.

#    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
#            -s $SYSLOG_CLIENT \
#            -d $IPADDR 514 -j ACCEPT

    # SYSLOG client (514)
    # -------------------

#    ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
#            -s $IPADDR 514 \
#            -d $SYSLOG_SERVER 514 -j ACCEPT

    # ------------------------------------------------------------------

    # AUTH server (113)
    # -----------------

    # Reject, rather than deny, the incoming auth port. (NET-3-HOWTO)

    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
            -s $ANYWHERE \
            -d $IPADDR 113 -j REJECT

    # ------------------------------------------------------------------
```

```
# SMTP client (25)
# ----------------
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
        -s $SMTP_SERVER 25 \
        -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
        -s $IPADDR $UNPRIVPORTS \
        -d $SMTP_SERVER 25 -j ACCEPT

# ------------------------------------------------------------------

# FTP server (20, 21)
# -------------------

# incoming request

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
        -s $ANYWHERE $UNPRIVPORTS \
        -d $IPADDR 21 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
        -s $IPADDR 21 \
        -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

# PORT MODE data channel responses
#
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
       -s $ANYWHERE $UNPRIVPORTS \
       -d $IPADDR 20 -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
       -s $IPADDR 20 \
       -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

# PASSIVE MODE data channel responses

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
        -s $ANYWHERE $UNPRIVPORTS \
        -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
        -s $IPADDR $UNPRIVPORTS \
        -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------
# OUTGOING TRACEROUTE
# -------------------
ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
        -s $IPADDR $TRACEROUTE_SRC_PORTS \
        -d $ANYWHERE $TRACEROUTE_DEST_PORTS -j ACCEPT

# ------------------------------------------------------------------------------
# Enable logging for selected denied packets

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
              -d $IPADDR -j DENY -l

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
              -d $IPADDR $PRIVPORTS -j DENY -l

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
```

```
            -d $IPADDR $UNPRIVPORTS -j DENY -l

  ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 5 -d $IPADDR -j DENY -l

  ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 13:255 -d $IPADDR -j DENY -l

# ------------------------------------------------------------------------

    ;;
 stop)
    echo -n "Shutting Firewalling Services: "

  # Remove all existing rules belonging to this filter
  ipchains -F

  # Reset the default policy of the filter to accept.
  ipchains -P input  ACCEPT
  ipchains -P output ACCEPT
  ipchains -P forward ACCEPT

  # Reset TCP SYN Cookie Protection to off.
  echo 0 >/proc/sys/net/ipv4/tcp_syncookies

  # Reset IP spoofing protection to off.
  # turn on Source Address Verification
  for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
     echo 0 > $f
  done

  # Reset ICMP Redirect Acceptance to on.
  for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
     echo 1 > $f
  done

  # Reset Source Routed Packets to on.
  for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
     echo 1 > $f
  done
          ;;
 status)
          echo -n "Now do you show firewalling stats?"
          ;;
 restart|reload)
          $0 stop
          $0 start
          ;;
 *)
          echo "Usage: firewall {start|stop|status|restart|reload}"
          exit 1
esac
```

Now, make this script executable and change its default permission:
[root@deep]# **chmod 700 /etc/rc.d/init.d/firewall**
[root@deep]# **chown 0.0 /etc/rc.d/init.d/firewall**

Create the symbolic rc.d links for your Firewall with the command:
[root@deep]# **chkconfig --add firewall**
[root@deep]# **chkconfig --level 345 firewall on**

Now, your firewall rules are configured to use System V init (System V init is in charge of starting all the normal processes that need to run at boot time) and it will be automatically started each time if your server reboot.

To stop manually the firewall on your system, use the command:
[root@deep]# **/etc/rc.d/init.d/firewall stop**

To start manually the firewall on your system, use the command:
[root@deep]# **/etc/rc.d/init.d/firewall start**

Configuration of the "/etc/rc.d/init.d/firewall" script file for the Mail Server

This is the configuration script file for our Mail Server machine. This configuration allow, unlimited traffic on the Loopback interface, ICMP, DNS Server and Client (53), SSH Server (22), SMTP Server and Client (25), IMAP server (143), and OUTGOING TRACEROUTE requests by default.

If you don't want some services listed in the firewall rules files for the Mail Server that I make ON by default, comment them out with a "#" at the beginning of the line. If you want some other services that I commented out with a "#", then remove the "#" at the beginning of their lines.

Create the **firewall** script file (touch /etc/rc.d/init.d/firewall) on your Mail Server and add:

```
#!/bin/sh
#
# --------------------------------------------------------------------------
# Last modified by Gerhard Mourani:  02-01-2000
# --------------------------------------------------------------------------
# Copyright (C) 1997, 1998, 1999  Robert L. Ziegler
#
# Permission to use, copy, modify, and distribute this software and its
# documentation for educational, research, private and non-profit purposes,
# without fee, and without a written agreement is hereby granted.
# This software is provided as an example and basis for individual firewall
# development.  This software is provided without warranty.
#
# Any material furnished by Robert L. Ziegler is furnished on an
# "as is" basis.  He makes no warranties of any kind, either expressed
# or implied as to any matter including, but not limited to, warranty
# of fitness for a particular purpose, exclusivity or results obtained
# from use of the material.
# --------------------------------------------------------------------------
#
# Invoked from /etc/rc.d/init.d/firewall.
# chkconfig: - 60 95
# description: Starts and stops the IPCHAINS Firewall \
#              used to provide Firewall network services.

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

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

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

# See how we were called.
case "$1" in
  start)
```

```
      echo -n "Starting Firewalling Services: "

# Some definitions for easy maintenance.

# ------------------------------------------------------------------------
#  EDIT THESE TO SUIT YOUR SYSTEM AND ISP.

EXTERNAL_INTERFACE="eth0"               # whichever you use
LOOPBACK_INTERFACE="lo"
IPADDR="208.164.186.2"
ANYWHERE="any/0"
NAMESERVER_1="208.164.186.1"            # Your primary name server
NAMESERVER_2="208.164.186.2"            # Your secondary name server

SYSLOG_SERVER="mail.openarch.com"   # Your syslog internal server
SYSLOG_CLIENT="208.164.168.0/24"        # Your syslog internal client

LOOPBACK="127.0.0.0/8"
CLASS_A="10.0.0.0/8"
CLASS_B="172.16.0.0/12"
CLASS_C="192.168.0.0/16"
CLASS_D_MULTICAST="224.0.0.0/4"
CLASS_E_RESERVED_NET="240.0.0.0/5"
BROADCAST_SRC="0.0.0.0"
BROADCAST_DEST="255.255.255.255"
PRIVPORTS="0:1023"
UNPRIVPORTS="1024:65535"

# ------------------------------------------------------------------------

# SSH starts at 1023 and works down to 513 for
# each additional simultaneous incoming connection.
SSH_PORTS="1022:1023"                   # range for SSH privileged ports

# traceroute usually uses -S 32769:65535 -D 33434:33523
TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS="33434:33523"

# ------------------------------------------------------------------------
# Default policy is DENY
# Explicitly accept desired INCOMING & OUTGOING connections

   # Remove all existing rules belonging to this filter
   ipchains -F

   # Set the default policy of the filter to deny.
   ipchains -P input  DENY
   ipchains -P output REJECT
   ipchains -P forward REJECT

# ------------------------------------------------------------------------

   # Enable TCP SYN Cookie Protection
   echo 1 >/proc/sys/net/ipv4/tcp_syncookies

   # Enable IP spoofing protection
   # turn on Source Address Verification
   for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
      echo 1 > $f
   done

   # Disable ICMP Redirect Acceptance
```

```
   for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
      echo 0 > $f
   done

   # Disable Source Routed Packets
   for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
      echo 0 > $f
   done

# ------------------------------------------------------------------------
# LOOPBACK

   # Unlimited traffic on the loopback interface.
   ipchains -A input  -i $LOOPBACK_INTERFACE -j ACCEPT
   ipchains -A output -i $LOOPBACK_INTERFACE -j ACCEPT

# ------------------------------------------------------------------------
# Network Ghouls
# Deny access to jerks

   # /etc/rc.d/rc.firewall.blocked contains a list of
   # ipchains -A input  -i $EXTERNAL_INTERFACE -s address -j DENY
   # rules to block from any access.

   # Refuse any connection from problem sites
   #if [ -f /etc/rc.d/rc.firewall.blocked ]; then
   #    . /etc/rc.d/rc.firewall.blocked
   #fi

# ------------------------------------------------------------------------
# SPOOFING & BAD ADDRESSES
# Refuse spoofed packets.
# Ignore blatantly illegal source addresses.
# Protect yourself from sending to bad addresses.

   # Refuse spoofed packets pretending to be from the external address.
   ipchains -A input  -i $EXTERNAL_INTERFACE -s $IPADDR -j DENY -l

   # Refuse packets claiming to be to or from a Class A private network
   ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_A -j DENY -l
   ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_A -j DENY -l
   ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_A -j REJECT -l
   ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_A -j REJECT -l

   # Refuse packets claiming to be to or from a Class B private network
   ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_B -j DENY -l
   ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_B -j DENY -l
   ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_B -j REJECT -l
   ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_B -j REJECT -l

   # Refuse packets claiming to be to or from a Class C private network
   ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_C -j DENY -l
   ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_C -j DENY -l
   ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_C -j REJECT -l
   ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_C -j REJECT -l

   # Refuse packets claiming to be from the loopback interface
   ipchains -A input  -i $EXTERNAL_INTERFACE -s $LOOPBACK -j DENY -l
   ipchains -A output -i $EXTERNAL_INTERFACE -s $LOOPBACK -j REJECT -l

   # Refuse broadcast address SOURCE packets
   ipchains -A input  -i $EXTERNAL_INTERFACE -s $BROADCAST_DEST -j DENY -l
```

```
ipchains -A input  -i $EXTERNAL_INTERFACE -d $BROADCAST_SRC -j DENY -l

# Refuse Class D multicast addresses (in.h) (NET-3-HOWTO)
# Multicast is illegal as a source address.
# Multicast uses UDP.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_D_MULTICAST -j DENY -l

# Refuse Class E reserved IP  addresses
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_E_RESERVED_NET -j DENY -l

# refuse addresses defined as reserved by the IANA
# 0.*.*.*, 1.*.*.*, 2.*.*.*, 5.*.*.*, 7.*.*.*, 23.*.*.*, 27.*.*.*
# 31.*.*.*, 37.*.*.*, 39.*.*.*, 41.*.*.*, 42.*.*.*, 58-60.*.*.*
# 65-95.*.*.*, 96-126.*.*.*, 197.*.*.*, 201.*.*.* (?), 217-223.*.*.*
ipchains -A input  -i $EXTERNAL_INTERFACE -s 1.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 2.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 5.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 7.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 23.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 27.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 31.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 37.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 39.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 41.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 42.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 58.0.0.0/7 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 60.0.0.0/8 -j DENY -l

#65: 01000001    - /3 includes 64 - need 65-79 spelled out
ipchains -A input  -i $EXTERNAL_INTERFACE -s 65.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 66.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 67.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 68.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 69.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 70.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 71.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 72.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 73.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 74.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 75.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 76.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 77.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 78.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 79.0.0.0/8 -j DENY -l

#80: 01010000   - /4 masks 80-95
ipchains -A input  -i $EXTERNAL_INTERFACE -s 80.0.0.0/4 -j DENY -l

# 96: 01100000    - /4 makses 96-111
ipchains -A input  -i $EXTERNAL_INTERFACE -s 96.0.0.0/4 -j DENY -l

#126: 01111110    - /3 includes 127 - need 112-126 spelled out
ipchains -A input  -i $EXTERNAL_INTERFACE -s 112.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 113.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 114.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 115.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 116.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 117.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 118.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 119.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 120.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 121.0.0.0/8 -j DENY -l
```

```
ipchains -A input  -i $EXTERNAL_INTERFACE -s 122.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 123.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 124.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 125.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 126.0.0.0/8 -j DENY -l

#217: 11011001    - /5 includes 216 - need 217-219 spelled out
ipchains -A input  -i $EXTERNAL_INTERFACE -s 217.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 218.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 219.0.0.0/8 -j DENY -l

#223: 11011111    - /6 masks 220-223
ipchains -A input  -i $EXTERNAL_INTERFACE -s 220.0.0.0/6 -j DENY -l

# ------------------------------------------------------------------------------
# ICMP

#    To prevent denial of service attacks based on ICMP bombs, filter
#    incoming Redirect (5) and outgoing Destination Unreachable (3).
#    Note, however, disabling Destination Unreachable (3) is not
#    advisable, as it is used to negotiate packet fragment size.

# For bi-directional ping.
#     Message Types:  Echo_Reply (0),  Echo_Request (8)
#     To prevent attacks, limit the src addresses to your ISP range.
#
# For outgoing traceroute.
#     Message Types:  INCOMING Dest_Unreachable (3), Time_Exceeded (11)
#     default UDP base: 33434 to base+nhops-1
#
# For incoming traceroute.
#     Message Types:  OUTGOING Dest_Unreachable (3), Time_Exceeded (11)
#     To block this, deny OUTGOING 3 and 11

#  0: echo-reply (pong)
#  3: destination-unreachable, port-unreachable, fragmentation-needed, etc.
#  4: source-quench
#  5: redirect
#  8: echo-request (ping)
# 11: time-exceeded
# 12: parameter-problem

ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 0 -d $IPADDR -j ACCEPT
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 3 -d $IPADDR -j ACCEPT
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 4 -d $IPADDR -j ACCEPT
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 11 -d $IPADDR -j ACCEPT
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 12 -d $IPADDR -j ACCEPT
ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s 208.164.186.0/24 8 -d $IPADDR -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
        -s $IPADDR 0 -d 208.164.186.0/24 -j ACCEPT
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
        -s $IPADDR 3 -d $ANYWHERE -j ACCEPT
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
        -s $IPADDR 4 -d $ANYWHERE -j ACCEPT
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
```

```
        -s $IPADDR 8 -d $ANYWHERE -j ACCEPT
    ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
        -s $IPADDR 12 -d $ANYWHERE -j ACCEPT
    ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
        -s $IPADDR 11 -d 208.164.186.0/24 -j ACCEPT

# ------------------------------------------------------------------------
    # UDP INCOMING TRACEROUTE
    # traceroute usually uses -S 32769:65535 -D 33434:33523

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
        -s 208.164.186.0/24 $TRACEROUTE_SRC_PORTS \
        -d $IPADDR $TRACEROUTE_DEST_PORTS -j ACCEPT -l

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
        -s $ANYWHERE $TRACEROUTE_SRC_PORTS \
        -d $IPADDR $TRACEROUTE_DEST_PORTS -j DENY -l

# ------------------------------------------------------------------------
    # DNS server
    # ----------

    # DNS: full server
    # server/client to server query or response

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
        -s $ANYWHERE $UNPRIVPORTS \
        -d $IPADDR 53 -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
        -s $IPADDR 53 \
        -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

    # DNS client (53)
    # ---------------
    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
        -s $NAMESERVER_1 53 \
        -d $IPADDR $UNPRIVPORTS -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
        -s $IPADDR $UNPRIVPORTS \
        -d $NAMESERVER_1 53 -j ACCEPT

    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
        -s $NAMESERVER_1 53 \
        -d $IPADDR $UNPRIVPORTS -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
        -s $IPADDR $UNPRIVPORTS \
        -d $NAMESERVER_1 53 -j ACCEPT

# ------------------------------------------------------------------------
    # TCP accept only on selected ports
    # ---------------------------------
    # ---------------------------------------------------------------

    # SSH server (22)
    # ---------------

    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
        -s $ANYWHERE $UNPRIVPORTS \
        -d $IPADDR 22 -j ACCEPT
```

```
   ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 22 \
         -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

   ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE $SSH_PORTS \
         -d $IPADDR 22 -j ACCEPT

   ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
         -s $IPADDR 22 \
         -d $ANYWHERE $SSH_PORTS -j ACCEPT

   # SSH client (22)
   # ---------------
#  ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
#        -s $ANYWHERE 22 \
#        -d $IPADDR $UNPRIVPORTS -j ACCEPT

#  ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
#        -s $IPADDR $UNPRIVPORTS \
#        -d $ANYWHERE 22 -j ACCEPT

#  ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
#        -s $ANYWHERE 22 \
#        -d $IPADDR $SSH_PORTS -j ACCEPT

#  ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
#        -s $IPADDR $SSH_PORTS \
#        -d $ANYWHERE 22 -j ACCEPT

   # ------------------------------------------------------------------

   # AUTH server (113)
   # -----------------

   # Reject, rather than deny, the incoming auth port. (NET-3-HOWTO)

   ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
         -s $ANYWHERE \
         -d $IPADDR 113 -j REJECT

   # ------------------------------------------------------------------

   # SYSLOG server (514)
   # -----------------

   # Provides full remote logging. Using  this feature you're able to
   # control all syslog messages on one host.

#  ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
#        -s $SYSLOG_CLIENT \
#        -d $IPADDR 514 -j ACCEPT

   # SYSLOG client (514)
   # -----------------

#  ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
#        -s $IPADDR 514 \
#        -d $SYSLOG_SERVER 514 -j ACCEPT

   # ------------------------------------------------------------------
```

```
    # SMTP server (25)
    # ----------------

    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
          -s $ANYWHERE $UNPRIVPORTS \
          -d $IPADDR 25 -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
          -s $IPADDR 25 \
          -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

    # SMTP client (25)
    # ----------------
    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
          -s $ANYWHERE 25 \
          -d $IPADDR $UNPRIVPORTS -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
          -s $IPADDR $UNPRIVPORTS \
          -d $ANYWHERE 25 -j ACCEPT

    # ------------------------------------------------------------------

    # IMAP server (143)
    # -----------------

    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
          -s $ANYWHERE $UNPRIVPORTS \
          -d $IPADDR 143 -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
          -s $IPADDR 143 \
          -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

    # ------------------------------------------------------------------

    # OUTGOING TRACEROUTE
    # -------------------
    ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
          -s $IPADDR $TRACEROUTE_SRC_PORTS \
          -d $ANYWHERE $TRACEROUTE_DEST_PORTS -j ACCEPT

# ------------------------------------------------------------------------
# Enable logging for selected denied packets

    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
                -d $IPADDR -j DENY -l

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
                -d $IPADDR $PRIVPORTS -j DENY -l

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
                -d $IPADDR $UNPRIVPORTS -j DENY -l

    ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
          -s $ANYWHERE 5 -d $IPADDR -j DENY -l

    ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
          -s $ANYWHERE 13:255 -d $IPADDR -j DENY -l

# ------------------------------------------------------------------------
```

```
    ;;
 stop)
    echo -n "Shutting Firewalling Services: "

  # Remove all existing rules belonging to this filter
  ipchains -F

  # Reset the default policy of the filter to accept.
  ipchains -P input  ACCEPT
  ipchains -P output ACCEPT
  ipchains -P forward ACCEPT

  # Reset TCP SYN Cookie Protection to off.
  echo 0 >/proc/sys/net/ipv4/tcp_syncookies

  # Reset IP spoofing protection to off.
  # turn on Source Address Verification
  for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
     echo 0 > $f
  done

  # Reset ICMP Redirect Acceptance to on.
  for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
     echo 1 > $f
  done

  # Reset Source Routed Packets to on.
  for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
     echo 1 > $f
  done
          ;;
 status)
          echo -n "Now do you show firewalling stats?"
          ;;
 restart|reload)
          $0 stop
          $0 start
          ;;
 *)
          echo "Usage: firewall {start|stop|status|restart|reload}"
          exit 1
esac
```

Now, make this script executable and change its default permission:
[root@deep]# **chmod 700 /etc/rc.d/init.d/firewall**
[root@deep]# **chown 0.0 /etc/rc.d/init.d/firewall**

Create the symbolic rc.d links for your Firewall with the command:
[root@deep]# **chkconfig --add firewall**
[root@deep]# **chkconfig --level 345 firewall on**

Now, your firewall rules are configured to use System V init (System V init is in charge of starting all the normal processes that need to run at boot time) and it will be automatically started each time if your server reboot.

To stop manually the firewall on your system, use the command:
[root@deep]# **/etc/rc.d/init.d/firewall stop**

To start manually the firewall on your system, use the command:

[root@deep]# **/etc/rc.d/init.d/firewall start**

Configuration of the "/etc/rc.d/init.d/firewall" script file for the Gateway Server

This is the configuration script file for our Gateway Server machine. This configuration allow, unlimited traffic on the Loopback interface, ICMP, DNS Server and Client (53), SSH Server and Client (22), HTTP Server and Client (80), HTTPS Server and Client (443), POP Client (110), NNTP NEWS Client (119), SMTP Server and Client (25), IMAP Server (143), IRC Client (6667), ICQ Client (4000), FTP Client (20, 21), RealAudio / QuickTime Client, and OUTGOING TRACEROUTE requests by default.

If you don't want some services listed in the firewall rules files for the Gateway Server that I make ON by default, comment them out with a "#" at the beginning of the line. If you want some other services that I commented out with a "#", then remove the "#" at the beginning of their lines. If you are configured Masquerading on your server, uncomment the modules necessary to masquerade their respective services that you need like ip_masq_irc.o, ip_masq_raudio.o, etc.

Create the **firewall** script file (touch /etc/rc.d/init.d/firewall) on your Gateway Server and add:

```
#!/bin/sh
#
# ------------------------------------------------------------------------
# Last modified by Gerhard Mourani:  02-01-2000
# ------------------------------------------------------------------------
# Copyright (C) 1997, 1998, 1999  Robert L. Ziegler
#
# Permission to use, copy, modify, and distribute this software and its
# documentation for educational, research, private and non-profit purposes,
# without fee, and without a written agreement is hereby granted.
# This software is provided as an example and basis for individual firewall
# development.  This software is provided without warranty.
#
# Any material furnished by Robert L. Ziegler is furnished on an
# "as is" basis.  He makes no warranties of any kind, either expressed
# or implied as to any matter including, but not limited to, warranty
# of fitness for a particular purpose, exclusivity or results obtained
# from use of the material.
# ------------------------------------------------------------------------
#
# Invoked from /etc/rc.d/init.d/firewall.
# chkconfig: - 60 95
# description: Starts and stops the IPCHAINS Firewall \
#              used to provide Firewall network services.

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

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

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

# See how we were called.
case "$1" in
  start)
        echo -n "Starting Firewalling Services: "

# Some definitions for easy maintenance.
```

```
# ------------------------------------------------------------------------
#  EDIT THESE TO SUIT YOUR SYSTEM AND ISP.

EXTERNAL_INTERFACE="eth0"               # whichever you use
LOCAL_INTERFACE_1="eth1"                # whichever you use
LOOPBACK_INTERFACE="lo"

IPADDR="208.164.186.1"
LOCALNET_1="192.168.1.0/24"             # whatever private range you use
ANYWHERE="any/0"
NAMESERVER_1="208.164.186.1"
NAMESERVER_2="208.164.186.2"

POP_SERVER="pop.videotron.ca"           # Your pop external server
NEWS_SERVER="news.videotron.ca"         # Your news external server
SYSLOG_SERVER="mail.openarch.com"       # Your syslog internal server

LOOPBACK="127.0.0.0/8"
CLASS_A="10.0.0.0/8"
CLASS_B="172.16.0.0/12"
CLASS_C="192.168.0.0/16"
CLASS_D_MULTICAST="224.0.0.0/4"
CLASS_E_RESERVED_NET="240.0.0.0/5"
BROADCAST_SRC="0.0.0.0"
BROADCAST_DEST="255.255.255.255"
PRIVPORTS="0:1023"
UNPRIVPORTS="1024:65535"

# ------------------------------------------------------------------------

# SSH starts at 1023 and works down to 513 for
# each additional simultaneous incoming connection.
SSH_PORTS="1022:1023"                   # range for SSH privileged ports

# traceroute usually uses -S 32769:65535 -D 33434:33523
TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS="33434:33523"

# ------------------------------------------------------------------------
# Default policy is DENY
# Explicitly accept desired INCOMING & OUTGOING connections

   # Remove all existing rules belonging to this filter
   ipchains -F

   # Set the default policy of the filter to deny.
   ipchains -P input  DENY
   ipchains -P output REJECT
   ipchains -P forward REJECT

   # set masquerade timeout to 10 hours for tcp connections
   ipchains -M -S 36000 0 0

   # Don't forward fragments. Assemble before forwarding.
   ipchains -A output -f -i $LOCAL_INTERFACE_1 -j DENY

# ------------------------------------------------------------------------

   # Enable TCP SYN Cookie Protection
   echo 1 >/proc/sys/net/ipv4/tcp_syncookies
```

```
   # Enable IP spoofing protection
   # turn on Source Address Verification
   for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
      echo 1 > $f
   done

   # Disable ICMP Redirect Acceptance
   for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
      echo 0 > $f
   done

   # Disable Source Routed Packets
   for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
      echo 0 > $f
   done

   # These modules are necessary to masquerade their respective services.
   /sbin/modprobe ip_masq_ftp.o
   /sbin/modprobe ip_masq_raudio.o ports=554,7070,7071,6970,6971
   /sbin/modprobe ip_masq_irc.o
   #/sbin/modprobe/ip_masq_vdolive.o
   #/sbin/modprobe/ip_masq_cuseeme.o
   #/sbin/modprobe/ip_masq_quake.o

# ------------------------------------------------------------------------
# LOOPBACK

   # Unlimited traffic on the loopback interface.
   ipchains -A input  -i $LOOPBACK_INTERFACE -j ACCEPT
   ipchains -A output -i $LOOPBACK_INTERFACE -j ACCEPT

# ------------------------------------------------------------------------
# Network Ghouls
# Deny access to jerks

   # /etc/rc.d/rc.firewall.blocked contains a list of
   # ipchains -A input  -i $EXTERNAL_INTERFACE -s address -j DENY
   # rules to block from any access.

   # Refuse any connection from problem sites
   #if [ -f /etc/rc.d/rc.firewall.blocked ]; then
   #    . /etc/rc.d/rc.firewall.blocked
   #fi

# ------------------------------------------------------------------------
# SPOOFING & BAD ADDRESSES
# Refuse spoofed packets.
# Ignore blatantly illegal source addresses.
# Protect yourself from sending to bad addresses.

   # Refuse spoofed packets pretending to be from the external address.
   ipchains -A input  -i $EXTERNAL_INTERFACE -s $IPADDR -j DENY -l

   # Refuse packets claiming to be to or from a Class A private network
   ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_A -j DENY -l
   ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_A -j DENY -l
   ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_A -j REJECT -l
   ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_A -j REJECT -l

   # Refuse packets claiming to be to or from a Class B private network
   ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_B -j DENY -l
   ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_B -j DENY -l
```

```
ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_B -j REJECT -l
ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_B -j REJECT -l

# Refuse packets claiming to be to or from a Class C private network
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_C -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -d $CLASS_C -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_C -j REJECT -l
ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_C -j REJECT -l

# Refuse packets claiming to be from the loopback interface
ipchains -A input  -i $EXTERNAL_INTERFACE -s $LOOPBACK -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -s $LOOPBACK -j REJECT -l

# Refuse broadcast address SOURCE packets
ipchains -A input  -i $EXTERNAL_INTERFACE -s $BROADCAST_DEST -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -d $BROADCAST_SRC -j DENY -l

# Refuse Class D multicast addresses (in.h) (NET-3-HOWTO)
# Multicast is illegal as a source address.
# Multicast uses UDP.
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_D_MULTICAST -j DENY -l

# Refuse Class E reserved IP  addresses
ipchains -A input  -i $EXTERNAL_INTERFACE -s $CLASS_E_RESERVED_NET -j DENY -l

# refuse addresses defined as reserved by the IANA
# 0.*.*.*, 1.*.*.*, 2.*.*.*, 5.*.*.*, 7.*.*.*, 23.*.*.*, 27.*.*.*
# 31.*.*.*, 37.*.*.*, 39.*.*.*, 41.*.*.*, 42.*.*.*, 58-60.*.*.*
# 65-95.*.*.*, 96-126.*.*.*, 197.*.*.*, 201.*.*.* (?), 217-223.*.*.*
ipchains -A input  -i $EXTERNAL_INTERFACE -s 1.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 2.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 5.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 7.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 23.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 27.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 31.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 37.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 39.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 41.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 42.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 58.0.0.0/7 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 60.0.0.0/8 -j DENY -l

#65: 01000001    - /3 includes 64 - need 65-79 spelled out
ipchains -A input  -i $EXTERNAL_INTERFACE -s 65.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 66.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 67.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 68.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 69.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 70.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 71.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 72.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 73.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 74.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 75.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 76.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 77.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 78.0.0.0/8 -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -s 79.0.0.0/8 -j DENY -l

#80: 01010000   - /4 masks 80-95
ipchains -A input  -i $EXTERNAL_INTERFACE -s 80.0.0.0/4 -j DENY -l
```

```
  # 96: 01100000    - /4 makses 96-111
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 96.0.0.0/4 -j DENY -l

  #126: 01111110    - /3 includes 127 - need 112-126 spelled out
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 112.0.0.0/8 -j DENY -l
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 113.0.0.0/8 -j DENY -l
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 114.0.0.0/8 -j DENY -l
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 115.0.0.0/8 -j DENY -l
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 116.0.0.0/8 -j DENY -l
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 117.0.0.0/8 -j DENY -l
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 118.0.0.0/8 -j DENY -l
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 119.0.0.0/8 -j DENY -l
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 120.0.0.0/8 -j DENY -l
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 121.0.0.0/8 -j DENY -l
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 122.0.0.0/8 -j DENY -l
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 123.0.0.0/8 -j DENY -l
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 124.0.0.0/8 -j DENY -l
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 125.0.0.0/8 -j DENY -l
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 126.0.0.0/8 -j DENY -l

  #217: 11011001    - /5 includes 216 - need 217-219 spelled out
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 217.0.0.0/8 -j DENY -l
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 218.0.0.0/8 -j DENY -l
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 219.0.0.0/8 -j DENY -l

  #223: 11011111    - /6 masks 220-223
  ipchains -A input  -i $EXTERNAL_INTERFACE -s 220.0.0.0/6 -j DENY -l

# ------------------------------------------------------------------------
# ICMP

  #    To prevent denial of service attacks based on ICMP bombs, filter
  #    incoming Redirect (5) and outgoing Destination Unreachable (3).
  #    Note, however, disabling Destination Unreachable (3) is not
  #    advisable, as it is used to negotiate packet fragment size.

  # For bi-directional ping.
  #     Message Types:  Echo_Reply (0),  Echo_Request (8)
  #     To prevent attacks, limit the src addresses to your ISP range.
  #
  # For outgoing traceroute.
  #     Message Types:  INCOMING Dest_Unreachable (3), Time_Exceeded (11)
  #     default UDP base: 33434 to base+nhops-1
  #
  # For incoming traceroute.
  #     Message Types:  OUTGOING Dest_Unreachable (3), Time_Exceeded (11)
  #     To block this, deny OUTGOING 3 and 11

  #  0: echo-reply (pong)
  #  3: destination-unreachable, port-unreachable, fragmentation-needed, etc.
  #  4: source-quench
  #  5: redirect
  #  8: echo-request (ping)
  # 11: time-exceeded
  # 12: parameter-problem

  ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 0 -d $IPADDR -j ACCEPT
  ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 3 -d $IPADDR -j ACCEPT
  ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
```

```
        -s $ANYWHERE 4 -d $IPADDR -j ACCEPT
    ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 11 -d $IPADDR -j ACCEPT
    ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s $ANYWHERE 12 -d $IPADDR -j ACCEPT
    ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
        -s 208.164.186.0/24 8 -d $IPADDR -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
        -s $IPADDR 0 -d 208.164.186.0/24 -j ACCEPT
    ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
        -s $IPADDR 3 -d $ANYWHERE -j ACCEPT
    ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
        -s $IPADDR 4 -d $ANYWHERE -j ACCEPT
    ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
        -s $IPADDR 8 -d $ANYWHERE -j ACCEPT
    ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
        -s $IPADDR 12 -d $ANYWHERE -j ACCEPT
    ipchains -A output -i $EXTERNAL_INTERFACE -p icmp \
        -s $IPADDR 11 -d 208.164.186.0/24 -j ACCEPT

# ------------------------------------------------------------------------------
    # UDP INCOMING TRACEROUTE
    # traceroute usually uses -S 32769:65535 -D 33434:33523

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
        -s 208.164.186.0/24 $TRACEROUTE_SRC_PORTS \
        -d $IPADDR $TRACEROUTE_DEST_PORTS -j ACCEPT -l

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
        -s $ANYWHERE $TRACEROUTE_SRC_PORTS \
        -d $IPADDR $TRACEROUTE_DEST_PORTS -j DENY -l

# ------------------------------------------------------------------------------
    # DNS server
    # ----------

    # DNS: full server
    # server/client to server query or response

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
        -s $ANYWHERE $UNPRIVPORTS \
        -d $IPADDR 53 -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
        -s $IPADDR 53 \
        -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

    # DNS client (53)
    # ---------------
    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
        -s $NAMESERVER_1 53 \
        -d $IPADDR $UNPRIVPORTS -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
        -s $IPADDR $UNPRIVPORTS \
        -d $NAMESERVER_1 53 -j ACCEPT

    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
        -s $NAMESERVER_1 53 \
        -d $IPADDR $UNPRIVPORTS -j ACCEPT
```

```
    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
          -s $IPADDR $UNPRIVPORTS \
          -d $NAMESERVER_1 53 -j ACCEPT

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
          -s $NAMESERVER_2 53 \
          -d $IPADDR $UNPRIVPORTS -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
          -s $IPADDR $UNPRIVPORTS \
          -d $NAMESERVER_2 53 -j ACCEPT

    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
          -s $NAMESERVER_2 53 \
          -d $IPADDR $UNPRIVPORTS -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
          -s $IPADDR $UNPRIVPORTS \
          -d $NAMESERVER_2 53 -j ACCEPT

# ------------------------------------------------------------------------------
    # TCP accept only on selected ports
    # --------------------------------
    # ------------------------------------------------------------------

    # SSH server (22)
    # ---------------

    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
          -s $ANYWHERE $UNPRIVPORTS \
          -d $IPADDR 22 -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
          -s $IPADDR 22 \
          -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
          -s $ANYWHERE $SSH_PORTS \
          -d $IPADDR 22 -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
          -s $IPADDR 22 \
          -d $ANYWHERE $SSH_PORTS -j ACCEPT

    # SSH client (22)
    # ---------------
    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
          -s $ANYWHERE 22 \
          -d $IPADDR $UNPRIVPORTS -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
          -s $IPADDR $UNPRIVPORTS \
          -d $ANYWHERE 22 -j ACCEPT

    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
          -s $ANYWHERE 22 \
          -d $IPADDR $SSH_PORTS -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
          -s $IPADDR $SSH_PORTS \
          -d $ANYWHERE 22 -j ACCEPT
```

```
# ------------------------------------------------------------------

# HTTP client (80)
# ---------------
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
       -s $ANYWHERE 80 \
       -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
       -s $IPADDR $UNPRIVPORTS \
       -d $ANYWHERE 80 -j ACCEPT

# ------------------------------------------------------------------

# HTTPS client (443)
# ------------------
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
       -s $ANYWHERE 443 \
       -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
       -s $IPADDR $UNPRIVPORTS \
       -d $ANYWHERE 443 -j ACCEPT

# ------------------------------------------------------------------

# POP client (110)
# ---------------
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
       -s $POP_SERVER 110 \
       -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
       -s $IPADDR $UNPRIVPORTS \
       -d $POP_SERVER 110 -j ACCEPT

# ------------------------------------------------------------------

# NNTP NEWS client (119)
# ----------------------
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
       -s $NEWS_SERVER 119 \
       -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
       -s $IPADDR $UNPRIVPORTS \
       -d $NEWS_SERVER 119 -j ACCEPT

# ------------------------------------------------------------------

# FINGER client (79)
# ------------------
#  ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
#         -s $ANYWHERE 79 \
#         -d $IPADDR $UNPRIVPORTS -j ACCEPT

#  ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
#         -s $IPADDR $UNPRIVPORTS \
#         -d $ANYWHERE 79 -j ACCEPT

# ------------------------------------------------------------------
```

```
# SYSLOG client (514)
# -----------------

#  ipchains -A output -i $LOCAL_INTERFACE_1 -p udp \
#          -s $IPADDR 514 \
#          -d $SYSLOG_SERVER 514 -j ACCEPT

# ------------------------------------------------------------------

# AUTH server (113)
# -----------------

# Reject, rather than deny, the incoming auth port. (NET-3-HOWTO)

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
        -s $ANYWHERE \
        -d $IPADDR 113 -j REJECT

# AUTH client (113)
# -----------------
#  ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
#          -s $ANYWHERE 113 \
#          -d $IPADDR $UNPRIVPORTS -j ACCEPT

#  ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
#          -s $IPADDR $UNPRIVPORTS \
#          -d $ANYWHERE 113 -j ACCEPT

# ------------------------------------------------------------------

# SMTP client (25)
# ----------------
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
        -s $ANYWHERE 25 \
        -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
        -s $IPADDR $UNPRIVPORTS \
        -d $ANYWHERE 25 -j ACCEPT

# ------------------------------------------------------------------

# IRC client (6667)
# -----------------
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
        -s $ANYWHERE 6667 \
        -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
        -s $IPADDR $UNPRIVPORTS \
        -d $ANYWHERE 6667 -j ACCEPT

# ------------------------------------------------------------------

# ICQ client (4000)
# -----------------
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
        -s $ANYWHERE 2000:4000 \
        -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
        -s $IPADDR $UNPRIVPORTS \
```

```
        -d $ANYWHERE 2000:4000 -j ACCEPT

ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
        -s $ANYWHERE 4000 \
        -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
        -s $IPADDR $UNPRIVPORTS \
        -d $ANYWHERE 4000 -j ACCEPT

# ------------------------------------------------------------------

# FTP client (20, 21)
# -------------------

# outgoing request
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
        -s $ANYWHERE 21 \
        -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
        -s $IPADDR $UNPRIVPORTS \
        -d $ANYWHERE 21 -j ACCEPT

# NORMAL mode data channel
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
       -s $ANYWHERE 20 \
       -d $IPADDR $UNPRIVPORTS -j ACCEPT

# NORMAL mode data channel responses
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp ! -y \
       -s $IPADDR $UNPRIVPORTS \
       -d $ANYWHERE 20 -j ACCEPT

# PASSIVE mode data channel creation
ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
       -s $IPADDR $UNPRIVPORTS \
       -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

# PASSIVE mode data channel responses
ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
       -s $ANYWHERE $UNPRIVPORTS \
       -d $IPADDR $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# RealAudio / QuickTime client
# ----------------------------

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
        -s $ANYWHERE 554 \
        -d $IPADDR $UNPRIVPORTS -j ACCEPT

ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
        -s $IPADDR $UNPRIVPORTS \
        -d $ANYWHERE 554 -j ACCEPT


# TCP is a more secure method:  7070:7071

ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
        -s $ANYWHERE 7070:7071 \
```

```
        -d $IPADDR $UNPRIVPORTS -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
        -s $IPADDR $UNPRIVPORTS \
        -d $ANYWHERE 7070:7071 -j ACCEPT


    # UDP is the preferred method:  6970:6999
    # For LAN machines, UDP requires the RealAudio masquerading module and
    # the ipmasqadm third-party software.

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
        -s $ANYWHERE $UNPRIVPORTS \
        -d $IPADDR 6970:6999 -j ACCEPT

    ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
        -s $IPADDR $UNPRIVPORTS \
        -d $ANYWHERE $UNPRIVPORTS -j ACCEPT

    # ------------------------------------------------------------------

    # WHOIS client (43)
    # -----------------
#    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp ! -y \
#          -s $ANYWHERE 43 \
#          -d $IPADDR $UNPRIVPORTS -j ACCEPT

#    ipchains -A output -i $EXTERNAL_INTERFACE -p tcp \
#          -s $IPADDR $UNPRIVPORTS \
#          -d $ANYWHERE 43 -j ACCEPT

    # ------------------------------------------------------------------

    # OUTGOING TRACEROUTE
    # -------------------
    ipchains -A output -i $EXTERNAL_INTERFACE -p udp \
        -s $IPADDR $TRACEROUTE_SRC_PORTS \
        -d $ANYWHERE $TRACEROUTE_DEST_PORTS -j ACCEPT

# ------------------------------------------------------------------------
# Unlimited traffic within the local network.

    # All internal machines have access to the firewall machine.

    ipchains -A input  -i $LOCAL_INTERFACE_1 -s $LOCALNET_1 -j ACCEPT
    ipchains -A output -i $LOCAL_INTERFACE_1 -d $LOCALNET_1 -j ACCEPT

# ------------------------------------------------------------------------
# Masquerade internal traffic.

    # All internal traffic is masqueraded externally.

    ipchains -A forward -i $EXTERNAL_INTERFACE -s $LOCALNET_1 -j MASQ


# ------------------------------------------------------------------------
# Enable logging for selected denied packets

    ipchains -A input  -i $EXTERNAL_INTERFACE -p tcp \
              -d $IPADDR -j DENY -l

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
```

```
                -d $IPADDR $PRIVPORTS -j DENY -l

    ipchains -A input  -i $EXTERNAL_INTERFACE -p udp \
                -d $IPADDR $UNPRIVPORTS -j DENY -l

    ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
            -s $ANYWHERE 5 -d $IPADDR -j DENY -l

    ipchains -A input  -i $EXTERNAL_INTERFACE -p icmp \
            -s $ANYWHERE 13:255 -d $IPADDR -j DENY -l

# ------------------------------------------------------------------------

        ;;
  stop)
        echo -n "Shutting Firewalling Services: "

    # Remove all existing rules belonging to this filter
    ipchains -F

    # Reset the default policy of the filter to accept.
    ipchains -P input  ACCEPT
    ipchains -P output ACCEPT
    ipchains -P forward ACCEPT

    # Reset TCP SYN Cookie Protection to off.
    echo 0 >/proc/sys/net/ipv4/tcp_syncookies

    # Reset IP spoofing protection to off.
    # turn on Source Address Verification
    for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
       echo 0 > $f
    done

    # Reset ICMP Redirect Acceptance to on.
    for f in /proc/sys/net/ipv4/conf/*/accept_redirects; do
       echo 1 > $f
    done

    # Reset Source Routed Packets to on.
    for f in /proc/sys/net/ipv4/conf/*/accept_source_route; do
       echo 1 > $f
    done
            ;;
  status)
            echo -n "Now do you show firewalling stats?"
            ;;
  restart|reload)
            $0 stop
            $0 start
            ;;
  *)
            echo "Usage: firewall {start|stop|status|restart|reload}"
            exit 1
esac
```

Now, make this script executable and change its default permission:
[root@deep]# **chmod 700 /etc/rc.d/init.d/firewall**
[root@deep]# **chown 0.0 /etc/rc.d/init.d/firewall**

Create the symbolic rc.d links for your Firewall with the command:

[root@deep]# **chkconfig --add firewall**
[root@deep]# **chkconfig --level 345 firewall on**

Now, your firewall rules are configured to use System V init (System V init is in charge of starting all the normal processes that need to run at boot time) and it will be automatically started each time if your server reboot.

To stop manually the firewall on your system, use the command:
[root@deep]# **/etc/rc.d/init.d/firewall stop**

To start manually the firewall on your system, use the command:
[root@deep]# **/etc/rc.d/init.d/firewall start**

Deny access to some address

Some time you know an address that you would like to block from any access on your server. You can do that by creating the **rc.firewall.blocked** file under "/etc/rc.d/" directory and uncomment the following lines in your firewall rules scripts file:

Edit your **firewall** scripts file (vi /etc/rc.d/init.d/firewall) and uncomment the following lines:

```
if [ -f /etc/rc.d/rc.firewall.blocked ]; then
   . /etc/rc.d/rc.firewall.blocked
fi
```

Create the **rc.firewall.blocked** file (touch /etc/rc.d/rc.firewall.blocked) and add inside this file all IP address you wan to block from any access on your server:

For example:
204.254.45.9
187.231.11.5

Further documentation

For more details, there are several man pages you can read:

$ ipchains (8)	- IP firewall administration
$ ipchains-restore (8)	- restore IP firewall chains from stdin
$ ipchains-save (8)	- save IP firewall chains to stdout

IPCHAINS Administrative Tools

The commands listed bellow are some that we use often in our regular use but much more exist and you must check the man page and documentation for more details and information.

ipchains

Ipchains is used to set up, maintain, and inspect the IP firewall rules in the Linux kernel. These rules can be divided into 4 different categories: the IP input chain, the IP output chain, the IP forwarding chain, and user defined chains.

- To list all rules in the selected chain, use the command:
  [root@deep]# **ipchains -L**

This command will list all rules in the selected chain. If no chain is selected, all chains are listed.

- To list all input rules in the selected chain, use the command:
  [root@deep]# **ipchains -L input**

This command will list all input rules in the selected chain.

- To list all output rules in the selected chain, use the command:
  [root@deep]# **ipchains -L output**

This command will list all output rules in the selected chain.

- To list all forward rules in the selected chain, use the command:
  [root@deep]# **ipchains -L forward**

This command will list all forward rules in the selected chain. This work only if you are configured Masquerading on your server.

- To list all masquerades rules in the selected chain, use the command:
  [root@deep]# **ipchains -ML**

This option allows viewing of the currently masqueraded connections. This work only if you are configured Masquerading on your server.

- To list all rules in numeric output in the selected chain, use the command:
  [root@deep]# **ipchains -nL**

This command will list all rules in numeric output. IP addresses and port numbers will be printed in numeric format. By default, the program will try to display them as host names, network names, or services (whenever applicable).

Part V Software's-Related Reference

In this Part

Compilers functionality
Securities Software
Servers Software

Chapter 8 Compilers Functionality

In this Chapter

The necessary packages
Why would we choose to use tarballs?
Compiling software on your system
Build and Install software on your system

Compilers functionality

Overview

Before we begin to explain how to compile and install server software with all the necessary securities and optimizations that we will need on our server, it is important to know the commands and programs we'll use often to do the job. First of all, we must ensure that we have the necessary packages needed to make compilation on our system. Those packages must be installed on your server or you'll not be able to compile programs.

The necessary packages

The following is the necessary packages needed to be able to make compilation on your system **after recompilation of your kernel**. Those software are on your Red Hat 6.1 Part 1 CD-ROM under RedHat/RPMS directory if there are not already installed.

[root@deep]# **mount /dev/cdrom /mnt/cdrom/**
[root@deep]# **cd /mnt/cdrom/RedHat/RPMS/**

autoconf-2.13-5.noarch.rpm
m4-1.4-12.i386.rpm
automake-1.4-5.noarch.rpm
dev86-0.14.9-1.i386.rpm
bison-1.28-1.i386.rpm
byacc-1.9-11.i386.rpm
cdecl-2.5-9.i386.rpm
cpp-1.1.2-24.i386.rpm
cproto-4.6-2.i386.rpm
ctags-3.2-1.i386.rpm
egcs-1.1.2-24.i386.rpm
ElectricFence-2.1-1.i386.rpm
flex-2.5.4a-7.i386.rpm
gdb-4.18-4.i386.rpm
glibc-devel-2.1.2-11.i386.rpm
make-3.77-6.i386.rpm
patch-2.5-9.i386.rpm

- The RPM command to install a RPM package on your system is:
  [root@deep]# **rpm -Uvh foo-1.0-2.i386.rpm**

- The RPM command to verify if package are or are not installed on your system is:
  [root@deep]# **rpm -q foo**

Once again, after installation and compilation of all programs that you need on your server, it's important to uninstall all sharp objects (compilers, etc) describe above. This will protect your system from unauthorized users trying to compile programs on your server without authorization.

Another thing to do is to move the "**rpm**" binary program in a safe place like the floppy disk for the same reason that above. Imagine some evil peoples trying to compile program on your server and realize that compilers are not available. He will switch to import programs RPM on the server and install it with the RPM commands. Hops surprise! RPM commands are not available too. Of course in the future if you need to install new software on your server that require RPM program, all you have to do is to put it from the floppy disk to his original place.

- To move RPM binary in the floppy disk, use the command:
  [root@deep]# **mount /dev/fd0 /mnt/floppy/**
  [root@deep]# **mv /bin/rpm /mnt/floppy**
  [root@deep]# **umount /mnt/floppy/**

- To put RPM binary to his original directory, use the command:
  [root@deep]# **mount /dev/fd0 /mnt/floppy/**
  [root@deep]# **cp /mnt/floppy/rpm /bin/**
  [root@deep]# **umount /mnt/floppy/**

NOTE: Never uninstall RPM program completely from your system or you will be unable to reinstall it again later since to install RPM or other software you need to have RPM commands available.

Why would we choose to use tarballs?

The Red Hat distribution of Linux are provided as RPM files. An RPM file, also known as a "package" is a way of distributing software so that it can be easily installed, upgraded, queried, and deleted. However, in the Unix world the defacto-standard for package distribution continues to be by way of so-called "tarballs". Tarballs are simply files that are readable with the "tar" utility. Installing from tar is usually significantly more tedious than using RPM. So why would we choose to do so?

1- Unfortunately, it takes a few weeks for developers to get the latest version of a package converted to RPM because many developers first release them as tarballs.

2- When developers release a new RPM, they include a lot options that often are not necessary. Developers don't know what options you will need and what you will not need, so they include the most used to fit the needs of every one.

3- Often RPM are not optimized for your specific processors, companies like Red Hat Linux build RPM based on a standard PC. This permit their RPM packages to be installed on all sort of computers since compiling programs for a i386 machine can fit on all systems.

4- Some time you download and install RPM, that other peoples around the world are build and make available for your purpose. This can pose conflicts in certain cases depending how this man are build the package, errors, security and all other problems describes above.

Compiling software on your system

Roughly speaking, a program is something a computer can execute. Somebody wrote the "source code" in a language he/she could understand. That might have been C (very likely) or some such thing. The program "source code" also makes sense to a compiler that converts the instructions into a binary file suited to whatever processor is wanted - e.g. a 386 or similar. A modern file format for these "executable" programs is Elf. The programmer shows his source to the compiler and gets a result of some sort. It's not at all uncommon that early attempts fail to compile, or having compiled, fail to act as expected. Half of programming is tracking down and fixing these problems (debugging).

More aspect and new words relating to compilation of a source code that you will see and use in this book include but are not limited to:

The Multiple Files

One-file programs are quite rare. Usually there are a number of files (say *.c) that are each compiled into object files (*.o) and then linked into an executable. The compiler is usually used to perform the linking (and it calls the 'ld' program behind the scenes).

The Makefiles
These are intended to aid consistency (you build your program the same way each time). They also often help with speed. Some compiles are quite long - tens of minutes for a large program. The 'make' program uses 'dependencies' in the Makefile to decide what parts of the program need to be recompiled. If you change one source file out of fifty you hope to get away with one compile and one link step, instead of starting from scratch. Be aware that the format includes tabs at the start of some lines (spaces won't do).

The Libraries
Programs can be linked not only to object files (*.o) but also to libraries (collections of object files). You will sometimes have to link to system-supplied libraries (e.g. -lm for the math's library, without which your mathematical C programs will produce nonsense). There are two forms of linking to libraries: static (the code goes in the executable file) and dynamic (the code is collected when the program starts to run). Large compiler manuals spend a page discussing the pros and cons of the two.

The Patches
Before, it was common for executable files to be given corrections without recompiling them. This practice has deservedly almost died out. Now people 'patch' source code, putting a change into files (usually changing a small proportion of the whole). Larry Wall's 'patch' program is used for this. Where different versions of a program are required small changes to code can be released, saving the trouble of having two large distributions.

The Errors in Compilation and Linking
These are often typos, omissions, and misuse of the language.... Checks that the right includes files are used for the functions you are calling. Unreferenced symbols are the sign of an incomplete link step. Also checks if the necessary development libraries or tools are installed on your system.

The Debugging
This is a large topic. It usually helps to have statements in the code that inform you of what is happening. To avoid drowning in output you might sometimes get them to print out only the first 3 passes in a loop. Checking that variables have passed correctly between modules often helps. Get familiar with your debugging tools.

Build and Install software on your system

You will see in the "Part VI Software's-Related-Reference" bellow that we use many different commands to build and install programs on the server. These commands are UNIX compatible and are used on all variant off *nix machines to compile and install software.

The procedure to compile and install tarballs software on you server follow:

1. First of all you must download the tarball from your trusted software archive site.

2. After downloading the tarball, change to the "/var/tmp/" directory (note that other paths are possible) and untar the archive by typing commands (as root) as in the following example:

   [root@deep]# **tar xzpf foo.tar.gz**

The above command will extract all files from the example “foo.tar.gz” compressed archive.

“x” option tells tar to extract all files from the archive.
“z” option tells tar that the archive is compressed with gzip.
“p” option maintains the original and permissions the files had as the archive was created.
“f” option tells tar that the very next argument is the file name.

Once the tarball has been decompressed into the appropriate directory, you will almost certainly find a “README” or a “INSTALL” file included with the newly decompressed files, with further instructions on how to prepare the software package for use. Likely, you will need to enter commands similar to the following example:

```
./configure
make
make install
```

The above commands “**./configure**” would configure the software to ensure your system has the necessary functionality and libraries to successfully compile the package, “**make**” compile all source files into executable binaries, and then “**make install**” install the binaries and any supporting files into the appropriate locations. Other specifics commands that you’ll see on our book for compilation and installation procedure will be:

```
make depend
strip
chown
```

The "**make depend**" command would build and make the necessary dependency of different files. The “**strip**” command would discard all symbols from the object files. This means that our binary file will be smaller in size. This will improve a bit the performance hit to the program since they will be fewer lines to read by the system when it’ll execute the binary. The "**chown**" command would set the correct files owner and group permission for the binaries.

NOTE: More commands will be explained in the concerned installation section.

At this part of our book, all software-listed on chapter 9 and 10 are optional and depend of what you wan to install or doing on your server. What king of job your server will do and for which part of your network Intranet/Internet etc. In other part it will be interesting for you to replace Telnet program with Ssh for secure remote administration. Another interesting program is Tripwire that aids system administrators and users in monitoring a designated set of files for any changes.

Chapter 9 Securities Software

In this Chapter

Linux sXid

Overview

sXid is an all in one suid/sgid monitoring program designed to be run from cron on a regular basis. Basically it tracks any changes in your s[ug]id files and folders. If there are any new ones, ones that aren't set any more, or they have changed bits or other modes then it reports the changes in an easy to read format via email or on the command line. sXid will automate the task to find all SUID/SGID on your server and report them to you. Once installed you forget it and it will make the job for you.

These installation instructions assume

Commands are Unix-compatible.
The source path is "/var/tmp" (other paths are possible).
Installations were tested on RedHat Linux 6.1.
All steps in the installation will happen in superuser account "root".
sXid version number is 4.0.1

Packages

sXid FTP Site: ftp://marcus.seva.net/pub/sxid/
You must be sure to download: sxid_4_0_1_tar.gz

Tarballs

It is a good idea to make a list of files on the system before you install Bind, and one afterwards, and then compare them using '**diff**' to find out what file it placed where. Simply run '**find /* > sxid1**' before and '**find /* > sxid2**' after you install the software, and use '**diff sxid1 sxid2 > sxid**' to get a list of what changed.

Compilation

Decompress the tarball (tar.gz).

```
[root@deep]# cp sxid_version_tar.gz /var/tmp/
[root@deep]# cd /var/tmp
[root@deep]# tar xzpf sxid_version_tar.gz
```

Compile and Optimize

Cd into the new sXid directory and type the following commands on your terminal:

```
make install
```

The above commands would configure the software to ensure your system has the necessary functionality and libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Cleanup after work

```
[root@deep]# cd /var/tmp
[root@deep]# rm -rf sxid-version/ sxid_version_tar.gz
```

The “rm” command will remove all the source files we have used to compile and install sXid. It will also remove the sXid compressed archive from the “/var/tmp” directory.

Configurations

All software we describe in our book "Linuxsos.pdf" will have a specific directory and subdirectory in a tar compressed archive named “floppy.tgz” containing file configurations for the specific program. If you get this archive file, you wouldn’t be obliged to reproduce the different configuration files bellow manually or cut and past them to create your configuration files. Whatever your decide to copy manually or get the files made to your convenience from the archive compressed files, it will be to your responsibility to modify, adjust for your needs and place the files related to sXid software to their appropriated places on your server machine, like show bellow. The server configuration files archive is located at the following Internet address: http://pages.infinit.net/lotus1/doc/opti/floppy.tgz

- To run sXid, the following file is require and must be create or copied to the appropriated directory on your server.

  Copy the **sxid.conf** file to the “/etc/” directory.

You can obtain configuration files listed bellow on our floppy.tgz archive. Copy the following files from the decompressed floppy.tgz archive to their appropriated places or copy and paste them directly from this book to the concerned file.

Configure the “/etc/sxid.conf” file

The configuration file for sXid (“/etc/sxid.conf”) allows you to set options that modify the operation of the program. It is well commented and very basic.

Step 1
Edit the **sxid.conf** file (vi /etc/sxid.conf) and set your needs:

```
# Configuration file for sXid
# Note that all directories must be absolute with no trailing /'s

# Where to begin our file search
SEARCH = "/"

# Which subdirectories to exclude from searching
EXCLUDE = "/proc /mnt /cdrom /floppy"

# Who to send reports to
EMAIL = "root"

# Always send reports, even when there are no changes?
ALWAYS_NOTIFY = "no"

# Where to keep interim logs. This will rotate 'x' number of
# times based on KEEP_LOGS below
LOG_FILE = "/var/log/sxid.log"

# How many logs to keep
KEEP_LOGS = "5"

# Rotate the logs even when there are no changes?
ALWAYS_ROTATE = "no"

# Directories where +s is forbidden (these are searched
```

```
# even if not explicitly in SEARCH), EXCLUDE rules apply
FORBIDDEN = "/home /tmp"

# Remove (-s) files found in forbidden directories?
ENFORCE = "yes"

# This implies ALWAYS_NOTIFY. It will send a full list of
# entries along with the changes
LISTALL = "no"

# Ignore entries for directories in these paths
# (this means that only files will be recorded, you
# can effectively ignore all directory entries by
# setting this to "/"). The default is /home since
# some systems have /home g+s.
IGNORE_DIRS = "/home"

# File that contains a list of (each on it's own line)
# of other files that sxid should monitor. This is useful
# for files that aren't +s, but relate to system
# integrity (tcpd, inetd, apache...).
# EXTRA_LIST = "/etc/sxid.list"

# Mail program. This changes the default compiled in
# mailer for reports. You only need this if you have changed
# it's location and don't want to recompile sxid.
# MAIL_PROG = "/usr/bin/mail"
```

Step 2
Place an entry into root's crontabs to make sXid run as a cronjob:
SXid will run from crond, basically it tracks any changes in your s[ug]id files and folders. If there are any new ones, ones that aren't set any more, or they have changed bits or other modes then it reports the changes. To add sxid in your cronjob you must edit the crontab and add the following line:

- To edit the crontab, use the command (as root):
  [root@deep]# **crontab -e**

```
# Sample crontab entry to run every day at 4am
0 4 * * * /usr/bin/sxid
```

Further documentation

For more details, there are several man pages you can read:

```
$ man sxid.conf (5)      - configuration settings for sxid
$ man sxid (1)           - check for changes in s[ug]id files and directories
```

sXid Administrative Tools

This program is meant to run as a cronjob. It must run once a day, but busy shell boxes may want to run it twice a day. You can also run this manually for spot checking.

- To run sxid manually, use the command:
  [root@deep]# **sxid -k**
  sXid Vers : 4.0.1
  Check run : Wed Dec 29 12:40:32 1999
  This host : mail.openarch.com

```
Spotcheck  : /home/admin
Excluding  : /proc /mnt /cdrom /floppy
Ignore Dirs: /home
Forbidden  : /home /tmp


No changes found
```

This checks for changes by recursing the current working directory. Log files will not be rotated and no email sent. All output will go to stdout.

Installed files

> /etc/sxid.conf
> /usr/bin/sxid
> /usr/man/man1/sxid.1
> /usr/man/man5/sxid.conf.5

Linux SSH1 Client/Server

Overview

SSH is a truly seamless and secure replacement of old, insecure remote login programs such as rlogin or rsh. According to the official SSH (Secure Shell) site, SSH is the secure login program that revolutionized remote management of networks hosts over the Internet. It is a powerful, very easy-to-use program that uses strong cryptography for protecting all transmitted confidential data, including passwords, binary files, and administrative commands. The major benefit of SSH1 is that it is completely free for both end users and commercial companies.

In our configuration we are configured sshd1 to support tcp-wrappers (inetd super server) for more security. SSH2 was originally free but is now under a commercial license, it is recommended to use SSH1 (free) instead of SSH2 (commercial). We provide in our configuration the both versions.

These installation instructions assume

Commands are Unix-compatible.
The source path is "/var/tmp" (other paths are possible).
Installations were tested on RedHat Linux 6.1.
All steps in the installation will happen in superuser account "root".
Ssh1 version number is 1.2.27

Packages

SSH1 Homepage: http://www.ssh.fi/
You must be sure to download: ssh-1.2.27.tar.gz

Tarballs

It is a good idea to make a list of files on the system before you install ssh1, and one afterwards, and then compare them using '**diff**' to find out what file it placed where. Simply run '**find /* > ssh1**' before and '**find /* > ssh2**' after you install the software, and use '**diff ssh1 ssh2 > ssh**' to get a list of what changed.

Compilation

Decompress the tarball (tar.gz).

```
[root@deep]# cp ssh-version.tar.gz /var/tmp
[root@deep]# cd /var/tmp
[root@deep]# tar xzpf ssh-version.tar.gz
```

Compile and Optimize

Cd into the new Ssh1 directory and type the following commands on your terminal:

```
CC="egcs" \
CFLAGS="-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions" \
./configure \
--prefix=/usr \
--with-etcdir=/etc/ssh \
--without-idea \
--enable-warnings \
--without-rsh \
--with-libwrap \
--disable-server-port-forwardings \
--disable-client-port-forwardings \
--disable-server-x11-forwarding \
--disable-client-x11-forwarding \
--disable-suid-ssh
```

This tells SSH1 to set itself up for this particular hardware setup with:

- Avoids patent problems in commercial use.
- Enable the -Wall (warning) option if using gcc/egcs.
- Do not use rsh under any conditions.
- Compile in libwrap (tcp_wrappers) support.
- Disable all port forwardings in server (except X11).
- Disable all port forwardings in client (except X11).
- Disable X11 forwarding in server.
- Disable X11 forwarding in client.
- Install ssh without suid bit.

```
[root@deep]# make clean
[root@deep]# make
[root@deep]# make install
```

The "**make clean**", erase all previous traces of a compilation so as to avoid any mistakes, then "**make**" compile all source files into executable binaries, and finally "**make install**" install the binaries and any supporting files into the appropriate locations.

Cleanup after work

```
[root@deep]# cd /var/tmp
[root@deep]# rm -rf ssh1-version/ ssh-version.tar.gz
```

The "rm" command will remove all the source files we have used to compile and install SSH1. It will also remove the SSH1 compressed archive from the "/var/tmp" directory.

Configurations

All software we describe in our book "Linuxsos.pdf" will have a specific directory and subdirectory in a tar compressed archive named “floppy.tgz” containing file configurations for the specific program. If you get this archive file, you wouldn’t be obliged to reproduce the different configuration files bellow manually or cut and past them to create your configuration files. Whatever your decide to copy manually or get the files make to your convenience from the archive compressed files, it will be to your responsibility to modify, adjust for your needs and place the files related to SSH1 software to their appropriated places on your server machine, like show bellow. The server configuration files archive is located at the following Internet address: http://pages.infinit.net/lotus1/doc/opti/floppy.tgz

- To run SSH1 Client/Server, the following files are require and must be create or copied to their appropriated directories on your server.

  Copy the **sshd_config** file to the “/etc/ssh/” directory.
  Copy the **ssh_config** file to the “/etc/ssh/” directory.

You can obtain configuration files listed bellow on our floppy.tgz archive. Copy the following files from the decompressed floppy.tgz archive to their appropriated places or copy and paste them directly from this book to the concerned file.

Configure the “/etc/ssh/ssh_config” file

The configuration file for ssh1 (“/etc/ssh/ssh_config”) allows you to set options that modify the operation of the client programs. The file contain keyword-value pairs, one per line, with keywords being case insensitive. Here are the more important keywords; a complete listing is available in the man page for ssh (1).

Edit the **ssh_config** file (vi /etc/ssh/ssh_config) and add:

```
# Site-wide defaults for various options

 Host *
   ForwardAgent no
   ForwardX11 no
   RhostsAuthentication no
   RhostsRSAAuthentication no
   RSAAuthentication yes
   TISAuthentication no
   PasswordAuthentication yes
   FallBackToRsh no
   UseRsh no
   BatchMode no
   Compression yes
   StrictHostKeyChecking no
   IdentityFile ~/.ssh/identity
   Port 22
   KeepAlive yes
   Cipher blowfish
   EscapeChar ~
```

This tells ssh_config file to set itself up for this particular configuration setup with:

*Host **

This option “Host” restricts declarations (up to the next Host keyword) to be only for those hosts that match one of the patterns given after the keyword. A single ´*´ as a pattern can be used to provide global defaults for all hosts.

ForwardAgent no
This option “ForwardAgent” specifies whether the connection to the authentication agent (if any) will be forwarded to the remote machine.

ForwardX11 no
This option “ForwardX11” specifies whether X11 connections will be automatically redirected over the secure channel and DISPLAY set.

RhostsAuthentication no
This option “RhostsAuthentication” specifies whether to try rhosts based authentication.

RhostsRSAAuthentication no
This option “RhostsRSAAuthentication” specifies whether to try rhosts based authentication with RSA host authentication.

RSAAuthentication yes
This option “RSAAuthentication” specifies whether to try RSA authentication.

TISAuthentication no
This option “TISAuthentication” specifies whether to try TIS authentication.

PasswordAuthentication yes
This option “PasswordAuthentication” specifies whether to use password authentication.

FallBackToRsh no
This option “FallBackToRsh” specifies that if connecting via ssh fails due to a connection refused error rsh should automatically be used instead.

UseRsh no
This option “UseRsh” specifies that rlogin/rsh should be used for this host.

BatchMode no
This option “BatchMode” if set to "yes", passphrase/password querying will be disabled. This option is useful in scripts and other batch jobs where you have no user to supply the password.

Compression yes
This option “Compression” specifies whether to use compression. Compression will improve communication speed.

StrictHostKeyChecking no
This option “StrictHostKeyChecking” if set to "yes", ssh will never automatically add host keys to the $HOME/.ssh/known_hosts file, and refuses to connect hosts whose host key has changed. This provides maximum protection against Trojan horse attacks.

IdentityFile ~/.ssh/identity
This option “IdentityFile” specifies the file from which the user's RSA authentication identity is read.

Port 22
This option “Port” specifies the port number to connect on the remote host.

KeepAlive yes

This option “KeepAlive” specifies whether the system should send keep alive messages to the other side. If they are sent, death of the connection or crash of one of the machines will be properly noticed.

Cipher blowfish
This option “Cipher” specifies the cipher to use for encrypting the session.

EscapeChar ~
This option “EscapeChar” sets the escape character.

Configure the “/etc/ssh/sshd_config” file

The configuration file for sshd1 (“/etc/ssh/sshd_config”) allows you to set options that modify the operation of the daemon. The files contain keyword-value pairs, one per line, with keywords being case insensitive. Here are the more important keywords; a complete listing is available in the man page for sshd (8).

Edit the **sshd_config** file (vi /etc/ssh/sshd_config) and add:

```
# This is ssh server systemwide configuration file.

Port 22
ListenAddress 192.168.1.1
HostKey /etc/ssh/ssh_host_key
RandomSeed /etc/ssh/ssh_random_seed
ServerKeyBits 1024
LoginGraceTime 600
KeyRegenerationInterval 3600
PermitRootLogin no
IgnoreRhosts yes
StrictModes yes
QuietMode no
X11Forwarding no
FascistLogging no
PrintMotd yes
KeepAlive yes
SyslogFacility AUTH
RhostsAuthentication no
RhostsRSAAuthentication no
RSAAuthentication yes
PasswordAuthentication yes
PermitEmptyPasswords no
AllowUsers admin
AllowHosts 192.168.1.4
```

This tells sshd_config file to set itself up for this particular configuration setup with:

Port 22
This option “Port” specifies the port number that sshd listens on.

ListenAddress 192.168.1.1
This option “ListenAddress” Specifies the ip address of the interface where the sshd server socket is bind.

HostKey /etc/ssh/ssh_host_key
This option “HostKey” specifies the file containing the private host key.

RandomSeed /etc/ssh/ssh_random_seed
This option “RandomSeed” specifies the file containing the random seed for the server; this file is created automatically and updated regularly.

ServerKeyBits 1024
This option “ServerKeyBits” defines the number of bits in the server key.

LoginGraceTime 600
This option “LoginGraceTime” specifies the time in second the server will wait before disconnecting if the user has not successfully logged in.

KeyRegenerationInterval 3600
This option “KeyRegenerationInterval” mean that the server key is automatically regenerated after this many seconds (if it has been used). The purpose of regeneration is to prevent decrypting captured sessions by later breaking into the machine and stealing the keys.

PermitRootLogin no
This option “PermitRootLogin” specifies whether the root can log in using ssh. Never say yes to this option.

IgnoreRhosts yes
This option “IgnoreRhosts” specifies that rhosts and shosts files will not be used in authentication.

StrictModes yes
This option “StrictModes” specifies whether ssh should check file modes and ownership of the user's home directory and rhosts files before accepting login. This is normally desirable because novices sometimes accidentally leave their directory or files world-writable.

QuietMode no
This option “QuietMode” specifies whether the system runs in quiet mode. In quiet mode, nothing is logged in the system log, except fatal errors.

X11Forwarding no
This option “X11Forwarding” specifies whether X11 forwarding is permitted.

FascistLogging no
This option “FascistLogging” specifies whether to use verbose logging. Verbose logging violates the privacy of users and is not recommended.

PrintMotd yes
This option “PrintMotd” specifies whether sshd should print “/etc/motd” when a user logs in interactively.

KeepAlive yes
This option “KeepAlive” specifies whether the system should send keep alive messages to the other side. If they are sent, death of the connection or crash of one of the machines will be properly noticed.

SyslogFacility AUTH
This option “SyslogFacility” gives the facility code that is used when logging messages from sshd.

RhostsAuthentication no
This option “RhostsAuthentication” specifies whether authentication using rhosts or “/etc/hosts.equiv” files is sufficient.

RhostsRSAAuthentication no

This option “RhostsRSAAuthentication” specifies whether rhosts or “/etc/hosts.equiv” authentication together with successful RSA host authentication is allowed.

RSAAuthentication yes
This option “RSAAuthentication” specifies whether pure RSA authentication is allowed.

PasswordAuthentication yes
This option “PasswordAuthentication” specifies whether password authentication is allowed.

PermitEmptyPasswords no
This option “PermitEmptyPasswords” when password authentication is allowed, it specifies whether the server allows login to accounts with empty password strings.

AllowUsers admin
This option “AllowUsers” can be followed by any number of user name patterns or user@host patterns, separated by spaces. Host name may be either the dns name or the ip address.

AllowHosts 192.168.1.4
This option “AllowHosts” can be followed by any number of host name patterns, separated by spaces. If specified, .shosts (and .rhosts and “/etc/hosts.equiv”) entries are only honored for hosts whose name matches one of the patterns. Servers are used to map the client's host into a canonical host name. If the name cannot be mapped, its IP-address is used as the host name.

Configure sshd1 to use tcp-wrappers inetd super server

Tcp-wrappers take cares to start and stop sshd1 server. Upon execution, inetd reads its configuration information from a configuration file which, by default, is “/etc/inetd.conf”. There must be an entry for each field of the configuration file, with entries for each field separated by a tab or a space.

Step 1
Edit the **inetd.conf** file (vi /etc/inetd.conf) and add the line:

```
ssh     stream  tcp     nowait  root    /usr/sbin/tcpd  sshd -i
```

NOTE: The -i parameter is important since is specifies that sshd is being run from inetd. Also, update your “inetd.conf” file by sending a SIGHUP signal (killall -HUP inetd) after adding the line.

```
[root@deep /root]# killall -HUP inetd
```

Step 2
Edit the **hosts.allow** file (vi /etc/hosts.allow) and add the line:

```
sshd: 192.168.1.4 win.openarch.com
```

Which mean client IP “192.168.1.4” with host name “win.openarch.com” is allowed to ssh on the server.

These "daemon" strings (for tcp-wrappers) are in use by sshd1:
sshdfwd-X11 (if you want to allow/deny X11-forwarding).
sshdfwd-<port-number> (for tcp-forwarding).
sshdfwd-<port-name> (port-name defined in /etc/services. Used in tcp-forwarding).

NOTE: If you do decide to switch to using ssh, make sure you install and use it on **all** your servers. Having ten secure servers and one insecure is a waste of time.

Further documentation

For more details, there are several man pages you can read:

```
$ man ssh-add1 (1)       - adds identities for the authentication agent
$ man ssh-agent1 (1)     - authentication agent
$ man ssh-keygen1 (1)    - authentication key pair generation
$ man ssh1 (1)           - secure shell client (remote login program)
$ man sshd1 (8)          - secure shell daemon
```

Ssh1 Per-User Configuration

Step 1
Create your private & public keys of local, by executing:
[root@deep]# **su username**
[username@deep]$ **ssh-keygen1**

The result should look like the following example:

```
Initializing random number generator...
Generating p:  .............................++ (distance 430)
Generating q:  .......................++ (distance 456)
Computing the keys...
Testing the keys...
Key generation complete.
Enter file in which to save the key (/home/username/.ssh/identity): [Press Enter]
Enter passphrase:
Enter the same passphrase again:
Your identification has been saved in /home/username/.ssh/identity.
Your public key is:
1024 37
14937757511251955533691120318477293862290049394715136511145806108870001764378494676831
29757784315853227236120610062314604405364871843677484233240919418480988907860997175244
46977589647127757030728779973708569993017043141563536333068888944038178461608592483844
59020215410275690305584653406336563558489976540218 1 username@deep.openarch.com
Your public key has been saved in /home/username/.ssh/identity.pub
```

NOTE: If you have multiple accounts you might want to create a separate key on each of them. You may want to have separate keys for:

- Your Mail server
- Your Web server
- Your GW server

This allows you to limit access between these servers, e.g. not allowing the Mail account to access your Web account or the machines in the GW. This enhances the overall security in the case any of authentication keys are compromised for some reason.

Step 2
Copy your public keys of local (**identity.pub**), to “/home/username/.ssh” directory of remote under the name, say, “authorized_keys”.

NOTE: One way to copy the file is to use the ftp command or you might need to send your public key in electronic mail to the administrator of the system. Just include the contents of the ~/.ssh/identity.pub file in the message.

If access to the remote system is still denied you should check the permissions of the following files on it:

- The home directory itself
- The ~/.ssh directory
- The ~/.ssh/authorized_keys file

The permissions should allow writing only by you (the owner). This example shows the permissions you could use.

```
[admin@deep]$ cd
[admin@deep admin]$ ls -ld . .ssh .ssh/authorized_keys
drwx------    5 admin    admin        1024  Nov 28 07:05 .
drwxr-xr-x    2 admin    admin        1024  Nov 29 00:02 .ssh
-rw-r--r--    1 admin    admin         342   Nov 29 00:02 .ssh/authorized_keys
```

Changing your pass-phrase
You can change the pass-phrase at any time by using the -p option of ssh-keygen.

- To change the pass-phrase, use the command:
  [root@deep]# **su username**
  [username@deep]$ **ssh-keygen1 -p**
  Enter file key is in (/home/username/.ssh/identity): [Press ENTER]
  Enter old passphrase:
  Key has comment 'username@deep.openarch.com'
  Enter new passphrase:
  Enter the same passphrase again:
  Your identification has been saved with the new passphrase.

SSH1 Users Tools

The commands listed bellow are some that we use often in our regular use but much more exist and you must check the man page and documentation for more details and information.

ssh1
Ssh1 (Secure Shell) is a program for logging into a remote machine and executing commands in a remote machine. It is intended to replace rlogin and rsh, and provide secure encrypted communications between two untrusted hosts over an insecure network. X11 connections and arbitrary TCP/IP ports can also be forwarded over the secure channel.

- To logging to a remote machine, use the command:
  [root@deep]# **ssh1 <login_name> <hostname>**
  For example:
  [root@deep]# **ssh1 username www.openarch.com**
  username@deep.openarch.com's password:
  Last login: Tue Oct 19 1999 18:13:00 -0400 from gate.openarch.com
  Welcome to www.openarch.com on Deepforest.

Where <login_name> is the name you use to connect to ssh server and <hostname> is the address of your ssh server.

scp1
You can copy files from the local system to a remote system or vice versa, or even between two remote systems using the scp command. An easy way of retrieving a copy of a remote file into the current directory follow.

- To copy files from remote to local system, use the command:
  [root@deep]# **su username**
  [username@deep]$ **scp1 -p <login_name@hostname>:/dir/for/file localdir/to/filelocation**
  For example:
  [username@deep]$ scp1 -p username@mail:/etc/test1 /tmp
  Enter passphrase for RSA key 'username@mail.openarch.com':
  test1 | 2 KB | 2.0 kB/s | ETA: 00:00:00 | 100%

- To copy files from local to remote system, use the command:
  [root@deep]# **su username**
  [username@deep]$ **scp1 -p localdir/to/filelocation <username@hostname>:/dir/for/file**
  For example:
  [username@deep]$ scp1 -p /usr/bin/test2 username@mail:/var/tmp
  username@mail's password:
  test2 | 7 KB | 7.9 kB/s | ETA: 00:00:00 | 100%

NOTE: The "-p" option indicates that the modification and access times as well as modes of the source file should be preserved on the copy. This is usually desirable.

Installed files

> /etc/ssh
> /etc/ssh/ssh_host_key
> /etc/ssh/ssh_host_key.pub
> /etc/ssh/ssh_config
> /etc/ssh/sshd_config
> /root/.ssh
> /root/.ssh/random_seed
> /usr/bin/ssh1
> /usr/bin/ssh
> /usr/bin/slogin
> /usr/bin/ssh-keygen1
> /usr/bin/ssh-keygen
> /usr/bin/ssh-agent1
> /usr/bin/ssh-agent
> /usr/bin/ssh-add1
> /usr/bin/ssh-add
> /usr/bin/scp1
> /usr/bin/scp
> /usr/bin/make-ssh-known-hosts1
> /usr/bin/make-ssh-known-hosts
> /usr/man/man1/scp1.1
> /usr/man/man1/ssh-keygen1.1
> /usr/man/man1/ssh-keygen.1
> /usr/man/man1/ssh-agent1.1
> /usr/man/man1/ssh-agent.1
> /usr/man/man1/ssh-add1.1
> /usr/man/man1/ssh-add.1
> /usr/man/man1/scp.1
> /usr/man/man1/slogin1.1
> /usr/man/man1/slogin.1
> /usr/man/man1/ssh1.1
> /usr/man/man1/ssh.1
> /usr/man/man1/make-ssh-known-hosts1.1
> /usr/man/man1/make-ssh-known-hosts.1
> /usr/man/man8/sshd1.8
> /usr/man/man8/sshd.8
> /usr/sbin/sshd1
> /usr/sbin/sshd

Free ssh clients for Windows

Putty

PuTTY (originally STel until it stopped being just Telnet) is a free implementation of Telnet and SSH for Win32 platforms (Win95 and WinNT have been tested; Win98 and even Win2000 are reported to work fine).

Packages

Putty Homepage: http://www.chiark.greenend.org.uk/~sgtatham/putty.html

Tera Term Pro and TTSSH

TTSSH is a free SSH client for Windows. It is implemented as an extension DLL for Teraterm Pro. Teraterm Pro is a superb free terminal emulator/telnet client for Windows, and its source is

available. TTSSH adds SSH capabilities to Teraterm Pro without sacrificing any of Teraterm's existing functionality. TTSSH is also free to download and use and its source is available too.

Packages
Tera Term Pro Homepage: http://hp.vector.co.jp/authors/VA002416/teraterm.html
TTSSH Homepage: http://www.zip.com.au/~roca/download.html

Linux SSH2 Client/Server

Overview

This is the SSH2 commercial version. We provide the configuration step of this software for people that still use it. In our configuration we are configured sshd2 to support tcp-wrappers (inetd super server) for security reason.

These installation instructions assume

Commands are Unix-compatible.
The source path is "/var/tmp" (other paths are possible).
Installations were tested on RedHat Linux 6.1.
All steps in the installation will happen in superuser account "root".
Ssh2 version number is 2.0.13

Packages

SSH2 Homepage: http://www.ssh.fi/
You must be sure to download: ssh-2.0.13.tar.gz

Tarballs

It is a good idea to make a list of files on the system before you install ssh2, and one afterwards, and then compare them using '**diff**' to find out what file it placed where. Simply run '**find /* > ssh1**' before and '**find /* > ssh2**' after you install the software, and use '**diff ssh1 ssh2 > ssh**' to get a list of what changed.

Compilation

Decompress the tarball (tar.gz).

```
[root@deep]# cp ssh-version.tar.gz /var/tmp
[root@deep]# cd /var/tmp
[root@deep]# tar xzpf ssh-version.tar.gz
```

Compile and Optimize

Cd into the new Ssh2 directory and type the following commands on your terminal:

```
CC="egcs" \
CFLAGS="-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions" \
./configure \
--prefix=/usr \
--without-ssh-agent1-compat \
--disable-suid-ssh-signer \
```

```
--disable-tcp-port-forwarding \
--disable-X11-forwarding \
--enable-tcp-nodelay \
--with-libwrap
```

This tells SSH2 to set itself up for this particular hardware setup with:

- Leave out ssh-agent1 compatibility.
- Install ssh-signer without suid bit.
- Disable port forwarding support.
- Disable X11 forwarding support.
- Enable TCP_NODELAY socket option.
- Compile in libwrap (tcp_wrappers) support.

```
[root@deep]# make clean
[root@deep]# make
[root@deep]# make install
[root@deep]# rm -f /usr/bin/ssh-askpass
```

The "**make clean**", command erase all previous traces of a compilation so as to avoid any mistakes, then "**make**" command compile all source files into executable binaries, and finally "**make install**" command install the binaries and any supporting files into the appropriate locations.

Cleanup after work

```
[root@deep]# cd /var/tmp
[root@deep]# rm -rf ssh2-version/ ssh-version.tar.gz
```

The "rm" command will remove all the source files we have used to compile and install SSH2. It will also remove the SSH2 compressed archive from the "/var/tmp" directory.

Configurations

All software we describe in our book "Linuxsos.pdf" will have a specific directory and subdirectory in a tar compressed archive named "floppy.tgz" containing file configurations for the specific program. If you get this archive file, you wouldn't be obliged to reproduce the different configuration files bellow manually or cut and past them to create your configuration files. Whatever your decide to copy manually or get the files made to your convenience from the archive compressed files, it will be to your responsibility to modify, adjust for your needs and place the files related to SSH2 software to their appropriated places on your server machine, like show bellow. The server configuration files archive is located at the following Internet address: http://pages.infinit.net/lotus1/doc/opti/floppy.tgz

- To run SSH2 Client/Server, the following files are require and must be create or copied to their appropriated directories on your server.

  Copy the **sshd2_config** file to the "/etc/ssh2/" directory.
  Copy the **ssh2_config** file to the "/etc/ssh2/" directory.
  Copy the **ssh** file to the "/etc/pam.d/" directory.

You can obtain configuration files listed bellow on our floppy.tgz archive. Copy the following files from the decompressed floppy.tgz archive to their appropriated places or copy and paste them directly from this book to the concerned file.

Configure the "/etc/ssh2/ssh2_config" file

The configuration file for ssh2 ("/etc/ssh2/ssh2_config") allows you to set options that modify the operation of the client programs. The files contain keyword-value pairs, one per line, with keywords being case insensitive. Here are the more important keywords; a complete listing is available in the man page for ssh2 (1).

Edit the **ssh2_config** file (vi /etc/ssh2/ssh2_config) and add:

```
# ssh2_config
# SSH 2.0 Client Configuration File

*:
    Port                    22
    Ciphers                 AnyStdCipher
    Compression             yes
    IdentityFile            identification
    AuthorizationFile       authorization
    RandomSeedFile          random_seed
    VerboseMode             no
    ForwardAgent            no
    ForwardX11              no
    PasswordPrompt          "%U's password: "
    Ssh1Compatibility       no
    Ssh1AgentCompatibility  none
    NoDelay                 yes
    KeepAlive               yes
    QuietMode               no
```

This tells ssh2_config file to set itself up for this particular configuration setup with:

Port 22
This option "Port" specifies the port number that sshd2 listens on.

Ciphers AnyStdCipher
This option "Ciphers" specifies the ciphers to use for encrypting the session. AnyStd allows only standard ciphers.

Compression yes
This option "Compression" specifies whether to use compression.

IdentityFile identification
This option "IdentityFile" specifies the name of the user's identification file.

AuthorizationFile authorization
This option "AuthorizationFile" specifies the name of the user's authorization file.

RandomSeedFile random_seed
This option "RandomSeedFile" specifies the name of the user's randomseed file.

VerboseMode no
This option "VerboseMode" causes ssh2 to print debugging messages about its progress. This is helpful in debugging connection, authentication, and configuration problems.

ForwardAgent no
This option "ForwardAgent" specifies whether the connection to the authentication agent (if any) will be forwarded to the remote machine.

ForwardX11 no
This option “ForwardX11” specifies whether X11 connections will be automatically redirected over the secure channel and DISPLAY set.

PasswordPrompt "%U's password: "
This option “PasswordPrompt” sets the password prompt, that the user sees when connecting to a host. Variables '%U' and '%H' can be used to give the user's login name and host, respectively.

Ssh1Compatibility no
This option “Ssh1Compatibility” specifies whether to use SSH1 compatibility code.

Ssh1AgentCompatibility none
This option “Ssh1AgentCompatibility” specifies whether to forward also SSH1 agent connection.

NoDelay yes
This option “NoDelay” if "yes", enable socket option TCP_NODELAY. This will improve network performance.

KeepAlive yes
This option “KeepAlive” specifies whether the system should send keep alive messages to the other side. If they are sent, death of the connection or crash of one of the machines will be properly noticed.

QuietMode no
This option “QuietMode” causes all warnings and diagnostic messages to be suppressed. Only fatal errors are displayed.

Configure the “/etc/ssh2/sshd2_config” file

The configuration file for sshd2 (“/etc/ssh2/sshd2_config”) allows you to set options that modify the operation of the daemon. The files contain keyword-value pairs, one per line, with keywords being case insensitive. Here are the more important keywords; a complete listing is available in the man page for sshd2 (8).

Edit the **sshd2_config** file (vi /etc/ssh2/sshs2_config) and add:

```
# sshd2_config
# SSH 2.0 Server Configuration File

*:
    Port                    22
    ListenAddress           192.168.1.1
    Ciphers                 AnyStdCipher
    IdentityFile            identification
    AuthorizationFile       authorization
    HostKeyFile             hostkey
    PublicHostKeyFile       hostkey.pub
    RandomSeedFile          random_seed
    ForwardAgent            no
    ForwardX11              no
    PasswordGuesses         3
    MaxConnections          5
    PermitRootLogin         no
    AllowedAuthentications  publickey,password
    RequiredAuthentications publickey,password
    VerboseMode             no
    PrintMotd               yes
    CheckMail               yes
```

```
    UserConfigDirectory         "%D/.ssh2"
    SyslogFacility              AUTH
    Ssh1Compatibility           no
    NoDelay                     yes
    KeepAlive                   yes
    UserKnownHosts              yes
    AllowHosts                  192.168.1.4
    DenyHosts                   *
    QuietMode                   no

# subsystem definitions

    subsystem-sftp              sftp-server
```

This tells sshd2_config file to set itself up for this particular configuration setup with:

Port 22
This option “Port” specifies the port number that sshd2 listens on.

ListenAddress 192.168.1.1
This option “ListenAddress” specifies the ip address of the interface where the sshd2 server socket is bind.

Ciphers AnyStdCipher
This option “Ciphers” specifies the ciphers to use for encrypting the session. AnyStd allows only standard ciphers.

IdentityFile identification
This option “IdentityFile” specifies the name of the user's identification file.

AuthorizationFile authorization
This option “AuthorizationFile” specifies the name of the user's authorization file.

HostKeyFile hostkey
This option “HostKeyFile” specifies the file containing the private host key (default /etc/ssh2/hostkey).

PublicHostKeyFile hostkey.pub
This option “PublicHostKeyFile” specifies the file containing the public host key (default /etc/ssh2/hostkey.pub).

RandomSeedFile random_seed
This option “RandomSeedFile” specifies the name of the user's randomseed file.

ForwardAgent no
This option “ForwardAgent” specifies whether the connection to the authentication agent (if any) will be forwarded to the remote machine.

ForwardX11 no
This option “ForwardX11” specifies whether X11 connections will be automatically redirected over the secure channel and DISPLAY set.

PasswordGuesses 3
This option “PasswordGuesses” specifies the number of tries that the user has when using password authentication.

MaxConnections 5
This option “MaxConnections” specifies the maximum number of connections sshd2 will handle simultaneously. This is useful in systems where spamming sshd2 with new connections can cause the system to become unstable or crash.

PermitRootLogin no
This option “PermitRootLogin” specifies whether the root can log in using ssh2. Never say yes to this option.

AllowedAuthentications publickey,password
This option “AllowedAuthentications” specifies the authentications methods, that are allowed to use. This is comma-separated list consisting of (currently) words password, public key and host based. Each specifies an authentication method. Default is "password,publickey". With RequiredAuthentications, sysadmin can force users to complete several authentications before they are considered authenticated.

RequiredAuthentications publickey,password
This option “RequiredAuthentications” related to “AllowedAuthentications”, this is used to specify what authentication methods the users must complete before continuing. This parameter has no default. Note: This parameter has to be a subset for “AllowedAuthentications”. Otherwise, the server denies connection every time.

VerboseMode no
This option “VerboseMode” causes sshd2 to print debugging messages about its progress. This is helpful in debugging connection, authentication, and configuration problems.

PrintMotd yes
This option “PrintMotd” specifies whether sshd2 should print “/etc/motd” when a user logs in interactively.

CheckMail yes
This option “CheckMail” specifies whether sshd should print information whether you have new mail or not when a user logs in interactively.

UserConfigDirectory "%D/.ssh2"
This option “UserConfigDirectory” specifies where user-specific configuration data should be fetched from. With this the administration can control whatever configuration parameters they wish that are normally the users' domain. This is given as a pattern string, which is expanded by sshd2. %D is the user's home directory, %U is user's login name, %IU is the user's user ID (uid) and %IG is his group ID (gid).

SyslogFacility AUTH
This option “SyslogFacility” gives the facility code that is used when logging messages from sshd2.

Ssh1Compatibility no
This option “Ssh1Compatibility” specifies whether to use SSH1 compatibility code.

NoDelay yes
This option “NoDelay” if "yes", enable socket option TCP_NODELAY. This will improve network performance.

KeepAlive yes
This option “KeepAlive” specifies whether the system should send keep alive messages to the other side. If they are sent, death of the connection or crash of one of the machines will be properly noticed.

UserKnownHosts yes
This option “UserKnownHosts” specifies whether user's $HOME/.ssh2/knownhosts/directory can be used to fetch hosts public keys when using "hostbased"-authentication.

AllowHosts 192.168.1.4
This option “AllowHosts” can be followed by any number of host name patterns, separated by spaces. If specified, login is allowed only from hosts whose name matches one of the patterns. ´*´ and ´?´ can be used as wildcards in the patterns. Normal name servers are used to map the client's host into a canonical host name. If the name cannot be mapped, its IP-address is used as the host name.

*DenyHosts *
This option “DenyHosts” can be followed by any number of host name patterns, separated by spaces. If specified, login is disallowed from the hosts whose name matches any of the patterns.

QuietMode no
This option “QuietMode” causes all warnings and diagnostic messages to be suppressed. Only fatal errors are displayed.

Configure sshd2 to use tcp-wrappers inetd super server

Tcp-wrappers take cares to start and stop sshd2 server. Upon execution, inetd reads its configuration information from a configuration file which, by default, is “/etc/inetd.conf”. There must be an entry for each field of the configuration file, with entries for each field separated by a tab or a space.

Step 1
Edit the **inetd.conf** file (vi /etc/inetd.conf) and add the line:

```
ssh     stream  tcp     nowait  root    /usr/sbin/tcpd  sshd -i
```

NOTE: The -i parameter is important since is specifies that sshd is being run from inetd. Also, update your “inetd.conf” file by sending a SIGHUP signal (killall -HUP inetd) after adding the line.

```
[root@deep /root]# killall -HUP inetd
```

Step 2
Edit the **hosts.allow** file (vi /etc/hosts.allow) and add the line:

```
sshd: 192.168.1.4 win.openarch.com
```

Which mean client “192.168.1.4” with host name “win.openarch.com” is allowed to ssh on the server.

These "daemon" strings (for tcp-wrappers) are in use by sshd2:
sshd, sshd2 (The name sshd2 was called with (usually "sshd")).
sshdfwd-X11 (if you want to allow/deny X11-forwarding).
sshdfwd-<port-number> (for tcp-forwarding).
sshdfwd-<port-name> (port-name defined in /etc/services. Used in tcp-forwarding).

NOTE: If you do decide to switch to using ssh, make sure you install and use it on **all** your servers. Having ten secure servers and one insecure is a waste of time.

Configuration of the "/etc/pam.d/ssh" file

Configure your "/etc/pam.d/ssh" file to use pam authentication.

Create the **ssh** file (touch /etc/pam.d/ssh) and add:

```
#%PAM-1.0
auth        required    /lib/security/pam_pwdb.so shadow
auth        required    /lib/security/pam_nologin.so
account     required    /lib/security/pam_pwdb.so
password    required    /lib/security/pam_cracklib.so
password    required    /lib/security/pam_pwdb.so use_authtok nullok md5 shadow
session     required    /lib/security/pam_pwdb.so
```

Further documentation

For more details, there are several man pages you can read:

```
$ man ssh-add2 (1)        - adds identities for the authentication agent
$ man ssh-agent2 (1)      - authentication agent
$ man ssh-keygen2 (1)     - authentication key pair generation
$ man ssh2 (1)            - secure shell client (remote login program)
$ man sshd2 (8)           - secure shell daemon
```

Ssh2 Per-User Configuration

Step 1

Create your private & public keys of local, by executing:
[root@deep]# **su username**
[username@deep]$ **ssh-keygen2**

Step 2

Create an "identification" file in your ".ssh2" directory on local:
[username@deep]$ **cd ~/.ssh2**
[username@deep]$ **echo "IdKey id_dsa_1024_a" > identification**

NOTE: It's optional to create an identification file on Remote.

Step 3

Copy your public key of Local (**id_dsa_1024_a.pub**) to ".ssh2" directory of remote under the name, say, "**Local.pub**".

Step 4

Create an "**authorization**" file in your ".ssh2" directory on Remote:
[username@deep]$ **touch authorization**

Step 5

Add the following one line to "authorization":
[username@deep]$ **vi authorization**
key Local.pub

SSH2 Users Tools

The commands listed bellow are some that we use often in our regular use but much more exist and you must check the man page and documentation for more details and information.

ssh2
Ssh2 (Secure Shell) is a program for logging into a remote machine and executing commands in a remote machine. It is intended to replace rlogin and rsh, and provide secure encrypted communications between two untrusted hosts over an insecure network. X11 connections and arbitrary TCP/IP ports can also be forwarded over the secure channel.

- To logging to a remote machine, use the command:
  [root@deep]# **ssh2 -l <login_name> <hostname>**
  For example:
  [root@deep]# **ssh2 -l username www.openarch.com**
  Passphrase for key "/home/username/.ssh2/id_dsa_1024_a" with comment "1024-bit dsa, username@deep.openarch.com, Tue Oct 19 1999 14:31:40 -0400":
  username's password:
  Last login: Tue Oct 19 1999 18:13:00 -0400 from gate.openarch.com
  Welcome to www.openarch.com on Deepforest.

Where <login_name> is the name you use to connect to ssh server and <hostname> is the address of your ssh server.

sftp2
sftp (Secure File Transfer) is a ftp-like client that can be used in file transfer over the network. sftp uses Ssh2 in data connections, so the file transport is secure. You must already be connected with ssh2 before use sftp2.

- To ftp over ssh2, use the following command:
  [username@deep]$ **sftp2 <hostname>**
  For example:
  [username@deep]$ **sftp2 www.openarch.com**
  local path : /home/username
  Passphrase for key "/home/username/.ssh2/id_dsa_1024_a" with comment "1024-bit dsa, username@deep.openarch.com, Tue Oct 19 1999 14:31:40 -0400":
  username's password:
  username's password:
  remote path : /home/username
  sftp>

Where <hostname> is the name of the server you wan to sftp.

Installed files

> /etc/pam.d/ssh
> /etc/ssh2
> /etc/ssh2/hostkey
> /etc/ssh2/hostkey.pub
> /etc/ssh2/sshd2_config
> /etc/ssh2/ssh2_config
> /root/.ssh2
> /root/.ssh2/random_seed
> /root/ssh2
> /usr/man/man1/ssh2.1
> /usr/man/man1/ssh-keygen2.1
> /usr/man/man1/ssh-add2.1
> /usr/man/man1/ssh-agent2.1
> /usr/man/man1/scp2.1
> /usr/man/man1/sftp2.1
> /usr/man/man8/sshd2.8
> /usr/man/man8/sshd.8
> /usr/bin/ssh2
> /usr/bin/scp2
> /usr/bin/sftp2
> /usr/bin/sftp-server2
> /usr/bin/ssh-agent2
> /usr/bin/ssh-keygen2
> /usr/bin/ssh-signer2
> /usr/bin/ssh-add2
> /usr/bin/ssh
> /usr/bin/ssh-agent
> /usr/bin/ssh-add
> /usr/bin/ssh-askpass
> /usr/bin/ssh-keygen

> /usr/man/man1/ssh.1
> /usr/man/man1/ssh-add.1
> /usr/man/man1/ssh-agent.1
> /usr/man/man1/ssh-keygen.1
> /usr/man/man1/scp.1
> /usr/man/man1/sftp.1
> /usr/bin/scp
> /usr/bin/sftp
> /usr/bin/sftp-server
> /usr/bin/ssh-signer
> /usr/sbin/sshd2
> /usr/sbin/sshd

Linux Tripwire 2.2.1

Overview

Tripwire's initial version, release 1.0, was originally designed in 1992. Tripwire software has been completely rewritten as of version 2.0. The rewritten code is not open source. Policies are no longer in the configuration file, and the policy language has also changed. The base 64 notation is also different. Tripwire 2.0 and later versions use four cryptographic signatures; the ASR offered eight signatures, a "no signature" option, and the ability to add a custom signature. Many of the ASR signatures are weak by current standards.

According to the official Tripwire site. Tripwire works at the most fundamental layer, protecting the servers and workstations that make up the corporate network. Tripwire works by first scanning a computer and creating a database of system files, a compact digital "snapshot" of the system in a known secure state. The user can configure Tripwire very precisely, specifying individual files and directories on each machine to monitor, or creating a standard template that can be used on all machines in an enterprise.

Once this baseline database is created, a system administrator can use Tripwire to check the integrity of a system at any time. By scanning the current system and comparing that information with the data stored in the database, Tripwire detects and reports any additions, deletions, or changes to the system outside of the specified boundaries. If these changes are valid, the administrator can update the baseline database with the new information. If malicious changes are found, the system administrator will instantly know which parts of which components of the network have been affected.

This version of Tripwire has significant product enhancements over previous versions of Tripwire. Some of the enhancements include:

- Multiple levels of reporting allow you to choose different levels of report detail.
- Syslog option sends information about database initialization, database update, policy update and integrity check to the syslog.
- Database performance has been optimized to increase the efficiency of integrity checks.
- Individual email recipients can be sent certain sections of a report.
- SMTP email reporting support.
- Email test mode enables you to verify that the email settings are correct.
- Ability to create multiple sections within a policy file to be executed separately.

These installation instructions assume

Commands are Unix-compatible.
The source path is "/var/tmp" (other paths are possible).
Installations were tested on RedHat Linux 6.1.
All steps in the installation will happen in superuser account "root".
Tripwire version number is 2.2.1

Packages

Tripwire Homepage: http://www.tripwiresecurity.com/
You must be sure to download: Tripwire_221_for_Linux_x86_tar.gz

Compilation Tripwire-2.2.1

Decompress the tarball (tar.gz).

```
[root@deep]# cp Tripwire_version_for_Linux_x86_tar.gz /var/tmp
[root@deep]# cd /var/tmp
[root@deep]# tar xzpf Tripwire_version_for_Linux_x86_tar.gz
```

NOTE: After the decompression of Tripwire you will see the following files in your "/var/tmp" directory related to Tripwire software: License.txt, README, Release_Notes, install.cfg, install.sh, the pkg directory and the Tripwire tar.gz file Tripwire_version_for_Linux_x86_tar.gz.

Compile and Optimize

During the installation procedure, you will:

1. Specify configuration options for default operation.
2. Plan for two passphrases to be assigned for your site and local keys.
3. Run the installation script.
4. Edit the configuration and policy files.
5. Troubleshoot configuration and policy file setup, as needed.
6. Initialize the Tripwire software database.

Configuration of the "/var/tmp/install.cfg" file

Recall that Tripwire version 2.2.1 is not open source, so you cannot compile and install it like other archives source files, instead you must modify the "install.cfg" file of tripwire (that will install automatically Tripwire software for you) to specify installation paths for your system. We must modify this file to be compliant with Red Hat file system structure and install Tripwire binaries under our PATH ENVIRONMENT VARIABLE.

Step 1
Edit the **install.cfg** file (vi /var/tmp/install.cfg) and change this file to look like:

```
#
# install.cfg
#
# default install.cfg for:
# Tripwire(R) 2.2.1 for Unix
#
# NOTE:  This is a Bourne shell script that stores installation
#       parameters for your installation.  The installer will
#       execute this file to generate your config file and also to
#       locate any special configuration needs for your install.
#       Protect this file, because it is possible for
#       malicious code to be inserted here
#
# To set your Root directory for install, set TWROOT= to something
# other than /usr/TSS as necessary.
#
#=====================================================

# If CLOBBER is true, then existing files are overwritten.
```

```
# If CLOBBER is false, existing files are not overwritten.
CLOBBER=false

# The root of the TSS directory tree.
TWROOT="/usr"

# Tripwire binaries are stored in TWBIN.
TWBIN="${TWROOT}/bin"

# Tripwire policy files are stored in TWPOLICY.
TWPOLICY="${TWROOT}/TSS/policy"

# Tripwire manual pages are stored in TWMAN.
TWMAN="${TWROOT}/man"

# Tripwire database files are stored in TWDB.
TWDB="${TWROOT}/TSS/db"

# The Tripwire site key files are stored in TWSITEKEYDIR.
TWSITEKEYDIR="${TWROOT}/TSS/key"

# The Tripwire local key files are stored in TWLOCALKEYDIR.
TWLOCALKEYDIR="${TWROOT}/TSS/key"

# Tripwire report files are stored in TWREPORT.
TWREPORT="${TWROOT}/TSS/report"

# This sets the default text editor for Tripwire.
TWEDITOR="/bin/vi"

# TWLATEPROMTING controls the point when tripwire asks for a password.
TWLATEPROMPTING=false

# TWLOOSEDIRCHK selects whether the directory should be monitored for
# properties that change when files in the directory are monitored.
TWLOOSEDIRCHK=false

# TWMAILNOVIOLATIONS determines whether Tripwire sends a no violation
# report when integrity check is run with --email-report but no rule
# violations are found.  This lets the admin know that the integrity
# was run, as opposed to having failed for some reason.
TWMAILNOVIOLATIONS=true

# TWEMAILREPORTLEVEL determines the verbosity of e-mail reports.
TWEMAILREPORTLEVEL=3

# TWREPORTLEVEL determines the verbosity of report printouts.
TWREPORTLEVEL=3

# TWSYSLOG determines whether Tripwire will log events to the system log
TWSYSLOG=false

########################################
# Mail Options - Choose the appropriate
# method and comment the other section
########################################

########################################
# SENDMAIL options - DEFAULT
#
# Either SENDMAIL or SMTP can be used to send reports via TWMAILMETHOD.
# Specifies which sendmail program to use.
```

```
#####################################

TWMAILMETHOD=SENDMAIL
TWMAILPROGRAM="/usr/lib/sendmail -oi -t"

#####################################
# SMTP options
#
# TWSMTPHOST selects the SMTP host to be used to send reports.
# SMTPPORT selects the SMTP port for the SMTP mail program to use.
#####################################

# TWMAILMETHOD=SMTP
# TWSMTPHOST="mail.domain.com"
# TWSMTPPORT=25

###############################################################################
# Copyright (C) 1998-2000 Tripwire (R) Security Systems, Inc. Tripwire (R) is a
# registered trademark of the Purdue Research Foundation and is licensed
# exclusively to Tripwire (R) Security Systems, Inc.
###############################################################################
```

NOTE: The file "install.cfg" is a Bourne shell script used by the installer to set configuration variables. These variables specify the target directories where the installer will copy files and what the installer should do if the installation process would overwrite existing Tripwire software files.

Step 2
Now we must run the installation script to install Tripwire binaries and related files in our system according to whether you are using default or custom configuration values.

- To run the installtion script to install Tripwire, use the following command:
  [root@deep tmp]# **./install.sh**

NOTE: The "install.sh" file is the installation script, which you run to begin installation.

Step 3
When Tripwire is installed in our system it will copy "License.txt", "README", and "Release_Notes" files under "/usr" directory. Of course after finishing reading those files you can safety remove them from you "/usr" directory with the following command:

- To remove those files from your system, use the following command:
  [root@deep tmp]# **rm -f /usr/Lincense.txt README Release_Notes**

Cleanup after work
[root@deep]# cd /var/tmp
[root@deep]# rm -rf License.txt README Release-Notes install.cfg install.sh pkg/
Tripwire_version_for_Linux_x86_tar.gz

The "**rm**" command will remove all related files and directory we have used to install Tripwire for Linux. It will also remove the Tripwire for Linux compressed archive from the "/var/tmp" directory.

Configurations

All software we describe in our book "Linuxsos.pdf" will have a specific directory and subdirectory in a tar compressed archive named “floppy.tgz” containing file configurations for the specific program. If you get this archive file, you wouldn’t be obliged to reproduce the different configuration files bellow manually or cut and past them to create your configuration files. Whatever your decide to copy manually or get the files make to your convenience from the archive compressed files, it will be to your responsibility to modify, adjust for your needs and place the files related to Tripwire software to their appropriated places on your server machine, like show bellow. The server configuration files archive is located at the following Internet address: http://pages.infinit.net/lotus1/doc/opti/floppy.tgz

- To run Tripwire for Linux, the following files are require and must be create or copied to their appropriated directories on your server.

  Copy the **twpol.txt** file to the “/usr/TSS/policy” directory.

You can obtain configuration files listed bellow on our floppy.tgz archive. Copy the following files from the decompressed floppy.tgz archive to their appropriated places or copy and paste them directly from this book to the concerned file.

Configuration of the “/usr/TSS/policy/twpol.txt” file

The “/usr/TSS/policy/twpol.txt” is a text policy file that specifies what files and directories, called system objects, to check. A rule specifies how to check the object you want to monitor, and a property is what specifies how to check. A property mask specifies individual properties of a file to examine during integrity checks. Attributes help refine how groups of rules work. Take a note that editing the policy file can take several iterations as you establish by experience what information you want to appear in the Tripwire reports.

Step1
You must modify the default policy file, or create your own. The “policyguide.txt” file under “/usr/TSS/policy” directory may be helpful. Open the policy file “twpol.txt” with a text editor (vi /usr/TSS/policy/twpol.txt) and change it to look like:

```
@@section GLOBAL
TWROOT="/usr";
TWBIN="/usr/bin";
TWPOL="/usr/TSS/policy";
TWDB="/usr/TSS/db";
TWSKEY="/usr/TSS/key";
TWLKEY="/usr/TSS/key";
TWREPORT="/usr/TSS/report";
HOSTNAME=deep.openarch.com;

@@section FS
SEC_CRIT      = $(IgnoreNone)-SHa;  # Critical files - we can't afford to miss any changes.
SEC_SUID      = $(IgnoreNone)-SHa;  # Binaries with the SUID or SGID flags set.
SEC_TCB       = $(ReadOnly);        # Members of the Trusted Computing Base.
SEC_BIN       = $(ReadOnly);        # Binaries that shouldn't change
SEC_CONFIG    = $(Dynamic);         # Config files that are changed infrequently but accessed often.
SEC_LOG       = $(Growing);         # Files that grow, but that should never change ownership.
SEC_INVARIANT = +pug;               # Directories that should never change permission or ownership.
SIG_LOW       = 33;                 # Non-critical files that are of minimal security impact
SIG_MED       = 66;                 # Non-critical files that are of significant security impact
SIG_HI        = 100;                # Critical files that are significant points of vulnerability


# Tripwire Binaries
```

```
(emailto = admin@openarch.com, rulename = "Tripwire Binaries", severity = $(SIG_HI))
{
  $(TWBIN)/siggen         -> $(ReadOnly);
  $(TWBIN)/tripwire       -> $(ReadOnly);
  $(TWBIN)/twadmin        -> $(ReadOnly);
  $(TWBIN)/twprint        -> $(ReadOnly);
}

# Tripwire Data Files - Configuration Files, Policy Files, Keys, Reports, Databases
(emailto = admin@openarch.com, rulename = "Tripwire Data Files", severity = $(SIG_HI))
{
  # NOTE: Removing the inode attribute because when Tripwire creates a backup
  # it does so by renaming the old file and creating a new one (which will
  # have a new inode number).  Leaving inode turned on for keys, which shouldn't
  # ever change.

  # NOTE: this rule will trigger on the first integrity check after database
  # initialization, and each integrity check afterward until a database update
  # is run, since the database file will not exist before that point.
  $(TWDB)                         -> $(Dynamic) -i;
  $(TWPOL)/tw.pol                 -> $(SEC_BIN) -i;
  $(TWBIN)/tw.cfg                 -> $(SEC_BIN) -i;
  $(TWLKEY)/$(HOSTNAME)-local.key   -> $(SEC_BIN) ;
  $(TWSKEY)/site.key              -> $(SEC_BIN) ;

  #don't scan the individual reports
  $(TWREPORT)                     -> $(Dynamic) (recurse=0);
}

# These files are critical to a correct system boot.
(emailto = admin@openarch.com, rulename = "Critical system boot files", severity = 100)
{
    /boot                         -> $(SEC_CRIT)  ;
    !/boot/System.map                       ;
    !/boot/module-info                     ;
}

# These files change the behavior of the root account
(emailto = admin@openarch.com, rulename = "Root config files", severity = 100)
{
    /root                         -> $(SEC_CRIT)   ;
    /root/.bash_history           -> $(SEC_LOG)    ;
}

# Commonly accessed directories that should remain static with regards to owner and group
(emailto = admin@openarch.com, rulename = "Invariant Directories", severity = $(SIG_MED))
{
  /             -> $(SEC_INVARIANT) (recurse = 0);
  /home         -> $(SEC_INVARIANT) (recurse = 0);
  /etc          -> $(SEC_INVARIANT) (recurse = 0);
  /chroot       -> $(SEC_INVARIANT) (recurse = 0);
  /cache        -> $(SEC_INVARIANT) (recurse = 0);
}

(emailto = admin@openarch.com, rulename = "Shell Binaries")
{
  /bin/bsh                        -> $(SEC_BIN);
  /bin/csh                        -> $(SEC_BIN);
  /bin/sh                         -> $(SEC_BIN);
}

# Rest of critical system binaries
```

```
(emailto = admin@openarch.com, rulename = "OS executables and libraries", severity = $(SIG_HI))
{
   /bin                          -> $(ReadOnly)  ;
   /lib                          -> $(ReadOnly)  ;
}

# Local files
(emailto = admin@openarch.com, rulename = "User binaries", severity = $(SIG_MED))
{
  /sbin                          -> $(SEC_BIN) (recurse = 1);
  /usr/sbin                      -> $(SEC_BIN) (recurse = 1);
  /usr/bin                       -> $(SEC_BIN) (recurse = 1);
}

# Temporary directories
(emailto = admin@openarch.com, rulename = "Temporary directories", recurse = false, severity =
$(SIG_LOW))
{
  /usr/tmp                       -> $(SEC_INVARIANT);
  /var/tmp                       -> $(SEC_INVARIANT);
  /tmp                           -> $(SEC_INVARIANT);
}

# Libraries
(emailto = admin@openarch.com, rulename = "Libraries", severity = $(SIG_MED))
{
  /usr/lib -> $(SEC_BIN);
}

# Include
(emailto = admin@openarch.com, rulename = "OS Development Files", severity = $(SIG_MED))
{
  /usr/include -> $(SEC_BIN);
}

# Shared
(emailto = admin@openarch.com, rulename = "OS Shared Files", severity = $(SIG_MED))
{
  /usr/share -> $(SEC_BIN);
}

# Kernel headers files
(emailto = admin@openarch.com, rulename = "Kernel Headers Files", severity = $( SIG_HI))
{
  /usr/src/linux-2.2.14 -> $(SEC_BIN);
}

# setuid/setgid root programs
(emailto = admin@openarch.com, rulename = "setuid/setgid", severity = $(SIG_HI))
{
   /bin/su -> $(SEC_SUID);
   /sbin/pwdb_chkpwd -> $(SEC_SUID);
   /sbin/dump -> $(SEC_SUID);
   /sbin/restore -> $(SEC_SUID);
   /usr/bin/at -> $(SEC_SUID);
   /usr/bin/passwd -> $(SEC_SUID);
   /usr/bin/suidperl -> $(SEC_SUID);
   /usr/bin/crontab -> $(SEC_SUID);
   /usr/sbin/sendmail -> $(SEC_SUID);
   /usr/bin/man -> $(SEC_SUID);
   /usr/bin/sperl5.00503 -> $(SEC_SUID);
   /usr/bin/slocate -> $(SEC_SUID);
```

```
   /usr/sbin/utempter -> $(SEC_SUID);
   /sbin/netreport -> $(SEC_SUID);
}

(emailto = admin@openarch.com, rulename = "Configuration Files")
{
  /etc/hosts                    -> $(SEC_CONFIG);
  /etc/inetd.conf               -> $(SEC_CONFIG);
  /etc/initlog.conf             -> $(SEC_CONFIG);
  /etc/inittab                  -> $(SEC_CONFIG);
  /etc/resolv.conf              -> $(SEC_CONFIG);
  /etc/syslog.conf              -> $(SEC_CONFIG);
}

(emailto = admin@openarch.com, rulename = "Security Control")
{
  /etc/group                    -> $(SEC_CRIT);
  /etc/security/                -> $(SEC_CRIT);
  /lib/security/                -> $(SEC_CRIT);
  /var/spool/cron               -> $(SEC_CRIT);
}

(emailto = admin@openarch.com, rulename = "Login Scripts")
{
  /etc/csh.login                -> $(SEC_CONFIG);
  /etc/profile                  -> $(SEC_CONFIG);
}

# These files change every time the system boots
(emailto = admin@openarch.com, rulename = "System boot changes", severity = $(SIG_HI))
{
    /dev/log                    -> $(Dynamic)   ;
    /dev/cua0                   -> $(Dynamic)   ;
    /dev/console                -> $(Dynamic)   ;
    /dev/tty2                   -> $(Dynamic)   ; # tty devices
    /dev/tty3                   -> $(Dynamic)   ; # are extremely
    /dev/tty4                   -> $(Dynamic)   ; # variable
    /dev/tty5                   -> $(Dynamic)   ;
    /dev/tty6                   -> $(Dynamic)   ;
    /dev/urandom                -> $(Dynamic)   ;
    /dev/initctl                -> $(Dynamic)   ;
    /var/lock/subsys            -> $(Dynamic)   ;
    /var/run                    -> $(Dynamic)   ; # daemon PIDs
    /var/log                    -> $(Dynamic)   ;
    /etc/ioctl.save             -> $(Dynamic)   ;
    /etc/.pwd.lock              -> $(Dynamic)   ;
    /etc/mtab                   -> $(Dynamic)   ;
    /lib/modules                -> $(Dynamic)   ;
}

# Critical configuration files
(emailto = admin@openarch.com, rulename = "Critical configuration files", severity = $(SIG_HI))
{
    /etc/conf.modules           -> $(ReadOnly)   ;
    /etc/crontab                -> $(ReadOnly)   ;
    /etc/cron.hourly            -> $(ReadOnly)   ;
    /etc/cron.daily             -> $(ReadOnly)   ;
    /etc/cron.weekly            -> $(ReadOnly)   ;
    /etc/cron.monthly           -> $(ReadOnly)   ;
    /etc/default                -> $(ReadOnly)   ;
    /etc/fstab                  -> $(ReadOnly)   ;
    /etc/group-                 -> $(ReadOnly) ;  # changes should be infrequent
```

```
    /etc/host.conf              -> $(ReadOnly)  ;
    /etc/hosts.allow            -> $(ReadOnly)  ;
    /etc/hosts.deny             -> $(ReadOnly)  ;
    /etc/lilo.conf              -> $(ReadOnly)  ;
    /etc/logrotate.conf         -> $(ReadOnly)  ;
    /etc/pwdb.conf              -> $(ReadOnly)  ;
    /etc/securetty              -> $(ReadOnly)  ;
    /etc/sendmail.cf            -> $(ReadOnly)  ;
    /etc/protocols              -> $(ReadOnly)  ;
    /etc/services               -> $(ReadOnly)  ;
    /etc/rc.d/init.d            -> $(ReadOnly)  ;
    /etc/rc.d                   -> $(ReadOnly)  ;
    /etc/motd                   -> $(ReadOnly)  ;
    /etc/passwd                 -> $(ReadOnly)  ;
    /etc/passwd-                -> $(ReadOnly)  ;
    /etc/profile.d              -> $(ReadOnly)  ;
    /etc/rpc                    -> $(ReadOnly)  ;
    /etc/sysconfig              -> $(ReadOnly)  ;
    /etc/shells                 -> $(ReadOnly)  ;
    /etc/nsswitch.conf          -> $(ReadOnly)  ;
}

# Critical devices
(emailto = admin@openarch.com, rulename = "Critical devices", severity = $(SIG_HI), recurse = false)
{
    /dev/kmem                   -> $(Device)  ;
    /dev/mem                    -> $(Device)  ;
    /dev/null                   -> $(Device)  ;
    /dev/zero                   -> $(Device)  ;
    /proc/devices               -> $(Device)  ;
    /proc/net                   -> $(Device)  ;
    /proc/tty                   -> $(Device)  ;
    /proc/sys                   -> $(Device)  ;
    /proc/cpuinfo               -> $(Device)  ;
    /proc/modules               -> $(Device)  ;
    /proc/mounts                -> $(Device)  ;
    /proc/dma                   -> $(Device)  ;
    /proc/filesystems           -> $(Device)  ;
    /proc/ide                   -> $(Device)  ;
    /proc/interrupts            -> $(Device)  ;
    /proc/ioports               -> $(Device)  ;
    /proc/scsi                  -> $(Device)  ;
    /proc/kcore                 -> $(Device)  ;
    /proc/self                  -> $(Device)  ;
    /proc/kmsg                  -> $(Device)  ;
    /proc/stat                  -> $(Device)  ;
    /proc/ksyms                 -> $(Device)  ;
    /proc/loadavg               -> $(Device)  ;
    /proc/uptime                -> $(Device)  ;
    /proc/locks                 -> $(Device)  ;
    /proc/version               -> $(Device)  ;
    /proc/meminfo               -> $(Device)  ;
    /proc/cmdline               -> $(Device)  ;
    /proc/misc                  -> $(Device)  ;
}
```

NOTE: This is a standard policy file, of course you must modify this file to fit your system and your specific needs.

Step 2
When you are ready to use your policy file for the first time, install it with the following command:

```
[root@deep]# twadmin --create-polfile /usr/TSS/policy/twpol.txt
Please enter your site passphrase:
Wrote policy file: /usr/TSS/policy/tw.pol
```

Securing Tripwire for Linux

Security Issue

Make sure that the integrity of the system you are running has not been compromised. For maximum confidence in your baseline database, you should generate operating system and application files from original media to ensure that you get a clean baseline.

It is recommend to delete the plain text copy of Tripwire configuration file named “twcfg.txt” located under “/usr/bin” directory to hide the locations of Tripwire’s files and prevent anyone from creating a second configuration file.

- To delete the plain copy of tripwire Configuration file, use the following command:
  [root@deep]# **rm -f /usr/bin/twcfg.txt**

Further documentation

For more details, there are several man pages you can read:

```
siggen (8)      - signature gathering routine for Tripwire
tripwire (8)    - a file integrity checker for UNIX systems
twadmin (8)     - Tripwire administrative and utility tool
twconfig (4)    - Tripwire configuration file reference
twfiles (5)     - overview of files used by Tripwire and file backup process
twintro (8)     - introduction to Tripwire software
twpolicy (4)    - Tripwire policy file reference
twprint (8)     - Tripwire database and report printer
```

Commands

The commands listed bellow are some that we use often in our regular use but much more exist and you must check the man page for more details and information.

Creating the database for the first time

In Database Initialization mode, Tripwire software builds a database of filesystem objects, based on the rules in the policy file. This database will serve as the baseline for later integrity checks.

The syntax for Database Initialization mode is:
[root@deep]# tripwire { --init }

- To initialize your database file, use the following command:
  [root@deep]# **tripwire --init**
  Please enter your local passphrase:
  Parsing policy file: /usr/TSS/policy/tw.pol
  Generating the database...
  *** Processing Unix File System ***
  Wrote database file: /usr/TSS/db/deep.openarch.com.twd
  The database was successfully generated.

NOTE: When this command has executed, the database is ready and you can check system integrity and review the report file.

Running the integrity check
The Integrity Check mode compares the current file system objects with their properties as recorded in the Tripwire database. Violations will be printed to stdout; the report file will be saved and can later be accessed by twprint.

The syntax for integrity check mode is:
[root@deep]# tripwire { --check }

- To run the integrity check mode, use the command:
  [root@deep]# **tripwire --check**

You can also run Tripwire in Interactive Check mode. In interactive mode you can automatically update your changes via the terminal. With the adding of "--interactive" syntax, the resulting report is opened in an editor for database updates.

- To run in interactive check mode, use the command:
  [root@deep]# **tripwire --check --interactive**

An email option enables you to send email. Running Tripwire with the option [--email-report] specifies that reports be emailed to the recipients designated in the policy file, using the options in the default configuration file.

- To run the integrity check mode and send email to the recipient, use the command:
  [root@deep]# **tripwire --check --email-report**

Updating the database after an integrity check
Database Update mode enables you to update the Tripwire database after an integrity check, if the violations discovered are actually valid. This update process saves you time by enabling you to update the database without having to regenerate it; even more importantly it enables selective updating, which cannot be done through regeneration. Since in our configuration of tripwire, the configuration file specifies the report filename using time-based variables, the report will not be found in a regular update mode. This happens because the $(DATE) variable will have changed to reflect the current time. To solve this problem, the report file must be specified on the command line with the -r or --twrfile argument.

The syntax for database update mode is:
[root@deep]# tripwire { --update -r}

- To update the database, use the command:
  [root@deep]# **tripwire --update -r /usr/TSS/report/deep.openarch.com-200001-021854.twr**

Where "-r" read the specified report file (deep.openarch.com-200001-021854.twr). This option is required since the REPORTFILE variable in the current configuration file uses $(DATE).

NOTE: In Database Update mode or Interactive Check mode, Tripwire software displays the report with a ballot box next to each policy violation. You can approve a change to the file system by leaving the "x" next to each policy violation. If you remove the "x" from the ballot box, the database will not be updated with the new value(s) for that object. After you exit the editor and provide the local passphrase, Tripwire software will update and save the database.

Updating the policy file
You can change the rules in the policy file, which will change the way that Tripwire software scans the system, and update the database without requiring a complete re-initialization. This can save a significant amount of time; even more importantly, it preserves security by keeping the policy file synchronized with the database it uses.

The syntax for policy update mode is:
[root@deep]# tripwire { --update-policy /path/to/new/policy/file}

- To update the policy file, use the command:
  [root@deep]# **tripwire --update-policy /usr/TSS/policy/newtwpol.txt**

By default, Policy Update mode runs with "--secure-mode high". You may encounter errors when running in high security mode if the file system has changed since the last database update, and if the changes still cause a violation in the new policy. This may happen if another adminis-trator is modifying files during the policy update process, for example. To accommodate this situation, after determining that all of the violations reported in high security mode are authorized, you can update the policy file in low security mode:

- To update the policy file in low security mode, use the command:
  [root@deep]# **tripwire --update-policy --secure-mode low /usr/TSS/policy/newtwpol.txt**

Installed files

> /usr/TSS
> /usr/bin
> /usr/bin/siggen
> /usr/bin/twprint
> /usr/bin/twadmin
> /usr/bin/tripwire
> /usr/bin/twcfg.txt
> /usr/bin/tw.cfg
> /usr/TSS/policy
> /usr/TSS/policy/policyguide.txt
> /usr/TSS/policy/twpol.txt
> /usr/TSS/policy/tw.pol
> /usr/TSS/policy/twpol.txt.bak
> /usr/TSS/report
> /usr/TSS/db
> /usr/TSS/key
> /usr/TSS/key/site.key
> /usr/TSS/key/deep.openarch.com-local.key
> /usr/man
> /usr/man/man4
> /usr/man/man4/twconfig.4
> /usr/man/man4/twpolicy.4
> /usr/man/man5
> /usr/man/man5/twfiles.5
> /usr/man/man8
> /usr/man/man8/siggen.8
> /usr/man/man8/tripwire.8
> /usr/man/man8/twadmin.8
> /usr/man/man8/twintro.8
> /usr/man/man8/twprint.8
> /usr/README
> /usr/Release_Notes
> /usr/License.txt

Linux Tripwire ASR 1.3.1

Overview
With the advent of increasingly sophisticated and subtle account break-ins on Unix systems, the need for tools to aid in the detection of unauthorized modification of files becomes clear. Tripwire is a tool that aids system administrators and users in monitoring a designated set of files for any changes. Used with system files on a regular (e.g., daily) basis, Tripwire can notify system administrators of corrupted or tampered files, so damage control measures can be taken in a timely manner.

Tripwire is a file and directory integrity checker, a utility that compares a designated set of files and directories against information stored in a previously generated database. Any differences are flagged and logged, including added or deleted entries. When run against system files on a regular basis, any changes in critical system files will be spotted -- and appropriate damage control measures can be taken immediately. With Tripwire, system administrators can conclude with a high degree of certainty that a given set of files remain free of unauthorized modifications if Tripwire reports no changes.

These installation instructions assume

Commands are Unix-compatible.
The source path is "/var/tmp" (other paths are possible).
Installations were tested on RedHat Linux 6.1.
All steps in the installation will happen in superuser account "root".
Tripwire version number is 1.3.1-1

Packages

Tripwire Homepage: http://www.tripwiresecurity.com/
You must be sure to download: Tripwire-1.3.1-1.tar.gz

Tarballs

It is a good idea to make a list of files on the system before you install it, and one afterwards, and then compare them using '**diff**' to find out what file it placed where. Simply run '**find /* > trip1**' before and '**find /* > trip2**' after you install the tarball, and use '**diff trip1 trip2 > trip**' to get a list of what changed.

Compilation Tripwire-1.3.1-1

Decompress the tarball (tar.gz).

```
[root@deep]# cp Tripwire-version.tar.gz /var/tmp
[root@deep]# cd /var/tmp
[root@deep]# tar xzpf Tripwire-version.tar.gz
```

Compile and Optimize

Cd into the new Tripwire directory and type the following on your terminal:

Edit the **utils.c** file (vi +462 src/utils.c) and change the line:

```
else if (iscntrl(*pcin)) {
To read:
else if (!(*pcin & 0x80) && iscntrl(*pcin)) {
```

Edit the **config.parse.c** file (vi +356 src/config.parse.c) and change the line:

```
rewind(fpout);
To read:
   else {
      rewind(fpin);
    }
```

Edit the **config.h** file (vi +106 include/config.h) and change the line:

#define CONFIG_PATH "/usr/local/bin/tw"
#define DATABASE_PATH "/var/tripwire"
To read:
#define CONFIG_PATH "/etc"
#define DATABASE_PATH "/var/spool/tripwire"

Edit the **config.h** file (vi +165 include/config.h) and change the line:

#define TEMPFILE_TEMPLATE "/tmp/twzXXXXXX"
To read:
#define TEMPFILE_TEMPLATE "/var/tmp/.twzXXXXXX"

Edit the **config.pre.y** file (vi +66 src/config.pre.y) and change the line:

#ifdef TW_LINUX
To read:
#ifdef TW_LINUX_UNDEF

Edit the **Makefile** file (vi +13 Makefile) and change the line:

DESTDIR = /usr/local/bin/tw
To read:
DESTDIR = /usr/sbin

DATADIR = /var/tripwire
To read:
DATADIR = /var/spool/tripwire

LEX = lex
To read:
LEX = flex

CC=gcc
To read:
CC=egcs

CFLAGS = -O
To read:
CFLAGS = -O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions

```
[root@deep]# make
[root@deep]# make install

[root@deep]# chmod 700 /var/spool/tripwire/
[root@deep]# chmod 500 /usr/sbin/tripwire
[root@deep]# chmod 500 /usr/sbin/siggen
[root@deep]# rm -f /usr/sbin/tw.config
```

The above commands "**make**" and "**make install**" would configure the software to ensure your system has the necessary functionality and libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

The “**chmod**” command will change the default mode of “tripwire” directory to be 700 (drwx------) only readable, writable, and executable by the super-user “root”. It will make the binary “/usr/sbin/tripwire” only readable, and executable by the super-user “root” (-r-x------) and finally make the “siggen” program under “/usr/sbin” directory only executable and readable by “root”.

The “**rm**” command will remove the file “tw.config” under “/usr/sbin”. We don’t need this file since we will create a new one under “/etc” directory later.

Cleanup after work

```
[root@deep]# cd /var/tmp
[root@deep]# rm -rf tw_ASR_version/ Tripwire-version.tar.gz
```

The “**rm**” command will remove all the source files we have used to compile and install Tripwire. It will also remove the Tripwire compressed archive from the “/var/tmp” directory.

Configurations

All software we describe in our book "Linuxsos.pdf" will have a specific directory and subdirectory in a tar compressed archive named “floppy.tgz” containing file configurations for the specific program. If you get this archive file, you wouldn’t be obliged to reproduce the different configuration files bellow manually or cut and past them to create your configuration files. Whatever your decide to copy manually or get the files make to your convenience from the archive compressed files, it will be to your responsibility to modify, adjust for your needs and place the files related to Tripwire software to their appropriated places on your server machine, like show bellow. The server configuration files archive is located at the following Internet address: http://pages.infinit.net/lotus1/doc/opti/floppy.tgz

- To run Tripwire, the following files are require and must be create or copied to their appropriated directories on your server.

  Copy the **tw.config** file to the “/etc” directory.
  Copy the **tripwire.verify** script to the “/etc/cron.daily” directory.

You can obtain configuration files listed bellow on our floppy.tgz archive. Copy the following files from the decompressed floppy.tgz archive to their appropriated places or copy and paste them directly from this book to the concerned file.

Configuration of the “/etc/tw.config” file

Tripwire runs in either of four modes: Database Generation, Integrity Checking, Database Update, and Interactive Update mode. In order to run Integrity Checking, Tripwire must have a database to compare against. To do that, you must first specify the set of files for Tripwire to monitor. This list is stored in the “/etc/tw.config” file.The “tw.config” file define all the directories that contain files that you want monitories.

Step 1
Create the **tw.config** file (touch /etc/tw.config) and add in this file all the directories that contain files that you want monitored. The format of the config file is described in its header and in the man page tw.config (5):

```
# Gerhard Mourani: gmourani@videotron.ca
# last updated: 1999/11/12

#  First, root's "home"
/root                           R
!/root/.bash_history
```

```
/                       R

# OS itself
/boot/vmlinuz           R

# critical boot resources
/boot                   R

# Critical directories and files
/chroot                 R
/etc                    R
/etc/inetd.conf         R
/etc/nsswitch.conf      R
/etc/rc.d               R
/etc/mtab               L
/etc/motd               L
/etc/group              R
/etc/passwd             L

# other popular filesystems
/usr                    R
/usr/local              R
/dev                    L-am
/usr/etc                R

# truncate home
=/home                  R

# var tree
=/var/spool             L
/var/log                L
/var/lib                L
/var/spool/cron         L
!/var/lock

# unusual directories
=/proc                  E
=/tmp
=/mnt/cdrom
=/mnt/floppy
```

Step 2
Now, for security reason change the mode of this file to be 0600 with the following command:
[root@deep]# **chmod 600 /etc/tw.config**

Configuration of the “/etc/tripwire.verify” script

A common setup for running Tripwire would mail the system administrator any output that it generates. However, some files on your system may change during normal operation, and this necessitates update of the Tripwire database. The “tripwire.verify” file is a small script executed by the crond program each day to scan your hard disk for possible changed files or directories and mail the result to the system administrator.

Step 1
Create the **tripwire.verify** script (touch /etc/cron.daily/tripwire.verify) and add in this script:

```
#!/bin/sh
/usr/sbin/tripwire -loosedir -q | (cat <<EOF
This is an automated report of possible file integrity changes, generated by
```

the Tripwire integrity checker. To tell Tripwire that a file or entire directory tree is valid, as root run:

/usr/sbin/tripwire -update [pathname|entry]

If you wish to enter an interactive integrity checking and verification session, as root run:

/usr/sbin/tripwire -interactive

Changed files/directories include:
EOF
cat
) | /bin/mail -s "File integrity report" root

Step 2
Now, make this script executable and change the mode to be 0700 with the following command:
[root@deep]# **chmod 700 /etc/cron.daily/tripwire.verify**

Securing Tripwire

Security Issue

For added security, it is recommended that the database (tw.db_[hostname]) file of Tripwire must be moved someplace (e.g. floppy) where it cannot be modified. Because data from Tripwire is only as trustworthy as its database, choose this with care.

I also recommend that you make a hardcopy printout of the database contents right away. In the event that you become suspicious of the integrity of the database, you will be able to manually compare information against this hardcopy.

Further documentation

For more details, there are several man pages you can read:

siggen (8)	- signature generation routine for Tripwire
tripwire (8)	- a file integrity checker for UNIX systems
tw.config (5)	- configuration file for Tripwire

Commands

The commands listed bellow are some that we use often in our regular use but much more exist and you must check the man page for more details and information.

Running Tripwire in Interactive mode

Running Tripwire in interactive mode is similar to the Integrity Checking mode. However, when a file or directory is encountered that has been added, deleted, or changed from what was recorded in the database, Tripwire asks the user whether the database entry should be updated. While this mode may be the most convenient way of keeping your database up-to-date, it requires that the user be "at the keyboard."

Step 1
First of all, you must run Tripwire with “tripwire --initialize” command. This will create a file called “tw.db_[hostname]” in the directory you specified to hold your databases (where [hostname] will be replaced with your machine hostname). Recall in order to run Integrity Checking, Tripwire must have a database to compare against so we first create the file information database.

- To create file information database, use the command:
  [root@deep]# **cd /var/spool/tripwire/**
  [root@deep]# **/usr/sbin/tripwire --initialize**

We move to the directory we specified to hold our database, then we create the file information database, which is used for all subsequent Integrity Checking.

Step 2
There are now two ways to update your Tripwire database. The first method is interactive, where Tripwire prompts the user whether each changed entry should be updated to reflect the current state of the file, while the second method is a command-line driven mode where specific files/entries are specified at run-time.

- To use the interactive mode, use the command:
  [root@deep]# **cd /var/spool/tripwire/database/**
  [root@deep]# **cp tw.db_myserverhostname /var/spool/tripwire/**
  [root@deep]# **cd ..**
  [root@deep]# **/usr/sbin/tripwire --interactive**
  Tripwire(tm) ASR (Academic Source Release) 1.3.1
  File Integrity Assessment Software
  (c) 1992, Purdue Research Foundation, (c) 1997, 1999 Tripwire
  Security Systems, Inc. All Rights Reserved. Use Restricted to
  Authorized Licensees.
  ### Phase 1: Reading configuration file
  ### Phase 2: Generating file list
  ### Phase 3: Creating file information database
  ### Phase 4: Searching for inconsistencies
  ###
  ### Total files scanned: 15722
  ### Files added: 34
  ### Files deleted: 42
  ### Files changed: 321
  ###
  ### Total file violations: 397
  ###
  added: -rwx------ root 22706 Dec 31 06:25:02 1999 /root/tmp/firewall
  ---> File: '/root/tmp/firewall'
  ---> Update entry? [YN(y)nh?]

In interactive mode, Tripwire first reports all added, deleted, and changed files, then allows the user to update the entry in the database.

Running Tripwire in Database Update mode
Tripwire supports incremental updates of its database on a per-file/directory or “tw.config” entry basis. Tripwire stores information in the database so it can associate any file in the database with the “tw.config” entry that generated it when the database was created. Running Tripwire in database update mode mixed with the “tripwire.verify” script file that mail the result to the system administrator will reduce the time of scanning the system. Instead of running Tripwire in Interactive mode and waiting for the long scan to finish, the script file “tripwire.verify” will scan the system and report via mail the result, then you run Tripwire in database update mode and change only single file that has changed.

As an example:
Therefore, if a single file has changed, you can:

[root@deep]# **tripwire -update /etc/newly.installed.file**

Or, if an entire set of files that made up an entry in the “tw.config” file changed, you can:

```
[root@deep]# tripwire -update /usr/lib/Package_Dir
```

In either case, Tripwire regenerates the database entries for every specified file. A backup of the old database is created in the “./databases” directory.

Installed files

> /etc/cron.daily/tripwire.verify
> /etc/tw.config
> /usr/man/man5/tw.config.5
> /usr/man/man8/siggen.8
> /usr/man/man8/tripwire.8
> /usr/sbin/tripwire
> /usr/sbin/siggen
> /var/spool/tripwire
> /var/spool/tripwire/tw.db_TEST

Alternatives to Tripwire

ViperDB
ViperDB was created as a smaller and faster option to Tripwire. ViperDB does not use a fancy all in-one database to keep records. Instead it uses a plaintext db which is stored in each "watched" directory. By using this there is no real one attack point for an attacker to focus his attention on. This coupled with the running of ViperDB every 5 minutes (via cron root job) decreases that likelihood that an attacker will be able to modify your "watched" filesystem while ViperDB is monitoring your system.

Packages
ViperDB Homepage: http://www.resentment.org/projects/viperdb/

FCHECK
FCHECK is a very stable PERL script written to generate and comparatively monitor a UNIX system against its baseline for any file alterations and report them through syslog, console, or any log monitoring interface. Monitoring events can be done in as little as one minute intervals if a system's drive space is small enough, making it very difficult to circumvent. This is a freely available open-source alternative to 'tripwire' that is time tested, and is easier to configure and use.

Packages
FCHECK Homepage: http://sites.netscape.net/fcheck/fcheck.html

Sentinel
Sentinel is a fast file/drive scanning utility similar to the Tripwire and Viper.pl utilities available. It uses a database similar to Tripwire, but uses a RIPEMD-160bit MAC check summing algorithm (no patents) which is more secure than the patented MD5 128 bit checksum. It should run on most unixes.

Packages
FCHECK Homepage: http://zurk.netpedia.net/zfile.html

Linux GnuPG

Overview

GnuPG - The GNU Privacy Guard. GnuPG is GNU's tool for secure communication and data storage. It can be used to encrypt data and to create digital signatures. It includes an advanced key management facility and is compliant with the proposed OpenPGP Internet standard as described in RFC2440.

Because GnuPG does not use any patented algorithm it cannot be compatible with PGP2 versions. PGP 2.x uses only IDEA (which is patented worldwide) and RSA (which is patented in the United States until Sep 20, 2000).

These installation instructions assume

Commands are Unix-compatible.
The source path is "/var/tmp" (other paths are possible).
Installations were tested on RedHat Linux 6.1.
All steps in the installation will happen in superuser account "root".
GnuPG version number is 1.0.1

Packages

GnuPG Homepage: http://www.gnupg.org/
You must be sure to download: gnupg-1_0_1_tar.gz

Tarballs

It is a good idea to make a list of files on the system before you install it, and one afterwards, and then compare them using '**diff**' to find out what file it placed where. Simply run '**find /* > pg1**' before and '**find /* > pg2**' after you install the tarball, and use '**diff pg1 pg2 > pg**' to get a list of what changed.

Compilation

Decompress the tarball (tar.gz).

```
[root@deep]# cp gnupg-version.tar.gz /var/tmp
[root@deep]# cd /var/tmp
[root@deep]# tar xzpf gnupg-version.tar.gz
```

Compile and Optimize

Cd into the new GnuPG dir and type the following on your terminal:

```
CC="egcs" \
CFLAGS="-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions" \
./configure \
--prefix=/usr \
--enable-shared

[root@deep]# make
[root@deep]# make check
[root@deep]# make install
[root@deep]# strip /usr/bin/gpg
```

The "**make**" command compile all source files into executable binaries, then the "**make check**" will run any self-tests that come with the package" and finally the "**make install**" command install the binaries and any supporting files into the appropriate locations. The "**strip**" command will reduce the size of the "gpg" binary for better performance.

Cleanup after work
[root@deep]# cd /var/tmp
[root@deep]# rm -rf gnupg-version/ gnupg-version.tar.gz

The "**rm**" command will remove all the source files we have used to compile and install GnuPG. It will also remove the GnuPG compressed archive from the "/var/tmp" directory.

Commands

The commands listed bellow are some that we use often in our regular use but much more exist and you must check the man page for more details and information.

Creating a key
First of all, we must create a new key-pair for our self if this is a first use of GunPG software.

Step 1

- To create a new key-pair (as root), use the following command:
  [root@deep]# **gpg --gen-key**
  gpg (GnuPG) 1.0.1; Copyright (C) 1999 Free Software Foundation, Inc.
  This program comes with ABSOLUTELY NO WARRANTY.
  This is free software, and you are welcome to redistribute it
  under certain conditions. See the file COPYING for details.

  gpg: /root/.gnupg: directory created
  gpg: /root/.gnupg/options: new options file created
  gpg: you have to start GnuPG again, so it can read the new options file
  This asks some questions and then starts key generation.

Step 2

- We start GnuPG again (as root) with the following command:
  [root@deep]# **gpg --gen-key**
  gpg (GnuPG) 1.0.1; Copyright (C) 1999 Free Software Foundation, Inc.
  This program comes with ABSOLUTELY NO WARRANTY.
  This is free software, and you are welcome to redistribute it
  under certain conditions. See the file COPYING for details.

  gpg: /root/.gnupg/secring.gpg: keyring created
  gpg: /root/.gnupg/pubring.gpg: keyring created
  Please select what kind of key you want:
  (1) DSA and ElGamal (default)
  (2) DSA (sign only)
  (4) ElGamal (sign and encrypt)
  Your selection? **1**
  DSA keypair will have 1024 bits.
  About to generate a new ELG-E keypair.
  minimum keysize is 768 bits
  default keysize is 1024 bits
  highest suggested keysize is 2048 bits
  What keysize do you want? (1024) **2048**
  Do you really need such a large keysize? **y**
  Requested keysize is 2048 bits
  Please specify how long the key should be valid.

```
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
      <n>y = key expires in n years
Key is valid for? (0) 0
correct (y/n)? y

You need a User-ID to identify your key; the software constructs the user id
from Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Gerhard Mourani
Email address: gmourani@videotron.ca
Comment: [Press Enter]
You selected this USER-ID:
    "Gerhard Mourani <gmourani@videotron.ca>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? o
You need a Passphrase to protect your secret key.

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
+++++..+++++++++++++++..+++++.+++++++++++++++++++++++++++++++++++++++..+++++++
+++.+++++++++++++++++++++++++.++++++++++++++...+++++++++++++++++++++++++.++++
+
..+++++>+++++...++++++++++++++>+++++.......>+++++.......<+++++.................
.........+++++^^^^
public and secret key created and signed.
```

A new key-pair is created (key pair: secret and public key) in the “root” home directory (~/root).

Importing keys

If you have received a key from someone else you can put it into your public keyring database in order to be able to use him/her key. This is called "importing".

- To import Public Keys to your keyring, use the following command:
  [root@deep]# **gpg --import <file>**
  As an example:
  [root@deep]# **gpg --import redhat2.asc**
  gpg: key DB42A60E: public key imported
  gpg: /root/.gnupg/trustdb.gpg: trustdb created
  gpg: Total number processed: 1
  gpg: imported: 1

New keys are appended to your keyring and already existing keys are updated. Note that GnuPG does not import keys that are not self-signed. In the above example we import the Public Key file “redhat2.asc” of the company Red Hat Linux downloadable from the Red Hat Internet site in our keyring.

Key signing

When you import keys and are sure that somebody is really the person they claim, you can start signing his/her keys.

- To sign a key for the compagny RedHat that we have added on our keyring above, use the following command:
  [root@deep]# **gpg --sign-key <UID>**
  As an example:
  [root@deep]# **gpg --sign-key RedHat**

  pub 1024D/DB42A60E created: 1999-09-23 expires: never trust: -/q
  sub 2048g/961630A2 created: 1999-09-23 expires: never
  (1) Red Hat, Inc <security@redhat.com>

  pub 1024D/DB42A60E created: 1999-09-23 expires: never trust: -/q
  Fingerprint: CA20 8686 2BD6 9DFC 65F6 ECC4 2191 80CD DB42 A60E

  Red Hat, Inc <security@redhat.com>

  Are you really sure that you want to sign this key
  with your key: "Gerhard Mourani <gmourani@videotron.ca>"

  Really sign? **y**

  You need a passphrase to unlock the secret key for
  user: "Gerhard Mourani <gmourani@videotron.ca>"
  1024-bit DSA key, ID E92D6C97, created 1999-12-30

  Enter passphrase:

You should only sign a key as being authentic when you are ABSOLUTELY SURE that the key is really authentic!!!. You should never sign a key based on any assumption.

Encrypt and decrypt
After installing everything and configuring everything in the way we want, we can start on encrypting and decrypting.

- To encrypt and sign data for user RedHat that we have added on our keyring above, use the following command:
  [root@deep]# **gpg -sear RedHat <file>**
  As an example:
  [root@deep]# **gpg -sear RedHat message-to-RedHat.txt**
  You need a passphrase to unlock the secret key for
  user: "Gerhard Mourani (Open Network Architecture) <gmourani@videotron.ca>"
  1024-bit DSA key, ID BBB4BA9B, created 1999-10-26
  Enter passphrase:

Which mean “s” for signing (To avoid the risk that somebody else claims to be you, it is very useful to sign everything you encrypt), “e” for encrypting, “a” to create ASCII armored output (“.asc” ready for sending by mail), “r” encrypt for user id name and <file> is the message you want to encrypt.

- To decrypt data, use the following command:
  [root@deep]# **gpg -d <file>**
  For example:
  [root@deep]# **gpg -d message-to-Gerhard.asc**
  You need a passphrase to unlock the secret key for
  user: "Gerhard Mourani (Open Network Architecture) <gmourani@videotron.ca>"
  2048-bit ELG-E key, ID 71D4CC44, created 1999-10-26 (main key ID BBB4BA9B)
  Enter passphrase:

Which mean “-d” is for decrypting and <file> is the message you want to decrypt.

Exporting keys
GnuPG has some options to help you publish public keys. This is called "exporting" a key. By exporting public keys you can broaden your horizon. This can be done by publishing it on your homepage, through a key server or any other method you can think of.

- To extract your public key in ASCII armored output, use the following command:
  [root@deep]# **gpg --export --armor > Public-key.asc**

Which mean “--export” is for extracting your Public-key from your pubring encrypted file, “--armor” to create ASCII armored output that you can mail, publish or put it on a web page and “> Public-key.asc” to put the result in a file that you’re named Public-key.asc.

Signing and checking
Everyone who knows your public key (you can and should publish your key by putting it on a key server, a web page, etc) is now able to check whether you really signed this text.

- To sign a message or a binary file, use the following command:
  [root@deep]# **gpg --detach-sign --armor <Data>**

You can write the signature in a separate file. It is highly recommended to use this option especially when signing binary files (like archives for instance). Also the --armor option can be extremely useful here. <data> can be a file or a binary file like tar archive.

- To check the signature of encrypted data, use the following command:
  [root@deep]# **gpg --verify <Data>**

GnuPG now checks whether the signature is valid and prints an appropriate message. If the signature is good, you know at least that the person (or machine) has access to the secret key, which corresponds to the published public key. When encrypted data has been signed as well, the signature is checked when the data is decrypted. This will only work (of course) when you own the public key of the sender. <data> is the same above.

Installed files

> /usr/bin/gpg
> /usr/lib/gnupg
> /usr/lib/gnupg/rndunix
> /usr/lib/gnupg/rndegd
> /usr/lib/gnupg/tiger
> /usr/man/man1/gpg.1
> /usr/share/gnupg
> /usr/share/gnupg/options.skel

Chapter 10 Servers Software

In this Chapter

Linux DNS and BIND Server

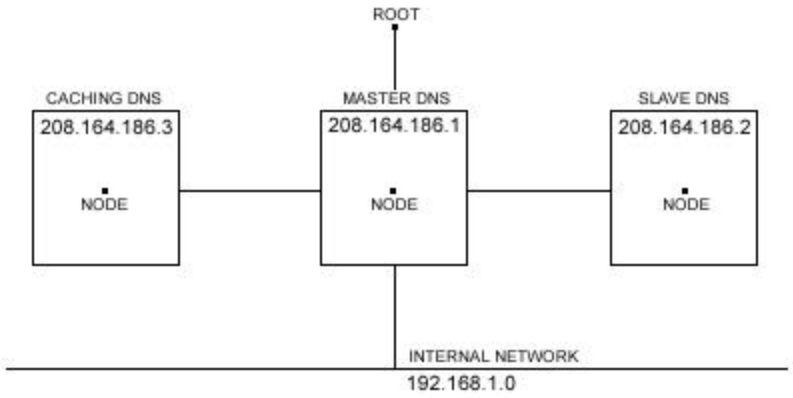
Overview

DNS is the **MOST** important network service for IP networks. All Unix **Client** machines should be configured to perform **caching functions** at minimum. Setting up a caching server for Client local machines will reduce load on the site's primary server. A caching only name server will find the answer to name queries and remember the answer the next time we need it. This will shorten the waiting time the next time significantly.

For security reason, it is very important that DNS doesn't exist between hosts on the corporate network and external hosts, it is far safer to simply use IP addresses to connect to external machines from the corporate network and vice-versa.

In our configuration and installation we'll run BIND as non root-user and in a chrooted environment. We also provide you three different configurations, one for a simple caching name server only, one for a slave and another one for a master name server.

The simple caching name server configuration will be used for your servers that not acts as a master or slave name server and the slave and master configurations will be used for your servers that act as a master name server and slave name server. Usually one of your server act as master, another one act as slave and the rest acts as simple caching name server.



This is the graphical representation of the DNS configuration we use on this book. We try to show you different setting (Caching Only DNS, Master DNS, and Slave DNS) on different servers. Lot possibilities exist and depend of your needs and network architecture.

These installation instructions assume

Commands are Unix-compatible.
The source path is "/var/tmp" (other paths are possible).
Installations were tested on RedHat Linux 6.1.
All steps in the installation will happen in superuser account "root".
Bind version number is 8.2.2-patchlevel5

Packages

Bind Homepage: http://www.isc.org/
You must be sure to download: bind-contrib.tar.gz, bind-doc.tar.gz, bind-src.tar.gz

Tarballs

It is a good idea to make a list of files on the system before you install Bind, and one afterwards, and then compare them using '**diff**' to find out what file it placed where. Simply run '**find /* > dns1**' before and '**find /* > dns2**' after you install the software, and use '**diff dns1 dns2 > dns**' to get a list of what changed.

Compilation

Decompress the tarball (tar.gz).

```
[root@deep]# mkdir /var/tmp/bind
[root@deep]# cp bind-contrib.tar.gz /var/tmp/bind/
[root@deep]# cp bind-doc.tar.gz /var/tmp/bind/
[root@deep]# cp bind-src.tar.gz /var/tmp/bind/
```

We create a directory named "bind" to handle the tar archives and copy them to this new directory.

Cd into the new bind directory (cd /var/tmp/bind) and decompress the tar files:

```
[root@deep]# tar xzpf bind-contrib.tar.gz
[root@deep]# tar xzpf bind-doc.tar.gz
[root@deep]# tar xzpf bind-src.tar.gz
```

Configure and Optimize

Edit the **Makefile.set** file (vi src/port/linux/Makefile.set) and add, modify:

```
'CC=egcs -D_GNU_SOURCE'
'CDEBUG=-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions'
'DESTBIN=/usr/bin'
'DESTSBIN=/usr/sbin'
'DESTEXEC=/usr/sbin'
'DESTMAN=/usr/man'
'DESTHELP=/usr/lib'
'DESTETC=/etc'
'DESTRUN=/var/run'
'DESTLIB=/usr/lib/bind/lib'
'DESTINC=/usr/lib/bind/include'
'LEX=flex -8 -l'
'YACC=yacc -d'
'SYSLIBS=-lfl'
'INSTALL=install'
'MANDIR=man'
'MANROFF=cat'
'CATEXT=$$N'
'PS=ps -p'
'AR=ar crus'
'RANLIB=:'
```

The first line represent the name of our GCC compiler (egcs), the second is our optimization flags. The "DESTLIB=" line specify the path of the library directory for Bind and the "DESTINC=" is where we put the include directory of Bind.

Compile and Optimize

Type the following commands on your terminal

```
[root@deep]# make -C src
[root@deep]# make clean all -C src SUBDIRS=../doc/man
[root@deep]# make install -C src
[root@deep]# make install -C src SUBDIRS=../doc/man
```

The “make” command compile all source files into executable binaries, and then “make install” install the binaries and any supporting files into the appropriate locations.

```
[root@deep]# strip /usr/bin/addr
[root@deep]# strip /usr/bin/dig
[root@deep]# strip /usr/bin/dnsquery
[root@deep]# strip /usr/bin/host
[root@deep]# strip /usr/bin/nslookup
[root@deep]# strip /usr/bin/nsupdate
[root@deep]# strip /usr/bin/mkservdb
[root@deep]# strip /usr/sbin/named
[root@deep]# strip /usr/sbin/named-xfer
[root@deep]# strip /usr/sbin/ndc
[root@deep]# strip /usr/sbin/dnskeygen
[root@deep]# strip /usr/sbin/irpd
[root@deep]# mkdir /var/named
```

The “strip” command would discard all symbols from the object files. This means that our binaries files will be smaller in size. This will improve a bit the performance hit to the program since they will be fewer lines to read by the system when it’ll execute the binary. The “mkdir” would create a new directory “/var/named”.

Cleanup after work

```
[root@deep]# cd /var/tmp
[root@deep]# rm -rf bind/
```

Will remove all the source files we have used to compile and install DNS/Bind.

Configurations

Configuration files for different services are very specific depending of your need and your network architecture. People can install DNS Server at home like a caching only DNS and company can install it with primary, secondary and caching DNS servers.

All software we describe in our book "Linuxsos.pdf" will have a specific directory and subdirectory in a tar compressed archive named “floppy.tgz” containing file configurations for the specific program. If you get this archive file, you wouldn’t be obliged to reproduce the different configuration files bellow manually or cut and past them to create your configuration files. Whatever your decide to copy manually or get the files make to your convenience from the archive compressed files, it will be to your responsibility to modify, adjust for your needs and place the files related to BIND/DNS software to their appropriated places on your server machine, like show bellow. The server configuration files archive is located at the following Internet address: http://pages.infinit.net/lotus1/doc/opti/floppy.tgz

- To run a caching only name server, the following files are require and must be create or copied to their appropriated directories on your server.

  Copy the **named.conf** file to the “/etc/” directory.
  Copy the **db.127.0.0** file to the “/var/named/” directory.
  Copy the **db.cache** file to the “/var/named/” directory.
  Copy the **named** script file to the “/etc/rc.d/init.d/” directory.

- To run a master name server, the following files are require and must be create or copied to their appropriated directories on your server.

  Copy the **named.conf** file to the "/etc/" directory.
  Copy the **db.127.0.0** file to the "/var/named/" directory.
  Copy the **db.cache** file to the "/var/named/" directory.
  Copy the **db.192.168.1** file to the "/var/named/" directory.
  Copy the **db.openarch** file to the "/var/named/" directory.
  Copy the **named** script file to the "/etc/rc.d/init.d/" directory.

- To run a slave name server, the following files are require and must be create or copied to their appropriated directories on your server.

  Copy the **named.conf** file to the "/etc/" directory.
  Copy the **db.127.0.0** file to the "/var/named/" directory.
  Copy the **db.cache** file to the "/var/named/" directory.
  Copy the **named** script file to the "/etc/rc.d/init.d/" directory.

You can obtain configuration files listed bellow on the "floppy.tgz" archive. Copy the following files from the decompressed "floppy.tgz" archive to their appropriated places or copy and paste them directly from this book to the concerned file.

Caching-only name Server

Caching-only name servers are servers not authoritative for any domains except 0.0.127.in-addr.arpa. A caching-only name server can look up names inside and outside your zone, as can primary and slave name servers. The difference is that when a caching-only name server initially looks up a name within your zone, it ends up asking one of the primary or slave names servers for your zone for the answer.

The necessary files to setup a simple caching name server are:
named.conf
db.127.0.0
db.cache
named script

Configuration of the "/etc/named.conf" file for a simple caching name server

Use this configuration for all servers machines on your network that doesn't act as a master or slave name server. Setting up a simple caching server for local machines will reduce load on the site's primary server. Many users on dialup connections use this configuration along with bind for such a purpose.

Create the **named.conf** file (touch /etc/named.conf) and add the following lines in the file:

```
options {
      directory "/var/named";
      forwarders { 208.164.186.1; 208.164.186.2; };
      forward only;
};

//
// a caching only nameserver config
zone "." in {
          type hint;
```

```
        file "db.cache";
};

zone "0.0.127.in-addr.arpa" in {
        type master;
        file "db.127.0.0";
};
```

In the "**forwarders**" line, 208.164.186.1 and 208.164.186.2 are the IP addresses of your Primary (Master) and Secondary (Slave) DNS server. They can also be the IP addresses of your ISP's DNS server and another DNS server respectively.

You may want to stop your name servers from even trying to contact an off-site server if their forwarder is down or doesn't respond. A "**forward only**" name servers doesn't try to contact other servers to find out information if the forwarder don't give it an answer. This is a security feature.

Configuration of the "/var/named/db.127.0.0" file for a simple caching name server

Use this configuration for all servers machines on your network that doesn't act as a master or slave name server. The "db.127.0.0" file cover the loopback network, the special address that host use to direct traffic to temselves. Create the following files in "/var/named/".

Create the **db.127.0.0** file (touch /var/named/db.127.0.0) and add the following lines in the file:

```
$TTL 345600
@       IN      SOA     localhost. root.localhost.  (
                                00              ; Serial
                                86400           ; Refresh
                                7200            ; Retry
                                2592000         ; Expire
                                345600 )        ; Minimum
        IN      NS      localhost.

1       IN      PTR     localhost.
```

Configuration of the "/var/named/db.cache" file for a simple caching name server

Before starting your DNS server you must take a copy of "**db.cache"** file and copy this file in the "/var/named/" directory. The "**db.cache"** tells your server where the servers for the "root" zone are.

Use the following command on another Unix computer in your organization to query a new **db.cache** file for your DNS Server or pick one from your Red Hat Linux CD-ROM source distribution:

```
[root@deep]# dig @.aroot-servers.net . ns > db.cache
```

Don't forget to copy the db.cache file to the "/var/named/" directory on your server where you're installing DNS server after retrieving it.

NOTE: Internal addresses like 192.168.1/24 are not included in the DNS configuration files for security reason. It is very important that DNS doesn't exist between hosts on the corporate network and external hosts.

Primary master name Server

A primary master name server for a zone reads the data for the zone from a file on it's host. Primary master name servers for a zone are authoritative for that zone.

The necessary files to setup a primary master name server are:
named.conf
db.127.0.0
db.208.164.186
db.openarch
db.cache
named script

Configuration of the "/etc/named.conf" file for a master name server

Use this configuration for the server machine on your network that act as a master name server. After compiling DNS, you need to set up a domain for your server - we'll be using "openarch.com" as an example domain, and assuming you are using 208.164.186.0. What we'll be doing is setting up your server as a primary with slave server DNS and restricting access to it. To do this, add the following lines to your "/etc/named.conf" (this assumes you are using the current version of bind, 8.2):

Create the **named.conf** file (touch /etc/named.conf) and add:

```
options {
        directory "/var/named";
        fetch-glue no;
        recursion no;
        allow-query { 208.164.186/24; 127.0.0/8; };
        allow-transfer { 208.164.186.2; };
        transfer-format many-answers;
};

// These files are not specific to any zone
zone "." in {
        type hint;
        file "db.cache";
};

zone "0.0.127.in-addr.arpa" in {
        type master;
        file "db.127.0.0";
};

// These are our primary zone files
zone "openarch.com" in {
        type master;
        file "db.openarch";
};

zone "186.164.208.in-addr.arpa" in {
        type master;
        file "db.208.164.186";
};
```

The "**fetch-glue no**" option can be used in conjunction with "**recursion no**" option to prevent the server's cache from growing or becoming corrupted. Also, disabling recursion puts your name servers into a passive mode, telling them never to send queries on behalf of other name servers or resolvers. A non-recursive name server is very difficult to spoof, since it doesn't send queries, and hence doesn't cache any data. This is a security feature.

In the "**allow-query**" line, 208.164.186/24 and 127.0.0/8 are the IP addresses allowed to ask ordinary questions to the server.

In the "**allow-transfer**" line, 208.164.186.2 is the IP address allowed to receive zone transfers from the server. You must ensure that only your real slave name servers can transfer zones from your name server. As the information provided is often used by spammers and IP spoofers.

NOTE: The options "recursion no", "allow-query", and "allow-transfer" in the "named.conf" file above are a security feature.

Configuration of the "/var/named/db.127.0.0" file for a master and slave name server

This configuration file can by used by master name server and slave name server. The "db.127.0.0" file cover the loopback network, the special address that host use to direct traffic to temselves. Create the following files in "/var/named/".

Create the **db.127.0.0** file (touch /var/named/db.127.0.0) and add:

```
; Revision History: April 22, 1999 - admin@mail.openarch.com
; Start of Authority (SOA) records.
$TTL 345600
@  IN  SOA  deep.openarch.com. admin.mail.openarch.com.  (
                        00              ; Serial
                        86400           ; Refresh
                        7200            ; Retry
                        2592000         ; Expire
                        345600 )        ; Minimum

; Name Server (NS) records.
     NS   deep.openarch.com.
     NS   mail.openarch.com.

; only One PTR record.
1      PTR  localhost.
```

Configuration of the "/var/named/db.208.164.186" file for a master name server

Use this configuration for the server machine on your network that act as a master name server. This file "db.208.164.186" map host names to addresses. Create the following files in "/var/named/".

Create the **db.208.164.186** file (touch /var/named/db.208.164.186) and add:

```
; Revision History: April 22, 1999 - admin@mail.openarch.com
; Start of Authority (SOA) records.
$TTL 345600
@  IN  SOA  deep.openarch.com. admin.mail.openarch.com.  (
                        00              ; Serial
                        86400           ; Refresh
                        7200            ; Retry
                        2592000         ; Expire
                        345600 )        ; Minimum

; Name Server (NS) records.
     NS   deep.openarch.com.
     NS   mail.openarch.com.

; Addresses Point to Canonical Names (PTR) for Reverse lookups
```

```
1  PTR     deep.openarch.com.
2  PTR     mail.openarch.com.
3  PTR     www.openarch.com.
```

Configuration of the "/var/named/db.openarch" file for a master name server

Use this configuration for the server machine on your network that act as a master name server. This file "db.openarch" map addresses to host names. Create the following file in "/var/named/".

Create the **db.openarch** file (touch /var/named/db.openarch) and add:

```
; Revision History: April 22, 1999 - admin@mail.openarch.com
; Start of Authority (SOA) records.
$TTL 345600
@  IN  SOA  deep.openarch.com. admin.mail.openarch.com.  (
                        00              ; Serial
                        86400           ; Refresh
                        7200            ; Retry
                        2592000         ; Expire
                        345600 )        ; Minimum

; Name Server (NS) records.
     NS   deep.openarch.com.
     NS   mail.openarch.com.

; Mail Exchange (MX) records.
     MX   0  mail.openarch.com.

; Address (A) records.
localhost               A     127.0.0.1
deep                    A     208.164.186.1
mail                    A     208.164.186.2
www                     A     208.164.186.3

; Aliases in Canonical Name (CNAME) records.
;www                    CNAME  deep.openarch.com.
```

Configuration of the "/var/named/db.cache" file for a master and slave name servers

Before starting your DNS server you must take a copy of "**db.cache"** file and copy this file in the "/var/named/" directory. The "**db.cache"** tells your server where the servers for the "root" zone are.

Use the following command on another Unix computer in your organization to query a new **db.cache** file for your DNS Server or pick one from your Red Hat Linux CD-ROM source distribution:

```
[root@deep]# dig @.aroot-servers.net . ns > db.cache
```

Don't forget to copy the "db.cache" file to the "/var/named/" directory on your server where you're installing DNS server after retrieving it.

Secondary slave name Server

A slave name server splits the load with the master server or handles the whole load if the master server is down. A slave name server loads its data over the network from another name server (usually the master name server). This process is called a zone transfer.

Necessary files to setup a secondary slave name server are:
named.conf
db.127.0.0
db.cache
named script

Configuration of the "/etc/named.conf" file for a slave name server

Use this configuration for the server machine on your network that act as a slave name server. You must modify the "named.conf" file on the slave name server host. Change every occurrence of primary to secondary except for "0.0.127.in-addr.arpa" and add a masters line with the IP address of the master server.

Create the **named.conf** file (touch /etc/named.conf) and add:

```
options {
        directory "/var/named";
        fetch-glue no;
        recursion no;
        allow-query { 208.164.186/24; 127.0.0/8; };
        allow-transfer { 208.164.186.1; };
        transfer-format many-answers;
};

// These files are not specific to any zone
zone "." in {
        type hint;
        file "db.cache";
};

zone "0.0.127.in-addr.arpa" in {
        type master;
        file "db.127.0.0";
};

// These are our slave zone files
zone "openarch.com" in {
        type slave;
        file "db.openarch";
        masters { 208.164.186.1; };
};

zone "186.164.208.in-addr.arpa" in {
        type slave;
        file "db.208.164.186";
        masters { 208.164.186.1; };
};
```

This tells the name server that it is a slave for the zone "openarch.com" and that it should track the version of this zone that is being kept on the host "208.164.186.1".

A slave name server doesn't need to retrieve all of its db files over the network; the overhead files, "db.127.0.0" and "db.cache", are the same as on a primary master, so keep a local copy on the slave.

Copy the "**db.127.0.0**" file from master name server to slave name server.
Copy the "**db.cache**" file from master name server to slave name server.

Configuration of the "/etc/rc.d/init.d/named" script file for all type of name server

This configuration script file can by used for all type of name server (caching, master or slave). Configure your "/etc/rc.d/init.d/named" script file to start and stop DNS/BIND daemons Server.

Create the **named** script file (touch /etc/rc.d/init.d/named) and add:

```
#!/bin/sh
#
# named         This shell script takes care of starting and stopping
#               named (BIND DNS server).
#
# chkconfig: - 55 45
# description: named (BIND) is a Domain Name Server (DNS) \
# that is used to resolve host names to IP addresses.
# probe: true

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

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

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

[ -f /usr/sbin/named ] || exit 0

[ -f /etc/named.conf ] || exit 0

RETVAL=0

# See how we were called.
case "$1" in
  start)
        # Start daemons.
        echo -n "Starting named: "
        daemon named
                RETVAL=$?
                [ $RETVAL -eq 0 ] && touch /var/lock/subsys/named
                echo
        ;;
  stop)
        # Stop daemons.
        echo -n "Shutting down named: "
        killproc named
                RETVAL=$?
                [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/named
        echo
        ;;
  status)
                /usr/sbin/ndc status
                exit $?
                ;;
  restart)
                $0 stop
                $0 start
                ;;
  reload)
                /usr/sbin/ndc reload
```

```
        exit $?
        ;;
probe)
        # named knows how to reload intelligently; we don't want linuxconf
        # to offer to restart every time
        /usr/sbin/ndc reload >/dev/null 2>&1 || echo start
        exit 0
        ;;

*)
    echo "Usage: named {start|stop|status|restart}"
    exit 1
esac

exit $RETVAL
```

Now, make this script executable and change its default permission:
[root@deep]# **chmod 700 /etc/rc.d/init.d/named**

Create the symbolic rc.d links for BIND/DNS with the command:
[root@deep]# **chkconfig --add named**

BIND/DNS script will not start automatically the daemon named when you reboot the server. You can change it default by executing the following command:

[root@deep]# **chkconfig --level 345 named on**

Start your DNS Server manually with the following command:
[root@deep]# **/etc/rc.d/init.d/named start**

Securing BIND/DNS

Running BIND in a chroot jail

This part focuses on preventing BIND from being used as a point of break-in to the system hosting it. Since BIND performs a relatively large and complex function, the potential for bugs that affect security is rather high. In fact, there have been exploitable bugs that allowed a remote attacker to obtain root access to hosts running BIND.

To minimize this risk, BIND can be run **as a non-root user**, which will limit any damage to what can be done as a normal user with a local shell. Of course, allowing what amounts to an anonymous guest account falls rather short of the security requirements for most DNS servers, so an additional step can be taken - that is, **running BIND in a chroot jail**.

The main benefit of a chroot jail is that the jail will limit the portion of the filesystem the daemon can see to the root directory of the jail. Additionally, since the jail only needs to support BIND, the programs available in the jail can be extremely limited. Most importantly, there is no need for setuid-root programs, which, given the right (or wrong...) bug, can be used to gain root access and break out of the jail.

NOTE: named program must be in a directory listed in your PATH environmental variable for this to work. If you're root, and indeed do have BIND installed; this should be the case. For the rest of the documentation, I'll assume the path of your original named is “/usr/sbin/named”.

Find the shared library dependencies of named. These will need to be copied into the chroot jail later.

[root@deep]# **ldd /usr/sbin/named**

libc.so.6 => /lib/libc.so.6 (0x40017000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)

Make a note of the files listed above; you will need these later.

Step 1:
Add a new user id and a new group id for running named. This is important because running it as root defeats the purpose of the jail, and using a different user id that already exists on the system can allow your services to access each others' resources. Think multi-layer security.

These are sample user and group id numbers. Check “/etc/passwd” and “/etc/group” files for a free uid/gid number. We'll use 53.

[root@deep]# **groupadd -g 53 named**
[root@deep]# **useradd -g 53 -u 53 named**

Step 2:
Set up the chroot environment. First, create the root directory of the jail. We've chosen “/chroot/named: because we want to put this on its own separate filesystem to prevent filesystem attacks. Early in our installation procedure we are create a special partition “/chroot” for this purpose.

[root@deep]# **/etc/rc.d/init.d/named stop** (only if named daemon is running)
[root@deep]# **mkdir -p /chroot/named**

Next, create the rest of directories like the following:

[root@deep]# **mkdir /chroot/named/dev**
[root@deep]# **mkdir /chroot/named/lib**
[root@deep]# **mkdir /chroot/named/etc**
[root@deep]# **mkdir -p /chroot/named/usr/sbin**
[root@deep]# **mkdir -p /chroot/named/var/run**

Copy the main configuration file, the zone files, the named and named-xfer programs:

[root@deep]# **cp /etc/named.conf /chroot/named/etc/**
[root@deep]# **mkdir /chroot/named/var/named**
[root@deep]# **cd /var/named ; cp -a . /chroot/named/var/named/**
[root@deep]# **mknod /chroot/named/dev/null c 1 3**
[root@deep]# **chmod 666 /chroot/named/dev/null**
[root@deep]# **cp /usr/sbin/named /chroot/named/usr/sbin/**
[root@deep]# **cp /usr/sbin/named-xfer /chroot/named/usr/sbin/**

IMPORTANT NOTE: The owner of the “/chroot/named/var/named” and all file in this directory must be the process name “named” under the slave server and only the slave server or you wouldn’t be able to make a zone transfer.

- To make “named” directory and all its files own by “named” process name under the slave server, use the command:
  [root@deep]# **chown -R named.named /chroot/named/var/named/**

- Set the immutable bit on “named.conf” file:
  [root@deep]# **cd /chroot/named/etc/**
  [root@deep]# **chattr +i named.conf**

Copy the shared libraries identified above to the chrooted lib directory:

[root@deep]# **cp /lib/libc.so.6 /chroot/named/lib/**
[root@deep]# **cp /lib/ld-linux.so.2 /chroot/named/lib/**

Copy the “localtime” and “nsswitch.conf” file to the jail so that log entries are adjusted for your local timezone properly:

[root@deep]# **cp /etc/localtime /chroot/named/etc/**
[root@deep]# **cp /etc/nsswitch.conf /chroot/named/etc/**

- Set the immutable bit on “nsswitch.conf” file:
  [root@deep]# **cd /chroot/named/etc/**
  [root@deep]# **chattr +i nsswitch.conf**

A file with the “+i” attribute cannot be modified: it cannot be deleted or renamed, no link can be created to this file and no data can be written to the file. Only the superuser can set or clear this attribute.

Step 3:
Tell syslogd about the new chrooted service:

Normally, processes talk to syslogd through “/dev/log”. As a result of the chroot jail, this won't be possible, so syslogd needs to be told to listen to “/chroot/named/dev/log”. To do this, edit the syslog startup script to specify additional places to listen.

Edit the **syslog** script (vi +24 /etc/rc.d/init.d/syslog) to change the line:

daemon syslogd -m 0
To read:
daemon syslogd -m 0 -a /chroot/named/dev/log

Step 4:
Edit the **named** script (vi /etc/rc.d/init.d/named) to change the line:

[-f /usr/sbin/named] || exit 0
To read:
[-f /chroot/named/usr/sbin/named] || exit 0

[-f /etc/named.conf] || exit 0
To read:
[-f /chroot/named/etc/named.conf] || exit 0

daemon named
To read:
daemon /chroot/named/usr/sbin/named -t /chroot/named/ -unamed -gnamed

-t tells named to start up using the new chroot environment,
-u specifies the user to run as,
-g specifies the group to run as.

Red Hat's init scripts' daemon() function doesn't allow alternate PID files to be specified, but that won't affect the operation of the start and stop actions of the named init scripts since they will be called from outside the chroot jail.

As of BIND 8.2 “ndc” command became a binary file, which renders the shipped ndc useless in this setting. To fix it, the BIND package must be compiled from source.

Keep in mind that these settings only apply to ndc. If you also want a new named and named xfer, the original settings should be used.

To do this, in the top level BIND source directory.

For ndc:
[root@deep]# **cp bind-src.tar.gz /vat/tmp**
[root@deep]# **cd /var/tmp/**
[root@deep]# **tar xzpf bind-src.tar.gz**
[root@deep]# **cd src**
[root@deep]# **cp port/linux/Makefile.set port/linux/Makefile.set-orig**

Edit the **Makefile.set** file (vi port/linux/Makefile.set) to make the changes listed bellow:

'CC=**egcs** -D_GNU_SOURCE'
'CDEBUG=**-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions'**
'DESTBIN=/usr/bin'
'DESTSBIN=/chroot/named/usr/sbin'
'DESTEXEC=/chroot/named/usr/sbin'
'DESTMAN=/usr/man'
'DESTHELP=/usr/lib'
'DESTETC=/etc'
'DESTRUN=/chroot/named/var/run'
'DESTLIB=/usr/lib/bind/lib'
'DESTINC=/usr/lib/bind/include'
'LEX=flex -8 -l'
'YACC=yacc -d'
'SYSLIBS=-lfl'
'INSTALL=install'
'MANDIR=man'
'MANROFF=cat'
'CATEXT=$$N'
'PS=ps -p'
'AR=ar crus'
'RANLIB=:'

The difference between the Makefile we used before and this one is that we modify the "DESTSBIN=", "DESTEXEC=", and "DESTRUN=" lines to point to the chrooted directory of DNS/BIND. With this modification, ndc program know where to find "named".

[root@deep]# **make clean**
[root@deep]# **make**
[root@deep]# **cp bin/ndc/ndc /usr/sbin/**
[root@deep]# cp: overwrite `/usr/sbin/ndc'? **y**
[root@deep]# **strip /usr/sbin/ndc**

We build the binary file then copy the result of ndc program to "/usr/sbin" and overwrite the old one. We don't forget to strip our new ndc binary for better performance.

It is a good idea to also build a new "named" binary now, to ensure the same version is used for both named and ndc.

For named:
[root@deep]# **cd /var/tmp/src**
[root@deep]# **cp port/linux/Makefile.set-orig port/linux/Makefile.set**
[root@deep]# cp: overwrite `port/linux/Makefile.set'? **y**

Edit the **Makefile.set** file (vi port/linux/Makefile.set) to make the changes listed bellow:

'CC=**egcs** -D_GNU_SOURCE'

'CDEBUG=**-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions'**
'DESTBIN=/usr/bin'
'DESTSBIN=/usr/sbin'
'DESTEXEC=/usr/sbin'
'DESTMAN=/usr/man'
'DESTHELP=/usr/lib'
'DESTETC=/etc'
'DESTRUN=/var/run'
'DESTLIB=/usr/lib/bind/lib'
'DESTINC=/usr/lib/bind/include'
'LEX=flex -8 -l'
'YACC=yacc -d'
'SYSLIBS=-lfl'
'INSTALL=install'
'MANDIR=man'
'MANROFF=cat'
'CATEXT=$$N'
'PS=ps -p'
'AR=ar crus'
'RANLIB=:'

[root@deep]# **rm -f .settings**
[root@deep]# **make clean**
[root@deep]# **make**
[root@deep]# **cp bin/named/named /chroot/named/usr/sbin**
[root@deep]# cp: overwrite `/chroot/named/usr/sbin/named'? **y**
[root@deep]# **cp bin/named-xfer/named-xfer /chroot/named/usr/sbin**
[root@deep]# cp: overwrite `/chroot/named/usr/sbin/named-xfer'? **y**
[root@deep]# **strip /chroot/named/usr/sbin/named**
[root@deep]# **strip /chroot/named/usr/sbin/named-xfer**

We remove the ".settings" file since the build system caches these variables, and we run the "make clean" command to be sure we have no stale trash laying about. After we build the "named" binary and copy it with "named-xfer" to the chrooted directory. Also we use the "strip" command for improving the performance of the new binaries.

Remove unnecessary files and directory

[root@deep]# **rm -f /usr/sbin/named**
[root@deep]# **rm -f /usr/sbin/named-xfer**
[root@deep]# **rm -f /etc/named.conf**
[root@deep]# **rm -rf /var/named/**

We remove the "named" and "named-xfer" binaries from the "/usr/sbin" directory, since the ones we will work with now in our daily use are located under the chroot directory. The same apply for "named.conf" file and "/var/named" directory.

Step 5:
Test the new chrooted configuration! Restart syslogd:

[root@deep]# **/etc/rc.d/init.d/syslog restart**

Now, start the new chrooted BIND:

[root@deep]# **/etc/rc.d/init.d/named start**

Make sure it's running as named and with the new arguments.

[root@deep]# **ps auxw | grep named**

named 11446 0.0 1.2 2444 1580 ? S 23:09 0:00 /chroot/named/usr/sbin/named -t /chroot/named/ -unamed -gnamed

The first column should be “named”, which is the user-id named is running under. The end of the line should be “named -t /chroot/named/ -unamed -gnamed”.

Cleanup after work
[root@deep]# rm -rf /var/tmp/bind/

NOTE: For security reason, it is very important that DNS doesn't exist between hosts on the corporate network and external hosts, it is far safer to simply use IP addresses to connect to external machines from the corporate network and vice-versa.

Zone transfers

Restrict Zone Transfers:
Restricting zone transfers prevents:

- Others from taxing the name server.
- Hackers from listing the contents of the zones.

Here is an example of what a “allow-transfer” option in the named.conf file would contain:

```
options {
        allow-transfer { 208.164.186.2; };
};
```

This control which slave name servers can transfers any zones from this name server. Note on restricting zone transfers: Remember to restrict zone transfers from slave name servers, not just the primary master.

Allow-query

Restrict the queries that your name servers accept to:

- The address they should come from.
- The zones they should ask about.

Here is an example of what a “allow-query” option in the named.conf file would contain:

```
options {
        allow-query { 208.164.186/24; 127.0.0/8; };
};
```

This specifies which IP addresses are allowed to send queries to the server. In particular, people who run Internet firewalls may have a legitimate need to hide certain parts of their name space from most of the world but to make if available to a limited audience.

Forward-only

You may want to stop your name servers from even trying to contact an off-site server if their forwarder is down or doesn’t respond. A “**forward only**” name servers doesn’t try to contact other servers to find out information if the forwarder don’t give it an answer.

Here is an example of what a “**forward only**” option in the named.conf file would contain:

```
options {
```

```
        forwarders { 205.151.222.250; 205.151.222.251; };
        forward-only;
};
```

In the "**forwarders**" line, 205.151.222.250 and 205.151.222.251 are the IP addresses of your ISP's DNS server and another DNS server respectively.

Further documentation

For more details, there are several man pages you can read:

```
$ man dnsdomainname (1)     - show the system's DNS domain name
$ man dnskeygen (1)         - generate public, private, and shared secret keys for DNS Security
$ man dnsquery (1)          - query domain name servers using resolver
$ man named (8)             - Internet domain name server (DNS)
```

DNS Administrative Tools

The commands listed bellow are some that we use often in our regular use but much more exist and you must check the man page and documentation for more details and information.

dig

The "**db.cache***"* tells your server where the servers for the "root" zone are. It must be updated periodically. The root name servers do not change very often, but they do change. A good practice is to check your "**db.cache***"* file every month or two.

- Use the following command to query a new *db.cache* file for your DNS Server:
  [root@deep]# **dig @.aroot-servers.net . ns > db.cache**
  Copy the db.cache file to */var/named/* after retrieving it.
  [root@deep]# **cp db.cache /var/named/**

Where **@.aroot-servers.net** is the address of the root server for query the new db.cache file and **db.cache** file is the name of your new db.cache file.

ndc

This command allows the system administrator to control the operation of a name server. If no command is given, ndc will prompt for commands until it reads EOF.

- Type ndc on your terminal and then help to see help on different command.
  [root@deep]# **ndc**
  [root@deep]# **ndc help**

DNS Users Tools

The commands listed bellow are some that we use often in our regular use but much more exist and you must check the man page and documentation for more details and information.

nslookup

Nslookup is a program to query Internet domain name servers. Nslookup has two modes: interactive and non-interactive. Interactive mode allows the user to query name servers for information about various hosts and domains or to print a list of hosts in a domain. Non-interactive mode is used to print just the name and requested information for a host or domain.

Interactive mode have a lot options, better approach will be to see the man page nslookup for information about Interactive mode or by typing help under nslookup Interactive mode.

- To enter under nslookup Interactive mode, use the command:
  [root@deep]# **nslookup** (and type help for more information about the use of nslookup).
  Default Server: deep.openarch.com
  Address: 208.164.186.3

  > **help**

- To run non-interactive mode, use the command:
  [root@deep]# **nslookup www.redhat.com**

Non-interactive mode is used when the name or Internet address of the host to be looked up is given as the first argument. The optional second argument specifies the host name or address of a name server.

dnsquery
The dnsquery program is a general interface to nameservers via BIND resolver library calls. This program is intended to be a replacement or supplement to program like nslookup.

- To query domain name servers using resolver, use the command:
  [root@deep]# **dnsquery <-n nameserver> <host>**
  For example:
  [root@deep]# **dnsquery -n localhost 192.168.1.2**

Where <-n nameserver> is the nameserver to be used in the query. *nameservers* can appear as either Internet addresses of the form w.x.y.z or can appear as domain names (Default: as specified in "/etc/resolv.conf"). <*host*> is the name of the host (or domain) of interest.

host
The host program looks for information about Internet hosts. It gets this information from a set of interconnected servers that are spread across the country. By default, it simply converts between host names and Internet addresses. However, with the ``-t" or ``-a" options, it can be used to find all of the information about this host that is maintained by the domain server.

- To look up host names using domain server, use the command:
  [root@deep]# **host <FQDN, domain names, host names, or host numbers>**
  For example:
  [root@deep]# **host deep.openarch.com**

Where <FQDN, domain names, host names, or host numbers> is either FDQN e.g. (deep.openarch.com), domain names e.g. (openarch.com), host names i.e. (deep) or host numbers e.g. (192.168.1.1).

- To find all of the information about host, use the command:
  [root@deep]# **host <-a domain names >**
  For example:
  [root@deep]# **host -a openarch.com**

Where <domain names> is e.g. (openarch.com). This options, can be used to find all of the information about this host that is maintained by the domain server.

- To list a complete domain, use the command:
  [root@deep]# **host <-l domain names >**
  For example:
  [root@deep]# **host -l openarch.com**

Where <domain names> is e.g. (openarch.com). This options, will give a complete download of the zone data for openarch.com, in the official master file format.

NOTE: ``-l'' is implemented by doing a complete zone transfer and then filtering out the information the you have asked for. This command should be used only if it is absolutely necessary.

Installed files

> /etc/rc.d/init.d/named
> /etc/rc.d/rc0.d/K45named
> /etc/rc.d/rc1.d/K45named
> /etc/rc.d/rc2.d/K45named
> /etc/rc.d/rc3.d/K45named
> /etc/rc.d/rc4.d/K45named
> /etc/rc.d/rc5.d/K45named
> /etc/rc.d/rc6.d/K45named
> /etc/named.conf
> /usr/bin/addr
> /usr/bin/nslookup
> /usr/bin/dig
> /usr/bin/dnsquery
> /usr/bin/host
> /usr/bin/nsupdate
> /usr/bin/mkservdb
> /usr/lib/bind
> /usr/lib/bind/include
> /usr/lib/bind/include/arpa
> /usr/lib/bind/include/arpa/inet.h
> /usr/lib/bind/include/arpa/nameser.h
> /usr/lib/bind/include/arpa/nameser_compat.h
> /usr/lib/bind/include/isc
> /usr/lib/bind/include/isc/eventlib.h
> /usr/lib/bind/include/isc/misc.h
> /usr/lib/bind/include/isc/tree.h
> /usr/lib/bind/include/isc/logging.h
> /usr/lib/bind/include/isc/heap.h
> /usr/lib/bind/include/isc/memcluster.h
> /usr/lib/bind/include/isc/assertions.h
> /usr/lib/bind/include/isc/list.h
> /usr/lib/bind/include/isc/dst.h
> /usr/lib/bind/include/isc/irpmarshall.h
> /usr/lib/bind/include/netdb.h
> /usr/lib/bind/include/resolv.h
> /usr/lib/bind/include/res_update.h
> /usr/lib/bind/include/irs.h
> /usr/lib/bind/include/irp.h
> /usr/lib/bind/include/hesiod.h
> /usr/lib/bind/include/sys
> /usr/lib/bind/include/net
> /usr/lib/bind/lib
> /usr/lib/bind/lib/libbind.a
> /usr/lib/bind/lib/libbind_r.a
> /usr/lib/nslookup.help
> /usr/man/man1/dig.1
> /usr/man/man1/host.1
> /usr/man/man1/dnsquery.1
> /usr/man/man1/dnskeygen.1
> /usr/man/man3/hesiod.3
> /usr/man/man3/gethostbyname.3
> /usr/man/man3/inet_cidr.3
> /usr/man/man3/resolver.3
> /usr/man/man3/getnetent.3
> /usr/man/man3/tsig.3
> /usr/man/man3/getaddrinfo.3
> /usr/man/man3/getipnodebyname.3
> /usr/man/man5/resolver.5
> /usr/man/man5/irs.conf.5
> /usr/man/man5/named.conf.5
> /usr/man/man7/hostname.7
> /usr/man/man7/mailaddr.7
> /usr/man/man8/named.8
> /usr/man/man8/ndc.8
> /usr/man/man8/named-xfer.8
> /usr/man/man8/named-bootconf.8
> /usr/man/man8/nslookup.8
> /usr/man/man8/nsupdate.8
> /usr/sbin/ndc
> /usr/sbin/named
> /usr/sbin/named-xfer
> /usr/sbin/irpd
> /usr/sbin/dnskeygen
> /usr/sbin/named-bootconf
> /var/named

Linux Sendmail Server

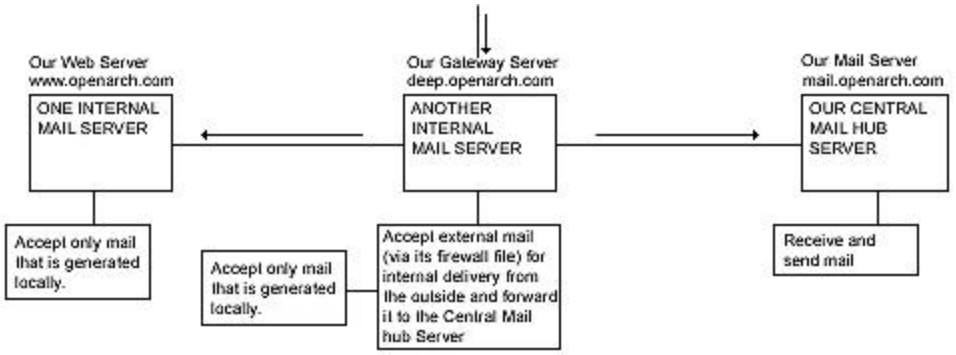
Overview

The Sendmail program is a very widely used Mail Transport Agent (MTA). MTAs send mail from one machine to another. Sendmail is not a client program, which you use to read your e-mail. Sendmail is a behind-the-scenes program, which actually moves your email over networks or the Internet to where you want it to go. Sendmail has been rather buggy and an easy mark for system crackers to exploit, although with the advent of version 8 sendmail, this becomes much more difficult.

In our configuration and installation we'll provide you two different configurations that you can setup for sendmail, one for a Central Mail Hub Relay, and another one for the local or neighbor client and server machines.

The Central Mail Hub Relay Server configuration will be used for your server where the assigned task is to send, receive and relay all mail for all local or neighbor client and server mail machines you may have on your network. A local or neighbor client and server machines refer to all other local server or client machines on your network that run sendmail and send all outgoing mail to the Central Mail Hub for future delivery. This king of internal client never receive mail directly via the Internet, instead all mail receiving from the Internet for those computers are keeps on the Mail Hub server. It is a good idea to run one Central Mail Hub Server for all computers on your network; this architecture will limit the task managements on internal server, client machines and improve the security of your site.

You can configure the neighbor sendmail so that it accept only mail that is generated locally, thus insulating neighbor machines for easier security. The Gateway server (outside the firewall or part of it) acts as a proxy and accepts external mail (via its Firewall file) that is destined for internal delivery from the outside and forwards it to the Central Mail Hub Server. Also note that the Gateway server is configured like a neighbor sendmail server to never accept incoming mail from the outside (Internet).



This is the graphical representation of the Sendmail configuration we use on this book. We try to show you different setting (Central Mail Hub Relay, and local or neighbor client and server machines) on different servers. Lot possibilities exist and depend of your needs and network architecture.

These installation instructions assume

Commands are Unix-compatible.
The source path is "/var/tmp" (other paths are possible).
Installations were tested on RedHat Linux 6.1.
All steps in the installation will happen in superuser account "root".
Sendmail version number is 8.9.3

Packages

Sendmail Homepage: http://www.sendmail.org/
You must be sure to download: sendmail.8.9.3.tar.gz

Tarballs

It is a good idea to make a list of files on the system before you install Sendmail, and one afterwards, and then compare them using '**diff**' to find out what file it placed where. Simply run '**find /* > send1**' before and '**find /* > send2**' after you install the software, and use '**diff send1 send2 > send**' to get a list of what changed.

Compilation

Decompress the tarball (tar.gz).

```
[root@deep]# cp sendmail.version.tar.gz /var/tmp
[root@deep]# cd /var/tmp
[root@deep]# tar xzpf sendmail.version.tar.gz
```

Configure

Cd into the new Sendmail directory and:

Edit the **linux.m4** file (vi +16 cf/ostype/linux.m4) and change:

```
define(`LOCAL_MAILER_PATH', /bin/mail.local)dnl
```

To read:

```
define(`PROCMAIL_MAILER_PATH', `/usr/bin/procmail')dnl
dnl define(`LOCAL_MAILER_FLAGS', `ShPfn')dnl
dnl define(`LOCAL_MAILER_ARGS', `procmail -a $h -d $u')dnl
define(`STATUS_FILE', `/var/log/sendmail.st')dnl
```

NOTE: Those steps are requiring only for a Central Mail Hub configuration. It will make sendmail to use procmail program as the local mailer delivery agent and define the status file "sendmail.st" to be found under "/var/log" directory.

Edit the **linux.m4** file (vi +16 cf/ostype/linux.m4) and add the line:

```
define(`STATUS_FILE', `/var/log/sendmail.st')dnl
```

NOTE: This step is requiring only for a local server, client sendmail configuration. It will define the status file "sendmail.st" to be found under "/var/log" directory.

Edit the **header.m4** file (vi +26 BuildTools/M4/header.m4) and change:

```
define(`confLIBSEARCH', `db bind resolv 44bsd')
```

To read:

```
define(`confLIBSEARCH', `db1 bind resolv 44bsd')
```

This change specifies the new Berkeley DB package installed on Linux.

Edit the **makemap.c** file (vi +30 makemap/makemap.c) and change:

```
# include <db.h>
```

To read:

```
# include <db_185.h>
```

Edit the **map.c** file (vi +29 src/map.c) and change:

include <db.h>
To read:
 # include <db_185.h>

Edit the **udb.c** file (vi +28 src/udb.c) and change:

include <db.h>
To read:
 # include <db_185.h>

Edit the **praliases.c** file (vi +37 praliases/praliases.c) and change:

include <db.h>
To read:
 # include <db_185.h>

Those four changes above specify the version (include <db_185.h>) of Berkeley DB package installed on Linux.

Edit the **Linux** file (vi BuildTools/OS/Linux) and remove the lines:

Remove the following lines:
define(`confSTDIR', `/etc')
define(`confHFDIR', `/usr/lib')
define(`confDEPEND_TYPE', `CC-M')
define(`confMANROOT', `/usr/man/man')

Edit the **Linux** file (vi BuildTools/OS/Linux) and add the lines:

Add the following lines:
define(`confSTDIR', `/var/log')
define(`confHFDIR', `/usr/lib')
define(`confDEPEND_TYPE', `CC-M')
define(`confMANROOT', `/usr/man/man')
define(`confSBINGRP', `root')
define(`confSBINMODE', `6755')
define(`confEBINDIR', `/usr/sbin')

Those lines macro define the variables like the location of the log, lib, man directory, the group name and mode of sendmail binary program under sbin directory.

Edit the **daemon.c** file (vi +1452 src/daemon.c) and change:

nleft = sizeof ibuf - 1;
To read:
nleft = sizeof(ibuf) - 1;

Edit the **smrsh.c** file (vi +61 smrsh/smrsh.c) and change:

define CMDDIR "/usr/adm/sm.bin"
To read:

 # define CMDDIR "/etc/smrsh"

This modification specifies the directory in which all commands must reside.

Edit the **smrsh.c** file (vi +69 smrsh/smrsh.c) and change:

define PATH "/bin:/usr/bin:/usr/ucb"
To read:
 # define PATH "/bin:/usr/bin"

This modification specifies the default search path for commands runs by “smrsh” program.

Compile and optimize

The Build script of Sendmail allows you to specify a site configuration file by using the -f flag like (Build -f ../BuildTools/Site/siteconfig.m4). A site configuration file contains definitions for system installation. We’ll build this site configuration files to suit our system installation and put it in the default “BuildTools/Site” sub-directory of Sendmail source distribution since the Build script will look for the default site configuration files in this directory.

Cd into the new Sendmail directory then creates the **siteconfig.m4** file (touch BuildTools/Site/siteconfig.m4) and adds the following lines inside this file:

```
define(`confMAPDEF', `-DNEWDB') (Require only for Mail Hub configuration)
define(`confENVDEF', `-DPICKY_QF_NAME_CHECK -DXDEBUG=0')
define(`confCC', `egcs')
define(`confOPTIMIZE', `-O9 -funroll-loops -ffast-math -malign-double -mcpu=pent
iumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions')
define(`confLIBS', `-lnsl')
define(`confLDOPTS', `-s')
define(`confMANOWN', `root')
define(`confMANGRP', `root')
define(`confMANMODE', `644')
define(`confMAN1SRC', `1')
define(`confMAN5SRC', `5')
define(`confMAN8SRC', `8')
```

This tells siteconfig.m4 file to set itself up for this particular configuration setup with:

define(`confMAPDEF', `-DNEWDB')
This macro option specifies database types to be included for the alias files and for general maps of Sendmail. In our configuration we’ll use the Berkeley db(3) both hash and btree forms database. The “define(`confMAPDEF', `-DNEWDB')” is only require for the Central Mail Hub configuration and it is not require for local server, client sendmail machines, since in our internal client sendmail machines we don’t use “aliases” database and the other general maps that needs this function “-DNEWDB”.

define(`confENVDEF', `-DPICKY_QF_NAME_CHECK -DXDEBUG=0')
This macro option is used primary to specify code that should either be specially included or excluded. With “-DPICKY_QF_NAME_CHECK“ defined, sendmail will log an error if the name of the “qf” file is incorrectly formed and will rename the “qf” file into a “Qf” file. The “-DXDEBUG=0 “ argument disable the step of additional internal checking during compile time.

define(`confCC', `egcs')
This macro option defines the C compiler to use for compilation of Sendmail. In our case we use the “egcs” C compiler for better optimization.

define(`confOPTIMIZE', `-O9 -funroll-loops -ffast-math -malign-double -mcpu=pent iumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions')
This macro option defines the flags passed to CC for optimization related to our specific CPU architecture.

define(`confLIBS', `-lnsl')
This macro option defines the -l flags passed to ld.

define(`confLDOPTS', `-s')
This macro option defines the linker options passed to ld.

define(`confMANOWN', `root')
This macro option defines the owner of installed man pages.

define(`confMANGRP', `root')
This macro option defines the group of installed man pages.

define(`confMANMODE', `644')
This macro option defines the mode of installed man pages.

define(`confMAN1SRC', `1')
This macro option defines the source for man pages installed in confMAN1.

define(`confMAN5SRC', `5')
This macro option defines the source for man pages installed in confMAN5.

define(`confMAN8SRC', `8')
This macro option defines the source for man pages installed in confMAN8.

[root@deep]# **cd /var/tmp/sendmail-version**
[root@deep]# **cd src**
[root@deep]# **sh Build -f ../BuildTools/Site/siteconfig.m4**
[root@deep]# **cd ..**
[root@deep]# **cd mailstats**
[root@deep]# **sh Build -f ../BuildTools/Site/siteconfig.m4**
[root@deep]# **cd ..**
[root@deep]# **cd makemap** (Require only for Mail Hub configuration)
[root@deep]# **sh Build -f ../BuildTools/Site/siteconfig.m4** (Require only for Mail Hub configuration)
[root@deep]# **cd ..**
[root@deep]# **cd praliases** (Require only for Mail Hub configuration)
[root@deep]# **sh Build -f ../BuildTools/Site/siteconfig.m4** (Require only for Mail Hub configuration)
[root@deep]# **cd ..**
[root@deep]# **cd smrsh**
[root@deep]# **sh Build -f ../BuildTools/Site/siteconfig.m4**
[root@deep]# **cd ..**

NOTE: The “sh Build” sendmail script will create a new directories named “obj.yourOS.youOSkernelversion.yourCPUarchitecture” for example “obj.Linux.2.2.13.i686” under each subdirectories program you may install and then creates links inside those directories to all the necessary source files and Makefiles.

[root@deep]# **make install -C src/obj.Linux.version.architecture**
[root@deep]# **make install -C mailstats/obj.Linux.version.architecture**
[root@deep]# **make install -C makemap/obj.Linux.version.architecture** (Only for Mail Hub configuration)
[root@deep]# **make install -C praliases/obj.Linux.version.architecture** (Only for Mail Hub configuration)
[root@deep]# **make install -C smrsh/obj.Linux.version.architecture**

[root@deep]# **ln -fs /usr/sbin/sendmail /usr/lib/sendmail**
[root@deep]# **strip /usr/sbin/mailstats**
[root@deep]# **strip /usr/sbin/makemap** (Only for Mail Hub configuration)
[root@deep]# **strip /usr/sbin/praliases** (Only for Mail Hub configuration)
[root@deep]# **strip /usr/sbin/smrsh**
[root@deep]# **strip /usr/sbin/sendmail**
[root@deep]# **chown 0.0 /usr/sbin/mailstats**
[root@deep]# **chown 0.0 /usr/sbin/makemap** (Only for Mail Hub configuration)
[root@deep]# **chown 0.0 /usr/sbin/praliases** (Only for Mail Hub configuration)
[root@deep]# **chown 0.0 /usr/sbin/smrsh**
[root@deep]# **chmod 511 /usr/sbin/smrsh**
[root@deep]# **install -d -m755 /var/spool/mqueue**
[root@deep]# **chown root.mail /var/spool/mqueue**
[root@deep]# **mkdir /etc/smrsh**
[root@deep]# **mkdir /etc/mail** (Only for Mail Hub configuration)

The **"sh Build -f"** command would build and make the necessary dependencies in "obj.Linux.version.architecture" of the different files require by sendmail before installation on your system.

The **"make install -C"** command would install sendmail, mailstats, makemap, praliases, smrsh binaries and links as well as the corresponding man pages on your system.

The **"ls -fs"** command would make a symbolic link of sendmail binary to "/usr/lib" directory. This is requiring since some programs hopes to find sendmail binary in this directory (/usr/lib).

The **"strip"** command would reduce the size of mailstats, praliases, sendmail, smrsh, and makemap binaries for optimum performance.

The "**install**" command would create the directory "mqueue" with permission 755 under "/var/spool". A mail message can be temporarily undeliverable for a wide variety of reasons. To ensure that such messages are eventually delivered, sendmail stores them in its queue directory until they can be delivered successfully.

The "**chown**" command would make UID and GID to "root" for files: mailstats, makemap, praliases, smrsh, and UID "root" GID "mail" for mqueue directory.

The "**mkdir**" command would create a "/etc/mail" and "/etc/smrsh" directories on your system.

NOTE: The programs "makemap", and "praliases" must only be installed on the Central Mail Hub Server". The "makemap" permit to create a database maps like the "/etc/aliases" or the "/etc/mail/access" files for sendmail. The "praliases" display the system mail aliases (the content of /etc/aliases file). Since it is better to only have one place like our Central Mail Hub to handle and manage all the db files in our network, then it is not necessary to use "makemap", and "praliases" programs and build db files on your other hosts in the network.

Configurations

All software we describe in our book "Linuxsos.pdf" will have a specific directory and subdirectory in a tar compressed archive named "floppy.tgz" containing file configurations for the specific program. If you get this archive file, you wouldn't be obliged to reproduce the different configuration files bellow manually or cut and past them to create your configuration files. Whatever your decide to copy manually or get the files made to your convenience from the archive compressed files, it will be to your responsibility to modify, adjust for your needs and place the files related to Sendmail software to their appropriated places on your server machine, like show bellow. The server configuration files archive is located at the following Internet address: http://pages.infinit.net/lotus1/doc/opti/floppy.tgz

- To run a Central Mail Hub Server, the following files are require and must be create or copied to their appropriated directories on your server.

  Copy the **access** file in the "/etc/mail/" directory.
  Copy the **aliases** file in the "/etc/" directory.
  Copy the **sendmail.cw** file in the "/etc/" directory.
  Copy the **sendmail.mc** file in the "/etc/" directory.
  Copy the **sendmail** file in the "/etc/sysconfig" directory.
  Copy the **sendmail** script file in the "/etc/rc.d/init.d/" directory.

- To run a Local or Neighbor Client, Server, the following files are require and must be create or copied to their appropriated directories on your server.

  Copy the **null.mc** file in the "/etc/" directory.
  Copy the **sendmail** file in the "/etc/sysconfig" directory.
  Copy the **sendmail** script file in the "/etc/rc.d/init.d/" directory.

You can obtain configuration files listed bellow on our floppy.tgz archive. Copy the following files from the decompressed floppy.tgz archive to their appropriated places or copy and paste them directly from this book to the concerned file.

The "/etc/mail/access and access.db" files for the Central Mail Hub

An "access" database can be created to accept or reject mail from selected domains. For example, you may choose to reject all mail originating from known spammers. Those files "access" and "access.db" are not require if you decide to use a central Mail Hub to handle all your mail. Also note that the use of a central Mail Hub will improve the security and the management of other server and client on your network that run sendmail.

Step 1
Create the **access** file (touch /etc/mail/access) and add the following lines:

```
# Description showing bellow for the format of this file comes from
# the Sendmail source distribution under "cf/README" file.
#
# The table itself uses e-mail addresses, domain names, and network
# numbers as keys. For example,
#
#       spammer@aol.com         REJECT
#       cyberspammer.com        REJECT
#       192.168.212             REJECT
#
# would refuse mail from spammer@aol.com, any user from cyberspammer.com
# (or any host within the cyberspammer.com domain), and any host on the
# 192.168.212.* network.
#
# The value part of the map can contain:
#
#       OK        Accept mail even if other rules in the
#                 running ruleset would reject it, for example,
#                 if the domain name is unresolvable.
#       RELAY     Accept mail addressed to the indicated domain or
#                 received from the indicated domain for relaying
#                 through your SMTP server.  RELAY also serves as
#                 an implicit OK for the other checks.
#       REJECT    Reject the sender or recipient with a general
#                 purpose message.
```

```
#       DISCARD    Discard the message completely using the
#                  $#discard mailer.  This only works for sender
#                  addresses (i.e., it indicates that you should
#                  discard anything received from the indicated
#                  domain).
#       ### any text    where ### is an RFC 821 compliant error code
#                  and "any text" is a message to return for
#                  the command.
#
# For example:
#
#       cyberspammer.com        550 We don't accept mail from spammers
#       okay.cyberspammer.com       OK
#       sendmail.org                OK
#       128.32                      RELAY
#
# would accept mail from okay.cyberspammer.com, but would reject mail
# from all other hosts at cyberspammer.com with the indicated message.
# It would allow accept mail from any hosts in the sendmail.org domain,
# and allow relaying for the 128.32.*.* network.
#
# You can also use the access database to block sender addresses based on
# the username portion of the address. For example:
#
#       FREE.STEALTH.MAILER@    550 Spam not accepted
#
# Note that you must include the @ after the username to signify that
# this database entry is for checking only the username portion of the
# sender address.
#
# If you use like we do in our "sendmail.mc macro configuration:
#
#       FEATURE(`blacklist_recipients')
#
# then you can add entries to the map for local users, hosts in your
# domains, or addresses in your domain which should not receive mail:
#
#       badlocaluser          550 Mailbox disabled for this username
#       host.mydomain.com     550 That host does not accept mail
#       user@otherhost.mydomain.com     550 Mailbox disabled for this recipient
#
# This would prevent a recipient of badlocaluser@mydomain.com, any
# user at host.mydomain.com, and the single address
# user@otherhost.mydomain.com from receiving mail. Enabling this
# feature will keep you from sending mails to all addresses that
# have an error message or REJECT as value part in the access map.
# Taking the example from above:
#
#       spammer@aol.com             REJECT
#       cyberspammer.com            REJECT
#
# Mail can't be sent to spammer@aol.com or anyone at cyberspammer.com.
#
# Now our configuration of access file,
# by default we allow relaying from localhost...
localhost.localdomain       RELAY
localhost                   RELAY
127.0.0.1                   RELAY
```

Step 2
Create the **access.db** file:

Remember, since “/etc/mail/access” is a database, after creating the text file as described above, you must use “makemap” program to create the database map.

- To create the “access database map”, use the following command:
  [root@deep]# **makemap hash /etc/mail/access.db < /etc/mail/access**

The “/etc/aliases and aliases.db” files for the Central Mail Hub

Aliasing is the process of converting one recipient name into another. One use is to convert a generic name (such as root) into a real username. Another is to convert one name into a list of many names (for mailing lists). For every envelope that lists a local user as a recipient, sendmail looks up that recipient’s name in the “aliases” file. Because sendmail may have to search through thousands of names in the “aliases” file, a version of the file is stored in a separate “db” database format file to significantly improve lookup speed. If you configure your sendmail to use a central server (Mail Hub) to handles all mail, you don’t need to install “aliases” and “aliases.db” files on the neighbor server or client machines.

Step 1
Create the **aliases** file (touch /etc/aliases) and add the following lines:

```
#
#     @(#)aliases    8.2 (Berkeley) 3/5/94
#
#  Aliases in this file will NOT be expanded in the header from
#  Mail, but WILL be visible over networks or from /bin/mail.
#
#     >>>>>>>>>>      The program "newaliases" must be run after
#     >> NOTE >>      this file is updated for any changes to
#     >>>>>>>>>>      show through to sendmail.
#

# Basic system aliases -- these MUST be present.
MAILER-DAEMON:      postmaster
postmaster:         root

# General redirections for pseudo accounts.
bin:            root
daemon:         root
nobody:         root

# Person who should get root's mail
#root:          admin
```

NOTE: Your aliases file is probably far more complex, but even so, note that the example shows minimum forms of aliases.

Step 2
Create the **aliases.db** file:
Since “/etc/aliases” is a database, after creating the text file as described above, you must use “makemap” program to create the database map.

- To create the “aliases database map”, use the following command:
  [root@deep]# **makemap hash /etc/aliases.db < /etc/aliases**

The "/etc/mail/virtusertable, domaintable, mailertable, and virtusertable.db, domaintable.db, mailertable.db" files for the Central Mail Hub

Some sites need to use multiple domain names when transitioning from an old domain to a new one. The domaintable feature enables such transitions to operate smoothly by rewriting the old domain to the new.

A virtusertable is a database that maps virtual domains into news addresses. A domain-specific form of aliasing, allowing multiple virtual domains to be hosted on one machine.

A mailertable is a database that maps "host.domain" names to special delivery agent and new domain name pairs. This can be used to override routing for particular domains.

- To create the **virtusertable, domaintable, mailertable, and their corresponding ".db"** files into "/etc/mail" directory, use the following commands:

```
[root@deep]# for map in virtusertable domaintable mailertable
> do
> touch /etc/mail/${map}
> chmod 0644 /etc/mail/${map}
> makemap hash /etc/mail/${map}.db < /etc/mail/${map}
> chmod 0644 /etc/mail/${map}.db
> done
```

The "/etc/sendmail.cw" file for the Central Mail Hub

The "/etc/sendmail.cw" file is read to obtain alternative names for the local host. One use for such a file might be to declare a list of hosts for which the local host is acting as the MX recipient. Also note that "sendmail.cw" file is requiring only on server that receive, forward and send mail like the central Mail Hub Server. On that machine we simply need to add the names of machines for which Mail Hub (mail.openarch.com) will handle mail to "/etc/sendmail.cw". Here is an example:

Create the **sendmail.cw** file (touch /etc/sendmail.cw) and add the following line:

```
# sendmail.cw - include all aliases for your machine here.
openarch.com
deep.openarch.com
www.openarch.com
win.openarch.com
mail.openarch.com
```

With this type of configuration, all mail sent will appear as if it were sent from "openarch.com", and any mail sent to "www.openarch.com" or the other hosts will be delivered to "mail.openarch.com" our mail Hub.

Please be aware that if you configure your system to masquerade as another any e-mail sent from your system to your system will be sent to the machine you are masquerading as. For example, in the above illustration, log files that are periodically sent to root@www.openarch.com by the cron daemon would be sent to root@mail.openarch.com.

The "/etc/sendmail.mc" file for the Central Mail Hub

A central Mail Hub can be a very busy machine, receiving and sending mail for many client machines, and because its role is broad, its configuration file is complex. Sendmail V8 uses the m4 macro preprocessor program to produce a Sendmail configuration file. The m4 program will produce the "/etc/sendmail.cf" configuration file by processing a file whose name ends in ".mc". We'll create this file (sendmail.mc) and put the necessary macros values in it to allow m4 program to processes (reads) its input and gathers definitions of macros, then replaces those macros with

their values and output the result to create our “sendmail.cf” file. Please refer to Sendmail documentation and README file under “cf” subdirectory of the V8 Sendmail source distribution for more information.

The “sendmail.cf” configuration file is the first file read by sendmail when it runs. Among the many items contained in that file are the locations of all the other files, the default permissions for those files and directories that sendmail needs. It contains options that modify sendmail’s behavior.

Step 1
Create the **sendmail.mc** file (touch /etc/sendmail.mc) and add the following lines:

```
divert(-1)
dnl This is the macro config file used to generate the /etc/sendmail.cf
dnl file. If you modify this file you will have to regenerate the
dnl /etc/sendmail.cf by running this macro config through the m4
dnl preprocessor:
dnl
dnl     cp sendmail.8.9.3.tar.gz /var/tmp
dnl     cd /var/tmp
dnl     tar xzpf sendmail.8.9.3.tar.gz
dnl     cd /var/tmp/sendmail-8.9.3/cf/cf
dnl     m4 ../m4/cf.m4 /etc/sendmail.mc > /etc/sendmail.cf
dnl
dnl You will need to have the sendmail source distribution for this to
dnl work.
divert(0)
define(`confDEF_USER_ID',``8:12'')
OSTYPE(`linux')
define(`confAUTO_REBUILD')
define(`confTO_CONNECT', `1m')
define(`confTRY_NULL_MX_LIST',true)
define(`confDONT_PROBE_INTERFACES',true)
define(`PROCMAIL_MAILER_PATH',`/usr/bin/procmail')
FEATURE(`smrsh',`/usr/sbin/smrsh')
FEATURE(mailertable)
FEATURE(`virtusertable',`hash -o /etc/mail/virtusertable')
FEATURE(redirect)
FEATURE(always_add_domain)
FEATURE(use_cw_file)
FEATURE(local_procmail)
FEATURE(nouucp)
MAILER(procmail)
MAILER(smtp)
FEATURE(`access_db')
FEATURE(`blacklist_recipients')
FEATURE(`rbl')
```

This tells sendmail.mc file to set itself up for this particular configuration setup with:

divert(-1) and divert(0)
The divert(-1) will delete the crud in the resulting output file and the divert(0) restores regular output.

define(`confDEF_USER_ID',``8:12'')
This configuration option specifies the default user id, in our case the user “mail” (see the /etc/passwd file).

OSTYPE(`linux’)

Support for various operating systems is supplied with the OSTYPE m4 command. Every “mc” file must declare the operating system with this command. This item is one of the minimal information require by the “mc” file.

define(`confAUTO_REBUILD')
This configuration option automatically rebuild alias file if needed.

define(`confTO_CONNECT', `1m')
This configuration option set the timeout waiting for an initial connect() to complete. This can only shorten connection timeouts; the kernel silently enforces an absolute maximum (which varies depending on the system).

define(`confTRY_NULL_MX_LIST',true)
This configuration option says that if the server is the best MX for a host and haven't made other arrangements, try connecting to the host directly.

define(`confDONT_PROBE_INTERFACES',true)
This configuration option if set, sendmail will _not_ insert the names and addresses of any local interfaces into the $=w class (list of known "equivalent" addresses).

define(`PROCMAIL_MAILER_PATH',`/usr/bin/procmail')
This configuration option set the path to the procmail program. This is also used by FEATURE(`local_procmail').

FEATURE(`smrsh',`/usr/sbin/smrsh')
This m4 macro enables the use of “smrsh” (sendmail restricted shell). Although Sendmail tries to be very safe about how it runs programs from the “/etc/aliases” and “~/.forward” files , it still may be vulnerable to some internal attacks. To limit the selection of programs that Sendmail is allowed to run, V8 of Sendmail now includes the “smrsh” (**S**end**m**ail **r**estricted **s**hell) program. This improves the ability of the local system administrator to control what gets run via e-mail. The default location for the “smrsh” program is “/usr/local/etc/smrsh”, since we are installed “smrsh” in another location, we need to add an argument to the smrsh feature for the new emplacement “/usr/sbin/smrsh”. The use of “smrsh” is recommended by CERT, so you are encouraged to use this feature as often as possible.

FEATURE(mailertable)
This m4 macro enables the use of “database selects new delivery agents”. A mailertable is a database that maps “host.domain” names to special delivery agent and new domain name pairs. The new domain names are used for routing but are not reflected in the headers of messages.

FEATURE(`virtusertable',`hash -o /etc/mail/virtusertable')
This m4 macro enables the use of “support for virtual domains”. A virtusertable is a database that maps virtual domains into new addresses.

FEATURE(redirect)
This m4 macro enables the use of “add support for address.REDIRECT”. The redirect feature allows aliases to be set up for retired accounts. Those aliases bounce with an indication of the new forwarding address. Note that the message is bounced and not forwarded. No notification is sent to the recipient’s new address.

FEATURE(always_add_domain)
This m4 macro enables the use of “add the local domain even on local mail”. The always_add_domain feature is safe and recommended. It ensures that all addresses that are locally delivered will be fully qualified.

FEATURE(use_cw_file)

This m4 macro enables the use of “use /etc/sendmail.cw file for local hostnames”. The use_cw_file feature causes the file “/etc/sendmail.cw” to be read to obtain alternative names for the local host. One use for such a file might be to declare a list of hosts for which the local host is acting as the MX recipient.

FEATURE(local_procmail)
This m4 macro enables the use of “use procmail as local delivery agent”. The procmail program can handle a user’s mail autonomously (for example, sorting incoming mail into folders based on subject) and can function as a Sendmail delivery agent.

FEATURE(nouucp)
This m4 macro enables the use of “eliminate all UUCP support”. If your site wants nothing to do with UUCP addresses, you can enable the nouucp feature. All the macros that relate to UUCP are ignored.

MAILER(procmail) and MAILER(smtp)
Delivery agents are not automatically declared. Instead, you must specify which ones you want to support and which ones to ignore with the MAILER m4 macro. MAILER(procmail) and MAILER(smtp) causes support for procmail, smtp, esmtp, smtp8 and relay delivery agents to be included.

FEATURE(`access_db')
Turns on the access database feature. The access db gives you the ability to allow or refuse to accept mail from specified domains for administrative reasons. For example, you may choose to reject all mail originating from known spammers.

FEATURE(`blacklist_recipients')
Turns on the ability to block incoming mail for certain recipient usernames, hostnames, or addresses. For example, you can block incoming mail to user nobody, host foo.mydomain.com, or guest@bar.mydomain.com.

FEATURE(`rbl')
This will cause sendmail to reject mail from any site in the Realtime Blackhole List database. If an argument is provided it is used as the name sever to contact; otherwise, the main RBL server at "rbl.maps.vix.com" is used, this is a database maintained in DNS of spammers. For details, see "http://maps.vix.com/rbl/".

NOTE: Sometimes, a domain may end up in the RBL list with which you wish to continue communications with. Perhaps it is vital for you to communicate with certain users at the black-listed domain. In this case, Sendmail allows you to override these domains to allow their e-mail to be received. Simply edit the "/etc/mail/access" file with the appropriate domain information. For example:

```
blacklisted.domain     OK
```

Step 2
Now that our macro configuration file “sendmail.mc” is create, we can build the sendmail configuration file “sendmail.cf” from these statements with the following commands:

```
[root@deep]# cd /var/tmp/sendmail-version/cf/cf/
[root@deep]# m4 ../m4/cf.m4 /etc/sendmail.mc > /etc/sendmail.cf
```

NOTE: Here, the “../m4/cf.m4” tells m4 program where to look for its default configuration file information.

The "/etc/null.mc" file for the local or neighbor client and server machines

Instead of having each individual server or workstation in a network handle its own mail, it can be advantageous to have powerful central server that handles all mail. Such a server is called a Mail Hub. The advantage of a central Mail Hub is:

- All incoming mail is sent to the hub, and no mail is sent directly to a client machine.
- All outgoing mail from clients is sent to the Hub, and the Hub then forwards that mail to its ultimate destination.
- All outgoing mail appears to come from a single server and no client's name needs to be known to the outside world.
- No client needs to run a sendmail daemon to listen for mail.

Step 1
Since our local clients machines never receive mail directly and send, relay all their mail through the Mail Hub server, we will create a special file called "null.mc", which, when later processed, will create a customized "sendmail.cf" configuration file that respond to this special setup for our neighbor or local server, client machines.

Create the **null.mc** file (touch /etc/null.mc) and add the following lines:

```
divert(-1)
dnl This is the macro config file used to generate the /etc/sendmail.cf
dnl file. If you modify this file you will have to regenerate the
dnl /etc/sendmail.cf by running this macro config through the m4
dnl preprocessor:
dnl
dnl     cp sendmail.8.9.3.tar.gz /var/tmp
dnl     cd /var/tmp
dnl     tar xzpf sendmail.8.9.3.tar.gz
dnl     cd /var/tmp/sendmail-8.9.3/cf/cf
dnl     m4 ../m4/cf.m4 /etc/null.mc > /etc/sendmail.cf
dnl
dnl You will need to have the sendmail source distribution for this to
dnl work.
divert(0)
OSTYPE(`linux')
FEATURE(`nullclient',`mail.openarch.com')
undefine(`ALIAS_FILE')
```

This tells null.mc file to set itself up for this particular configuration setup with:

divert(-1) and divert(0)
The divert(-1) will delete the crud in the resulting output file and the divert(0) restores regular output.

OSTYPE(`linux')
Support for various operating systems is supplied with the OSTYPE m4 command. Every "mc" file must declare the operating system with this command. This item is one of the minimal information require by the "mc" file.

FEATURE(`nullclient',`mail.openarch.com')
This configuration option set your server or client to never receive mail directly, send their mail as though the Central Mail Hub and they relay all mail through that server rather than sending directly. This is a special case -- it creates a stripped down configuration file containing nothing but support for forwarding all mail to a central hub via a local SMTP-based network. The argument is the name of that hub. The argument `**mail.openarch.com**' is the canonical name of

the Mail Hub. You should, of course, change this canonical name to reflect your Mail Hub Server for example: FEATURE(`nullclient',` **my.mailhub.com**').

undefine(`ALIAS_FILE')
This configuration option prevent the nullclient version of sendmail from trying to access "/etc/aliases" and "/etc/aliases.db" files. With the adding of this line, you don't need to have "aliases" file on all your internal client machines. Aliases file is require only on the Mail Hub Server for all server and client aliases on the network.

Now that our macro configuration file "null.mc" is create, we can build the sendmail configuration file "sendmail.cf" from these statements in all our neighbor server, client machines with the following commands:

[root@client]# **cd /var/tmp/sendmail-version/cf/cf/**
[root@client]# **m4 ../m4/cf.m4 /etc/null.mc > /etc/sendmail.cf**

Step 2
No mail should ever again be delivered to your local machine. Since there will be no incoming mail connections, you no longer needed to run a sendmail daemon on your neighbor or local server, client machines.

To stop sendmail daemon to run on your neighbor or local server, client machines, edit the "/etc/sysconfig/sendmail" file and change the line that read:

DAEMON=yes
To read:
DAEMON=no

NOTE: The "QUEUE=1h" under "/etc/sysconfig/sendmail" file cause sendmail to process the queue once every 1 hour. We leave that line in place because sendmail still needs to process the queue periodically in case the Mail Hub is down.

Step 3
Local machines never use aliases, access, or other maps database. Since all maps file database are located and used on the Central Mail Hub Server for all local machines we may have on the network, we can safety remove the following command and man pages from all our local machines.

/usr/bin/newaliases
/usr/man/man1/newaliases.1
/usr/man/man5/aliases.5

- To remove the following files from your system, use the command:
  [root@client]# **rm -f /usr/bin/newaliases**
  [root@client]# **rm -f /usr/man/man1/newaliases.1**
  [root@client]# **rm -f /usr/man/man5/aliases.5**

Step 4
Remove the unnecessary Procmail program from all your local sendmail server, client. Since local machines send all internal and outgoing mail to the mail Hub Server for future delivery, we don't need to use complex local delivery agent program like Procmail to do the job. Instead we can use the "/bin/mail" program.

- To remove Procmail from your system, use the command:
  [root@client]# **rpm -e procmail**

Configuration of the "/etc/sysconfig/sendmail" file for all configuration

The "/etc/sysconfig/sendmail" file is used to specify SENDMAIL configuration information like if sendmail must run as a daemon and listen for mail or not, how must time to wait before sending a warning if messages in queue directory has not been delivered.

Create the **sendmail** file (touch /etc/sysconfig/sendmail) and add:

```
DAEMON=yes
QUEUE=1h
```

The "DAEMON=yes" instruct sendmail to run as a daemon. This line is useful when sendmail client machines are configured to not accept mail from outside and forward all local mail to a central hub, and not run as a daemon for better security. If you are configured your server or client machines in this way, all you have to do is to replace the "DAEMON=yes" to "DAEMON=no".

Mail is usually placed into the queue because it could not be transmitted immediately. The "QUEUE=1h" set the time interval before sends a warning to the sender, if the messages has not been delivered.

Configuration of the "/etc/rc.d/init.d/sendmail" script file for all configuration

Configure your "/etc/rc.d/init.d/sendmail" script file to start and stop Sendmail daemon Server.

Create the **sendmail** script file (touch /etc/rc.d/init.d/sendmail) and add:

```
#!/bin/sh
#
# sendmail     This shell script takes care of starting and stopping
#              sendmail.
#
# chkconfig: 2345 80 30
# description: Sendmail is a Mail Transport Agent, which is the program \
#              that moves mail from one machine to another.
# processname: sendmail
# config: /etc/sendmail.cf
# pidfile: /var/run/sendmail.pid

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

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

# Source sendmail configuration.
if [ -f /etc/sysconfig/sendmail ] ; then
      . /etc/sysconfig/sendmail
else
      DAEMON=yes
      QUEUE=1h
fi

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

[ -f /usr/sbin/sendmail ] || exit 0
```

```
RETVAL=0

# See how we were called.
case "$1" in
  start)
        # Start daemons.

        echo -n "Starting sendmail: "
        /usr/bin/newaliases > /dev/null 2>&1
        for i in virtusertable access domaintable mailertable ; do
           if [ -f /etc/mail/$i ] ; then
              makemap hash /etc/mail/$i < /etc/mail/$i
           fi
        done
        daemon /usr/sbin/sendmail $([ "$DAEMON" = yes ] && echo -bd) \
                        $([ -n "$QUEUE" ] && echo -q$QUEUE)
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/sendmail
        ;;
  stop)
        # Stop daemons.
        echo -n "Shutting down sendmail: "
        killproc sendmail
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/sendmail
        ;;
  restart|reload)
        $0 stop
        $0 start
        RETVAL=$?
        ;;
  status)
        status sendmail
        RETVAL=$?
        ;;
  *)
        echo "Usage: sendmail {start|stop|restart|status}"
        exit 1
esac

exit $RETVAL
```

Now, make this script executable and change its default permission:
[root@deep]# **chmod 700 /etc/rc.d/init.d/sendmail**

Create the symbolic rc.d links for Sendmail with the command:
[root@deep]# **chkconfig --add sendmail**

Start your Sendmail Server manually with the following command
[root@deep]# **/etc/rc.d/init.d/sendmail start**

Cleanup after work

[root@deep]# cd /var/tmp
[root@deep]# rm -rf sendmail-version/ sendmail.version.tar.gz

The “rm” command will remove all the source files we have used to compile and install sendmail. It will also remove the Sendmail compressed archive from the “/var/tmp” directory.

Securing Sendmail

The Sendmail restricted shell "smrsh"

The smrsh program is intended as a replacement for "/bin/sh" in the program mailer definition of sendmail. The smrsh software is a restricted shell utility that provides the ability to specify, through the "/etc/smrsh" directory, an explicit list of executable programs. Briefly, even if a "bad guy" can get sendmail to run a program without going through an alias or forward file, smrsh limits the set of programs that he or she can execute. When used in conjunction with sendmail, smrsh effectively limits sendmail's scope of program execution to only those programs specified in smrsh's directory. If you are follow what we do above, smrsh program is already compiled and installed on your computer under "/usr/sbin/smrsh".

Step 1
The first thing we need to do is to determine the list of commands that "smrsh" should allow sendmail to run. By default we include but not limited to:

"/bin/mail" (if you have it installed on your system)
"/usr/bin/procmail" (if you have it installed on your system)

NOTE: You should NOT include interpreter programs such as sh(1), csh(1), perl(1), uudecode(1) or the stream editor sed(1) in your list of acceptable commands.

Step 2
You will next need to populate the "/etc/smrsh" directory with the programs that are allowable for sendmail to execute. To prevent duplicate programs and make a nice job, it is better to establish links to the allowable programs from "/etc/smrsh" rather than copy programs to this directory.

- To allow the **mail** program "/bin/mail", use the command:
  [root@deep]# **cd /etc/smrsh**
  [root@deep]# **ln -s /bin/mail mail**

- To allow the **procmail** program "/usr/bin/procmail", use the command:
  [root@deep]# **cd /etc/smrsh**
  [root@deep]# **ln -s /usr/bin/procmail procmail**

Would allow the **mail** and **procmail** programs to be run from a user's ".forward" file or an "aliases" which uses the "|program" syntax.

Step 3
We can now configure sendmail to use the restricted shell. The program mailer is defined by a single line in the sendmail configuration file, "/etc/sendmail.cf". You must modify this single line "Mprog" definition in the sendmail.cf file, by replacing the "/bin/sh" specification with "/usr/sbin/smrsh".

Edit the sendmail.cf file (vi /etc/sendmail.cf) and change the line:

For example:
Mprog, P=/bin/sh, F=lsDFMoqeu9, S=10/30, R=20/40, D=$z:/, T=X-Unix, A=sh -c $u
Which should be changed to:
Mprog, P=**/usr/sbin/smrsh**, F=lsDFMoqeu9, S=10/30, R=20/40, D=$z:/, T=X-Unix, A=sh -c $u

- Now re-start the sendmail process manually with the following command:
  [root@deep]# **/etc/rc.d/init.d/sendmail restart**

NOTE: In our “sendmail.mc” for the Mail Hub Server configuration file above, we are already configured this line “Mprog” to use the restricted shell “/usr/sbin/smrsh” with the m4 macro “FEATURE(`smrsh',`/usr/sbin/smrsh')”, so don’t be surprised if you see the “/usr/sbin/smrsh” specification is already set in your “/etc/sendmail.cf” file for the Mail Hub relay. Instead use the technique show above for other “/etc/sendmail.cf” files in your network like the one for the nullclient “local or neighbor client and server machines” that use the “/etc/null.mc” macro configuration file to generate the “/etc/sendmail.cf” file.

The “/etc/aliases” file

The aliases file can easily be used to gain privileged status if it wrongly or carelessly administered. For example, many vendors used to ship systems with a “decode” alias in the aliases file. This practice is becoming less common.

The intention is to provide an easy way for users to transfer binary files using mail. At the sending site the user converts the binary to ASCII with “uuencode”, then mails the result to the “decode” alias at the receiving site. That alias pipes the mail message through the “/usr/bin/uuencode” program, which converts the ASCII back into the original binary file.

Remove the “decode” alias. Similarly, every alias that executes a program -that you did not place there yourself and check completely- should be questioned and probably removed. For this change to take effect you will need to run:

[root@deep]# **/usr/bin/newaliases**

Edit the aliases file (vi /etc/aliases) and remove the following lines:

```
# Basic system aliases -- these MUST be present.
MAILER-DAEMON:  postmaster
postmaster:     root

# General redirections for pseudo accounts.
bin:            root
daemon:         root
games:          root ← remove this line.
ingres:         root ← remove this line.
nobody:         root
system:         root ← remove this line.
toor:           root ← remove this line.
uucp:           root ← remove this line.

# Well-known aliases.
manager:        root ← remove this line.
dumper:         root ← remove this line.
operator:       root ← remove this line.

# trap decode to catch security attacks
decode:         root ← remove this line.

# Person who should get root's mail
#root:          marc
```

Don’t forget to run “**/usr/bin/newaliases”** for this change to take effect.

Prevent your Sendmail being abused by unauthorized users
The very latest versions of Sendmail (8.9.3) include powerful Anti-Spam features, which can help prevent your mail server being abused by unauthorized users. To do that, edit your "/etc/sendmail.cf" file and make a change to the configuration file to block off spammers.

Edit the sendmail.cf file (vi /etc/sendmail.cf) and change the line:

O PrivacyOptions=authwarnings
To read:
O PrivacyOptions=authwarnings,**noexpn,novrfy**

Setting "noexpn" causes sendmail to disallow all SMTP "EXPN" commands, it also causes sendmail to reject all SMTP "VERB" commands. Setting "novrfy" causes sendmail to disallow all SMTP "VRFY" commands. The change prevents spammers from using the ``EXPN'' and ``VRFY'' commands in sendmail. Unethical individuals too often abuse these commands.

The SMTP greeting message
When sendmail accepts an incoming SMTP connection it sends a greeting message to the other host. This message identifies the local machine and is the first thing it sends to say it is ready.

Edit the sendmail.cf file (vi /etc/sendmail.cf) and change the line:

O SmtpGreetingMessage=$j Sendmail $v/$Z; $b
To read:
O SmtpGreetingMessage=$j Sendmail $v/$Z; $b **NO UCE C=xx L=xx**

- Now re-start the sendmail process manually for the change to take effect:
  [root@deep]# **/etc/rc.d/init.d/sendmail restart**

The change modifies the banner, which Sendmail displays upon receiving a connection. You should replace the ``xx'' in the ``C=xx L=xx'' entries with your country and location codes. For example, in my case, I would use ``C=CA L=QC'' for Canada, Quebec. The latter change doesn't actually affect anything, but was recommended by folks in the "news.admin.net-abuse.email" newsgroup as a legal precaution.

Restrict who may examine the queue's contents
Ordinarily, anyone may examine the mail queue's contents by using the "mailq" command. To restrict who may examine the queue's contents, specify "restrictmailq" in the "/etc/sendmail.cf" file. With this option, sendmail allow only users who are in the same group as the group ownership of the queue directory (root) to examine the contents. This allows the queue directory to be fully protected with mode 0700 yet for selected users to still be able to see its contents.

Edit the sendmail.cf file (vi /etc/sendmail.cf) and change the line:

O PrivacyOptions=authwarnings,noexpn,novrfy
To read:
O PrivacyOptions=authwarnings,noexpn,novrfy,**restrictmailq**

- Now we change the mode of our queue directory to be fully protected:
  [root@deep]# **chmod 0700 /var/spool/mqueue**

NOTE: We are already added "noexpn and novrfy" option to our line "PrivacyOptions=" in sendmail.cf file. Now we continue by adding the "restrictmailq" option to this line.

Any no privileged user who attempts to examine the mail queue content will get this message:

[user@deep]$ **/usr/bin/mailq**
You are not permitted to see the queue

Limit queue processing to "root"
Ordinarily, anyone may process the queue with the "-q" switch. To limit queue processing to "root" and the owner of the queue directory, specify "restrictqrun" in the "/etc/sendmail.cf" file.

Edit the sendmail.cf file (vi /etc/sendmail.cf) and change the line:

O PrivacyOptions=authwarnings,noexpn,novrfy,restrictmailq
To read:
O PrivacyOptions=authwarnings,noexpn,novrfy,restrictmailq,**restrictqrun**

Any no privileged user who attempts to process the queue will get this message:
[user@deep]$ **/usr/sbin/sendmail -q**
You do not have permission to process the queue

Set the immutable bit on important Sendmail files
Important Sendmail files can be set immutable for better security with the "chattr" command. A file with the "+i" attribute cannot be modified, it cannot be deleted or renamed, no link can be created to this file and no data can be written to the file. Only the superuser can set or clear this attribute.

Set the immutable bit on "sendmail.cf" file:
[root@deep]# **chattr +i /etc/sendmail.cf**

Set the immutable bit on "sendmail.cw" file:
[root@deep]# **chattr +i /etc/sendmail.cw**

Set the immutable bit on "sendmail.mc" file:
[root@deep]# **chattr +i /etc/sendmail.mc**

Set the immutable bit on "null.mc" file:
[root@deep]# **chattr +i /etc/null.mc**

Set the immutable bit on "aliases" file:
[root@deep]# **chattr +i /etc/aliases**

Set the immutable bit on "access" file:
[root@deep]# **chattr +i /etc/mail/access**

Further documentation
For more details, there are several man pages you can read:

$ man aliases (5)	- aliases file for sendmail
$ man makemap (8)	- create database maps for sendmail
$ man sendmail (8)	- an electronic mail transport agent
$ man mailq (1)	- print the mail queue
$ man newaliases (1)	- rebuild the data base for the mail aliases file
$ man mailstats (8)	- display mail statistics
$ man praliases (8)	- display system mail aliases

Sendmail Administrative Tools
The commands listed bellows are some that we use often in our regular use but much more exist and you must check the man page and documentation for more details and information.

newaliases
Newaliases rebuilds the random access database for the mail aliases file “/etc/aliases”. It must be run each time this file is changed in order for the change to take effect. The “newaliases” command is identical to “sendmail -bi” command.

- To run the newaliases, use the command:
  [root@deep]# **/usr/bin/newaliases**

makemap
Makemap creates the database maps used by the keyed map lookups in sendmail. It reads input from the standard input and outputs them to the indicated mapname. The makemap command must be used only when you need to create a new database for file like aliases, access, or domaintable, mailertable, and virtusertable.

- To run the makemap to create new database for access, use the command:
  [root@deep]# **makemap hash /etc/mail/access.db < /etc/mail/access**

Where <hash> is the database format, makemap can handles up to three different database formats, They may be “hash”, “btree” and “dbm”. The </etc/mail/access.db> is the location and the name of the new database that will be crested. The </etc/mail/access> is the location of the file from where makemap will read from the standard input file.

mailq
The mailq utility prints a summary of the mail messages queued for future delivery.

- To print a summary of the mail messages, use the command:
  [root@deep]# **mailq**

Sendmail Users Tools

The commands listed bellows are some that we use often in our regular use but much more exist and you must check the man page and documentation for more details and information.

mailstats
The mailstats utility displays the current mail statistics.

- To displays the current mail statistics, use the command:
  [root@deep]# **mailstats**
  Statistics from Tue Dec 14 20:31:48 1999
  M msgsfr bytes_from msgsto bytes_to msgsrej msgsdis Mailer
  8 7 7K 7 7K 0 0 local
  ==========
  T 7 7K 7 7K 0 0

praliases
The praliases utility displays the current system aliases, one per line, in no particular order.

- To displays the current system aliases, use the command:
  [root@deep]# **praliases**
  postmaster:root
  daemon:root
  root:admin

```
@:@
mailer-daemon:postmaster
bin:root
nobody:root
www:root
```

Installed files for Sendmail Central Mail Hub

```
> /etc/rc.d/init.d/sendmail
> /etc/rc.d/rc0.d/K30sendmail
> /etc/rc.d/rc1.d/K30sendmail
> /etc/rc.d/rc2.d/S80sendmail
> /etc/rc.d/rc3.d/S80sendmail
> /etc/rc.d/rc4.d/S80sendmail
> /etc/rc.d/rc5.d/S80sendmail
> /etc/rc.d/rc6.d/K30sendmail
> /etc/sysconfig/sendmail
> /etc/mail
> /etc/mail/access
> /etc/mail/virtusertable
> /etc/mail/virtusertable.db
> /etc/mail/domaintable
> /etc/mail/domaintable.db
> /etc/mail/mailertable
> /etc/mail/mailertable.db
> /etc/mail/access.db
> /etc/smrsh
> /etc/aliases
> /etc/sendmail.cw
> /etc/sendmail.mc
> /etc/aliases.db
> /etc/sendmail.cf
> /usr/bin/newaliases
> /usr/bin/mailq
> /usr/bin/hoststat
> /usr/bin/purgestat
> /usr/lib/sendmail.hf
> /usr/lib/sendmail
> /usr/man/man1/mailq.1
> /usr/man/man1/newaliases.1
> /usr/man/man5/aliases.5
> /usr/man/man8/sendmail.8
> /usr/man/man8/mailstats.8
> /usr/man/man8/makemap.8
> /usr/man/man8/praliases.8
> /usr/man/man8/smrsh.8
> /usr/sbin/sendmail
> /usr/sbin/mailstats
> /usr/sbin/makemap
> /usr/sbin/praliases
> /usr/sbin/smrsh
> /var/log/sendmail.st
> /var/spool/mqueue
```

Installed files for Sendmail local server or client

```
> /etc/rc.d/init.d/sendmail
> /etc/rc.d/rc0.d/K30sendmail
> /etc/rc.d/rc1.d/K30sendmail
> /etc/rc.d/rc2.d/S80sendmail
> /etc/rc.d/rc3.d/S80sendmail
> /etc/rc.d/rc4.d/S80sendmail
> /etc/rc.d/rc5.d/S80sendmail
> /etc/rc.d/rc6.d/K30sendmail
> /etc/sysconfig/sendmail
> /etc/null.mc
> /etc/sendmail.cf
> /usr/bin/newaliases
> /usr/bin/mailq
> /usr/bin/hoststat
> /usr/bin/purgestat
> /usr/lib/sendmail.hf
> /usr/lib/sendmail
> /usr/man/man1/mailq.1
> /usr/man/man1/newaliases.1
> /usr/man/man5/aliases.5
> /usr/man/man8/sendmail.8
> /usr/man/man8/mailstats.8
> /usr/sbin/sendmail
> /usr/sbin/mailstats
> /var/log/sendmail.st
> /var/spool/mqueue
```

Linux OPENSSL Server

Overview

The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, fully featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols with full-strength cryptography.

The project is managed by a worldwide community of volunteers that use the Internet to communicate, plan, and develop the OpenSSL toolkit and its related documentation.

Cryptography Advantages

Data Confidentiality
When a message is encrypted, the input plaintext is transformed by an algorithm into enciphered text that hides the meaning of the message. This process involves a secret key that is used to encrypt and later decrypt the data. Without the secret key, the encrypted data is meaningless. With cryptography, only the secret data encryption key has to be transmitted by a secure method; the encrypted text can be sent via any public mechanism.

Data Integrity
Cryptography is also used to protect the integrity of data. For example, a cryptographic checksum, called a message authentication code (MAC), can be calculated on arbitrary user supplied text. The text and MAC are then sent to the receiver. The receiver of the message can verify the trial MAC appended to a message by recalculating the MAC for the message, using the appropriate secret key and verifying that it exactly equals the trial MAC.

Authentication
Another use of cryptography is in personal identification, where the user knows a secret which can serve to authenticate his identity.

Patents

Various companies hold various patents for various algorithms in various locations around the world. _YOU_ are responsible for ensuring that your use of any algorithms is legal by checking if there are any patents in your country. The file contains some of the patents that we know about or are rumored to exist. This is not a definitive list.

RSA Data Security holds software patents on the RSA and RC5 algorithms. If their ciphers are used inside the USA (and Japan?), you must contact RSA Data Security for licensing conditions. Their web page is http://www.rsa.com/.

RC4 is a trademark of RSA Data Security, so use of this label should perhaps only be used with RSA Data Security's permission.

The IDEA algorithm is patented by Ascom in Austria, France, Germany, Italy, Japan, Netherlands, Spain, Sweden, Switzerland, UK and the USA. They should be contacted if that algorithm is to be used; their web page is http://www.ascom.ch/.

These installation instructions assume

Commands are Unix-compatible.
The source path is “/var/tmp” (other paths are possible).
Installations were tested on RedHat Linux 6.1.
All steps in the installation will happen in superuser account “root”.
OpenSSL version number is 0.9.4

Tarballs

It is a good idea to make a list of files on the system before you install Openssl, and one afterwards, and then compare them using ‘**diff**’ to find out what file it placed where. Simply run

'**find /* > ssl1**' before and '**find /* > ssl2**' after you install the software, and use '**diff ssl1 ssl2 > ssl**' to get a list of what changed.

Packages

OpenSSL Homepage: http://www.openssl.org/
You must be sure to download: openssl-0.9.4.tar.gz

Compilation

Decompress the tarball (tar.gz).

```
[root@deep]# cp openssl_version.tar.gz /var/tmp
[root@deep]# cd /var/tmp
[root@deep]# tar xzpf openssl_version.tar.gz
```

Compile and Optimize

Cd into the new Openssl directory and type the following commands on your terminal:

Step 1
Edit the **c_rehash** file (vi +11 tools/c_rehash) and change the line:

```
DIR=/usr/local/ssl
```
To read:
DIR=/usr

The changed line above will build and install OpenSSL in the default location "/usr".

Step 2
By default OpenSSL source files suppose that your Perl program directory is located under the "/usr/local/bin/perl" directory. We must modify the "#!/usr/local/bin/perl" line in all scripts that rely on perl to reflect our Perl directory under Red Hat Linux to be "/usr/bin".

[root@deep]# **perl util/perlpath.pl /usr/bin** (where your perl program reside).

Step 3
OpenSSL must to know where to find the necessary source libraries of OpenSSL to compile successfully it require files. With the command bellow, we set the PATH ENVIRONMENT VARIABLE to the default directory where we are uncompressed the OpenSSL source files.

[root@deep]# **export LD_LIBRARY_PATH=`pwd`**

```
CC="egcs" \
./Configure linux-elf -DSSL_FORBID_ENULL \
--prefix=/usr  \
--openssldir=/etc/ssl
```

NOTE: The "-DSSL_FORBID_ENULL" option is requiring for not allowing null encryption for security reasons.

Edit the **Makefile.ssl** file (vi +52 Makefile.ssl) and add:

CFLAG= -DTHREADS -D_REENTRANT -DSSL_FORBID_ENULL -DL_ENDIAN -DTERMIO **-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions** -Wall -DSHA1_ASM -DMD5_ASM -DRMD160_ASM

This is our optimization flag for the compilation of OpenSSL software on the server.

Edit the **Makefile.ssl** file (vi +77 Makefile.ssl) and add:

PROCESSOR= **686**

NOTE: If you have a Pentium, put: 586, a PentiumPro/II/III, put 686, a 486, put 486.

```
[root@deep]# make -f Makefile
[root@deep]# make test
[root@deep]# make install

[root@deep]# mv /etc/ssl/misc/*  /usr/bin/
[root@deep]# rm -rf  /etc/ssl/misc/
[root@deep]# rm -rf  /etc/ssl/lib/
[root@deep]# rm -f  /usr/bin/CA.pl
[root@deep]# rm -f  /usr/bin/CA.sh
[root@deep]# install -m 644 libRSAglue.a  /usr/lib/
[root@deep]# install -m 644 rsaref/rsaref.h  /usr/include/openssl/
[root@deep]# strip /usr/bin/openssl
[root@deep]# mkdir -p /etc/ssl/crl
```

The "**make -f**" command will build the OpenSSL libraries (libcrypto.a and libssl.a) and the OpenSSL binary "openssl". The libraries will be built in the top-level directory, and the binary will be in the "apps" directory.

After a successful build, the "**make test**" will test the libraries and finaly the "**make install**" will create the installation directory and install OpenSSL.

The "**mv**" command would move all files under the "/etc/ssl/misc/" directory to the "/usr/bin/" directory. These files are binary and must be located under "/usr/bin/" since in our system, all binary files are keep in this directory. Also putting these files in the "/usr/bin/" directory will keep them on our PATH ENVIRONMENT VARIABLE.

The "**rm**" command would remove the "/etc/ssl/misc/" and "/etc/ssl/lib/" directories from our system since files that was on these directories are now located in other place. Also it will remove the "CA.pl" and "CA.sh" files that are a small scripts used to create you own CA certificate. Those scripts related to "openssl ca" commands has some strange requirements and the default OpenSSL config doesn't allow one easily to use ``openssl ca'' directly. So we'll create the "sign.sh" script program later to replace them.

Cleanup after work

```
[root@deep]# cd /var/tmp
[root@deep]# rm -rf openssl-version/ openssl_version.tar.gz
```

The "rm" command will remove all the source files we have used to compile and install OpenSSL. It will also remove the OpenSSL compressed archive from the "/var/tmp" directory.

Configuration

All software we describe in our book "Linuxsos.pdf" will have a specific directory and subdirectory in a tar compressed archive named “floppy.tgz” containing file configurations for the specific program. If you get this archive file, you wouldn’t be obliged to reproduce the different configuration files bellow manually or cut and past them to create your configuration files. Whatever your decide to copy manually or get the files made to your convenience from the archive compressed files, it will be to your responsibility to modify, adjust for your needs and place the files related to OpenSSL software to their appropriated places on your server machine, like show bellow. The server configuration files archive is located at the following Internet address: http://pages.infinit.net/lotus1/doc/opti/floppy.tgz

- To run OpenSSL Server, the following files are require and must be create or copied to their appropriated directories on your server.

  Copy the **openssl.cnf** file to the “/etc/ssl/” directory.
  Copy the **sign.sh** script file to the “/usr/bin/” directory.

You can obtain configuration files listed bellow on our floppy.tgz archive. Copy the following files from the decompressed floppy.tgz archive to their appropriated places or copy and paste them directly from this book to the concerned file.

Configuration of the “/etc/ssl/openssl.cnf” file

This is the general configuration file for openssl program where you can configure, expiration date of your keys, the name of your organization, the address etc. The configurations you must change will be in the **[CA_default]** and **[req_distinguished_name]** sections.

Edit the **openssl.cnf** file (vi /etc/ssl/openssl.cnf) and add or modify:

```
# OpenSSL example configuration file.
# This is mostly being used for generation of certificate requests.
#

RANDFILE                = $ENV::HOME/.rnd
oid_file                = $ENV::HOME/.oid
oid_section             = new_oids

# To use this configuration file with the "-extfile" option of the
# "openssl x509" utility, name here the section containing the
# X.509v3 extensions to use:
# extensions            =
# (Alternatively, use a configuration file that has only
# X.509v3 extensions in its main [= default] section.)

[ new_oids ]

# We can add new OIDs in here for use by 'ca' and 'req'.
# Add a simple OID like this:
# testoid1=1.2.3.4
# Or use config file substitution like this:
# testoid2=${testoid1}.5.6

####################################################################
[ ca ]
default_ca      = CA_default            # The default ca section

####################################################################
[ CA_default ]

dir             = /etc/ssl              # Where everything is kept
```

```
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/ca.db.index      # database index file.
new_certs_dir   = $dir/ca.db.certs      # default place for new certs.

certificate     = $dir/certs/ca.crt     # The CA certificate
serial          = $dir/ca.db.serial     # The current serial number
crl             = $dir/crl.pem          # The current CRL
private_key     = $dir/private/ca.key   # The private key
RANDFILE        = $dir/ca.db.rand       # private random number file

x509_extensions = usr_cert              # The extentions to add to the cert

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crl_extensions = crl_ext

default_days    = 365                   # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md      = md5                   # which md to use.
Preserve        = no                    # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy          = policy_match

# For the CA policy
[ policy_match ]
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

####################################################################
[ req ]
default_bits            = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions = v3_ca # The extentions to add to the self signed cert

[ req_distinguished_name ]
countryName             = Country Name (2 letter code)
countryName_default     = CA
countryName_min         = 2
countryName_max         = 2
```

```
stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Quebec

localityName                    = Locality Name (eg, city)
localityName_default            = Montreal

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = Open Network Architecture

# we can do this but it is not needed normally :-)
#1.organizationName             = Second Organization Name (eg, company)
#1.organizationName_default     = World Wide Web Pty Ltd

organizationalUnitName          = Organizational Unit Name (eg, section)
organizationalUnitName_default  = Internet Department

commonName                      = Common Name (eg, YOUR name)
commonName_default              = www.openarch.com
commonName_max                  = 64

emailAddress                    = Email Address
emailAddress_default            = admin@openarch.com
emailAddress_max                = 40

# SET-ex3                       = SET extension number 3

[ req_attributes ]
challengePassword               = A challenge password
challengePassword_min           = 4
challengePassword_max           = 20

unstructuredName                = An optional company name

[ usr_cert ]

# These extensions are added when 'ca' signs a request.

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.

basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.

# This is OK for an SSL server.
# nsCertType                    = server

# For an object signing certificate this would be used.
# nsCertType = objsign

# For normal client use this is typical
# nsCertType = client, email

# and for everything including object signing:
# nsCertType = client, email, objsign

# This is typical in keyUsage for a client certificate.
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# This will be displayed in Netscape's comment listbox.
nsComment                       = "OpenSSL Generated Certificate"
```

```
# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always

# This stuff is for subjectAltName and issuerAltname.
# Import the email address.
# subjectAltName=email:copy

# Copy subject details
# issuerAltName=issuer:copy

#nsCaRevocationUrl              = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName

[ v3_ca]

# Extensions for a typical CA


# PKIX recommendation.

subjectKeyIdentifier=hash

authorityKeyIdentifier=keyid:always,issuer:always

# This is what PKIX recommends but some broken software chokes on critical
# extensions.
#basicConstraints = critical,CA:true
# So we do this instead.
basicConstraints = CA:true

# Key usage: this is typical for a CA certificate. However since it will
# prevent it being used as an test self-signed certificate it is best
# left out by default.
# keyUsage = cRLSign, keyCertSign

# Some might want this also
# nsCertType = sslCA, emailCA

# Include email address in subject alt name: another PKIX recommendation
# subjectAltName=email:copy
# Copy issuer details
# issuerAltName=issuer:copy

# RAW DER hex encoding of an extension: beware experts only!
# 1.2.3.5=RAW:02:03
# You can even override a supported extension:
# basicConstraints= critical, RAW:30:03:01:01:FF

[ crl_ext ]
# CRL extensions.
# Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.

# issuerAltName=issuer:copy
authorityKeyIdentifier=keyid:always,issuer:always
```

NOTE: This file “openssl.cnf” already exist on your server when you compile and install OpenSSL program and can be found under “/etc/ssl/” directory. You don’t need to change all the default options set in this file, the configurations you must usually change will be in the **[CA_default]** and **[req_distinguished_name]** sections only.

Create the “/usr/bin/sign.sh” program file

The “openssl ca” commands has some strange requirements and the default OpenSSL config doesn't allow one easily to use “openssl ca" directly. So we’ll create this “sign.sh” program to replace it.

Create the **sign.sh** program file (touch /usr/bin/sign.sh) and add on this file:

```
#!/bin/sh
##
##  sign.sh -- Sign a SSL Certificate Request (CSR)
##  Copyright (c) 1998-1999 Ralf S. Engelschall, All Rights Reserved.
##

#   argument line handling
CSR=$1
if [ $# -ne 1 ]; then
    echo "Usage: sign.sign <whatever>.csr"; exit 1
fi
if [ ! -f $CSR ]; then
    echo "CSR not found: $CSR"; exit 1
fi
case $CSR in
   *.csr ) CERT="`echo $CSR | sed -e 's/\.csr/.crt/'`" ;;
      * ) CERT="$CSR.crt" ;;
esac

#   make sure environment exists
if [ ! -d ca.db.certs ]; then
    mkdir ca.db.certs
fi
if [ ! -f ca.db.serial ]; then
    echo '01' >ca.db.serial
fi
if [ ! -f ca.db.index ]; then
    cp /dev/null ca.db.index
fi

#   create an own SSLeay config
cat >ca.config <<EOT
[ ca ]
default_ca              = CA_own
[ CA_own ]
dir                     = /etc/ssl
certs                   = /etc/ssl/certs
new_certs_dir           = /etc/ssl/ca.db.certs
database                = /etc/ssl/ca.db.index
serial                  = /etc/ssl/ca.db.serial
RANDFILE                = /etc/ssl/ca.db.rand
certificate             = /etc/ssl/certs/ca.crt
private_key             = /etc/ssl/private/ca.key
default_days            = 365
default_crl_days        = 30
default_md              = md5
preserve                = no
policy                  = policy_anything
```

```
[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
EOT

#  sign the certificate
echo "CA signing: $CSR -> $CERT:"
openssl ca -config ca.config -out $CERT -infiles $CSR
echo "CA verifying: $CERT <-> CA cert"
openssl verify -CAfile /etc/ssl/certs/ca.crt $CERT

#  cleanup after SSLeay
rm -f ca.config
rm -f ca.db.serial.old
rm -f ca.db.index.old

#  die gracefully
exit 0
```

Now, make this program executable and change its default permission:
[root@deep]# **chmod 755 /usr/bin/sign.sh**

NOTE: You can also find this program "sign.sh" in the mod_ssl distribution under "mod_ssl-version/pkg.contrib/" subdirectory or on our floppy.tgz archive file. Also note that the section **[CA_own]** must be changed to refect your own environment and don't forget to change the " **openssl verify -CAfile /etc/ssl/certs/ca.crt $CERT**" line to.

Securing Openssl

Make your keys "Read and Write" only by the super-user "root". This is important since no one need to touch this files.

- To make your keys "read and Write" only by "root", use the command:
  [root@deep]# **chmod 600 /etc/ssl/certs/ca.crt**
  [root@deep]# **chmod 600 /etc/ssl/certs/server.crt**
  [root@deep]# **chmod 600 /chroot/httpd/etc/ssl/private/ca.key**
  [root@deep]# **chmod 600 /chroot/httpd/etc/ssl/private/server.key**

Commands

The commands listed bellows are some that we use often in our regular use but much more exist and you must check the man page and documentation for more details and information. As an example we will show you how to create a certificates for your Apache Web Server.

NOTE: All commands listed bellow are assumed to be made in **"/etc/ssl/"** directory.

1.1 Create a RSA private key protected with a passphrase for your Apache Server.

```
[root@deep]# openssl genrsa -des3 -out server.key 1024
Generating RSA private key, 1024 bit long modulus
.....................+++++
.....+++++
e is 65537 (0x10001)
Enter PEM pass phrase:
```

```
Verifying password - Enter PEM pass phrase:
```

Please backup this **server.key** file and remember the pass-phrase you had to enter at a secure location.

1.2 Generate a Certificate Signing Request (CSR) with the server RSA private key.

```
[root@deep]# openssl req -new -key server.key -out server.csr
Enter PEM pass phrase:
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]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [Open Network Architecture]:
Organizational Unit Name (eg, section) [Internet Department]:
Common Name (eg, YOUR name) [www.openarch.com]:
Email Address [admin@openarch.com]:

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

You now have to send this Certificate Signing Request (CSR) to a Certifying Authority (CA) for signing. The result is then a real Certificate, which can be used for Apache. Here you have to options: First you can let the CSR sign by a commercial CA like Verisign or Thawte. Then you usually have to post the CSR into a web form, pay for the signing and await the signed Certificate you then can store into a server.crt file. Second you can use your own CA and now have to sign the CSR yourself by this CA. See bellow on how to sign a CSR with your CA yourself.

Make sure you enter the FQDN “Fully Qualified Domain Name” of the server when OpenSSL prompts you for the “CommonName”, i.e. when you generate a CSR for a website which will be later accessed via https://www.mydomain.com/, enter www.mydomain.com here.

1.3 Create a RSA private key for your ca (CA).

```
[root@deep]# openssl genrsa -des3 -out ca.key 1024
Generating RSA private key, 1024 bit long modulus
.....+++++
.....+++++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
```

Please backup this **ca.key** file and remember the pass-phrase you had to enter at a secure location.

1.4 Create a self-signed (CA) certificate (x509 structure) with the RSA key of the CA.

```
[root@deep]# openssl req -new -x509 -days 365 -key ca.key -out ca.crt
```

```
Enter PEM pass phrase:
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]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [Open Network Architecture]:
Organizational Unit Name (eg, section) [Internet Department]:CA Marketing
Common Name (eg, YOUR name) [www.openarch.com]:
Email Address [admin@openarch.com]:
```

```
[root@deep]# mv server.key private/
[root@deep]# mv ca.key private/
[root@deep]# mv ca.crt certs/
```

NOTE: The “req” command creates a self-signed certificate when the -x509 switch is used.

1.5 Signing a certificate request. (We create and use our own Certificate Authority (CA))

Prepare a script for signing which is needed because the “openssl ca" command has some strange requirements and the default OpenSSL config doesn't allow one easily to use “openssl ca" directly. So a script named **sign.sh** is distributed with the floppy disk under openssl directory. Use this script for signing.

Now you can use this CA to sign server CSR's in order to create real SSL Certificates for use inside an Apache Webserver (assuming you already have a server.csr at hand):

```
[root@deep]# /usr/bin/sign.sh server.csr
Using configuration from ca.config
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName               :PRINTABLE:'CA'
stateOrProvinceName       :PRINTABLE:'Quebec'
localityName              :PRINTABLE:'Montreal'
organizationName          :PRINTABLE:'Open Network Architecture'
organizationalUnitName    :PRINTABLE:'Internet Department'
commonName                :PRINTABLE:'www.openarch.com'
emailAddress              :IA5STRING:'admin@openarch.com'
Certificate is to be certified until Dec  1 14:59:29 2000 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: server.crt <-> CA cert
server.crt: OK
```

This signs the CSR and results in a server.crt file.

```
[root@deep]# mv server.crt certs/
```

Now you have two files: **server.key** and **server.crt**. These now can be for example used as following inside your Apache's **httpd.conf** file:

SSLCertificateFile /etc/ssl/certs/server.crt
SSLCertificateKeyFile /etc/ssl/private/server.key

The **server.csr** file is no longer needed.
[root@deep]# **rm -f server.csr**

Installed files

> /etc/ssl
> /etc/ssl/crl
> /etc/ssl/certs
> /etc/ssl/private
> /etc/ssl/openssl.cnf
> /usr/bin/openssl
> /usr/bin/c_rehash
> /usr/bin/sign.sh
> /usr/bin/c_hash
> /usr/bin/c_info
> /usr/bin/c_issuer
> /usr/bin/c_name
> /usr/bin/der_chop
> /usr/include/openssl
> /usr/include/openssl/e_os.h
> /usr/include/openssl/e_os2.h
> /usr/include/openssl/crypto.h
> /usr/include/openssl/tmdiff.h
> /usr/include/openssl/opensslv.h
> /usr/include/openssl/opensslconf.h
> /usr/include/openssl/ebcdic.h
> /usr/include/openssl/md2.h
> /usr/include/openssl/md5.h
> /usr/include/openssl/sha.h
> /usr/include/openssl/mdc2.h
> /usr/include/openssl/hmac.h
> /usr/include/openssl/ripemd.h
> /usr/include/openssl/des.h
> /usr/include/openssl/rc2.h
> /usr/include/openssl/rc4.h
> /usr/include/openssl/rc5.h
> /usr/include/openssl/idea.h
> /usr/include/openssl/blowfish.h
> /usr/include/openssl/cast.h
> /usr/include/openssl/bn.h
> /usr/include/openssl/rsa.h
> /usr/include/openssl/dsa.h
> /usr/include/openssl/dh.h
> /usr/include/openssl/buffer.h
> /usr/include/openssl/bio.h
> /usr/include/openssl/stack.h
> /usr/include/openssl/safestack.h
> /usr/include/openssl/lhash.h
> /usr/include/openssl/rand.h
> /usr/include/openssl/err.h
> /usr/include/openssl/objects.h
> /usr/include/openssl/evp.h
> /usr/include/openssl/asn1.h
> /usr/include/openssl/asn1_mac.h
> /usr/include/openssl/pem.h
> /usr/include/openssl/pem2.h
> /usr/include/openssl/x509.h
> /usr/include/openssl/x509_vfy.h
> /usr/include/openssl/x509v3.h
> /usr/include/openssl/conf.h
> /usr/include/openssl/txt_db.h
> /usr/include/openssl/pkcs7.h
> /usr/include/openssl/pkcs12.h
> /usr/include/openssl/comp.h
> /usr/include/openssl/ssl.h
> /usr/include/openssl/ssl2.h
> /usr/include/openssl/ssl3.h
> /usr/include/openssl/ssl23.h
> /usr/include/openssl/tls1.h
> /usr/include/openssl/rsaref.h
> /usr/lib/libcrypto.a
> /usr/lib/libssl.a
> /usr/lib/libRSAglue.a
> /var/lock/subsys/named

Linux Imap & Pop Server

Overview

A mail server is a server that is running one or more of the following: an IMAP server, a POP3 server, a POP2 server, and an SMTP server. For now, I'm only going to cover installing IMAP4, POP3, and POP2, which all come in a single package. An example of SMTP server is Sendmail (widely used).

POP stands for “Post Office Protocol” and simply allows you to list messages, retrieve them, and delete them. There are many POP servers for Linux available, the stock one that ships with most distributions is ok for the majority of users. IMAP is POP on steroids. It allows you to easily maintain multiple accounts, have multiple people access one account, leave mail on the server, just download the headers, or bodies and no attachments, and so on.

IMAP is ideal for anyone on the go or with serious email needs. The default POP and IMAP servers that most distributions ship (bundled together into a single package named imapd oddly enough) fulfill most needs.

These installation instructions assume

Commands are Unix-compatible.
The source path is “/var/tmp” (other paths are possible).
Installations were tested on RedHat Linux 6.1.
All steps in the installation will happen in superuser account “root”.
Imap version number is 4.5

Packages

Imap Homepage: http://www.washington.edu/imap/
You must be sure to download: imap-4.5.tar.Z

Tarballs

It is a good idea to make a list of files on the system before you install Imap, and one afterwards, and then compare them using ‘**diff**’ to find out what file it placed where. Simply run ‘**find /* > imap1**’ before and ‘**find /* > imap2**’ after you install the software, and use ‘**diff imap1 imap2 > imap**’ to get a list of what changed.

Compilation

Decompress the tarball (tar.Z).

```
[root@deep]# cp imap-version.tar.Z /var/tmp
[root@deep]# cd /var/tmp
[root@deep]# tar xzpf imap-version.tar.Z
```

Compile and Optimize

Cd into the new Imap directory and type the following commands on your terminal:

Edit the **Makefile** file (vi +698 src/osdep/unix/Makefile) and change:

```
sh -c '(test -f /usr/include/sys/statvfs.h -a $(OS) != sc5 -a $(OS) != sco) && $(LN) flocksun.c flockbsd.c ||
$(LN) flocksv4.c flockbsd.c'
```

To read:

```
sh -c '(test -f /usr/include/sys/statvfs.h -a $(OS) != sc5 -a $(OS) != sco -a $(OS) != lnx) && $(LN)
flocksun.c flockbsd.c || $(LN) flocksv4.c flockbsd.c'
```

This modification will change the “sys/stavfs” file. This file with the new glibc 2.1 of Linux is different from what is available on the Sun.

Edit the **Makefile** file (vi +355 src/osdep/unix/Makefile) and change:

BASECFLAGS="-g -fno-omit-frame-pointer -O6 -DNFSKLUDGE" \
To read:
**BASECFLAGS="-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions -DNFSKLUDGE" **

This is our optimization flag for the compilation of IMAP/POP software on the server.

Edit the **Makefile** file (vi +112 src/osdep/unix/Makefile) and change:

BUILDOPTIONS= EXTRACFLAGS="$(EXTRACFLAGS)"\
To read:
BUILDOPTIONS= EXTRACFLAGS= -DDISABLE_POP_PROXY=1 -DIGNORE_LOCK_EACCES_ERRORS=1"$(EXTRACFLAGS)"

By default, the ipop[23]d servers offer POP->IMAP proxy access, which allow a POP client to access mail on an IMAP server by using the POP server as a go-between. Setting the "-DDISABLE_POP_PROXY=1" option disables this facility.

The "-DIGNORE_LOCK_EACCES_ERRORS=1" option disable the "Mailbox vulnerable -- directory must have 1777 protection" warning which occurs if an attempt to create a mailbox lock file fails due to an EACCES error.

Edit the **Makefile** file (vi +58 src/osdep/unix/Makefile) and change:

ACTIVEFILE=/usr/lib/news/active
To read:
ACTIVEFILE=/var/lib/news/active

SPOOLDIR=/usr/spool
To read:
SPOOLDIR=/var/spool

RSHPATH=/usr/ucb/rsh
To read:
RSHPATH=/usr/bin/rsh

The "ACTIVEFILE=" line specify the path of the "active" directory for IMAP/POP, the "SPOOLDIR=" is where we put the "spool" directory of Linux IMAP/POP, and the "RSHPATH=" specify the path of "rsh" directory on our system. It is important to note that we don't use rsh services on our server but even so we specify the right directory of "rsh".

Edit the **Makefile** file (vi +85 src/osdep/unix/Makefile) and change:

CC=cc
To read:
CC=egcs

This line represent the name of our GCC compiler we will use to compile IMAP/POP software, in our case (egcs).

```
[root@deep]# make lnp

[root@deep]# mv ./c-client/c-client.a ./c-client/libimap.a
[root@deep]# install -m 644 c-client/libimap.a /usr/lib/
[root@deep]# install -m 644 ./src/ipopd/ipopd.8c /usr/man/man8/ipopd.8c
```

```
[root@deep]# install -m 644 ./src/imapd/imapd.8c /usr/man/man8/imapd.8c
[root@deep]# install -s -m 755 ./ipopd/ipop2d /usr/sbin/
[root@deep]# install -s -m 755 ./ipopd/ipop3d /usr/sbin/
[root@deep]# install -s -m 755 ./imapd/imapd /usr/sbin/
[root@deep]# mkdir -p /usr/include/imap
[root@deep]# install -m 644 ./c-client/*.h /usr/include/imap/
[root@deep]# install -m 644 ./src/osdep/tops-20/shortsym.h /usr/include/imap/
[root@deep]# chown root.mail /usr/sbin/ipop2d
[root@deep]# chown root.mail /usr/sbin/ipop3d
[root@deep]# chown root.mail /usr/sbin/imapd
```

The above commands would configure the software to ensure your system has the necessary functionality and libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Take a note that "**make lnp**" command above will configure your Linux system with Pluggable Authentication Modules (PAM) capabilities for better security.

The "**mkdir**" command would create a new directory named "imap" under "/usr/include". This new directory "imap" will keep all header files related to imapd program "c-client/*", and "shortsym.h" files.

The "**chown**" command would change the ownership of binaries program "ipop2d", "ipop3d", and "imapd" to be owner by the super-user "root", be group owner by the user "mail".

NOTE: For security reason, if you use only imapd service remove ipop2d and ipop3d binaries from your server. The same apply for ipopd, if you use only ipopd service remove imapd binary from your server. If you are intended to use imapd and ipopd services on your server then keep the both binaries.

Cleanup after work

```
[root@deep]# cd /var/tmp
[root@deep]# rm -rf imap-version/ imap-version.tar.Z
```

The "rm" command will remove all the source files we have used to compile and install IMAP/POP. It will also remove the IMAP/POP compressed archive from the "/var/tmp" directory.

Configurations

All software we describe in our book "Linuxsos.pdf" will have a specific directory and subdirectory in a tar compressed archive named "floppy.tgz" containing file configurations for the specific program. If you get this archive file, you wouldn't be obliged to reproduce the different configuration files bellow manually or cut and past them to create your configuration files. Whatever your decide to copy manually or get the files made to your convenience from the archive compressed files, it will be to your responsibility to modify, adjust for your needs and place the files related to IMAP/POP software to their appropriated places on your server machine, like show bellow. The server configuration files archive is located at the following Internet address: http://pages.infinit.net/lotus1/doc/opti/floppy.tgz

- To run IMAP/POP server, the following files are require and must be create or copied to their appropriated directories on your server.

  Copy the **imap** file to the "/etc/pam.d/" directory if you're intended to use **imapd** service. Copy the **pop** file to the "/etc/pam.d/' directory if you're intended to use **popd** service.

You can obtain configuration files listed bellow on our floppy.tgz archive. Copy the following files from the decompressed floppy.tgz archive to their appropriated places or copy and paste them directly from this book to the concerned file.

Configuration of the "/etc/pam.d/imap" file

Configure your "/etc/pam.d/imap" file to use pam authentication.

Create the **imap** file (touch /etc/pam.d/imap) and add:

```
#%PAM-1.0
auth            required    /lib/security/pam_pwdb.so shadow nullok
account         required    /lib/security/pam_pwdb.so
```

NOTE: This file is only requiring if you're intended to use IMAP service.

Configuration of the "/etc/pam.d/pop" file

Configure your "/etc/pam.d/pop" file to use pam authentication.

Create the **pop** file (touch /etc/pam.d/pop) and add:

```
#%PAM-1.0
auth            required    /lib/security/pam_pwdb.so shadow nullok
account         required    /lib/security/pam_pwdb.so
```

NOTE: This file is only requiring if you're intended to use POP service.

Securing IMAP/POP

Do you really need IMAP/POP service?

The IMAP/POP program is a very common exploit method for attackers as some versions contain a serious and easily exploited buffer overrun that allows remote execution commands as root. Make sure you have or update your daemon to version 4.5. Some POP servers also don't report failed logins, so an attacker can brute force passwords and you will never know. If yours does this you should upgrade.

Be aware that IMAP/POP programs use plaintext passwords by default. Anyone running a sniffer program along your network path can grab your username/password and use them to log in as you. It is not because you use an IMAP/POP mail reader on your LINUX system mean you need to run an IMAP/POP server locally. Check your configuration and if you use a remote/external IMAP/POP server shut off or uninstall the local daemon on your system.

Also if you are intended to use a web interface to read your mail via the Internet (WebMail) it is a good idea to use the SSL protocol to encrypt the communication with the IMAP/POP server. See the part V "Software's-Related Reference" in chapter 10 "Server Software" under the section "Linux Apache Web Server" for more information on the topic.

Further documentation

For more details, there are several man pages you can read:

```
$ man imapd (8C)        - Internet Message Access Protocol server
$ man ipopd (8C)        - Post Office Protocol server
```

Installed files

> /usr/include/imap
> /usr/include/imap/dummy.h
> /usr/include/imap/env.h
> /usr/include/imap/env_unix.h
> /usr/include/imap/fdstring.h
> /usr/include/imap/flstring.h
> /usr/include/imap/fs.h
> /usr/include/imap/ftl.h
> /usr/include/imap/imap4r1.h
> /usr/include/imap/linkage.h
> /usr/include/imap/lockfix.h
> /usr/include/imap/mail.h
> /usr/include/imap/mbox.h
> /usr/include/imap/mbx.h
> /usr/include/imap/mh.h
> /usr/include/imap/misc.h
> /usr/include/imap/mmdf.h
> /usr/include/imap/mtx.h
> /usr/include/imap/mx.h
> /usr/include/imap/netmsg.h
> /usr/include/imap/news.h
> /usr/include/imap/newsrc.h
> /usr/include/imap/nl.h
> /usr/include/imap/nntp.h
> /usr/include/imap/os_a32.h
> /usr/include/imap/os_a41.h
> /usr/include/imap/os_aix.h
> /usr/include/imap/os_aos.h
> /usr/include/imap/os_art.h
> /usr/include/imap/os_asv.h
> /usr/include/imap/os_aux.h
> /usr/include/imap/os_bsd.h
> /usr/include/imap/os_bsi.h
> /usr/include/imap/os_cvx.h
> /usr/include/imap/os_d-g.h
> /usr/include/imap/os_drs.h
> /usr/include/imap/os_dyn.h
> /usr/include/imap/os_hpp.h
> /usr/include/imap/os_isc.h
> /usr/include/imap/os_lnx.h
> /usr/include/imap/os_lyn.h
> /usr/include/imap/os_mct.h
> /usr/include/imap/os_mnt.h
> /usr/include/imap/os_nxt.h
> /usr/include/imap/os_os4.h
> /usr/include/imap/os_osf.h
> /usr/include/imap/os_ptx.h
> /usr/include/imap/os_pyr.h
> /usr/include/imap/os_qnx.h
> /usr/include/imap/os_s40.h
> /usr/include/imap/os_sc5.h
> /usr/include/imap/os_sco.h
> /usr/include/imap/os_sgi.h
> /usr/include/imap/os_shp.h
> /usr/include/imap/os_slx.h
> /usr/include/imap/os_sol.h
> /usr/include/imap/os_sos.h
> /usr/include/imap/os_sun.h
> /usr/include/imap/os_sv2.h
> /usr/include/imap/os_sv4.h
> /usr/include/imap/os_ult.h
> /usr/include/imap/os_vu2.h
> /usr/include/imap/osdep.h
> /usr/include/imap/phile.h
> /usr/include/imap/pop3.h
> /usr/include/imap/pseudo.h
> /usr/include/imap/rfc822.h
> /usr/include/imap/smtp.h
> /usr/include/imap/tcp.h
> /usr/include/imap/tcp_unix.h
> /usr/include/imap/tenex.h
> /usr/include/imap/unix.h
> /usr/include/imap/utf8.h
> /usr/include/imap/shortsym.h
> /usr/lib/libimap.a
> /usr/man/man8/ipopd.8c
> /usr/man/man8/imapd.8c
> /usr/sbin/ipop2d
> /usr/sbin/ipop3d
> /usr/sbin/imapd
> /etc/pam.d/imap
> /etc/pam.d/pop

Linux MM – Shared Memory Library

Overview

Build the MM Shared Memory library when you want shared memory support in Apache/EAPI. For instance this allows mod_ssl to use a high-performance RAM-based session cache instead of a disk-based one.

These installation instructions assume

Commands are Unix-compatible.
The source path is “/var/tmp” (other paths are possible).
Installations were tested on RedHat Linux 6.1.

All steps in the installation will happen in superuser account “root”.
Mm version number is 1.0.12

Packages

MM Homepage: http://www.engelschall.com/sw/mm/
You must be sure to download: mm-1.0.12.tar.gz

Tarballs

It is a good idea to make a list of files on the system before you install MM, and one afterwards, and then compare them using ‘**diff**’ to find out what file it placed where. Simply run ‘**find /* > mm1**’ before and ‘**find /* > mm2**’ after you install the software, and use ‘**diff mm1 mm2 > mm**’ to get a list of what changed.

Compilation

Decompress the tarball (tar.gz).

```
[root@deep]# cp mm_version.tar.gz /var/tmp
[root@deep]# cd /var/tmp
[root@deep]# tar xzpf mm_version.tar.gz
```

Compile

Cd into the new mm directory and type the following commands on your terminal:

```
./configure \
--disable-shared  \
--prefix=/usr
```

This tells MM to set itself up for this particular hardware setup with:

- Disable shared libraries.

```
[root@deep]# make
[root@deep]# make test
[root@deep]# make install
```

NOTE: The “**make test**” command would make some important tests on the program to verify that it work and respond properly before the installation.

Cleanup after work

```
[root@deep]# cd /var/tmp
[root@deep]# rm -rf mm-version/ mm_version.tar.gz
```

The “rm” command will remove all the source files we have used to compile and install mm. It will also remove the mm compressed archive from the “/var/tmp” directory.

Further documentation

For more details, there are several man pages you can read:

MM (3)	- Shared Memory Library
mm-config (1)	- MM library configuration/build utility

Installed files

/usr/bin/mm-config
/usr/include/mm.h
/usr/lib/libmm.la
/usr/lib/libmm.a
/usr/man/man1/mm-config.1
/usr/man/man3/mm.3

Linux Samba Server

Overview

Samba is the protocol by which a lot of PC-related machines share files and printers and other information such as lists of available files and printers. Operating systems that support this natively include Windows 95/98/NT, OS/2, and Linux and add on packages that achieve the same thing are available for DOS, Windows, VMS, Unix of all kinds, MVS, and more.

Apple Macs and some Web Browsers can speak this protocol as well. Alternatives to SMB include Netware, NFS, AppleTalk, Banyan Vines, Decnet etc; many of these have advantages but none are both public specifications and widely implemented in desktop machines by default.

These installation instructions assume

Commands are Unix-compatible.
The source path is "/var/tmp" (other paths are possible).
Installations were tested on RedHat Linux 6.1.
All steps in the installation will happen in superuser account "root".
Samba version number is 2.0.6

Packages

Samba Homepage: http://us1.samba.org/samba/samba.html
You must be sure to download: samba-2.0.6.tar.gz

Tarballs

It is a good idea to make a list of files on the system before you install Samba, and one afterwards, and then compare them using '**diff**' to find out what file it placed where. Simply run '**find /* > smb1**' before and '**find /* > smb2**' after you install the software, and use '**diff smb1 smb2 > smb**' to get a list of what changed.

Compilation

Decompress the tarball (tar.gz).

```
[root@deep]# cp samba.version.tar.gz /var/tmp
[root@deep]# cd /var/tmp
[root@deep]# tar xzpf samba.version.tar.gz
```

Configure

Cd into the new Samba directory and then cd into the “sources” subdirectory.

Edit the **smbsh.in** file (vi +3 smbwrapper/smbsh.in) and change:

SMBW_LIBDIR=${SMBW_LIBDIR-@builddir@/smbwrapper}
To read:
SMBW_LIBDIR=${SMBW_LIBDIR**-/usr/bin**}

This will locate the “lib” directory to be under “/usr/bin” directory.

Edit the **Makefile.in** file (vi +28 Makefile.in) and change:

SBINDIR = @bindir@
To read:
SBINDIR = @sbindir@

VARDIR = @localstadir@
To read:
VARDIR = /var/log/samba

This will specify that our ”sbin” directory for binaries files will be located in the ”/usr/sbin” directory and the ”/var” directory for Samba log files will be under ”/var/log/samba” subdirectory.

Edit the **convert_smbpasswd** file (vi +10 script/convert_smbpasswd) and change:

nawk 'BEGIN {FS=":"}
To:
gawk 'BEGIN {FS=":"}

This will specify to use the GNU version of the awk text processing utility instated of the Bell Labs research version of awk program for the smbpasswd file. This file “convert_smbpasswd” convert a samba 1.9.18 smbpasswd file format into a Samba 2.0 smbpasswd file format.

Edit the **include.h** file (vi +655 include/include.h) and remove:

Remove the lines:
```
#ifdef strcat
#undef strcat
#endif /* strcat */
#define strcat(dest,src) __ERROR__XX__NEVER_USE_STRCAT___;
```

Edit the **smbmount.c** file (vi +99 client/smbmount.c) and change:

```
static void close_our_files(int client_fd)
{
        int i;
        for (i = 0; i < 256; i++) {
                if (i == client_fd) continue;
                close(i);
        }
```
To read:
```
static void close_our_files(int client_fd)
{
        struct rlimit limits;
```

```
int i;

getrlimit(RLIMIT_NOFILE,&limits);
for (i = 0; i < limits.rlim_max; i++) {
        if (i == client_fd) continue;
        close(i);
}
```

The two steps above for the “include.h” and “smbmount.c” files will make them compatible for the Red Hat glibc 2.1 library.

Compile and optimize

Type the following commands on your terminal:

```
CC="egcs" \
./configure \
--prefix=/usr \
--libdir=/etc \
--with-lockdir=/var/lock/samba \
--with-privatedir=/etc \
--with-swatdir=/usr/share/swat \
--with-pam \
--with-mmap
```

NOTE: The option “--with-mmap” can give a large boost to performance on some machines, on others it makes not difference at all, and on some it may reduce performance.

This tells Samba to set itself up for this particular hardware setup with:

- Include PAM password database support.
- Include experimental MMAP support (improve performance).

```
[root@deep]# make all
[root@deep]# make install

[root@deep]# install -m 755 script/mksmbpasswd.sh  /usr/bin/
[root@deep]# rm -rf /usr/share/swat/ (if like me, you don't like to configure samba in HTML)
[root@deep]# rm -f /usr/sbin/swat
[root@deep]# rm -f /usr/man/man8/swat.8
[root@deep]# mkdir -p /var/lock/samba
[root@deep]# mkdir -p /var/spool/samba (only require for printer sharing)
[root@deep]# chmod 1777 /var/spool/samba/ (only require for printer sharing)
```

The “**install**” command would install the script “mksmbpasswd.sh” under “/usr/bin/” directory. This script is needed to setup Samba user allowed to connect on our server via the “smbpasswd” file. See later on this documentation how to setup and use Samba password.

The “**rm**” command would remove the “/usr/share/swat” directory and all the files under it, it also remove the “swat” binary program under “/usr/sbin/”. The SWAT program is a web-based configuration utility that permit you to configure the “smb.conf” file via html. The utility listens on TCP port 901 of your server. Of course, in order to use the SWAT utility you will need to have a web server running, such as Apache. The SWAT utility can open a security breach on your server and for this reason I recommend to not use it and remove it.

The “**mkdir**” command would create a “/var/spool/samba/” directory on your system for all printer sharing jobs you may have. Of course this directory is only require is you’re intended to use Samba printer sharing over your LAN. Since we are not configured our Samba server to use

printer sharing, we do not need to create this directory (“/var/spool/samba/”) on our server and we do not need to use the command “chmod” to change the “sticky” bit in “/var/spool/samba” so only the file's owner can delete a given file in this directory.

NOTE: These installations assume that you are running Shadow passwords/pam. (You really should be!). If you are follow our Linux installation, this must already be set.

Cleanup after work
[root@deep]# cd /var/tmp
[root@deep]# rm -rf samba-version/ samba.version.tar.gz

The “rm” command will remove all the source files we have used to compile and install Samba. It will also remove the Samba compressed archive from the “/var/tmp” directory.

Configurations

Configuration files for different services are very specific depending of your need and your network architecture. Someone can install Samba Server and have just one client connection and other can install it with 1000 connections.

All software we describe in our book "Linuxsos.pdf" will have a specific directory and subdirectory in a tar compressed archive named “floppy.tgz” containing file configurations for the specific program. If you get this archive file, you wouldn’t be obliged to reproduce the different configuration files bellow manually or cut and past them to create your configuration files. Whatever your decide to copy manually or get the files made to your convenience from the archive compressed files, it will be to your responsibility to modify, adjust for your needs and place the files related to Samba software to their appropriated places on your server machine, like show bellow. The server configuration files archive is located at the following Internet address: http://pages.infinit.net/lotus1/doc/opti/floppy.tgz

- To run a Samba server, the following files are require and must be create or copied to their appropriated directories on your server.

  Copy the **smb.conf** and **lmhosts** files in the “/etc/” directory.
  Copy the **smb** script file in the “/etc/rc.d/init.d/” directory.
  Copy the **samba** file in the “/etc/logrotate.d/” directory.
  Copy the **samba** file in the “/etc/pam.d/” directory.

You can obtain configuration files listed bellow on our floppy.tgz archive. Copy the following files from the decompressed floppy.tgz archive to their appropriated places or copy and paste them directly from this book to the concerned file.

Configuration of the “/etc/smb.conf” file

The “/etc/smb.conf” file is the main configuration file for Samba server, you can specify which directory you want to access from windows machine, which IP addresses are authorized and so on. The first few lines of the file under the [global] line, contain global configuration directives, which are common to all shares (unless they are over-ridden on a per-share basis), followed by share sections.

In our example we are created just one directory “[tmp]” and are allowed one machine IP address to connect on the Samba server. Also, we don’t use printer sharing capability over Samba and Windows on our server.

Edit the **smb.conf** file (vi /etc/smb.conf) and add:

```
[global]

        workgroup = OPENARCH
        server string = Samba %h
        encrypt passwords = True
        security = user
        smb passwd file = /etc/smbpasswd
        log file = /var/log/samba/log.%m
        socket options = IPTOS_LOWDELAY TCP_NODELAY
        domain master = Yes
        local master = Yes
        preferred master = Yes
        os level = 65
        dns proxy = No
        name resolve order = lmhosts host bcast
        bind interfaces only = True
        interfaces = eth0 192.168.1.1
        hosts deny = ALL
        hosts allow = 192.168.1.4 127.0.0.1
        debug level = 1
        create mask = 0640
        directory mask = 0750
        level2 oplocks = True
        wide links = no
        read raw = no

[homes]
        comment = Home Directories
        browseable = no
        read only = no
        invalid users = root bin daemon nobody named sys tty disk mem kmem users

[tmp]
        comment = Temporary File Space
        path = /tmp
        read only = No
        valid users = admin
        invalid users = root bin daemon nobody named sys tty disk mem kmem users
```

This tells smb.conf file to set itself up for this particular configuration setup with:

[global]

workgroup = OPENARCH
The workgroup your server will appear to be in when queried by clients.

server string = Samba %h
The string that you wish to show to your users, a “%h” will be replaced with the hostname of the server you connect to.

encrypt passwords = True
This set encrypted password will be negotiated with the client instead of plain text password. Sniffer program will not be able to detect your password when it is encrypted. This option always must be set to True for security reason.

security = user

With user-level security, a client must first "log-on" with a valid username and password or the connection will be refused. This mean, a valid username and password for the client must exit in your “/etc/passwd” file on the Samba server or the connection from the client will fail.

smb passwd file = /etc/smbpasswd
This option “smb passwd file” sets the path to the encrypted smbpasswd file. The smbpasswd file is a copy of the “/etc/passwd” file containing valid username and password of client allowed to connect to the Samba server. The Samba software read this file when a connection is requested.

log file = /var/log/samba/log.%m
This option “log file” with the exentention “%m” allow you to have separate log files for each user or machine that log on your Samba server.

socket options = IPTOS_LOWDELAY TCP_NODELAY
This option “socket options” tune the connection for a local network and improve the performance of the Samba server for transferring files.

domain master = Yes
This option “domain master” identifies (nmbd) the Samba server daemon as a domain master browser for its given workgroup. This option, usually must be set to “Yes” only on one Samba server for its given workgroup for all other samba server on the same network and workgroup.

local master = Yes
This option “local master” allows (nmbd) the Samba server daemon to try and become a local master browser on a subnet. Like the above, usually this option must be set to “Yes” only on one Samba server that act as a local master on a subnet for all the other Samba server on your network.

preferred master = Yes
This option “preferred master” controls if (nmbd) the Samba server daemon is a preferred master browser for its workgroup. Once again must usually be set to “Yes” on one server for all the other on your network.

os level = 65
This option “os level” determines whether (nmbd) the Samba server daemon has a chance of becoming a local master browser for the WORKGROUP in the local broadcast area. The number 65 will win against any NT Server. If you have a NT Server on your network and want to set your Linux Samba server to be and win NT server for becoming a local master browser for the workgroup in the local broadcast area then you must set the “os level” option to 65. Also this option must be set on one Linux samba server and must be disable on all other Linux samba server you may have on your network.

dns proxy = No
This option “dns proxy” if set to “Yes” specifies that (nmbd) the Samba server daemon when acting as a WINS server and finding that a Net BIOS name has not been registered, should treat the Net BIOS name word-for-word as a DNS name and do a lookup with the DNS server for that name on behalf of the name-querying client. Since we are not configured the Samba server for acting as a WINS server, we don’t need to set this option to Yes.

name resolve order = lmhosts host bcast
This option “name resolve order” determine what naming services and in what order to resolve host names to IP addresses.

bind interfaces only = True

This option “bind interfaces only” if set to True, allows to limit what interfaces on a machine will serve smb requests. This is a security feature. The configuration “interfaces = eth0 192.168.1.1” bellow complete this option.

interfaces = eth0 192.168.1.1
This option “interfaces” allows you to override the default network interfaces list that Samba will use for browsing, name registration and other NBT traffic. By default Samba will query the kernel for the list of all active interfaces and use any interfaces except 127.0.0.1 that are broadcast capable. With this option, Samba will only listen on interface “eth0” on the IP address 192.168.1.1. This is a security feature and complte the above configuration (bind interfaces only = True).

hosts deny = ALL
Hosts listed here “hosts deny” are NOT permitted access to services unless the specific services have their own lists to override this one. We deny access to all hosts by default and allow specific hosts in the (hosts allow =) option bellow.

hosts allow = 192.168.1.4 127.0.0.1
This option “hosts allow” is a comma, space, or tab delimited set of hosts which are permitted to access a service. We allow host 192.168.1.4 and our localhost 127.0.0.1 to access the samba server. Note that localhost must always be set or you will receive some error messages.

debug level = 1
This option “debug level” allows the debug level (logging level) to be specified in the “smb.conf” file. If you set the debug level higher than 2 then you may suffer a large drop in performance. This is because the server flushes the log file after each operation, which can be very expensive.

create mask = 0640
This option “create mask” set the necessary permissions according to the mapping from DOS modes to UNIX permissions. With this option set to 0640, all files copying or creating from the Windows systems to the Unix system will have a permission of 0640 by default.

directory mask = 0750
This option “directory mask” set the octal modes which are used when converting DOS modes to UNIX modes when creating UNIX directories. With this option set to 0750, all directories copying or creating from the Windows systems to the Unix system will have a permission of 0750 by default.

level2 oplocks = True
This option “level2 oplocks” will increases the performance for many accesses of files that are not commonly written (such as application .EXE files).

wide links = no
This option “wide links” controls whether or not links in the UNIX file system may be followed by the server. Links that point to areas within the directory tree exported by the server are always allowed; this parameter controls access only to areas that are outside the directory tree being exported. It’s recommended to disable it for better security.

read raw = no
This option “read raw” controls whether or not the server will support the raw read SMB requests when transferring data to clients. Note that memory mapping is not used by the "read raw" operation. Thus you may find memory mapping is more effective if you disable "read raw" using "read raw = no" like we do.

[tmp]

comment = Temporary File Space
This option “comment” is a text field that is seen next to a share when a client does a queries the server, either via the network neighborhood or via "net view" to list what shares are available.

path = /tmp
This option “path” specifies a directory to which the user of the service is to be given access.

read only = No
This option “read only” specifies if users should be allowed to read only files or not.

valid users = admin
This option “valid users” is a list of users that should be allowed to login to this service.

invalid users = root bin daemon nobody named sys tty disk mem kmem users
This option “invalid users” is a list of users that should not be allowed to login to this service. This is really a ″paranoid″ check to absolutely ensure an improper setting does not breach your security.

Configuration of the “/etc/lmhosts” file

Configure your “/etc/lmhosts” file. The “lmhosts” file is the Samba Net BIOS name to IP address mapping file. It is very similar to the “/etc/hosts” file format, except that the hostname component must correspond to the Net BIOS naming format.

Create the **lmhosts** file (touch /etc/lmhosts) and add:

```
# Sample Samba lmhosts file.
#
127.0.0.1       localhost
192.168.1.1     deep
192.168.1.4     win
```

In our example, this file contains three IP to Net BIOS name mappings. The localhost (127.0.0.1), client named deep (192.168.1.1) and client named win (192.168.1.4).

Configuration of the “/etc/rc.d/init.d/smb” script file

Configure your “/etc/rc.d/init.d/smb” script file to start and stop Samba smbd and nmbd daemons Server.

Create the **smb** script file (touch /etc/rc.d/init.d/smb) and add:

```
#!/bin/sh
#
# chkconfig: - 91 35
# description: Starts and stops the Samba smbd and nmbd daemons \
#              used to provide SMB network services.

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

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

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

# Check that smb.conf exists.
```

```
[ -f /etc/smb.conf ] || exit 0

RETVAL=0

# See how we were called.
case "$1" in
  start)
        echo -n "Starting SMB services: "
        daemon smbd -D
        RETVAL=$?
        echo
        echo -n "Starting NMB services: "
        daemon nmbd -D
        RETVAL2=$?
        echo
        [ $RETVAL -eq 0 -a $RETVAL2 -eq 0 ] && touch /var/lock/subsys/smb || \
          RETVAL=1
        ;;
  stop)
        echo -n "Shutting down SMB services: "
        killproc smbd
        RETVAL=$?
        echo
        echo -n "Shutting down NMB services: "
        killproc nmbd
        RETVAL2=$?
        [ $RETVAL -eq 0 -a $RETVAL2 -eq 0 ] && rm -f /var/lock/subsys/smb
        echo ""
        ;;
  restart)
        $0 stop
        $0 start
        RETVAL=$?
        ;;
  reload)
      echo -n "Reloading smb.conf file: "
        killproc -HUP smbd
        RETVAL=$?
        echo
        ;;
  status)
        status smbd
        status nmbd
        RETVAL=$?
        ;;
  *)
        echo "Usage: $0 {start|stop|restart|status}"
        exit 1
esac

exit $RETVAL
```

Now, make this script executable and change its default permission:
[root@deep]# **chmod 700 /etc/rc.d/init.d/smb**

Create the symbolic rc.d links for Samba with the command:
[root@deep]# **chkconfig --add smb**

Samba script will not start automatically the smbd and nmbd daemon when you reboot the server. You can change it default by executing the following command:
[root@deep]# **chkconfig --level 345 smb on**

Start your Samba Server manually with the following command:
[root@deep]# **/etc/rc.d/init.d/smb start**

Configuration of the "/etc/pam.d/samba" file

Configure your "/etc/pam.d/samba" file to use pam authentication.

Create the **samba** file (touch /etc/pam.d/samba) and add:

```
Auth            required    /lib/security/pam_pwdb.so nullok shadow
Account         required    /lib/security/pam_pwdb.so
```

Configuration of the "/etc/logrotate.d/samba" file

Configure your "/etc/logrotate.d/samba" file to rotate each week your log files automatically.

Create the **samba** file (touch /etc/logrotate.d/samba) and add:

```
/var/log/samba/log.nmb {
   notifempty
   missingok
   postrotate
      /usr/bin/killall -HUP nmbd
   endrotate
}

/var/log/samba/log.smb {
   notifempty
   missingok
   postrotate
      /usr/bin/killall -HUP smbd
   endrotate
}
```

Securing Samba

Create an encrypted password file

The "/etc/smbpasswd" file is the **Samba** encrypted password file. It contains the username, Unix user id and the SMB hashed passwords of the user, as well as account flag information and the time the password was last changed.

Important: To create a Samba account you must first have a valid Linux account for them. Generate the smbpasswd file from your "/etc/passwd" file using the following command:

[root@deep]# **cat /etc/passwd | mksmbpasswd.sh > /etc/smbpasswd**

Create the user account:

[root@deep]# **smbpasswd -a username** (remember that "username" must be a valid Linux account).
New SMB password:
Retype new SMB password:
Password changed for user username.

[root@deep]# **chmod 600 /etc/smbpasswd**
[root@deep]# **testparm** (this will verify the smb.conf file for error).

NOTE: See ENCRYPTION.txt in samba/doc/texts/ for more information.

Immunize important configuration files
The immutable bit can be used to prevent accidentally deleting or overwriting a file that must be protected. It also prevents someone from creating a symbolic link to this file. Once your "smb.conf" and "lmhosts" files have been configured, it's a good idea to immunize them with command like:

```
[root@deep]# chattr +i /etc/smb.conf
[root@deep]# chattr +i /etc/lmhosts
```

Further documentation

For more details, there are several man pages you can read:

```
$ man Samba (7)         - A Windows SMB/CIFS fileserver for UNIX
$ man smb.conf (5)      - The configuration file for the Samba suite
$ man smbclient (1)     - ftp-like client to access SMB/CIFS resources on servers
$ man smbd (8)          - server to provide SMB/CIFS services to clients
$ man smbmnt (8)        - mount smb file system
$ man smbmount (8)      - mount smb file system
$ man smbpasswd (5)     - The Samba encrypted password file
$ man smbpasswd (8)     - change a users SMB password
$ man smbrun (1)        - interface program between smbd and external programs
$ man smbsh (1)         - Allows access to Windows NT filesystem using UNIX commands
$ man smbstatus (1)     - report on current Samba connections
$ man smbtar (1)        - shell script for backing up SMB shares directly to UNIX tape drives
$ man smbumount (8)     - umount for normal users
$ man testparm (1)      - check an smb.conf configuration file for internal correctness
$ man testprns (1)      - check printer name for validity with smbd
```

Samba Administrative Tools

The commands listed bellow are some that we use often in our regular use but much more exist and you must check the man page and documentation for more details and information.

smbstatus
smbstatus is a very simple program to list the current Samba connections.

- To report on current Samba connections, use the command:
  [root@deep]# **smbstatus**

  ```
  Samba version 2.0.6
  Service     uid     gid     pid     machine
  ----------------------------------------------
  tmp         webmaster webmaster  3995   gate     (192.168.1.3) Sat Sep 25 19:40:54 1999

  No locked files

  Share mode memory usage (bytes):
    1048464(99%) free + 56(0%) used + 56(0%) overhead = 1048576(100%) total
  ```

Samba Users Tools

The commands listed bellow are some that we use often in our regular use but much more exist and you must check the man page and documentation for more details and information.

smbclient
smbclient is a client that can talk to an SMB/CIFS server. If offers an interface similar to that of the FTP program. Operations include things like getting files from the server to the local machine, putting files from the local machine to the server, retrieving directory information from the server and so on.

- To connect to a Windows machine, use the command:
  [root@deep]# **smbclient *//sbmserver/sharename -U username***
  [root@deep]# smbclient //gate/tmp -U webmaster
  Password:
  Domain=[OPENARCH] OS=[Windows NT 4.0] Server=[NT LAN Manager 4.0]
  Smb: \>

-Where *//sbmserver* is the name of the server you want to connect to. */sharename* is the directory on this server you want to connect to and, *-U* is your username on this machine.

Installed files

> /etc/rc.d/init.d/smb
> /etc/rc.d/rc0.d/K35smb
> /etc/rc.d/rc1.d/K35smb
> /etc/rc.d/rc2.d/K35smb
> /etc/rc.d/rc3.d/K35smb
> /etc/rc.d/rc4.d/K35smb
> /etc/rc.d/rc5.d/K35smb
> /etc/rc.d/rc6.d/K35smb
> /etc/codepages
> /etc/lmhosts
> /etc/pam.d/samba
> /etc/smb.conf
> /etc/MACHINE.SID
> /etc/logrotate.d/samba
> /etc/smbpasswd
> /usr/bin/smbclient
> /usr/bin/smbspool
> /usr/bin/testparm
> /usr/bin/testprns
> /usr/bin/smbstatus
> /usr/bin/rpcclient
> /usr/bin/smbpasswd
> /usr/bin/make_smbcodepage
> /usr/bin/nmblookup
> /usr/bin/make_printerdef
> /usr/bin/smbtar
> /usr/bin/addtosmbpass
> /usr/bin/convert_smbpasswd
> /usr/bin/mksmbpasswd.sh
> /usr/man/man1/nmblookup.1
> /usr/man/man1/make_smbcodepage.1
> /usr/man/man1/smbclient.1
> /usr/man/man1/smbrun.1
> /usr/man/man1/smbsh.1
> /usr/man/man1/smbstatus.1
> /usr/man/man1/smbtar.1
> /usr/man/man1/testparm.1
> /usr/man/man1/testprns.1
> /usr/man/man5/lmhosts.5
> /usr/man/man5/smb.conf.5
> /usr/man/man5/smbpasswd.5
> /usr/man/man7/samba.7
> /usr/man/man8/nmbd.8
> /usr/man/man8/smbd.8
> /usr/man/man8/smbmnt.8
> /usr/man/man8/smbmount.8
> /usr/man/man8/smbpasswd.8
> /usr/man/man8/smbspool.8
> /usr/man/man8/smbumount.8
> /usr/sbin/smbd
> /usr/sbin/nmbd
> /var/log/samba
> /var/lock/samba

Linux OpenLDAP Server

Overview
LDAP (Lightweight Directory Access Protocol) is an open-standard protocol for accessing information services. The protocol runs over Internet transport protocols, such as TCP, and can be used to access stand-alone directory servers or X.500 directories.

These installation instructions assume

Commands are Unix-compatible.
The source path is "/var/tmp" (other paths are possible).
Installations were tested on RedHat Linux 6.1.
All steps in the installation will happen in superuser account "root".
OpenLDAP version number is 1_2_8

Packages

OpenLDAP Homepage: http://www.openldap.org/
You must be sure to download: openldap-1_2_8.tgz

Tarballs

It is a good idea to make a list of files on the system before you install OpenLDAP, and one afterwards, and then compare them using '**diff**' to find out what file it placed where. Simply run '**find /* > ldap1**' before and '**find /* > ldap2**' after you install the software, and use '**diff ldap1 ldap2 > ldap**' to get a list of what changed.

Compilation

Decompress the tarball (tar.gz).

```
[root@deep]# cp openldap-version.tgz /var/tmp
[root@deep]# cd /var/tmp/
[root@deep]# tar xzpf openldap-version.tgz
```

Compile and Optimize

Cd into the new OpenLDAP directory and type the following commands on your terminal:

Edit the **string.h** file (vi +52 include/ac/string.h) and remove the lines:

```
#else
        /* some systems have strdup(), but fail to declare it */
        extern char *(strdup)();
```

The following lines above doesn't apply to our Linux system and must be removed.

```
CC="egcs" \
CFLAGS="-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-
frame-pointer -fno-exceptions" \
./configure \
--prefix=/usr \
--libexecdir=/usr/sbin \
--localstatedir=/var/run \
--sysconfdir=/etc \
--enable-shared \
--with-gnu-ld
```

This tells OpenLDAP to set itself up for this particular hardware setup with:

- Build shared libraries.
- Assume the C compiler uses GNU ld.

```
[root@deep]# make depend
[root@deep]# make
[root@deep]# cd tests/
[root@deep]# make
[root@deep]# cd ..
[root@deep]# make install
```

The "**make depend**" command would build and make the necessary dependency of different files, "**make**" compile all source files into executable binaries, and then "**make install**" install the binaries and any supporting files into the appropriate locations.

The "**make**" command under "/test" directory would do some important test to verify the functionality of your LDAP server before the installation. If some tests fails, you'll need to fixes the problems before continuing the installation.

```
[root@deep]# install -d -m 700 /var/ldap
[root@deep]# echo localhost > /etc/openldap/ldapserver
[root@deep]# strip /usr/lib/liblber.so.1.0.0
[root@deep]# strip /usr/lib/libldap.so.1.0.0
[root@deep]# strip /usr/lib/libldap.a
[root@deep]# strip /usr/lib/liblber.a
[root@deep]# strip /usr/sbin/in.xfingerd
[root@deep]# strip /usr/sbin/go500
[root@deep]# strip /usr/sbin/go500gw
[root@deep]# strip /usr/sbin/mail500
[root@deep]# strip /usr/sbin/rp500
[root@deep]# strip /usr/sbin/fax500
[root@deep]# strip /usr/sbin/rcpt500
[root@deep]# strip /usr/sbin/slapd
[root@deep]# strip /usr/sbin/ldif2ldbm
[root@deep]# strip /usr/sbin/ldif2index
[root@deep]# strip /usr/sbin/ldif2id2entry
[root@deep]# strip /usr/sbin/ldif2id2children
[root@deep]# strip /usr/sbin/ldbmcat
[root@deep]# strip /usr/sbin/ldif
[root@deep]# strip /usr/sbin/centipede
[root@deep]# strip /usr/sbin/ldbmtest
[root@deep]# strip /usr/sbin/slurpd
[root@deep]# strip /usr/bin/ud
[root@deep]# strip /usr/bin/ldapadd
[root@deep]# strip /usr/bin/ldapsearch
[root@deep]# strip /usr/bin/ldapmodify
[root@deep]# strip /usr/bin/ldapmodrdn
[root@deep]# strip /usr/bin/ldappasswd
[root@deep]# strip /usr/bin/ldapdelete
```

The "**install**" command above will create a new directory named "ldap" under "/var" directory and will set its mode to be readable, writable, and executable only by the super-user "root" (700) for security reasons.

The "**strip**" command would discard all symbols from the object files. This means that our binaries files will be smaller in size. This will improve a bit the performance hit to the program since they will be fewer lines to read by the system when it'll execute the binary.

Cleanup after work

```
[root@deep]# cd /var/tmp
[root@deep]# rm -rf ldap openldap-version.tgz
```

The “rm” command will remove all the source files we have used to compile and install OpenLDAP. It will also remove the OpenLDAP compressed archive from the “/var/tmp” directory.

Configurations

All software we describe in our book "Linuxsos.pdf" will have a specific directory and subdirectory in a tar compressed archive named “floppy.tgz” containing file configurations for the specific program. If you get this archive file, you wouldn’t be obliged to reproduce the different configuration files bellow manually or cut and past them to create your configuration files. Whatever your decide to copy manually or get the files made to your convenience from the archive compressed files, it will be to your responsibility to modify, adjust for your needs and place the files related to OpenLDAP software to their appropriated places on your server machine, like show bellow. The server configuration files archive is located at the following Internet address: http://pages.infinit.net/lotus1/doc/opti/floppy.tgz

- To run OpenLDAP server, the following files are require and must be create or copied to their appropriated directories on your server.

  Copy the **slapd.conf** file in the “/etc/openldap/” directory.
  Copy the **ldap** script file in the “/etc/rc.d/init.d/” directory.

You can obtain configuration files listed bellow on our floppy.tgz archive. Copy the following files from the decompressed floppy.tgz archive to their appropriated places or copy and paste them directly from this book to the concerned file.

Configuration of the “/etc/ldap/slapd.conf” file

The “/etc/openldap/slapd.conf” file is the main config file for configuring ldap server, permission, password, database type, database location and so on.

Edit the **slap.conf** file (vi /etc/openldap/slapd.conf) and add:

```
#
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include         /etc/openldap/slapd.at.conf
include         /etc/openldap/slapd.oc.conf
schemacheck     off
#referral        ldap://ldap.itd.umich.edu

pidfile         /var/run/slapd.pid
argsfile        /var/run/slapd.args

########################################################################
# ldbm database definitions
########################################################################

database        ldbm
suffix          "o=openarch, c=com"
directory       /var/ldap
rootdn          "cn=admin, o=openarch, c=com"
rootpw          secret
# cleartext passwords, especially for the rootdn, should
# be avoid.  See slapd.conf(5) for details.

# ldbm indexed attribute definitions
index cn,sn,uid
```

```
index objectclass pres,eq
index default none
# ldbm access control definitions
defaultaccess read
access to attr=userpassword
        by self write
        by dn="cn=admin, o=openarch, c=com" write
        by * compare
```

You should be sure to set the following options in your “slapd.conf” file above before starting the slapd daemon program:

suffix *<dn>*

This option says what entries are to be held by this database. You should set this to the DN of the root of the subtree you are trying to create.

For example:
```
suffix      "o=openarch, c=com"
```

You should be sure to specify a directory where the index files should be created

directory *<directory>*

For example:
```
directory   /var/ldap
```

You need to make it so you can connect to slapd as somebody with permission to add entries. This is done through the following two options in the database definition:

rootdn *<dn>*

rootpw *<passwd>* /* Remember to use crypto password here !!! */

For example:
```
rootdn      "cn=admin, o=openarch, c=com"
rootpw      secret
```

These options specify a DN and password that can be used to authenticate as the "superuser" entry of the database (e.g. the entry allowed to do anything). The DN and password specified here will always work, regardless of whether the entry named actually exists or has the password given.

Finally, you should make sure that the database definition contains the index definitions you want:

index {<attrlist> | default} [pres,eq,approx,sub,none]

For example, to index the cn, sn, uid and objectclass attributes the following index configuration lines could be used.

For example:
```
index cn,sn,uid
index objectclass pres,eq
index default none
```

Configuration of the “/etc/rc.d/init.d/ldap” script file

Configure your “/etc/rc.d/init.d/ldap” script file to start and stop Ldap Server.

Create the **ldap** script file (touch /etc/rc.d/init.d/ldap) and add:

```
#!/bin/sh
#
# ldap    This shell script takes care of starting and stopping
#         ldap servers (slapd and slurpd).
#
# chkconfig: - 70 40
# description: LDAP stands for Lightweight Directory Access Protocol, used \
#         for implementing the industry standard directory services.
# processname: slapd
# config: /etc/openldap/slapd.conf
# pidfile: /var/run/slapd.pid

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

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

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

[ -f /usr/sbin/slapd ] || exit 0
[ -f /usr/sbin/slurpd ] || exit 0

RETVAL=0

# See how we were called.
case "$1" in
  start)
    # Start daemons.
    echo -n "Starting ldap: "
    daemon slapd
        RETVAL=$?
        if [ $RETVAL -eq 0 ]; then
      if grep -q "^replogfile" /etc/openldap/slapd.conf; then
                daemon slurpd
                RETVAL=$?
                [ $RETVAL -eq 0 ] && pidof slurpd | cut -f 1 -d " " > /var/run/slurpd
      fi
        fi
    echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/ldap
    ;;
  stop)
    # Stop daemons.
        echo -n "Shutting down ldap: "
        killproc slapd
        RETVAL=$?
        if [ $RETVAL -eq 0 ]; then
          if grep -q "^replogfile" /etc/openldap/slapd.conf; then
                killproc slurpd
                RETVAL=$?
          fi
        fi
        echo
    if [ $RETVAL -eq 0 ]; then
```

```
            rm -f /var/lock/subsys/ldap
            rm -f /var/run/slapd.args
        fi
    ;;
  status)
        status slapd
        RETVAL=$?
        if [ $RETVAL -eq 0 ]; then
            if grep -q "^replogfile" /etc/openldap/slapd.conf; then
                status slurpd
                RETVAL=$?
            fi
        fi
        ;;
  restart)
        $0 stop
        $0 start
        RETVAL=$?
        ;;
  reload)
        killproc -HUP slapd
        RETVAL=$?
        if [ $RETVAL -eq 0 ]; then
            if grep -q "^replogfile" /etc/openldap/slapd.conf; then
                killproc -HUP slurpd
                RETVAL=$?
            fi
        fi
        ;;
  *)
        echo "Usage: $0 start|stop|restart|status}"
        exit 1
esac

exit $RETVAL
```

Now, make this script executable and change its default permission:
[root@deep]# **chmod 700 /etc/rc.d/init.d/ldap**

Create the symbolic rc.d links for OpenLDAP with the command:
[root@deep]# **chkconfig --add ldap**

OpenLDAP script will not start automatically the slapd daemon when you reboot the server. You can change it default by executing the following command:
[root@deep]# **chkconfig --level 345 ldap on**

Start your OpenLDAP Server manually with the following command:
[root@deep]# **/etc/rc.d/init.d/ldap start**

Securing OpenLDAP

Immunize important configuration files
The immutable bit can be used to prevent accidentally deleting or overwriting a file that must be protected. It also prevents someone from creating a symbolic link to this file. Once your "slapd.conf" file have been configured, it's a good idea to immunize it with command like:

[root@deep]# **chattr +i /etc/openldap/slapd.conf**

Further documentation

For more details, there are several man pages you can read:

$ man ldapd (8)	- LDAP X.500 Protocol Daemon
$ man ldapdelete (1)	- ldap delete entry tool
$ man ldapfilter.conf (5)	- configuration file for LDAP get filter routines
$ man ldapfriendly (5)	- data file for LDAP friendly routines
$ man ldapmodify, ldapadd (1)	- ldap modify entry and ldap add entry tools
$ man ldapmodrdn (1)	- ldap modify entry RDN tool
$ man ldappasswd (1)	- change the password of an LDAP entry
$ man ldapsearch (1)	- ldap search tool
$ man ldapsearchprefs.conf (5)	- configuration file for LDAP search preference routines
$ man ldaptemplates.conf (5)	- configuration file for LDAP display template routines
$ man ldif (5)	- LDAP Data Interchange Format
$ man slapd (8)	- Stand-alone LDAP Daemon
$ man slapd.conf (5)	- configuration file for slapd, the stand-alone LDAP daemon
$ man slurpd (8)	- Standalone LDAP Update Replication Daemon
$ man ud (1)	- interactive LDAP Directory Server query program

OpenLDAP Creation and Maintenance Tools

Creating a database off-line

This method is best if you have many thousands of entries to create, which would take an unacceptably long time using the ldapadd method. This tool read the slapd configuration file and an input file containing a text representation of the entries to add.

An input file containing a text representation of entries named “my-data-file” is available bellow for our example.

- To create a database off-line, use the command:
  [root@deep]# **ldif2ldbm -i <inputfile> -f <slapdconfigfile>**
  [root@deep]# ldif2ldbm -i my-data-file -f /etc/openldap/slapd.conf

Where <inputfile> specifies the LDIF input file containing the entries to add in text form. <slapdconfigfile> specifies the slapd configuration file that tells where to create the indexes, what indexes to create, etc.

NOTE: Our slapd daemon is not started in this mode of creation.

The LDIF input file (input file containing a text representation of entries)

When you install OpenLDAP for the first time and have a big archive to put in your OpenLDAP software, it’s always a god idea to put all your database in a input file containing a text representation of entries and add this text file in your OpenLDAP with the command show above.

Create the **my-data-file** file (touch /tmp/my-data-file) and add as an example in this file the following lines:

```
dn: o=openarch, c=com
o: openarch
objectclass: organization

dn: cn=Ronald Smith, o=openarch, c=com
cn: Ronald Smith
sn: Smith
telephonenumber: (480) 757-5856
title: Operator.
objectclass: top
```

objectclass: person

dn: cn=Anthony Bay, o=openarch, c=com
cn: Anthony Bay
sn: Bay
homephone: (410) 896-3786
mobile: (410) 833-0590
mail: abay@openarch.com
objectclass: top
objectclass: person

dn: cn=George Parker, o=openarch, c=com
cn: George Parker
sn: Parker
telephonenumber: (414) 389-5695
fax: (414) 778-8785
mobile: (414) 470-8669
description: E-Commerce.
objectclass: top
objectclass: person

This example text file is for show you how to convert you archives information databases to a LDAP entries before add it to your new OpenLDAP database. More option type exist and can by created to feet your needs, consult your OpenLDAP documentation or book for more information.

Creating a database over LDAP
With this method you use the ldapadd tool to add entries, just like you would once the database is created. For example, to add a “Europe Mourani” entry using the ldapadd tool, you could create a file called “/tmp/newentry” with the contents:

Create the **newentry** file (touch /tmp/newentry) and add in this file the contents:

cn=Europe Mourani, o=openarch, c=com
cn=Europe Mourani
sn=Mourani
mail=emourani@old.com
description=Marketing relation.
objectClass=top
objectClass=person

- And then use a command like this to actually create the entry:
  [root@deep]# **ldapadd -f /tmp/newentry -D "cn=admin, o=openarch, c=com" -W**
  Enter LDAP Password :

The above command assumes that you have set rootdn to "cn=admin, o=openarch, c=com" and rootpw to "secret". You will be prompted to enter the password.

ldapmodify
The ldapmodify command opens a connection to an LDAP server, binds, and modifies or adds entries. The entry information is read from standard input or from file through the use of the -f option.

Input file format for the ldapmodify command.

Assuming that the file /tmp/entry exists and has the contents:
cn=Europe Mourani, o=openarch, c=com
- mail=emourani@old.com # will delete the old mail address for Europe Mourani in the database.

+mail=emourani@new.com # will add the new mail address for Europe Mourani in the database.

- To modify the contents of Ldap database, use the command:
  [root@deep]# **ldapmodify -D 'cn=Admin, o=openarch, c=com' -W -f *<inputfile>***

Will replace the contents of the "Europe Mourani" entry's mail attribute with the value emourani@new.com

OpenLDAP Users Tools

The commands listed bellow are some that we use often in our regular use but much more exist and you must check the man page and documentation for more details and information.

Search on LDAP for entries

ldapsearch opens a connection to an LDAP server, binds, and performs a search using the filter.

- To search on Ldap database for entries, use the command:
  [root@deep]# **ldapsearch -b *<dn>* *<attrs>***
  [root@deep]# **ldapsearch -b 'o=openarch.com' 'cn=a*'**

Will retrieve all entries and values beginning by the letter **a** and will printed to standard output.

Installed files

> /etc/openldap
> /etc/openldap/ldap.conf
> /etc/openldap/ldap.conf.default
> /etc/openldap/ldapfilter.conf
> /etc/openldap/ldapfilter.conf.default
> /etc/openldap/ldaptemplates.conf
> /etc/openldap/ldaptemplates.conf.default
> /etc/openldap/ldapsearchprefs.conf
> /etc/openldap/ldapsearchprefs.conf.default
> /etc/openldap/slapd.conf
> /etc/openldap/slapd.conf.default
> /etc/openldap/slapd.at.conf
> /etc/openldap/slapd.at.conf.default
> /etc/openldap/slapd.oc.conf
> /etc/openldap/slapd.oc.conf.default
> /etc/openldap/ldapserver
> /etc/rc.d/init.d/ldap
> /etc/rc.d/rc0.d/K40ldap
> /etc/rc.d/rc1.d/K40ldap
> /etc/rc.d/rc2.d/K40ldap
> /etc/rc.d/rc3.d/S70ldap
> /etc/rc.d/rc4.d/S70ldap
> /etc/rc.d/rc5.d/S70ldap
> /etc/rc.d/rc6.d/K40ldap
> /usr/bin/ud
> /usr/bin/ldapsearch
> /usr/bin/ldapmodify
> /usr/bin/ldapdelete
> /usr/bin/ldapmodrdn
> /usr/bin/ldappasswd
> /usr/bin/ldapadd
> /usr/include/ldap.h
> /usr/include/lber.h
> /usr/include/ldap_cdefs.h
> /usr/man/man3/ldap_open.3
> /usr/man/man3/ldap_errlist.3
> /usr/man/man3/ldap_err2string.3
> /usr/man/man3/ldap_first_attribute.3
> /usr/man/man3/ldap_next_attribute.3
> /usr/man/man3/ldap_first_entry.3
> /usr/man/man3/ldap_next_entry.3
> /usr/man/man3/ldap_count_entries.3
> /usr/man/man3/ldap_friendly.3
> /usr/man/man3/ldap_friendly_name.3
> /usr/man/man3/ldap_free_friendlymap.3
> /usr/man/man3/ldap_get_dn.3
> /usr/man/man3/ldap_explode_dn.3
> /usr/man/man3/ldap_explode_dns.3
> /usr/man/man3/ldap_dn2ufn.3
> /usr/man/man3/ldap_is_dns_dn.3
> /usr/man/man3/ldap_get_values.3
> /usr/man/man3/ldap_get_values_len.3
> /usr/man/man3/ldap_value_free.3
> /usr/man/man3/ldap_value_free_len.3
> /usr/man/man3/ldap_count_values.3
> /usr/man/man3/ldap_count_values_len.3
> /usr/man/man3/ldap_getfilter.3
> /usr/man/man3/ldap_init_getfilter.3
> /usr/man/man3/ldap_init_getfilter_buf.3
> /usr/man/man3/ldap_getfilter_free.3
> /usr/man/man3/ldap_getfirstfilter.3
> /usr/man/man3/ldap_getnextfilter.3
> /usr/man/man3/ldap_setfilteraffixes.3
> /usr/man/man3/ldap_build_filter.3
> /usr/man/man3/ldap_modify.3
> /usr/man/man3/ldap_modify_s.3
> /usr/man/man3/ldap_mods_free.3
> /usr/man/man3/ldap_modrdn.3

> /usr/include/disptmpl.h
> /usr/include/srchpref.h
> /usr/lib/liblber.so.1.0.0
> /usr/lib/liblber.so.1
> /usr/lib/liblber.so
> /usr/lib/liblber.la
> /usr/lib/liblber.a
> /usr/lib/libldap.so.1.0.0
> /usr/lib/libldap.so.1
> /usr/lib/libldap.so
> /usr/lib/libldap.la
> /usr/lib/libldap.a
> /usr/man/man1/ud.1
> /usr/man/man1/ldapdelete.1
> /usr/man/man1/ldapmodify.1
> /usr/man/man1/ldapadd.1
> /usr/man/man1/ldapmodrdn.1
> /usr/man/man1/ldappasswd.1
> /usr/man/man1/ldapsearch.1
> /usr/man/man3/cldap_close.3
> /usr/man/man3/cldap_open.3
> /usr/man/man3/cldap_search_s.3
> /usr/man/man3/cldap_setretryinfo.3
> /usr/man/man3/lber-decode.3
> /usr/man/man3/lber-encode.3
> /usr/man/man3/ldap.3
> /usr/man/man3/cldap.3
> /usr/man/man3/ldap_abandon.3
> /usr/man/man3/ldap_add.3
> /usr/man/man3/ldap_add_s.3
> /usr/man/man3/ldap_bind.3
> /usr/man/man3/ldap_bind_s.3
> /usr/man/man3/ldap_simple_bind.3
> /usr/man/man3/ldap_simple_bind_s.3
> /usr/man/man3/ldap_kerberos_bind_s.3
> /usr/man/man3/ldap_kerberos_bind1.3
> /usr/man/man3/ldap_kerberos_bind1_s.3
> /usr/man/man3/ldap_kerberos_bind2.3
> /usr/man/man3/ldap_kerberos_bind2_s.3
> /usr/man/man3/ldap_unbind.3
> /usr/man/man3/ldap_unbind_s.3
> /usr/man/man3/ldap_set_rebind_proc.3
> /usr/man/man3/ldap_cache.3
> /usr/man/man3/ldap_enable_cache.3
> /usr/man/man3/ldap_disable_cache.3
> /usr/man/man3/ldap_destroy_cache.3
> /usr/man/man3/ldap_flush_cache.3
> /usr/man/man3/ldap_uncache_entry.3
> /usr/man/man3/ldap_uncache_request.3
> /usr/man/man3/ldap_set_cache_options.3
> /usr/man/man3/ldap_charset.3
> /usr/man/man3/ldap_set_string_translators.3
> /usr/man/man3/ldap_enable_translation.3
> /usr/man/man3/ldap_translate_from_t61.3
> /usr/man/man3/ldap_translate_to_t61.3
> /usr/man/man3/ldap_t61_to_8859.3
> /usr/man/man3/ldap_8859_to_t61.3
> /usr/man/man3/ldap_compare.3
> /usr/man/man3/ldap_compare_s.3
> /usr/man/man3/ldap_delete.3
> /usr/man/man3/ldap_delete_s.3
> /usr/man/man3/ldap_disptmpl.3
> /usr/man/man3/ldap_modrdn_s.3
> /usr/man/man3/ldap_modrdn2.3
> /usr/man/man3/ldap_modrdn2_s.3
> /usr/man/man3/ldap_init.3
> /usr/man/man3/ldap_result.3
> /usr/man/man3/ldap_msgfree.3
> /usr/man/man3/ldap_search.3
> /usr/man/man3/ldap_search_s.3
> /usr/man/man3/ldap_search_st.3
> /usr/man/man3/ldap_searchprefs.3
> /usr/man/man3/ldap_init_searchprefs.3
> /usr/man/man3/ldap_init_searchprefs_buf.3
> /usr/man/man3/ldap_free_searchprefs.3
> /usr/man/man3/ldap_first_searchobj.3
> /usr/man/man3/ldap_next_searchobj.3
> /usr/man/man3/ldap_sort.3
> /usr/man/man3/ldap_sort_entries.3
> /usr/man/man3/ldap_sort_values.3
> /usr/man/man3/ldap_sort_strcasecmp.3
> /usr/man/man3/ldap_ufn.3
> /usr/man/man3/ldap_ufn_search_s.3
> /usr/man/man3/ldap_ufn_search_c.3
> /usr/man/man3/ldap_ufn_search_ct.3
> /usr/man/man3/ldap_ufn_setprefix.3
> /usr/man/man3/ldap_ufn_setfilter.3
> /usr/man/man3/ldap_ufn_timeout.3
> /usr/man/man3/ldap_url.3
> /usr/man/man3/ldap_is_ldap_url.3
> /usr/man/man3/ldap_url_parse.3
> /usr/man/man3/ldap_free_urldesc.3
> /usr/man/man3/ldap_url_search.3
> /usr/man/man3/ldap_url_search_s.3
> /usr/man/man3/ldap_url_search_st.3
> /usr/man/man5/ldap.conf.5
> /usr/man/man5/ldapfilter.conf.5
> /usr/man/man5/ldapfriendly.5
> /usr/man/man5/ldapsearchprefs.conf.5
> /usr/man/man5/ldaptemplates.conf.5
> /usr/man/man5/ldif.5
> /usr/man/man5/slapd.conf.5
> /usr/man/man5/slapd.replog.5
> /usr/man/man5/ud.conf.5
> /usr/man/man8/centipede.8
> /usr/man/man8/chlog2replog.8
> /usr/man/man8/edb2ldif.8
> /usr/man/man8/go500.8
> /usr/man/man8/go500gw.8
> /usr/man/man8/in.xfingerd.8
> /usr/man/man8/ldapd.8
> /usr/man/man8/ldbmcat.8
> /usr/man/man8/ldif.8
> /usr/man/man8/ldif2ldbm.8
> /usr/man/man8/ldif2index.8
> /usr/man/man8/ldif2id2entry.8
> /usr/man/man8/ldif2id2children.8
> /usr/man/man8/mail500.8
> /usr/man/man8/fax500.8
> /usr/man/man8/rcpt500.8
> /usr/man/man8/slapd.8
> /usr/man/man8/slurpd.8
> /usr/sbin/ldif
> /usr/sbin/in.xfingerd

> /usr/man/man3/ldap_init_templates.3
> /usr/man/man3/ldap_init_templates_buf.3
> /usr/man/man3/ldap_free_templates.3
> /usr/man/man3/ldap_first_disptmpl.3
> /usr/man/man3/ldap_next_disptmpl.3
> /usr/man/man3/ldap_oc2template.3
> /usr/man/man3/ldap_tmplattrs.3
> /usr/man/man3/ldap_first_tmplrow.3
> /usr/man/man3/ldap_next_tmplrow.3
> /usr/man/man3/ldap_first_tmplcol.3
> /usr/man/man3/ldap_next_tmplcol.3
> /usr/man/man3/ldap_entry2text.3
> /usr/man/man3/ldap_entry2text_search.3
> /usr/man/man3/ldap_vals2text.3
> /usr/man/man3/ldap_entry2html.3
> /usr/man/man3/ldap_entry2html_search.3
> /usr/man/man3/ldap_vals2html.3
> /usr/man/man3/ldap_error.3
> /usr/man/man3/ldap_perror.3
> /usr/man/man3/ld_errno.3
> /usr/man/man3/ldap_result2error.3
> /usr/sbin/go500
> /usr/sbin/go500gw
> /usr/sbin/mail500
> /usr/sbin/rp500
> /usr/sbin/fax500
> /usr/sbin/xrpcomp
> /usr/sbin/rcpt500
> /usr/sbin/slapd
> /usr/sbin/ldif2ldbm
> /usr/sbin/ldif2index
> /usr/sbin/ldif2id2entry
> /usr/sbin/ldif2id2children
> /usr/sbin/ldbmcat
> /usr/sbin/centipede
> /usr/sbin/ldbmtest
> /usr/sbin/slurpd
> /usr/share/openldap
> /usr/share/openldap/ldapfriendly
> /usr/share/openldap/go500gw.help
> /usr/share/openldap/rcpt500.help
> /var/ldap

Linux PostgreSQL Database Server

Overview

Postgres, developed originally in the UC Berkeley Computer Science Department, pioneered many of the object-relational concepts now becoming available in some commercial databases. It provides SQL92/SQL3 language support, transaction integrity, and type extensibility. PostgreSQL is a public domain, open source descendant of this original Berkeley code.

These installation instructions assume

Commands are Unix-compatible.
The source path is “/var/tmp” (other paths are possible).
Installations were tested on RedHat Linux 6.1.
All steps in the installation will happen in superuser account “root”.
PostgreSQL version number is 6_5_3
egcs-c++-1.1.2-24.i386.rpm package must be installed on your system.

Packages

PostgreSQL Homepage: http://www.postgresql.org/
You must be sure to download: postgresql-6_5_3_tar.gz

Tarballs

It is a good idea to make a list of files on the system before you install it, and one afterwards, and then compare them using ‘**diff**’ to find out what file it placed where. Simply run ‘**find /* > sql1**’ before and ‘**find /* > sql2**’ after you install the tarball, and use ‘**diff sql1 sql2 > sql**’ to get a list of what changed.

Compilation

Decompress the tarball (tar.gz).

```
[root@deep]# cp postgresql-version_tar.gz /var/tmp
[root@deep]# cd /var/tmp
[root@deep]# tar xzpf postgresql-version_tar.gz
```

Compile and Optimize

Step 1
First of all, create the Postgres Superuser Account (**postgres** is commonly used).

- To create the Postgres account, use the command:
  [root@deep]# **useradd -M -o -r -d /var/lib/pgsql -s /bin/bash -c "PostgreSQL Server" -u 40 postgres >/dev/null 2>&1 || :**

NOTE: The 2>&1 send the contents of stderr to stdout.

Step 2
Before compiling PosgreSQL program, you must verify that egcs-c++-1.1.2-24.i386.rpm package is installed on your system. The egcs-c++-1.1.2-24.i386.rpm package is located on you Red Hat 6.1 CD-ROM under “RedHat/RPMS” directory. After compilation and installation of PostgreSQL you can remove the egcs-c++-1.1.2-24.i386.rpm package from your system.

- To verify if egcs-c++-1.1.2-24.i386.rpm is installed, use the command:
  [root@deep]# **rpm -q egcs-c++**

- To install egcs-c++-1.1.2-24.i386.rpm, use the command:
  [root@deep]# **rpm -Uvh egcs-c++-1.1.2-24.i386.rpm**

Step 3
Cd into the new PosgreSQL directory and type the following commands on your terminal:

```
[root@deep]# cd src
CC="egcs" \
./configure \
--prefix=/usr \
--enable-locale
```

This tells PostgreSQL to set itself up for this particular hardware setup with:

- Enable locale support.

Edit the **Makefile.global** file (vi +210 Makefile.global) and change:

```
CFLAGS= -I$(SRCDIR)/include -I$(SRCDIR)/backend
To read:
CFLAGS= -I$(SRCDIR)/include -I$(SRCDIR)/backend   -O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions
```

This is our optimization flag for PostgreSQL Server. Of course you must change it to fit your system and CPU architecture.

```
[root@deep]# make all
[root@deep]# cd ..
[root@deep]# make -C src install
[root@deep]# make -C src/man install
[root@deep]# mkdir -p /usr/include/pgsql
[root@deep]# mv /usr/include/access  /usr/include/pgsql/
[root@deep]# mv /usr/include/commands  /usr/include/pgsql/
[root@deep]# mv /usr/include/executor  /usr/include/pgsql/
[root@deep]# mv /usr/include/lib  /usr/include/pgsql/
[root@deep]# mv /usr/include/libpq  /usr/include/pgsql/
[root@deep]# mv /usr/include/libpq++  /usr/include/pgsql/
[root@deep]# mv /usr/include/port  /usr/include/pgsql/
[root@deep]# mv /usr/include/utils  /usr/include/pgsql/
[root@deep]# mv /usr/include/fmgr.h /usr/include/pgsql/
[root@deep]# mv /usr/include/os.h /usr/include/pgsql/
[root@deep]# mv /usr/include/config.h /usr/include/pgsql/
[root@deep]# mv /usr/include/c.h /usr/include/pgsql/
[root@deep]# mv /usr/include/postgres.h /usr/include/pgsql/
[root@deep]# mv /usr/include/postgres_ext.h /usr/include/pgsql/
[root@deep]# mv /usr/include/libpq-fe.h /usr/include/pgsql/
[root@deep]# mv /usr/include/libpq-int.h /usr/include/pgsql/
[root@deep]# mv /usr/include/ecpgerrno.h /usr/include/pgsql/
[root@deep]# mv /usr/include/ecpglib.h /usr/include/pgsql/
[root@deep]# mv /usr/include/ecpgtype.h /usr/include/pgsql/
[root@deep]# mv /usr/include/sqlca.h /usr/include/pgsql/
[root@deep]# mv /usr/include/libpq++.H /usr/include/pgsql/
[root@deep]# mkdir -p /usr/lib/pgsql
[root@deep]# mv /usr/lib/*source /usr/lib/pgsql/
[root@deep]# mv /usr/lib/*sample /usr/lib/pgsql/
[root@deep]# mkdir -p /var/lib/pgsql
[root@deep]# chown -R postgres.postgres /var/lib/pgsql/
[root@deep]# chmod 755 /usr/lib/libpq.so.2.0
[root@deep]# chmod 755 /usr/lib/libecpg.so.3.0.0
[root@deep]# chmod 755 /usr/lib/libpq++.so.3.0
[root@deep]# strip /usr/bin/postgres
[root@deep]# strip /usr/bin/postmaster
[root@deep]# strip /usr/bin/ecpg
[root@deep]# strip /usr/bin/pg_id
[root@deep]# strip /usr/bin/pg_version
[root@deep]# strip /usr/bin/pg_dump
[root@deep]# strip /usr/bin/pg_passwd
[root@deep]# strip /usr/bin/psql
[root@deep]# rm -f /usr/lib/global1.description
[root@deep]# rm -f /usr/lib/local1_template1.description
```

The “**make**” command compile all source files into executable binaries, and then “**make install**” install the binaries and any supporting files into the appropriate locations. The “mkdir” will create a new directory named “pgsql” under the “/usr/include” and “/usr/lib” directories then we will move all subdirectories and files related to PostgreSQL under “/usr/include” and “/usr/lib” directories to the “/usr/include/pgsql” and “/usr/lib/pgsql” directories respectively with the command “**mv**”.

The "**chown**" command would set the correct files owner and group permission for the “/var/lib/pgsql” directory. The “**strip**” command would discard all symbols from the object files. This means that our binary file will be smaller in size. This will improve a bit the performance hit to the program since they will be fewer lines to read by the system when it’ll execute the binary.

The “**rm**” command will remove the “global1.description” and ”local1_template1.description” files that are not needed by our PosgreSQL program.

Create the database installation from your Postgres superuser account

```
[root@deep]# su postgres
[postgres@deep]# initdb --pglib=/usr/lib/pgsql --pgdata=/var/lib/pgsql

We are initializing the database system with username postgres (uid=40).
This user will own all the files and must also own the server process.

Creating Postgres database system directory /var/lib/pgsql/base

Creating template database in /var/lib/pgsql/base/template1

Creating global classes in /var/lib/pgsql/base

Adding template1 database to pg_database...

Vacuuming template1
Creating public pg_user view
Creating view pg_rules
Creating view pg_views
Creating view pg_tables
Creating view pg_indexes
Loading pg_description

[postgres@deep]# chmod 640 /var/lib/pgsql/pg_pwd
[postgres@deep]# exit
```

NOTE: Do not create the database installation as “root”! This would be a major security hole.

Cleanup after work

```
[root@deep]# cd /var/tmp
[root@deep]# rm -rf postgresql-version/ postgresql-version_tar.gz
```

Remove the egcs-c++-1.1.2-24.i386.rpm package to make space.

```
[root@deep]# rpm -e egcs-c++
```

The “rm” command will remove all the source files we have used to compile and install PostgreSQL. It will also remove the PostgreSQL compressed archive from the “/var/tmp” directory.

The “rpm -e” command will remove the egcs-c++ package we have installed to compile the PosgreSQL Server. Note that egcs-c++ package is requiring only for compiling program like PostgreSQL and can be uninstalled after successfully compilation of PostgreSQL safety.

Configurations

All software we describe in our book "Linuxsos.pdf" will have a specific directory and subdirectory in a tar compressed archive named “floppy.tgz” containing file configurations for the specific program. If you get this archive file, you wouldn’t be obliged to reproduce the different configuration files bellow manually or cut and past them to create your configuration files. Whatever your decide to copy manually or get the files made to your convenience from the archive compressed files, it will be to your responsibility to modify, adjust for your needs and place the files related to PostgreSQL software to their appropriated places on your server machine, like show bellow. The server configuration files archive is located at the following Internet address: http://pages.infinit.net/lotus1/doc/opti/floppy.tgz

- To run PostgreSQL Database server, the following file is require and must be create or copied to the appropriated directory on your server.

  Copy the **postgresql** script file in the "/etc/rc.d/init.d/" directory.

You can obtain configuration files listed bellow on our floppy.tgz archive. Copy the following files from the decompressed floppy.tgz archive to their appropriated places or copy and paste them directly from this book to the concerned file.

Configuration of the "/etc/rc.d/init.d/postgresql" script file

Configure your "/etc/rc.d/init.d/postgresql" script file to start and stop PostgreSQL Server.

Create the **postgresql** script file (touch /etc/rc.d/init.d/postgresql) and add:

```
#! /bin/sh
# postgresql    This is the init script for starting up the PostgreSQL
#               server

# chkconfig: 345 85 15
# description: Starts and stops the PostgreSQL backend daemon that handles \
#              all database requests.
# processname: postmaster
# pidfile: /var/run/postmaster.pid
#

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

# Get config.
. /etc/sysconfig/network

# Check that networking is up.
# Pretty much need it for postmaster.
[ ${NETWORKING} = "no" ] && exit 0

[ -f /usr/bin/postmaster ] || exit 0

# This script is slightly unusual in that the name of the daemon (postmaster)
# is not the same as the name of the subsystem (postgresql)

# See how we were called.
case "$1" in
  start)
        echo -n "Checking postgresql installation: "
        # Check for the PGDATA structure
        if [ -f /var/lib/pgsql/PG_VERSION ] && [ -d /var/lib/pgsql/base/template1 ]
        then
        # Check version of existing PGDATA

                if [ `cat /var/lib/pgsql/PG_VERSION` != '6.5' ]
                then
                        echo "old version. Need to Upgrade."
                        echo "See /usr/doc/postgresql-6.5.2/README.rpm for more information."
                        exit 1
                else
                        echo "looks good!"
                fi

        # No existing PGDATA! Initdb it.
```

```
    else
        echo "no database files found."
        if [ ! -d /var/lib/pgsql ]
        then
            mkdir -p /var/lib/pgsql
            chown postgres.postgres /var/lib/pgsql
        fi
        su -l postgres -c '/usr/bin/initdb --pglib=/usr/lib/pgsql --pgdata=/var/lib/pgsql'
    fi

    # Check for postmaster already running...
    pid=`pidof postmaster`
    if [ $pid ]
    then
        echo "Postmaster already running."
    else
        #all systems go -- remove any stale lock files
        rm -f /tmp/.s.PGSQL.* > /dev/null
        echo -n "Starting postgresql service: "
        su -l postgres -c '/usr/bin/postmaster -i -S -D/var/lib/pgsql'
        sleep 1
        pid=`pidof postmaster`
        if [ $pid ]
        then
            echo -n "postmaster [$pid]"
            touch /var/lock/subsys/postgresql
            echo $pid > /var/run/postmaster.pid
            echo
        else
            echo "failed."
        fi
    fi
    ;;
 stop)
    echo -n "Stopping postgresql service: "
    killproc postmaster
    sleep 2
    rm -f /var/run/postmaster.pid
    rm -f /var/lock/subsys/postgresql
    echo
    ;;
 status)
    status postmaster
    ;;
 restart)
    $0 stop
    $0 start
    ;;
 *)
    echo "Usage: postgresql {start|stop|status|restart}"
    exit 1
esac

exit 0
```

Now, make this script executable and change its default permission:
[root@deep]# **chmod 700 /etc/rc.d/init.d/postgresql**

Create the symbolic rc.d links for PostgreSQL with the command:
[root@deep]# **chkconfig --add postgresql**

Start your new PostgreSQL manually with the following command:
[root@deep]# **/etc/rc.d/init.d/postgresql start**

Commands

The commands listed bellow are some that we use often in our regular use but much more exist and you must check the man page for more details and information.

Client connections can be restricted by IP address and/or user name via the "**pg_hba.conf"** file in PG_DATA.

- To define a new user in your database, run the **createuser** utility program:
  [root@deep]# su postgres
  [postgres@deep]$ createuser
  Enter name of user to add ---> admin
  Enter user's postgres ID or RETURN to use unix user ID: 500 ->
  Is user "admin" allowed to create databases (y/n) y
  Is user "admin" a superuser? (y/n) y
  createuser: admin was successfully added

- To remove a user in your database, run the **destroyuser** utility program:
  [root@deep]# su postgres
  [postgres@deep]$ destroyuser
  Enter name of user to delete ---> admin
  destroyuser: delete of user admin was successful.

- To create a new database, run the **createdb** utility program:
  [root@deep]# su postgres
  [postgres@deep]$ createdb dbname (the name of the database).

  or with the Postgres terminal monitor program (psql)
  [root@deep]# su admin
  [admin@deep]$ psql template1
  Welcome to the POSTGRESQL interactive sql monitor:
  Please read the file COPYRIGHT for copyright terms of POSTGRESQL
  [PostgreSQL 6.5.3 on i686-pc-linux-gnu, compiled by egcs]

  type \? for help on slash commands
  type \q to quit
  type \g or terminate with semicolon to execute query
  You are currently connected to the database: template1

  template1➔ create database foo;
  CREATEDB

Other useful Postgres terminal monitor program (psql) are:

- To connect to the new database, use the command:
  template1➔ \c foo
  connecting to new database: foo
  foo➔

- To create a table, use the command:
  foo➔ create table bar (i int4, c char(16));
  CREATE
  foo➔

- To inspect the new table, use the command:

```
foo➔ \d bar
Table    = bar
+----------------------------------+----------------------------------+-----------+
|          Field                   |          Type                    | Length    |
+----------------------------------+----------------------------------+-----------+
| l                                | int4                             |         4 |
| c                                | char()                           |        16 |
+----------------------------------+----------------------------------+-----------+
foo➔
```

- To drop a table, index, view, use the command:
  foo➔ drop table table_name;
  foo➔ drop index index_name;
  foo➔ drop view view_name;

- To insert into: (once a table is created, it can be filled using the command…)
  foo➔ insert into table_name (name_of_attr1, name_of_attr2, name_of_attr3)
  foo➔ values (value1, value2, value3);

Installed files

> /etc/rc.d/init.d/postgresql
> /etc/rc.d/rc0.d/K15postgresql
> /etc/rc.d/rc1.d/K15postgresql
> /etc/rc.d/rc2.d/K15postgresql
> /etc/rc.d/rc3.d/S85postgresql
> /etc/rc.d/rc4.d/S85postgresql
> /etc/rc.d/rc5.d/S85postgresql
> /etc/rc.d/rc6.d/K15postgresql
> /usr/bin/postgres
> /usr/bin/postmaster
> /usr/bin/ecpg
> /usr/bin/pg_id
> /usr/bin/pg_version
> /usr/bin/psql
> /usr/bin/pg_dump
> /usr/bin/pg_dumpall
> /usr/bin/pg_upgrade
> /usr/bin/pg_passwd
> /usr/bin/cleardbdir
> /usr/bin/createdb
> /usr/bin/createlang
> /usr/bin/createuser
> /usr/bin/destroydb
> /usr/bin/destroylang
> /usr/bin/destroyuser
> /usr/bin/initdb
> /usr/bin/vacuumdb
> /usr/bin/initlocation
> /usr/bin/ipcclean
> /usr/include/lib
> /usr/include/lib/dllist.h
> /usr/include/pgsql
> /usr/include/pgsql/access
> /usr/include/pgsql/access/attnum.h
> /usr/include/pgsql/commands
> /usr/include/pgsql/commands/trigger.h
> /usr/include/pgsql/executor
> /usr/include/pgsql/executor/spi.h
> /usr/include/pgsql/libpq
> /usr/include/pgsql/libpq/pqcomm.h
> /usr/include/pgsql/libpq/libpq-fs.h
> /usr/include/pgsql/libpq++
> /usr/include/pgsql/libpq++/pgconnection.h
> /usr/include/pgsql/libpq++/pgdatabase.h
> /usr/man/manl/begin.l
> /usr/man/manl/close.l
> /usr/man/manl/cluster.l
> /usr/man/manl/commit.l
> /usr/man/manl/copy.l
> /usr/man/manl/create_aggregate.l
> /usr/man/manl/create_database.l
> /usr/man/manl/create_function.l
> /usr/man/manl/create_index.l
> /usr/man/manl/create_language.l
> /usr/man/manl/create_operator.l
> /usr/man/manl/create_rule.l
> /usr/man/manl/create_sequence.l
> /usr/man/manl/create_table.l
> /usr/man/manl/create_trigger.l
> /usr/man/manl/create_type.l
> /usr/man/manl/create_user.l
> /usr/man/manl/create_version.l
> /usr/man/manl/create_view.l
> /usr/man/manl/declare.l
> /usr/man/manl/delete.l
> /usr/man/manl/drop.l
> /usr/man/manl/drop_aggregate.l
> /usr/man/manl/drop_database.l
> /usr/man/manl/drop_function.l
> /usr/man/manl/drop_index.l
> /usr/man/manl/drop_language.l
> /usr/man/manl/drop_operator.l
> /usr/man/manl/drop_rule.l
> /usr/man/manl/drop_sequence.l
> /usr/man/manl/drop_table.l
> /usr/man/manl/drop_trigger.l
> /usr/man/manl/drop_type.l
> /usr/man/manl/drop_user.l
> /usr/man/manl/drop_view.l
> /usr/man/manl/end.l
> /usr/man/manl/explain.l
> /usr/man/manl/fetch.l
> /usr/man/manl/grant.l
> /usr/man/manl/insert.l
> /usr/man/manl/listen.l
> /usr/man/manl/load.l
> /usr/man/manl/lock.l
> /usr/man/manl/move.l

> /usr/include/pgsql/libpq++/pgtransdb.h
> /usr/include/pgsql/libpq++/pgcursordb.h
> /usr/include/pgsql/libpq++/pglobject.h
> /usr/include/pgsql/port
> /usr/include/pgsql/port/linux
> /usr/include/pgsql/utils
> /usr/include/pgsql/utils/geo_decls.h
> /usr/include/pgsql/utils/elog.h
> /usr/include/pgsql/utils/palloc.h
> /usr/include/pgsql/utils/mcxt.h
> /usr/include/pgsql/fmgr.h
> /usr/include/pgsql/os.h
> /usr/include/pgsql/config.h
> /usr/include/pgsql/c.h
> /usr/include/pgsql/postgres.h
> /usr/include/pgsql/postgres_ext.h
> /usr/include/pgsql/libpq-fe.h
> /usr/include/pgsql/libpq-int.h
> /usr/include/pgsql/ecpgerrno.h
> /usr/include/pgsql/ecpglib.h
> /usr/include/pgsql/ecpgtype.h
> /usr/include/pgsql/sqlca.h
> /usr/include/pgsql/libpq++.H
> /usr/lib/libpq.a
> /usr/lib/libpq.so.2.0
> /usr/lib/libpq.so.2
> /usr/lib/libpq.so
> /usr/lib/libecpg.a
> /usr/lib/libecpg.so.3.0.0
> /usr/lib/libecpg.so.3
> /usr/lib/libecpg.so
> /usr/lib/libpq++.a
> /usr/lib/libpq++.so.3.0
> /usr/lib/libpq++.so.3
> /usr/lib/libpq++.so
> /usr/lib/plpgsql.so
> /usr/lib/pgsql
> /usr/lib/pgsql/global1.bki.source
> /usr/lib/pgsql/local1_template1.bki.source
> /usr/lib/pgsql/pg_geqo.sample
> /usr/lib/pgsql/pg_hba.conf.sample
> /usr/man/man1/cleardbdir.1
> /usr/man/man1/createdb.1
> /usr/man/man1/createuser.1
> /usr/man/man1/destroydb.1
> /usr/man/man1/destroyuser.1
> /usr/man/man1/ecpg.1
> /usr/man/man1/initdb.1
> /usr/man/man1/initlocation.1
> /usr/man/man1/ipcclean.1
> /usr/man/man1/pg_dump.1
> /usr/man/man1/pg_dumpall.1
> /usr/man/man1/pg_passwd.1
> /usr/man/man1/pg_upgrade.1
> /usr/man/man1/postgres.1
> /usr/man/man1/postmaster.1
> /usr/man/man1/psql.1
> /usr/man/man3/catalogs.3
> /usr/man/man3/libpq.3
> /usr/man/man5/pg_hba.conf.5
> /usr/man/manl
> /usr/man/manl/abort.l
> /usr/man/manl/alter_table.l
> /usr/man/manl/alter_user.l
> /usr/man/manl/notify.l
> /usr/man/manl/reset.l
> /usr/man/manl/revoke.l
> /usr/man/manl/rollback.l
> /usr/man/manl/select.l
> /usr/man/manl/set.l
> /usr/man/manl/show.l
> /usr/man/manl/sql.l
> /usr/man/manl/update.l
> /usr/man/manl/vacuum.l
> /var/lib/pgsql
> /var/lib/pgsql/base
> /var/lib/pgsql/base/template1
> /var/lib/pgsql/base/template1/pg_proc
> /var/lib/pgsql/base/template1/pg_type
> /var/lib/pgsql/base/template1/pg_attribute
> /var/lib/pgsql/base/template1/pg_class
> /var/lib/pgsql/base/template1/pg_inherits
> /var/lib/pgsql/base/template1/pg_index
> /var/lib/pgsql/base/template1/pg_statistic
> /var/lib/pgsql/base/template1/pg_operator
> /var/lib/pgsql/base/template1/pg_opclass
> /var/lib/pgsql/base/template1/pg_am
> /var/lib/pgsql/base/template1/pg_amop
> /var/lib/pgsql/base/template1/pg_amproc
> /var/lib/pgsql/base/template1/pg_language
> /var/lib/pgsql/base/template1/pg_aggregate
> /var/lib/pgsql/base/template1/pg_ipl
> /var/lib/pgsql/base/template1/pg_inheritproc
> /var/lib/pgsql/base/template1/pg_rewrite
> /var/lib/pgsql/base/template1/pg_listener
> /var/lib/pgsql/base/template1/pg_description
> /var/lib/pgsql/base/template1/pg_attribute_relid_attnam_index
> /var/lib/pgsql/base/template1/pg_attribute_relid_attnum_index
> /var/lib/pgsql/base/template1/pg_attribute_attrelid_index
> /var/lib/pgsql/base/template1/pg_proc_oid_index
> /var/lib/pgsql/base/template1/pg_proc_proname_narg_type_index
> /var/lib/pgsql/base/template1/pg_proc_prosrc_index
> /var/lib/pgsql/base/template1/pg_type_oid_index
> /var/lib/pgsql/base/template1/pg_type_typname_index
> /var/lib/pgsql/base/template1/pg_class_oid_index
> /var/lib/pgsql/base/template1/pg_class_relname_index
> /var/lib/pgsql/base/template1/pg_attrdef
> /var/lib/pgsql/base/template1/pg_attrdef_adrelid_index
> /var/lib/pgsql/base/template1/pg_relcheck
> /var/lib/pgsql/base/template1/pg_relcheck_rcrelid_index
> /var/lib/pgsql/base/template1/pg_trigger
> /var/lib/pgsql/base/template1/pg_trigger_tgrelid_index
> /var/lib/pgsql/base/template1/pg_description_objoid_index
> /var/lib/pgsql/base/template1/PG_VERSION
> /var/lib/pgsql/base/template1/pg_user
> /var/lib/pgsql/base/template1/pg_rules
> /var/lib/pgsql/base/template1/pg_views
> /var/lib/pgsql/base/template1/pg_tables
> /var/lib/pgsql/base/template1/pg_indexes
> /var/lib/pgsql/pg_variable
> /var/lib/pgsql/pg_database
> /var/lib/pgsql/pg_shadow
> /var/lib/pgsql/pg_group
> /var/lib/pgsql/pg_log
> /var/lib/pgsql/PG_VERSION
> /var/lib/pgsql/pg_hba.conf
> /var/lib/pgsql/pg_geqo.sample
> /var/lib/pgsql/pg_pwd

Linux Squid Proxy Server

Overview

A few proxy-server programs are on the market. These proxy-servers have two main drawbacks: they are commercial software and they don't support ICP. The excellent Apache web server has included a proxy-cache module since its 1.2. This module is a very interesting option: it's free, and works with the most popular web server on the Net. However, it doesn't use ICP, and its robustness is not comparable to the best choice for a proxy-cache server: SQUID.

Squid consists in these programs:

squid: the main proxy server
dnsserver: a DNS lookup program that performs single, blocking DNS operations
unlinkd: a program to delete files in the background from the cache directory.

It also provides a CGI program, designed to be run through a web interface, that outputs statistics about its configuration and performance and allows some management capabitlities.

Squid is a high-performance proxy-cache server and it is the result of efforts by numerous individuals from the Internet community. Development is led by Duane Wessels of the National Laboratory for Applied Network Research and funded by the National Science Foundation. Squid is derived from the "cached" software from the ARPA-funded Harvest research project. Squid offers high-performance caching of web clients, and also supports FTP, Gopher, and HTTP requests. It stores hot objects in RAM, and mantains a robust database of objects in disk directories. Squid also supports the SSL protocol for proxying secure connections and has a complex access control mechanism. In addition, Squid can be hierarchically linked to other Squid-based proxy servers for streamlined caching of pages.

In our compilation and configuration we'll configure Squid to run as an httpd-accelerator to get more performance. In accelerator mode, the Squid server acts as a reverse proxy cache: it accepts client requests, serves them out of cache if possible, or requests them from the origin server for which it is the reverse proxy. You move the server away from port 80 (or whatever your published port is), and substitute the accelerator, which then pulls the HTTP data from the "real" HTTP server (only the accelerator needs to know where the real server is). The outside world sees no difference (apart from an increase in speed, with luck).

These installation instructions assume

Commands are Unix-compatible.
The source path is "/var/tmp" (other paths are possible).
Installations were tested on RedHat Linux 6.1.
All steps in the installation will happen in superuser account "root".
Squid version number is 2_3_STABLE1

Packages

Squid Homepage: http://squid.nlanr.net/
You must be sure to download: squid-2_3_STABLE1-src_tar.gz

Tarballs

It is a good idea to make a list of files on the system before you install Squid, and one afterwards, and then compare them using '**diff**' to find out what file it placed where. Simply run '**find /* >**

squid1' before and '**find /* > squid2**' after you install the software, and use '**diff squid1 squid2 > squid**' to get a list of what changed.

Compilation

Decompress the tarball (tar.gz).

```
[root@deep]# cp squid-version_STABLEz-src_tar.gz /var/tmp
[root@deep]# cd /var/tmp
[root@deep]# tar xzpf squid-version_STABLEz-src_tar.gz
```

Configure and Optimize

Squid Proxy Server can't run as superuser root, for this reason we'll create a special user with less privilege for running Squid Proxy Server.

```
[root@deep]# /usr/sbin/useradd -d /cache/ -r -s /dev/null squid >/dev/null 2>&1
[root@deep]# mkdir /cache/
[root@deep]# chown -R squid.squid /cache/
```

First of all, we add the user "squid" to the "/etc/passwd" file, then we create the "/cache" directory if this directory doesn't exist and only if it doesn't exist. Finally we change the owner of our directory "cache" to be the user "squid".

NOTE: Usually we don't need to make the command (**mkdir /cache/**) because we are already create this directory when we are partitioning our disk. If this partition doesn't exist so execute this command to create the directory.

Cd into the new Squid directory and type the following commands on your terminal:

Edit the **Makefile.in** file (vi +18 icons/Makefile.in) and change the line:

```
DEFAULT_ICON_DIR = $(sysconfdir)/icons
To read:
DEFAULT_ICON_DIR = $(libexecdir)/icons
```

We change the variable (sysconfdir) to be (libexecdir). With this modification, the "icons" directory of Squid will be located under the "/usr/lib/squid" directory.

Edit the **Makefile.in** file (vi +34 src/Makefile.in) and change the lines:

```
DEFAULT_CACHE_LOG = $(localstatedir)/logs/cache.log
To read:
DEFAULT_CACHE_LOG = $(localstatedir)/log/squid/cache.log

DEFAULT_ACCESS_LOG = $(localstatedir)/logs/access.log
To read:
DEFAULT_ACCESS_LOG = $(localstatedir)/log/squid/access.log

DEFAULT_STORE_LOG = $(localstatedir)/logs/store.log
To read:
DEFAULT_STORE_LOG = $(localstatedir)/log/squid/store.log

DEFAULT_PID_FILE = $(localstatedir)/logs/squid.pid
To read:
DEFAULT_PID_FILE = $(localstatedir)/run/squid.pid
```

DEFAULT_SWAP_DIR = $(localstatedir)/cache
To read:
DEFAULT_SWAP_DIR = **/cache**

DEFAULT_ICON_DIR = $(sysconfdir)/icons
To read:
DEFAULT_ICON_DIR = **$(libexecdir)**/icons

We change the default location of “cache.log”, “access.log”, and “store.log” files to be located under “/var/log/squid” directory, then we put the pid file of Squid under “/var/run” directory and finally locate the “icons” directory of Squid under “/usr/lib/squid/icons” with the variable (libexecdir) above.

malloc

Many users have found improved performance when linking Squid with an external malloc library, such as GNU malloc. To make Squid use GNU malloc follows these simple steps:

1. Download the GNU malloc source, available from one of The GNU FTP Mirror sites (http://www.gnu.org/order/ftp.html).

2. Compile GNU malloc
   [root@deep]# tar xzpf malloc.tar.gz
   [root@deep]# cd malloc
   [root@deep]# vi Makefile and uncomment the line: CPPFLAGS = -DUSG # in the Makefile
   [root@deep]# export CC=egcs
   [root@deep]# make

3. Copy libmalloc.a to your system's library directory and be sure to name it *libgnumalloc.a*.
   [root@deep]# **cp libmalloc.a /usr/lib/libgnumalloc.a**

4. (Optional) Copy the GNU malloc.h to your system's include directory and be sure to name it *gnumalloc.h*. This step is not required, but if you do this, then Squid will be able to use the *mstat()* function to report memory usage statistics on the cachemgr info page.
   [root@deep]# **cp malloc.h /usr/include/gnumalloc.h**

Compile and Optimize

Return into the new Squid directory and type the following commands on your terminal:

[root@deep]# **export CACHE_HTTP_PORT=80**

```
CC="egcs" \
CFLAGS="-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-
frame-pointer -fno-exceptions" \
./configure \
--prefix=/usr \
--exec-prefix=/usr \
--bindir=/usr/sbin \
--libexecdir=/usr/lib/squid \
--localstatedir=/var \
--sysconfdir=/etc/squid \
--enable-cache-digests \
--enable-poll \
--disable-ident-lookups \
--enable-truncate \
--enable-underscores \
--enable-heap-replacement
```

This tells Squid to set itself up for this particular hardware setup with:

- Use Cache Digests to improve performance.

- Enable poll() instead of select().

- Disable-ident-lookups allows you to remove code that performs Ident (RFC 931) lookups.

- Enable-truncate uses truncate() instead of unlink() when removing cache files. Truncate gives a little performance improvement. Also take a note that Truncate uses more filesystem inodes than unlink.

- Enable-underscores. Squid by default rejects any host names with _ in their name to conform to Internet standards. If you disagree with this you may allow _ in hostnames by using this switch, provided that the resolver library on the host where Squid runs does not reject _ in hostnames.

- Enable-heap-replacement. This option allows you to use various cache replacement algorithms, instead of the standard LRU algorithm. See below for more explication.

```
[root@deep]# make -f makefile
[root@deep]# make install
[root@deep]# mkdir -p /var/log/squid
[root@deep]# rm -rf /var/logs/
[root@deep]# chown squid.squid /var/log/squid/
[root@deep]# chmod 750 /var/log/squid/
[root@deep]# chmod 750 /cache/
[root@deep]# rm -f /usr/sbin/RunCache
[root@deep]# rm -f /usr/sbin/RunAccel
[root@deep]# strip /usr/sbin/squid
[root@deep]# strip /usr/sbin/client
[root@deep]# strip /usr/lib/squid/dnsserver
[root@deep]# strip /usr/lib/squid/unlinkd
[root@deep]# strip /usr/lib/squid/cachemgr.cgi
```

The “make -f” command will compile all source files into executable binaries, and “make install” will install the binaries and any supporting files into the appropriate locations. The “mkdir” will create a new directory named “squid” under “/var/log”. The “rm -rf” command will remove the “/var/logs” directory since this directory has been created to handle the log files related to Squid that we have moved to “/var/log/squid” location. The “chown” will change the owner of “/var/log/squid” to be the user squid and “chmod” command will make the mode of “squid” and “cache” directories to be (0750/drwxr-x---) for security reason.

Take a note that we remove the small scripts named “RunCache” and “RunAccel” which take care to start Squid in caching mode or accelerator mode since we use a better script named “squid” located under “/etc/rc.d/init.d/” directory to take advantage of Linux system V. The **“**strip**”** command would reduce the size of binaries for optimum performance.

What is a Cache Digest?

A Cache Digest is a summary of the contents of an Internet Object Caching Server. It contains, in a compact (i.e. compressed) format, an indication of whether or not particular URLs are in the cache. A "lossy" technique is used for compression, which means that very high compression factors can be achieved at the expense of not having 100% correct information.

Cleanup after work

```
[root@deep]# cd /var/tmp
[root@deep]# rm -rf squid-version/ squid-version_STABLEz-src_tar.gz
```

[root@deep]# rm -rf malloc/ malloc.tar.gz (if your are used the malloc)

The “rm” command will remove all the source files we have used to compile and install Squid and Malloc. It will also remove the Squid and Malloc compressed archive from the “/var/tmp” directory.

Configurations

All software we describe in our book "Linuxsos.pdf" will have a specific directory and subdirectory in a tar compressed archive named “floppy.tgz” containing file configurations for the specific program. If you get this archive file, you wouldn’t be obliged to reproduce the different configuration files bellow manually or cut and past them to create your configuration files. Whatever your decide to copy manually or get the files made to your convenience from the archive compressed files, it will be to your responsibility to modify, adjust for your needs and place the files related to Squid software to their appropriated places on your server machine, like show bellow. The server configuration files archive is located at the following Internet address: http://pages.infinit.net/lotus1/doc/opti/floppy.tgz

- To run Squid server, the following files are require and must be create or copied to their appropriated directories on your server.

  Copy the **squid.conf** file in the “/etc/squid/” directory.
  Copy the **squid** script file in the “/etc/rc.d/init.d/” directory.
  Copy the **squid** file in the “/etc/logrotate.d/” directory.

You can obtain configuration files listed bellow on our floppy.tgz archive. Copy the following files from the decompressed floppy.tgz archive to their appropriated places or copy and paste them directly from this book to the concerned file.

Configuration of the “/etc/squid/squid.conf” file as a httpd-accelerator

Configure your “/etc/squid/squid.conf” file to be in httpd-accelerator mode. If the Web Server run on the same server where Squid is installed, you must put your httpd (Apache) daemon running on port 81. With Apache you can do it by assign the line (Port 81) in “httpd.conf”. If the Web Server (Apache) run on other server in your network like we do, you can keep the same port number (80) for Apache, since Squid Proxy will bind on different IP number where the port (80) is not already in use.

Edit the **squid.conf** file (vi /etc/squid/squid.conf) and add:

```
http_port 80
icp_port 0
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 16 MB
cache_dir ufs /cache 200 16 256
emulate_httpd_log on
redirect_rewrites_host_header off
replacement_policy GDSF
half_closed_clients off
acl all src 0.0.0.0/0.0.0.0
http_access allow all
cache_mgr admin
cache_effective_user squid
cache_effective_group squid
httpd_accel_host 208.164.186.3
httpd_accel_port 80
```

log_icp_queries off
buffered_logs on

This tells squid.conf file to set itself up for this particular configuration setup with:

http_port 80
This option “http_port” specify the port number where Squid will listen for HTTP client requests. With this configuration, client will have the illusion to be connected to the Apache Web Server (usualy port 80).

icp_port 0
This option “icp_port” specify the port number where Squid sends and receives ICP requests to and from neighbour caches. 0 disable this option since we are configured our proxy to be an accelerator for our Web Server and we don’t use neighbour caches for proxy server.

acl QUERY urlpath_regex cgi-bin \? and no_cache deny QUERY
These options “acl QUERY urlpath_regex cgi-bin \? and no_cache deny QUERY” are used to force certain objects to never be cached like files under cgi-bin. This is a security feature.

cache_mem 16 MB
This option “cache_mem” specifies the ideal amount of memory to be used for: In-Transit objects, Hot Objects, Negative-Cached objects. This is an optimization feature. Add here the amount of memory (RAM memory) to devote to caching. Warning: Squid uses much more than this value. Rule of thumb: if you have N megabytes free for Squid, put N/3 here.

cache_dir ufs /cache 200 16 256
This option “cache_dir” specify the kind of storage system to use. Most everyone will want to use (ufs) as the type, the location of your cache directory (/cache), the amount of disk space (MB) to use under this directory (200MB), the number of first-level subdirectories which will be created under the directory (16 first-level), and the number of second-level subdirectories which will be created under each first-level directory (256 second-level).

emulate_httpd_log on
This option “emulate_httpd_log” if set to on, the Cache can emulate the log file format which many “httpd” programs use. This is very useful if you want to use program like Webalizer that analyze Web Server log file.

redirect_rewrites_host_header off
By default Squid rewrites any Host: header in redirected requests. If you are running a accelerator then this may not be a wanted effect of a redirector. This option is used to bypass the redirectors if the load becomes too high. Only use this if the redirectors are used for “optimizations” and not access controls.

replacement_policy GDSF
The cache replacement policy parameter determines which objects are evicted (replaced) when disk space is needed. When compiling Squid with “--enable-heap-replacement” you can choose between two new, enhanced policies:

GDSF: Greedy-Dual Size Frequency
LFUDA: Least Frequently Used with Dynamic Aging

Both of these policies perform better than the original Squid LRU policy which is used by default if you are not specifying the “--enable-heap-replacement” option during the compile time.

The GDSF policy optimizes object-hit rate by keeping smaller popular objects in cache so it has a better chance of getting a hit. It achieves a lower byte hit rate than LFUDA though since it evicts larger (possibly popular) objects.

The LFUDA policy keeps popular objects in cache regardless of their size and thus optimizes byte hit rate at the expense of hit rate since one large, popular object will prevent many smaller, slightly less popular objects from being cached.

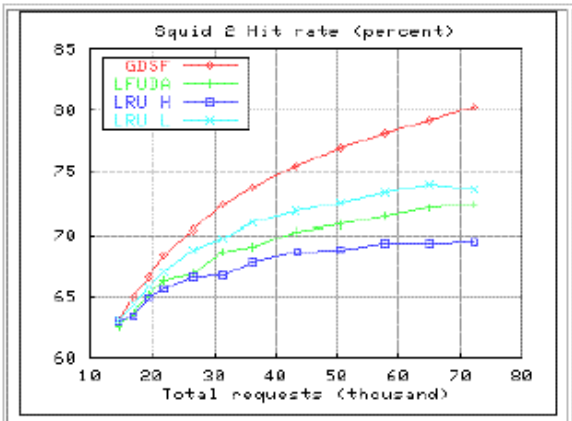


Figure 1: SPECWeb Hit Rate

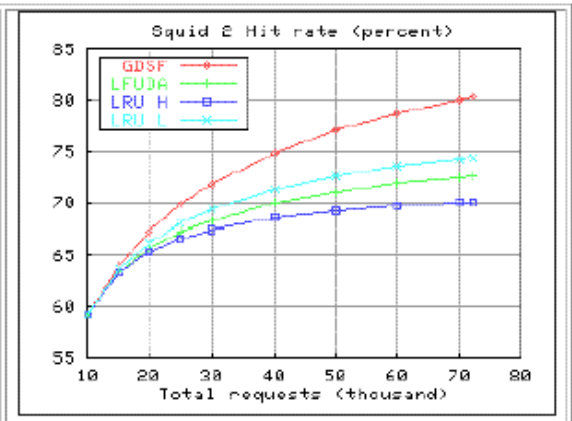


Figure 2: `http` Hit Rate

Figures 1 and 2 show the hit rate of the Squid 2 implementations under the SPECWeb workload and the `http` workload. In hit rate there is little difference between the original SPECWeb validation and this revalidation: the relative ordering has not changed and the shape of the curves is broadly similar. There is greater variance in the policies in the SPECWeb version and note that the run length of the `http` test is slightly greater.

half_closed_clients off
This option “half_closed_clients” if set to off, tell Squid to immediately close client connections when read(2) returns no more data to read because, some clients may shutdown the sending side of their TCP connections, while leaving their receiving sides open. Sometimes, Squid can not tell the difference between a half-closed and a fully-closed TCP connection.

acl all src 0.0.0.0/0.0.0.0 and http_access allow all
These options “acl and http_access” defining an Access List. See your documentation for more information. Our “acl” and “http_access” allow every one to connect on the proxy server since we use this proxy to accelerated our public Web Server.

cache_mgr admin
This option “cache_mgr” specify the email-address of local cache manager who will receive mail if the cache dies.

cache_effective_user squid and cache_effective_group squid
This option “cache_effective_user and cache_effective_group” specify the UID?GID the cache will run on. This is a security feature and you must never run Squid as a “root” user. In our configuration we use the UID “squid” and the GID “squid”.

httpd_accel_host 208.164.186.3 and httpd_accel_port 80

These options "httpd_accel_host and httpd_accel_port" define the host name and port number where the real HTTP Server is. In our configuration, the real HTTP Server is on the IP address 208.164.186.3 (www.openarch.com) and on the port (80). The www.openarch.com is another host on our network and since the Squid Server don't reside on the same host of Apache HTTP Web Server, we can use the port (80) for our Squid Proxy Server, the port (80) for our Apache Web Server and the illusion is perfect.

log_icp_queries off
This option "log_icp_queries" specify if you want ICP queries to be logged to "access.log" or not. Since we don't use ICP, we turn this option off.

buffered_logs on
This option "buffered_logs" if turned "on" can speed up the writing of some log files slightly. This is an optimization feature.

Configuration of the "/etc/rc.d/init.d/squid" script file as a httpd-accelerator

Configure your "/etc/rc.d/init.d/squid" script file to start and stop Squid Internet Object Cache. This script have been modified to setup swap cache for Squid in "/cache" instate of "/var/spool/squid".

Create the **squid** script file (touch /etc/rc.d/init.d/squid) and add:

```
#!/bin/bash
# squid             This shell script takes care of starting and stopping
#                   Squid Internet Object Cache
#
# chkconfig: - 90 25
# description: Squid - Internet Object Cache. Internet object caching is \
#          a way to store requested Internet objects (i.e., data available \
#          via the HTTP, FTP, and gopher protocols) on a system closer to the \
#          requesting site than to the source. Web browsers can then use the \
#          local Squid cache as a proxy HTTP server, reducing access time as \
#          well as bandwidth consumption.
# pidfile: /var/run/squid.pid
# config: /etc/squid/squid.conf

PATH=/usr/bin:/sbin:/bin:/usr/sbin
export PATH

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

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

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

# check if the squid conf file is present
[ -f /etc/squid/squid.conf ] || exit 0

# determine the name of the squid binary
[ -f /usr/sbin/squid ] && SQUID=squid
[ -z "$SQUID" ] && exit 0

# determine which one is the cache_swap directory
CACHE_SWAP=`sed -e 's/#.*//g' /etc/squid/squid.conf | \
        grep cache_dir | sed -e 's/cache_dir//' | \
        cut -d ' ' -f 2`
[ -z "$CACHE_SWAP" ] && CACHE_SWAP=/cache
```

```
# default squid options
# -D disables initial dns checks. If you most likely will not to have an
#    internet connection when you start squid, uncomment this
#SQUID_OPTS="-D"

RETVAL=0
case "$1" in
start)
   echo -n "Starting $SQUID: "
   for adir in $CACHE_SWAP; do
      if [ ! -d $adir/00 ]; then
              echo -n "init_cache_dir $adir... "
              $SQUID -z -F 2>/dev/null
          fi
   done
   $SQUID $SQUID_OPTS &
   RETVAL=$?
   echo $SQUID
   [ $RETVAL -eq 0 ] && touch /var/lock/subsys/$SQUID
   ;;

stop)
   echo -n "Stopping $SQUID: "
   $SQUID -k shutdown &
   RETVAL=$?
   if [ $RETVAL -eq 0 ] ; then
          rm -f /var/lock/subsys/$SQUID
          while : ; do
                  [ -f /var/run/squid.pid ] || break
                  sleep 2 && echo -n "."
          done
          echo "done"
   else
          echo
   fi
   ;;

reload)
   $SQUID $SQUID_OPTS -k reconfigure
   exit $?
   ;;

restart)
   $0 stop
   $0 start
   ;;

status)
   status $SQUID
   $SQUID -k check
   exit $?
   ;;

probe)
   exit 0;
   ;;

*)
   echo "Usage: $0 {start|stop|status|reload|restart}"
   exit 1
esac
```

```
exit $RETVAL
```

Now, make this script executable and change its default permission:
[root@deep]# **chmod 700 /etc/rc.d/init.d/squid**

Create the symbolic rc.d links for Squid with the command:
[root@deep]# **chkconfig --add squid**

By default squid script will not start automatically the proxy server on Red Hat Linux when you reboot the server. You can change it default by executing the following command:
[root@deep]# **chkconfig --level 345 squid on**

Start your new Squid Proxy Server manually with the following command:
[root@deep]# **/etc/rc.d/init.d/squid start**

Configuration of the "/etc/logrotate.d/squid" file

Configure your "/etc/logrotate.d/squid" file to rotate each week your log files automatically.

Create the **squid** file (touch /etc/logrotate.d/squid) and add:

```
/var/log/squid/access.log {
   weekly
   rotate 5
   copytruncate
   compress
   notifempty
   missingok
}
/var/log/squid/cache.log {
   weekly
   rotate 5
   copytruncate
   compress
   notifempty
   missingok
}

/var/log/squid/store.log {
   weekly
   rotate 5
   copytruncate
   compress
   notifempty
   missingok
# This script asks squid to rotate its logs on its own.
# Restarting squid is a long process and it is not worth
# doing it just to rotate logs
   postrotate
     /usr/sbin/squid -k rotate
   endscript
}
```

Securing Squid

More control on mounting a file system

You can have more control on mounting a file system like "/cache" with some nifty options like noexec, nodev, and nosuid. This can be setup in "/etc/fstab":

Edit your **fstab** file (vi /etc/fstab) and add:

/dev/sda8 /cache ext2 nosuid,nodev,noexec 1 2

Assuming “/dev/sda8” is the partition where your “/cache” for Squid live.

Which mean for <nodev> do not interpret character or block special devices on the file system, for <nosuid> do not allow set-user-identifier or set-group-identifier bits to take effect and for <noexec> do not allow execution of any binaries on the mounted file system. Applying this procedure on the partition where Squid Cache reside will help to eliminate the possibility of DEV, SUID/SGID, and execution of any binaries.

Immunize important configuration files

The immutable bit can be used to prevent accidentally deleting or overwriting a file that must be protected. It also prevents someone from creating a symbolic link to this file. Once your “squid.conf” file have been configured, it’s a good idea to immunize it with command like:

[root@deep]# **chattr +i /etc/squid/squid.conf**

Optimizing Squid

The noatime attribute

Linux has a mount option for filesystems call **noatime**. This option can be added to the mount options field in “/etc/fstab”. When a filesystem is mounted with this option, read accesses to the file will no longer result in an update to the atime information associated with the file. The atime info is generally not all that useful, so the lacks of updates to this field are not often relevant.

The importance of the **noatime** setting is that it eliminates the need to make writes to the filesystem for files, which are simply being read. Since writes tend to be somewhat expensive, this can result in measurable performance gains. Note that the wtime information will continue to be updated anytime the file is written to.

Edit the **fstab** file (vi /etc/fstab) and add:

E.I: /dev/sda8 /cache ext2 nosuid,nodev,noexec,**noatime** 1 2

Assuming ”/dev/sda8” is the partition where your cache directory for Squid reside on the server.

Reboot your system and then test your results with the command:

[root@deep]# **reboot**
[root@deep]# **cat /proc/mounts**

The bdflush parameter

This documentation is for the sysctl files in “/proc/sys/vm” and is valid for Linux kernel version 2.2. The files in this directory can be used to tune the operation of the virtual memory (VM) subsystem of the Linux kernel, and one of the files (bdflush) also has a little influence on disk usage.

This file (bdflush) controls the operation of the bdflush kernel daemon. We generally use this command to improve filesystem performance:

echo "100 1200 128 512 15 5000 500 1884 2">/proc/sys/vm/bdflush

Add the above commands to “/etc/rc.d/rc.local” and you’ll not have to type it again the next time you reboot your system.

Be changing some values from the default and the system seems more responsive, e.g. it waits a little more to write to disk and thus avoids some disk access contention.

Look at “/usr/src/linux/Documentation/sysctl/vm.txt” for more information on how to improve kernel parameters related to virtual memory, disk cache, swap, etc.

The ip_local_port_range parameter
This documentation is for the sysctl files in “/proc/sys/net/ipv4/ip_local_port_range” and is valid for Linux kernel version 2.2. ip_local_port_range - 2 INTEGERS.

Defines the local port range that is used by TCP and UDP to choose the local port. The first number is the first (port), the second the last local port number. For high-usage systems change this to 32768-61000.

echo “32768 61000” > /proc/sys/net/ipv4/ip_local_port_range

Add the above commands to “**/etc/rc.d/rc.local”** file and you’ll not have to type it again the next time if you reboot your system.

Physical memory
The most important resource for Squid is physical memory. Your processor does not need to be ultra-fast. Your disk system will be the major bottleneck, so fast disks are important for high-volume caches. Do not use IDE disks if you can help it.

Installed files

> /etc/squid
> /etc/squid/mib.txt
> /etc/squid/squid.conf.default
> /etc/squid/squid.conf
> /etc/squid/mime.conf.default
> /etc/squid/mime.conf
> /etc/squid/errors
> /etc/squid/errors/ERR_ACCESS_DENIED
> /etc/squid/errors/ERR_CACHE_ACCESS_DENIED
> /etc/squid/errors/ERR_CACHE_MGR_ACCESS_DENIED
> /etc/squid/errors/ERR_CANNOT_FORWARD
> /etc/squid/errors/ERR_CONNECT_FAIL
> /etc/squid/errors/ERR_DNS_FAIL
> /etc/squid/errors/ERR_FORWARDING_DENIED
> /etc/squid/errors/ERR_FTP_DISABLED
> /etc/squid/errors/ERR_FTP_FAILURE
> /etc/squid/errors/ERR_FTP_FORBIDDEN
> /etc/squid/errors/ERR_FTP_NOT_FOUND
> /etc/squid/errors/ERR_FTP_PUT_CREATED
> /etc/squid/errors/ERR_FTP_PUT_ERROR
> /etc/squid/errors/ERR_FTP_PUT_MODIFIED
> /etc/squid/errors/ERR_FTP_UNAVAILABLE
> /etc/squid/errors/ERR_INVALID_REQ
> /etc/squid/errors/ERR_INVALID_URL
> /etc/squid/errors/ERR_LIFETIME_EXP
> /etc/squid/errors/ERR_NO_RELAY
> /etc/squid/errors/ERR_ONLY_IF_CACHED_MISS
> /etc/rc.d/rc4.d/S90squid
> /etc/rc.d/rc5.d/S90squid
> /etc/rc.d/rc6.d/K25squid
> /etc/logrotate.d/squid
> /usr/lib/squid
> /usr/lib/squid/dnsserver
> /usr/lib/squid/unlinkd
> /usr/lib/squid/cachemgr.cgi
> /usr/lib/squid/icons
> /usr/lib/squid/icons/anthony-binhex.gif
> /usr/lib/squid/icons/anthony-bomb.gif
> /usr/lib/squid/icons/anthony-box.gif
> /usr/lib/squid/icons/anthony-box2.gif
> /usr/lib/squid/icons/anthony-c.gif
> /usr/lib/squid/icons/anthony-compressed.gif
> /usr/lib/squid/icons/anthony-dir.gif
> /usr/lib/squid/icons/anthony-dirup.gif
> /usr/lib/squid/icons/anthony-dvi.gif
> /usr/lib/squid/icons/anthony-f.gif
> /usr/lib/squid/icons/anthony-image.gif
> /usr/lib/squid/icons/anthony-image2.gif
> /usr/lib/squid/icons/anthony-layout.gif
> /usr/lib/squid/icons/anthony-link.gif
> /usr/lib/squid/icons/anthony-movie.gif
> /usr/lib/squid/icons/anthony-pdf.gif
> /usr/lib/squid/icons/anthony-portal.gif
> /usr/lib/squid/icons/anthony-ps.gif

> /etc/squid/errors/ERR_READ_ERROR
> /etc/squid/errors/ERR_READ_TIMEOUT
> /etc/squid/errors/ERR_SHUTTING_DOWN
> /etc/squid/errors/ERR_SOCKET_FAILURE
> /etc/squid/errors/ERR_TOO_BIG
> /etc/squid/errors/ERR_UNSUP_REQ
> /etc/squid/errors/ERR_URN_RESOLVE
> /etc/squid/errors/ERR_WRITE_ERROR
> /etc/squid/errors/ERR_ZERO_SIZE_OBJECT
> /etc/rc.d/init.d/squid
> /etc/rc.d/rc0.d/K25squid
> /etc/rc.d/rc1.d/K25squid
> /etc/rc.d/rc2.d/K25squid
> /etc/rc.d/rc3.d/S90squid
> /usr/lib/squid/icons/anthony-quill.gif
> /usr/lib/squid/icons/anthony-script.gif
> /usr/lib/squid/icons/anthony-sound.gif
> /usr/lib/squid/icons/anthony-tar.gif
> /usr/lib/squid/icons/anthony-tex.gif
> /usr/lib/squid/icons/anthony-text.gif
> /usr/lib/squid/icons/anthony-unknown.gif
> /usr/lib/squid/icons/anthony-xbm.gif
> /usr/lib/squid/icons/anthony-xpm.gif
> /usr/sbin/RunCache
> /usr/sbin/RunAccel
> /usr/sbin/squid
> /usr/sbin/client
> /var/log/squid

Linux Apache Web Server

Overview

Apache is a full-featured web server with full support for the HTTP 1.1 standard, password authenticated web pages, and many other features. Apache is one of the most popular web servers available, and provides performance equal or better to commercial servers.

Because Apache is a complex package there are a lot of installation variants and options. For this different documents exists which explain special things: For more information's read this document when you want to install Apache under Unix. Here, I explain how to compile and optimize Apache with a lot options like mod_ssl, mod_perl, mod_php3, LDAP… because I don't want to make several individual document's from each one. Fill free to compile what you want, e.g. For Apache + PHP3 but without mod_ssl or PostgreSQL or… follow the section that speak about Apache and PHP3 and skip the rest.

As you noticed above there are a lot of possibilities, variants and options for installing Apache. So, in the following we provide some step-by-step examples where you can see how to build Apache with other third-party modules. For simplification we assume some prerequisites for each example. If these don't fit your situation you have to adjust the steps.

This session is geared for new Apache Webmasters, and describes how to install an Apache server and get it running. It also covers some basic ways in which you can adjust the configuration to improve the server's performance. In our configuration and installation we'll run Apache as non root-user and in a chrooted environment for optimal security.

These installation instructions assume

Commands are Unix-compatible.
The source path is "/var/tmp" (other paths are possible).
Installations were tested on RedHat Linux 6.1.
All steps in the installation will happen in superuser account "root".
Apache version number is 1.3.9
Mod_Perl version number is 1.21
Mod_SSL version number is 2_4_10-1_3_9
PHP version number is 3_0_13
MM version number is 1.0.12

Packages

Apache Homepage: http://www.apache.org/
Mod_ssl Homepage: http://www.modssl.org/
OpenSSL Homepage: http://www.openssl.org/
MM Homepage: http://www.engelschall.com/sw/mm/
Imap Homepage: http://www.washington.edu/imap/
PostgreSQL Homepage: http://www.postgresql.org/index.html
OpenLDAP Homepage: http://www.openldap.org/
Mod_perl Homepage: http://perl.apache.org/
Mod_php Homepage: http://www.php.net/

Prerequisites

OpenSSL should be already installed on your system (e.g. if you want Apache+mod_ssl)
PosgreSQL should be already installed on your system (e.g. if you want Apache+PHP3+Pgsql)
Mm should be already installed on your system (e.g. if you want Apache+mod_ssl+mm)
OpenLDAP should be already installed on your system (e.g. is you want Apache+PHP3+LDAP)
Perl should be already installed on your system (e.g. if you want Apache+mod_perl)
Imap should be already installed on your system (e.g. if you want Apache+PHP3+imap)

Why do I need to use MM?
Build the MM Shared Memory library when you want shared memory support in Apache/EAPI. For instance this allows mod_ssl to use a high-performance RAM-based session cache instead of a disk-based one.

Tarballs

It is a good idea to make a list of files on the system before you install Apache, and one afterwards, and then compare them using '**diff**' to find out what file it placed where. Simply run '**find /* > apache1**' before and '**find /* > apache2**' after you install the software's, and use '**diff apache1 apache2 > apache**' to get a list of what changed.

Compilation

Decompress the tarballs (tar.gz).

```
[root@deep]# cp apache_version.tar.gz /var/tmp
[root@deep]# cp mod_ssl-version-version.tar.gz /var/tmp
[root@deep]# cp mod_perl-version.tar.gz /var/tmp
[root@deep]# cp php-version.tar.gz /var/tmp
[root@deep]# cd /var/tmp/
[root@deep]# tar xzpf apache_version.tar.gz
[root@deep]# tar xzpf mod_ssl-version-version_tar.gz
[root@deep]# tar xzpf mod_perl-version.tar.gz
[root@deep]# tar xzpf php-version_tar.gz
```

Compile and Optimize

Apply mod-ssl to Apache source tree.
Cd into the new mod_ssl directory (cd "mod_ssl-version") and type the following commands on your terminal:

```
CC="egcs" \
CFLAGS="-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions" \
./configure \
```

```
--with-apache=../apache_1.3.9 \
--with-crt=/etc/ssl/certs/server.crt \
--with-key=/etc/ssl/private/server.key
```

Cd into the new Apache directory *(cd ../apache-version)* and type the following commands on your terminal:

[root@deep]# vi +331 src/include/httpd.h and change:

#define HARD_SERVER_LIMIT 256
To
#define HARD_SERVER_LIMIT **1024**

Pre-configure Apache for PHP3's configure step.
Cd into the new Apache directory (cd "../apache-version") and type the following commands on your terminal:

```
CC="egcs" \
OPTIM="-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions" \
CFLAGS="-DDYNAMIC_MODULE_LIMIT=0" \
./configure \
--prefix=/home/httpd \
--bindir=/usr/bin \
--sbindir=/usr/sbin \
--libexecdir=/usr/lib/apache \
--includedir=/usr/include/apache \
--sysconfdir=/etc/httpd/conf \
--localstatedir=/var \
--runtimedir=/var/run \
--logfiledir=/var/log/httpd \
--datadir=/home/httpd \
--proxycachedir=/var/cache/httpd \
--mandir=/usr/man
```

Configure PHP3 and apply it to the Apache source tree.
Cd into the new php3 directory (cd "../php-version") and type the following commands on your terminal:

Edit the **php3_pgsql.h** file (vi +46 functions/php3_pgsql.h) and change the lines:

#include <libpq-fe.h>
#include <libpq/libpq-fs.h>
To read:
#include </usr/include/pgsql/libpq-fe.h>
#include </usr/include/pgsql/libpq/libpq-fs.h>

```
CC="egcs" \
CFLAGS="-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions -I/usr/include/openssl" \
./configure \
--prefix=/usr \
--with-config-file-path=/etc/httpd \
--with-imap \
--enable-safe-mode \
--with-exec-dir=/usr/bin \
--disable-debug \
--with-apache=../apache_1.3.9 \
```

--with-pgsql \ (if you want database PostgreSQL)
--with-ldap \ (if you want LDAP database light directory)
--enable-memory-limit=yes \
--enable-debug=no

[root@deep]# make
[root@deep]# make install

Apply mod_perl to Apache source tree and build/install the Perl-side of mod_perl.
Cd into the new mod_perl directory (cd "../mod_perl-version") and type the following commands on your terminal:

perl Makefile.PL \
EVERYTHING=1 \
APACHE_SRC=../apache_1.3.9/src \
USE_APACI=1 \
PREP_HTTPD=1 \
DO_HTTPD=1

[root@deep]# make
[root@deep]# make install

Build/Install Apache with mod_ssl + mm + mod_perl and PHP3.
Cd into the new Apache directory (cd "../apache-version") and type the following commands on your terminal:

SSL_BASE=SYSTEM \ (require for mod_ssl).
EAPI_MM=SYSTEM \ (require for mm).
CC="egcs" \
OPTIM="-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions" \
CFLAGS="-DDYNAMIC_MODULE_LIMIT=0" \
./configure \
--prefix=/home/httpd \
--bindir=/usr/bin \
--sbindir=/usr/sbin \
--libexecdir=/usr/lib/apache \
--includedir=/usr/include/apache \
--sysconfdir=/etc/httpd/conf \
--localstatedir=/var \
--runtimedir=/var/run \
--logfiledir=/var/log/httpd \
--datadir=/home/httpd \
--proxycachedir=/var/cache/httpd \
--mandir=/usr/man \
--add-module=src/modules/experimental/mod_mmap_static.c \ (if you are intended to use mod_mmap).
--add-module=src/modules/standard/mod_auth_db.c \ (if you are intended to use mod_auth).
--enable-module=ssl \ (require for mod_ssl).
--enable-rule=SSL_SDBM \ (require for mod_ssl).
--disable-rule=SSL_COMPAT \ (require for mod_ssl).
--activate-module=src/modules/php3/libphp3.a \ (require for php).
--enable-module=php3 \ (require for php).
--activate-module=src/modules/perl/libperl.a \ (require for mod_perl).
--enable-module=perl \ (require for mod_perl).
--disable-module=include \
--disable-module=status \
--disable-module=userdir \
--disable-module=negotiation \
--disable-module=autoindex \
--disable-module=asis \
--disable-module=imap \

```
--disable-module=env \
--disable-module=actions
```

This tells Apache to set itself up for this particular hardware setup with:

- module mod_mmap to improve performance.
- module mod_auth for password authentication security.
- module mod_ssl for data encryptions and secure commerce.
- module mod_php3 for php capability and improve the load of web pages build in php.
- module mod_perl for better security and performance than cgi script.
- disable module include
- disable module status
- disable module userdir
- disable module negotiation
- disable module autoindex
- disable module asis
- disable module imap
- disable module env
- disable module actions

NOTE: Removing all modules that you don't need will improve the performance of your Apache Web Server.

```
[root@deep]# make
[root@deep]# make install
[root@deep]# rm -f /usr/sbin/apachectl
[root@deep]# rm -f /usr/man/man8/apachectl.8
[root@deep]# rm -rf /home/httpd/icons/
[root@deep]# rm -rf /home/httpd/htdocs/
[root@deep]# cd /var/tmp/php-version
[root@deep]# install -m 644 php3.ini.dist /etc/httpd/php.ini
[root@deep]# rm -rf /etc/httpd/conf/ssl.crl/
[root@deep]# rm -rf /etc/httpd/conf/ssl.crt/
[root@deep]# rm -rf /etc/httpd/conf/ssl.csr/
[root@deep]# rm -rf /etc/httpd/conf/ssl.key/
[root@deep]# rm -rf /etc/httpd/conf/ssl.prm/
[root@deep]# rm -f /etc/httpd/conf/srm.conf srm.conf.default access.conf access.conf.default
```

Cleanup after work

```
[root@deep]# cd /var/tmp
[root@deep]# rm -rf apache-version/ apache-version.tar.gz mod_ssl-version-version/ mod_ssl-version-version.tar.gz php-version/ php-version.tar.gz mod_perl-version/ mod_perl-version.tar.gz
```

Configurations

Configuration files for different services are very specific depending of your need and your network architecture. Someone can install Apache Server for showing web pages only, other can install it with database connectivity or ebusiness with ssl protocol ect. In this book, we provide you an "httpd.conf" file setting with php, perl, ssl, ldap, and password authentication to show you different possibility.

We'll focus on optimization and security of these files and let all specific adjustments to your tastes. So you will need to read documentation that comes with these software and understand them.

All software we describe in our book "Linuxsos.pdf" will have a specific directory and subdirectory in a tar compressed archive named "floppy.tgz" containing file configurations for the specific program. If you get this archive file, you wouldn't be obliged to reproduce the different

configuration files bellow manually or cut and past them to create your configuration files. Whatever your decide to copy manually or get the files made to your convenience from the archive compressed files, it will be to your responsibility to modify, adjust for your needs and place the files related to Apache software to their appropriated places on your server machine, like show bellow. The server configuration files archive is located at the following Internet address: http://pages.infinit.net/lotus1/doc/opti/floppy.tgz

- To run Apache server, the following files are require and must be create or copied to their appropriated directories on your server.

  Copy the **httpd.conf** file to the "/etc/httpd/conf/" directory.
  Copy the **apache** file to the "/etc/logrotate.d/" directory.
  Copy the **httpd** script file to the "/etc/rc.d/init.d/" directory.

You can obtain configuration files listed bellow on our floppy.tgz archive. Copy the following files from the decompressed floppy.tgz archive to their appropriated places or copy and paste them directly from this book to the concerned file.

Configuration of the "/etc/httpd/conf/httpd.conf" file

Configure your"/etc/httpd/conf/httpd.conf" file. The following example is a minimal configuration for Apache with SSL protocol.

Edit the **httpd.conf** file (vi /etc/httpd/conf/httpd.conf) and add:

```
ServerType standalone
ResourceConfig /dev/null
AccessConfig /dev/null
ServerRoot "/etc/httpd"
PidFile /var/run/httpd.pid
Timeout 300
KeepAlive On
MaxKeepAliveRequests 0
KeepAliveTimeout 15
MinSpareServers 16
MaxSpareServers 64
StartServers 16
MaxClients 512
MaxRequestsPerChild 100000
Port 80

<IfDefine SSL>
Listen 192.168.1.1:80
Listen 192.168.1.1:443
</IfDefine>

User www
Group www
ServerAdmin admin@openarch.com
ServerName www.openarch.com
DocumentRoot "/home/httpd/ona"
Include conf/mmap.conf

<Directory />
   Options None
   AllowOverride None
   Order deny,allow
   Deny from all
</Directory>
```

```
<Directory /home/httpd>
   Options None
   AllowOverride None
   Order allow,deny
   Allow from all
</Directory>

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

DirectoryIndex index.htm index.php index.html default.html index.cgi
UseCanonicalName On
TypesConfig /etc/httpd/conf/mime.types
DefaultType text/plain
HostnameLookups Off

ErrorLog /var/log/httpd/error_log
LogLevel warn
LogFormat "%h %v %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
/etc/httpd/conf/httpd.conf
SetEnvIf Request_URI \.gif$ gif-image
CustomLog /var/log/httpd/access_log combined env=!gif-image
ServerSignature Off

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

AddType application/x-httpd-php3 .php3
AddType application/x-httpd-php3-source .phps

ErrorDocument 500 "The server made a boo boo.
ErrorDocument 404 http://www.openarch.com/error.htm
ErrorDocument 403 "Access Forbidden -- Go away.
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

<IfDefine SSL>
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl
</IfDefine>

<IfModule mod_ssl.c>
SSLPassPhraseDialog  builtin
SSLSessionCache        dbm:/var/run/ssl_scache
SSLSessionCacheTimeout  300
SSLMutex  file:/var/run/ssl_mutex
SSLRandomSeed startup builtin
SSLRandomSeed connect builtin
SSLLog      /var/log/httpd/ssl_engine_log
```

```
SSLLogLevel warn
</IfModule>

<IfDefine SSL>
<VirtualHost _default_:443>
DocumentRoot "/home/httpd/ona"
ServerName www.openarch.com
ServerAdmin admin@openarch.com
ErrorLog /var/log/httpd/error_log
SSLEngine on
SSLCipherSuite ALL:!ADH:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
SSLCertificateFile    /etc/ssl/certs/server.crt
SSLCertificateKeyFile /etc/ssl/private/server.key
SSLCACertificatePath    /etc/ssl/certs
SSLCACertificateFile    /etc/ssl/certs/ca.crt
SSLCARevocationPath     /etc/ssl/crl
SSLVerifyClient none
SSLVerifyDepth  10

SSLOptions +ExportCertData +StrictRequire
SetEnvIf User-Agent ".*MSIE.*" nokeepalive ssl-unclean-shutdown
SetEnvIf Request_URI \.gif$ gif-image
CustomLog /var/log/httpd/ssl_request_log \
        "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b" env=!gif-image
</VirtualHost>
</IfDefine>
```

NOTE: if you use the mod_php3 module with your Apache server don't forget to include in your "/etc/httpd/conf/httpd.conf" file the following line:

```
AddType application/x-httpd-php3 .php3
AddType application/x-httpd-php3-source .phps
```

NOTE: if you use the mod_perl module with your Apache server don't forget to include in your "/etc/httpd/conf/httpd.conf" file the following line to be able to see status of your different perls modules on the server:

```
<Location /perl-status>
        SetHandler  perl-script
        PerlHandler Apache::Status
        Order deny,allow
        Deny from all
        Allow from 192.168.1.3
 </Location>
```

Configuration of the "/etc/logrotate.d/apache" file

Configure your "/etc/logrotate.d/apache" file to rotate each week your log files automatically.

Create the **apache** file (touch /etc/logrotate.d/apache) and add:

```
/var/log/httpd/access_log {
   missingok
   postrotate
      /usr/bin/killall -HUP httpd
   endscript
}

/var/log/httpd/error_log {
   missingok
```

```
    postrotate
        /usr/bin/killall -HUP httpd
    endscript
}

/var/log/httpd/ssl_request_log {
    missingok
    postrotate
        /usr/bin/killall -HUP httpd
    endscript
}
```

Configuration of the "/etc/rc.d/init.d/httpd" script file

Configure your "/etc/rc.d/init.d/httpd" script file to start and stop Apache server.

Create the **httpd** script file (touch /etc/rc.d/init.d/httpd) and add:

```
#!/bin/sh
#
# Startup script for the Apache Web Server
#
# chkconfig: 345 85 15
# description: Apache is a World Wide Web server.  It is used to serve \
#              HTML files and CGI.
# processname: httpd
# pidfile: /var/run/httpd.pid
# config: /etc/httpd/conf/httpd.conf


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

# See how we were called.
case "$1" in
  start)
        echo -n "Starting httpd: "
        daemon httpd -DSSL
        echo
        touch /var/lock/subsys/httpd
        ;;
  stop)
        echo -n "Shutting down http: "
        killproc httpd
        echo
        rm -f /var/lock/subsys/httpd
        rm -f /var/run/httpd.pid
        ;;
  status)
        status httpd
        ;;
  restart)
        $0 stop
        $0 start
        ;;
  reload)
        echo -n "Reloading httpd: "
        killproc httpd -HUP
        echo
        ;;
  *)
```

```
        echo "Usage: $0 {start|stop|restart|reload|status}"
        exit 1
esac

exit 0
```

Now, make this script executable and change its default permission:
[root@deep]# **chmod 700 /etc/rc.d/init.d/httpd**

Create the symbolic rc.d links for Apache with the command:
[root@deep]# **chkconfig --add httpd**

Start your new Apache server manually with the following command:
[root@deep]# **/etc/rc.d/init.d/httpd start**

NOTE: The “-DSSL” option will start Apache on mode SSL. If you wan to start Apache on regular mode remove the “-DSSL” option near “daemon httpd”.

Securing Apache

There are several important configuration options that affect security. The first is the user that the web server runs as (User and Group, in httpd.conf). It is best to choose a user that has as few privileges possible - ideally, a user created just for the purpose of running the web server daemon. Create this user (We generally call it www) and ensure that it has minimal access to the system and its functions.

[root@deep]# **groupadd -g 80 www**
[root@deep]# **useradd -g 80 -u 80 www**

Change some important permission files and directory of your Web Server.

[root@deep]# **chmod 511 /usr/sbin/httpd**
[root@deep]# **chmod 750 /etc/httpd/conf/**
[root@deep]# **chmod 750 /var/log/httpd/**

You can create an http subdirectory which is modifiable by other users -- since root never executes any files out of there, and shouldn't be creating files in there (**e.g. “/home/httpd/ona”**).

Automatic indexing

By default, Apache usually comes with automatic indexing of directories enabled (IndexOptions in httpd.conf). This means any requests for a directory that don't find a index file, will build an index of what is in the directory. In many cases, this is a security issue, as you many only want people seeing files that you specifically link to.

To turn this off, you need to remove read permissions from the directory (but not the files inside) - that is, **chmod 311**. Depending on the ownership of the directory, it should look like:

[root@deep]# **cd /home/httpd/**
[root@deep]# **chmod 311 ona**
[root@deep]# **ls -la**

```
d-wx--x--x   13 webmaster webmaster     1024 Jul 28 08:12 ona
```

Then, any requests for this protected directory should return an error message something like:

Forbidden

You don't have permission to access “/ona/” on this server.

More control on mounting a file system
You can have more control on mounting a file system like “/chroot” with some nifty options like nosuid (for chroot file system refer to section “Running Apache in a chroot jail” bellow). This can be setup in “/etc/fstab”:

Edit the **fstab** file (vi /etc/fstab) and add depending of your needs:

/dev/sda7 /chroot ext2 nosuid 1 2

-Which mean for <*nosuid*> do not allow set-user-identifier or set-group-identifier bits to take. Applying this procedure on the partition where Apache reside will help to eliminate the possibility of SUID/SGID.

Create the .dbmpasswd password file for authentication
Needed only if you thing that you'll use an access file authentication for your site.

[root@deep]# **chmod 750 /usr/bin/dbmmanage**
[root@deep]# **/usr/bin/dbmmanage /etc/httpd/.dbmpasswd adduser username**

-Where <.dbmpasswd> is the name of the password file, <username> is the name of the user you want to add in your “.dbmpasswd” file.

Immunize important configuration files
The immutable bit can be used to prevent accidentally deleting or overwriting a file that must be protected. It also prevents someone from creating a symbolic link to this file. Once your “httpd.conf” file have been configured, it's a good idea to immunize it with command like:

[root@deep]# **chattr +i /etc/httpd/conf/httpd.conf**

Running Apache in a chroot jail
This part focuses on preventing Apache from being used as a point of break-in to the system hosting it. Apache by default run **as a non-root user**, which will limit any damage to what can be done as a normal user with a local shell. Of course, allowing what amounts to an anonymous guest account falls rather short of the security requirements for most Apache servers, so an additional step can be taken - that is, **running Apache in a chroot jail**.

The main benefit of a chroot jail is that the jail will limit the portion of the filesystem the daemon can see to the root directory of the jail. Additionally, since the jail only needs to support Apache, the programs available in the jail can be extremely limited. Most importantly, there is no need for setuid-root programs, which, given the right (or wrong...) bug, can be used to gain root access and break out of the jail.

Chrooting apache is no easy task and has a tendency to break things. Before we embark on this, we need to first decide whether it is beneficial for you to do so. Some pros and cons are but most certainly not limited to:

Pros:

- If apache is ever compromised, the attacker will not have access to the entire file system. Also make sure apache is running as its own unique user/group, and not something that's overly used, such as "nobody." Consider the scenario where a web server is running as

nobody or any other overly used UID/GID and compromised. The cracker can now access any other processes running as nobody from within the chroot.

- Poorly written CGI scripts that may, for example, allow someone to email your "/etc/passwd" file will not work.

Cons:

- The extra libraries you'll need to have in the chroot. Using hardlinks or compiling your binaries statically will help here. Static binaries are also nifty because they eliminate the possibility -- however infinitesimal it may be -- of someone replacing your libc or other shared library with some hostile wrapper lib and having apache run it unknowingly.

- Generally speaking, the more nifty stuff your web server does, the more difficult it will be to not break that functionality. For example, if you use any Perl/CGI, you will need to copy the needed binaries and perl libraries to the appropriate spot within the chroot space. The same applies for SSL, PHP and other.

The chrooted configuration listed bellow suppose that you're compiled your Apache server with **mod_ssl**, **mod_php** and **mod_auth_db** for password authentication. The differences of what you're compiled with your Apache web server resides in which libraries and binaries we'll need to copy in our chrooted lib directory.

Remember that if you've compiled Apache to use **mod_perl** you must copy the needed binary and perl libraries to the chrooted directory. Perl reside in "/usr/lib/perl5" directory and in the case you use perl, copy the perl directories to "/chroot/httpd/usr/lib/perl5/". Don't forget to create this directory ("/chroot/httpd/usr/lib/perl5") in your chrooted structure before coping.

Find the shared library dependencies of httpd. These will need to be copied into the chroot jail later.

[root@deep]# **ldd /usr/sbin/httpd**

```
libpam.so.0 => /lib/libpam.so.0 (0x40016000)
libm.so.6 => /lib/libm.so.6 (0x4001f000)
libdl.so.2 => /lib/libdl.so.2 (0x4003b000)
libcrypt.so.1 => /lib/libcrypt.so.1 (0x4003e000)
libnsl.so.1 => /lib/libnsl.so.1 (0x4006b000)
libresolv.so.2 => /lib/libresolv.so.2 (0x40081000)
libdb.so.3 => /lib/libdb.so.3 (0x40090000)
libc.so.6 => /lib/libc.so.6 (0x400cb000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Make a note of the files listed above; you will need these later.

Step 1:
Add a new user id and a new group id if this is not already done for running httpd. This is important because running it as root defeats the purpose of the jail, and using a different user id that already exists on the system can allow your services to access each others' resources. Think multi-layer security.

These are sample user and group id numbers. Check the "/etc/passwd" and "/etc/group" files for a free uid/gid number. We'll use 80.

[root@deep]# **groupadd -g 80 www**
[root@deep]# **useradd -g 80 -u 80 www**

Step 2:

Set up the chroot environment. First we need to create the chrooted Apache structure. We use "/chroot/httpd" for the chrooted Apache. The "/chroot/httpd" is just a directory on a different partition where we've decided to put apache for more security.

[root@deep]# **/etc/rc.d/init.d/httpd stop** ← if Apache is already installed and run on your system.
[root@deep]# **mkdir /chroot/httpd**

Next, create the rest of directories like the following:

[root@deep]# **mkdir /chroot/httpd/dev**
[root@deep]# **mkdir /chroot/httpd/lib**
[root@deep]# **mkdir /chroot/httpd/etc**
[root@deep]# **mkdir -p /chroot/httpd/usr/sbin**
[root@deep]# **mkdir -p /chroot/httpd/var/run**
[root@deep]# **mkdir -p /chroot/httpd/var/log/httpd**
[root@deep]# **chmod 750 /chroot/httpd/var/log/httpd/**
[root@deep]# **mkdir -p /chroot/httpd/home/httpd**

Copy the main configuration directory, the configuration files, the cgi-bin directory, the root directory and the httpd program:

[root@deep]# **cp -r /etc/httpd /chroot/httpd/etc/**
[root@deep]# **cp -r /home/httpd/cgi-bin /chroot/httpd/home/httpd/**
[root@deep]# **cp -r /home/httpd/your-DocumentRoot /chroot/httpd/home/httpd/**
[root@deep]# **mknod /chroot/httpd/dev/null c 1 3**
[root@deep]# **chmod 666 /chroot/httpd/dev/null**
[root@deep]# **cp /usr/sbin/httpd /chroot/httpd/usr/sbin/**
[root@deep]# **cp -r /etc/ssl /chroot/httpd/etc/** ← require only if you use mod_ssl.
[root@deep]# **chmod 600 /chroot/httpd/etc/ssl/certs/ca.crt**
[root@deep]# **chmod 600 /chroot/httpd//etc/ssl/certs/server.crt**
[root@deep]# **chmod 600 /chroot/httpd/etc/ssl/private/ca.key**
[root@deep]# **chmod 600 /chroot/httpd/etc/ssl/private/server.key**

We need the "/chroot/httpd/etc", "/chroot/httpd/dev", "/chroot/httpd/lib", "/chroot/httpd/usr/sbin", "/chroot/httpd/var/run", "/chroot/httpd/home/httpd" and "/chroot/httpd/var/log/httpd" directories because, from the point of the chroot, we're sitting at "/".

Since we are compiled apache to use shared libraries; we need to install them into the chroot directory structure. Use **ldd /chroot/httpd/usr/sbin/httpd** to find out which libraries are needed. The output (depending of what you've compiled with Apache) will be something similar to:

libpam.so.0 => /lib/libpam.so.0 (0x40016000)
libm.so.6 => /lib/libm.so.6 (0x4001f000)
libdl.so.2 => /lib/libdl.so.2 (0x4003b000)
libcrypt.so.1 => /lib/libcrypt.so.1 (0x4003e000)
libnsl.so.1 => /lib/libnsl.so.1 (0x4006b000)
libresolv.so.2 => /lib/libresolv.so.2 (0x40081000)
libdb.so.3 => /lib/libdb.so.3 (0x40090000)
libc.so.6 => /lib/libc.so.6 (0x400cb000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)

Copy the shared libraries identified above:

[root@deep]# **cp /lib/libpam.so.0 /chroot/httpd/lib/**
[root@deep]# **cp /lib/libm.so.6 /chroot/httpd/lib/**
[root@deep]# **cp /lib/libdl.so.2 /chroot/httpd/lib/**
[root@deep]# **cp /lib/libcrypt.so.1 /chroot/httpd/lib/**
[root@deep]# **cp /lib/libnsl* /chroot/httpd/lib/**
[root@deep]# **cp /lib/libresolv* /chroot/httpd/lib/**
[root@deep]# **cp /lib/libdb.so.3 /chroot/httpd/lib/**

[root@deep]# **cp /lib/libc.so.6 /chroot/httpd/lib/**
[root@deep]# **cp /lib/ld-linux.so.2 /chroot/httpd/lib/**

You'll also need the following extra libraries for some network functions like resolving:

[root@deep]# **cp /lib/libnss_compat* /chroot/httpd/lib/**
[root@deep]# **cp /lib/libnss_dns* /chroot/httpd/lib/**
[root@deep]# **cp /lib/libnss_files* /chroot/httpd/lib/**

We now need to copy passwd and group files inside the "/chroot/httpd/etc" chrooted directory. The concept here is how ftpd uses passwd and group files.

[root@deep]# **cp /etc/passwd /chroot/httpd/etc/**
[root@deep]# **cp /etc/group /chroot/httpd/etc/**

Next, remove all entries except for the user that apache runs as in both files (passwd and group). You will also need "**/etc/resolv.conf", "/etc/nsswitch.conf"** and "**/etc/hosts"** files.

Edit the **passwd** file (vi /chroot/httpd/etc/passwd) and delete all entries except the user apache run as (in our configuration is "www"):

- Set the immutable bit on "passwd" file:
  [root@deep]# **cd /chroot/httpd/etc/**
  [root@deep]# **chattr +i passwd**

Edit the **group** file (vi /chroot/httpd/etc/group) and delete all entries except the group apache run as (in our configuration is "www"):

- Set the immutable bit on "group" file:
  [root@deep]# **cd /chroot/httpd/etc/**
  [root@deep]# **chattr +i group**

- Set the immutable bit on "httpd.conf" file:
  [root@deep]# **cd /chroot/httpd/etc/httpd/conf/**
  [root@deep]# **chattr +i httpd.conf**

With the immutable bit set, files cannot be deleted or renamed, no link can be created to this file and no data can be written to the file. Only the superuser can set or clear this attribute.

[root@deep]# **cp /etc/resolv.conf /chroot/httpd/etc/**
[root@deep]# **cp /etc/hosts /chroot/httpd/etc/**
[root@deep]# **cp /etc/nsswitch.conf /chroot/httpd/etc/**

- Set the immutable bit on "resolv.conf" file:
  [root@deep]# **cd /chroot/httpd/etc/**
  [root@deep]# **chattr +i resolv.conf**

- Set the immutable bit on "hosts" file:
  [root@deep]# **cd /chroot/httpd/etc/**
  [root@deep]# **chattr +i hosts**

- Set the immutable bit on "nsswitch.conf" file:
  [root@deep]# **cd /chroot/httpd/etc/**
  [root@deep]# **chattr +i nsswitch.conf**

Copy the localtime file to the jail so that log entries are adjusted for your local timezone properly:

[root@deep]# **cp /etc/localtime /chroot/httpd/etc/**

Remove unnecessary old files and directories:

[root@deep]# **rm -rf /var/log/httpd/**
[root@deep]# **rm -rf /etc/httpd/**
[root@deep]# **rm -rf /home/httpd/**
[root@deep]# **rm -f /usr/sbin/httpd**

Step 3:
Tell syslogd about the new chrooted service.

Normally, processes talk to syslogd through “/dev/log”. As a result of the chroot jail, this won't be possible, so syslogd needs to be told to listen to “/chroot/httpd/dev/log”. To do this, edit the syslog startup script to specify additional places to listen.

Edit the **syslog** script (vi /etc/rc.d/init.d/syslog) to change the line:

daemon syslogd -m 0
To read:
daemon syslogd -m 0 -a /chroot/httpd/dev/log

Step 4:
Edit the **httpd** script (vi /etc/rc.d/init.d/httpd) to change the line:

daemon httpd
To read:
/usr/sbin/chroot /chroot/httpd/ /usr/sbin/httpd -DSSL

rm -f /var/run/httpd.pid
To read:
rm -f /chroot/httpd/var/run/httpd.pid

Red Hat's init scripts' daemon() function doesn't allow alternate PID files to be specified, but that won't affect the operation of the start and stop actions of the httpd init scripts since they will be called from outside the chroot jail.

Step 5:
Test the new chrooted configuration! Restart syslogd:

root@deep]# **/etc/rc.d/init.d/syslog stop**
root@deep]# **/etc/rc.d/init.d/syslog start**

Now, start the new chrooted Apache:

Whew, we're finished! Try it out. **/etc/rc.d/init.d/httpd start**. If you don't get any errors, do a **ps auwx | grep httpd** and see if we're running. If so, lets check to make sure it's chrooted by picking out one of the process numbers of httpd and doing **ls -la /proc/that_process_number/root/**. If you see:

dev
etc
home
lib
usr
var

congratulations!

As mentioned above, if you use Perl, you'll need to copy or hardlinks any system libraries, perl libraries “/usr/lib/perl5”, and binaries into the chroot area. The same applies for SSL, PHP and other.

Configuration of the new “/etc/logrotate.d/apache” file

Now Apache logs file reside on “/chroot/var/log/httpd” directory instead of “/var/log/httpd”, for this reason we need to modify our “/etc/logrotate.d/httpd” file to point to the new chrooted directory. Also we’re compiled Apache with mod_ssl and we’ll add one more line to permit logrotate program to rotate the ssl_request_log file. Configure your “/etc/logrotate.d/apache” file to rotate each week your log files automatically.

Create the **apache** file (touch /etc/logrotate.d/apache) and add:

```
/chroot/httpd/var/log/httpd/access_log {
   missingok
   postrotate
      /usr/bin/killall -HUP /chroot/httpd/usr/sbin/httpd
   endscript
}

/chroot/httpd/var/log/httpd/error_log {
   missingok
   postrotate
      /usr/bin/killall -HUP /chroot/httpd/usr/sbin/httpd
   endscript
}

/chroot/httpd/var/log/httpd/ssl_request_log {
   missingok
   postrotate
      /usr/bin/killall -HUP /chroot/httpd/usr/sbin/httpd
   endscript
}

/chroot/httpd/var/log/httpd/ssl_engine_log {
   missingok
   postrotate
      /usr/bin/killall -HUP /chroot/httpd/usr/sbin/httpd
   endscript
}
```

Optimizing Apache

The static file

For static file, compile **mod_mmap_static** (if you are follow what I described in the compilation time above, this is already done --add-module-../mod_mmap_static.c) into Apache (see http://www.apache.org/docs/mod/mod_mmap_static.html) and configure Apache to memory-map the static documents, e.g. by creating a config file like this as root:

```
[root@deep]# find /home/httpd/htdocs -type f -print | sed -e 's/.*/mmapfile &/' > /etc/httpd/conf/mmap.conf
```

And including “mmap.conf” in your Apache config file like this:
[root@deep]# vi /etc/httpd/conf/httpd.conf and add the line: **Include conf/mmap.conf** somewhere in the “httpd.conf” file.

The noatime
Linux has a mount option for filesystems call **noatime**. This option can be added to the mount options field in "/etc/fstab". When a filesystem is mounted with this option, read accesses to the file will no longer result in an update to the atime information associated with the file. The atime info is generally not all that useful, so the lacks of updates to this field are not often relevant.

The importance of the **noatime** setting is that it eliminates the need to make writes to the filesystem for files, which are simply being read. Since writes tend to be somewhat expensive, this can result in measurable performance gains. Note that the wtime information will continue to be updated anytime the file is written to.

Edit the **fstab** file (vi /etc/fstab) and add noatime option to:

```
/dev/sda7       /chroot       ext2     nosuid,nodev,noatime       1 2
```

Assuming "/dev/sda7" is your partition where the "/chroot" live.

Reboot your system and then test your results with the command:

```
[root@deep]# reboot
[root@deep]# cat /proc/mounts
```

The ip_local_port_range parameter
This documentation is for the sysctl files in "/proc/sys/net/ipv4/ip_local_port_range" and is valid for Linux kernel version 2.2. ip_local_port_range - 2 INTEGERS.

Defines the local port range that is used by TCP and UDP to choose the local port. The first number is the first (port), the second the last local port number. For high-usage systems change this to 32768-61000.

```
echo "32768 61000" > /proc/sys/net/ipv4/ip_local_port_range
```

Add the above commands to "/etc/rc.d/rc.local" file and you'll not have to type it again the next time if you reboot your system.

Installed files

```
> /etc/rc.d/init.d/httpd
> /etc/rc.d/rc0.d/K15httpd
> /etc/rc.d/rc1.d/K15httpd
> /etc/rc.d/rc2.d/K15httpd
> /etc/rc.d/rc3.d/S85httpd
> /etc/rc.d/rc4.d/S85httpd
> /etc/rc.d/rc5.d/S85httpd
> /etc/rc.d/rc6.d/K15httpd
> /etc/logrotate.d/apache
> /etc/httpd
> /etc/httpd/conf
> /etc/httpd/conf/httpd.conf.default
> /etc/httpd/conf/httpd.conf
> /etc/httpd/conf/mime.types.default
> /etc/httpd/conf/mime.types
> /etc/httpd/conf/magic.default
> /etc/httpd/conf/magic
> /etc/httpd/php.ini
> /home/httpd
> /usr/include/apache/ap_md5.h
> /usr/include/apache/ap_mm.h
> /usr/include/apache/ap_mmn.h
> /usr/include/apache/ap_sha1.h
> /usr/include/apache/buff.h
> /usr/include/apache/compat.h
> /usr/include/apache/conf.h
> /usr/include/apache/explain.h
> /usr/include/apache/fnmatch.h
> /usr/include/apache/hsregex.h
> /usr/include/apache/http_conf_globals.h
> /usr/include/apache/http_config.h
> /usr/include/apache/http_core.h
> /usr/include/apache/http_log.h
> /usr/include/apache/http_main.h
> /usr/include/apache/http_protocol.h
> /usr/include/apache/http_request.h
> /usr/include/apache/http_vhost.h
> /usr/include/apache/httpd.h
```

> /home/httpd/cgi-bin
> /home/httpd/cgi-bin/printenv
> /home/httpd/cgi-bin/test-cgi
> /usr/bin/htpasswd
> /usr/bin/htdigest
> /usr/bin/dbmmanage
> /usr/include/apache
> /usr/include/apache/xml
> /usr/include/apache/xml/asciitab.h
> /usr/include/apache/xml/hashtable.h
> /usr/include/apache/xml/iasciitab.h
> /usr/include/apache/xml/latin1tab.h
> /usr/include/apache/xml/nametab.h
> /usr/include/apache/xml/utf8tab.h
> /usr/include/apache/xml/xmldef.h
> /usr/include/apache/xml/xmlparse.h
> /usr/include/apache/xml/xmlrole.h
> /usr/include/apache/xml/xmltok.h
> /usr/include/apache/xml/xmltok_impl.h
> /usr/include/apache/alloc.h
> /usr/include/apache/ap.h
> /usr/include/apache/ap_compat.h
> /usr/include/apache/ap_config.h
> /usr/include/apache/ap_config_auto.h
> /usr/include/apache/ap_ctx.h
> /usr/include/apache/ap_ctype.h
> /usr/include/apache/ap_hook.h
> /usr/include/apache/multithread.h
> /usr/include/apache/rfc1413.h
> /usr/include/apache/scoreboard.h
> /usr/include/apache/util_date.h
> /usr/include/apache/util_md5.h
> /usr/include/apache/util_script.h
> /usr/include/apache/util_uri.h
> /usr/include/apache/os.h
> /usr/include/apache/os-inline.c
> /usr/lib/apache
> /usr/man/man1/htpasswd.1
> /usr/man/man1/htdigest.1
> /usr/man/man1/dbmmanage.1
> /usr/man/man8/ab.8
> /usr/man/man8/httpd.8
> /usr/man/man8/logresolve.8
> /usr/man/man8/rotatelogs.8
> /usr/man/man8/apxs.8
> /usr/sbin/httpd
> /usr/sbin/ab
> /usr/sbin/logresolve
> /usr/sbin/rotatelogs
> /usr/sbin/apxs
> /var/log/httpd
> /var/cache
> /var/cache/httpd

Optional component to install with Apache

Devel-Symdump
The perl module Devel::Symdump provides a convenient way to inspect perl's symbol table and the class hierarchie within a running program. From version 2.00, this module needs at least perl5.003. To build and install it, please follow this step.

Packages
Devel-Symdump Homepage: http://www.perl.com/CPAN/modules/by-module/Devel/

```
[root@deep]# cp Devel-Symdump-version.tar.gz /var/tmp/
[root@deep]# tar xzpf Devel-Symdump-version.tar.gz
```

Cd into the new devel-Symdump directory and type the following commands on your terminal:

```
[root@deep]# perl Makefile.PL
[root@deep]# make
[root@deep]# make test
[root@deep]# make install
```

Cleanup after work
```
[root@deep]# cd /var/tmp
[root@deep]# rm -rf devel-Symdump.version/ Devel-Symdump-version.tar.gz
```

CGI.pm
This is CGI.pm, an easy-to-use Perl5 library for writing World Wide Web CGI scripts. Older version of this software exists by default on your system and are buggy, please update to version 2.51 at least. To install this module, please follow this step.

Packages
CGI.pm Homepage: http://stein.cshl.org/WWW/software/CGI/cgi_docs.html

[root@deep]# cp CGI.pm.tar.gz /var/tmp/
[root@deep]# tar xzpf CGI.pm.tar.gz

Cd into the new CGI.pm directory and type the following commands on your terminal:

[root@deep]# perl Makefile.PL
[root@deep]# make
[root@deep]# make test
[root@deep]# make install

Cleanup after work
[root@deep]# cd /var/tmp
[root@deep]# rm -rf CGI.pm/ CGI.pm.tar.gz

FormMail
FormMail is a universal WWW form to E-mail gateway. There is only one required form input tag, which must be specified in order for this script to work with your existing forms. To install this script, please follow this step.

Packages
Formmail Homepage: http://www.worldwidemart.com/scripts/formmail.shtml

[root@deep]# cp formmail.tar.gz /var/tmp/
[root@deep]# tar xzpf formmail.tar.gz

Cd into the new Formmail directory and edit the following file:

[root@deep]# vi Formmail.pl

@referers = (**'openarch.com','192.168.1.1'**);
This array allows you to define the domains that you will allow forms to reside on and use your FormMail script.

The script, FormMail.pl, needs to be placed in your server's cgi-bin and the www user must have the ability to read/execute the script.

[root@deep]# cp Formmail.pl /home/httpd/cgi-bin/ (or wherever your CGIs live).
Cd to cgi-bin directory
[root@deep]# chmod 750 Formmail.pl
[root@deep]# chown 0.99 Formmail.pl

Cleanup after work
[root@deep]# cd /var/tmp
[root@deep]# rm -rf formmail/ formmail.tar.gz

Webalizer
The Webalizer is a web server log file analysis program, which produces usage statistics in HTML format for viewing with a browser. The results are presented in both columnar and graphical format, which facilitates interpretation.

Packages
Webalizer Homepage: http://www.mrunix.net/webalizer/

```
[root@deep]# cp webalizer-version-src.tgz /var/tmp/
[root@deep]# tar xzpf webalizer-version-src.tgz
```

The Webalizer requires the GD graphics library by Tom Boutell. If you don't already have it, install it from the Red Hat Linux 6.1 CD-ROM.

```
[root@deep]# rpm -Uvh gd-devel-version.i386.rpm
```

Cd into the new Webalizer directory and type the following commands on your terminal:

```
CC="egcs" \
CFLAGS="-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-frame-pointer -fno-exceptions" \
./configure \
--prefix=/usr
[root@deep]# make
[root@deep]# make install
[root@deep]# mkdir /home/httpd/usage
```

Add the following lines in your httpd.conf file:

```
Alias /usage/ "/home/httpd/usage/"
 <Directory "/home/httpd/usage">
        Order deny,allow
        Deny from all
        Allow from 192.168.1.3
</Directory>
```

Cleanup after work

```
[root@deep]# cd /var/tmp
[root@deep]# rm -rf webalizer-version/ webalizer-version-src.tgz
```

FAQ-O-Matic

The Faq-O-Matic is a CGI-based system that automates the process of maintaining a FAQ (or Frequently Asked Questions list). It allows visitors to your FAQ to take part in keeping it up-to-date. To install this program, please follow this step.

Packages

FAQ-O-Matic Homepage: http://www.dartmouth.edu/~jonh/ff-serve/cache/1.html

```
[root@deep]# cp FAQ-O-Matic-version.tar.gz /var/tmp/
[root@deep]# tar xzpf FAQ-O-Matic-version.tar.gz
```

First of all update your “CGI.pm” program to version 2.51 (see above) at least before installing “FAQ-O-Matic” then, cd into the new FAQ-O-Matic directory and type the following commands on your terminal:

```
[root@deep]# perl Makefile.PL
[root@deep]# make
[root@deep]# make install
[root@deep]# mv fom /home/httpd/cgi-bin/ (or wherever your CGIs live).
[root@deep]# mkdir -p /home/httpd/cgi-bin/fom-meta
[root@deep]# mkdir -p /home/httpd/faqomatic
[root@deep]# chown 0.www /home/httpd/cgi-bin/fom
[root@deep]# chown -R www.www /home/httpd/cgi-bin/fom-meta/
[root@deep]# chown -R www.www /home/httpd/faqomatic/
```

Add the following lines in your httpd.conf file:

```
Alias /faqomatic/ "/home/httpd/faqomatic/"
<Directory "/home/httpd/faqomatic">
        Order allow,deny
        Allow from all
</Directory>

Alias /bags/ "/home/httpd/faqomatic/bags/"
<Directory "/home/httpd/faqomatic/bags">
        Order allow,deny
        Allow from all
</Directory>

Alias /cache/ "/home/httpd/faqomatic/cache/"
<Directory "/home/httpd/faqomatic/cache">
        Order allow,deny
        Allow from all
</Directory>

Alias /item/ "/home/httpd/faqomatic/item/"
<Directory "/home/httpd/faqomatic/item">
        Order allow,deny
        Allow from all
</Directory>
```

Restart you webserver with the following command:
[root@deep]# **/etc/rc.d/init.d/httpd restart**

Netscape http://localhost/cgi-bin/fom (or whatever browser you prefer). (Or whatever the URL would be to execute the CGI).
Enter your temporary password
Create the fom-meta directory
Click first on "**Define configuration parameters**" and configure
Configure the following for example under: "**Mandatory**: Server directory configuration"
$serverBase= http://www.openarch.com
$cgiURL= /cgi-bin/fom
$serveDir= /home/httpd/faqomatic/
$serveURL= /faqomatic/
configure the rest as you need

Cleanup after work
[root@deep]# cd /var/tmp
[root@deep]# rm -rf FAQ-OMatic-version/ FAQ-OMatic-version.tar.gz

Webmail IMP
IMP is an IMAP Webmail client.
To install this program, please follow this step.

Packages
Webmail IMP Homepage: http://www.horde.org/imp/

[root@deep]# cp horde-version.tar.gz /home/httpd/
[root@deep]# tar xzpf horde-version.tar.gz
[root@deep]# mv horde-version horde
[root@deep]# cp imp-version.tar.gz /home/httpd/horde/
Change into the "horde" directory (cd /home/httpd/horde/), and untar/gzip imp-version.tar.gz
[root@deep]# tar xzpf imp-version.tar.gz

```
[root@deep]# rm -f imp-version.tar.gz
[root@deep]# mv imp-version imp
```
Add the following lines in your httpd.conf file:

```
Alias /horde/ "/home/httpd/horde/"
<Directory "/home/httpd/horde">
        Order allow,deny
        Allow from all
</Directory>

Alias /imp/ "/home/httpd/horde/imp/"
<Directory "/home/httpd/horde/imp">
        Options None
        Order allow,deny
        Allow from all
</Directory>
```

Restart you webserver with the following command:
[root@deep]# **/etc/rc.d/init.d/httpd restart**

New setup engine named “setup.php3” give people the ability to configure IMP via the web. For security reasons, it is disabled by default, but you can enable if by saying:

Cd into horde directory (cd /home/httpd/horde/) and type the following on your terminal
[root@deep]# sh ./install.sh

You should be able to point your browser to http:// <your imp server>/<your horde home>/setup.php3. At this point you can walk through the graphical setup program and configure all aspects of IMP.

When you are done be sure to disable it again!
Cd horde directory (cd /home/httpd/horde/) and type the following on your terminal
[root@deep]# sh ./secure.sh

Cleanup after work
[root@deep]# cd /var/tmp
[root@deep]# rm -rf horde-version.tar.gz

Linux IPX Netware ™ Client

Overview

Internet Packet exchange is a protocol used by the Novell corporation to provide internetworking support for their NetWare™ product.

Ncpfs is a filesystem, which understands the Novell NetWare(TM) NCP protocol. Functionally, NCP is used for NetWare the way NFS is used in the TCP/IP world. For a Linux system to mount a NetWare filesystem, it needs a special mount program. The ncpfs package contains such a mount program plus other tools for configuring and using the ncpfs filesystem. You must verify that ncpfs is installed on your system. Use the command **rpm -q ncpfs**.

The **ipxutils** package includes utilities (ipx_configure, ipx_internal_net, ipx_interface, and ipx_route) necessary for configuring and debugging IPX interfaces and networks under Linux. IPX

is the low-level protocol used by Novell's NetWare file server system to transfer data. You must verify that ipxutils is installed on your system. Use the command **rpm –q ipxutils**.

These installation instructions assume

Commands are Unix-compatible.
Installations were tested on RedHat Linux 6.1 Server.
All steps in the installation will happen in superuser account “root”.
ncpfs version number is 2.2.0.12
ipxutils version number is 2.2.0.12

Build a kernel with IPX support and NCP protocol

The first thing you need to do is ensure that your kernel has been built with IPX support enabled and NCP protocol. In the 2.2.14 kernel version you need ensure that you have answered **Y** to the following question:

The IPX protocol (CONFIG_IPX) [N] **Y**
NCP filesystem support (CONFIG_NCP_PS) [N] **Y**
Packet signatures (CONFIG_NCPFS_PACKET_SIGNING) [N] **Y**
Clear remove/delete inhibit when needed (CONFIG_NCPFS_STRONG) [N] **Y**
Use NFS namespace if available (CONFIG_NCPFS_NFS_NS) [N] **Y**
Use LONG (OS/2) namespace if available (CONFIG_NCPFS_OS2_NS) [No] **Y**
Allow mounting of volume subdirectories (CONFIG_NCPFS_MOUNT_SUBDIR) [N] **Y**
Enable symbolic links and execute flags (CONFIG_NCPFS_EXTRAS) [N] **Y**

Trying to set up an IPX only network interface with no TCP/IP

You can have the interface active without any protocols bound to it. Instead of using ifconfig to place IP numbers, etc. Simply say ifconfig ethN up

Assuming you want to setup eth1 as an IPX only network interface without TCP/IP, you must check that the following file exist (ifcfg-eth1) and be sure the lines IPADDR, NETMASK, NETWORK and BROADCAST contain no values:

[root@deep]# vi /etc/sysconfig/network-scripts/ifcfg-eth1

DEVICE=eth1
IPADDR=
NETMASK=
BROADCAST=
ONBOOT=no
BOOTPROTO=none
USERCTL=no

After, all you have to do is to restart the network daemon with the command: **/etc/rc.d/init.d/network restart** and make up the Ethernet card eth1 with the command**: ifconfig eth1 up**.

Since Linux by default configure the IPX protocol on both interfaces (eth0 and eth1) and make the first card it found the primary IPX interface (eth0) you must to deactivate the IPX protocol on this card (eth0) because we want to use our second card (eth1) to transfer IPX data’s.

To make this, use the command: **ipx_interface del eth0 802.3**. Now if you make an ifconfig, you will see that our network interfaces are configured in this way: eth0 TCP/IP protocol only and eth1 IPX protocol only.

Ncpfs User Commands

For automatic setting of the interface configuration and primary interface, use the command
[root@deep]# **ipx_configure –auto_interface=on –auto_primary=on**

To mount a Novell™ server or volume, use the command
[root@deep]# **ncpmount –S DMS01 /mnt/netware –U username –P passwd**

This command will mount fileserver DMS01, with a login id of username with password passwd, under the /mnt/netware directory. If you don't specify the –P option you will be prompted for a password.

To umount the /mnt/netware directory, use the command
[root@deep]# **ncpumount /mnt/netware**

To see a list of all of the Novell fileserver on your network, use the command
[root@deep]# **slist**

To copy files, use the command
[root@deep]# **ncopy file1 [file2…] directory**

There are a number of files related to the Linux IPX support that are located within the /proc filesystem. They are:

/proc/net/ipx_interface
This file contains information about the IPX interfaces configured on your machine. These may have been configured manually by command or automatically detected and configured.

/proc/net/ipx_route
This file contains a list of the routes that exist in the IPX routing table. These routes may have been added manually by command or automatically by an IPX routing daemon.

/proc/net/ipx
This file is a list of the IPX sockets that are currently open for use on the machine.

Linux FTP Server

Overview

Using the File Transfer Protocol (FTP) is a popular way to transfer files from machine to machine across a network. Clients and servers have been written for all the popular platforms, thereby often making FTP the most convenient way of performing file transfers.

You can configure FTP servers in many different ways. One is as a private user-only site, which is the default configuration for the FTP server; A private FTP server allows users on the system only to be able to connect via FTP and access their files. You can place access controls on these users so that certain users can be explicitly denied or granted access.

An other kind of FTP server is anonymous. An anonymous FTP server allows anyone on the network to connect to it and transfer files without having an account. Due to the potential security risk involved with this setup, you should take precautions to allow access only to certain directories on the system.

The configuration I will cover here is an FTP server that allow FTP to semi-secure areas of a Unix filesystem (chroot'd Guest FTP access). This configuration allow users to have access to, for instance, Web site directories without allowing them to get into higher levels.

These installation instructions assume

Commands are Unix-compatible.
Installations were tested on RedHat Linux 6.1 Server.
All steps in the installation will happen in superuser account "root".
wu-ftpd version number is 2.6.0

Packages

Wu-ftpd Homepage: http://www.wu-ftpd.org/
You must be sure to download: wu-ftpd-2.6.0.tar.gz

Compilation

Decompress the tarball (tar.gz).

```
[root@deep]# cp wu-ftpd-version.tar.gz /var/tmp
[root@deep]# cd /var/tmp
[root@deep]# tar xzpf wu-ftpd-version.tar.gz
```

Compile and Optimize

Cd into the new Wu-ftpd directory and type the following on your terminal:

Edit the **ftpcount.c** file (vi +241 src/ftpcount.c) and change the line:

```
#if defined (LINUX)
```

To read:

```
#if defined (LINUX_BUT_NOT_REDHAT_6_0)
```

Edit the **pathnames.h.in** file (vi +42 src/pathnames.h.in) and change the line:

```
#define _PATH_EXECPATH  "/bin/ftp-exec"
```

To read:

```
#define _PATH_EXECPATH  "/usr/bin/ftp-exec"
```

We change the "/bin" directory of "ftp-exec" to be under "/usr/bin".

```
CC="egcs" \
CFLAGS="-O9 -funroll-loops -ffast-math -malign-double -mcpu=pentiumpro -march=pentiumpro -fomit-
frame-pointer -fno-exceptions" \
./configure \
--prefix=/usr \
--sysconfdir=/etc \
--localstatedir=/var \
--disable-dnsretry \
--enable-quota \
--enable-pam \
--disable-daemon \
--disable-newlines \
--disable-virtual \
```

```
--disable-plsm \
--disable-pasvip \
--disable-anonymous \
--enable-ls \
--enable-numericuid
```

This tells Wu-ftpd to set itself up for this particular hardware setup with:

- Don't retry failed DNS lookups.
- Add QUOTA support (if your OS supports it).
- Add PAM support.
- Don't allow running as standalone daemon.
- Suppress some extra blank lines.
- Don't support virtual servers.
- Disable PID lock sleep messages (for busy sites).
- Don't require same IP for passive connections.
- Don't allow anonymous ftp access.
- Use the internal ls (EXPERIMENTAL).
- Internal ls displays UID instead of username (faster).

```
make
make install
install -m 755 util/xferstats /usr/sbin
touch /var/log/xferlog
chmod 600 /var/log/xferlog
cd /usr/sbin
ln -sf in.ftpd /usr/sbin/wu.ftpd
ln -sf in.ftpd /usr/sbin/in.wuftpd
strip /usr/bin/ftpcount
strip /usr/bin/ftpwho
strip /usr/sbin/in.ftpd
strip /usr/sbin/ftpshut
strip /usr/sbin/ckconfig
strip /usr/sbin/ftprestart
```

The above commands “make” and “make install” would configure the software to ensure your system has the necessary functionality and libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

The “install -m” will install the program xferstats used to see static about transferred files and the “touch” command will create the log file for xferstats under “/var/log” directory. The “chmod” will change the mode of “xferlog” file to be readable and writable only by the super-user “root”. After, we create symbolic links for “in.ftpd” binary and finally strip all binaries related to Wu-ftpd to reduce their sizes for better performance.

Cleanup after work

```
[root@deep]# cd /var/tmp
[root@deep]# rm -rf wu-ftpd-version/ wu-ftpd-version.tar.gz
```

The “rm” command will remove all the source files we have used to compile and install Wu-ftpd. It will also remove the Wu-ftpd compressed archive from the “/var/tmp” directory.

Setup an FTP user account for each user without shells

First of all, create a new user for this purpose, this user will be the user allowing to connect to your ftp server. This has to be separate from a regular user account with unlimited access, because of how the "chroot" environment works. Chroot makes it appear from the user's

perspective as if the level of the filesystem you've placed them in is the top level of the file system.

Step 1
Use the following command to create users in the "/etc/passwd" file. This step must be doing for each additional new users you allow to access your ftp server.

```
[root@deep]# mkdir /home/ftp
[root@deep]# useradd -d /home/ftp/ftpadmin/ -s /dev/null ftpadmin > /dev/null 2>&1
[root@deep]# passwd ftpadmin
Changing password for user ftpadmin
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

Step 2
Edit the "/etc/shells" file and add a no existent shell name like "null" for example. This fake shell will limit access on the system for ftp users.

```
[root@deep]# vi /etc/shells

/bin/bash
/bin/sh
/bin/ash
/bin/bsh
/bin/tcsh
/bin/csh
/dev/null ← This is our added no existent shell
```

Step 3
Now, edit your "/etc/passwd" file and add manually the "/./" to divides "/home/ftp" directory to "/ftpadmin" directory where the user "ftpadmin" should be automatically chdir. This step must be doing for each ftp users you add to your passwd file.

Edit the **passwd** file (vi /etc/passwd) and add:

```
ftpadmin:x:502:502::/home/ftp/ftpadmin/:/dev/null
To read:
ftpadmin:x:502:502::/home/ftp/./ftpadmin/:/dev/null
                              ^^
```

The account is "ftpadmin", but you'll notice the path to the home directory is a bit odd. The first part "/home/ftp/" indicates the filesystem that should be considered their new root. The dot divides that from the directory they should be automatically chdir (change directory'd) into, "/ftpadmin/".

The "/dev/null" part disables their login as a regular user. With this modification, user "ftpadmin" have now a fake shell instead of a real shell resulting to a limited access on the system.

Setup a chroot user environment

What you're essentially doing is creating a skeleton root file system with enough components necessary (binaries, password files, etc.) to allow Unix to do a chroot when the user logs in. Make note that if you use the "**--enable-ls**" option during compilation like seen above, the "/home/ftp/bin", and "/home/ftp/lib" directories are not required since this new option allows WU-FTP to use its own "ls" function. I still continue to demonstrate the old method to people that

prefer to copy “/bin/ls” to the chroot’d ftp directory (“/home/ftp/bin”) and create the appropriated library related to “ls”.

Step 1:
First create all the necessary chrooted environment directories:

[root@deep]# **mkdir /home/ftp/dev**
[root@deep]# **mkdir /home/ftp/etc**
[root@deep]# **mkdir /home/ftp/bin** (require only if you are not using the “--enable-ls” option)
[root@deep]# **mkdir /home/ftp/lib** (require only if you are not using the “--enable-ls” option)

Step 2
Change the new directories permission to 0511:

[root@deep]# **chmod 0511 /home/ftp/dev**
[root@deep]# **chmod 0511 /home/ftp/etc**
[root@deep]# **chmod 0511 /home/ftp/bin** (require only if you are not using the “--enable-ls” option)
[root@deep]# **chmod 0511 /home/ftp/lib** (require only if you are not using the “--enable-ls” option)

The “chmod” command will make our chrooted “dev”, “etc”, “bin”, and “lib” directories readable and executable by the super-user “root” and executable by the user-group and all users.

Step 3
Copy the "/bin/ls" binary to "/home/ftp/bin" directory and change the permission of “ls” to 0111. (You don't want users to be able to modify the binaries):

[root@deep]# **cp /bin/ls /home/ftp/bin** (require only if you are not using the “--enable-ls” option)
[root@deep]# **chmod 0111 /bin/ls /home/ftp/bin/ls** (require only if you are not using the “--enable-ls” option)

Step 4:
Find the shared library dependencies of “ls” program:

[root@deep]# **ldd /bin/ls** (require only if you are not using the “--enable-ls” option)

libc.so.6 => /lib/libc.so.6 (0x00125000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x00110000)

Copy the shared libraries identified above to your new “lib” directory under “/home/ftp” directory:

[root@deep]# **cp /lib/libc.so.6 /home/ftp/lib/** (require only if you are not using the “--enable-ls” option)
[root@deep]# **cp /lib/ld-linux.so.2 /home/ftp/lib/** (require only if you are not using the “--enable-ls” option)

These library is needed to make “ls” to work.

NOTE: The steps 3 and 4 above are requiring only if you want to use the “ls” binary program of Linux instead of the “--enable-ls” option that use the new internal ls capability of Wu-ftpd.

Step 5
Create your “/home/ftp/dev/null” file:

[root@deep]# **mknod /home/ftp/dev/null c 1 3**
[root@deep]# **chmod 666 /home/ftp/dev/null**

Step 6:
Copy the “group” and “passwd” files in “/home/ftp/etc” directory. This should not be the same as your true ones.

[root@deep]# **cp /etc/passwd /home/ftp/etc/**
[root@deep]# **cp /etc/group /home/ftp/etc/**

Edit the **passwd** file (vi /home/ftp/etc/passwd) and delete all entries except the user “root” and all your allowed FTP users. It is very important that the passwd file in the chroot environment should have entries like:

root:x:0:0:root:**/:/dev/null**
ftpadmin:x:502:502::**/ftpadmin/:/dev/null**

Edit the **group** file (vi /home/ftp/etc/group) and delete all entries except the user “root” and all your allowed FTP users. The group file should correspond to your normal group file:

root:x:0:root
ftpadmin:x:502:

Configurations

All software we describe in our book "Linuxsos.pdf" will have a specific directory and subdirectory in a tar compressed archive named “floppy.tgz” containing file configurations for the specific program. If you get this archive file, you wouldn’t be obliged to reproduce the different configuration files bellow manually or cut and past them to create your configuration files. Whatever your decide to copy manually or get the files made to your convenience from the archive compressed files, it will be to your responsibility to modify, adjust for your needs and place the files related to Wu-FTP software to their appropriated places on your server machine, like show bellow. The server configuration files archive is located at the following Internet address: http://pages.infinit.net/lotus1/doc/opti/floppy.tgz

- To run an FTP server, the following files are require and must be create or copied to their appropriated directories on your server.

  Copy the **ftpaccess** file in the “/etc/” directory.
  Copy the **ftpusers** file in the “/etc/” directory.
  Copy the **ftphosts** file in the “/etc/” directory.
  Copy the **ftpgroups** file in the “/etc/” directory.
  Copy the **ftpconversion** file in the “/etc/” directory.
  Copy the **ftp** file in the “/etc/pam.d/” directory.
  Copy the **ftpd** file in the “/etc/logrotate.d/” directory.

You can obtain configuration files listed bellow on our floppy.tgz archive. Copy the following files from the decompressed floppy.tgz archive to their appropriated places or copy and paste them directly from this book to the concerned file.

Configuration of the “/etc/ftpaccess” file

The “/etc/ftpaccess” file is used to configure the operation of “ftpd”. This file is the primary means of controlling who, how many users access your server and other important security configurations. Each line in the file controls either defines an attribute or sets its value. With non-anonymous chroot access, you want to create a set of “guestgroups”, each of which corresponds directly to entries in the "/home/ftp/etc/group" file.

Create the **ftpaccess** file (touch /etc/ftpaccess) and add in this file:

```
class   openarch   guest   208.164.186.*

limit openarch 20 MoTuWeTh,Fr0000-1800 /home/ftp/.too_many.msg
email admin@openarch.com

loginfails 3

readme  README*    login
readme  README*    cwd=*

message /home/ftp/.welcome.msg  login
message .message                cwd=*

compress        yes             all
tar             yes             all
chmod           yes             guest
delete          yes             guest
overwrite       yes             guest
rename          yes             guest

log commands real,guest
log transfers real,guest inbound,outbound

guestgroup ftpadmin
guestgroup webmaster

# We don't want users being able to upload into these areas.
upload /home/ftp/* /    no
upload /home/ftp/* /etc no
upload /home/ftp/* /dev no

# We'll prevent downloads with noretrieve.
noretrieve /home/ftp/etc
noretrieve /home/ftp/dev

log security real,guest

guest-root /home/ftp ftpadmin webmaster
restricted-uid ftpadmin webmaster
restricted-gid ftpadmin webmaster

greeting terse
Keepalive yes
noretrieve .notar
```

Now, change its default permission to be 600:
[root@deep]# **chmod 600 /etc/ftpaccess**

This tells ftpaccess file to set itself up for this particular configuration setup with:

class
The class command defines a class of users who can access your FTP server. You can define as many classes as you want. Each class line comes in the form

class <classname> <typelist> <addrglob>

where <classname> is the name of the class you are defining, <typelist> is the type of user you are allowing into the class, and <addrglob> is the range of IP addresses allowed access to that class.

The <typelist> is a comma-delimited list in which each entry has one of three values: **anonymous, guest, or real**. Anonymous users are, of course, any users who connect to the server as user anonymous or ftp and want to access only publicly available files. Guest users are special because they do not have accounts on the system per se, but they do have special access to key parts of the guest group. Real users must have accounts on the FTP server and are authenticated accordingly.

<addrglob> takes the form of a regular expression where * implies all sites.

The line **class openarch guest 208.164.186.***

Allows only guest users with accounts on the FTP server access to their accounts via FTP if they are coming from the address 208.164.186.*

limit
The limit command allows you to control the number of users who log in to the system via FTP by class and time of day. The format of the limit command is:

limit <class> <n> <times> <message_file>

Where <class> is the class to limit, <n> is the maximum number of people allowed in that class, <times> is the time during which the limit is in effect, and <message_file> is the file that should be displayed to the client when the maximum limit is reached.

The format of the <times> parameter is in the form of a comma-delimited string, where each option is for a separate day. Sunday through Saturday take the form Su, Mo, Tu, We, Th, Fr, and Sa, respectively, and all the weekdays can be referenced as Wk. Time should be kept in military format without a colon separating the hours and minutes. The dash character specifies a range.

For example, to limit the class **openarch** to 20 users from Monday through Thursday, all day, and Friday from midnight to 6:00 p.m., you would use the following limit line:

limit openarch 20 MoTuWeTh,Fr0000-1800 /home/ftp/.too_many.msg

In this case, if the limit is hit, the contents of the file “/home/ftp/.too_many.msg” are displayed to the connecting user.

loginfails
The loginfails command allows you to set the number of failed login attempts clients can make before disconnecting them. You can set it by using the command:

loginfails <n>

Where <n> is the number of attempts. For example, the following line disconnects a user from the FTP server after three failed attempts:

loginfails 3

readme
The readme command allows you to specify the conditions under which clients are notified that a certain file in their current directory was last modified. This command can take the form:

readme <path> <when>

Where <path> is the name of the file to alert the clients about (for example README), <when> is the condition under which to display the message.

The <when> parameter should take one of two forms: either LOGIN or CWD=<dir>. If it is LOGIN, the message is displayed upon a successful login. If the parameter is set to CWD=<dir>, then the message id displayed when client enter the <dir> directory.

Remember that when you're specifying a path for anonymous users, the file must be relative to the anonymous FTP directory.

message
The message command allows you to set up special messages to be sent to the clients when they either log in or change into a certain directory. You can specify multiple messages. The format of this command is:

message <path> <when>

Where <path> is the full pathname to the file to be displayed, .<when> is similar to the <when> in the readme command.

Remember that when messages are triggered by an anonymous user, the message path needs to be relative to the anonymous FTP directory.

An example is:
message /home/ftp/.welcome.msg LOGIN

compress, tar, chmod, delete, overwrite, rename
If you don't specify the following directives, they default to yes for everybody. What you're doing here is giving permission for these guestgroups to chmod, delete, overwrite, and rename files, and you're allowing everybody to use compress and tar.

For example:

compress	**yes**	**all**
tar	**yes**	**all**
chmod	**yes**	**guest**
delete	**yes**	**guest**
overwrite	**yes**	**guest**
rename	**yes**	**guest**

log commands
Enables logging of individual commands by users for security purpose. The format of this command is:

log commands <typelist>

Where <typelist> is a comma-separated list specifying which kinds of users should be logged (**anonymous, guest, real**).

For example, to log all **real** and **guest** individual commands, you would use the following:

log commands real,guest

The resulting logs are stored in the “/var/log/message” file.

log transfers
You probably should log all transfers for security purposes. The format of this command is:

log transfers <typelist> <directions>

Where <typelist> is a comma-separated list specifying which kinds of users should be logged (**anonymous, guest, real**), and <directions> is a comma-separated list specifying which direction the transfers must take in order to be logged. The two directions you can choose to log are inbound and outbound.

For example, to log all **real** and **guest** transfers that are both inbound and outbound, you would use the following:

log transfers real,guest inbound,outbound

The resulting logs are stored in the “/var/log/xferlog” file.

guestgroup
This command allow you to specify all of your guestgroups, one per line. The "/home/ftp/etc/group" file has entries for each of these groups, each of which has just one member.

For example:
guestgroup ftpadmin
guestgroup webmaster

log security
Enables logging of violations of security rules (noretrieve, .notar, ...) for real, guest and/or anonymous users.

log security <typelist>

Where <typelist> is a comma-separated list of any of the keywords "**anonymous**", "**guest**" and "**real**". If the "real" keyword is included, logging will be done for users using FTP to access real accounts, and if the "anonymous" keyword is included logging will done for users using anonymous FTP. The "guest" keyword matches guest access accounts.

For example:
log security real,guest

restricted-uid, restricted-gid and guest-root
These clauses control whether or not **real** or **guest** users will be allowed access to areas on the FTP site outside their home directories. An example of the use of these clauses shows their intended use.

guest-root <root-dir>
restricted-uid <uid-range>
restricted-gid <gid-range>

For example:
guest-root /home/ftp ftpadmin webmaster

restricted-uid ftpadmin webmaster
restricted-gid adminftp webmaster

<root-dir> specified the chroot() path for users. Multiple uid ranges may be given on the line. If a guest-root is chosen for the user, the user's home directory in the “<root-dir>/etc/passwd” file is used to determine the initial directory and their home directory in the system-wide “/etc/passwd” is not used. While both “ftpadmin” and “webmaster” are chroot'd to “/home/ftp”, they cannot access each other's files because they are restricted to their home directories.

greeting full|brief|terse
Allows you to control how much information is given out before the remote user logs in.

greeting full|brief|terse

‘greeting full' is the default and shows the hostname and daemon version. 'greeting brief' whose shows the hostname. 'greeting terse' simply says "FTP server ready".

For example:
greeting terse

keepalive <yes|no>
Set the TCP SO_KEEPALIVE option for data sockets. This can be used to control network disconnect. Yes: set it. No: use system default (usually off). It is a good idea to set this.

Keepalive yes

Configuration of the “/etc/ftphosts” file

The “/etc/ftphosts” file establishes rule on a per-user basis defining whether users are allowed to log in from certain hosts or whether users are denied access when they try to log in from other hosts.

Create the **ftphosts** file (touch /etc/ftphosts) and add for example in the file:

```
# Example host access file
#
# Everything after a '#' is treated as comment,
# empty lines are ignored
allow ftpadmin 208.164.186.1 208.164.186.2 208.164.186.4
deny ftpadmin 208.164.186.5
```

Now, change its default permission to be 600:
[root@deep]# **chmod 600 /etc/ftphosts**

Each line in the file can be one of two commands:

allow <username> <addrglob>
or
deny <username> <addrglob>

Where the allow command allows the user specified in <username> to connect via FTP from the explicitly listed addresses in <addrglob>. You can list multiple addresses.

The deny command explicitly denies the specified user <username> from the sites listed in <addrglob>. You can list multiple sites.

Configuration of the "/etc/ftpusers" file

The "/etc/ftpusers" file specifies those users that are NOT allowed to connect to your ftpd.

Create the **ftpusers** file (touch /etc/ftpusers) and add in this file:

```
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
```

Now, change its default permission to be 600:
[root@deep]# **chmod 600 /etc/ftpusers**

Configuration of the "/etc/ftpconversions" file

The "/etc/ftpconversions" is a file who contain instruction that permit to compress files on demand before the transfer.

Create the **ftpconversions** file (touch /etc/ftpconversions) and add in this file:

```
:.Z:  :  :/bin/compress -d -c %s:T_REG|T_ASCII:O_UNCOMPRESS:UNCOMPRESS
:   : :.Z:/bin/compress -c %s:T_REG:O_COMPRESS:COMPRESS
:.gz: :  :/bin/gzip -cd %s:T_REG|T_ASCII:O_UNCOMPRESS:GUNZIP
:   : :.gz:/bin/gzip -9 -c %s:T_REG:O_COMPRESS:GZIP
:   : :.tar:/bin/tar -c -f - %s:T_REG|T_DIR:O_TAR:TAR
:   : :.tar.Z:/bin/tar -c -Z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+COMPRESS
:   : :.tar.gz:/bin/tar -c -z -f - %s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+GZIP
:   : :.crc:/bin/cksum %s:T_REG::CKSUM
:   : :.md5:/bin/md5sum %s:T_REG::MD5SUM
```

Now, change its default permission to be 600:
[root@deep]# **chmod 600 /etc/ftpconversions**

Configuration of the "/etc/pam.d/ftp" file

Configure your "/etc/pam.d/ftp" file to use pam authentication.

Create the **ftp** file (touch /etc/pam.d/ftp) and add:

```
#%PAM-1.0
auth      required    /lib/security/pam_listfile.so item=user sense=deny file=/etc/ftpusers onerr=succeed
auth      required    /lib/security/pam_pwdb.so shadow nullok
auth      required    /lib/security/pam_shells.so
account   required    /lib/security/pam_pwdb.so
session   required    /lib/security/pam_pwdb.so
```

Configuration of the "/etc/logrotate.d/ftpd" file

Configure your "/etc/logrotate.d/ftpd" file to rotate each week your log files automatically.

Create the **ftpd** file (touch /etc/logrotate.d/ftpd) and add:

```
/var/log/xferlog {
   # ftpd doesn't handle SIGHUP properly
   nocompress
}
```

Configure ftpd to use tcp-wrappers inetd super server

Tcp-wrappers take cares to start and stop ftpd server. Upon execution, inetd reads its configuration information from a configuration file which, by default, is "/etc/inetd.conf". There must be an entry for each field of the configuration file, with entries for each field separated by a tab or a space.

Edit the **inetd.conf** file (vi /etc/inetd.conf) and add or verify the existence of the line:

ftp stream tcp nowait root /usr/sbin/tcpd in.ftpd -l -a

NOTE: Update your "inetd.conf" file by sending a SIGHUP signal (killall -HUP inetd) after adding the line.

[root@deep /root]# **killall -HUP inetd**

Edit the **hosts.allow** file (vi /etc/hosts.allow) and add for example the line:

in.ftpd: 192.168.1.4 win.openarch.com

Which mean client IP "192.168.1.4" with host name "win.openarch.com" is allowed to ftp on the server.

FTP Administrative Tools

ftpwho

ftpwho displays all active users on the system connected through FTP. The output of the command is in the format of the "/bin/ps" command. The format of this command is:

<pid> <time> <tty> <connection details>

Where <pid> is the process ID of the FTP daemon handling the transfer; <time> indicates when the user have been connected to the FTP server; <tty> is always a question mark (?) because the connection is coming from FTP, not Telnet; and finally, <connection details> tells where the connection is coming from, who is the user, and what that user's current function is.

The following is an example of output from ftpwho.

```
[root@deep]# ftpwho
Service class openarch:
 5443 ?       S      0:00 ftpd: win.openarch.com: admin: IDLE
  -  1 users ( 20 maximum)
```

Here, you can see that one user is logged in (20 users are allowed to connect) and this user has the username admin who claims to be win.openarch.com.

ftpcount
ftpcount, which is a simplified version of ftpwho, shows only the total count of users logged in to the system and the maximum number of users allowed. A sample output from ftpcount shows the following:

```
[root@deep]# ftpcount
Service class openarch        -  1 users ( 20 maximum)
```

Securing FTP

ENSURE that you have set up a file "/etc/ftpusers" which specifies those users that are NOT allowed to connect to your ftpd. This should include, as a MINIMUM, the entries: root, bin, daemon, adm, lp, sync, shutdown, halt, mail, news, uucp, operator, games, nobody and ALL vendor supplied accounts.

To disable anonymous ftp, remove the user **ftp** from your password file. Verify that **anonftp-version.i386.rpm** package is not installed on your system whit the command:

```
[root@deep]# rpm -q anonftp.
```

The upload command

By default, the WU-FTPD server will grant upload privileges to all guest users. The example users are chroot'd to "/home/ftp" and cannot access any area of the filesystem outside that directory structure. What we're interested in, then, is simply protecting the areas in the chroot directory structure we want to keep the users out of.

In our installation, there will be "bin", "etc", "dev", and "lib", subdirectories in the "/home/ftp" directory. We don't want users being able to upload into these areas. In case something happens to the permissions on them, you should deny upload privileges in your "/etc/ftpaccess" file. In our case, we'll say the following:

```
upload /home/ftp/* /    no
upload /home/ftp/* /etc no
upload /home/ftp/* /dev no
upload /home/ftp/* /bin no (require only if you are not using the "--enable-ls" option)
upload /home/ftp/* /lib no (require only if you are not using the "--enable-ls" option)
```

The noretrieve command

It is also a good idea to prevent downloads of those subdirectories in the "/home/ftp" directory with the command "noretrieve" in your "/etc/ftpaccess" file.

```
noretrieve /home/ftp/etc
noretrieve /home/ftp/dev
noretrieve /home/ftp/bin (require only if you are not using the "--enable-ls" option)
noretrieve /home/ftp/lib (require only if you are not using the "--enable-ls" option)
```

The special file ".notar"

Whether you allow on-the-fly tar'ing of directories or not, you should make sure an end-run cannot be made using tar in all area where the upload is not permit. To do so, create the special file '.notar' in each directory and in the FTP directory.

```
[root@deep]# touch /home/ftp/.notar
[root@deep]# chmod 0 /home/ftp/.notar
[root@deep]# touch /home/ftp/etc/.notar
```

[root@deep]# **chmod 0 /home/ftp/etc/.notar**
[root@deep]# **touch /home/ftp/dev/.notar**
[root@deep]# **chmod 0 /home/ftp/dev/.notar**
[root@deep]# **touch /home/ftp/bin/.notar** (require only if you are not using the “--enable-ls” option)
[root@deep]# **chmod 0 /home/ftp/bin/.notar** (require only if you are not using the “--enable-ls” option)
[root@deep]# **touch /home/ftp/lib/.notar** (require only if you are not using the “--enable-ls” option)
[root@deep]# **chmod 0 /home/ftp/lib/.notar** (require only if you are not using the “--enable-ls” option)

The zero-length “.notar” file can confuse some web clients and FTP proxies, so let's mark it irretrievable to solve the problem. Add the following lines to your “/etc/ftpaccess” file.

noretrieve .notar

Installed files

> /etc/ftphosts
> /etc/ftpusers
> /etc/ftpaccess
> /etc/pam.d/ftp
> /etc/ftpconversions
> /etc/ftpgroups
> /etc/logrotate.d/ftpd
> /usr/bin/ftpcount
> /usr/bin/ftpwho
> /usr/man/man1/ftpcount.1
> /usr/man/man1/ftpwho.1
> /usr/man/man5/ftpaccess.5
> /usr/man/man5/ftphosts.5
> /usr/man/man5/ftpconversions.5
> /usr/man/man5/xferlog.5
> /usr/man/man8/ftpd.8
> /usr/man/man8/ftpshut.8
> /usr/man/man8/ftprestart.8
> /usr/sbin/in.ftpd
> /usr/sbin/ftpshut
> /usr/sbin/ckconfig
> /usr/sbin/ftprestart
> /usr/sbin/xferstats
> /usr/sbin/wu.ftpd
> /usr/sbin/in.wuftpd
> /var/log/xferlog

Part VI Backup-Related reference

In this Part

Backup and Restore Procedures

Chapter 11 Backup and restore procedures
In this Chapter

Server Backup Procedures
Server restore Procedures

Backup and Restore Procedures

Server Backup Procedures

A secure and reliable server is closely related to performing regular backups. Regular backups should be considered one of a top priorities. Failures probably will occur. They may be caused by hardware failure, power outages, or human error. If you are hosting users on your system, you will most certainly be requested to restore an inadvertently deleted files.

If you perform regular backups, you will hopefully reduce the possibility of such loss files. The safest method of doing backups is to record them in a location separate from your Linux system like network, tape, removable drive, writable CD-ROM, ect.

There are a variety of methods of performing backups with Linux. These include command-line tools included with every Linux distribution, such as “dd”, “dump”, “cpio”, as well as “tar”. Also available is text-based utility, such as “Amanda”, which is designed to add a more user-friendly interface to the backup and restore procedures. Finally, commercial backup utility is also available, such as “BRU”.

Obviously, the procedures for performing a backup and restore will differ depending on your choice of a backup solution. For this reason, we will discuss methods for performing backups with the tool we use most: “tar”, which is a command-line backup tool largely portable across *nix systems.

The following command will perform a backup of your entire Linux server onto the ``/archive/'' file system, with the exception of the ``/proc/'' pseudo-filesystem, any mounted file systems in "/mnt/", the ``/archive/'' file system (no sense backing up our backup sets!), as well as Squid's rather large cache files (which are, a waste of backup media and unnecessary to back up):

Caution: When backing up your file systems, do not include the "/proc" pseudo-filesystem! The files in "/proc" are not actually files but are simply file-like links which describe and point to kernel data structures, also do not include the “/mnt” and “/archive” files.

- To perform a backup of your entire system, use the command:
  [root@deep]# **cd /**
  [root@deep]# **tar -zcvpf /archive/full-backup-`date '+%d-%B-%Y'`.tar.gz **
  **--directory / --exclude=proc --exclude=mnt --exclude=archive **
  --exclude=cache .

z specifies the backup data will be compressed with “gzip” program.
c specifies an archive file is begin created.
v display a list of files as they get.
p preserve permissions; file protection information will be “remembered”.
f states that the very next argument will be the name of the archive file or device being written.

Notice how a filename which contains the current date is derived, simply by enclosing the “date” command between two back-quote characters. A common naming convention is to add a “tar” suffix for non-compressed archives, and a “tar.gz” suffix for compressed ones.

The ”--directory” option tells tar to first switch to the following directory path (the “/” directory in this example) prior to starting the backup. The “--exclude” options tell tar not to bother backing up the specified directories or files. Finally, the “.” character at the end of the command tells tar that it should back up everything in the current directory.

Another example, this time writing only the specified file systems onto a SCSI tape drive follows:

- To perform a backup of only the specified files onto a SCSI tape drive, use the command:
  [root@deep]# **cd /**
  [root@deep]# **tar -cvpf /dev/nst0 --label="Backup set created on `date '+%d-%B-%Y'`." **
  --directory / etc home chroot

In the above command, notice that the "z" (compress) option is not used for this example, but we strongly recommend writing compressed data to tape. Because the tape drive is a character device, it is not possible to specify an actual file name. Therefore, the file name used as an argument to tar is simply the name of the device, "/dev/nst0", the first tape device on the SCSI bus.

NOTE: The "/dev/nst0" device does not rewind after the backup set is written; therefore it is possible to write multiple sets on one tape. You may also refer to the device as ``/dev/st0", in which case the tape is automatically rewound after the backup set is written.

Since we aren't able to specify a filename for the backup set, the "--label" option can be used to write some information about the backup set into the archive file itself. Finally, only the files contained in the "/etc/", "/home/", and "/chroot", are written to the tape.

When working with tapes, you can use the following commands to rewind, and eject your tape:

[root@deep]# **mt -f /dev/nst0 rewind**
[root@deep]# **mt -f /dev/nst0 offline**

Server Restore Procedures

The one thing that is more important than performing regular backups is having them available when it comes time to recover an important file! As discussed above, the procedures for performing a restore will differ depending on your choice of a backup solution. In this section, we will discuss methods for restoring files which have been backed up with “tar”.

The following command will restore all files from the “full-backup-14-November-1999.tar.gz” archive, which is an example backup of our Linux server as created in the example commands shown above.

- To restore a backup of your entire system, use the command:
  [root@deep]# **cd /**
  [root@deep]# **tar -zxvpf /archive/full-backup-14-November-1999.tar.gz**

The above command extracts all files contained in the compressed archive, preserving original file ownership and permissions.

z option tells tar that the archive is compressed with gzip.
x option stands for extract.
v display a list of files as they get.
p preserve permissions; file protection information will be “remembered”.
f states that the very next argument will be the name of the archive file or device.

If you do not need to restore all files contained in the archive, you can specify one or more files that you wish to restore, as in the following example:

- To specify one or more files that you wish to restore, use the command:
  [root@deep]# **cd /**
  [root@deep]# **tar -zxvpf /archive/full-backup-09-October-1999.tar.gz **

```
etc/passwd usr/sbin/chpasswd
```

The above command restores the “etc/passwd” and “usr/sbin/chpasswd” files from the archive.

If you are trying to restore only one or a few files from your archive, you will not be successful unless you specify the file name and directory path exactly as stored in the archive. The following example might help out:

```
[root@deep]# cd /
[root@deep]# tar -ztvpf /archive/full-backup-09-October-1999.tar.gz \
             | grep -i chpasswd
```

In the above example, all files contained in the archive are listed by file name. The resulting output is then piped to the “grep” command using grep's “i” option to ignore mixed case, displaying any files containing the word “chpasswd” in either the directory path or file name. Once you determine the exact file name you wish to restore, you can then specify it for extraction in a regular tar command expression like above.

NOTE: When creating an archive file, “tar” will strip leading “/” (slash) characters from file path names. This means that restore files may not end up in the same locations they were backed up from. Therefore, to solve the problem the solution is to change to the “/” root directory before make all backup or all restore. Another solution is to restore the desired files under a different directory, and then compare, move, or update the files to their original locations afterward.

Caution: If you have a files on your system set with the immutable bit, using the “chattr” command, these files will not be remembered with the immutable bit from your restore backup. You must to reset it immutable with the command “chattr” after the backup is completed.

Further documentation

For more details, there is man page you can read:

tar (1) - The GNU version of the tar archiving utility

Alternatives to tar backup

AMANDA

AMANDA (Advanced Maryland Automatic Network Disk Archiver) is a backup system that allows the administrator of a LAN to set up a single master backup server to back up multiple hosts to a single large capacity tape drive. AMANDA users native dump and/or GNU tar facilities and can back up a large number of workstations running multiple versions of Unix. Recent versions can also use SAMBA to back up Microsoft Windows 95/NT hosts.

Packages

AMANDA Homepage: http://www.cs.umd.edu/projects/amanda/

BRU

BRU ™ is the Unix Backup and Restore Utility designed for maximum reliability and flexibility. It makes Unix backups easy, fast and safe. BRU is much more than just a replacement for tar and cpio, it's really a backup system with a multitude of options.

Packages

BRU Homepage: http://www.bru.com/

Part VII Appendixes
In this part

Appendix A
In this part

Tweaks, Tips and Administration tasks

Tweaks, Tips and Administration tasks

Some of the tips in this presentation are specific to Linux systems. Most are applicable to UNIX system in general.

1.0 The “du” utility

You can use the "**du**" utility to help determine required space. For example, to determine the requirements of the "/var/log/" and "/home/" directory trees, type:

```
[root@deep]# du -sh /var/log /home
1.0M    /var/log
66M     /home
```

Bear in mind that the above command will report the actual size of your data. Nice and neat and in human readable form. Now that you know “/home” is using 66M you can "cd /home" then "du -sh *" to locate where the largest files are.

NOTE: You can add to your crontab this command so that every day you get emailed a disk space list so you can monitor it without logging in constantly.

1.1 Find out the route that the packets sent from your machine to a remote host

If you want to find out the route that the packets sent from your machine to a remote host, simply issue the command: (traceroute hostname) where hostname is the name or ip address of the host that you want to trace.

```
[root@deep]# traceroute www.redhat.com
```

1.2 Display the number of time you Web pages has been acceded:

To display the number of times your page has been accessed use this command:

```
[root@deep]# grep "GET / HTTP" /var/log/httpd/access_log | wc -l
```

1.3 Shut down most services altogether

As root, you can shut down most services with the following command:

```
[root@deep]# killall httpd smbd nmbd sendmail named
```

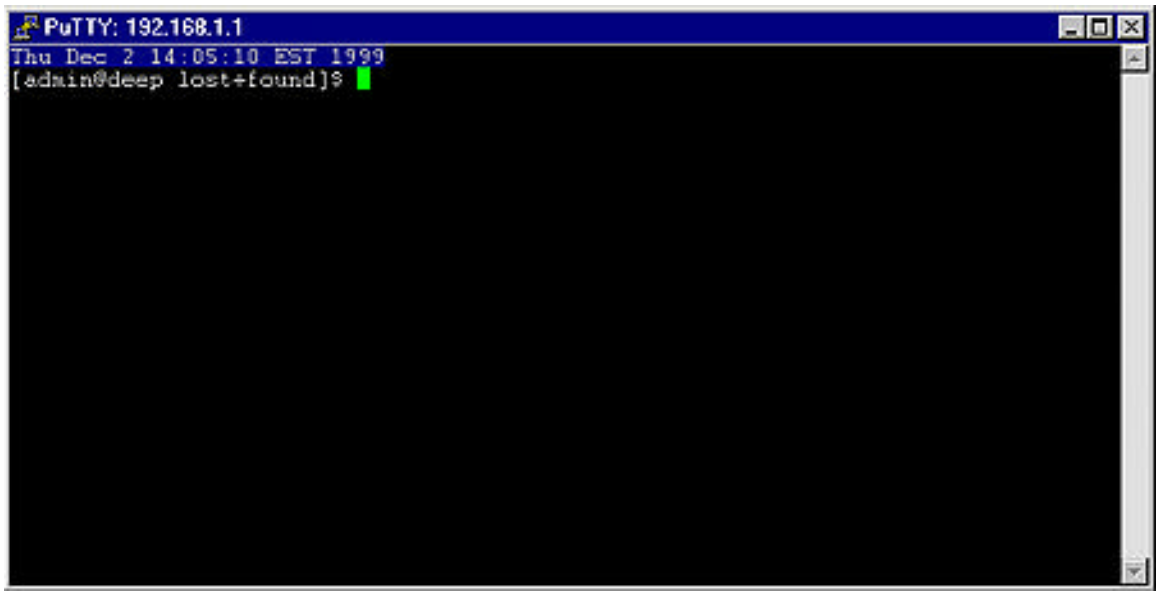
The above command will shut down the Apache server, Samba services, e-mail server, and DNS server.

1.4 Want a clock on the top of your terminal for all user?

Edit the profile file (vi /etc/profile) and add the following line:

```
PROMPT_COMMAND='echo -ne
"\0337\033[2;999r\033[1;1H\033[00;44m\033[K"`date`"\033[00m\0338"'
```

The result will look like:



1.5 Do you have "lsof" installed?
If not, install it and execute "lsof -i". This should list which ports you have open on your machine. Lsof is a great tool, however a much easier way to check for open tcp ports is 'netstat -an |grep LISTEN'. The nice thing about lsof is that it will tell you which processes are listening on a given port. Netstat only tells you which ports are open, not what process is attached.

[root@deep]# **lsof -i**

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
Inetd	344	root	4u	IPv4	327		TCP	*:ssh (LISTEN)
sendmail	389	root	4u	IPv4	387		TCP	*:smtp (LISTEN)
smbd	450	root	5u	IPv4	452		TCP	deep.openarch.com:netbios-ssn (LISTEN)
nmbd	461	root	5u	IPv4	463		UDP	*:netbios-ns
nmbd	461	root	6u	IPv4	465		UDP	*:netbios-dgm
nmbd	461	root	8u	IPv4	468		UDP	deep.openarch.com:netbios-ns
nmbd	461	root	9u	IPv4	470		UDP	deep.openarch.com:netbios-dgm
named	2599	root	4u	IPv4	3095		UDP	*:32771
named	2599	root	20u	IPv4	3091		UDP	localhost.localdomain:domain
named	2599	root	21u	IPv4	3092		TCP	localhost.localdomain:domain (LISTEN)
named	2599	root	22u	IPv4	3093		UDP	deep.openarch.com:domain
named	2599	root	23u	IPv4	3094		TCP	deep.openarch.com:domain (LISTEN)

1.6 Run commands on remote via ssh protocol without logging in
The ssh command can also be used to run commands on remote systems without logging in. The output of the command is displayed and control returns to the local system. Here is an example which will display all the users logged in on the remote system.

[admin@deep]$ **ssh mail.openarch.com who**
admin@mail.openarch.com's password:

root	tty1	Dec	2	14:45
admin	tty2	Dec	2	14:45
wahib	pts/0	Dec	2	11:38

1.7 Filename Completion
Tab filename completion allows you to type in portions of a filename or program, and then press [TAB], and it will complete the filename for you. If there's more than one file or program that starts

with what you already typed in, it will beep, and then when you press [TAB] again it lists all the files that start with what you initially type. Play around with it a little bit to see what its limits are, how it can work for what you do, etc.

Appendix B
In this part

Obtaining Requests for Comments (RFCs)

Obtaining Requests for Comments (RFCs)

Requests for Comments (RFCs) is an ongoing set of documents issued by the Internet Engineering Task Force (IETF) at the Network Information Center (NIC) that presents new protocols and establishes standards for the Internet protocol suite. Each such document defines an aspect of protocol regarding the Internet. We have listed bellow all the RFCs that pertain to this book. RFCs are available from the following site: http://www.cis.ohio-state.edu/rfc/

RFC706
On the Junk Mail Problem.

RFC733
Standard for the Format of ARPA Network Text Messages.

RFC768
User Datagram Protocol (UDP).

RFC791
Internet Protocol (IP).

RFC792
Internet Control Message Protocol (ICMP).

RFC793
Transmission Control Protocol (TCP).

RFC805
Computer Mail Meting Notes.

RFC821
Simple Mail Transfert Protocol (SMTP).

RFC822
Standard for the Format of ARPA Internet Text Massages.

RFC934
Proposed Standard for Message Encapsulation.

RFC950
IP Subnet Extention.

RFC959
File Transfer Protocol (FTP).

RFC976
UUCP Mail Interchange Format Standard.

RFC1034
Domain Names: Concepts and Facilities.

RFC1036
Standard for Interchange of USENET Message.

RFC1058

Routing Information Protocol (RIP).

RFC1112
Internet Group Multicast Protocol (IGMP).

RFC1122
Requirement for Internet Host—Communication Layers.

RFC1123
Requirements for Internet Host—Application and Support.

RFC1137
Mapping Between Full RFC 822 and RFC 822 with Restricted Encoding.

RFC1153
Digest Message Format.

RFC1155
Structure of Management Information (SMI).

RFC1157
Simple Network Management Protocol (SNMP).

RFC1176
Interactive Mail Access Protocol: Version 2.

RFC1310
The Internet Standards Process.

RFC1319
MD2 Message-Digest Algorithm.

RFC1320
MD4 Message-Digest Algorithm.

RFC1321
MD5 Message-Digest Algorithm.

RFC1343
User Agent Configuration Mechanism for Multimedia Mail Format Information.

RFC1344
Implications of MIME for Internet Mail Gateways.

RFC1345
Character Mnemonics and Character Sets.

RFC1421
Privacy Enhancement for Internet Electronic Mail: Part I—Message Encipherment and authentication Procedures.

RFC1422
Privacy Enhancement for Internet Electronic Mail: Part II—Certificate-based key Management.

RFC1423

Privacy Enhancement for Internet Electronic Mail: Part III—Algorithms, modes, and identifiers [Draft].

RFC1428
Transmition of Internet Mail from Just-Send-8 to 8bit-SMTP/MIME.

RFC1492
An Access Control Protocol, Sometimes Called TACACS.

RFC1495
Mapping Between X.400(1988)/ISO 10021 and RFC 822.

RFC1496
X.400 1988 to 1984 Downgrading.

RFC1505
Encoding Header Field for Internet Messages.

RFC1510
The Kerberos Network Authentication Service (V5).

RFC1519
Classless Inter-Domain Routing (CIDR) Assignment and Aggregation Strategy.

RFC1521
MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies (MIME).

RFC1522
Representation of Non-ASCII Text in Internet Message Headers.

RFC1566
Mail Monitoring MIB.

RFC1579
Firewall-Friendly FTP.

RFC1583
Open Shortest Path First Routing V2 (OSPF2).

RFC1625
WAIS over Z39.50-1988.

RFC1631
The IP Network Address Translator (NAT).

RFC1652
SMTP Service Extentions for 8bit-MIMEtransport.

RFC1661
Point-to-Point Protocol (PPP).

RFC1711
Classifications in E-mail Routing.

RFC1725

Post Office Protocol, Version 3 (POP)3.

RFC1738
Uniform Resource Locators (URL).

RFC1739
A Primer on Internet and TCP/IP Tools.

RFC1796
Not All RFCs are Standards.

RFC1830
SMTP Services Extentions for Transmission of Large and Binary MIME Messages.

RFC1844
Multimedia E-mail (MIME) User Agent checklist.

RFC1845
SMTP Service Extension for Checkpoint/Restart.

RFC1846
SMTP 521 Reply Code.

RFC1854
SMTP Service Extention for command pipelining.

RFC1855
Netiquette Guidelines.

RFC1864
The content-MD5 Header.

RFC1866
Hypertext Markup Language - 2.0.

RFC1869
SMTP Service Extensions.

RFC1870
SMTP Service Extension for Message Size Declaration.

RFC1872
The MIME Multipart/Related Content-type.

RFC1873
Message/External-Body Content-ID Access-type.

RFC1883
Internet Protocol, Version 6 (Ipv6) Specification.

RFC1884
IP Version 6 Addressing Atchitecture.

RFC1886
DNS Extentions to support IP version 6.

RFC1891
SMTP Service Extension for Delivery Status Notifications.

RFC1892
The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages.

RFC1893
Enhanced Mail System Status Codes.

RFC1894
An Extensible Message Format for Delivery Status Notifications.

RFC1918
Address Allocation for Private Internets.

RFC1928
SOCKS Protocol Version 5.

RFC1929
Username/Password Authentication for SOCKS V5.

RFC1961
GSS-API Authentication Method for SOCKS Version 5.

RFC2003
IP Encapsulation within IP.

RFC2028
The Organizations Involved in the IETF Standards Process.

RFC2060
Internet Message Access Protocol – Version 4rev1 (IMAP4).

RFC2104
HMAC: Keyed-Hashing for Message Authentication.

RFC2138
Remote Authentication Dial In User Service (RADIUS).

RFC2200
Internet Official Protocol Standards.

RFC2305
A Simple Mode of Facsimile Using Internet Mail.

RFC2313
PKCS 1: RSA Encryption Version 1-5.

RFC2314
PKCS 10: Certification Request Syntax Version 1-5.

RFC2315
PKCS 7: Cryptographic Message Syntax Version 1-5.