Ubuntu Server Guide

Ubuntu Documentation Project <*ubuntu-doc@lists.ubuntu.com*>

Ubuntu Server Guide

by Ubuntu Documentation Project <*ubuntu-doc@lists.ubuntu.com*>

Abstract

An introduction to installing and configuring server applications on Ubuntu.

Credits and License

The following Ubuntu Documentation Team authors maintain this document:

- Bhuvaneswaran Arumugam

The Ubuntu Server Guide is also based on the contributions of:

- Robert Stoffers
- Brian Shumate
- Rocco Stanzione

Disclaimer

Table of Contents

List of Tables

About This Guide

1. Conventions

The following notes will be used throughout the book:

A note presents interesting, sometimes technical, pieces of information related to the surrounding discussion.

A tip offers advice or an easier way of doing something.

A caution alerts the reader to potential problems and helps avoid them.

A warning advises the reader of a hazard that may arise in a given scenario.

Cross-reference conventions for print will be displayed as follows:

- Links to other documents or websites will look like *this* [http://www.ubuntu.com].

PDF, HTML, and XHTML versions of this document will use hyperlinks to handle cross-referencing.

Type conventions will be displayed as follows:

- File names or paths to directories will be shown in `monospace`.
- Commands that you type at a Terminal command prompt will be shown as:

  ```
  command to type
  ```

- Options that you click, select, or choose in a user interface will be shown in `monospace` type.

Menu selections, mouse actions, and keyboard short-cuts:

- A sequence of menu selections will be displayed as follows: File → Open
- Mouse actions shall assume a right-handed mouse configuration. The terms “click” and “double-click” refer to using the left mouse button. The term “right-click” refers to using the right mouse button. The term “middle-click” refers to using the middle mouse button, pressing down on the scroll wheel, or pressing both the left and right buttons simultaneously, based on the design of your mouse.
- Keyboard shortcut combinations will be displayed as follows: **Ctrl**-**N** .Where the conventions for “Control”, “Shift,” and “Alternate” keys will be **Ctrl**, **Shift**, and **Alt**, respectively, and shall mean the first key is to be held down while pressing the second key.

2. Contributing and Feedback

This book is developed by the *Ubuntu Documentation Team* [https://wiki.ubuntu.com/DocumentationTeam]. *You* can contribute to this document by sending ideas or comments to the Ubuntu Documentation Team mailing list. Information about the team, its mailing lists, projects, etc. can be found on the *Ubuntu Documentation Team Website* [https://wiki.ubuntu.com/DocumentationTeam].

If you see a problem with this document, or would like to make a suggestion, you can simply file a bug report at the *Ubuntu Bugtracker* [https://launchpad.net/products/ubuntu-doc/+bugs]. Your help is vital to the success of our documentation!

Many thanks,

-Your Ubuntu Documentation Team

Chapter 1. Introduction

Welcome to the *Ubuntu Server Guide*!

The *Ubuntu Server Guide* contains information on how to install and configure various server applications on your Ubuntu system to fit your needs. It is a step-by-step, task-oriented guide for configuring and customizing your system. This manual discusses many intermediate topics such as the following:

- Network Configuration
- Apache2 Configuration
- Databases
- Windows Networking

This manual is divided into the following main categories:

- Installation
- Package Management
- Networking
- Windows Networking

This guide assumes you have a basic understanding of your Ubuntu system. If you need detailed help installing Ubuntu, refer to the Ubuntu Installation Guide.

HTML and PDF versions of the manual are available online at *the Ubuntu Documentation website* [http://help.ubuntu.com].

You can buy this guide in book form from *our Lulu store* [http://www.lulu.com/ubuntu-doc]. You will only pay for the price of printing and postage.

Chapter 2. Installation

This chapter provides a quick overview of installing Ubuntu 6.06.1 LTS Server Edition. For more detailed instructions, please refer to the Ubuntu Installation Guide.

1. Preparing to Install

This section explains various aspects to consider before starting the installation.

1.1. System Requirements

Ubuntu 6.06.1 LTS Server Edition supports three (3) major architectures: Intel x86, AMD64, and PowerPC. The table below lists recommended hardware specifications. Depending on your needs, you might manage with less than this. However, most users risk being frustrated if they ignore these suggestions.

Table 2.1. Recommended Minimum Requirements

Install Type	RAM	Hard Drive Sp
Server	64 megabytes	500 megabytes

The default profile for the Ubuntu 6.06.1 LTS Server Edition is shown below. Once again, the size of the installation will greatly depend on the services you install during setup. For most administrators, the default services are suitable for general server use.

Server

This is a small server profile, which provides a common base for all sorts of server applications. It's minimal and designed to have the desired services added on top, such as file/print services, web hosting, email hosting, etc. For these services at least 500MB of disk space would suffice, but consider adding more space depending on the services you'd like to host with your server.

Remember that these sizes don't include all the other materials which are usually to be found, such as user files, mail, logs, and data. It is always best to be generous when considering the space for your own files and data.

1.2. Backing Up

- Before you start, make sure to back up every file that is now on your system. If this is the first time a non-native operating system has been installed on your computer, it's quite likely you will need to re-partition your disk to make room for Ubuntu. Any time you partition your disk, you should be prepared to lose everything on the disk should you make a mistake or something goes wrong during partitioning such as power loss to the system. The programs used in installation are quite reliable, and most have seen years of use, but they also perform destructive actions, and one mistake in use can result in loss of your valuable data.

  If you are creating a multi-boot system, make sure that you have the distribution media of any other present operating systems on hand. Especially if you repartition your boot drive, you might find that you have to reinstall your operating system's boot loader, or in many cases the whole operating system itself and all files on the affected partitions.

2. Installing from CD

Insert your installation CD into your CD-ROM drive and reboot the computer. The installation system is started immediately when booting from the CD-ROM. Once initialized, your first screen will appear.

At this point, read the text on the screen. You may want to read the help screen provided by the installation system. To do this, press F1.

To perform a default server installation, select “Install to the hard disk” and press **Enter**. The installation process will be started. Simply follow the on-screen instructions, and your Ubuntu system will be installed.

Alternatively, to install a LAMP server (Linux, Apache, MySQL, PHP/Perl/Python), select “Install a LAMP server”, and follow the instructions.

Chapter 3. Package Management

Ubuntu features a comprehensive package management system for the installation, upgrade, configuration, and removal of software. In addition to providing access to an organized base of over 17,000 software packages for your Ubuntu computer, the package management facilities also feature dependency resolution capabilities and software update checking.

Several tools are available for interacting with Ubuntu's package management system, from simple command-line utilities which may be easily automated by system administrators, to a simple graphical interface which is easy to use by those new to Ubuntu.

1. Introduction

Ubuntu's package management system is derived from the same system used by the Debian GNU/Linux distribution. The package files contain all of the necessary files, meta-data, and instructions to implement a particular functionality or software application on your Ubuntu computer.

Debian package files typically have the extension '.deb', and typically exist in *repositories* which are collections of packages found on various media, such as CD-ROM discs, or online. Packages are normally of the pre-compiled binary format; thus installation is quick and requires no compiling of software.

Many complex packages use the concept of *dependencies*. Dependencies are additional packages required by the principal package in order to function properly. For example, the speech synthesis package Festival depends upon the package festvox-kalpc16k, which is a package supplying one of the voices used by the application. In order for Festival to function, all of the dependencies must be installed in conjunction with the principal Festival package. The software management tools in Ubuntu will do this automatically.

2. Apt-Get

The apt-get command is a powerful command-line tool used to work with Ubuntu's *Advanced Packaging Tool* (APT) performing such functions as installation of new software packages, upgrade of existing software packages, updating of the package list index, and even upgrading the entire Ubuntu system.

Being a simple command-line tool, apt-get has numerous advantages over other package management tools available in Ubuntu for server administrators. Some of these advantages include ease of use over simple terminal connections (SSH) and the ability to be used in system administration scripts, which can in turn be automated by the cron scheduling utility.

Some examples of popular uses for the apt-get utility:

- **Install a Package**: Installation of packages using the apt-get tool is quite simple. For example, to install the network scanner *nmap*, type the following:

  ```
  sudo apt-get install nmap
  ```

- **Remove a Package**: Removal of a package or packages is also a straightforward and simple process. To remove the nmap package installed in the previous example, type the following:

  ```
  sudo apt-get remove nmap
  ```

  

  Multiple Packages: You may specify multiple packages to be installed or removed, separated by spaces.

- **Update the Package Index**: The APT package index is essentially a database of available packages from the repositories defined in the `/etc/apt/sources.list` file. To update the local package index with the latest changes made in repositories, type the following:

  ```
  sudo apt-get update
  ```

- **Upgrade Packages**: Over time, updated versions of packages currently installed on your computer may become available from the package repositories (for example security updated). To upgrade your system, first update your package index as outlined above, and then type:

  ```
  sudo apt-get upgrade
  ```

  If a package needs to install or remove new dependencies when being upgraded, it will not be upgraded by the *upgrade* command. For such an upgrade, it is necessary to use the *dist-upgrade* command.

  Also, you may upgrade your entire Ubuntu system from one revision to another with dist-upgrade. For example, to upgrade from Ubuntu version 5.10 to version 6.06.1 LTS, you would first ensure the version 6.06.1 LTS repositories replace the existing 5.10 repositories in your computer's

`/etc/apt/sources.list`, then simply issue the apt-get update command as detailed above, and finally, perform the actual upgrade by typing:

```
sudo apt-get dist-upgrade
```

After a fairly considerable amount of time, your computer will be upgraded to the new revision. Typically, some post-upgrade steps would be required as detailed in the upgrade notes for the revision you are upgrading to.

Actions of the apt-get command, such as installation and removal of packages, are logged in the /var/log/dpkg.log log file.

For further information about the use of APT, read the comprehensive *Debian APT User Manual* [http://www.debian.org/doc/user-manuals#apt-howto] or type:

```
apt-get help
```

3. Aptitude

Aptitude is a menu-driven, text-based front-end to the *Advanced Packaging Tool* (APT) system. Many of the common package management functions, such as installation, removal, and upgrade, are performed in Aptitude with single-key commands, which are typically lowercase letters.

Aptitude is best suited to use in a non-graphical terminal environment to ensure proper functioning of the command keys. You may start Aptitude as a normal user with the following command at a terminal prompt:

```
sudo aptitude
```

When Aptitude starts, you will see a menu bar at the top of the screen and two panes below the menu bar. The top pane contains package categories, such as *New Packages* and *Not Installed Packages*. The bottom pane contains information related to the packages and package categories.

Using Aptitude for package management is relatively straightforward, and the user interface makes common tasks simple to perform. The following are examples of popular package management functions as performed in Aptitude:

- **Install Packages**: To install a package, locate the package via the Not Installed Packages package category, for example, by using the keyboard arrow keys and the **ENTER** key, and highlight the package you wish to install. After highlighting the package you wish to install, press the **+** key, and the package entry should turn *green*, indicating it has been marked for installation. Now press **g** to be presented with a summary of package actions. Press **g** again, and you will be prompted to become root to complete the installation. Press **ENTER** which will result in a Password: prompt. Enter your user password to become root. Finally, press **g** once more and you'll be prompted to download the package. Press **ENTER** on the *Continue* prompt, and downloading and installation of the package will commence.
- **Remove Packages**: To remove a package, locate the package via the Installed Packages package category, for example, by using the keyboard arrow keys and the **ENTER** key, and highlight the package you wish to remove. After highlighting the package you wish to install, press the **-** key, and the package entry should turn *pink*, indicating it has been marked for removal. Now press **g** to be presented with a summary of package actions. Press **g** again, and you will be prompted to become root to complete the installation. Press **ENTER** which will result in a Password: prompt. Enter your user password to become root. Finally, press **g** once more, and you'll be prompted to download the package. Press **ENTER** on the *Continue* prompt, and removal of the package will commence.
- **Update Package Index**: To update the package index, simply press the **u** key and you will be prompted to become root to complete the installation. Press **ENTER** which will result in a Password: prompt. Enter your user password to become root. Updating of the package index will commence. Press **ENTER** on the OK prompt when the download dialog is presented to complete the process.

- **Upgrade Packages**: To upgrade packages, perform the update of the package index as detailed above, and then press the **U** key to mark all upgradeable packages. Now press **g** whereby you'll be presented with a summary of package actions. Press **g** again, and you will be prompted to become root to complete the installation. Press **ENTER** which will result in a Password: prompt. Enter your user password to become root. Finally, press **g** once more, and you'll be prompted to download the package. Press **ENTER** on the *Continue* prompt, and upgrade of the packages will commence.

The first column of information displayed in the package list in the top pane, when actually viewing packages lists the current state of the package, and uses the following key to describe the state of the package:

- **i**: Installed package.
- **c**: Package not installed, but package configuration remains on system
- **p**: Purged from system
- **v**: Virtual package
- **B**: Broken package
- **u**: Unpacked files, but package not yet configured
- **C**: Half-configured- Configuration failed and requires fix
- **H**: Half-installed- Removal failed and requires fix

To exit Aptitude, simply press the **q** key and confirm you wish to exit. Many other functions are available from the Aptitude menu by pressing the **F10** key.

4. Configuration

Configuration of the *Advanced Packaging Tool* (APT) system repositories is stored in the /etc/apt/sources.list configuration file. An example of this file is referenced here, along with information on adding or removing repository references from the file.

Here [../sample/sources.list] is a simple example of a typical `/etc/apt/sources.list` file.

You may edit the file to enable repositories or disable them. For example, to disable the requirement of inserting the Ubuntu CD-ROM whenever package operations occur, simply comment out the appropriate line for the CD-ROM, which appears at the top of the file:

```
# no more prompting for CD-ROM please
# deb cdrom:[Ubuntu 6.06 _Dapper Drake_ - Release i386 (20060329.1)]/ dapper main restricted
```

5. Extra Repositories

In addition to the officially supported package repositories available for Ubuntu, there exist additional community-maintained repositories which add thousands more potential packages for installation. Two of these additional repositories are most popular, and are the *Universe* and *Multiverse* repositories. These repositories are not officially supported by Ubuntu, which is why they are not enabled by default, but they generally provide packages which are safe for use with your Ubuntu computer.

Packages in the Multiverse repository often have licensing issues that prevent them from being distributed with a free operating system, and they may be illegal in your locality.

Be advised that neither the *Universe* or *Multiverse* repositories contain officially supported packages. In particular, there may not be security updates for these packages.

Many other package sources are available, sometimes even offering only one package, as in the case of package sources provided by the developer of a single application. You should always be very careful and cautious when using non-standard package sources, however. Research the source and packages carefully before performing any installation, as some package sources and their packages could render your system unstable or non-functional in some respects.

To enable the *Universe* and *Multiverse* repositories, edit the `/etc/apt/sources.list` file and uncomment the appropriate lines:

```
# We want Multiverse and Universe repositories, please

deb http://archive.ubuntu.com/ubuntu dapper universe multiverse
deb-src http://archive.ubuntu.com/ubuntu dapper universe multiverse
```

5.1. References

Adding Repositories Howto (Ubuntu Wiki) [https://wiki.ubuntu.com/AddingRepositoriesHowto]

Chapter 4. Networking

Networks consist of two or more devices, such as computer systems, printers, and related equipment which are connected by either physical cabling wireless links for the purpose of sharing and distributing information among the connected devices.

This section of the Ubuntu Server Guide provides general and specific information pertaining to networking, including an overview of network concepts and detailed discussion of popular network protocols and server applications.

1. Network Configuration

Ubuntu ships with a number of graphical utilities to configure your network devices. This document is geared toward server administrators and will focus on managing your network on the command line.

1.1. Ethernet

Most ethernet configuration is centralized in a single file, `/etc/network/interfaces`. If you have no ethernet devices, only the loopback interface will appear in this file, and it will look something like this:

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback
address 127.0.0.1
netmask 255.0.0.0
```

If you have only one ethernet device, eth0, and it gets its configuration from a DHCP server, and it should come up automatically at boot, only two additional lines are required:

```
auto eth0
iface eth0 inet dhcp
```

The first line specifies that the eth0 device should come up automatically when you boot. The second line means that interface (“iface”) eth0 should have an IPv4 address space (replace “inet” with “inet6” for an IPv6 device) and that it should get its configuration automatically from DHCP. Assuming your network and DHCP server are properly configured, this machine's network should need no further configuration to operate properly. The DHCP server will provide the default gateway (implemented via the route command), the device's IP address (implemented via the ifconfig command), and and DNS servers used on the network (implemented in the `/etc/resolv.conf` file.)

To configure your ethernet device with a static IP address and custom configuration, some more information will be required. Suppose you want to assign the IP address 192.168.0.2 to the device eth1, with the typical netmask of 255.255.255.0. Your default gateway's IP address is 192.168.0.1. You would enter something like this into `/etc/network/interfaces`:

```
iface eth1 inet static
 address 192.168.0.2
 netmask 255.255.255.0
 gateway 192.168.0.1
```

In this case, you will need to specify your DNS servers manually in `/etc/resolv.conf`, which should look something like this:

```
search mydomain.com
nameserver 192.168.0.1
```

```
nameserver 4.2.2.2
```

The *search* directive will append mydomain.com to hostname queries in an attempt to resolve names to your network. For example, if your network's domain is mydomain.com and you try to ping the host “mybox”, the DNS query will be modified to “mybox.mydomain.com” for resolution. The *nameserver* directives specifiy DNS servers to be used to resolve hostnames to IP addresses. If you use your own nameserver, enter it here. Otherwise, ask your Internet Service Provider for the primary and secondary DNS servers to use, and enter them into `/etc/resolv.conf` as shown above.

Many more configurations are possible, including dialup PPP interfaces, IPv6 networking, VPN devices, etc. Refer to man 5 interfaces for more information and supported options. Remember that `/etc/network/interfaces` is used by the ifup/ifdown scripts as a higher level configuration scheme than may be used in some other Linux distributions, and that the traditional, lower level utilities such as ifconfig, route, and dhclient are still available to you for ad hoc configurations.

1.2. Managing DNS Entries

This section explains how to configure the nameserver to use when resolving IP address to hostnames and vice versa. It does not explain how to configure the system as a name server.

To manage DNS entries, you can add, edit, or remove DNS names from the `/etc/resolv.conf` file. A *sample file* [../sample/resolv.conf] is given below:

```
search com
nameserver 204.11.126.131
nameserver 64.125.134.133
nameserver 64.125.134.132
nameserver 208.185.179.218
```

The search key specifies the string which will be appended to an incomplete hostname. Here, we have mentioned it as com. So, when we run: **ping ubuntu** it would be interpreted as **ping ubuntu.com**.

The nameserver key specifies the nameserver IP address. It will be used to resolve the given IP address or hostname. This file can have multiple nameserver entries. The nameservers will be used by the network query in the same order.

If the DNS server names are retrieved dynamically from DHCP or PPPOE (retrieved from your ISP), do not add nameserver entries in this file. It will be updated automatically.

1.3. Managing Hosts

To manage hosts, you can add, edit, or remove hosts from `/etc/hosts` file. The file contains IP addresses and their corresponding hostnames. When your system tries to resolve a hostname to an IP address or determine the hostname for an IP address, it refers to the `/etc/hosts` file before using the name servers. If the IP address is listed in the `/etc/hosts` file, the name servers are not used. This behavior can be modified by editing `/etc/nsswitch.conf` at your peril.

If your network contains computers whose IP addresses are not listed in DNS, it is recommended that you add them to the `/etc/hosts` file.

2. TCP/IP

The Transmission Control Protocol and Internet Protocol (TCP/IP) is a standard set of protocols developed in the late 1970s by the Defense Advanced Research Projects Agency (DARPA) as a means of communication between different types of computers and computer networks. TCP/IP is the driving force of the Internet, and thus it is the most popular set of network protocols on Earth.

2.1. TCP/IP Introduction

The two protocol components of TCP/IP deal with different aspects of computer networking. *Internet Protocol*, the "IP" of TCP/IP is a connectionless protocol which deals only with network packet routing using the *IP datagram* as the basic unit of networking information. The IP datagram consists of a header followed by a message. The *Transmission Control Protocol* is the "TCP" of TCP/IP and enables network hosts to establish connections which may be used to exchange data streams. TCP also guarantees that the data between connections is delivered and that it arrives at one network host in the same order as sent from another network host.

2.2. TCP/IP Configuration

The TCP/IP protocol configuration consists of several elements which must be set by editing the appropriate configuration files, or deploying solutions such as the Dynamic Host Configuration Protocol (DHCP) server which in turn, can be configured to provide the proper TCP/IP configuration settings to network clients automatically. These configuration values must be set correctly in order to facilitate the proper network operation of your Ubuntu system.

The common configuration elements of TCP/IP and their purposes are as follows:

- **IP address** The IP address is a unique identifying string expressed as four decimal numbers ranging from zero (0) to two-hundred and fifty-five (255), separated by periods, with each of the four numbers representing eight (8) bits of the address for a total length of thirty-two (32) bits for the whole address. This format is called *dotted quad notation*.
- **Netmask** The Subnet Mask (or simply, *netmask*) is a local bit mask, or set of flags which separate the portions of an IP address significant to the network from the bits significant to the *subnetwork*. For example, in a Class C network, the standard netmask is 255.255.255.0 which masks the first three bytes of the IP address and allows the last byte of the IP address to remain available for specifying hosts on the subnetwork.
- **Network Address** The Network Address represents the bytes comprising the network portion of an IP address. For example, the host 12.128.1.2 in a Class A network would use 12.0.0.0 as the network address, which uses twelve (12) to represent the first byte of the IP address, (the network part) and zeroes (0) in all of the remaining three bytes to represent the potential host values. Network hosts using the very common private and non-routable IP addresses such as 192.168.1.100 would in turn use a Network Address of 192.168.1.0, which specifies the first three bytes of the Class C 192.168.1 network and a zero (0) for all the possible hosts on the network.

- **Broadcast Address** The Broadcast Address is an IP address which allows network data to be sent simultaneously to all hosts on a given subnetwork rather than specifying a particular network host. The standard general broadcast address for IP networks is 255.255.255.255, but this broadcast address cannot be used to send a broadcast message to every host on the Internet because routers block it. A more appropriate broadcast address is set to match a specific subnetwork. For example, on the popular private Class C IP network, 192.168.1.0, the broadcast address should be configured as 192.168.1.255. Broadcast messages are typically produced by network protocols such as the Address Resolution Protocol (ARP) and the Routing Information Protocol (RIP).
- **Gateway Address** A Gateway Address is the IP address through which a particular network, or host on a network, may be reached. If one network host wishes to communicate with another network host, and that host is not located on the same network, then a *gateway* must be used. In many cases, the Gateway Address will be that of a router on the same network, which will in turn pass traffic on to other networks or hosts, such as Internet hosts. The value of the Gateway Address setting must be correct, or your system will not be able to reach any hosts beyond those on the same network.
- **Nameserver Address** Nameserver Addresses represent the IP addresses of Domain Name Service (DNS) systems, which resolve network hostnames into IP addresses. There are three levels of Nameserver Addresses, which may be specified in order of precedence: The *Primary* Nameserver, the *Secondary* Nameserver, and the *Tertiary* Nameserver. In order for your system to be able to resolve network hostnames into their corresponding IP addresses, you must specify valid Nameserver Addresses which you are authorized to use in your system's TCP/IP configuration. In many cases these addresses can and will be provided by your network service provider, but many free and publicly accessible Nameservers are available for use, such as the Level3 (Verizon) servers with IP addresses from 4.2.2.1 to 4.2.2.6.

The IP address, Netmask, Network Address, Broadcast Address, and Gateway Address are typically specified via the appropriate directives in the file `/etc/network/interfaces`. The Nameserver Addresses are typically specified via *nameserver* directives in the file `/etc/resolv.conf`. For more information, view the system manual page for `interfaces` or `resolv.conf` respectively, with the following commands typed at a terminal prompt:

Access the system manual page for `interfaces` with the following command:

```
man interfaces
```

Access the system manual page for `resolv.conf` with the following command:

```
man resolv.conf
```

2.3. IP Routing

IP routing is a means of specifying and discovering paths in a TCP/IP network along which network data may be sent. Routing uses a set of *routing tables* to direct the forwarding of network data packets from their source to the destination, often via many intermediary network nodes known as *routers*. IP Routing is the principal mode of path discovery on the Internet. There are two primary forms of IP Routing: *Static Routing* and *Dynamic Routing*.

Static routing involves manually adding IP routes to the system's routing table, and this is usually done by manipulating the routing table with the route command. Static routing enjoys many advantages over dynamic routing, such as simplicity of implementation on smaller networks, predictability (the routing table is always computed in advance, and thus the route is precisely the same each time it used), and low overhead on other routers and network links due to the lack of a dynamic routing protocol. However, static routing does present some disadvantages as well. For example, static routing is limited to small networks and does not scale well. Static routing also fails completely to adapt to network outages and failures along the route due to the fixed nature of the route.

Dynamic Routing depends on large networks with multiple possible IP routes from a source to a destination and makes use of special routing protocols, such as the Router Information Protocol (RIP), which handle the automatic adjustments in routing tables that make dynamic routing possible. Dynamic routing has several advantages over static routing, such as superior scalability and the ability to adapt to failures and outages along network routes. Additionally, there is less manual configuration of the routing tables, since routers learn from one another about their existence and available routes. This trait also eliminates the possibility of introducing mistakes in the routing tables via human error. Dynamic routing is not perfect, however, and presents disadvantages such as heightened complexity and additional network overhead from router communications, which does not immediately benefit the end users, but still consumes network bandwidth.

2.4. TCP and UDP

TCP is a connection-based protocol, offering error correction and guaranteed delivery of data via what is known as *flow control*. Flow control determines when the flow of a data stream needs to be stopped, and previously sent data packets should to be re-sent due to problems such as *collisions*, for example, thus ensuring complete and accurate delivery of the data. TCP is typically used in the exchange of important information such as database transactions.

The User Datagram Protocol (UDP), on the other hand, is a *connectionless* protocol which seldom deals with the transmission of important data because it lacks flow control or any other method to ensure reliable delivery of the data. UDP is commonly used in such applications as audio and video streaming, where it is considerably faster than TCP due to the lack of error correction and flow control, and where the loss of a few packets is not generally catastrophic.

2.5. ICMP

The Internet Control Messaging Protocol (ICMP) is an extension to the Internet Protocol (IP) as defined in the Request For Comments (RFC) #792 and supports network packets containing control, error, and informational messages. ICMP is used by such network applications as the ping utility, which can determine the availability of a network host or device. Examples of some error messages returned by ICMP which are useful to both network hosts and devices such as routers, include *Destination Unreachable* and *Time Exceeded*.

2.6. Daemons

Daemons are special system applications which typically execute continuously in the background and await requests for the functions they provide from other applications. Many daemons are network-centric; that is, a large number of daemons executing in the background on an Ubuntu system may provide network-related functionality. Some examples of such network daemons include the *Hyper Text Transport Protocol Daemon* (httpd), which provides web server functionality; the *Secure SHell Daemon* (sshd), which provides secure remote login shell and file transfer capabilities; and the *Internet Message Access Protocol Daemon* (imapd), which provides E-Mail services.

3. Firewall Configuration

The Linux kernel includes the *Netfilter* subsystem, which is used to manipulate or decide the fate of network traffic headed into or through your server. All modern Linux firewall solutions use this system for packet filtering.

3.1. Firewall Introduction

The kernel's packet filtering system would be of little use to administrators without a userspace interface to manage it. This is the purpose of iptables. When a packet reaches your server, it will be handed off to the Netfilter subsystem for acceptance, manipulation, or rejection based on the rules supplied to it from userspace via iptables. Thus, iptables is all you need to manage your firewall if you're familiar with it, but many frontends are available to simplify the task.

3.2. IP Masquerading

The purpose of IP Masquerading is to allow machines with private, non-routable IP addresses on your network to access the Internet through the machine doing the masquerading. Traffic from your private network destined for the Internet must be manipulated for replies to be routable back to the machine that made the request. To do this, the kernel must modify the *source* IP address of each packet so that replies will be routed back to it, rather than to the private IP address that made the request, which is impossible over the Internet. Linux uses *Connection Tracking* (conntrack) to keep track of which connections belong to which machines and reroute each return packet accordingly. Traffic leaving your private network is thus "masqueraded" as having originated from your Ubuntu gateway machine. This process is referred to in Microsoft documentation as Internet Connection Sharing.

This can be accomplished with a single iptables rule, which may differ slightly based on your network configuration:

```
sudo iptables -t nat -A POSTROUTING -s 192.168.0.0/16 -o ppp0 -j MASQUERADE
```

The above command assumes that your private address space is 192.168.0.0/16 and that your Internet-facing device is ppp0. The syntax is broken down as follows:

- -t nat -- the rule is to go into the nat table
- -A POSTROUTING -- the rule is to be appended (-A) to the POSTROUTING chain
- -s 192.168.0.0/16 -- the rule applies to traffic originating from the specified address space
- -o ppp0 -- the rule applies to traffic scheduled to be routed through the specified network device
- -j MASQUERADE -- traffic matching this rule is to "jump" (-j) to the MASQUERADE target to be manipulated as described above

Each chain in the filter table (the default table, and where most or all packet filtering occurs) has a default *policy* of ACCEPT, but if you are creating a firewall in addition to a gateway device, you may have set the policies to DROP or REJECT, in which case your masqueraded traffic needs to be allowed through the FORWARD chain for the above rule to work:

```
sudo iptables -A FORWARD -s 192.168.0.0/16 -o ppp0 -j ACCEPT
sudo iptables -A FORWARD -d 192.168.0.0/16 -m state --state ESTABLISHED,RELATED -i ppp0 -j ACCEPT
```

The above commands will allow all connections from your local network to the Internet and all traffic related to those connections to return to the machine that initiated them.

3.3. Tools

There are many tools available to help you construct a complete firewall without intimate knowledge of iptables. For the GUI-inclined, Firestarter is quite popular and easy to use, and fwbuilder is very powerful and will look familiar to an administrator who has used a commercial firewall utility such as Checkpoint FireWall-1. If you prefer a command-line tool with plain-text configuration files, Shorewall is a very powerful solution to help you configure an advanced firewall for any network. If your network is relatively simple, or if you don't have a network, ipkungfu should give you a working firewall "out of the box" with zero configuration, and will allow you to easily set up a more advanced firewall by editing simple, well-documented configuration files. Another interesting tool is fireflier, which is designed to be a desktop firewall application. It is made up of a server (fireflier-server) and your choice of GUI clients (GTK or QT), and behaves like many popular interactive firewall applications for Windows.

3.4. Logs

Firewall logs are essential for recognizing attacks, troubleshooting your firewall rules, and noticing unusual activity on your network. You must include logging rules in your firewall for them to be generated, though, and logging rules must come before any applicable terminating rule (a rule with a target that decides the fate of the packet, such as ACCEPT, DROP, or REJECT). For example:

```
sudo iptables -A INPUT -m state --state NEW -p tcp --dport 80 -j LOG --log-prefix "NEW_HTTP_CONN: "
```

A request on port 80 from the local machine, then, would generate a log in dmesg that looks like this:

```
[4304885.870000] NEW_HTTP_CONN: IN=lo OUT= MAC=00:00:00:00:00:00:00:00:00:00:00:00:08:00 SRC=127.0.
```

The above log will also appear in `/var/log/messages`, `/var/log/syslog`, and `/var/log/kern.log`. This behavior can be modified by editing `/etc/syslog.conf` appropriately or by installing and configuring ulogd and using the ULOG target instead of LOG. The ulogd daemon is a userspace server that listens for logging instructions from the kernel specifically for firewalls, and can log to any file you like, or even to a PostgreSQL or MySQL database. Making sense of your firewall logs can be simplified by using a log analyzing tool such as fwanalog, fwlogwatch, or lire.

4. OpenSSH Server

4.1. Introduction

This section of the Ubuntu Server Guide introduces a powerful collection of tools for the remote control of networked computers and transfer of data between networked computers, called *OpenSSH*. You will also learn about some of the configuration settings possible with the OpenSSH server application and how to change them on your Ubuntu system.

OpenSSH is a freely available version of the Secure Shell (SSH) protocol family of tools for remotely controlling a computer or transferring files between computers. Traditional tools used to accomplish these functions, such as telnet or rcp, are insecure and transmit the user's password in cleartext when used. OpenSSH provides a server daemon and client tools to facilitate secure, encrypted remote control and file transfer operations, effectively replacing the legacy tools.

The OpenSSH server component, sshd, listens continuously for client connections from any of the client tools. When a connection request occurs, sshd sets up the correct connection depending on the type of client tool connecting. For example, if the remote computer is connecting with the ssh client application, the OpenSSH server sets up a remote control session after authentication. If a remote user connects to an OpenSSH server with scp, the OpenSSH server daemon initiates a secure copy of files between the server and client after authentication. OpenSSH can use many authentication methods, including plain password, public key, and Kerberos tickets.

4.2. Installation

Installation of the OpenSSH client and server applications is simple. To install the OpenSSH client applications on your Ubuntu system, use this command at a terminal prompt:

```
sudo apt-get install openssh-client
```

To install the OpenSSH server application, and related support files, use this command at a terminal prompt:

```
sudo apt-get install openssh-server
```

4.3. Configuration

You may configure the default behavior of the OpenSSH server application, sshd, by editing the file `/etc/ssh/sshd_config`. For information about the configuration directives used in this file, you may view the appropriate manual page with the following command, issued at a terminal prompt:

```
man sshd_config
```

There are many directives in the sshd configuration file controlling such things as communications settings and authentication modes. The following are examples of configuration directives that can be changed by editing the `/etc/ssh/ssh_config` file.

Prior to editing the configuration file, you should make a copy of the original file and protect it from writing so you will have the original settings as a reference and to reuse as necessary.

Copy the `/etc/ssh/sshd_config` file and protect it from writing with the following commands, issued at a terminal prompt:

```
sudo cp /etc/ssh/sshd_config /etc/ssh/sshd_config.original
sudo chmod a-w /etc/ssh/sshd_config.original
```

The following are examples of configuration directives you may change:

- To set your OpenSSH to listen on TCP port 2222 instead of the default TCP port 22, change the Port directive as such:

  Port 2222

- To have sshd allow public key-based login credentials, simply add or modify the line:

  PubkeyAuthentication yes

  in the `/etc/ssh/sshd_config` file, or if already present, ensure the line is not commented out.

- To make your OpenSSH server display the contents of the `/etc/issue.net` file as a pre-login banner, simply add or modify the line:

  Banner /etc/issue.net

  in the `/etc/ssh/sshd_config` file.

After making changes to the `/etc/ssh/sshd_config` file, save the file, and restart the sshd server application to effect the changes using the following command at a terminal prompt:

```
sudo /etc/init.d/ssh restart
```

Many other configuration directives for sshd are available for changing the server application's behavior to fit your needs. Be advised, however, if your only method of access to a server is ssh, and you make a mistake in configuring sshd via the `/etc/ssh/sshd_config` file, you may find you are locked out of the server upon restarting it, or that the sshd server refuses to start due to an incorrect configuration directive, so be extra careful when editing this file on a remote server.

4.4. References

OpenSSH Website [http://www.openssh.org/]

Advanced OpenSSH Wiki Page [https://wiki.ubuntu.com/AdvancedOpenSSH]

5. FTP Server

File Transfer Protocol (FTP) is a TCP protocol for uploading and downloading files between computers. FTP works on a client/server model. The server component is called an *FTP daemon*. It continuously listens for FTP requests from remote clients. When a request is received, it manages the the login and sets up the connection. For the duration of the session it executes any of commands sent by the FTP client.

Access to an FTP server can be managed in two ways:

- Anonymous
- Authenticated

In the Anonymous mode, remote clients can access the FTP server by using the default user account called 'anonymous" or "ftp" and sending an email address as the password. In the Authenticated mode a user must have an account and a password. User access to the FTP server directories and files is dependent on the permissions defined for the account used at login. As a general rule, the FTP daemon will hide the root directory of the FTP server and change it to the FTP Home directory. This hides the rest of the file system from remote sessions.

5.1. vsftpd - FTP Server Installation

vsftpd is an FTP daemon available in Ubuntu. It is easy to install, set up, and maintain. To install vsftpd you can run the following command:

```
sudo apt-get install vsftpd
```

5.2. vsftpd - FTP Server Configuration

You can edit the vsftpd configuration file, `/etc/vsftpd.conf`, to change the default settings. By default only anonymous FTP is allowed. If you wish to disable this option, you should change the following line:

```
anonymous_enable=YES
```

to

```
anonymous_enable=NO
```

By default, local system users are not allowed to login to FTP server. To change this setting, you should uncomment the following line:

```
#local_enable=YES
```

By default, users are allowed to download files from FTP server. They are not allowed to upload files to FTP server. To change this setting, you should uncomment the following line:

```
#write_enable=YES
```

Similarly, by default, the anonymous users are not allowed to upload files to FTP server. To change this setting, you should uncomment the following line:

```
#anon_upload_enable=YES
```

The configuration file consists of many configuration parameters. The information about each parameter is available in the configuration file. Alternatively, you can refer to the man page, **man 5 vsftpd.conf** for details of each parameter.

Once you configure vsftpd you can start the daemon. You can run following command to run the vsftpd daemon:

```
sudo /etc/init.d/vsftpd start
```

Please note that the defaults in the configuration file are set as they are for security reasons. Each of the above changes makes the system a little less secure, so make them only if you need them.

6. Network File System (NFS)

NFS allows a system to share directories and files with others over a network. By using NFS, users and programs can access files on remote systems almost as if they were local files.

Some of the most notable benefits that NFS can provide are:

- Local workstations use less disk space because commonly used data can be stored on a single machine and still remain accessible to others over the network.
- There is no need for users to have separate home directories on every network machine. Home directories could be set up on the NFS server and made available throughout the network.
- Storage devices such as floppy disks, CDROM drives, and USB Thumb drives can be used by other machines on the network. This may reduce the number of removable media drives throughout the network.

6.1. Installation

At a terminal prompt enter the following command to install the NFS Server:

```
sudo apt-get install nfs-kernel-server
```

6.2. Configuration

You can configure the directories to be exported by adding them to the `/etc/exports` file. For example:

```
/ubuntu  *(ro,sync,no_root_squash)
/home    *(rw,sync,no_root_squash)
```

You can replace * with one of the hostname formats. Make the hostname declaration as specific as possible so unwanted systems cannot access the NFS mount.

To start the NFS server, you can run the following command at a terminal prompt:

```
sudo /etc/init.d/nfs-kernel-server start
```

6.3. NFS Client Configuration

Use the mount command to mount a shared NFS directory from another machine, by typing a command line similar to the following at a terminal prompt:

```
sudo mount example.hostname.com:/ubuntu /local/ubuntu
```

The mount point directory `/local/ubuntu` must exist. There should be no files or subdirectories in the `/local/ubuntu` directory.

An alternate way to mount an NFS share from another machine is to add a line to the `/etc/fstab` file. The line must state the hostname of the NFS server, the directory on the server being exported, and the directory on the local machine where the NFS share is to be mounted.

The general syntax for the line in `/etc/fstab` file is as follows:

```
example.hostname.com:/ubuntu /local/ubuntu nfs rsize=8192,wsize=8192,timeo=14,intr
```

6.4. References

Linux NFS faq [http://nfs.sourceforge.net/]

7. Dynamic Host Configuration Protocol (DHCP)

The Dynamic Host Configuration Protocol (DHCP) is a network service that enables host computers to be automatically assigned settings from a server as opposed to manually configuring each network host. Computers configured to be DHCP clients have no control over the settings they receive from the DHCP server, and the configuration is transparent to the computer's user.

The most common settings provided by a DHCP server to DHCP clients include:

- IP-Address and Netmask
- DNS
- WINS

However, a DHCP server can also supply configuration properties such as:

- Host Name
- Domain Name
- Default Gateway
- Time Server
- Print Server

The advantage of using DHCP is that changes to the network, for example a change in the address of the DNS server, need only be changed at the DHCP server, and all network hosts will be reconfigured the next time their DHCP clients poll the DHCP server. As an added advantage, it is also easier to integrate new computers into the network, as there is no need to check for the availability of an IP address. Conflicts in IP address allocation are also reduced.

A DHCP server can provide configuration settings using two methods:

MAC Address
: This method entails using DHCP to identify the unique hardware address of each network card connected to the network and then continually supplying a constant configuration each time the DHCP client makes a request to the DHCP server using that network device.

Address Pool
: This method entails defining a pool (sometimes also called a range or scope) of IP addresses from which DHCP clients are supplied their configuration properties dynamically and on a fist come first serve basis. When a DHCP client is no longer on the network for a specified period, the configuration is expired and released back to the address pool for use by other DHCP Clients.

Ubuntu is shipped with both DHCP server and client. The server is dhcpd (dynamic host configuration protocol daemon). The client provided with Ubuntu is dhclient and should be installed on all computers required to be automatically configured. Both programs are easy to install and configure and will be automatically started at system boot.

7.1. Installation

At a terminal prompt, enter the following command to install dhcpd:

```
sudo apt-get install dhcpd
```

You will see the following output, which explains what to do next:

```
Please note that if you are installing the DHCP server for the first
time you need to configure. Please stop (/etc/init.d/dhcp
stop) the DHCP server daemon, edit /etc/dhcpd.conf to suit your needs
and particular configuration, and restart the DHCP server daemon
(/etc/init.d/dhcp start).

You also need to edit /etc/default/dhcp to specify the interfaces dhcpd
should listen to. By default it listens to eth0.

NOTE: dhcpd's messages are being sent to syslog. Look there for
diagnostics messages.

Starting DHCP server: dhcpd failed to start - check syslog for diagnostics.
```

7.2. Configuration

The error message the installation ends with might be a little confusing, but the following steps will help you configure the service:

Most commonly, what you want to do is assign an IP address randomly. This can be done with settings as follows:

```
# Sample /etc/dhcpd.conf
# (add your comments here)
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option domain-name "mydomain.org";

subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.10 192.168.1.100;
range 192.168.1.150 192.168.1.200;
}
```

This will result in the DHCP server giving a client an IP address from the range 192.168.1.10-192.168.1.100 or 192.168.1.150-192.168.1.200. It will lease an IP address for 600 seconds if the client doesn't ask for a specific time frame. Otherwise the maximum (allowed) lease will be 7200 seconds. The server will also "advise" the client that it should use 255.255.255.0 as its subnet mask, 192.168.1.255 as its broadcast address, 192.168.1.254 as the router/gateway and 192.168.1.1 and 192.168.1.2 as its DNS servers.

If you need to specify a WINS server for your Windows clients, you will need to include the netbios-name-servers option, e.g.

```
option netbios-name-servers 192.168.1.1;
```

Dhcpd configuration settings are taken from the DHCP mini-HOWTO, which can be found *here* [http://www.tldp.org/HOWTO/DHCP/index.html].

7.3. References

DHCP FAQ [http://www.dhcp-handbook.com/dhcp_faq.html]

8. Domain Name Service (DNS)

Domain Name Service (DNS) is an Internet service that maps IP addresses and fully qualified domain names (FQDN) to one another. In this way, DNS alleviates the need to remember IP addresses. Computers that run DNS are called *name servers*. Ubuntu ships with BIND (Berkley Internet Naming Daemon), the most common program used for maintaining a name server on GNU/Linux.

8.1. Installation

At a terminal prompt, enter the following command to install dns:

```
sudo apt-get install bind
```

8.2. Configuration

The DNS configuration files are stored in the `/etc/bind` directory. The primary configuration file is `/etc/bind/named.conf`. The content of the default configuration file is shown below:

```
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind/README.Debian for information on the
// structure of BIND configuration files in Debian for BIND versions 8.2.1
// and later, *BEFORE* you customize this configuration file.
//

include "/etc/bind/named.conf.options";

// reduce log verbosity on issues outside our control
logging {
 category lame-servers { null; };
 category cname { null; };
};

// prime the server with knowledge of the root servers
zone "." {
        type hint;
        file "/etc/bind/db.root";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912

zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
```

```
        type master;
        file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};

// add local zone definitions here
include "/etc/bind/named.conf.local";
```

The include line specifies the filename which contains the DNS options. The directory line in the options file tells DNS where to look for files. All files BIND uses will be relative to this directory.

The file named `/etc/bind/db.root` describes the root name servers in the world. The servers change over time and must be maintained now and then.

The zone section defines a master server, and it is stored in a file mentioned against file tag. Every zone file contains 3 resource records (RRs): an SOA RR, an NS RR and a PTR RR. SOA is short of Start of Authority. The "@" is a special notation meaning the origin. NS is the Name Server RR. PTR is Domain Name Pointer. To start the DNS server, run the following command from a terminal prompt:

```
sudo /etc/init.d/bind start
```

You can refer to the documentation mentioned in the references section for details.

8.3. References

DNS HOWTO [http://www.tldp.org/HOWTO/DNS-HOWTO.html]

9. CUPS - Print Server

The primary mechanism for Ubuntu printing and print services is the **Common UNIX Printing System** (CUPS). This printing system is a freely available, portable printing layer which has become the new standard for printing in most GNU/Linux distributions.

CUPS manages print jobs and queues and provides network printing using the standard Internet Printing Protocol (IPP), while offering support for a very large range of printers, from dot-matrix to laser and many in between. CUPS also supports PostScript Printer Description (PPD) and auto-detection of network printers, and features a simple web-based configuration and administration tool.

9.1. Installation

To install CUPS on your Ubuntu computer, simply use sudo with the the apt-get command and give the packages to install as the first parameter. A complete CUPS install has many package dependencies, but they may all be specified on the same command line. Enter the following at a terminal prompt to install CUPS:

```
sudo apt-get install cupsys cupsys-client
```

Upon authenticating with your user password, the packages should be downloaded and installed without error. Upon the conclusion of installation, the CUPS server will be started automatically. For troubleshooting purposes, you can access CUPS server errors via the error log file at: `/var/log/cups/error_log`. If the error log does not show enough information to troubleshoot any problems you encounter, the verbosity of the CUPS log can be increased by changing the **LogLevel** directive in the configuration file (discussed below) to "debug" or even "debug2", which logs everything, from the default of "info". If you make this change, remember to change it back once you've solved your problem, to prevent the log file from becoming overly large.

9.2. Configuration

The Common UNIX Printing System server's behavior is configured through the directives contained in the file `/etc/cups/cupsd.conf`. The CUPS configuration file follows the same syntax as the primary configuration file for the Apache HTTP server, so users familiar with editing Apache's configuration file should feel at ease when editing the CUPS configuration file. Some examples of settings you may wish to change initially will be presented here.

Prior to editing the configuration file, you should make a copy of the original file and protect it from writing, so you will have the original settings as a reference, and to reuse as necessary.

Copy the `/etc/cups/cupsd.conf` file and protect it from writing with the following commands, issued at a terminal prompt:

```
sudo cp /etc/cups/cupsd.conf /etc/cups/cupsd.conf.original
sudo chmod a-w /etc/cups/cupsd.conf.original
```

- **ServerAdmin**: To configure the email address of the designated administrator of the CUPS server, simply edit the `/etc/cups/cupsd.conf` configuration file with your preferred text editor, and modify the *ServerAdmin* line accordingly. For example, if you are the Administrator for the CUPS server, and your e-mail address is 'bjoy@somebigco.com', then you would modify the ServerAdmin line to appear as such:

  ```
  ServerAdmin bjoy@somebigco.com
  ```

For more examples of configuration directives in the CUPS server configuration file, view the associated system manual page by entering the following command at a terminal prompt:

```
man cupsd.conf
```

Whenever you make changes to the `/etc/cups/cupsd.conf` configuration file, you'll need to restart the CUPS server by typing the following command at a terminal prompt:

```
sudo /etc/init.d/cupsys restart
```

Some other configuration for the CUPS server is done in the file `/etc/cups/cups.d/ports.conf`:

- **Listen**: By default on Ubuntu, the CUPS server installation listens only on the loopback interface at IP address *127.0.0.1*. In order to instruct the CUPS server to listen on an actual network adapter's IP address, you must specify either a hostname, the IP address, or optionally, an IP address/port pairing via the addition of a Listen directive. For example, if your CUPS server resides on a local network at the IP address *192.168.10.250* and you'd like to make it accessible to the other systems on this subnetwork, you would edit the `/etc/cups/cups.d/ports.conf` and add a Listen directive, as such:

  ```
  Listen 127.0.0.1:631           # existing loopback Listen
  Listen /var/run/cups/cups.sock # existing socket Listen
  Listen 192.168.10.250:631      # Listen on the LAN interface, Port 631 (IPP)
  ```

  In the example above, you may comment out or remove the reference to the Loopback address (127.0.0.1) if you do not wish cupsd to listen on that interface, but would rather have it only listen on the Ethernet interfaces of the Local Area Network (LAN). To enable listening for all network interfaces for which a certain hostname is bound, including the Loopback, you could create a Listen entry for the hostname *socrates* as such:

  ```
  Listen socrates:631  # Listen on all interfaces for the hostname 'socrates'
  ```

  or by omitting the Listen directive and using *Port* instead, as in:

```
Port 631  # Listen on port 631 on all interfaces
```

9.3. References

CUPS Website [http://www.cups.org/]

10. HTTPD - Apache2 Web Server

Apache is the most commonly used Web Server on GNU/Linux systems. Web Servers are used to serve Web Pages requested by client computers. Clients typically request and view Web Pages using Web Browser applications such as Firefox, Opera, or Mozilla.

Users enter a Uniform Resource Locator (URL) to point to a Web server by means of its Fully Qualified Domain Name (FQDN) and a path to the required resource. For example, to view the home page of the *Ubuntu Web site* [http://www.ubuntu.com] a user will enter only the FQDN. To request specific information about *paid support* [http://www.ubuntu.com/support/supportoptions/paidsupport], a user will enter the FQDN followed by a path.

The most common protocol used to transfer Web pages is the Hyper Text Transfer Protocol (HTTP). Protocols such as Hyper Text Transfer Protocol over Secure Sockets Layer (HTTPS), and File Transfer Protocol (FTP), a protocol for uploading and downloading files, are also supported.

Apache Web Servers are often used in combination with the MySQL database engine, the HyperText Preprocessor (PHP) scripting language, and other popular scripting languages such as Python and Perl. This configuration is termed LAMP (Linux, Apache, MySQL and Perl/Python/PHP) and forms a powerful and robust platform for the development and deployment of Web-based applications.

10.1. Installation

The Apache2 web server is available in Ubuntu Linux. To install Apache2:

- At a terminal prompt enter the following command:

```
sudo apt-get install apache2
```

10.2. Configuration

Apache is configured by placing *directives* in plain text configuration files. The main configuration file is called `apache2.conf`. In addition, other configuration files may be added using the *Include* directive, and wildcards can be used to include many configuration files. Any directive may be placed in any of these configuration files. Changes to the main configuration files are only recognized by Apache2 when it is started or restarted.

The server also reads a file containing mime document types; the filename is set by the *TypesConfig* directive, and is `mime.types` by default.

The default Apache2 configuration file is `/etc/apache2/apache2.conf` . You can edit this file to configure the Apache2 server. You can configure the port number, document root, modules, log files, virtual hosts, etc.

10.2.1. Basic Settings

This section explains Apache2 server essential configuration parameters. Refer to the *Apache2 Documentation* [http://httpd.apache.org/docs/2.0/] for more details.

- Apache2 ships with a virtual-host-friendly default configuration. That is, it is configured with a single default virtual host (using the *VirtualHost* directive) which can modified or used as-is if you have a single site, or used as a template for additional virtual hosts if you have multiple sites. If left alone, the default virtual host will serve as your default site, or the site users will see if the URL they enter does not match the *ServerName* directive of any of your custom sites. To modify the default virtual host, edit the file `/etc/apache2/sites-available/default`. If you wish to configure a new virtual host or site, copy that file into the same directory with a name you choose. For example, **sudo cp /etc/apache2/sites-available/default /etc/apache2/sites-available/mynewsite** Edit the new file to configure the new site using some of the directives described below.
- The *ServerAdmin* directive specifies the email address to be advertised for the server's administrator. The default value is webmaster@localhost. This should be changed to an email address that is delivered to you (if you are the server's administrator). If your website has a problem, Apache2 will display an error message containing this email address to report the problem to. Find this directive in your site's configuration file in /etc/apache2/sites-available.
- The *Listen* directive specifies the port, and optionally the IP address, Apache2 should listen on. If the IP address is not specified, Apache2 will listen on all IP addresses assigned to the machine it runs on. The default value for the Listen directive is 80. Change this to 127.0.0.1:80 to cause Apache2 to listen only on your loopback interface so that it will not be available to the Internet, to (for example) 81 to change the port that it listens on, or leave it as is for normal operation. This directive can be found and changed in its own file, `/etc/apache2/ports.conf`
- The *ServerName* directive is optional and specifies what FQDN your site should answer to. The default virtual host has no ServerName directive specified, so it will respond to all requests that do not match a ServerName directive in another virtual host. If you have just acquired the domain name ubunturocks.com and wish to host it on your Ubuntu server, the value of the ServerName directive in your virtual host configuration file should be ubunturocks.com. Add this directive to the new virtual host file you created earlier (`/etc/apache2/sites-available/mynewsite`).

  

  You may also want your site to respond to www.ubunturocks.com, since many users will assume the www prefix is appropriate. Use the *ServerAlias* directive for this. You may also use wildcards in the ServerAlias directive. For example, **ServerAlias *.ubunturocks.com** will cause your site to respond to any domain request ending in .ubunturocks.com.
- The *DocumentRoot* directive specifies where Apache should look for the files that make up the site. The default value is /var/www. No site is configured there, but if you uncomment the *RedirectMatch* directive in `/etc/apache2/apache2.conf` requests will be redirected to /var/www/apache2-default where the default Apache2 site awaits. Change this value in your site's virtual host file, and remember to create that directory if necessary!

The /etc/apache2/sites-available directory is **not** parsed by Apache2. Symbolic links in /etc/apache2/sites-enabled point to "available" sites. Use the a2ensite (Apache2 Enable Site) utility to create those symbolic links, like so: **sudo a2ensite mynewsite** where your site's configuration file is `/etc/apache2/sites-available/mynewsite`. Similarly, the a2dissite utility should be used to disable sites.

10.2.2. Default Settings

This section explains configuration of the Apache2 server default settings. For example, if you add a virtual host, the settings you configure for the virtual host take precedence for that virtual host. For a directive not defined within the virtual host settings, the default value is used.

- The *DirectoryIndex* is the default page served by the server when a user requests an index of a directory by specifying a forward slash (/) at the end of the directory name.

  For example, when a user requests the page http://www.example.com/this_directory/, he or she will get either the DirectoryIndex page if it exists, a server-generated directory list if it does not and the Indexes option is specified, or a Permission Denied page if neither is true. The server will try to find one of the files listed in the DirectoryIndex directive and will return the first one it finds. If it does not find any of these files and if Options Indexes is set for that directory, the server will generate and return a list, in HTML format, of the subdirectories and files in the directory. The default value, found in `/etc/apache2/apache2.conf` is " index.html index.cgi index.pl index.php index.xhtml". Thus, if Apache2 finds a file in a requested directory matching any of these names, the first will be displayed.
- The *ErrorDocument* directive allows you to specify a file for Apache to use for specific error events. For example, if a user requests a resource that does not exist, a 404 error will occur, and per Apache2's default configuration, the file `/usr/share/apache2/error/HTTP_NOT_FOUND.html.var` will be displayed. That file is not in the server's DocumentRoot, but there is an Alias directive in `/etc/apache2/apache2.conf` that redirects requests to the /error directory to /usr/share/apache2/error/. To see a list of the default ErrorDocument directives, use this command: **grep ErrorDocument /etc/apache2/apache2.conf**
- By default, the server writes the transfer log to the file /var/log/apache2/access.log. You can change this on a per-site basis in your virtual host configuration files with the *CustomLog* directive, or omit it to accept the default, specified in `/etc/apache2/apache2.conf`. You may also specify the file to which errors are logged, via the *ErrorLog* directive, whose default is `/var/log/apache2/error.log`. These are kept separate from the transfer logs to aid in troubleshooting problems with your Apache2 server. You may also specify the *LogLevel* (the default value is "warn") and the *LogFormat* (see `/etc/apache2/apache2.conf` for the default value).
- Some options are specified on a per-directory basis rather than per-server. Option is one of these directives. A Directory stanza is enclosed in XML-like tags, like so:

```
<Directory /var/www/mynewsite>
    ...
  </Directory>
```

The Options directive within a Directory stanza accepts one or more of the following values (among others), separated by spaces:

- **ExecCGI** - Allow execution of CGI scripts. CGI scripts are not executed if this option is not chosen.

Most files should not be executed as CGI scripts. This would be very dangerous. CGI scripts should kept in a directory separate from and outside your DocumentRoot, and only this directory should have the ExecCGI option set. This is the default, and the default location for CGI scripts is /usr/lib/cgi-bin.

- **Includes** - Allow server-side includes. Server-side includes allow an HTML file to *include* other files. This is not a common option. See *the Apache2 SSI Howto* [http://httpd.apache.org/docs/2.0/howto/ssi.html] for mor information.
- **IncludesNOEXEC** - Allow server-side includes, but disable the #exec and #include commands in CGI scripts.
- **Indexes** - Display a formatted list of the directory's contents, if no DirectoryIndex (such as index.html) exists in the requested directory.

For security reasons, this should usually not be set, and certainly should not be set on your DocumentRoot directory. Enable this option carefully on a per-directory basis only if you are certain you want users to see the entire contents of the directory.

- **Multiview** - Support content-negotiated multiviews; this option is disabled by default for security reasons. See the *Apache2 documentation on this option* [http://httpd.apache.org/docs/2.0/mod/mod_negotiation.html#multiviews].
- **SymLinksIfOwnerMatch** - Only follow symbolic links if the target file or directory has the same owner as the link.

10.2.3. Virtual Hosts Settings

Virtual hosts allow you to run different servers for different IP addresses, different host names, or different ports on the same machine. For example, you can run the website for http://www.example.com and http://www.anotherexample.com on the same Web server using virtual hosts. This option corresponds to the <VirtualHost> directive for the default virtual host and IP-based virtual hosts. It corresponds to the <NameVirtualHost> directive for a name-based virtual host.

The directives set for a virtual host only apply to that particular virtual host. If a directive is set server-wide and not defined within the virtual host settings, the default setting is used. For example, you can define a Webmaster email address and not define individual email addresses for each virtual host.

Set the DocumentRoot directive to the directory that contains the root document (such as index.html) for the virtual host. The default DocumentRoot is `/var/www`.

The ServerAdmin directive within the VirtualHost stanza is email the address used in the footer of error pages if you choose to show a footer with an email address on the error pages.

10.2.4. Server Settings

This section explains how to configure basic server settings.

LockFile - The LockFile directive sets the path to the lockfile used when the server is compiled with either USE_FCNTL_SERIALIZED_ACCEPT or USE_FLOCK_SERIALIZED_ACCEPT. It must be stored on the local disk. It should be left to the default value unless the logs directory is located on an NFS share. If this is the case, the default value should be changed to a location on the local disk and to a directory that is readable only by root.

PidFile - The PidFile directive sets the file in which the server records its process ID (pid). This file should only be readable by root. In most cases, it should be left to the default value.

User - The User directive sets the userid used by the server to answer requests. This setting determines the server's access. Any files inaccessible to this user will also be inaccessible to your website's visitors. The default value for User is www-data.

Unless you know exactly what you are doing, do not set the User directive to root. Using root as the User will create large security holes for your Web server.

The Group directive is similar to the User directive. Group sets the group under which the server will answer requests. The default group is also www-data.

10.2.5. Apache Modules

Apache is a modular server. This implies that only the most basic functionality is included in the core server. Extended features are available through modules which can be loaded into Apache. By default, a base set of modules is included in the server at compile-time. If the server is compiled to use dynamically loaded modules, then modules can be compiled separately, and added at any time using the LoadModule directive. Otherwise, Apache must be recompiled to add or remove modules. Ubuntu compiles Apache2 to allow the dynamic loading of modules. Configuration directives may be conditionally included on the presence of a particular module by enclosing them in an <IfModule> block. You can install additional Apache2 modules and use them with your Web server. You can install Apache2 modules using the apt-get command. For example, to install the Apache2 module for MYSQL authentication, you can run the following command from a terminal prompt:

```
sudo apt-get install libapache2-mod-auth-mysql
```

Once you install the module, the module will be available in the `/etc/apache2/mods-available` directory. You can use the a2enmod command to enable a module. You can use the a2dismod command to disable a module. Once you enable the module, the module will be available in the the `/etc/apache2/mods-enabled` directory.

10.3. HTTPS Configuration

The mod_ssl module adds an important feature to the Apache2 server - the ability to encrypt communications. Thus, when your browser is communicating using SSL encryption, the https:// prefix is used at the beginning of the Uniform Resource Locator (URL) in the browser navigation bar.

The mod_ssl module is available in apache2-common package. If you have installed this package, you can run the following command from a terminal prompt to enable the mod_ssl module:

```
sudo a2enmod ssl
```

10.3.1. Certificates and Security

To set up your secure server, use public key cryptography to create a public and private key pair. In most cases, you send your certificate request (including your public key), proof of your company's identity, and payment to a Certificate Authority (CA). The CA verifies the certificate request and your identity, and then sends back a certificate for your secure server.

Alternatively, you can create your own self-signed certificate. Note, however, that self-signed certificates should not be used in most production environments. Self-signed certificates are not automatically accepted by a user's browser. Users are prompted by the browser to accept the certificate and create the secure connection.

Once you have a self-signed certificate or a signed certificate from the CA of your choice, you need to install it on your secure server.

10.3.2. Types of Certificates

You need a key and a certificate to operate your secure server, which means that you can either generate a self-signed certificate or purchase a CA-signed certificate. A CA-signed certificate provides two important capabilities for your server:

- Browsers (usually) automatically recognize the certificate and allow a secure connection to be made without prompting the user.
- When a CA issues a signed certificate, it is guaranteeing the identity of the organization that is providing the web pages to the browser.

Most Web browsers that support SSL have a list of CAs whose certificates they automatically accept. If a browser encounters a certificate whose authorizing CA is not in the list, the browser asks the user to either accept or decline the connection.

You can generate a self-signed certificate for your secure server, but be aware that a self-signed certificate does not provide the same functionality as a CA-signed certificate. A self-signed certificate is not automatically recognized by most Web browsers, and a self-signed certificate does not provide any guarantee concerning the identity of the organization that is providing the website. A CA-signed

certificate provides both of these important capabilities for a secure server. The process of getting a certificate from a CA is fairly easy. A quick overview is as follows:

1. Create a private and public encryption key pair.
2. Create a certificate request based on the public key. The certificate request contains information about your server and the company hosting it.
3. Send the certificate request, along with documents proving your identity, to a CA. We cannot tell you which certificate authority to choose. Your decision may be based on your past experiences, or on the experiences of your friends or colleagues, or purely on monetary factors.

   Once you have decided upon a CA, you need to follow the instructions they provide on how to obtain a certificate from them.
4. When the CA is satisfied that you are indeed who you claim to be, they send you a digital certificate.
5. Install this certificate on your secure server, and begin handling secure transactions.

Whether you are getting a certificate from a CA or generating your own self-signed certificate, the first step is to generate a key.

10.3.3. Generating a Certificate Signing Request (CSR)

To generate the Certificate Signing Request (CSR), you should create your own key. You can run the following command from a terminal prompt to create the key:

```
openssl genrsa -des3 -out server.key 1024
```

```
Generating RSA private key, 1024 bit long modulus
.....................++++++
.................++++++
unable to write 'random state'
e is 65537 (0x10001)
Enter pass phrase for server.key:
```

You can now enter your passphrase. For best security, it should at least contain eight characters. The minimum length when specifying -des3 is four characters. It should include numbers and/or punctuation and not be a word in a dictionary. Also remember that your passphrase is case-sensitive.

Re-type the passphrase to verify. Once you have re-typed it correctly, the server key is generated and stored in `server.key` file.

You can also run your secure web server without a passphrase. This is convenient because you will not need to enter the passphrase every time you start your secure web server. But it is highly insecure and a compromise of the key means a compromise of the server as well.

In any case, you can choose to run your secure web server without a passphrase by leaving out the -des3 switch in the generation phase or by issuing the following command at a terminal prompt:

```
openssl rsa -in server.key -out server.key.insecure
```

Once you run the above command, the insecure key will be stored in the `server.key.insecure` file. You can use this file to generate the CSR without passphrase.

To create the CSR, run the following command at a terminal prompt:

```
openssl req -new -key server.key -out server.csr
```

It will prompt you enter the passphrase. If you enter the correct passphrase, it will prompt you to enter Company Name, Site Name, Email Id, etc. Once you enter all these details, your CSR will be created and it will be stored in the `server.csr` file. You can submit this CSR file to a CA for processing. The CAN will use this CSR file and issue the certificate. On the other hand, you can create self-signed certificate using this CSR.

10.3.4. Creating a Self-Signed Certificate

To create the self-signed certificate, run the following command at a terminal prompt:

```
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

The above command will prompt you to enter the passphrase. Once you enter the correct passphrase, your certificate will be created and it will be stored in the `server.crt` file.

If your secure server is to be used in a production environment, you probably need a CA-signed certificate. It is not recommended to use self-signed certificate.

10.3.5. Installing the Certificate

You can install the key file `server.key` and certificate file `server.crt` or the certificate file issued by your CA by running following commands at a terminal prompt:

```
sudo cp server.crt /etc/ssl/certs
sudo cp server.key /etc/ssl/private
```

You should add the following four lines to the `/etc/apache2/sites-available/default` file or the configuration file for your secure virtual host. You should place them in the *VirtualHost* section. They should be placed under the *DocumentRoot* line:

```
SSLEngine on

SSLOptions +FakeBasicAuth +ExportCertData +CompatEnvVars +StrictRequire

SSLCertificateFile /etc/ssl/certs/server.crt
SSLCertificateKeyFile /etc/ssl/private/server.key
```

HTTPS should listen on port number 443. You should add the following line to the `/etc/apache2/ports.conf` file:

```
Listen 443
```

10.3.6. Accessing the Server

Once you install your certificate, you should restart your web server. You can run the following command at a terminal prompt to restart your web server:

```
sudo /etc/init.d/apache2 restart
```

You should remember and enter the passphrase every time you start your secure web server.

You will be prompted to enter the passphrase. Once you enter the correct passphrase, the secure web server will be started. You can access the secure server pages by typing https://your_hostname/url/ in your browser address bar.

10.4. References

Apache2 Documentation [http://httpd.apache.org/docs/2.0/]

Mod SSL Documentation [http://www.modssl.org/docs/]

11. Squid - Proxy Server

Squid is a full-featured web proxy cache server application which provides proxy and cache services for Hyper Text Transport Protocol (HTTP), File Transfer Protocol (FTP), and other popular network protocols. Squid can implement caching and proxying of Secure Sockets Layer (SSL) requests and caching of Domain Name Server (DNS) lookups, and perform transparent caching. Squid also supports a wide variety of caching protocols, such as Internet Cache Protocol, (ICP) the Hyper Text Caching Protocol, (HTCP) the Cache Array Routing Protocol (CARP), and the Web Cache Coordination Protocol. (WCCP)

The Squid proxy cache server is an excellent solution to a variety of proxy and caching server needs, and scales from the branch office to enterprise level networks while providing extensive, granular access control mechanisms and monitoring of critical parameters via the Simple Network Management Protocol (SNMP). When selecting a computer system for use as a dedicated Squid proxy, or caching servers, ensure your system is configured with a large amount of physical memory, as Squid maintains an in-memory cache for increased performance.

11.1. Installation

At a terminal prompt, enter the following command to install the Squid server:

```
sudo apt-get install squid squid-common
```

11.2. Configuration

Squid is configured by editing the directives contained within the `/etc/squid/squid.conf` configuration file. The following examples illustrate some of the directives which may be modified to affect the behavior of the Squid server. For more in-depth configuration of Squid, see the References section.

Prior to editing the configuration file, you should make a copy of the original file and protect it from writing so you will have the original settings as a reference, and to re-use as necessary.

Copy the `/etc/squid/squid.conf` file and protect it from writing with the following commands entered at a terminal prompt:

```
sudo cp /etc/squid/squid.conf /etc/squid/squid.conf.original
sudo chmod a-w /etc/squid/squid.conf.original
```

- To set your Squid server to listen on TCP port 8888 instead of the default TCP port 3128, change the http_port directive as such:

  http_port 8888

- Change the visible_hostname directive in order to give the Squid server a specific hostname. This hostname does not necessarily need to be the computer's hostname. In this example it is set to *weezie*

  visible_hostname weezie
- Again, Using Squid's access control, you may configure use of Internet services proxied by Squid to be available only users with certain Internet Protocol (IP) addresses. For example, we willll illustrate access by users of the 192.168.42.0/24 subnetwork only:

  Add the following to the **bottom** of the ACL section of your `/etc/squid/squid.conf` file:

  acl fortytwo_network src 192.168.42.0/24

  Then, add the following to the **top** of the http_access section of your `/etc/squid/squid.conf` file:

  http_access allow fortytwo_network
- Using the excellent access control features of Squid, you may configure use of Internet services proxied by Squid to be available only during normal business hours. For example, we'll illustrate access by employees of a business which is operating between 9:00AM and 5:00PM, Monday through Friday, and which uses the 10.1.42.0/42 subnetwork:

  Add the following to the **bottom** of the ACL section of your `/etc/squid/squid.conf` file:

  acl biz_network src 10.1.42.0/24 acl biz_hours time M T W T F 9:00-17:00

  Then, add the following to the **top** of the http_access section of your `/etc/squid/squid.conf` file:

  http_access allow biz_network biz_hours

After making changes to the `/etc/squid/squid.conf` file, save the file and restart the squid server application to effect the changes using the following command entered at a terminal prompt:

```
sudo /etc/init.d/squid restart
```

11.3. References

Squid Website [http://www.squid-cache.org/]

12. Version Control System

Version control is the art of managing changes to information. It has long been a critical tool for programmers, who typically spend their time making small changes to software and then undoing those changes the next day. But the usefulness of version control software extends far beyond the bounds of the software development world. Anywhere you can find people using computers to manage information that changes often, there is room for version control.

12.1. Subversion

Subversion is an open source version control system. Using Subversion, you can record the history of source files and documents. It manages files and directories over time. A tree of files is placed into a central repository. The repository is much like an ordinary file server, except that it remembers every change ever made to files and directories.

12.1.1. Installation

To access Subversion repository using the HTTP protocol, you must install and configure a web server. Apache2 is proven to work with Subversion. Please refer to the HTTP subsection in the Apache2 section to install and configure Apache2. To access the Subversion repository using the HTTPS protocol, you must install and configure a digital certificate in your Apache 2 web server. Please refer to the HTTPS subsection in the Apache2 section to install and configure the digital certificate.

To install Subversion, run the following command from a terminal prompt:

```
sudo apt-get install subversion libapache2-svn
```

12.1.2. Server Configuration

This step assumes you have installed above mentioned packages on your system. This section explains how to create a Subversion repository and access the project.

12.1.2.1. Create Subversion Repository

The Subversion repository can be created using the following command from a terminal prompt:

```
svnadmin create /path/to/repos/project
```

12.1.3. Access Methods

Subversion repositories can be accessed (checked out) through many different methods --on local disk, or through various network protocols. A repository location, however, is always a URL. The table describes how different URL schemas map to the available access methods.

Table 4.1. Access Methods

Schema	Access Method
file://	direct repository access (on local disk)
http://	Access via WebDAV protocol to Subversion-aware Apache2 web server
https://	Same as http://, but with SSL encryption
svn://	Access via custom protocol to an svnserve server
svn+ssh://	Same as svn://, but through an SSH tunnel

In this section, we will see how to configure Subversion for all these access methods. Here, we cover the basics. For more advanced usage details, refer to the *svn book* [http://svnbook.red-bean.com/].

12.1.3.1. Direct repository access (file://)

This is the simplest of all access methods. It does not require any Subversion server process to be running. This access method is used to access Subversion from the same machine. The syntax of the command, entered at a terminal prompt, is as follows:

```
svn co file:///path/to/repos/project
```

or

```
svn co file://localhost/path/to/repos/project
```

If you do not specify the hostname, there are three forward slashes (///) -- two for the protocol (file, in this case) plus the leading slash in the path. If you specify the hostname, you must use two forward slashes (//).

The repository permissions depend on filesystem permissions. If the user has read/write permission, he can checkout from and commit to the repository.

12.1.3.2. Access via WebDAV protocol (http://)

To access the Subversion repository via WebDAV protocol, you must configure your Apache 2 web server. You must add the following snippet in your `/etc/apache2/apache2.conf` file:

```
<Location /svn>
 DAV svn
 SVNPath /path/to/repos
 AuthType Basic
 AuthName "Your repository name"
 AuthUserFile /etc/subversion/passwd
 <LimitExcept GET PROPFIND OPTIONS REPORT>
 Require valid-user
```

```
</LimitExcept>
</Location>
```

Next, you must create the `/etc/subversion/passwd` file. This file contains user authentication details. To add an entry, i.e. to add a user, you can run the following command from a terminal prompt:

```
htpasswd2 /etc/subversion/passwd user_name
```

This command will prompt you to enter the password. Once you enter the password, the user is added. Now, to access the repository you can run the following command:

```
svn co http://servername/svn
```

The password is transmitted as plain text. If you are worried about password snooping, you are advised to use SSL encryption. For details, please refer next section.

12.1.3.3. Access via WebDAV protocol with SSL encryption (https://)

Accessing Subversion repository via WebDAV protocol with SSL encryption (https://) is similar to http:// except that you must install and configure the digital certificate in your Apache2 web server.

You can install a digital certificate issued by a signing authority like Verisign. Alternatively, you can install your own self-signed certificate.

This step assumes you have installed and configured a digital certificate in your Apache 2 web server. Now, to access the Subversion repository, please refer to the above section! The access methods are exactly the same, except the protocol. You must use https:// to access the Subversion repository.

12.1.3.4. Access via custom protocol (svn://)

Once the Subversion repository is created, you can configure the access control. You can edit the `/path/to/repos/project/conf/svnserve.conf` file to configure the access control. For example, to set up authentication, you can uncomment the following lines in the configuration file:

```
# [general]
# password-db = passwd
```

After uncommenting the above lines, you can maintain the user list in the passwd file. So, edit the file `passwd` in the same directory and add the new user. The syntax is as follows:

```
username = password
```

For more details, please refer to the file.

Now, to access Subversion via the svn:// custom protocol, either from the same machine or a different machine, you can run svnserver using svnserve command. The syntax is as follows:

```
$ svnserve -d --foreground -r /path/to/repos
```

```
# -d -- daemon mode
# --foreground -- run in foreground (useful for debugging)
# -r -- root of directory to serve

For more usage details, please refer to:
$ svnserve --help
```

Once you run this command, Subversion starts listening on default port (3690). To access the project repository, you must run the following command from a terminal prompt:

```
svn co svn://hostname/project project --username user_name
```

Based on server configuration, it prompts for password. Once you are authenticated, it checks out the code from Subversion repository. To synchronize the project repository with the local copy, you can run the **update** sub-command. The syntax of the command, entered at a terminal prompt, is as follows:

```
cd project_dir ; svn update
```

For more details about using each Subversion sub-command, you can refer to the manual. For example, to learn more about the co (checkout) command, please run the following command from a terminal prompt:

```
svn co help
```

12.1.3.5. Access via custom protocol with SSL encryption (svn+ssh://)

The configuration and server process is same as in the svn:// method. For details, please refer to the above section. This step assumes you have followed the above step and started the#Subversion server using svnserve command.

It is also assumed that the ssh server is running on that machine and that it is allowing incoming connections. To confirm, please try to login to that machine using ssh. If you can login, everything is perfect. If you cannot login, please address it before continuing further.

The svn+ssh:// protocol is used to access the Subversion repository using SSL encryption. The data transfer is encrypted using this method. To access the project repository (for example with a checkout), you must use the following command syntax:

```
svn co svn+ssh://hostname/var/svn/repos/project
```

You must use the full path (/path/to/repos/project) to access the Subversion repository using this access method.

Based on server configuration, it prompts for password. You must enter the password you use to login via ssh. Once you are authenticated, it checks out the code from the Subversion repository.

12.2. CVS Server

CVS is a version control system. You can use it to record the history of source files.

12.2.1. Installation

At a terminal prompt, enter the following command to install cvs:

```
sudo apt-get install cvs
```

After you install cvs, you should install xinetd to start/stop the cvs server. At the prompt, enter the following command to install xinetd:

```
sudo apt-get install xinetd
```

12.2.2. Configuration

Once you install cvs, the repository will be automatically initialized. By default, the repository resides under the /var/lib/cvs directory. You can change this path by running following command:

```
cvs -d /your/new/cvs/repo init
```

Once the initial repository is set up, you can configure xinetd to start the CVS server. You can copy the following lines to the `/etc/xinetd/cvspserver` file.

```
service cvspserver
{
     port = 2401
     socket_type = stream
     protocol = tcp
     user = root
     wait = no
     type = UNLISTED
     server = /usr/bin/cvs
     server_args = -f --allow-root /var/lib/cvs pserver
     disable = no
}
```

Be sure to edit the repository if you have changed the default repository (/var/lib/cvs) directory.

Once you have configured xinetd you can start the cvs server by running following command:

```
sudo /etc/init.d/xinetd start
```

You can confirm that the CVS server is running by issuing the following command:

```
sudo netstat -tap | grep cvs
```

When you run this command, you should see the following line or something similar:

```
tcp        0      0 *:cvspserver            *:* LISTEN
```

From here you can continue to add users, add new projects, and manage the CVS server.

CVS allows the user to add users independently of the underlying OS installation. Probably the easiest way is to use the Linux Users for CVS, although it has potential security issues. Please refer to the CVS manual for details.

12.2.3. Add Projects

This section explains how to add new project to the CVS repository. Create the directory and add necessary document and source files to the directory. Now, run the following command to add this project to CVS repository:

```
cd your/project
cvs import -d :pserver:username@hostname.com:/var/lib/cvs -m "Importing my project to CVS repositor
```

You can use the CVSROOT environment variable to store the CVS root directory. Once you export the CVSROOT environment variable, you can avoid using -d option to above cvs command.

The string *new_project* is a vendor tag, and *start* is a release tag. They serve no purpose in this context, but since CVS requires them, they must be present.

When you add a new project, the CVS user you use must have write access to the CVS repository (/var/lib/cvs). By default, the src group has write access to the CVS repository. So, you can add the user to this group, and he can then add and manage projects in the CVS repository.

12.3. References

Subversion Home Page [http://subversion.tigris.org/]

Subversion Book [http://svnbook.red-bean.com/]

CVS Manual [http://ximbiot.com/cvs/manual/cvs-1.11.21/cvs_toc.html]

13. Databases

Ubuntu provides two Database servers. They are:

- MySQL™
- PostgreSQL

They are available in the main repository. This section explains how to install and configure these database servers.

13.1. MySQL

MySQL is a fast, multi-threaded, multi-user, and robust SQL database server. It is intended for mission-critical, heavy-load production systems as well as for embedding into mass-deployed software.

13.1.1. Installation

To install MySQL, run the following command from a terminal prompt:

```
sudo apt-get install mysql-server mysql-client
```

Once the installation is complete, the MySQL server should be started automatically. You can run the following command from a terminal prompt to check whether the MySQL server is running:

```
sudo netstat -tap | grep mysql
```

When you run this command, you should see the following line or something similar:

```
tcp        0      0 localhost.localdomain:mysql           *:* LISTEN -
```

If the server is not running correctly, you can type the following command to start it:

```
sudo /etc/init.d/mysql restart
```

13.1.2. Configuration

By default, the administrator password is not set. Once you install MySQL, the first thing you must do is to configure the MySQL administrator password. To do this, run the following commands:

```
sudo mysqladmin -u root password newrootsqlpassword
```

```
sudo mysqladmin -u root -h localhost password newrootsqlpassword
```

You can edit the `/etc/mysql/my.cnf` file to configure the basic settings -- log file, port number, etc. Refer to `/etc/mysql/my.cnf` file for more details.

13.2. PostgreSQL

PostgreSQL is an object-relational database system that has the features of traditional commercial database systems with enhancements to be found in next-generation DBMS systems.

13.2.1. Installation

To install PostgreSQL, run the following command in the command prompt:

```
sudo apt-get install postgresql
```

Once the installation is complete, you should configure the PostgreSQL server based on your needs, although the default configuration is viable.

13.2.2. Configuration

By default, connection via TCP/IP is disabled. PostgreSQL supports multiple client authentication methods. By default, IDENT authentication method is used. Please refer *the PostgreSQL Administrator's Guide* [http://www.postgresql.org/docs/8.1/static/admin.html].

The following discussion assumes that you wish to enable TCP/IP connections and use the MD5 method for client authentication. PostgreSQL configuration files are stored in the `/etc/postgresql/<version>/main` directory. For example, if you install PostgreSQL 7.4, the configuration files are stored in the `/etc/postgresql/7.4/main` directory.

To configure ident authentication, add entries to the `/etc/postgresql/7.4/main/pg_ident.conf` file.

To enable TCP/IP connections, edit the file `/etc/postgresql/7.4/main/postgresql.conf`

Locate the line *#tcpip_socket = false* and change it to *tcpip_socket = true*. You may also edit all other parameters, if you know what you are doing! For details, refer to the configuration file or to the PostgreSQL documentation.

By default, the user credentials are not set for *MD5* client authentication. So, first it is necessary to configure the PostgreSQL server to use *trust* client authentication, connect to the database, configure the password, and revert the configuration back to use *MD5* client authentication. To enable *trust* client authentication, edit the file `/etc/postgresql/7.4/main/pg_hba.conf`

Comment out all the existing lines which use *ident* and *MD5* client authentication and add the following line:

```
local   all         postgres                          trust sameuser
```

Then, run the following command to start the PostgreSQL server:

```
sudo /etc/init.d/postgresql start
```

Once the PostgreSQL server is successfully started, run the following command at a terminal prompt to connect to the default PostgreSQL template database

```
psql -U postgres -d template1
```

The above command connects to PostgreSQL database *template1* as user *postgres*. Once you connect to the PostgreSQL server, you will be at a SQL prompt. You can run the following SQL command at the psql prompt to configure the password for the user *postgres*.

```
template1=# ALTER USER postgres with encrypted password 'your_password';
```

After configuring the password, edit the file `/etc/postgresql/7.4/main/pg_hba.conf` to use *MD5* authentication:

Comment the recently added *trust* line and add the following line:

```
local   all         postgres                          md5 sameuser
```

The above configuration is not complete by any means. Please refer *the PostgreSQL Administrator's Guide* [http://www.postgresql.org/docs/8.1/static/admin.html] to configure more parameters.

14. Email Services

The process of getting an email from one person to another over a network or the Internet involves many systems working together. Each of these systems must be correctly configured for the process to work. The sender uses a *Mail User Agent* (MUA), or email client, to send the message through one or more *Mail Transfer Agents* (MTA), the last of which will hand it off to a *Mail Delivery Agent* (MDA) for delivery to the recipient's mailbox, from which it will be retrieved by the recipient's email client, usually via a POP3 or IMAP server.

14.1. Postfix

Postfix is the default Mail Transfer Agent (MTA) in Ubuntu. It attempts to be fast and easy to administer and secure. It is compatible with the MTA sendmail. This section explains how to install and configure postfix. It also explains how to set it up as an SMTP server using a secure connection (for sending emails securely).

14.1.1. Installation

To install postfix with SMTP-AUTH and Transport Layer Security (TLS), run the following command:

```
sudo apt-get install postfix
```

Simply press return when the installation process asks questions, the configuration will be done in greater detail in the next stage.

14.1.2. Basic Configuration

To configure postfix, run the following command:

```
sudo dpkg-reconfigure postfix
```

The user interface will be displayed. On each screen, select the following values:

- Ok
- Internet Site
- NONE
- mail.example.com
- mail.example.com, localhost.localdomain, localhost
- No
- 127.0.0.0/8
- Yes
- 0
- +

- all

Replace mail.example.com with your mail server hostname.

14.1.3. SMTP Authentication

The next steps are to configure postfix to use SASL for SMTP AUTH. Rather than editing the configuration file directly, you can use the **postconf** command to configure all postfix parameters. The configuration parameters will be stored in `/etc/postfix/main.cf` file. Later if you wish to re-configure a particular parameter, you can either run the command or change it manually in the file.

1. Configure Postfix to do SMTP AUTH using SASL (saslauthd):

```
postconf -e 'smtpd_sasl_local_domain ='
postconf -e 'smtpd_sasl_auth_enable = yes'
postconf -e 'smtpd_sasl_security_options = noanonymous'
postconf -e 'broken_sasl_auth_clients = yes'
postconf -e 'smtpd_recipient_restrictions = permit_sasl_authenticated,permit_mynetworks,reject_
postconf -e 'inet_interfaces = all'
echo 'pwcheck_method: saslauthd' >> /etc/postfix/sasl/smtpd.conf
echo 'mech_list: plain login' >> /etc/postfix/sasl/smtpd.conf
```

2. Next, configure the digital certificate for TLS. When asked questions, follow the instructions and answer appropriately.

```
openssl genrsa -des3 -rand /etc/hosts -out smtpd.key 1024
chmod 600 smtpd.key
openssl req -new -key smtpd.key -out smtpd.csr
openssl x509 -req -days 3650 -in smtpd.csr -signkey smtpd.key -out smtpd.crt
openssl rsa -in smtpd.key -out smtpd.key.unencrypted
mv -f smtpd.key.unencrypted smtpd.key
openssl req -new -x509 -extensions v3_ca -keyout cakey.pem -out cacert.pem -days 3650
mv smtpd.key /etc/ssl/private/
mv smtpd.crt /etc/ssl/certs/
mv cakey.pem /etc/ssl/private/
mv cacert.pem /etc/ssl/certs/
```

You can get the digital certificate from a certificate authority. Alternatively, you can create the certificate yourself. Refer to *Section 10.3.4, “Creating a Self-Signed Certificate” [p. 52]* for more details.

3. Configure Postfix to do TLS encryption for both incoming and outgoing mail:

```
postconf -e 'smtpd_tls_auth_only = no'
postconf -e 'smtp_use_tls = yes'
postconf -e 'smtpd_use_tls = yes'
postconf -e 'smtp_tls_note_starttls_offer = yes'
postconf -e 'smtpd_tls_key_file = /etc/ssl/private/smtpd.key'
postconf -e 'smtpd_tls_cert_file = /etc/ssl/certs/smtpd.crt'
```

```
postconf -e 'smtpd_tls_CAfile = /etc/ssl/certs/cacert.pem'
postconf -e 'smtpd_tls_loglevel = 1'
postconf -e 'smtpd_tls_received_header = yes'
postconf -e 'smtpd_tls_session_cache_timeout = 3600s'
postconf -e 'tls_random_source = dev:/dev/urandom'
postconf -e 'myhostname = mail.example.com'
```

After you run all the commands, the SMTP AUTH is configured with postfix. The self-signed cerficiate is created for TLS and it is configured with postfix.

Now, the file `/etc/postfix/main.cf` should look like *this* [../sample/postfix_configuration].

The postfix initial configuration is complete. Run the following command to start postfix daemon:

```
sudo /etc/init.d/postfix start
```

Now the postfix daemon is installed, configured and run successfully. Postfix supports SMTP AUTH as defined in *RFC2554* [ftp://ftp.isi.edu/in-notes/rfc2554.txt]. It is based on *SASL* [ftp://ftp.isi.edu/in-notes/rfc2222.txt]. However it is still necessary to set up SASL authentication before you can use SMTP.

14.1.4. Configuring SASL

The libsasl2, sasl2-bin and libsasl2-modules are necessary to enable SMTP AUTH using SASL. You can install these applications if you have not installed them already.

```
apt-get install libsasl2 sasl2-bin
```

A few changes are necessary to make it work properly. Because Postfix runs chrooted in `/var/spool/postfix`, SASL needs to be configured to run in the false root (`/var/run/saslauthd` becomes `/var/spool/postfix/var/run/saslauthd`):

```
mkdir -p /var/spool/postfix/var/run/saslauthd
rm -rf /var/run/saslauthd
```

To activate saslauthd, edit the file `/etc/default/saslauthd`, and change or add the START variable. In order to configure saslauthd to run in the false root, add the PWDIR, PIDFILE and PARAMS variables. Finally, configure the MECHANISMS variable to your liking. The file should look like this:

```
# This needs to be uncommented before saslauthd will be run
# automatically
START=yes

PWDIR="/var/spool/postfix/var/run/saslauthd"
PARAMS="-m ${PWDIR}"
PIDFILE="${PWDIR}/saslauthd.pid"
```

```
# You must specify the authentication mechanisms you wish to use.
# This defaults to "pam" for PAM support, but may also include
# "shadow" or "sasldb", like this:
# MECHANISMS="pam shadow"

MECHANISMS="pam"
```

If you prefer, you can use **shadow** instead of **pam**. This will use MD5 hashed password transfer and is perfectly secure. The username and password needed to authenticate will be those of the users on the system you are using on the server.

Next, update the dpkg "state" of `/var/spool/portfix/var/run/saslauthd`. The saslauthd init script uses this setting to create the missing directory with the appropriate permissions and ownership:

```
dpkg-statoverride --force --update --add root sasl 755 /var/spool/postfix/var/run/saslauthd
```

14.1.5. Testing

SMTP AUTH configuration is complete. Now it is time to start and test the setup. You can run the following command to start the SASL daemon:

```
sudo /etc/init.d/saslauthd start
```

To see if SMTP-AUTH and TLS work properly, run the following command:

```
telnet mail.example.com 25
```

After you have established the connection to the postfix mail server, type:

```
ehlo mail.example.com
```

If you see the following lines among others, then everything is working perfectly. Type **quit** to exit.

```
250-STARTTLS
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250 8BITMIME
```

14.2. Exim4

Exim4 is is another Message Transfer Agent (MTA) developed at the University of Cambridge for use on Unix systems connected to the internet. Exim can be installed in place of sendmail, although the configuration of exim is quite different to that of sendmail.

14.2.1. Installation

To install exim4, run the following command:

```
sudo apt-get install exim4 exim4-base exim4-config
```

14.2.2. Configuration

To configure exim4, run the following command:

```
sudo dpkg-reconfigure exim4-config
```

The user interface will be displayed. The user interface lets you configure many parameters. For example, In exim4 the configuration files are split among multiple files. If you wish to have them in one file you can configure accordingly in this user interface.

All the parameters you configure in the user interface are stored in `/etc/exim4/update-exim4.conf.conf` file. If you wish to re-configure, either you re-run the configuration wizard or manually edit this file using your favourite editor. Once you configure, you can run the following command to generate the master configuration file:

```
sudo update-exim4.conf
```

The master configuration file, is generated and it is stored in `/var/lib/exim4/config.autogenerated`.

At any time, you should not edit the master configuration file, `/var/lib/exim4/config.autogenerated` manually. It is updated automatically every time you run **update-exim4.conf**

You can run the following command to start exim4 daemon.

```
sudo /etc/init.d/exim4 start
```

TODO: This section should cover configuring SMTP AUTH with exim4.

14.3. Dovecot Server

Dovecot is a Mail Delivery Agent, written with security primarily in mind. It supports the major mailbox formats: mbox or Maildir. This section explain how to set it up as an imap or pop3 server.

14.3.1. Installation

To install dovecot, run the following command in the command prompt:

```
sudo apt-get install dovecot-common dovecot-imapd dovecot-pop3d
```

14.3.2. Configuration

To configure dovecot, you can edit the file `/etc/dovecot/dovecot.conf`. You can choose the protocol you use. It could be pop3, pop3s (pop3 secure), imap and imaps (imap secure). A description of these protocols is beyond the scope of this guide. For further information,

refer to the wikipedia articles on *POP3* [http://en.wikipedia.org/wiki/POP3] and *IMAP* [http://en.wikipedia.org/wiki/Internet_Message_Access_Protocol].

IMAPS and POP3S are more secure that the simple IMAP and POP3 because they use SSL encryption to connect. Once you have chosen the protocol, amend the following line in the file `/etc/dovecot/dovecot.conf`:

```
protocols = pop3 pop3s imap imaps
```

It enables the protocols when dovecot is started. Next, add the following line in pop3 section in the file `/etc/dovecot/dovecot.conf`:

```
pop3_uidl_format = %08Xu%08Xv
```

Next, choose the mailbox you use. Dovecot supports **maildir** and **mbox** formats. These are the most commonly used mailbox formats. They both have their own benefits and they are discussed on *the dovecot website* [http://dovecot.org/doc/configuration.txt].

Once you have chosen your mailbox type, edit the file `/etc/dovecot/dovecot.conf` and change the following line:

```
default_mail_env = maildir:~/Maildir # (for maildir)
or
default_mail_env = mbox:~/mail:INBOX=/var/spool/mail/%u # (for mbox)
```

You should configure your Mail Trasport Agent (MTA) to transfer the incoming mail to this type of mailbox if it is different from the one you have configured.

Once you have configured dovecot, start the dovecot daemon in order to test your setup:

```
sudo /etc/init.d/dovecot start
```

If you have enabled imap, or pop3, you can also try to log in with the commands **telnet localhost pop3** or **telnet localhost** imap2. If you see something like the following, the installation has been successful:

```
bhuvan@rainbow:~$ telnet localhost pop3
Trying 127.0.0.1...
Connected to localhost.localdomain.
Escape character is '^]'.
+OK Dovecot ready.
```

14.3.3. Dovecot SSL Configuration

To configure dovecot to use SSL, you can edit the file `/etc/dovecot/dovecot.conf` and amend following lines:

```
ssl_cert_file = /etc/ssl/certs/dovecot.pem
ssl_key_file = /etc/ssl/private/dovecot.pem
ssl_disable = no
disable_plaintext_auth = no
```

The **cert** and **key** files are created automatically by dovecot when you install it. Please note that these keys are not signed and will give "bad signature" errors when connecting from a client. To avoid this, you can use commercial certificates, or even better, you can use your own SSL certificates.

14.3.4. Firewall Configuration for an Email Server

To access your mail server from another computer, you must configure your firewall to allow connections to the server on the necessary ports.

- IMAP - 143
- IMAPS - 993
- POP3 - 110
- POP3S - 995

14.4. Mailman

Mailman is an open source program for managing electronic mail discussions and e-newsletter lists. Many open source mailing lists (including all the *Ubuntu mailing lists* [http://lists.ubuntu.com]) use Mailman as their mailing list software. It is powerful and easy to install and maintain.

14.4.1. Installation

Mailman provides a web interface for the administrators and users. So, it requires apache with mod_perl support. Mailman uses an external mail server to send and receive emails. It works perfectly with the following mail servers:

- Postfix
- Exim
- Sendmail
- Qmail

We will see how to install mailman, the apache web server and the Exim mail server. If you wish to install mailman with a different mail server, please refer to the references section.

14.4.1.1. Apache2

To install apache2 you refer to *Section 10.1, “Installation” [p. 45]*.

14.4.1.2. Exim4

To install Exim4 you run the following commands at a terminal prompt:

```
sudo apt-get install exim4
sudo apt-get install exim4-base
sudo apt-get install exim4-config
```

Once exim4 is installed, the configuration files are stored in the `/etc/exim4` directory. In ubuntu, by default, the exim4 configuration files are split across different files. You can change this behavior by changing the following variable in the `/etc/exim4/update-exim4.conf` file:

- dc_use_split_config='true'

14.4.1.3. Mailman

To install Mailman, run following command at a terminal prompt:

```
sudo apt-get install mailman
```

It copies the installation files in /var/lib/mailman directory. It installs the CGI scripts in /usr/lib/cgi-bin/mailman directory. It creates *list* linux user. It creates the *list* linux group. The mailman process will be owned by this user.

14.4.2. Configuration

This section assumes you have successfully installed mailman, apache2, and exim4. Now you just need to configure them.

14.4.2.1. Apache2

Once apache2 is installed, you can add the following lines in the `/etc/apache2/apache2.conf` file:

```
Alias /images/mailman/ "/usr/share/images/mailman/"
Alias /pipermail/ "/var/lib/mailman/archives/public/"
```

Mailman uses apache2 to render its CGI scripts. The mailman CGI scripts are installed in the /usr/lib/cgi-bin/mailman directory. So, the mailman url will be http://hostname/cgi-bin/mailman/. You can make changes to the `/etc/apache2/apache2.conf` file if you wish to change this behavior.

14.4.2.2. Exim4

Once Exim4 is installed, you can start the Exim server using the following command from a terminal prompt:

```
sudo apt-get /etc/init.d/exim4 start
```

In order to make mailman work with exim4, you need to configure exim4. As mentioned earlier, by default, exim4 uses multiple configuration files of different types. For details, please refer to the *Exim* [http://www.exim.org] website. To run mailman, we should add new a configuration file to the following configuration types:

- Main

- Transport
- Router

Exim creates a master configuration file by sorting all these mini configuration files. So, the order of these configuration files is very important.

14.4.2.3. Main

All the configuration files belonging to the main type are stored in the `/etc/exim4/conf.d/main/` directory. You can add the following content to a new file, named `04_exim4-config_mailman`:

```
# start
# Home dir for your Mailman installation -- aka Mailman's prefix
# directory.
# On Ubuntu this should be "/var/lib/mailman"
# This is normally the same as ~mailman
MM_HOME=/var/lib/mailman
#
# User and group for Mailman, should match your --with-mail-gid
# switch to Mailman's configure script.  Value is normally "mailman"
MM_UID=list
MM_GID=list
#
# Domains that your lists are in - colon separated list
# you may wish to add these into local_domains as well
domainlist mm_domains=hostname.com
#
# -=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
#
# These values are derived from the ones above and should not need
# editing unless you have munged your mailman installation
#
# The path of the Mailman mail wrapper script
MM_WRAP=MM_HOME/mail/mailman
#
# The path of the list config file (used as a required file when
# verifying list addresses)
MM_LISTCHK=MM_HOME/lists/${lc::$local_part}/config.pck
# end
```

14.4.2.4. Transport

All the configuration files belonging to transport type are stored in the `/etc/exim4/conf.d/transport/` directory. You can add the following content to a new file named `40_exim4-config_mailman`:

```
  mailman_transport:
   driver = pipe
   command = MM_WRAP \
               '${if def:local_part_suffix \
```

```
                    {${sg{$local_part_suffix}{-(\\w+)(\\+.*)?}{\$1}}} \
                    {post}}' \
              $local_part
   current_directory = MM_HOME
   home_directory = MM_HOME
   user = MM_UID
   group = MM_GID
```

14.4.2.5. Router

All the configuration files belonging to router type are stored in the `/etc/exim4/conf.d/router/` directory. You can add the following content in to a new file named `101_exim4-config_mailman`:

```
mailman_router:
 driver = accept
 require_files = MM_HOME/lists/$local_part/config.pck
 local_part_suffix_optional
 local_part_suffix = -bounces : -bounces+* : \
                     -confirm+* : -join : -leave : \
                     -owner : -request : -admin
 transport = mailman_transport
```

The order of main and transport configuration files can be in any order. But, the order of router configuration files must be the same. This particular file must appear before the 200_exim4-config_primary file. These two configuration files contain same type of information. The first file takes the precedence. For more details, please refer to the references section.

14.4.2.6. Mailman

Once mailman is installed, you can run it using the following command:

```
sudo /etc/init.d/mailman start
```

Once mailman is installed, you should create the default mailing list. Run the following command to create the mailing list:

```
sudo /usr/sbin/newlist mailman
```

```
Enter the email address of the person running the list: bhuvan at ubuntu.com
Initial mailman password:
To finish creating your mailing list, you must edit your /etc/aliases (or
equivalent) file by adding the following lines, and possibly running the
`newaliases' program:

## mailman mailing list
mailman:              "|/var/lib/mailman/mail/mailman post mailman"
mailman-admin:        "|/var/lib/mailman/mail/mailman admin mailman"
```

```
mailman-bounces:      "|/var/lib/mailman/mail/mailman bounces mailman"
mailman-confirm:      "|/var/lib/mailman/mail/mailman confirm mailman"
mailman-join:         "|/var/lib/mailman/mail/mailman join mailman"
mailman-leave:        "|/var/lib/mailman/mail/mailman leave mailman"
mailman-owner:        "|/var/lib/mailman/mail/mailman owner mailman"
mailman-request:      "|/var/lib/mailman/mail/mailman request mailman"
mailman-subscribe:    "|/var/lib/mailman/mail/mailman subscribe mailman"
mailman-unsubscribe:  "|/var/lib/mailman/mail/mailman unsubscribe mailman"

Hit enter to notify mailman owner...

#
```

We have configured exim to recognize all emails from mailman. So, it is not mandatory to make any new entries in `/etc/aliases`. If you have made any changes to the configuration files, please ensure that you restart those services before continuing to next section.

14.4.3. Administration

We assume you have a default installation. The mailman cgi scripts are still in /usr/lib/cgi-bin/mailman/ directory. Mailman provides a web based administration facility. To access this page, point your browser to the following url:

http://hostname/cgi-bin/mailman/admin

The default mailing list, *mailman*, will appear in this screen. If you click the mailing list name, it will ask for your authentication password. If you enter the correct password, you will be able to change administrative settings of this mailing list. You can create a new mailing list using command line utility (**/usr/sbin/newlist**). Alternatively, you can create a new mailing list using web interface.

14.4.4. Users

Mailman provides a web based interface for users. To access this page, point your browser to the following url:

http://hostname/cgi-bin/mailman/listinfo

The default mailing list, *mailman*, will appear in this screen. If you click the mailing list name, it will display the subscription form. You can enter your email address, name (optional), and password to subscribe. An email invitation will be sent to you. You can follow the instructions in the email to subscribe.

14.4.5. References

GNU Mailman - Installation Manual [http://www.list.org/mailman-install/index.html]

HOWTO - Using Exim 4 and Mailman 2.1 together [http://www.exim.org/howto/mailman21.html]

Chapter 5. Windows Networking

Computer networks are often comprised of diverse systems, and while operating a network made up entirely of Ubuntu desktop and server computers would certainly be fun, some network environments must consist of both Ubuntu and Microsoft® Windows® systems working together in harmony. This section of the Ubuntu Server Guide introduces principles and tools used in configuring your Ubuntu Server for sharing network resources with Windows computers.

1. Introduction

Successfully networking your Ubuntu system with Windows clients involves providing and integrating with services common to Windows environments. Such services assist the sharing of data and information about the computers and users involved in the network, and may be classified under three major categories of functionality:

- **File and Printer Sharing Services**. Using the Server Message Block (SMB) protocol to facilitate the sharing of files, folders, volumes, and the sharing of printers throughout the network.
- **Directory Services**. Sharing vital information about the computers and users of the network with such technologies as the Lightweight Directory Access Protocol (LDAP) and Microsoft Active Directory®.
- **Authentication and Access**. Establishing the identity of a computer or user of the network and determining the information the computer or user is authorized to access using such principles and technologies as file permissions, group policies, and the Kerberos authentication service.

Fortunately, your Ubuntu system may provide all such facilities to Windows clients and share network resources among them. One of the principle pieces of software your Ubuntu system includes for Windows networking is the SAMBA suite of SMB server applications and tools. This section of the Ubuntu Server Guide will briefly introduce the installation and limited configuration of the SAMBA suite of server applications and utilities. Additional, detailed documentation and information on SAMBA is beyond the scope of this documentation, but exists on the *SAMBA website* [http://www.samba.org].

2. Installing SAMBA

At the prompt enter the following command to install the SAMBA server applications:

```
sudo apt-get install samba
```

3. Configuring SAMBA

You may configure the SAMBA server by editing the `/etc/samba/smb.conf` file to change the default settings or add new settings. More information about each setting is available in the comments of the `/etc/samba/smb.conf` file or by viewing the `/etc/samba/smb.conf` manual page from the prompt with the following command typed at a terminal prompt:

```
man smb.conf
```

Prior to editing the configuration file, you should make a copy of the original file and protect it from writing so you will have the original settings as a reference and to re-use as necessary.

Backup the `/etc/samba/smb.conf` file:

```
sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.original
```

Now, edit the `/etc/samba/smb.conf` file and make your changes.

3.1. Server

In addition to the SAMBA suite of file and printer sharing server applications, Ubuntu also includes other powerful server applications designed to provide additional network server functionality to Windows clients similar to the functionality provided by actual Windows servers. For example, Ubuntu offers centralized management of network resources such as computers and users via Directory Services, and facilitates the identification, and authorization of computers and users via Authentication Services.

The following sections will discuss SAMBA and the supporting technologies, such as Lightweight Directory Access Protocol (LDAP) server, and Kerberos authentication server in more detail. You will also learn about some of the available configuration directives available the SAMBA configuration file which facilitate network integration with Windows clients and servers.

3.1.1. Active Directory

Active Directory is a proprietary implementation of Directory Services by Microsoft, and is used to provide a means to share information about network resources and users. In addition to providing a centralized source of such information, Active Directory also acts as a centralized authentication security authority for the network. Active directory combines capabilities traditionally found in separate, specialized directory systems to simplify integration, management, and security of network resources. The SAMBA package may be configured to use Active Directory services from a Windows Domain Controller.

3.1.1.1. LDAP

The LDAP server application provides Directory Services functionality to Windows computers in a manner very similar to Microsoft Active Directory services. Such services include managing the identities and relationships of computers, users, and groups of computers or users that participate in the network, and providing a consistent means to describe, locate, and manage these resources. The freely available implementation of LDAP available for your Ubuntu system is called *OpenLDAP*. The server daemons responsible for handling OpenLDAP directory requests and the propagation of directory data from one LDAP server to another on Ubuntu, are slapd and slurpd. OpenLDAP may be used in conjunction with SAMBA to provide File, Print, and Directory services in much the same way a Windows Domain Controller does so long as SAMBA is compiled with LDAP support.

3.1.1.2. Kerberos

The Kerberos authentication security system is a standardized service for providing authentication to computers and users by means of a centralized server which grants encrypted authorization tickets accepted for authorization by any other computer using Kerberos. Benefits of Kerberos authentication include mutual authentication, delegated authentication, interoperability, and simplified trust management. The primary server daemons for handling the Kerberos authentication and Kerberos database administration on Ubuntu are krb5kdc and kadmin. SAMBA may use Kerberos as a mechanism for authenticating computers and users against a Windows Domain Controller. To do so, the Ubuntu system must have Kerberos installed, and the `/etc/samba/smb.conf` must be modified to select the the proper *realm* and *security* mode. For example, edit the `/etc/samba/smb.conf` file and add the values:

realm = DOMAIN_NAME

security = ADS

to the file, and save the file.

Be sure to replace the token DOMAIN_NAME in the example above with the actual name of your specific Windows Domain.

You will need to restart the SAMBA daemons to effect these changes. Restart the SAMBA daemons with the following command entered at a terminal prompt:

```
sudo /etc/init.d/samba restart
```

3.1.2. Computer Accounts

Computer Accounts are used in Directory Services to uniquely identify computer systems participating in a network, and are even treated in the same manner as users in terms of security. Computer accounts may have passwords just as user accounts do, and are subject to authorization to network resources in the same manner as user accounts. For example, if a network user, with a valid

account for a particular network attempts to authenticate with a network resource from a computer which does not have a valid computer account, depending upon policies enforced on the network, the user may be denied access to the resource if the computer the user is attempting authentication from is considered to be an unauthorized computer.

A computer account may be added to the SAMBA password file, provided the name of the computer being added exists as a valid user account in the local password database first. The syntax for adding a computer or machine account to the SAMBA password file is to use the smbpasswd command from a terminal prompt as follows:

```
sudo smbpasswd -a -m COMPUTER_NAME
```

Be sure to replace the token COMPUTER_NAME in the example above with the actual name of the specific computer you wish to add a machine account for.

3.1.3. File Permissions

File Permissions define the explicit rights a computer or user has to a particular directory, file, or set of files. Such permissions may be defined by editing the `/etc/samba/smb.conf` file and specifying the explicit permissions of a defined file share. For example, if you have defined a SAMBA share called *sourcedocs* and wish to give *read-only* permissions to the group of users known as *planning*, but wanted to allow writing to the share by the group called *authors* and the user named *richard*, then you could edit the `/etc/samba/smb.conf` file, and add the following entries under the *[sourcedocs]* entry:

read list = @planning

write list = @authors, richard

Save the `/etc/samba/smb.conf` for the changes to take effect.

Another possible permission is to declare *administrative* permissions to a particular shared resource. Users having administrative permissions may read, write, or modify any information contained in the resource the user has been given explicit administrative permissions to. For example, if you wanted to give the user *melissa* administrative permissions to the example *sourcedocs* share, you would edit the `/etc/samba/smb.conf` file, and add the following line under the *[sourcedocs]* entry:

admin users = melissa

Save the `/etc/samba/smb.conf` for the changes to take effect.

3.2. Clients

Ubuntu includes client applications and capabilities for accessing network resources shared with the SMB protocol. For example, a utility called smbclient allows for accessing remote shared

file-systems, in a manner similar to a File Transfer Protocol (FTP) client. To access a shared folder resource known as *documents* offered by a remote Windows computer named *bill* using smbclient for example, one would enter a command similar to the following at the prompt:

```
smbclient //bill/documents -U <username>
```

You will then be prompted for the password for the user name specified after the -U switch, and upon successful authentication, will be presented with a prompt where commands may be entered for manipulating and transferring files in a syntax similar to that used by non-graphical FTP clients. For more information on the smbclient utility, read the utility's manual page with the command:

```
man smbclient
```

Local mounting of remote network resources using the SMB protocol is also possible using the mount command. For example, to mount a shared folder named *project-code* on a Windows server named *development* as the user *dlightman* to your Ubuntu system's /mnt/pcode mount-point, you would issue this command at the prompt:

```
mount -t smbfs -o username=dlightman //development/project-code /mnt/pcode
```

You will then be prompted for the user password, and after successfully authenticating, the contents of the shared resource will be available locally via the mount-point specified as the last argument to the mount command. To disconnect the shared resource, simply use the umount command as you would with any other mounted file system. For example:

```
umount /mnt/pcode
```

3.2.1. User Accounts

User Accounts define persons with some level of authorization to use certain computer and network resources. Typically, in a network environment, a user account is provided to each person allowed to access a computer or network, where policies and permissions then define what explicit rights that user account has access to. To define SAMBA network users for your Ubuntu system, you may use the smbpasswd command. For example to add a SAMBA user to your Ubuntu system with the user name *jseinfeld*, you would enter this command at the prompt:

```
smbpasswd -a jseinfeld
```

The smbpasswd application will then prompt you to enter a password for the user:

```
New SMB password:
```

Enter the password you wish to set for the user, and the smbpasswd application will ask you to confirm the password:

```
Retype new SMB password:
```

Confirm the password, and smbpasswd will add the entry for the user to the SAMBA password file.

3.2.2. Groups

Groups define a collection of computers or users which have a common level of access to particular network resources and offer a level of granularity in controlling access to such resources. For example, if a group *qa* is defined and contains the users *freda*, *danika*, and *rob* and a second group *support* is defined and consists of users *danika*, *jeremy*, and *vincent* then certain network resources configured to allow access by the *qa* group will subsequently enable access by freda, danika, and rob, but not jeremy or vincent. Since the user *danika* belongs to both the *qa* and *support* groups, she will be able to access resources configured for access by both groups, whereas all other users will have only access to resources explicitly allowing the group they are part of.

When defining groups in the SAMBA configuration file, `/etc/samba/smb.conf` the recognized syntax is to preface the group name with an "@" symbol. For example, if you wished to define a group named *sysadmin* in a certain section of the `/etc/samba/smb.conf`, you would do so by entering the group name as **@sysadmin**.

3.2.3. Group Policy

Group Policy defines certain SAMBA configuration settings pertaining to the Domain or Workgroup computer accounts belong to, and other global settings for the SAMBA server. For example, if the SAMBA server belongs to a Workgroup of Windows computers called *LEVELONE*, then the `/etc/samba/smb.conf` could be edited, and the following value changed accordingly:

workgroup = LEVELONE

Save the file and restart the SAMBA daemons to affect the change.

Other important global policy settings include the *server string* which defines the NETBIOS server name reported by your Ubuntu system to other machines on the Windows-based network. This is the name your Ubuntu system will be recognized as by Windows clients and other computers capable of browsing the network with the SMB protocol. Additionally, you may specify the name and location of the SAMBA server's log file by using the *log file* directive in the `/etc/samba/smb.conf` file.

Some of the additional directives governing global group policy include specification of the global nature of all shared resources. For example, placing certain directives under the *[global]* heading of the `/etc/samba/smb.conf` file will affect all shared resources unless an overriding directive is placed under a particular shared resource heading. You specify all shares are browseable by all clients on the network by placing a *browseable* directive, which takes a Boolean argument, under the *[global]* heading in the `/etc/samba/smb.conf`. That is, if you edit the file and add the line:

browseable = true

under the *[global]* section of `/etc/samba/smb.conf`, then all shares provided by your Ubuntu system via SAMBA will be browseable by all authorized clients, unless a specific share contains a *browseable = false* directive, which will override the global directive.

Other examples which work in a similar manner, are the *public* and *writeable* directives. The *public* directive accepts a Boolean value and decides whether a particular shared resource is visible by all clients, authorized or not. The *writeable* directive also takes a Boolean value and defines whether a particular shared resource is writable by any and all network clients.

Appendix A. Creative Commons by Attribution-ShareAlike 2.0

CREATIVE COMMONS CORPORATION IS NOT A LAW FIRM AND DOES NOT PROVIDE LEGAL SERVICES. DISTRIBUTION OF THIS LICENSE DOES NOT CREATE AN ATTORNEY-CLIENT RELATIONSHIP. CREATIVE COMMONS PROVIDES THIS INFORMATION ON AN "AS-IS" BASIS. CREATIVE COMMONS MAKES NO WARRANTIES REGARDING THE INFORMATION PROVIDED, AND DISCLAIMS LIABILITY FOR DAMAGES RESULTING FROM ITS USE.

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

1. **Definitions.**
   a. **"Collective Work"** means a work, such as a periodical issue, anthology or encyclopedia, in which the Work in its entirety in unmodified form, along with a number of other contributions, constituting separate and independent works in themselves, are assembled into a collective whole. A work that constitutes a Collective Work will not be considered a Derivative Work (as defined below) for the purposes of this License.
   b. **"Derivative Work"** means a work based upon the Work or upon the Work and other pre-existing works, such as a translation, musical arrangement, dramatization, fictionalization, motion picture version, sound recording, art reproduction, abridgment, condensation, or any other form in which the Work may be recast, transformed, or adapted, except that a work that constitutes a Collective Work will not be considered a Derivative Work for the purpose of this License. For the avoidance of doubt, where the Work is a musical composition or sound recording, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered a Derivative Work for the purpose of this License.
   c. **"Licensor"** means the individual or entity that offers the Work under the terms of this License.
   d. **"Original Author"** means the individual or entity who created the Work.
   e. **"Work"** means the copyrightable work of authorship offered under the terms of this License.
   f. **"You"** means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received

express permission from the Licensor to exercise rights under this License despite a previous violation.

g. **"License Elements"** means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.

2. **Fair Use Rights.** Nothing in this license is intended to reduce, limit, or restrict any rights arising from fair use, first sale or other limitations on the exclusive rights of the copyright owner under copyright law or other applicable laws.

3. **License Grant.** Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

   a. to reproduce the Work, to incorporate the Work into one or more Collective Works, and to reproduce the Work as incorporated in the Collective Works;

   b. to create and reproduce Derivative Works;

   c. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission the Work including as incorporated in Collective Works;

   d. to distribute copies or phonorecords of, display publicly, perform publicly, and perform publicly by means of a digital audio transmission Derivative Works.

   e. For the avoidance of doubt, where the work is a musical composition:

      i. **"Performance Royalties Under Blanket Licenses."** Licensor waives the exclusive right to collect, whether individually or via a performance rights society (e.g. ASCAP, BMI, SESAC), royalties for the public performance or public digital performance (e.g. webcast) of the Work.

      ii. **"Mechanical Rights and Statutory Royalties."** Licensor waives the exclusive right to collect, whether individually or via a music rights society or designated agent (e.g. Harry Fox Agency), royalties for any phonorecord You create from the Work ("cover version") and distribute, subject to the compulsory license created by 17 USC Section 115 of the US Copyright Act (or the equivalent in other jurisdictions).

   f. **"Webcasting Rights and Statutory Royalties."** For the avoidance of doubt, where the Work is a sound recording, Licensor waives the exclusive right to collect, whether individually or via a performance-rights society (e.g. SoundExchange), royalties for the public digital performance (e.g. webcast) of the Work, subject to the compulsory license created by 17 USC Section 114 of the US Copyright Act (or the equivalent in other jurisdictions).

   The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. All rights not expressly granted by Licensor are hereby reserved.

4. **Restrictions.** The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may distribute, publicly display, publicly perform, or publicly digitally perform the Work only under the terms of this License, and You must include a copy of, or the Uniform Resource Identifier for, this License with every copy or phonorecord of the Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Work that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Work itself to be made subject to the terms of this License. If You create a Collective Work, upon notice from any Licensor You must, to the extent practicable, remove from the Collective Work any reference to such Licensor or the Original Author, as requested. If You create a Derivative Work, upon notice from any Licensor You must, to the extent practicable, remove from the Derivative Work any reference to such Licensor or the Original Author, as requested.

b. You may distribute, publicly display, publicly perform, or publicly digitally perform a Derivative Work only under the terms of this License, a later version of this License with the same License Elements as this License, or a Creative Commons iCommons license that contains the same License Elements as this License (e.g. Attribution-ShareAlike 2.0 Japan). You must include a copy of, or the Uniform Resource Identifier for, this License or other license specified in the previous sentence with every copy or phonorecord of each Derivative Work You distribute, publicly display, publicly perform, or publicly digitally perform. You may not offer or impose any terms on the Derivative Works that alter or restrict the terms of this License or the recipients' exercise of the rights granted hereunder, and You must keep intact all notices that refer to this License and to the disclaimer of warranties. You may not distribute, publicly display, publicly perform, or publicly digitally perform the Derivative Work with any technological measures that control access or use of the Work in a manner inconsistent with the terms of this License Agreement. The above applies to the Derivative Work as incorporated in a Collective Work, but this does not require the Collective Work apart from the Derivative Work itself to be made subject to the terms of this License.

c. If you distribute, publicly display, publicly perform, or publicly digitally perform the Work or any Derivative Works or Collective Works, You must keep intact all copyright notices for the Work and give the Original Author credit reasonable to the medium or means You are utilizing by conveying the name (or pseudonym if applicable) of the Original Author if supplied; the title of the Work if supplied; to the extent reasonably practicable, the Uniform Resource Identifier, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and in the case of a Derivative Work, a credit identifying the use of the Work in the Derivative Work (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). Such credit may be implemented in any reasonable manner; provided, however, that in the case

of a Derivative Work or Collective Work, at a minimum such credit will appear where any other comparable authorship credit appears and in a manner at least as prominent as such other comparable authorship credit.

5. **Representations, Warranties and Disclaimer**

   UNLESS OTHERWISE AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE MATERIALS, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. **Limitation on Liability.** EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. **Termination**

   a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Derivative Works or Collective Works from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.

   b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. **Miscellaneous**

   a. Each time You distribute or publicly digitally perform the Work or a Collective Work, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.

   b. Each time You distribute or publicly digitally perform a Derivative Work, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.

   c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without

further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.

d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.

Creative Commons is not a party to this License, and makes no warranty whatsoever in connection with the Work. Creative Commons will not be liable to You or any party on any legal theory for any damages whatsoever, including without limitation any general, special, incidental or consequential damages arising in connection to this license. Notwithstanding the foregoing two (2) sentences, if Creative Commons has expressly identified itself as the Licensor hereunder, it shall have all rights and obligations of Licensor.

Except for the limited purpose of indicating to the public that the Work is licensed under the CCPL, neither party will use the trademark "Creative Commons" or any related trademark or logo of Creative Commons without the prior written consent of Creative Commons. Any permitted use will be in compliance with Creative Commons' then-current trademark usage guidelines, as may be published on its website or otherwise made available upon request from time to time.

Creative Commons may be contacted at *http://creativecommons.org/*.

Appendix B. GNU Free Documentation License

Version 1.2, November 2002

Free Software Foundation, Inc.
51 Franklin St, Fifth Floor,
Boston,
MA
02110-1301
USA

Version 1.2, November 2002

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent

copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

GNU FDL Modification Conditions

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the *Addendum [p. 96]* below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network

location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in *section 4 [p. 93]* above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in

parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

> ## Sample Invariant Sections list
>
> Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

> ## Sample Invariant Sections list
>
> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.