Chapter 10

Server Configuration

This chapter discusses the basic configuration of various servers. These configurations provide a fundamental system that will operate, but do not elaborate on enhancements or improved security. It is the intent here to establish an operational system, where one can then enhance.

The configurations included in this chapter contain a little of the server operation and configuration. By utilizing this approach it is believed that the user will better understand the reasons for various configuration requirements. The servers presented are ones that might be utilized in a small business. This is a long chapter!

More advanced server requirements are detailed in Chapter 18.

Concepts Learned in this Chapter

- Internet Services and Protocol Identifiers
- Activating Server Functions
- Server Configuration for the more popular services

Table of Contents

10.1 Internet Services and Protocols

Before one can fully understand the transmission of data across the Internet, a basic knowledge of the various services and protocols needs to be understood.

10.1.1 Internet Protocols

Data is transferred across the Internet using specified protocols – rules that specify what information, and where it is located in each packet that carries data. These protocols vary depending upon the type of information to be carried, such as email, web, ftp and many others. This protocol is specified in the Data Link MAC Frame. A few of the more useful protocols include:

Protocol	Protocol ID
ip	0
icmp	1
tcp	6
egp	8
igp	9
udp	17
eigrp	88
ospf	89

10.1.2 Internet Services

Besides having to know how data is formatted, a standard set of ports, alias for the server name, have been assigned. One can view the port as a set of office doors in a (very) long hallway. Each door / port is assigned to a specific task. The Internet Service is specified in the Network Packet. Some of the more important doors that should be memorized include:

Port	Service
20/21	ftp
22	ssh
23	telnet
25	smtp
69	tftp
53	name server
80	http (web)
110	pop-3
220	imap3
2049	nfs
n/a	dhcp

dhcp does not have a service port because it is used prior to IP being activated, it uses the MAC address.

An IP Address plus the Port number is referred to as a **socket**. For example:

192.168.1.10:25 Socket for Internet Mail services

10.2 Activating Services

The bootup process is controlled by multiple startup scripts. The first file that is run by the Kernel is **init**, which starts the boot initiation process (recall the pstree command). Through the process, the various server functions are initiated.

In order to simplify the initialization process, all of the scripts for each server process is located in the **/etc/rc.d/init.d** directory in the Red Hat distribution. Each script can be used to start or stop a specific server process. The standard calls for the scripts to be in the /etc/init.d directory. Red does provide a symbolic link from /etc/init.d to /etc/rc.d/init.d .

During normal operation, we can issue the command:

/etc/rc.d/init.d/{server function} {start | stop | restart}

to either start, stop, or restart the specified server function. Examples include:

nfsd	Network File System
named	Domain Name Service
httpd	Web Server

Instead of remembering that long command sequence, there is an easier shortcut method. As an alternative, issue the command:

service {server function} {start | stop | restart}

The command **service** is an alias for **/etc/rc.d/init.d** .

Whenever a modification is made to the configuration of a server function, in order for it to be activated, the server function must be terminated and reinitiated. This causes the modified configuration file to be read as part of the startup process. Note that the whole system does not need to be rebooted.

10.2.1 CLI Interface

Sometimes we need to modify how we set our system when specifying the various server functions that were installed (or not installed) during the initial installation. First we must verify which services we are running. Red Hat provides a command called **chkconfig**, whereas other Linux vendors do not necessarily support this command. Issue the command:

$ **chkconfig – – list**

This will generate a list of all services and their operational status, something like the following:

keytable	**0:off**	**1:off**	**2:on**	**3:on**	**4:on**	**5:on**	**6:off**
network	**0:off**	**1:off**	**2:on**	**3:on**	**4:on**	**5:on**	**6:off**
httpd	**0:off**	**1:off**	**2:off**	**3:on**	**4:on**	**5:on**	**6:off**
named	**0:off**	**1:off**	**2:off**	**3:on**	**4:on**	**5:on**	**6:off**
linuxconf	**0:off**	**1:off**	**2:off**	**3:on**	**4:on**	**5:on**	**6:off**
inet	**0:off**	**1:off**	**2:off**	**3:on**	**4:on**	**5:on**	**6:off**
routed	**0:off**	**1:off**	**2:off**	**3:off**	**4:off**	**5:off**	**6:off**
smb	**0:off**	**1:off**	**2:off**	**3:off**	**4:off**	**5:off**	**6:off**

Recall from the **/etc/inittab** file the various run level options:

0 System halt
1 Single User Mode
2 Multi-User without NFS

3 Multi-User
4 May be user defined
5 X Windows Mode
6 Reboot

Single user mode – minimum configuration

If we wish to add a server service that was not previously installed, we issue the command:

$ **chkconfig – –add {service_name}**

If we are presently in run level 3, then issuing the command:

$ **chkconfig – –add routed**

would add the routed service (although it is available by default), then

$ **chkconfig – –list routed**

we get

routed 0:off 1:off 2:off 3:on 4:off 5:off 6:off

Note that only run level 3 is turned on, because that is the level the system was operating in. If we wish to change a different level, we can issue the command:

$ **chkconfig – –add routed – –level 5**

Now level 5 will be added to the operational list.

There are instances where we need to remove a service. This might be when our usage has grown and we need to split one of the services off to a new host. Issue the command:

$ **chkconfig – –del named**

which will terminate the service in level 3. Again, to terminate the service in another level, use the command:

$ **chkconfig – –del named – –level 5**

The above discussion is in general an observation of which services are turned on or off, with Red Hat version 7, this is much more important. By default in version 7, most services are turned off to enhance security, and must be activated if the service is to be operational. Note at the end of the file, you can observe another list of services that are above and beyond the service activation. These are separate services that are activated, with enhanced security, are available to be modified in the **/etc/xinetd/** directory. Services there include ***telnet, ftp, finger,*** and others may be enabled.

When we run the **chkconfig – –list** command, it will create a very long list – which is much longer than one can see easily. Assuming you know which service you are interested in, you should use the grep filtering command to be a little more selective. This command is formatted as:

$ **chkconfig – –list | grep {service}**

Before the services may be fully used, they must be activated. This can be done by either rebooting (not preferred) or by issuing the command **xinetd** .

If a service is already listed in the chkconfig listing, you may have to activate it. This is accomplished by issuing the command:

$ **chkconfig {server function} on**

Another option is to give the command:

$ chkconfig – –named on or
$ chkconfig – –named off

This will turn on or off levels 3, 4, and 5.

Before the service is fully active, we must perform two actions. The first is to re-read the services file. This is performed by issuing the **xinetd** command to restart the function. Finally we need to restart the service, since it was off when we booted the computer. This is performed by one of two methods:

$ /etc/rc.d/init.d/{service} restart or
$ service {service} restart
$ xinetd

The service has now been activated without rebooting the system.

10.2.2 GUI Interface

As an alternative to using the CLI mode for modifying the server operation, we desire to utilize a graphical format. Two options are available.

10.2.2.1 ntsysv

The first option is to bring up a screen similar to the one used during the installation. Issue the command:

$ **ntsysv**

We can now modify which services we want for the current run level. To select a different run level, issue the command:

ntsysv – –level 5

and do the same thing.

10.2.2.2 serviceconf

Another activation utility is **serviceconf**. This is very similar in functional design to **ntsysv**, but provides a more modern GUI. Additionally it provides icons for initializing the service, and a description box to describe each of the services. This is the recommended GUI interface.

The service may be initiated by entering the command **serviceconf** from a terminal window or by selecting the KDE (GNOME) menu – System Settings – Services.

After the service has been checked, the setting must be **saved** and **restarted**. Note that you must set the **RunLevel** for both levels 3 and 5 to make it totally functional.

10.2.2.3 linuxconf

Alternatively, we can issue the command:

$ **linuxconf**

to bring up the full administrative mode.

Select the following:

1. **Control**
2. **Control Panel**
3. **Control service activity**

You can now scroll through the list to view the status of each service. By hitting the ENTER key for a specified service, you can select Enable, Start or Restart a selected service. Additionally, a description of the service and the package name is also provided.

Finally the last option is when you are in the X Windows process. From a terminal window, type in:

$ control-panel

This will open a vertical window of several icons. One of the icons will be "Run Levels". Click on it and it will open another panel that displays all of the run levels and the services that are running in each. From a list on the left, you can choose a service that you wish to add, remove, start or stop.

This is probably the best option to use as you can modify any of the levels as desired as visually indicated and you can modify any of the levels in an easy method.

linuxconf is fairly old, but provides many functions. Unfortunately it is becoming outdated and is slowly being depreciated. Hence it may not be available when you install. If it is not available, first look on the vendor CDs under the RPM directory, or download it from the Internet.

10.2.3 Service Troubleshooting

If a service fails to activate, or does not function properly, then you need to figure out what you did wrong. The best resource is to observe possible errors in the /var/log/messages file. At the end of the file you should observe when a service was restarted, and possible errors if such exist. From whatever directory you are in, issue the command:

$ cat /var/log/messages

You will have displayed the last 24 lines of the file that you can observe for possible errors.

10.3 Server Requirements

What services are required are dependent upon what you need to do. If you are at home and you want to have a minimum system, your requirements will be small. Alternatively, if you are responsible for a major business, you might be responsible for a number of servers that support the business on the Internet and the employees.

Fundamental server requirements might include:

Backup	FTP	Telnet
Database	Mail	TFTP
DHCP	NFS	Web
DNS	Samba	SSH

Those servers that are normally seen by the Internet would include FTP, Mail, Telnet, and Web, the others would reside behind the protection of a firewall for use by the business. Although we have specified a number of servers, not all are necessary – it depends upon the business requirements. Note that setting up any host as a Telnet server is asking for disaster – opening yourself as a security risk!

Setting up servers for a business is definitely an individual evaluation. A given system may support one or all server functions. This is dependent upon how much traffic a given server function must support. For certain functions, such as web server, we might create a cluster of servers all doing the same service in order to distribute the load and serve the customer faster.

We have noted earlier that systems may either have a static or dynamic address. As a general rule if we elect to utilize dynamic addressing, we will address only user stations. Servers will always be assigned a static address.

Although not normally considered as a server, a router is in fact a computer running a variation of the Unix OS set up with multiple NIC cards and serial interfaces. Its function is to provide routing services for different information. Surprisingly, the power requirements of a basic router are quite low, a 386 processor is capable of supporting multiple DS-1 serial connections.

A topic that is extremely importance, yet often looked at last, is security. Various server functions exist, but if not necessary, should not be activated. A server function that should not be activated unless absolutely needed is Telnet, as this is an open invitation to crackers to gain access to your system and disrupt your business. The second server on the list is FTP. This system is also vulnerable to attack and should be set up with high restrictions for access. Issues of security will be discussed during the setup of each server in later labs.

10.4 Network File Server

The **Network File System,** or **NFS,** was the original means that ***Sun Microsystems*** developed in the 1980 to allow various Unix directory systems to be shared across a network. NFS utilizes the service port 2049.

It is an excellent method of making another host's file system, or part there of, to appear if it were your local drive. NFS is capable of supporting only a Unix / Linux based system. For support on the Windows environment one must use Samba. Note that the use of NFS is basically an insecure environment, as there are no security properties once the connection is established.

The advantages of this method are that we can set up one server to support files for multiple users, or maintain data on a remote system for security. The disadvantage is that all transactions with this file require data to flow over the network, thereby add to the general network load.

10.4.1 NFS Background Processes

To provide NFS services from any host, three processes must be utilized:

rpc.portmapper
: Maps calls made from another host to the correct NFS daemons.

rpc.nfsd
: Translates NFS request into actual request on the local file system.

rpc.mountd
Used to mount and umount file systems.

These three programs are normally installed and loaded at boot time. To check if they have been installed, issue the command:

$ rpcinfo –p

You should have an output that appears something like the following:

program	***vers***	***proto***	***port***	
100000	***2***	***tcp***	***111***	***portmapper***
100000	***2***	***udp***	***111***	***portmapper***
100005	***1***	***udp***	***821***	***mountd***
100005	***1***	***tcp***	***823***	***mountd***
100003	***2***	***udp***	***2049***	***nfs***
100003	***2***	***tcp***	***2049***	***nfs***

Specifically, we are looking for the mountd and the nfs entries.
These provide all of the registered RPC programs running on the local host. To check if RPC programs are running on a remote host, use the command:

$ rpcinfo -p *hostname*

If NFS is not installed on your system, it needs installation, two options are available for installation. The first option is to mount the CDROM and change to the **/mnt/cdrom/RedHat/RPMS** directory. Issue the following command:

$ rpm -ivh nfs-serv.rpm

Using the second method is easier to install NFS using **yum**. From the CLI, enter the following command:

$ yum –y install nfs*

This will install the NFS server and all dependencies.

10.4.2 Server Setup

A server will normally be set up to allow a specific set of directories to be seen by other Unix / Linux systems. The administrator will designate an existing directory, or may create a new one, as we will do in the Lab practice.
Lets assume that you want to share an existing directory where all data is stored, and that it is **/nfs/data**. On the server, the **/etc/exports** file is used to specify what directories are to be shared. It is set up with the following line:

/directory/to/export host1*(permissions)*
host2*(permissions)*

Multiple directories may be set up in the file, each on a separate line, with many other remote hosts set up to have connectivity.
The **/directory/to/export** is the absolute path name of the directory that is to be seen by other hosts. If the line ends up being longer than the standard lines permits, you can use the standard continuation character (the backslash - \) to continue on the next line.
In our example above, the first part of the line will be: **/nfs/data**
A host may be specified in one of four ways:

1. By hostname (requires a hosts file entry)
2. **@group,** where group is the specific network group (requires a group to be established)
3. Wildcards in the hostname. This way a group of users will all have access to the same directory. For example one might have ***.department.ourlab.com**, so each member of a department would see the remote directory.
4. By using an IP address range, which sets up the initial IP address and establishes a range by use of a subnet mask such as 192.168.102.0/255.255.255.0 .
5. Alternatively, the range may be listed as 192.168.102.1/24. This format is limited in range because there may be conflicts in the first available address and the sub-network mask. (The /24 format does not work when using RH 6.0.)

The permissions are optional, several of which are:

rw	Read and Write access
ro	Read only access
noaccess	Denies access to all subdirectories below the listed directory.
no_root-squash	Acknowledge and trust the client's root account

By default, if no options are specified, then the “ro” option will be applied. As a general rule, one should specify the “ro” permission to prevent a warning message.

To review the full options available, review the **info exports** page on your Linux system.

This will allow everyone in the class to mount your directory from the server.

Alternatively, we could also use the following lines in the exports file:

/example 192.168.102.{stn-id}
this limits access to one station

/example 192.168.102.{stn-id}(ro)
this limits station to read only

/example 192.168.102.0/255.255.255.0(rw)
this allows all stations on the 192.168.102 network to read and write to a file

If we have entered the following into our **/etc/hosts** file the line:

“**192.168.102.149 prof**”**,**

then we might also use:

/nfs/data prof

If we wish to set up two users and specify different permissions for both,, we could use the following line:

/nfs/data prof(rw), student(ro)

After the **exports** file has been modified, it needs to be reread. Issue the command:

exportfs –av (do not put a space between the a and v)

This generates a signal to the rpc.nfsd and rpc.montd daemons to reread the **/etc/exports** file and update the internal tables and provides a display of what was done. It also modifies the **/var/lib/nfs/xtab** (Red Hat version 6 and earlier) or **/var/lib/nfs/etab** (Red Hat version 7 and later).

(Red Hat does not quite follow the standard here, but they do have symbolic links that points from /etc/init.d to /etc/rc.d/init.d.)

Finally, we need to make sure that remote users are allowed access. View the file **hosts.deny** and verify if any specific user or group is restricted from access. In particular, confirm that the line:

ALL:=ALL

does **not** exist. If it does, remove it! By default, it should not exist, but we need to make sure. With this file, we could deny any other specific or group of users. (This might be of special use for certain users on your network.)

Even though we have restarted the server, it still is not activated so that remote users may access it. Several actions are required to activate or update the nfs service.

Now we must verify that the nfsd server daemon is active. Issue the command:

$ chkconfig --list | grep nfs

If the service is not active (levels 3, 4, and 5), that is they should indicate an "on" condition. If the condition is "off" (as it should be by default), issue the command:

$ chkconfig nfs on

Now the system has been configured to be a NFS server, but there are two more things to do. When the system booted, it read the activity status for each server (chkconfig) and found that it was configured for nfs to be off, so now we need to reread the server configurations. To activate them, we need to issue the command:

$ xinetd

Last – but not least – you need to again restart the nfs service. Issue the command:

/etc/init.d/nfs stop
/etc/init.d/nfs start or alternatively
/etc/init.d/nfs restart or
service nfs restart

10.4.3 Troubleshooting NFS

Sometimes the installation of a server does not go as smoothly as desired. Then one needs to perform a little troubleshooting. Here are a few suggestions.

10.4.3.1 Red Hat 6 and before:

Some times NFS does not work when we expect that it should. This is observed during the Client Setup. If this condition is seen, change to the **/var/lib/nfs** directory. There you will find the **xtab** file. Before you make a change to the exportfs (assuming no other changes have been make), this file

should contain the desired path and users. If something else is found, delete the inappropriate lines.

10.4.3.2 Red Hat 7 and later:

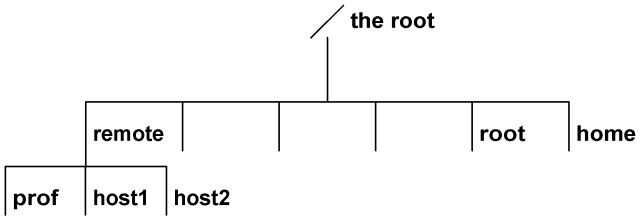
The same applies as just noted, except there was a change in the file that was written to. Now you will find a new file, **etab**, that is used instead of xtab. Otherwise the concept is identical. You will note that the xtab file still exists, but it is not used.

Note: For Red Hat 7.2 to function as an NFS server, the system must be rebooted – one of the very few cases.

10.4.4 Client Setup

In order to complete our example, we need to set up a directory where we want to have the remote directories reside. Setting up the directory in this format allows the Host user to keep the directory structure in a simple and traceable format. Issue the following command:

$ mkdir /remote



After you have created the remote directory, it is highly recommended that you create sub-directories – one for each nfs server. This will allow you to unmount a specific remote nfs server if necessary. This point has been learned from experience – in order to unmount a remote server, it must be its own directory; if you mount several remote servers in the same directory, you will not be able to unmount any of them.

In order for the NFS directory to be mounted on a local client, the mount command must be issued. We need to issue the command:

mount servername:/exported/directory dir/to/mount/to

For our example, assume we have the directory **/nfs/data/** on the **prof** host and we want to mount it on your system under the **/remote** directory, then issue the commands:

mkdir /remote/prof
mount prof:/nfs/data /remote/prof

Note that we must provide the full directory path for both the server and local directories.

As an additional verification that your system is working properly as a server, mount your own directory under the remote directory. For example, create a directory:

mkdir /remote/self

Then issue the command:

mount {your-IP-Address}:/nfs/data /remote/self

You should now be able to change to the directory:

cd /remote/self
ls

If everything was done correctly, you would now be able to see the contents of your own /nfs/data directory on your system under the /remote/self directory. If all is working correctly, then you have configured your system as an nfs server, and other Unix / Linux systems will be able to access your system.

To unmount the directory, you would go to the /data directory and issue the command:

umount {servername}:/{mounted-directory}

where ***servername*** is the nfs server host name (or IP address), and ***mounted-directory*** is the directory path that was previously mounted. You are not able to unmount the nfs directory if you are within it. For our example, you would issue the command:

umount /remote/self

If you now change to the /remote/prof directory and do a listing, you should see nothing. Again do the remote mount and confirm that you are able to see the files and directories.

This allows us to have the remote system appear as ours, but we do not want to go through the process every time we boot up. Linux uses a special file to automatically load the files at bootup. This is the **/etc/fstab** file. The format of the data in the **fstab** file is as follows:

/dev/device /dir/to/mount ftype parameters fs_freq fs_passno

These are:

/dev/device	The device or directory to be mounted. For a NFS system, it is in the form **servername:/dir/exported**.
/dir/to/mount	The location at which the file system is to be mounted on the local host.
ftype	The file system type. For a NFS mount, it will be **nfs.**
parameters	These are the parameters needed to mount the device or directory. Typical parameters include **rw, intr,** and **bg.**
fs-freq	This is used by the dump to determine whether a file system needs to be dumped.
fs_passno	This is used by the **fsck** program to determine the order to check disks at boot time.

As an example, append to the **fstab** to include:

prof /example /remote nfs rw 0 0

We now have set up the workstation to automatically connect to the NFS server during its boot process and to add the remote directory. Save and close file

10.4.4.1 Client Problem

Note that there is a special problem that exists in a NFS configuration when two or more users access the same file. If only one person has the file open, then they can write their information to the file and close it. If two or more have the file open, after the first person modifies the file and saves it, the second person modifies the version that they have and saves it – hence overwriting the previously saved file and therefore erasing the changes made to it. There is no method under NFS to lock a file.

There exists another problem during the shutdown of your system. If the server has been shut down prior to you shutting down, your system will hang in an attempt to disconnect the nfs service. The only solution is to just power down your system. For this reason, it is not a good idea to shut down an nfs server unless absolutely necessary.

10.5 Samba Server[1]

In order to allow Linux and MS Windows to work together, a Linux system must be configured to accept the MS **Server Message Block (SMB)**, which MS uses to support NetBIOS. SMB is the protocol that IBM and Microsoft created for NetBIOS so that one host would be able to see other hosts from the Network Neighborhood. It provides a request – response protocol, so it can be quite "talky" over a network. The netbios protocol utilizes port 137, 138 and 139.

After a Linux system has been set up with Samba, it can operate as a file server, print server – even as a Primary Domain Server or Backup Domain Server. It integrates the Linux system into Microsoft / PC network architecture.

The process and code was first developed by Andrew Tridell in Australia in an attempt to link his wife's windows system to his Linux computer. To do this he wrote a packet sniffer and reverse engineered the MS smb process. The use of "smb" was copyrighted, so Andrew did a dictionary search and came up with the term **samba –** which has no special meaning.

10.5.1 smb Protocol

smb is a protocol originally defined by IBM and subsequently enhanced by Microsoft in order to establish a local network architecture for interconnecting various Hosts. It is a client-server, request-response protocol. Clients connect to servers using NetBIOS over TCP/IP as specified in RFC –1001 and RFC – 1002. Once a connection has been established, smb commands can be sent to the server, allowing access to shares, files and printers. Note that NetBIOS was originally used by IBM and Microsoft for just the local network – it did not require TCP/IP.

Logging onto a server is a process of authentication. On request for services, the client requests access, is challenged, and if the password is correct, is issued a User ID (UID). All subsequent requests for information to the server must include the UID in order to be authenticated for access.

The first part is to allow the MS Windows systems access to Unix/Linux directories and files.

---

[1] ***SAMS Teach Yourself Samba in 24 Hours*** by Gerald Carter, **Sams / Macmillan Publisbing,** ISBN 0-672-31609-9

First we need to test your system to determine if SAMBA has been installed during the initial installation. If you performed a full installation, then the installation process is not required. Issue the command:

$ rpm –q samba

If it is not installed, perform the installation with the command (SAMBA is located on the CDROM/RedHat/RPMS):

$ rpm –ivh samba(version number).i386.rpm

Several directories and files are created during the installation:

/usr/sbin/smbd Samba client connections
/usr/sbin/nmbd NetBIOS server
/usr/bin/smbclient Client to access SMB server
etc/samba/smb.conf Samba configuration file

To implify the installation, issue the command:

$ yum –y install samba*

In preparation for our example, and to demonstrate how the system works, we need to create some directories, users, and a group.

10.5.2 Users

First, to demonstrate the process, we will create two new users on the system, such as **prof** and **instructor.** Display the **/etc/passwd** file to observe that they are now installed. Note that the field for the password is **:!!:** (or 'x' if you have a shadow file set up), representing that a password has not yet been created. Now generate a password for each as **ourlab** by issuing the command:

$ passwd prof and
$ passwd instructor

In the **/etc/group** file, we will find a list of users and groups that have been established by the system. Creating a new user is automatically added to the list with a group id in the 500 range. There should be two lines just added that are something like:

prof:x:501 and
instructor:x:502

The "x" means that the password is contained in the shadow passwd file.

10.5.3 Groups

Although it is not necessary to create a group for access to a directory that is to be shared to a MS Windows system, it is administratively a better approach. By using this method, if you, as the administrator, need to add or delete a user from access, you only need to modify the group and not the smb configuration (which we will get into shortly).

First we need to manually edit the group file to add a new group - there are two methods that we can use to add a group. The first method is to use one of the commands:

$ groupadd class or
$ groupadd –g 2001 class

The first option says to add a new group with the name of class, where the system will generate its own group number. The second option allows you to specify the group number that you want to use. Otherwise they are identical.

At this time, we have the users prof and instructor as users, but later we will be limiting access to our directory to the group called ***class***. We must populate the group class. The proper way of doing this is to issue the command:

$ usermod –G class prof and
$ usermod –G class instructor

The second alternative to using the moduser command is to manually edit the **/etc/group** file and add the user group along with the group members. Using **vi** or **pico,** go to the end of the file and add the line:

class:x:2001:prof, instructor

Notice that in editing the file manually we can automatically add the users with the same entry process, but if we use the **groupadd** command, then we will still need to manually add users to the group.

The **:x:** means that there is no password for the group, which is the preferred case (why would one have a password to be a member of a group unless you are in a very secure environment?).

10.5.4 Directory to be Shared

Now we wish to create a test directory that will be set up as a shared resource. From the **root** ("*/*")directory, issue the commands:

```
$ mkdir samba
$ cd samba
$ mkdir smb
$ cd smb
$ mkdir students
```

Note that for this example, we have created a new directory, but we would normally set up our system to share an existing directory.

Finally, we want to create a temporary file in the students directory to prove that we got there and to confirm where we are. Issue the command:

```
$ cd students
$ nano myfile
```

Create the following text:

```
Name:       {your name}
Station:    {station id}
IP Address: {192.168.102.stnid}
```

Save the file and close.

This is an example for the class setup, you will naturally set up something different on a business system.

Now we want to set up the **students** directory to be shareable to other systems. To fully understand what is happening, we will issue additional commands so that we can follow what is happening. Change to the

/samba/smb directory and monitor what happens. We need to issue the following commands:

$ ls –l
drwxr-xr-x n root root size date time filename
$ chgrp class /samba/smb/students
$ ls –l
drwxr-xr-x n root class size date time filename

This makes the directory **students** a member of the group **class,** did you notice the difference.

Now issue the command:

$ chmod 0770 /samba/smb/students
$ ls –l
drwxrwx- - - n root class size date time filename

Do you see the difference in the line from the previous printout?

This changes the access rights of the directory **students** as owner full access (rwx), group full access (rwx), and the rest of the world nothing (0). Now issue the command:

$ chmod g+s /samba/smb/students
$ ls –l
drwxrws- - - ***n root class size date time filename***

This sets the group for the directory **students** as shareable.

10.5.5 Modifying the smb Configuration File

For protection, we need to backup the file that was created during the installation, the **/etc/samba/smb.conf** file. Issue the command:

$ cp smb.conf smb.conf.bak

Now we need to define several elements of our computer so that it may be identified to the Windows environment.

First we need to really get into the meat of the configuration of the **smb.conf** file – some definitions. You are referenced to the **info smb.conf** manual pages for more detail. These will provide additional insight as to the options that are available in the configuration of the file. Two that we are particularly interested in are:

The following are a few variables that will be used in the configuration:

%u user name of the current service (logged in user)
%H home directory of the user given by %u
%S sets up sharable user set

10.5.5.1 Global Section

We need to set up some variables that we will use in the configuration of the smb.conf file.

netbios name = eagle{stn-id}
workgroup = nest

The first represents the name that we are going to give to the computer. In the Windows NetBIOS system, every computer is provided a unique name – no two host computers can have the same name. In this class example, we are

adding our station id to the eagle name in order to make sure we are each different. This line is not presently included in the existing file and will have to be added. Setting NetBIOS name is an option, if not used the system will use the hostname.

Second we are setting up our workgroup as nest. This is the same as a workgroup of accounting or engineering. Each department in a company is typically setup as a separate workgroup and the members of each department are appropriately assigned. For ease in the class, let's just create one workgroup.

The last variable sets the security of the system as to that of the user for Windows environment. This value should already be in the default configuration.

We must insure that the **smb.conf** file is properly configured. This file is set up in a similar manner as an MS Windows **-.ini** file. Note to monitor the changes that you are being told to do - other changes may cause your system to lock up. The first section should appear as ("#" lines are commented out):

```
[global]
  netbios name = eagle{stn-id}          (you will need to add this line)
  printing = bsd
  printcap = /etc/printcap load
  load printers = yes
  guest account = nobody
  lock directory = /var/lock/samba
  share modes = yes
  workgroup = nest                      (must be set to local workgroup))
```

Most of the lines already exist, and may be spread over quite a few other details.

Further down the file you will find two lines (in previous releases these were remarked out (**;**)). If the two lines are remarked out, then we will be set up to transmit passwords across the network in a clear text mode – but this is a bad practice! If the following two lines are remarked out, remove the semi-colon so that the statements will be active.

```
encrypt password = yes
smb passwd file = /etc/smbpasswd          Red Hat release 6
smb passwd file = /etc/samba/smbpasswd    Red Hat release 7
```

The default installation of smb file includes the following active lines:

```
log file
max log
security = user                       (sets local user security)
socket options
dns proxy = no
```

There are many other options available, most of which are commented out. These provide the administrator the ability to set up the system to serve different functions, such as a Primary Domain (i.e. – you don't need NT).

10.5.5.2 Homes Section

The second section sets up access rights, it should be set up as:

```
[homes]
```

```
   comment = Linux Home Directories    {modify this line}
   browseable = yes                    {change to allow it users to browser}
   path = %H                           {add}
   writeable = yes                     {add}
   valid users = %S                    {add}
   locking = no                              {add}
#  readonly = no                       {remark out}
#  preserve case = yes                       {remark out}
#  short preserve case = yes                 {remark out}
   create mode = 0750
   directory mode = 0770               {add}
```

Make sure everything is spelled correctly, otherwise errors will occur.

10.5.5.3 Printer Section

The third section sets up the printers as:

```
[printers]
   comment = All Printers
   path = /var/spool/samba
   browseable = no
   printable = yes
#  public = no
   writeable = no
#  create mode = 0700
   guest ok = no
```

No modifications should be necessary.

10.5.5.4 Public Section

The forth section sets up a directory being shared. For our initial example, we will comment it out. Many examples are provided for user's review.

```
#[public]
#  path = /path/directory     {the directory you want to set up as sharable}
#  public = yes
#  readonly = yes
#  printable = no
```

This section should be fully remarked out, and is typically shown as an example for setting up your system.

10.5.5.5 Temp Section

Another shared directory providing full rights may be set up as – it may be commented out:

```
;[temp]
;  path = /tmp                      {the path and directory you want}
;  readonly = yes
;  public   yes
```

This section should be fully remarked out, and is typically shown as an example for setting up your system.

10.5.5.6 Private Section

To set up a private access directory, you would use the following. Again it is commented out.

```
;[private]
;  path = /private/directory          {the path and directory you want}
;  valid users= username 1   usename2   username3
;  public = no
;  writeable = yes
;  printable = no
;  create mask = 0765
```

This section should be fully remarked out, and is typically shown as an example for setting up your system.

Other sections may exist, and are typically commented out (have **;** or **#** in front of the line). They are set up as examples that you may utilize for your own purposes.

10.5.5.7 Class Public Directory

Let's create a new section for our directory, we need to add the following lines:

```
# Class setup
[Eagle's Nest]
  comment = Shared Directory
  path = /samba/smb/students
  writeable = yes
  browseable = yes
  valid users = @class          {the @ means it is a group}
  locking = no                  {for a database, only one user may
                                access a file at a time}
 create mode = 0770             {User may create a file with read, write,
                                and execute properties for owner and
                                group}
 directory mode = 0770          {User may create a directory with read,
                                write, and execute properties for owner
                                and group}
```

The above sets the path to the shared directory, limiting access to only the members to the group that have rights.

Lets review what we have entered:

# Class setup	just a comment for our reference
[Eagle's Nest]	denotes the beginning of our section, and is the name that appears on a MS Windows system
comment	an ignored line – a comment
writeable	specifies whether the files may be written to
browseable	specifies whether the files may be viewed

valid users	specifies either users or groups that may access
locking	specifies if multiple users may access
create mode	if a file is created, specifies the attributes
directory mode	if a directory is created, specifies the attributes

We are now finished modifying the file**.** Save the file and close.

When you created the new section, [Eagle's Nest], what appears in the MS Windows network Neighborhood is "Eagle's Nest" for the shared folder. There is one limitation, this directory name has a maximum length of 13 characters. Any more than that and the remainder will be truncated, and the server access will be denied.

10.5.6 Testing Configuration

As part of the process, there is a procedure that allows us to test the smb.conf file. Issue the command:

$ testparm smb.conf

If all went correctly, we will get a listing noting the sections that were added and the values. If there were errors, you will note that portions of the file were ignored. By observing the comments, you should be able to determine where the errors might be and where to look. Most errors will likely be typos. Some of these are hard to see on a first look, but on intense review one can find them. The output of the testparm listing does not point to the exact error, but does provide an excellent pointer to the problem. Finding the problem often takes experience and practice – it is not easy!

10.5.7 Active Ports

If there are no errors in the smb.conf file, we are fundamentally done with the configuration. All that is left is to make sure the service is activated and updated. But prior to doing this, we want to first make sure that the ports are activated.

Next we need to check out the **/etc/services** file to insure the following lines exist (if not - add them – by default, they should all be available):

```
netbios-ns    137/tcp
netbios-ns    137/udp
netbios-dgm   138/tcp
netbios-dgm   138/udp
netbios-ssn   139/tcp
netbios-ssn   139/udp
```

These will allow the smb protocol packets to be recognized by the Linux Samba Server.

If you originally installed Samba during the installation, is probably already operational with the original configuration. You can verify its operation by looking for the process id. Issue the command:

$ ps –aux | grep smbd and
$ ps –aux | grep nmbd

Look for the smbd and nmbd processes. If they exist, then we need to stop the process and re-start it.

10.5.8 Creating Samba Password File

Microsoft encrypts a user's password in a different way from Unix or Linux. Because of this, a separate file is required to maintain the MS encrypted passwords – this file is called **smbpasswd**. Initially, the **smbpasswd** does not exit, so we will use a special program called a **script** that will perform a specific function. In this case, we need to convert our **passwd** file to the **smbpasswd** file.

Now issue the command:

10.5.8.1 Red Hat 6 smbpasswd File

$ cat /etc/passwd | mksmbpasswd.sh > /etc/smbpasswd

10.5.8.2 Red Hat 7 (and later) smbpasswd File

$ cat /etc/passwd | mksmbpasswd.sh > /etc/samba/smbpasswd

What we are doing is to take the results of the concatenate (cat) function and **pipe** (pass through) it to the script program **mksmbpasswd.sh**. The result is then directed to a file called **smbpasswd** rather than to the screen. This creates the file, but does not set the password for any individual user.

View the **passwd** or shadow file – observe that the encrypted passwords have been exists and that they are different. Now look at the **/etc/samba/smbpasswd** file – observe that the passwords for ***prof*** and ***instructor*** are a series of "xxxxx's". The user exists but the password has not yet been entered.

10.5.8.3 smbpasswd Security

Now we need to change the attributes of this file so that only the root administrator has access. Issue the commands:

We need to insure that only the system administrator has the rights to read or modify the smbpasswd file. Perform the following if necessary.

$ ls –l | grep smbpass*
- r w x r - x r - x n root root size time date smbpasswd
$ chmod 600 /etc/smbpasswd
$ ls –l | grep smbpass*
-rw - - - - - - - n root root size time date smbpasswd

The above is normally the default to insure that only the administrator, but must be verified to insure system security.

10.5.8.4 User smb Password

We now need to generate the passwords for our two users. Issue the commands:

$ smbpasswd prof and
$ smbpasswd instructor

For each, issue the password of **ourlab**.

10.5.8.5 MS Password Identity

Finally, we need to again look at the **smbpasswd** file and observe the encrypted passwords for both ***prof*** and ***instructor***. For each, set the password to **ourlab**. Now issue the following command:

```
$ cat   /etc/shadow | grep   prof
$ cat   /etc/shadow | grep   instructor
$ cat   /etc/samba/smbpasswd | grep   prof
$ cat   /etc/samba/smbpasswd | grep   instructor
```

Observe the differences between them – the passwords in the passwd file are different, whereas the passwords in the smbpasswd file are the same. This is when the password for each user is “ourlab”.

10.5.9 New Samba User

You later come upon the situation that you want to add a user to the **smbpasswd** file – you can not recreate the file with the above process because you would delete the previous users passwords (which you have naturally forgotten). To add a new user, issue the command:

```
$ smbpasswd   –a   {newusername}
```

After which you can apply the new password to the user with the command:

```
$ smbpasswd   {newusername}
```

10.5.10 Password Transmission

Microsoft Windows are designed inherently to transmit passwords across the network in an encrypted mode. As we have set up the smb.conf file, it is designed to pass the password in an encrypted text – so we are ok. Prior to version 7, the Red Hat default was to send the password in clear text – a definite conflict to the network operation. We have several options here, which will be discussed as appropriate.

10.5.10.1 First Method – Not Preferred

The first solution was originally used in the implementation of Samba, but is NOT recommended on any system except for a private system (and then it is VERY highly discouraged). This method is in accordance to the Microsoft TechNotes Q187228. Realize that if a password is transmitted in a clear text mode, then anyone with a network monitor is capable of collecting and recording all passwords. This is a very bad idea!

Within the Windows Registry is an entry for transmitting the password as clear text. This document does not specify the exact key for security reasons, but may be obtained from an MSCE.

10.5.10.2 Second Method – Preferred

As an alternative to modifying each Windows system (what if you have 100 users using Windows) is to modify the Linux system to accept encrypted passwords. THIS IS THE PREFERRED METHOD. By modifying the smb.conf file to accept encryption, you do it once and the passwords are transmitted across the network in an encrypted mode.

For Red Hat 6, if we had not originally set up the system for encryption, we would have to open the smb.conf file with pico and locate the line **encrypt passwords**. The default is set to **no.** Change this to **yes.** This line will probably be commented (**";**") out, remove the semicolon to enable the line. For Red Hat 7 and later, the default is for enabled encryption.

Now locate the line that contains **smb password file** (should be directly below the password line). This should be set to **/etc/smbpasswd** (**/etc/samba/smbpasswd** in Red Hat version 7 and later). This line is also probably commented out, remove the semicolon to enable this line.

10.5.11 Restarting Samba

Now we are almost done – but we still need to check that the service is activated.

10.5.11.1 Activating Samba Service

Issue the command:

$ chkconfig – – list | grep smb

Note that for Run Levels 3, 4 and 5, they need to be turned on. If they are not, issue the command:

$ chkconfig smb on then
$ chkconfig – – list | grep smb

and verify that the smb service has been set to on for the Run Levels.

Finally, issue the command:

$ xinetd

to re-read the information and activating the service.

10.5.11.2 Restarting SMB

As before, we need to reread the **smb.conf** file to accept the changes. Issue the commands:

$ /etc/rc.d/init.d/smb stop and
$ /etc/rc.d/init.d/smb start or the single command
$ /etc/rc.d/init.d/smb restart

Alternatively (because you want to do it faster), issue the command:

$ service smb restart

If you should wish to stop the Samba's smb, issue the command:

$ /etc/init.d/smb stop

10.5.12 MS Windows Setup

There is one last process that needs to be done - setting up Microsoft Windows to be able to see the files.

Before MS Windows can see the Unix / Linux system, it too must be configured. Quite often the following have already been set up, but you want to make sure of it.

10.5.12.1 MS Windows Sharing

We must make several modifications to the MS system in order for it to see the Linux Samba Server.

10.5.12.2 Network Neighborhood Configuration

First action is to set up MS Windows to be shareable.

Open Network Neighborhood Properties (right click Network Neighborhood – click on Properties).

Under the Network Neighborhood Properties Configuration Tab, select:

Primary Network Logon set to

Client for Microsoft Networks

Click the ***File and Print Sharing*** button and click the options for file and printer access.

Second, on the Identifier tab, select:

User Name to **prof** This needs to be set different on each host

Workgroup to **nest** This needs to be the same on all hosts

Third, on the Access Control tab, select:

Control Access set to

Share-level Access Control and

click **File and Print Sharing.**

10.5.12.3 MS Windows System Name

Lets make a special note of the Identifier tab. We are specifying it to prof for this example, if it is different, then we must add that user name to the Linux system and add it to our group that will have access to the desired directory. We must log onto the system as the user prof or instructor in this example.

Naturally, Windows will now require you to reboot the system before the changed configuration will take effect. (Maybe they will learn to upgrade the system so that it will not require rebooting.)

On rebooting, you will be required to login in order to log on to the system. You must log into the system with one of the usernames that you created on the Unix / Linux system – either **prof** or **instructor** and give the proper password which is that for the appropriate user in the Linux system. If you hit the cancel button - you will not have access to the network at all! Remember that the password for prof and instructor for this setup is "ourlab".

It is a requirement that the MS Windows system name be the user name in the smbpasswd file on each Samba Server, and that the MS users know the password entered on the Samba Server (it does not have to be the same as that used to log on from the MS system). It is also required that the MS username be a member of the group that was established.

Be prepared to wait up to 15 minutes before the Windows system may see the Linux computer(s). This is because the smb enacts an update only once every 15 minutes, so it may take a while. One way of improving the lookup speed is to do a search for a specific computer. Select find – search for computer, and then put in the Linux hostname (eaglexx).

10.5.13 It should be working "?"

Now we can go to the Windows Host and open up the Network Neighborhood to view the Linux systems that have Samba enabled. If all worked well, it works.

From an MS Windows system, we will be able to click to the Network Neighborhood, click on Entire Network, where we should now see the group called Nest. Upon clicking on Nest, we should see all of the Linux Samba computers that are available. If you are not on the MS Windows system, you need to investigate what errors have been made on your system – and correct them. At the MS PC, we can click on a specific Linux system and we will be given access to the shared directory (students) and the files that reside therein. At this point you should be able to see, and open the text file that you created earlier, **myfile.**

10.5.14 A Quick Test

We can perform a quick test of our configuration by entering the following command:

$ smbclient –L {your-IP-Address}

At the password request, hit ENTER. You will obtain a configuration of your system under samba.

Remember that this is an example of a setup, and will be different on a system at a business or at your home.

10.5.15 A GUI Interface for Configuring Samba

There may be an easier way to do all of this in the near future. Starting with RedHat 6.0, there is a graphical means of configuring the smb.conf file. Before we can use it, we need to do a little preliminary work.

In the **/etc/services** file, verify that the following line exists (generally it does):

swat 901/tcp #Add swat service used via inetd

Note that there are some great benefits to using SWAT, but some caution must be used. The original smb.conf file is great for examples. Unfortunately it is written over when SWAT saves the new configuration, therefore, make sure the original smb.conf file is backed up before using SWAT.

10.5.15.1 RedHat 6

Next, check the **/etc/inetd.conf** file for the following line. Add it if necessary.

swat stream tcp nowait.400 root /usr/sbin/swat swat

10.5.15.2 RedHat 7 and later

Create the file **/etc/xinetd.d/swat**, (if it does not exist):

```
service swat
(
        type        = INTERNAL
        id          = echo-stream
        socket_type = stream
        protocol    = tcp
        user        = root
        wait        = no
```

disable = no
)

Assuming that the Samba server has been previously configured as a web (http server), then you will be able to modify the Samba configuration using the SWAT utility. To access from a remote system, enter into the URL of a browser:
{Samba IP Address}:901

Finally, edit the **/etc/hosts.allow** file. Add the following line:
swat 127.0.0.1 192.168.102.
or swat: ALL
Before Swat can be utilized, since we have made changes to the inetd file, we need to restart it by generating a HUP signal. This causes the system to reread the inetd configuration file (the one just modified). Issue the command:
Now start the X Windows session and start **Konqueror**. At the URL enter:
http://192.168.102.{your-stn ip}:901/

This will start a program called **Swat.** From this you can select the various configurations for configuring your smb.conf file. What is notable is that if you have a question regarding the configuration of a specific parameter, you can obtain help directly for that specific item by clicking on the ***help*** pointer to the left of each option field.
On initiating Swat, you will need to log in and provide a password - this should be root and your local password.
Under Netscape, a screen will open with the letters ***samba*** at the top of the home page and seven buttons directly below it. These allow you to set configuration of the ***home, global, shares, printers, status, view,*** and ***password.***
Each of these may by modified as appropriate. The simple configuration is presented one the selected page, but you may make advanced options by clicking on the ***Advanced View*** button.
Before the new configuration will work, we need do a restart. Select the ***Status*** button. From here you need to stop and restart the services smbd and nmbd.

10.5.16 Troubleshooting

Although making Samba work is not a problem, it does require some experience to become proficient. Getting samba to work often runs into problems, make sure that you performed the following tasks:

1. Create user that will access samba.
2. Set password for the new user as a Linux user (not necessary for just samba).
3. Create a group that the user will be on.
4. Add the new user to the group.
5. Add / configure shared directory.
    a. Create directory structure.
    b. Specify group name to shared directory.
    c. Specify attributes of shared directory.
    d. Specify group share of shared directory.

6. Edit smb.conf file.
   a. Make smb.conf backup file.
   b. Create netbios name.
   c. Edit workgroup name.
   d. Verify encrypt password = yes and not commented out.
   e. Verify smb passwd file = /etc/samba/smbpasswd and not commented out.
   f. Create shared section.
      I. Create [Shared Name]
      II. Set path.
      III. Set writable
      IV. Set browseable
      V. Set valid users
      VI. Set locking
      VII. Set Create Mode
      VIII. Set Directory Mode
7. Run testparm to check for errors.
8. Restart smb.
9. Create / Update smbpasswd file.
   a. Create
      cat /etc/passwd | mksmbpasswd.sh > /etc/samba/smbpasswd
   b. Update
      I. New user.
         smbpasswd -a userid
      II. Existing user.
         smbpasswd userid

10.6 Telnet Server

For Red Hat 7 and later, by default the telnet server is not active. Earlier version the server was active – a major security risk.

10.6.1 Installing Telnet

If telnet must be installed, issue the following command:

$ yum –y install telnet*

As was demonstrated in an earlier lab, telnet allows a user on one host to log onto a different host from a remote location. This concept is very good, except that there is a major security risk – once one is logged onto the system, the remote user may modify or delete any file they wish. Therefore as a general rule, except for special cases, telnet should not be enabled. Telnet utilizes service port 23.

To enable telnet, issue the following commands:

1. **chkconfig – –list | grep telnet**
   Search the list for the value of the telnet agent – will typically be at the end of the file. Note that the default value for disable is yes.
2. **cd /etc/xinetd.d**

3. **nano telnet**
   a. The new structure of the xinetd files provides for additional security and features. One line in the list is **disable**. Red Hat by default has set the value to ***yes***, which will prevent your system from providing ftp services.
      i. Change the **disable** value to **no**.
      ii. Save and exit the file.
   b. Instead of editing the telnet file, issue the command:
      chkconfig telnet on
4. Again issue the **chkconfig – –list** command. Note that ftp has now been enable. Although the service was enabled, it is still not operational. For this, we need to restart the service.
5. In MS Windows tradition, you could reboot the computer. But Linux has a better way of doing it – just re-read the ***xinetd*** file and restart. Issue the command:
   xinetd

Users will now be able to telnet into your system to obtain files.

10.7 FTP Server[2,3]

Often we wish to set up a single server to allow others the ability to download or upload information via an **ftp** transaction. FTP utilizes service ports 20 and 21. The connection between the client and server is set up using port 21, and the data transfer uses port 20.

The server must be set up to accept three different types of users: real, guest and an anonymous login. The normal and guest users use a normal password that has been assigned to them, whereas the anonymous use logs using the username “anonymous” or “ftp” (either will work) and with the password normally being the user's email address. The password must be tested against collected information (the email address) prior to the login being successful. In our lab environment, the password may be anything other than a blank.

10.7.1 Installation

If you properly installed the appropriate packages during the installation process, the ftp server has already been set up. If not, then you need to install two software packages from the CD installation disk.

To see if the ftp services were set during installation, issue the command:

$ rpm –q ftp

If installed, you will observe the packages of vsftp and anonftp.

If it was not installed, mount the CD, then change directory to the CD. Do an **ls** command to display the contents of the CD, which should include the **RedHat** directory. Then do another **ls** to display the contents of the RedHat directory, which should include a RPMS directory.

Next, do an **ls vs*,** this should list a file something like

vsftpd-x.x-y.rpm where the x and y's are numbers

[2] Red Hat Linux 7 Server: Mohammed J. Kabir; M&T Books / IDG Books
[3] Red Hat Linux – The Complete Reference; ; Osborne / McGraw Hill

Later versions of Red Hat (Fedora) Linux use **vsftpd**.
Then do an **ls anon*,** this should list a file something like
anonftp-x.x-y.rpm where the x and y's are numbers

To install these files, issue the commands:
$ rpm –ivh vsftpd-x.x-y.rpm and
$ rpm –ivh anonftp-x.x-y.rpm

Make sure that you type the filename exactly as you found it when doing the ls command.
If the package was previously installed, it will not be reinstalled.
We need to make sure that remote users are allowed access. View the file **hosts.deny** for its contents. If there are any restrictions, note that such user or group will not be allowed access. Specifically we need to verify that the line:
ALL:=ALL

does not exist. If it does, remove it!
Basically, after the installation, no other configuration requirements exist for a fundamental system. With the exception of activating ftp, you are now ready to go.
The previous application, wu-ftpd, was the daemon originally used by Red Hat. This application has been changed to **vsftpd**. The process of installation and everything else is virtually identical.
If vsftpd is not installed, then it can be installed using the command:
$ yum –y install vftpd*

An even more secure ftp server application is **proftp**. If not available on the installation CD's, it may be downloaded off of the Internet.
For the rest of this section, we will refer to **vsftpd**.

10.7.2 Server Activation

To activate the ftp process, we first need to check if it is active. Issue the command:
$ chkconfig – – list | grep ftp

you should obtain:
vsftpd 1: off 2: off 3: off 4: off 5: off 6: off

representing the six run levels and whether the ftp server daemon is operational. We need to make it active, so issue the command:
$ chkconfig vsftpd on

If you again do the chkconfig listing for ftp, you will observe that run levels 3, 4, and 5 are now “on”.
To test out the operation, you can ftp to yourself to see the operation. Perform the following:
$ ftp 192.168.102.(station-ip)

You will see a greeting statement from your host and a request for a username, type in **anonymous.**

Next you will be requested for a password, nearly anything will be accepted, but you should type in your email address.

You are now at the directory of **/var/ftp/**. If you issue an **ls** (directory list) command, you will see a list of files that are at that location which may be downloaded. Since you are presently on your own machine, there is no need to transfer a file.

Make sure that there is a file on that directory that others may download for practice.

An additional note, vsftp is not the most secure system. A better FTP server agent is “Pro-FTP”. This software provides enhanced security features. At this time, installation of it is not being reviewed. Other ftp server agents also exist.

Note that current versions of Fedora use the vsftpd daemon to provide the FTP service.

10.7.3 FTP Users

An FTP Server may be set up to support three classes of users – real, guest, and anonymous.

10.7.3.1 Real User

Real users access an FTP server to their own home directory. They have full rights to read and write to the directory. With vsftp, they may also move to any other directory on the system where they are allowed. By default, vsftp considers any named user as real. If using pro-ftp, the reverse is true, a specific named user must be allocated real user rights.

The setup and configuration of vsftp is specified in six files, their functions are presented below, and then the specifics for setting up an anonymous and guest user.

10.7.3.1.1 /etc/services

This defines the two ports used for ftp services (and all of the other services), namely ports 20 and 21. **xinetd** uses these values to know which ports are assigned to ftp.

10.7.3.1.2 /etc/xinetd.d/wu-ftpd (/etc/xinetd.d/vs-ftpd)

As previously noted, this file defines the line for ***disable*** is set to **no**, otherwise the service is not functional. After changes have been made, we need to re-read the file, issue the command **xinetd**.

10.7.3.1.3. /etc/ftpaccess

This is the primary configuration file for the ftp service. It consists of various command lines of the format:

Keyword option(s)

We will review the various keywords with their syntax, and their function.

1. **class all real, guest, anonymous**
   This specifies the class name and which users (real, guest, or anonymous) are members of it. A new line for each class name is

made in the file. In this example (default), all users are part of the class *all*.

User real: A user that has a valid userid in the password file.

User guest: A user who logs in as real user but is provided given guest privileges.

User anonymous: A user who logs in as anonymous and has very limited privileges.

2. **deny addrglob message_file**
   This specifies the group of users that are denied service and the message that is issued to them when they attempt to log in.
3. **limit class n times message_file**
   This limits the number of simultaneous users that may log into the server, what days they log in, and the message file that they receive if too many attempt to log in.
4. **noretrieve file_list**
   This lists files that are not allowed to be downloaded.
5. **loginfails n**
   Specifies the number of times a user may attempt to log in before the connection is terminated.
6. **private yes / no**
   Specifies whether an extended list of commands is allowed to be issued. Due to security reasons, the default should be *no*.
7. **guestgroup groupname(s)**
   This is a list of groups that are considered as guest group accounts.
8. **autogroup groupname group(s)**
   This is a list of groups that have their groupid modified if the user belongs to a valid group.
9. **banner filename**
   A file that displays a message prior to login.
10. **email user@host**
    A directive for the email address of the administrator of the server.
11. **message path/message_file [when class(s)]**
    Path and message file when a user successfully logs in.
12. **readme path/filename [when class(s)]**
    Displays the existence and date attributes for the specified filename.
13. **log transfers file-list directions**
    Specifies which files that are transferred either into or out of the server are logged.
14. **log commands command list**
    Specifies commands that are logged.

15. **chmod yes / no userlist**
    Specifies if a real, guest, or anonymous user may modify file attributes. Default should be ***no***.
16. **delete yes / no userlist**
    Specifies if a real, guest, or anonymous user may delete a file. Default should be ***no***.
17. **overwrite yes / no userlist**
    Specifies if a real, guest, or anonymous user may overwrite a file. Default should be ***no***.
18. **rename yes / no userlist**
    Specifies if a real, guest, or anonymous user may rename a file. Default should be ***no***.
19. **passwd-check none / trivial / rfc822 (enforce / warn)**
    Specifies the type of password required for an anonymous user and if the rule is to be enforced or just given a warning. The default should be to enforce the rfc822 rule.
20. **upload user-home-directory upload-directory yes / no owner group-name file-attributes (nodirs)**
    Specifies if a user is allowed to upload files to the upload-directory. Default attribute is 0600. If a use is not permitted to create a directory, the option (nodirs) must be specified.
21. **alias string dir**
    Sets up an alias name for a (long) path-directory.
22. **chpath dir**
    Specifies another path-directory that is not in normal home directory that a user may change to.
23. **compress yes / no class(es)**
    Specifies whether various user class(es) are allowed to compress files.
24. **tar yes / no class(es)**
    Specifies whether various user class(es) are allowed to tar files.

10.7.3.1.4 /etc/ftpconversions

This file specifies the FTP server's conversion database. Changes are normally not required.

10.7.3.1.5 /etc/ftpgroups

This file is only necessary if the non-standard **SITE** commands are permitted. Due to security reasons, the administrator should not have to worry about this file.

10.7.3.1.6 /etc/ftphosts

Controls FTP access to specific accounts from various hosts. This is set up to allow a user to log in to the FTP server from various hosts on the same LAN.

10.7.3.2 Guest User Setup

Setting up a Guest User is a combination of a real user and anonymous user. After creating a user, perform the following modifications to convert them to a guest user.

1. Copy the **/var/ftp/bin, /etc, /lib, and /pub** directories to the **/home/{guestuser}** directory. Use the command:
   cp –r /var/ftp/bin /home/{guestuser}/
   The –r specifies that the directory (/bin) and its contents are to be copied to the guest users directory. Do this for all four directories.
2. Edit the **/home/{guestuser}/etc/passwd** file for the following:
   a. Remove the line for the ftp user.
   b. Copy the ftp user line from the **/etc/passwd** file to the **/home/{guestuser}/etc/passwd** file. Then make it look like the following:
   {guestuser}:*:{guestid}:{guestgrpid}:/home/{guestuser}/./:/bin/true
   (guestid = 14, guestgrpid = 50?)
3. Change the owner and group of the **/home/{guestuser} to {guestuser}**.
4. Change the directory attributes for **/home/{guestuser}** to 750.
5. Change the owner and group of **/home/{guestuser}/bin, /etc, /lib, and /pub** to **root. root** (chown –R root. Root *)
6. Change the bin, etc, and lib directories to execute only (chmod –R 111 *).
7. In the etc, bin, and lib directories, changes the owner and group name to root.
8. In the bin, etc, and lib directories, change the first attributes to read only (444).
9. Edit the **/etc/ftpaccess** file to add the **{guestuser}** as a guest group. The group name for **{guestuser}** is identical to the guest user name. Add the line:
   guest group {guest name}
10. If you wish to allow the **{guestuser}** to upload files, do the following:
    a. Create directory **/home/{guestuser}/incoming**
    b. Change the owner and group for incoming to **root / {user group}**.
    c. Change the attribute of incoming to **1760**.
    d. Add the following line to the **/etc/ftpaccess** file:
       upload /home/{guestuser}/incoming yes root {group user} 0600 nadirs
    e. Make sure the guestuser is included in at least one of the **/etc/ftpaccess** class definitions.

By default, a username in vsftp must be specified a a guest user, in pro-ftp, by default, a user is considered as a guest user unless set to real. If using pro-ftp, they are not allowed out of their home directory, or more properly specified, they are “chrooted” to their home directory.

10.7.3.3 Anonymous User

The application **anonftp-(version).rpm** must be installed. This installs four directories in the ftp home directory, **./bin, ./etc, ./lib, and ./pub**

An anonymous user is treated as a root user with the home ftp directory, and therefore cannot see the real /bin, /etc, or /lib directories. We therefore must create a set of "limited" directories and files in the ftp home directory.

Directories:

- /bin Provides for file compression, if it is allowed to be used.
- /etc Contains standard files that the ftp anonymous user will need. Many files are abbreviated to limit the user abilities.
- /lib Contains standard programs that the ftp user will need..

If you desire to allow an anonymous user to upload a file, which is NOT recommended, do the following (do these from the ftp home directory):

1. Create the directory **./ftp/incoming**.
2. Change the owner of the incoming directory to **root.ftp**
   (chown –R root.ftp incoming)
3. Change the attribute of the incoming directory to **1733**.
   (chmod –R 1733 incoming)
4. Add the following line to the **/etc/ftpaccess** file:
   upload $HOME /incoming yes root ftp 0600 nodir
5. Make sure that the anonymous user is included in at least one of the **/etc/ftpaccess** class definitions.

The most significant change between Red Hat 6 and 7 was the relocation of the anonymous ftp user home directory location. To confirm the exact location of the home directory, one should verify the home directory location from the ftp user as specified in the passwd file.

10.7.3.1 Red Hat 6

The location of the ftp home directory for Red Hat 6 and earlier was **/home/ftp**. For the following discussion, replace the variable with:

$HOME = /home/ftp

10.7.3.2 Red Hat 7

The location of the ftp home directory for Red Hat 7 is **/var/ftp**. For the following discussion, replace the variable with:

$HOME = /var/ftp

Anonymous ftp is by default a slight security risk, so make sure the following conditions exist.

1. In the **/etc/passwd** file, make sure the ftp user has the following line:
   ftp:*:14:50:FTP User:$HOME:/bin/true
2. **$HOME/bin** directory must have **root** as it owner and group.
3. **$HOME/bin** must be set to execute only (chmod –R 111 bin).
4. All files in the /bin directory need to be owned by root.
5. **$HOME/etc** directory must have **root** as its owner and group.
6. **$HOME/etc** must be set to read only (chmod –R 444 *)

7. All files in the /etc directory need to be owned by root.
8. Never copy the **/etc/passwd** file to the **$HOME/etc** directory. The **/etc/passwd** file maintains a full list of all users, whereas the **/ftp/etc/passwd** file is abbreviated for security.
9. Edit the **/etc/shell** file. Add the following line to the end:
   /bin/true

10.8 Trivial FTP Server

When we do a File Transfer, we normally desire to have the transfer verified. This requires the use of the Transfer Control Protocol (TCP). Commonly it is designed to go across the Internet where one might incur possible errors. TFTP uses the User Datagram Protocol (UDP). TFTP utilizes service port 69.

10.8.1 Installation of TFTP

The easiest method of installing TFTP is to use yum. Issue the following command:

$ yum –y install tftp*

There is no configuration for the default configuration, although a few minor changes may be made if desired.

Prior to transferring data to the server, the server must be activated. By normal configuration of Red Hat, it is deactivated for security reasons. First issue the command:

chkconfig –list | grep ftp

You should observe something like the following:

vsftp on
gftp off
tftp off

Now issue the commands:

chkconfig tftp on
xinetd

We have now instated the tftp server and re-read the file activate the system.

Note that we need to be careful as to having tftp active – is may be a potential security problem.

10.8.2 Using TFTP

If we restrict our transfer to our local network, which typically does not incur errors, we could utilize a different protocol called Trivial File Transfer Protocol – TFTP. TFTP utilizes the User Datagram Protocol (UDP).

TFTP does a transfer of data directly across the network by issuing the data without setting up a session or checking for errors. Hence it is quite fast.

In order to achieve this, the TFTP server must be configured prior to data delivery to know where to put the data.

Originally TFTP was designed for booting a diskless system – where the boot kernel resided on a server, transferring it across the network to the diskless system and then booting. This required the data be stored in a dedicated

location in memory. If we do have a disk system, then the data is stored in a specified directory, specifically the **/tftpboot** directory. Hence when we transfer any data to our system, it will be automatically stored in the /tftpboot directory exclusively.

If we are working with Cisco routers, we are able save the configuration data by using the command at the privileged prompt. To save the running configuration, we issue the command:

$ copy run tftp

The Cisco router will first query you for the tftp server. We typically enter the IP address, but alternatively enter a hostname if it is stored in our host table on that router. Second, you will be queried for the file name, giving you a default in square brackets. We normally accept this value. Finally, the running-configuration file is transferred. Note that we have not had a way of specifying the location.

On a Unix / Linux tftp server, as we noted before, the data is stored in the /tftpboot directory. We are not able to modify this setup. There is one very important issue that must be taken into account – the file name must exist in the /tftpboot directory. If it does not exist, the incoming file will be refused.

Prior to transferring the file from the Cisco router, note the filename specified in the second question. Once noted, create a filename of zero length in the /tftpboot by using the command:

$ touch filename

We can now proceed with the data transfer.

If we do a long listing of the contents of the /tftpboot directory after the file has been completed, we will observe that it now has some contents. This file is in ASCII format, thus we can read it directly using our favorite display utility.

At a later date, when we desire to upload our stored configuration from the tftp server to the Cisco router, we issue the command:

$ copy tftp run

Again, the router will ask you for the tftp server IP address and file name (with the default name given). The router then retrieves the file from /tftpboot directory. Note that by default, when a query is made of the server, it automatically directs the query to the /tftpboot directory.

10.9 HTTP Server

Web Services, using the Hyper-Text Transport Protocol (HTTP), provide all those wonderful screens of information (ignoring the value of some). In order to provide this service, we must install the appropriate software to provide the web page. One of the most popular software packages is Apache. It is used on over 60 percent of all web servers, including support for MS Windows. HTTP, the protocol for the web, utilizes service port 80.

10.9.1 Apache Installation

There are two options for installing the Apache software. The CD's that you received have all of the Apache software and configuration files as complied by Red Hat. Alternatively, allowing for the fact that Apache is continually

undergoing updates, the version provided on the CD may be out of date. In a future write up we will learn how to install the latest version.

First, check out if Apache was installed during the initial installation. Issue the command:

$ rpm –q apache

Hopefully you will get back the response of:

apache –(version number)

this means that apache is already installed on your system.

Now check to see if the **httpd** process is running by issuing the command:

$ ps aux | grep httpd

This will give a list of process that are operational that contain the text httpd.

If apache was not initially installed, then we need to install it. We must first mount the Red Hat Installation disk on the PC. Go to:

/cdrom/RedHat/RPMS.

Do a listing of all files that start with an **apa*.** You should have something such as:

apache-2.3.3-1.i386.rpm

We now need to install this package. If we attempt to install a package on a Linux system that is already installed, we will get an error message back indicating such. Perform the following command:

$ rpm –ivh apache–2.3.3–1.i386.rpm (modify for the latest version)

You should see a list of lines that indicate the progress of the installation.

Again, you can take the easy way to install apache, use yum.

$ yum –y install http*

10.9.2 Apache Configuration

During the installation, the process installed files in the directory **/etc/httpd,** the default directory when using Red Hat. Other installations install apache in different directories. For Red Hat, you should find the directory **/etc/httpd/conf.**

10.9.2.1 Apache Red Hat 6

For Red Hat release 6, change to **/home/httpd** directory.

10.9.2.2 Apache Red Hat 7 (and later)

For Red Hat release 7 change to the **/var/www/html** directory, and list the files and directories. The /var/www directory is the “business” directory – that is the directory where you go without having to specify an alternate directory. It is also known as the “**root directory**”.

The configuration directory for Apache is **/var/www/httpd**. It is not secure since anyone can change to the **/var/www/httpd** directory. If you change to this directory and do a listing, you should have the following directories:

cgi-bin html icons

10.9.3 A Simple Web Page

Change to the **/var/www/html** directory and do another listing, you should have:

index.html manual poweredby.gif

In the Fedora Core editions, the index.html file does not exist, so you will have to create it. Use your favorate editor.

Now we need to edit the **index.html** file using either the **vi** or **pico (nano)** editor. If the file exists, find the line:

<H1 ALIGN = "CENTER"> {some other information} </H1>

If the file does not exist, then use the text editor to create the **index.html** file and enter the following lines:

```
<html>
<title>{Your name}'s Test Page</title>
<body>
<h1 align = “center”> {put in your name} </h1>
</body>
</html>
```

The above is an extremely simple code. If you know more html code, you may embellish the code to be more creative.

Replace or insert **some other information** with your name to make it distinctive. This way we will now when we get to your page. You can make additional modifications if you desire, but be aware of what you may be doing may cause weird results to the page (but will not cause any harm). Save and close the file. It is not the intent here to teach you html coding, but how to get a page working on your system. If you know basic html, you may have creative fun.

10.9.4 Server Configuration

Change to the **/etc/httpd/conf** directory. Edit the file **httpd.conf.** Find the line (do a search in nano with the **^w** command):

#ServerName new.hosts.name

Change this to:

ServerName {StnNbr}.ourlab.com

It really does not matter what is included in this line, as long as the line is there. The **#** originally is a remark statement, which must be removed. In later versions, this modification is no longer required.

In scanning the file, you should note a number of other parameters that are available to the administrator. These include:

HostnameLookup	Lists of everyone who has access if set to yes
Users	nobody really means everyone, can limit who has access
Group	nobody really means everyone
ScrverAdmin	specifies local administrator
ErrorLog	Where to log errors if such occurs

Other directives of interest include **TimeOut, KeepAlive, MaxKeeepAlive_Timeout, MinSpareServers, MaxSpareServers, StartServers,** and **MaxClients.** These effect the general operation of the server. In general you do not need to change these values unless you need to optimize performance.

In earlier versions of apache, the above modifications were required. In current versions, no modifications are required. Basically install, create your index.html file, and run.

Save the modified file and close.

10.9.5 Apache Service Activation

By default, the web server is disabled. Issue the command:

$ chkconfig – –list | grep http

to determine the operating status. It will probably specify that levels 3, 4, and 5 are off. Turn the on with the command:

$ chkconfig httpd on

Finally, we need to re-read the chkconfig file, issue the command:

$ xinetd

to re-read the Internet services file.

10.9.6 Restarting the Apache Configuration

Whenever you make a change to the configuration files for apache, you need to stop and restart the process. To stop apache, issue the command:

$ /etc/init.d/httpd stop

Then to restart the process, issue the command:

$ /etc/init.d/httpd start

(Alternatively, you can use the command:

$ /etc/init.d/httpd restart or

$ service httpd restart).

10.9.7 Verifying Web Page

With **X Windows running**, start Firefox or one of the other browsers.

At the URL line, type in the line:

http://l92.168.102.{StnNbr} Enter your personal IP address

Your page should immediately be observed.

If you wish to set up multiple different accounts on the same server, create appropriate user directories under the **/home/** such as **/home/instructor,** which will be the instructor's home directory on your system. For ease, copy the file **index.html** to this directory, which must modified. For now edit the file removing your name and replace it with **Professor's Page at Station Number {your station id}**.

10.9.8 Web System Security

As always, we need to be concerned about the security of system. There is a special password file set up for the different users. The very first time a new user is entered into the file, the file must be created. This done with the command:

$ htpasswd –c path/filename username

Typically the **path/filename** will be:

/etc/httpd/conf/users

After the first time, use the command:

$ htpasswd path/filename username

This will append new users to the existing file. Passwords are encrypted, but the coding is different from that used in the general passwd file.

To create groups which may use to update a web page file, you need to create a new file, **/etc/httpd/conf/groups.**

Set the comments to be:

groupname: usernamel username2 username3 ...

As an example, you might create an user group, with users:

ourlab: drice dduck mmouse tboss gwill

The **.htaccess** file (a line in the **srm.conf** file) specifies the authorized users file and groupname. This file must be in every **httpd/html/user** directory.

10.9.9 Web Page Location

Previous to version 7, Red Hat maintained the web pages in the user's home directory. It is now maintained in the **/var/www/html** directory. For a single web page system, the html files would be located there; for a multi web page system, sub directories would be set up for each user in their home directory.

10.9.10 Apache Configuration

Even with the change to version 7, there have been a few additional modifications to the configuration. Initially, the setup required configuration of three different files, but this has subsequently reduced to just one – nice. If one requires adding special features, the other files may also be modified but are not required as all of the changes are incorporated into the first file. The three files are (in order of use) are:

1. httpd.conf
2. srm.conf
3. access.conf

10.9.10.1 httpd.conf

This file specifies the server configuration. It is divided into three sections:

1. Apache server directives
2. Server directives
3. Virtual Host settings

Per the latest design, all configuration are incorporated into **/etc/httpd/conf/httpd.conf**, **srm.conf** and **access.conf** files. srm.conf and access.conf are initially empty and do not require modification.

httpd.conf is a very long file when you realize that it has a great deal of documentation – otherwise it is just a list of configuration statements. Of the list, only a few need to be initially modified for a basic system. With additional research, many other additional options are available.

- **ServerName localhost**
  Initially commented out, it must be modified to the name that is returned to a client. The name must be a valid DNS registered name. Alternatively, the IP address may be entered as "http://192.168.102.201" (put in your own correct address). For our lab, except for the removing of the "**#**", no other modifications are required. We can modify it to our own machine if we are a registered domain name, and in class we will, otherwise, do not modify.
- **DocumentRoot "/var/www/html"**
  This is the directory base where all web pages are maintained. If multiple pages are to be maintained, create subdirectories below it. This is also set below in the ***Virtual Host*** section. For our lab, do not modify. Also note that other programs will refer to the "DocumentRoot". This is where it is specified. Some other distributions of Linux will use a different path.
- **<Directory "/var/www/html">**
  This setting must be the same as ***DocumentRoot*** For our lab, do not modify.
- **UserDir**
  This section of the configuration file sets up users for their own web page. This will be discussed in the next section.

This completes the modification of the **httpd.conf** file. After editing, save and exit.

Verify that **httpd** is running by running the command:

$ chkconfig - - list | grep httpd

If it is not set to on, issue the command:

$ chkconfig httpd on

Before the service can be used, we need to restart the process. Issue the command:

$ xinetd

The web service is now operational.

10.9.11 Setting up a Personal Home Web Page

So you want to set up a personal web page on your serve in addition to the company page. Recall that the company page is maintained in the **/var/www/html** directory. User web pages are maintained in the user's home directory, in a subdirectory specified as **public_html**.

10.9.11.1 Apache Version 1.3

Setting up a user for their own personal web page is very simple. To gain access to another user's personal web page that is located in their home directory, we need to enter the URL of :

http://192.168.102.{StnNbr}/~{username}/

The "**~**" means "user home directory" or /home/username.

Initially this directory for the web page does not exist, so it must be created. As the administrator, or the specific user setting up their own web page, issue the following commands:

```
$ cd   /home/username
$ mkdir   public_html
$ chmod   755   public_html
```

Additionally, permissions must be modified to allow access to the directory. First modify the username directory:

```
$ chmod   711   /home/username
```

Second, the index file within the public_html directory must also have its permissions set to 755. Issue the command:

```
$ chmod   755   index.html
```

This will allow the outside world to view and execute (php or perl programs) in the user's personal web page.

To access the personal web page, enter the URL in your web browser:

URL-Name/~username/

Now if you go to the site http:192.168.102.149/~prof/ (assuming that the instructor as created a web page there), you will see the user's page.

The user's personal web page will now open. Note that the terminating "/" is required.

10.9.11.2 Apache Version 2

Starting with Apache version 2, additional security was added to the configuration file in the **/etc/httpd/conf/httpd.conf** file. A search in the file for the keyword "UserDir" that includes documentation (comments), and you will find a line with the following:

```
</Directory>
#
# UserDir: The name of the directory that is appended onto a user's home
# directory if a ~user request is received
#
# The path to the end user account 'public_html' must be accessible
# to the webserver userid.  This usually means that ~userid must have
# permissions of 711, ~userid/public_html must have permissions of 755,
# and the documents contained therein must be world readable.  Otherwise,
# the client will only receive a "403 Forbidden" message.
#
```

```
    #  See also: http://httpd.apache.org/docs/misc/FAQ.html#forbidden
    #
    <IfModule mod_userdir.c>
       #
       # UserDir is disabled by default since it can confirm the presence
       # of a username on the system (depending on home directory
       # permissions).
       #
       UserDir  disable
       #
       # To enable requests to /~user/ to serve the user's public_html
       # directory, remove the "UserDir disable" line above, and uncomment
       # the following line instead:
       #
       #UserDir  public_html
    </IfModule>
    #
    #
```

To enable a user to access their home directory, make the following changes:

```
    <IfModule mod_userdir.c>
       #
       # UserDir is disabled by default since it can confirm the presence
       # of a username on the system (depending on home directory
       # permissions).
       #
       # UserDir  disable
       #
       # To enable requests to /~user/ to serve the user's public_html
       # directory, remove the "UserDir disable" line above, and uncomment
       # the following line instead:
       #
       UserDir  public_html

    </IfModule>
```

As before, create the directory **./public_html**. Remember to change the permissions for the user's directory, public_html directory, and index.html file to allow the world (other) to be able to read it. Do not all any user to have write privileges, or you will set up a major security risk. Also make sure that the owner of both the public_html directory and the index.html file is set to the owners username (they typically will not be if you create them as the administrator.

Within the public_html directory, create an index.html file with the contents of what the user desires. Alternatively, instruct the user on how to ftp his / her data to the directory. The user will now be able to view the web page with the URL of **http://your_URL/~username/**.

10.10 DNS Server

The **Domain Name System (DNS)**[4,5,6,7,8] is a network replacement for the local /etc/hosts file. The intent is to support a collection of IP Addresses / Names gained off of the Internet. It is designed to obtain either an IP address given a URL, or the URL given the IP address. DNS is a hierarchal structure of name servers that are delegated a level of authority to provide a specific level of information. DNS utilizes service port 53 (name server).

Recall that we can edit our local host file to include all of the local address resolution that we desire, but this would be rather impractical to add every site on the network. DNS acts as the host name resolver for all of the Internet hosts. The big difference is that DNS does not hold all of the addresses, but must do a request for the one system that holds all the addresses for a specific local network.

Lets assume that we have the following host systems on our local network performing the specified function. Each of these machines will have a static address because we want them to be resolvable by the Internet.

DNS Server	192.168.102.150	ns.ourlab.com
Mail Server	192.168.102.151	mx.ourlab.com
Web Server	192.168.102.152	www.ourlab.com
ftp Server	192.168.102.153	ftp.ourlab.com

10.10.1 DNS Installation

The installation of DNS is straightforward to either install from the CDs or via **yum**. The application goes by the name of **BIND**, **Berkeley Internet Name Daemon**. As can be observed, it was developed at the University of California, Berkeley. First, test to see if it has been installed by issuing the command:

rpm – q bind

If installed, this will return the software version number. If not installed, then the bind applications may be installed from one of the CDs by issuing the command:

rpm – ivh bind*

This will install DNS and all of the associated files that are required. After the installation, the latest version should be obtained to insure that all security features are incorporated.

To install using **yum**, issue the command:

yum – y install bind*

This will install the latest version of the application.

10.10.2 Network Setup

The Internet is set up on a numeric (binary) addressing system, where every host is located on a specified local address. Because it is difficult to remember

---

[4] **Using Linux – Special Edition** by Jack Tackett and Steven Burnett; Que

[5] **Red Hat Linux Unleashed** by David Pitts and Bill Ball; SAMS

[6] **Red Hat Linux 6 Server** by Mohammed Kabir, IDG Books

[7] **DNS and Bind** by Paul Albitz & Cricket Liu, O'Reilly

[8] **Red Hat Linux – The Complete Reference, 2nd Ed.** by Richard Petersen, Osborne / McGraw Hill,

the numeric address, we have developed a method or system where we may assign a name to a local network and to a specific host, called the URL.

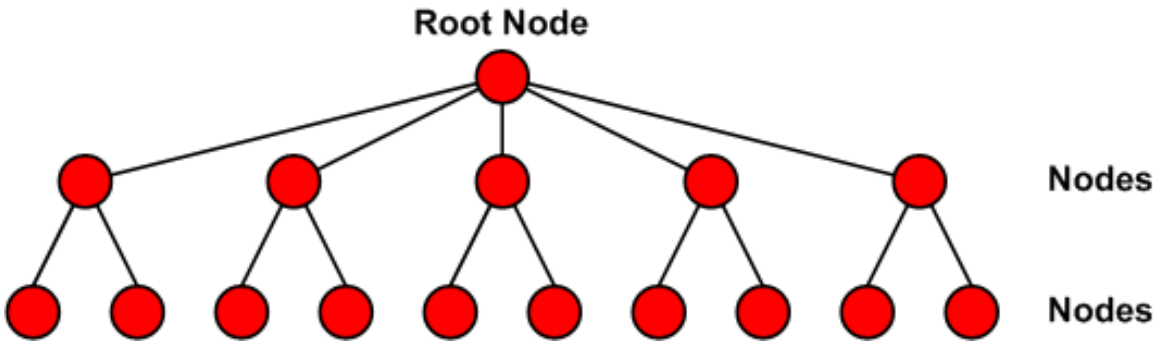


Figure 10-1: Nodes on at Tree Network

The naming convention is based on a tree structure – becoming more detailed the further one travels down the tree. This is shown in Figure 1.

Each node is called a **zone** and includes all nodes or zones below it. An individual host is a station within a **zone** on a local network.

Now to find another station within the Internet, knowing its name, we first look at our local **/etc/hosts** file, but this often does not include the address of the place we wish to view or go to. To solve this, we create a **Domain Name System** , or **DNS**, that we ask for a name to numeric address resolution. Often our local DNS host will not know the answer to our name request, so it then must request the information from another DNS system that is in higher authority. Through a number of name resolution questions, we will eventually be able to obtain our answer as to what the numeric address is.

The highest zone is known as the **"."**, below this top zone exists the zone of **gov, mil, edu, com, org, net** and others. Figure 2 shows the base structure. At present there are 13 top level DNS servers, a surprising "non-diverse" locations around the United States. These 13 servers are established for redundancy, but have been shown in the past to have been shut down by attacks.

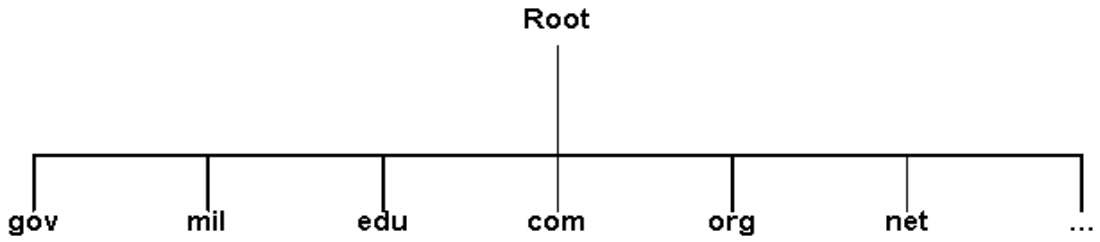


Figure 10-2: Common Top Level Zones

Each of these points is a zone made up of all of the other zones below it.

Beneath the **gov** zone we have **whitehouse, senate, house, nasa, hud** and many others. Beneath **edu** we have **ucla, ucb, ucsb, uo, utd, uta, udallas** and **devry**, plus nearly every other university and college in America. Again each of these is considered a lower level zone.

Each of these zones are reserved for certain types of services, but that does not mean that they might be used inappropriately. It is obvious that ".gov" and

".mil" are reserved by the government, ".edu" is reserved for valid educational institutions, ".com" is for commercial business, ".org" is for non-profit organizations, and ".net" is for Internet network providers that do not provide other commercial services. Of course, there are many businesses that occupy more than one **Top Level Domain (TLD)**.

Now beneath the devry zone we have **kc, pom, phx, dal** and the other campuses. Our tree to the devry dallas would appear as shown in Figure 3.

The domain name for our examples, we will set up to be:

ourlab.com

As another example, say we have a business that is located across multiple states, and we wish to designate each separately. We might have:

Primary name: **ourbiz.com**

Now to designate each state, we could add:

Massachusetts: **ma.ourbiz.com**
California: **ca.ourbiz.com**
Washington: **wa.ourbiz.com**

In addition, each state has an engineering and HR department. So we add:

eng.ma.ourbiz.com	**eng.ca.ourbiz.com**	**eng.wa.biz.com**
hr.ma.ourbiz.com	**hr.ca.ourbiz.com**	**hr.wa.ourbiz.com**

At this point we have defined smaller zones – we have now defined **Fully Qualified Domain Names**,or **FQDN**, but still have not specified any specific servers. As we noted before for our example, common servers will include DNS, Web, Mail and FTP. Others might also be included. We could have a set of these servers for each department, although that would not be optimum, but we very well might have one for each state. Lets add to our example, and say we have the following servers:

ns.ma.ourbiz.com	**ns.ca.ourbiz.com**	**ns.wa.ourbiz.com**
web.ma.ourbiz.com	**web.ca.ourbiz.com**	**web.wa.ourbiz.com**
mail.ma.ourbiz.com	**mail.ca.ourbiz.com**	**mail.wa.ourbiz.com**
ftp.ma.ourbiz.com	**ftp.ca.ourbiz.com**	**ftp.wa.ourbiz.com**

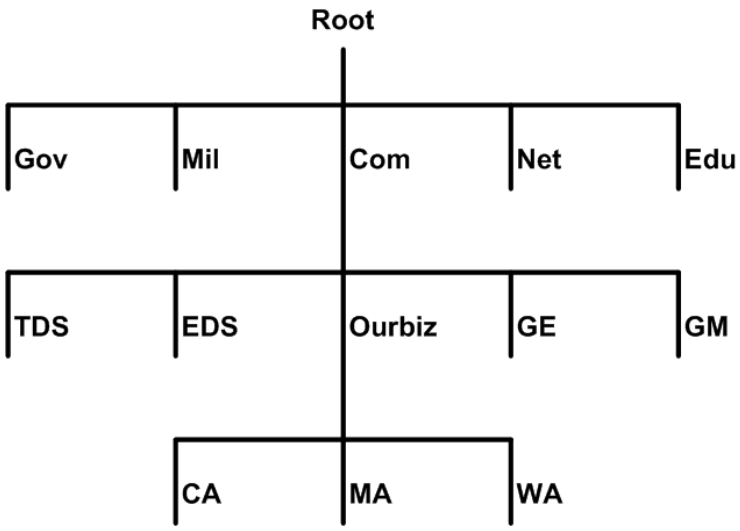


Figure 10-3: Ourbiz Tree

We do not write the top root **"."** zone – it is understood. There is a name server sitting at the root location that provides the IP address to the second level (.com, .net, .edu, etc.) name servers. In fact, there are 13 root servers spread across the United States to provide redundancy and service diversity. Below the root servers are various companies that act in the place of ICANN (**Internet Corporation for Assigned Names and Numbers)**, which is the agency responsible for issuing domain names. In our example, all registered education institutions are addressed on the name server at the .edu location and commercial business are in the .com location. Here we have the address for example ourbiz.com, with further resolution to the different states.

In the corporate office of Ourbiz, we would locate the primary DNS server. This is the Name Server that the Internet ".com" server would point to. In this server, we would provide resolution to the CA, MA, and WA DNS servers. Thus in each state, we would have another DNS server that would provide information to the local mail, web, mail, and ftp servers.

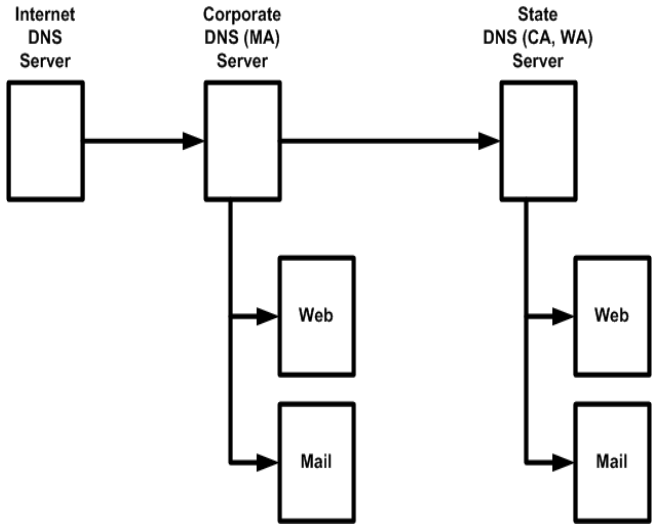


Figure 10-4: DNS Pointer System

To be properly configured, every zone must maintain a DNS server, and should properly be set up with a backup server in case the primary should fail. By specification of Internic (Internet's Network Information Center), a FQDN (Fully Qualified Domain Name) must maintain a master DNS server and a slave, or backup, DNS server. In our example, the DNS server at CA could also serve as the backup to the Corporate (MA) DNS server.

As of 1998, a new format of the DNS configuration was implemented; the general installation uses a process called **Bind**. The present version of Bind is the series 9.X, whereas the previous version was of the series 8.X, that being preceded by Bind series 4.X. Newer installations should use version 9.X, although older version of Bind 8.X and 4.X still exist.

There are three levels of DNS support – Local caching, Master service, and Secondary service, known as **Slave**. Local Caching is intended to support an isolated LAN. A Master system is used with a small business with Internet connection. A larger business with its own dedicated IP address must maintain a Secondary system to support naming in case of failure of the Master system. According to the Internic rules, in order to have your own domain name, you must maintain a master and secondary DNS system.

The installation of DNS through **Bind** creates a series of files, typically using the base name of **named**.

The following is a process for setting up a DNS server for a small enterprise. Simply said, this is a complex setup that requires a lot of explanation. Additional research is required to appreciate all of the options and the full capability of the DNS server.

10.10.3 BIND Version 9

In setting up the DNS system under the Version 8 and 9 formats, a number of files must be set up. For a major system, the number of files might be quite large, but the management is a little more straightforward than in version 4. Version 9 supports even higher enhanced security issues over Bind 8.

The following files must be set up for a system:

named.conf	Specifies what files are used
forward domain zone file	Specifies the IP to URL conversion
reverse domain zone file	Specifies the URL to IP conversion
127 reverse zone file	Specifies the localhost to IP conversion
local forward zone	Specifies the local URL to IP conversion
cache zone file	Specifies Internet Name Servers

We normally set up our DNS server(s) to only support our local network. By this we mean that it acts as the local network (zone) host file that may be interrogated by the Internet. We desire to specify in it machines that are on our network. Naturally there are always exceptions to this rule.

10.10.3.1 /etc/named.conf

Lets start by providing the named.conf file, in the /etc directory, in is general format as an example, and then go back and try to explain it (it is very obvious – after you know what you are doing □). Note that the convention shown here is slightly different from that you may find in the default files on a system, the default configurations for clarification.

```
#named.conf
# security key option
controls {
        inet 127.0.0.1   allow  { localhost; }  keys  { rcndckey; };
};
# this is the encrypted key that allows transfer to the slave dns server
include   "/etc/rndc.key";
options   {
        directory   "/var/named";
};
# cache zone file – required for cache only
zone   "."   {
        type  hint;
        file   "named.ca";
};

# localhost forward zone file – required for cache only
zone   "localhost"   {
        type  master;
        file   "localhost.zone";
};

# 127 localhost reverse zone file – required for cache only
zone   "0.0.127-in-addr.arpa"   {
        type  master;
        file   "0.0.127.in-addr.arpa.zone";
};

# forward domain zone file
zone   "domain.com"   {
                                        type  master;
        file   "domain.com.zone";
};

# reverse domain zone file
zone   "102.168.192.in-addr.arpa"   {
        type  master;
        file   " 'rev.addr.in-addr.arpa.zone' ";
                                PUT IN YOUR OWN REVERSE IP
};
```

10.10.3.1.1 named.conf Option Details

The first file that must be configured is the **named.conf** file. It has the following format:

```
keyword {
        [details of the condition];
};
```

Four different keywords are given in our example file, namely **options, inet, include,** and **zone**, although others exist.

The first keyword that must be used is **options**. It has the basic format as:

```
options {
        directory   "pathname" ;
        statistics-interval   number ;
        forwarders  { IP-Address; } ;
        forward   (only | first) ;
};
```

The directory line specifies where all other files may be found. This is considered the base path if the other file locations are to be relative to this location. Although not absolutely necessary, it is better to assume that it is. For earlier distributions, it directly specified the default path of "**/var/named**". To improve security, this value still exists, but the true path for the additional files is "**/var/named/chroot/var/named**". Note that the directory of "chroot" exists in the path; chroot holds a special place in Linux, setting up a "jail", meaning that if a person should be able to ftp into that directory, they will not be able to exit. Now in Fedora, all of the dns files are located in the chrooted path.

The statistics-interval line specifies how often information is written to the log. This log is very important if we are to maintain control in case of a problem. The default is set to 60 minutes, but may be varied if appropriate. This line is optional.

The forwarder line specifies where a query is to be forwarded to if it is unable to be resolved on this server. More than one remote server may be specified, separated by a semicolon. This line is optional, but should be used if we need to query the ISP DNS name server.

The forward line is used only if the forwarder line exists. If set to **first**, it will forward the query to the list immediately and will do a look up on itself only if the remote servers do not respond in the specified time (timeout). If set to **only**, then a local lookup will not be performed.

The **inet** option specifies a security code is to be used. This security code limits who is allowed to download update information – namely the slave dns server. We can place the security code in this line, or in an external file.

This is where the last option, **include** comes into play. It specifies the encrypted key that is to be used and where it is located. For the secondary dns serve to work, this specific code must be copied to that system. If you list the /etc/rndc.key file, you will observe the encrypted security code. Note also that only *root* and the group *named* may read the file.

The last option of the **named.conf** file is **zone**. It has the basic format as:

```
zone   "domain_name"  {
        type   level ;
        file   "path/filename" ;
};
```

Multiple zones may be set up to configure specific requirements.

The **type** line specifies the function of the zone. The following options exist:

hint	The zone domain name is **"."** . This sets up the cache process. Here we set up the IP Address for primary Internet name servers. The filename is typically **named.ca**.
master	This zone domain_name file is the primary authority name server for the specified domain. The path/filename must

point to a specific definition file. The file name is not critical, it can be anything you want, just make sure that the name specified in this section exists in the directory specified in the option section.

slave The zone domain_name file is the backup or secondary name server for the specified domain. For a slave, the **master** line may optionally be included, which specifies the IP Address of the server to which it obtains updates. It typically points to the same filename as for the master level.

In the **named.conf** file, for a business that has a DNS system, two zone files must be created for each zone that is established. The first file specifies name-to-IP Address resolution and the second specifies the IP Address-to-name resolution – commonly referred to as the reverse DNS. There is no absolute required naming process, although the administrator should establish a uniform naming convention to allow easy user interpretation.

10.10.3.2 /var/named zone files

For each zone specified in the **named.conf** file, an additional file must be created in the /pathname specified in the directory option portion of the named.conf. The pathname is typically **/var/named**, as specified in the directory option of the named.conf file. As noted previously, to enhance security, these files are now located in the **/var/named/chroot/var/named** directory.

From the master zone record in the **named.conf** file we might have the following:

In each of the following, we show the email address of "admin.ourlab.com.". We normally put in a username that is forwarded to the system administrator without putting in their real username – we do not want to want to broadcast our username across the Internet.

10.10.3.2.1 Forward Domain Zone File

From the named.conf file:

```
zone "ourlab.com" {
        type master;
        file "ourlab.com.zone";
};
```

We create the **/var/named/ourlab.com.zone** file:

```
$TTL  86400
@     IN    SOA   namesvr.ourlab.com.          admin.ourlab.com
(
                  2002021501 ;     Serial       change on update
                  7200       ;     Refresh      2 hrs
                  3600       ;     Retry        1 hr
                  43200      ;     Expire             12 hrs
                  86400      ;     Minimum      1 day
                  )
      IN    NS              namesvr.ourlab.com.
      IN    MX    0         mail.ourlab.com.
```

```
namesvr     IN     A          192.168.102.150
mail        IN     A          192.168.102.151
web         IN     A          192.168.102.152
ftp         IN     A          192.168.102.153
www         IN     CNAME      web.ourlab.com.
ns          IN     CNAME      namesvr.ourlab.com.
mx          IN     CNAME      mail.ourlab.com.
```

Above we have included the ns and mx in the conical name. This is required if we want to have a remote client to be able to search for the name server and mail server by their Internet generic names.

Note that the server names (namesvr, mail …) must start as the first character for each line, otherwise the server will not work.

10.10.3.2.2 Reverse Domain Zone File

From the named.conf file:

```
zone  "102.168.192.in-addr.arpa" in {
        type  master;
        file  "102.168.192.in-addr.arpa.zone";
};
```

We create the **/var/named/102.168.192.in-addr.arpa.zone** file:

```
$TTL  86400
@     IN     SOA   namesvr.ourlab.com.      admin.ourlab.com.  (
                   2002021501  ;     Serial      change on update
                   7200        ;     Refresh     2 hrs
                   3600        ;     Retry       1 hr
                   43200       ;     Expire            12 hrs
                   86400       ;     Minimum     1 day
                   )
      IN     NS          namesvr.ourlab.com.
150   IN     PTR         namesvr.ourlab.com.
151   IN     PTR         mail.ourlab.com.
152   IN     PTR         web.ourlab.com.
153   IN     PTR         ftp.ourlab.com.
```

10.10.3.2.3 127 Localhost Reverse File

From the named.conf file:

```
zone "0.0.127.in-addr.arpa"  {
        type master;
        file "0.0.127.in-arpa.zone";
};
```

We create the **/var/named/0.0.127.in-addr.arpa.zone** file:

```
$TTL  86400
@     IN     SOA   localhost.    admin.localhost.   (
                   1999121001        ;     sequence number
                   8H                ;     refresh rate (8 hours)
                   2H                ;     retry (2 hours)
```

```
                        1W              ;       expire (1 week)
                        1D      )       ;       minimum ttl (1 day)
        IN      NS              namesvr.ourlab.com.
1       IN      PTR             namesvr.ourlab.com.
```

10.10.3.2.4 Localhost Forward Zone File

From the named.conf file:

```
zone "localhost" {
        type master;
        file "localhost.zone";
}
```

We create the **/var/named/localhost.zone** file:

```
$TTL 86400
@       IN      SOA     localhost.    admin.localhost.   (
                        2002021501      ;       serial number
                        8H              ;       refresh rate (8 hours)
                        2H              ;       retry (2 hours)
                        1W              ;       expire (1 week)
                        1D      )       ;       minimum ttl (1 day)
                IN      localhost.
                IN      A       127.0.0.1
```

10.10.3.2.5 Cache File

From the named.conf file:

```
zone    "." {
        type hint;
        file    "named.ca";
}
```

The **/var/named/named.ca** file is printed out later, and no changes are made for the default configuration.

In review, the first zone provides forward name resolution for the specified zone, pointing us to the file with the details. The second zone provides for the reverse name resolution for the specified zone. Both zone specifications are required for a designated zone.

In the above example, we have created a bogus IP Address for the Ourlab corporation using a segment of 192.168.X.Y private address. Note in the second zone file the IP Address is given in reverse and that only three octets are specified.

Next we need to provide the local host reverse file. This is the zone "0.0.127-in.addr.arp" grouping. Again we specify the type (master) and the filename, which is typically called "named.local". As a general rule, we need to change the default "localhost" over to our local domain (FQDN) name.

The last file that we must look at is the local reverse file, or the localhost.zone file. In this file we again, in general, only need to change the "localhost" entry to our local domain name.

Finally we have the cache file. As a general rule, you never need to do anything with this portion, but do need to make a periodic check to see if it has

been updated – and if so, down load the new version and replace the existing "old" named.ca file.

10.10.4 Naming File Construction

Now we want to look at the structure of for of the five main files, so we can understand what is happening. Each will have a similar structure. By default on installation, the basic files – ***named.local***, ***localhost.zone*** and ***named.ca*** generally already exist. We will need to create the remaining two. These three files are the minimum needed to establish a "caching" file DNS system.

10.10.5 Start of Authority Record

The first record put into a zone file is typically the **Start of Authority (SOA)**. There must be one and only one SOA record per zone file. It has the format of:

```
@    IN    SOA  nameserver. contact-email-address. (
                 serial_number     ; Serial
                 refresh_number    ; Refresh
                 retry_number      ; Retry
                 expire-number     ; Expire
                 minimum_number ;  Minimum Time to Live
                 )
```

10.10.5.1 Statement of Authority (SOA)

The first field is the name_field. It is always set to the **@** character and need not be repeated in the following records. The next field is **IN** for Internet. The third field specifies the **Start of Authority**. We read these three fields as:

Domain-Name (**@**) on the **IN**ternet **S**tatment **o**f **A**uthority.

The "**@**" symbol is to be read as your domain name.

The forth field specifies the domain name server – note that it must terminate with a **"."** . The last field specifies the email address of the administrator of the DNS server. This should be a generic name for security and privacy issues. Note that we replace the email "@" character with a "**.**" . Because in the file the "@" character has a reserved meaning and again this must terminate with the **"."**. At this time, the email address is required for future requirements, but is presently not used.

10.10.5.2 Serial Number

The second line is a serial number for the file. Every time the file is modified, the Serial Number must be incremented or changed. A common practice is to create a serial number with the format of YYYYMMDDNN. Other name servers will interrogate this file and record its data and serial number, and on future interrogations will compare the recorded SN to the existing – and if different will record the new data, or if the same will reset its timer without downloading the file.

> Note: The maximum length of the serial number is 10 digits. Anything longer than this will cause an error when the service is restarted.

10.10.5.3 Refresh

Line three specifies how often a secondary name server should check if it needs to be updated. The number is normally in seconds, but may also be specified in hours (h), days (d) or weeks (w).

10.10.5.4 Retry

Line four specifies how long a secondary server should wait before attempting to obtain data on a previously failed transfer. Again the value is typically in seconds.

10.10.5.5 Expire

Line five specifies how long a secondary server is to use the zone data, after which the existing data is considered to be expired. This value should be at a minimum of 4 times the Refresh rate to insure that the secondary server does not have expired data. The value is also given typically in seconds.

10.10.5.6 Time to Live

The last line specifies the general Time To Live (TTL) value that is used for all other records in the file unless otherwise specified.

10.10.5.7 Name Server Record

Now we must create the two specified files. They will both be similar in format, but provide different functions. The syntax of these files is of the format:

[name] [ttl] addr-class record-type specified-data

The ***name*** field is always is the name of the domain record, and **it must always start in column 1**. Typically only the first resource record is needed for the ***name*** field, and this may be replaced with the **@** character, which specifies to use the domain name specified in the zone block of the **named.conf** file. The **@** character is not required.

The ***ttl***, or Time to Live, field is optional. It specifies how long the data is to be stored before being updated. If the field is blank, the default value is taken from the SOA resource record (detailed below).

The ***addr-class*** is the address class field. The primary value used in most records is **IN**, for Internet addresses and other information.

The ***record-type*** specifies the type of resource record. This will be explained below.

The ***specific-data*** specifies the appropriate information for that resource record. This will be explained below.

The second type of record type in a zone field is the **Name Server (NS)**. Its general syntax is:

IN NS *name-server-hostname.*

Here we no longer need to specify the name and TTL fields, as they are specified in the SOA record. Multiple name servers may be specified for each zone, and in fact it is preferred (required) by Internic to have two for a registered domain name zone. By this rule, there would be two DNS servers at ourlab.com (Oakbridge – Chicago), but only one is required at ourlab.com, since the slave DNS server could be the one at Oakbridge.

10.10.5.8 Address Record

The third record type in a zone file is the **Address (A)**. This is used to specify the IP Address for a specific hostname. It has the syntax of:

hostname IN A IP-Address

There should be at least one **A** Record per zone file.

10.10.5.9 Domain Name Pointer Record

The forth-record type in a zone file is the **Domain Name Pointer (PTR)**. This record type is used in the reverse zone file to resolve an IP Address to a hostname. The syntax is:

IP-Address IN PRT hostname.

10.10.5.10 Canonical Name Record

The fifth record type in a zone file is the **Canonical Name (CNAME)**. Using this record type, we can specify alternative hostname (an alias) for the official (canonical) hostname. The syntax is:

alias-name IN CNAME canonical-hostname.

Commonly, one does not name a server by the server-function that it performs. For example, one will typically not call the web server “www”. For example, one might name the server “homebase”. Across the Internet, one would normally refer to the web server as “www”, so the administrator would add the line:

www IN CNAME homebase.ourlab.com.

10.10.5.11 Mail Exchange Record

The last record type is the **Mail Exchange (MX)**. It is used to specify a host that is the SMTP mail server. The syntax is:

IN MX *preference-value mail-server-hostname.*

The preference-value is used to provide a priority when we have multiple mail servers. If we only have one, set it to zero.

10.10.6 Slave DNS Server

We must add a block of code to our **named.conf** file to specify the Secondary Server if it is to be configured.

```
zone  “ourlab.com” (
        type slave;
        file  “zn.ourlab.com”;
);
```

The secondary zone file acts like the primary one, but is used as a backup to the primary. Configuration of the secondary zone file is similar, but must be modified to account for the fact that it is the secondary. Creating this file is easier than creating the Primary zone file because a process has been created to automatically generate it by using the zone file from the Primary Server. When logged onto the Secondary DNS server, from the **/var/named** directory, issue the following command:

named–xfer –z ourlab.com –f sv.ourlab.com –s 0 ns.ourlab.com

where:

named-xfer	is the program that converts the file
–z	specifies the zone name of the source
ourlab.com	is the specified zone name
–f	specifies zone file name
sv.ourlab.com	is the file name in the Primary zone
–s	specifies the name of the server where the file resides

The file will be transferred and appropriately modified as a secondary server file.

10.10.7 Cache File

The caching file actually tells the system where to go to find all of the other Internet names. This file is very cryptic because it does not really say what type of name you are looking for. The original intent was for each specified system to provide the DNS function for a specified top-level zone (gov, mil, edu, com, net , org and others). Since then the distinction has blurred a little, because you can go to several different systems to resolve a name.

A typical **named.ca** file follows, but this should be checked on a periodic basis because it may be updated. This can be researched off of the Internet. There are thirteen network DNS servers across the United States, unfortunately they are clustered in just several locations. Since the number of servers generally does not change, the file will typically remain static.

Each server is updated every time a new domain name is registered (done in batch). The entry in each contains only the base domain name, and points to another DNS server that must provide additional resolution of server naming. Every nation around the world must have its set of top-level DNS servers. For example, England would have their set, and the servers in the US would point to the English set for any domain name ending in ".uk".

```
;  The latest updated version of this file may be obtained from
;  InterNIC at
;  ftp://ftp.rs.internic.net/domain/named.root
;
;       formerly  ns.internic.net
.               3600000      IN      NS     A.ROOT-SERVERS.NET.
A.ROOT-SERVER.NET.  3600000      A       198.41.0.4

;       formerly  ns1.isi.edu
.               3600000      IN      NS     B.ROOT-SERVERS.NET.
B.ROOT-SERVER.NET.  3600000      A       128.9.0.107

;       formerly  c.psi.net
.               3600000      IN      NS     C.ROOT-SERVERS.NET.
C.ROOT-SERVER.NET.  3600000      A       192.33.4.12

;       formerly  terp.umd.edu
```

```
.                3600000      IN      NS      D.ROOT-SERVERS.NET.
D.ROOT-SERVER.NET.   3600000      A       128.8.10.90

;       formerly  ns.nasa.gov
.                3600000      IN      NS      E.ROOT-SERVERS.NET.
E.ROOT-SERVER.NET.   3600000      A       192.203.230.10

;       formerly  ns.isc.org
.                3600000      IN      NS      F.ROOT-SERVERS.NET.
F.ROOT-SERVER.NET.   3600000      A       192.5.5.241

;       formerly  ns.nic.ddn.mil
.                3600000      IN      NS      G.ROOT-SERVERS.NET.
G.ROOT-SERVER.NET.   3600000      A       192.112.36.4

;       formerly  aos.arl.army.mil
.                3600000      IN      NS      H.ROOT-SERVERS.NET.
H.ROOT-SERVER.NET.   3600000      A       128.63.2.53

;       formerly  nic.nordu.net
.                3600000      IN      NS      I.ROOT-SERVERS.NET.
I.ROOT-SERVER.NET.   3600000      A       192.36.148.17

;       temporarily housed at NSI (InterNIC)
.                3600000      IN      NS      J.ROOT-SERVERS.NET.
J.ROOT-SERVER.NET.   3600000      A       198.41.0.10

;       housed in LINX, operated by RIPE NCC
.                3600000      IN      NS      K.ROOT-SERVERS.NET.
K.ROOT-SERVER.NET.   3600000      A       193.0.14.129

;       temporarily housed at ISI (IANA)
.                3600000      IN      NS      L.ROOT-SERVERS.NET.
L.ROOT-SERVER.NET.   3600000      A       198.32.64.12

;       housed in Japan, operated by WIDE
.                3600000      IN      NS      M.ROOT-SERVERS.NET.
M.ROOT-SERVER.NET.   3600000      A       202.12.27.33
```

eof

The **named.ca** file is a standard file created by the network. For the latest version, it should be downloaded from the network. It is subject to change so the administrator must insure that the latest version is maintained.

We go to the Internet (IN) to the Name Server (NS) with the name of **A.ROOT-SERVERS.NET**. Each one of these locations maintains the highest level of DNS name resolution, so that if we do not know the name one our local DNS server, we go directly to one of the above named servers. This format sets the Name Server address immediately below the name server name.

An important point about this file, it may be subject to frequent update by Internic and must be downloaded on a periodic basis.

10.10.8 Setting up the Workstation

In order for the local host workstation to resolve the DNS lookup, two files need to be modified.

10.10.8.1 host.conf file

The first file that must be modified is the **/etc/host.conf** file. This file tells the resolver what services to use and in what order they are to be implemented.

```
# Sample /etc/host.conf file
#
# Lookup names via the /etc/hosts file first then fall back to DNS.

order  hosts,  bind
                {the following lines are not necessary to host.conf file, but
                may be configured if desired}
# We don't have machines with multiple addresses
multi  off                                      ;Optional
# check for IP Address spoofing
nospoof  on                                     ;Optional
# and warn us if someone attempts to spoof
alert  on                                       ;Optional
# Trim the ourlab.com domain name for host lookups
trim  ourlab.com                                ;Optional
```

This example is a general resolver configuration for our local domain. The resolver looks up the host names by using the local **/etc/hosts** file first and then tries the DNS. This may also be set up to be:

```
order   bind,   hosts
```

where we will now check out what the nameserver has first, then look to the local hosts file see what it has. This configuration is not the best because if bind fails, you may not be able to resolve to the hosts file.

Spoofing is where someone tries to access a host while using a local name rather than their own. By setting **nospoof on**, we create an extra load on the network, but we insure that someone trying to access our network is using his or her proper address. If someone does try to access our system while using a spoof IP Address, then the system will issue an **alert**.

10.10.8.2 resolv.conf file

The **/etc/resolv.conf** file specifies the names for the local host and where we need to look for the DNS server. An example of the **/etc/resolv.conf** file is:

```
# /etc/resolv.conf for  ourlab.com
#
# Set our local domain name
server   ourlab.com
# Specify our primary name server
```

nameserver 192.168.102.150 << put in the correct IP Address

Note that we must specify the IP Address of the nameserver. If we use a name, and we look to the DNS first, then we will never be able to resolve the name to an address. If a valid address does not exist, then the client will not be able to obtain a name resolution from the DNS server.

During the normal operation of our system, we must have a valid external nameserver IP address. Up to three nameserver addresses may exist.

10.10.9 Doing it an easier way – well maybe

Naturally there has to be an easier way – using the GUI built into X Windows. If you open the System menu and then DNSConfigurator, you will find that it will lead you through the setup directly. But wait, there is a catch. The set of files that it produces may have several errors. Namely it leaves out a few lines that are very important – specifically that for the nameserver. You need to review the files and verify that all is correct and make corrections as necessary. In the past, it has deleted files that were created.

The GUI window has a set of fields that you need to fill in. This is obvious if you know what you are doing, and in some cases it will prompt you if you enter an error. Experimenting does no harm. Just make sure that you verify that all of the required files exist and the content of each is correct. Make sure you test it before just accepting what it does.

10.10.10 Testing the DNS System

After all is set and done, you need to test the configuration. The easiest way to do this is to run various tests on it. There are three commands available to test the configuration:

nslookup	outdated command by Unix / Linux standards, but will work to be backward compatible with Microsoft
dig	full data reporting for the query
host	basic response of DNS information

Each of these commands uses the format of:

$ dig –x IP-Address or
$ dig URL name

Utilizing the command format of dig IP-Address does not work. It returns the Authority server, but does not answer the question. Using the dig –x IP-Address does work.

10.10.11 Summary

All of the above can be confusing because it is fragmented. Lets do a consolidation of our lab network. When we get down to the nuts and bolts, it really is a lot easier than all of the above makes out. Of course, there are a lot of issues that have not been addressed here and require further research to be fully proficient at its administration.

10.10.11.1 Lab */etc/named.conf* file

```
options {
        directory   "/var/named";
```

```
};
controls {
        inet 127.0.0.1   allow  { localhost; }  keys  { rndckey; };
};
#  IN Domain Forward File
zone   "ourlab.com"  {
        type master;
        file  "ourlab.com.zone";
};
#  IN Domain Reverse File
zone  "102.168.192.in-addr.arpa"  {
        type master;
        file  "102.168.192.in-addr.arpa.zone";
};
#  Local Forward File
zone "localhost"  {
        type master;
        file  "localhost.zone";
};
#  Local Reverse File
zone "0.0.127.in-addr.arpa"  {
        type master;
        file "0.0.127.in-addr.arpa.zone";                    was named.local
};
#  Cache File
zone  "."  {
        type hint;
        file "named.ca";
};
include "/etc/rndc.key";
```

eof

10.10.11.2 Lab zone files

Make sure while entering the configuration files, that while I show the left hand side indented for these pages, them must not be indented on your text file. This has been done for clarity purposes. Recall that the files are now located in the **/var/named/chroot/var/named** directory for improved security.

10.10.11.2.1 Zone ourlab.com.zone domain forward file

```
$TTL  86400
@       IN      SOA   namesvr.ourlab.com.        admin.ourlab.com.  (
                      1999122700      ;     Serial        change on update
                      7200            ;     Refresh       2 hrs
                      3600            ;     Retry         1 hr
                      43200           ;     Expire        12 hrs
                      86400           ;     Minimum       1 day
                      )

                IN    NS          namesvr.ourlab.com.
                      IN    MX    0       mail.ourlab.com.
namesvr         IN    A           192.168.102.150      ;{put in real IP Address}
mail                  IN    A     192.168.102.151      ;{put in real IP Address}
web                   IN    A     192.168.102.152
```

```
ftp             IN      A       192.168.102.153
www             IN      CNAME   web.ourlab.com.
tftp            IN      CNAME   ftp.ourlab.com.
ns              IN      CNAME   namesvr.ourlab.com.
mx              IN      CNAME   mail.ourlab.com.
                {comment out undesired servers if appropriate}
eof
```

Notice that two lines, starting with “ns” and “mx” are included as conical names. From experience it has been found that it is a good idea to include these lines, as someone can then issue the request:

$ dig ns.yahoo.com

10.10.11.2.2 Zone 102.168.192.in-addr.arpa.zone domain reverse zone

```
$TTL  86400
@     IN    SOA   namesvr.ourlab.com.     admin.ourlab.com.  (
                  1999122700 ;      Serial
                  2h         ;      Refresh
                  1h         ;      Retry
                  12h        ;      Expire
                  86400      ;      Minimum
                  )

@     IN    NS    namesvr.ourlab.com.
150   IN    PTR   namesvr.ourlab.com.     ;{put in real IP Address}
151   IN    PTR   mail.ourlab.com.        ;{put in real IP Address}
152   IN    PTR   web.ourlab.com.
153   IN    PTR   ftp.ourlab.com.
                  {comment out undesired servers if appropriate}
eof
```

10.10.11.2.3 Zone 0.0.127.in-addr.arpa.zone local reverse file

```
$TTL  86400
@     IN    SOA   namesvr.ourlab.com.          admin.ourlab.com.
(
                  1999121001        ;     sequence number
                  8H                ;     refresh rate (8 hours)
                  2H                ;     retry (2 hours)
                  1W                ;     expire (1 week)
                  1D    )           ;     minimum ttl (1 day)

      IN    NS          namesvr.ourlab.com.
1     IN    PTR         namesvr.ourlab.com.
eof
```

10.10.11.2.4 Zone localhost.zone local forward file

```
$TTL  86400
@     IN    SOA   namesvr.ourlab.com.          admin.ourlab.com.
(
                  1999121001        ;     sequence number
                  8H                ;     refresh rate (8 hours)
```

```
                        2H              ;       retry (2 hours)
                        1W              ;       expire (1 week)
                        1D      )       ;       minimum ttl (1 day)

                IN      NS              namesvr.ourlab.com.
                IN      A               127.0.0.1
```

eof

10.10.11.2.5 Zone named.ca

This file is identical to the one specified above and is too long; hence it does not need to be repeated. No changes are required.

10.10.12 Activating the DNS Server

Finally, we need to either turn on the DNS service or must restart the service after it has been modified.

By default, all of the following will already exist.

1. We need to activate the named daemon. Issue the command to see if it is active:

   $ chkconfig – – list | grep named

   We observe that run levels 3, 4, and 5 are “off”.

2. Now issue the command to activate the service:

   # chkconfig named on

3. Then again issue the command to verify that the service is turned on.

   # chkconfig – – list | grep named

4. Finally, to fully activate the service, issue the command:

   # xinetd

10.10.12.1 Restarting the DNS Server

You must also issue the restart command on named:

 # /etc/init.d/named restart or
 # service named restart

Remember to modify the serial number of each modified zone file or the new configuration will not be loaded by the secondary (slave) DNS server (if such exists).

10.10.13 Client Setup

The client must be properly configured in order to access a DNS server.

REQUIRED

Client /etc/host.conf file

```
# /etc/host.conf for    ourlab.com
# Lookup names via DNS first then fall back to the /etc/hosts file

order   hosts,   bind                   ; or may be   bind   hosts
```

OPTIONAL

```
        # The following configurations are not required and are generally
        # not included
        # We don't have machines with multiple addresses
        multi  off                                  ;
        # check for IP Address spoofing
        nospoof  on                                 ;
        # and warn us if someone attempts to spoof
        #alert  on                                  ;
        # Trim the ourlab.com domain name for host lookups
        trim  ourlab.com                            ;
eof
```

Client /etc/resolv.conf file

```
        # /etc/resolv.conf for   ourlab.com
        #
        search ourlab.com
        # Specify our primary name server
        nameserver   192.168.102.150      {this is set to the "real" server IP, which
eof                                       for setting up the DNS server  should be
                                          your own IP}
```

10.10.14 Testing the DNS Server

Running the process **nslookup** is one of the most common tools for testing the operation of the DNS server. Several other utilities are available for installation by installing the rpm distribution of **bind-utils**.

To look up the domain of yahoo.com using nslookup, issuing the command:

$ nslookup yahoo.com

responds:

```
        Server:         208.137.0.177
        Address:        208.137.0.177#53

        Non-authoritative answer:
        Name:           yahoo.com
        Address: 66.218.71.112
        Name:           yahoo.com
        Address: 66.218.71.113
```

nslookup was considered an obsolete command – but will be around forever! (MS Windows only works with nslookup.) The new commands that replace it are **dig** and **host**. Dig provides the user with the full data, whereas host provides a short basic report.

To use host, give the command:

$ host yahoo.com

responds:

```
        yahoo.com. has address 66.218.71.113
        yahoo.com. has address 66.218.71.112
```

To use dig, give the command:

$ dig yahoo.com

responds:

```
; <<>> DiG 9.1.3 <<>> yahoo.com
;; global options:  printcmd
;; Got  answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18358
;; fla  gs: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 5, ADDITIONAL: 5

;; QUESTION SECTION:
;yahoo.com.                    IN      A


;; ANSWER SECTION:
yahoo.com.          1106    IN      A       66.218.71.113
yahoo.com.          1106    IN      A       66.218.71.112

;; AUTHORITY SECTION:
yahoo.com.          9561    IN      NS      NS1.yahoo.com.
yahoo.com.          9561    IN      NS      NS2.yahoo.com.
yahoo.com.          9561    IN      NS      NS3.yahoo.com.
yahoo.com.          9561    IN      NS      NS4.yahoo.com.
yahoo.com.          9561    IN      NS      NS5.yahoo.com.

;; ADDITIONAL SECTION:
NS1.yahoo.com.          9578    IN      A       66.218.71.63
NS2.yahoo.com.          5402    IN      A       209.132.1.28
NS3.yahoo.com.          169645          IN      A       217.12.4.104
NS4.yahoo.com.          169645          IN      A       63.250.206.138
NS5.yahoo.com.          169645          IN      A       64.58.77.85


;; Query time: 63 msec
;; SERVER: 208.137.0.177#53(208.137.0.177)
;; WHEN: Tue May 21 10:24:42 2002
;; MSG SIZE  rcvd: 229
```

In the header we can observe what may be expected in the remainder of the message.

```
->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18358
;; fla  gs: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 5, ADDITIONAL: 5
```

We see that we have made a query, there were no errors, for one query we received two answers, five authorities and 5 additional lines of information. At the bottom of the response we also observe that the responding DNS server was 208.137.0.177 and was 229 bytes long.

10.10.15 Troubleshooting DNS

All of the above is nice – if it works. When doing the "dig" command, you get a question section, but no answer section. You have a configuration problem. Make sure that the DNS server is thoroughly tested to insure that any other station or query from the Internet works properly, otherwise other Internet problems will be created.

1. For the dig, did it respond with a query time – server with the IP address of another system? Verify the setup of the **/etc/resolv.conf** file.
2. Look at the end of the **/var/log/messages** file (use tail). Now you need to search the report for possible problems. If there is a simple error, you will be able to see the file that is in error and the line number – now just count down the lines in the file and look for the error. Since you just restarted the service, the problem will be at the end. You should be able to observe where the service was stopped and then restarted – plus some error messages.
   a. Are all files included and spelled correctly? This includes your typing in the **/etc/named.conf** file and all of the **/var/named/** files.
   b. If you see the notation "filename:XX:", the XX represents the line number within the file "filename". Check for typo errors and proper syntax.

There is one application tool that is available for testing the various files of the DNS configuration. The test is issued as:

$ nslint (options)

where the options are:

-c filename	Specifies a different filename if the default named.conf file is not used
-d	Verbose display of performed checks

If nslint is not installed, it may be obtained from the site **ftp://ftp.ee.lbl.gov/nslint.tar.gz** or from **rpmfind.net** .

10.11 DHCP Server

We often wish to assign an IP address to a host on an automatic basis without having to manually insert the address. This might be beneficial if we have a limited number of IP addresses and a greater number of hosts that are not active all of the time. Because the dhcp protocol is used during the boot up process, it communicates by utilizing the Internic's MAC address.

DHCP stands for Dynamic Host Configuration Protocol. It is used to control vital networking parameters of hosts (running clients) with the help of a server. DHCP is backward compatible with BOOTP. For more information see RFC 2131 (old RFC 1531) and other. (See Internet Resources section at the end of the document). You can also read DHCP FAQ (**http://web.syr.edu/jmwobus/comfaqs/dhcp.faq.html**).

The following is derived from the mini DHCP HOWTOs.

10.11.1 dhcp Server Installation

The first thing that we must do is to install the dhcp service to our system. Issue the commands:

```
cd /mnt/cdrom/RedHat/RPMS/
rpm -ivh dhcp-2.0b1pl6-6.i386.rpm        {substitute the correct version}
```

The above rpm file may differ in revision number from the one shown here, assuming you have a newer issue of the file.

If the service is already installed, you will get an appropriate message indicating such.

One file must be modified and a second created.

Once again, we can use the yum installation process:

```
$ yum –y install dhcp*
```

10.11.1.1 dhcpd.conf File

Create the following file if it does not already exist in the **/etc** directory:

```
nano dhcpd.conf
                                          comments not put in file
default-lease-time 600;                   lease time is 10 minutes
max-lease-time 7200;                      maximum time is 2 hours


option subnet-mask 255.255.255.0;
option broadcast-address 192.168.102.255;      this is for our lab
setup
option routers 192.168.102.1;             we do not have a router,
                                          so comment out
option domain-name-servers 192.168.102.188;
option domain-name "ourlab.com";          tell the system our domain
                                          name
ddns-update-style ad-hoc;                 Version 9 update only
subnet 192.168.102.0 netmask 255.255.255.0 {
        range 192.168.102.201 192.168.102.250;
}
```

This specifies that we want the stations to be in the address range of 201 to 250. If we wanted to provide two ranges of address range, then we need the following lines instead.

```
range 192.168.102.101 192.168.102.130;
range 192.168.102.201 192.168.102.230;
```

This will allow assignment of IP addresses between 101 to 130 or 201 to 230. Addresses between 131 and 200 would not be assigned, we assume that they are assigned as static addresses to specific hosts.

Since we are in a Lab environment, we do not need to put the line in.

Be very careful of your typing, syntax is very important and an error will prevent the server from operating. In particular, make sure you use the curly brackets rather than parentheses.

When a client host requests an IP address, one will be assigned from the 192.168.102.201 to .250 range. IP addresses between 0 to 200; and 251 and above will not be issued to requesting hosts.

In our example, if we had 80 hosts, only the first 60 requesting station would be issued an IP address, the rest would not be issued one, and hence will not be operational over the network.

10.11.2 dhcpd.leases File

Before dhcp can work, it must have someplace to store the address information that it has assigned. This file is only created. Linux uses this file to keep what IP addresses have been assigned to which machine. It opens, writes, reads, and closes, but is not able to create the file on its own. Issue the command:

```
# touch   /etc/dhcpd.leases
```

10.11.3 Starting / Restarting Service

Before we can utilize the service, it must be activated. Remember that there may only be one dhcp server on the network.

```
$ chkconfig   - - list | grep dhcp
  dhcpd              1:off   2:off   3:off   4:off   5:off   6:off
$ chkconfig   dhcpd   on
$ xinetd
```

The service is now activated to turn on when a system boots.

Now we need to restart the service, specifically because we have made changes to the configuration.

```
$ service   dhcpd   restart
```

The dhcp server is now operational.

10.11.4 dhcp Client

DHCPd configuration under RedHat 5.0+ is really easy. All you need to do is start the Control Panel by typing:

control-panel or from an X Windows terminal, enter **netcfg**.

at the command line. Then:

- **Select "Network Configuration"**
- **Click on Interfaces**
- **Click Add**
- **Select Ethernet**
- **In the Edit Ethernet/Bus Interface select "Activate interface at boot time" as well as select DHCP as Interface configuration protocol**

Alternatively, you can also use the **netcfg** application. This interface is a little easier to use and performs the same thing. (Unfortunately, the netcfg application is no longer supported in Red Hat 7 and later – great application, major loss.)

If you have some normal number under inet. addr you are set. If you see 0.0.0.0 don't despair, it is a temporary setting before dhcpcd acquires the IP address. If even after few minutes you are seeing 0.0.0.0 please check out

``troubleshooting''. DHCPd is a daemon and will stay running as long as you have your machine on. Every three hours it will contact the DHCP server and try to renew the IP address lease. It will log all the messages in the syslog (**/var/log/syslog**).

One final thing. You need to specify your nameservers. There are two ways to do it, you can either ask your provider to provide you with the addresses of your name server and then put those in the /etc/resolv.conf or DHCPd will obtain the list from the DHCP server and will build a resolv.conf in /etc/dhcpc. This example to uses DHCPd's resolv.conf by doing the following:

Back up your old /etc/resolv.conf

$ cp /etc/resolv.conf /etc/resolv.conf.OLD

If directory /etc/dhcpc doesn't exist create it

$ mkdir /etc/dhcpc

Make a link from /etc/dhcpc/resolv.conf to /etc/resolv.conf

$ ln –s /etc/dhcpc/resolv.conf /etc/resolv.conf

There is a program (dhcp) that may be used to test the dhcp server. By issuing the appropriate command, the client transmits a dhcp request, and view the responses. The program is obtained from **www.mavetju.org/download dhcping-1.2.tar** (get latest version). After installation, issue the command:

$ dhcping –s 192.168.102.255

Some stations may require their specific IP address to check.

10.11.5 Testing a DHCP Server

An application exists to test the operation of a DHCP server, called dhcping. Installation is specified in the Appendix.

10.11.6 Testing a DHCP Client [9]

A DHCP Client may be tested by issuing the command:

$ pump (options)

where the options include:

-c file	Specifies different configuration file if not the default **/etc/pump.conf**
-h host	Host name of request
-i interface	Interface to be configured
-k	Terminates the program
-l	Leases time that is requested
-R	Renew lease on expiration
-s	Interface status
-d	Stops /etc/resolv.conf file from changing
-?	Help

The utility may be found on the **rpmfind.net** site.

[9] Linux Networking, Smart Certify Direct, Thompson Learning

10.12 Mail Server[10]

Probably the number one functional usage of the Internet is email. When the Internet was first established (slightly before Vice President Gore took credit for it), it was used to transfer mail and files between various universities and the military using several mail programs and an ftp program. The transfer of mail between mail servers utilizes service port 25 (smtp protocol). The collection of mail by the client uses either the POP3 or IMAP protocols, using service port 110 or 220.

There are two mail server applications that we will review. The oldest is **sendmail**, the grandfather of the mail servers. We will also review a newer mail server called **postfix**. Another option is **qmail**, which has similarities to postfix (i.e. much simpler to configure), but that will not be reviewed here.

There are actually two different programs that make the full email process work. These are the MUA and MTU agents. In addition, there are three protocols that are involved.

10.12.1 Sendmail Installation

Installation of Sendmail is straightforward. To install using the rpm files, find the multiple sendmail files on the appropriate CD and issue the command:

rpm –ivh sendmail*

Alternatively, yum may be used to install sendmail. To install, issue the command:

yum –y install sendmail*

Upon completion of either method, the application will be completely installed, including the pop and imap protocols for mail retreval.

10.12.2 Mail Software

There are three different types of mail programs that are required to make the transfer of mail work. They are the **MUA, MTU,** and **MDU**.

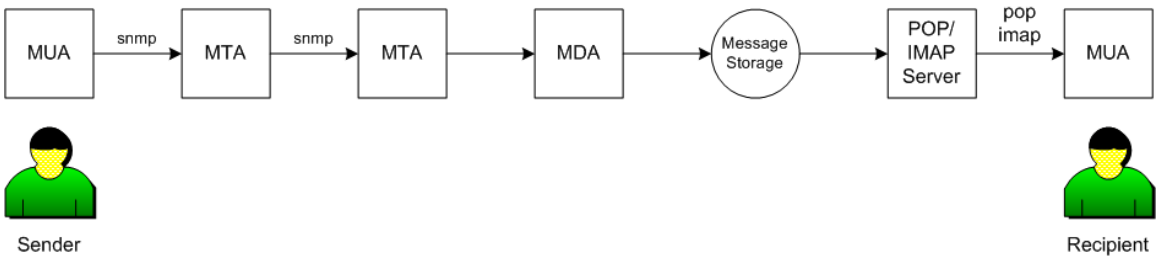


Figure 10-5: Mail Delivery Agents

10.12.2.1 Mail User Agent

The **MUA, Mail User Agent**, is the program that allows the user to create and receive mail. An example is Netscape Communicator. When working in the CLI mode, a common program is **mail**. This is covered elsewhere in more detail later. From a remote system, a user uses either a POP3 (present level) or IMAP protocol for transferring mail from a remote mail server to their system.

[10] Linux Administration – A Beginner's Guide; Steve Shah; Osborne – McGraw Hill

POP3 (or the older POP2) is used to transfer mail to a local system and then automatically delete it from the mail server. IMAP is used to read mail off of the server without removing it from the server. If you have a standalone system to collect your mail, then you typically want to configure your system for POP3, whereas if you are traveling and may use multiple computers, you would want to use IMAP.

10.12.2.2 Mail Transfer Unit

The **MTU, Mail Transfer Unit**, is responsible for accepting, transferring, and holding mail until it can be delivered. Sendmail is an example of an MTU. Whereas the MUA is used to only collect the mail, the MTU actually transfers the mail across the Internet between mail servers.

10.12.2.3 Message Delivery Agent

The last agent is the **MDA**, or **Message Delivery Agent**. This particular application is responsible for setting up the mail for delivery to the recipient. As we will see, it takes received mail at the recipient's ISP and prepares if for delivery by storing it in the appropriate location so that the recipient may collect it. The storage location may be the mail server, or another server that is set up exclusively for mail delivery.

10.12.3 Mail Protocols

Three protocols are used to transfer the email. These are ***Simple Mail transfer Protocol (SMTP), Post Office Protocol (POP), and Internet Mail Application Protocol (IMAP)***.

10.12.3.1 SMTP

SMTP, or **Simple Mail Transfer Protocol**, is used by a mail server to transfer messages from itself to another mail server. It is not responsible for making mail readable. SMTP uses Port 25 for information transfer.

A distinction must be made, as there are commonly two SMTP applications, one for transferring data between mail servers, and a second that is used to transfer mail from the sender's MUA to the MTA (sender to mail server). Both use the SMTP protocol. In our discussion, the application that transfers mail between mail servers may be either Sendmail or Postfix for our discussion. The software used to transfer mail from the MUA to the MTA may be any number of application packages, including MS Outlook, Netscape, Thunderbird, Mozilla, Eudora, Sendmail (user's software, not MTA), Mail, and many others.

10.12.3.2 POP

POP, or Post Office Protocol, is the protocol for delivering and reading a user’s mail, and is incorporated into virtually all client mail programs. POP (specifically version 3) uses Port 110.

As an example, you create a mail message on your computer using the **sendmail** or **mail** program. When you connect to your mail server, the mail is forwarded to the recipient’s mail server using the SMTP protocol, and held in a queue until it can be delivered. Finally, the recipient connects to their mail server and collects the mail from the mail server using the POP3 protocol.

Typically, when the recipient collects his or her mail, it is copied to the recipient's system and automatically deleted from the POP3 server.

10.12.3.3 IMAP

The alternative to using the POP is IMAP, or Internet Mail Application Protocol. One commonly observes this protocol when connecting to Hotmail, Yahoo, or Netscape (and many others. Using a web browser, one observes the received mail and reads it. The read mail is not deleted from the server, or downloaded to the recipient's system. The recipient must manually delete the mail in order to eliminate it.

10.12.4 MTU Mail Application

Two mail server programs are reviewed, **sendmail** and **postfix**.

10.12.4.1 Sendmail

One of the earliest mail programs was Sendmail. Over the years it has been improved and enhanced. Today it is a very robust program – but it comes at a stiff price of being fairly difficult to manage for the enhanced features. Because Sendmail was one of the first programs to be developed – and because it is fundamentally free, it is the most widely used program. Sendmail is considered a monolithic program, that is, it is a single program that does everything, and has one primary configuration file. This concept is what makes the program very difficult to work with.

For a basic default configuration, there are three parts that need to be addressed for **sendmail** to work.

1. Configuration file
2. Queue directory
3. Alias file

10.12.4.1.1 Mail Queue

When messages are received by **sendmail**, they must be queued in one of two locations.

If a message is to be forwarded to another MTU, it is stored in the **/var/spool/mqueue** directory. Queuing may be due to the remote MTU not being available (server or link down) or the cost of delivery may be less at a different hour.

When a message is received at the terminating MTU it must be stored until the recipient MUA can download it. These messages are stored as a single file in **/var/spool/mail/** directory.

In order for these files to be accessible by the **sendmail** program, several requirements must be set up for specific directories. Each of the following directories must be set up as owned by ***root*** (chmod) and not writable by either group or world (other) (chmod go-w or chmod 755).

/usr **/var** **/var/spool** **/var/spool/mqueue** and **/var/spool/mail**

10.12.4.1.2 Sendmail Configuration File

The Sendmail Configuration file provides all of the ”rules” of how to send the mail. Simply stated, this is a very complex process that requires extensive research to fully understand all of the features available. For now, the default basic configuration will work fine with a few minor changes. It has been said that you are not a network administrator until you have tackled Sendmail – and never want to do it again.

The Red Hat installation of Sendmail is pre-configured as a single-server, single-office server with spam filtering enabled. This will demonstrate a basic system without extra options – which are numerous. The user is referred to ***Sendmail*** (O'Reilly & Associates) (commonly known as the "Bat Book").

10.12.4.1.3 sendmail.cf File

The configuration file is **sendmail.cf**, located in the **/etc/mail** directory. This file is quite long and very difficult to read or understand, but after six months of diligent study . . . .

If the sendmail server is unable to resolve our Fully Qualified Domain Name (FQDN), we need to make one change to our file. Using a text editor, open **/etc/sendmail.cf**, and find the lines (use ^W in nano):

```
#  my official domain name
#… define this only if send mail cannot automatically determine your
domain
#Dj$w.Foo.Com
```

Then change the last line to read (this is optional):

```
Dj$w.ourlab.com
```

For our lab, we do not need to change the setting since the system is able to resolve the domain name. Remember, this is the FQDN, not the mail server host.

Naturally for your own company, you would put in your own domain name.

As an option, sendmail allows the server to masquerade for another domain. Find the lines that read:

```
# who I masquerade as
DM
```

And we could change the second line to read:

```
DMourlab.com
```

Again, we do not need to change the line as we are not masquerading as any other system.

Finally, our server can act as the mail server for multiple domains, such as if the ISP provided commercial web hosting. There are two options for configuring this feature, internal to sendmail.cf (hard way), or through a pointer to an external file, where the domain names are listed (easy way). RedHat is pre-configured to look for an external file. This is typically located just preceding the domain name:

```
Cwlocalhost
# file containing names of the hosts for which we receive mail
Fw/etc/mail/local-host-names
```

One change that is required if we wish to have remote systems access our mail server. The default setting for mail service is to limit access only from the single system, that is one must telnet to the server, but that is not the normally desired process. We must find the location of the following line:

```
O   DaemonPortOptions=Port=smtp, Addr=127.0.0.1, Name=MTA
```

Do a search for '127.0.0.1'. Now we need to comment out the above statement and add one that looks identical – but without the Addr=127.0.0.1 portion. We should have the following:

 # Removed single system
 # O DaemonPortOptions=Port=smtp, Addr=127.0.0.1, Name=MTA
O DaemonPortOptions=Port=smtp, Name=MTA

We can now send and receive mail from a remote system with respect to sendmail.

10.12.4.1.4 local-hosts-names File

Now edit the **/etc/mail/local-hosts-names** file. This is a simple, short file, as it lists the names of our domain. Set up the file to appear as the following:

 # /etc/mail/local-hosts-names – include all aliases for your machine
mx.ourlab.com {put in your station host name instead of the "mx"}
mail.ourlab.com
ourlab.com

The final minimum configuration is to make sure that **/etc/mail/local-hosts-names** file contains the following:

localhost
localhost.ourlab.com

We are basically all set for a simple mail server.

10.12.4.1.5 Alias File

Sendmail has the ability to forward mail. This is a database type file that provides for address conversion. For this review, it will be considered an advanced topic and not discussed further. For now it is empty.

10.12.4.1.6 DNS Server Requirements

Previously we set up the DNS server to support our network operation. We need to make sure of a setting before the sendmail server will operate properly for our domain. Our entry in the forward domain zone file (domain.com.zone) for the mail server should appear like the following:

IN MX 0 mail.FQDN

Make sure it reads MX 'ZERO' – not the letter "O"

If you put in the name of the server host, DNS will think you are specifying the FQDN as the host name.

It is important that the Forward Domain File be properly configured for INternet Addresses. In our example here, we are using our own host to serve as the mail server, so we need to set the Addresses as (top part not shown):

Forward Domain Zone File

	IN	**NS**		**namesvr.ourlab.com.**
	IN	**MX**	**0**	**mail.ourlab.com.**
namesvr	**IN**	**A**		**192.168.102.{name-server-Station-#}**
mail	**IN**	**A**		**192.168.102.{mail-server-Station-#}**
web	**IN**	**A**		**192.168.102.{web-server-Station-#}**

```
ftp     IN   A       192.168.102.{ftp-server-Station-#}
www     IN   CNAME   web.ourlab.com.
tftp    IN   CNAME   ftp.ourlab.com.
ns      IN   CNAME   namesvr.ourlab.com.
mx      IN   CNAME   mail.ourlab.com.
```

For the Reverse Domain File, we need to make sure of the following configuration:

Reverse Domain Zone File

```
@     IN   NS    namesvr.ourlab.com.
150   IN   PTR   namesvr.ourlab.com.    ;{put in real IP Address}
151   IN   PTR   mail.ourlab.com.       ;{put in real IP Address}
152   IN   PTR   web.ourlab.com.
153   IN   PTR   ftp.ourlab.com.
```

Make sure that you set the last octet to the proper value for each of the servers, this very important for the mail server or the boot process will be delayed.

Notice that if we do a dig on mx.ourlab.com, it will point to the mail.ourlab.com server, which in turn points to the namesvr.ourlab.com server. This is required in our example because we are using just one host. In normal operation, these would probably be separate systems.

10.12.4.1.7 Activating Sendmail Server

Before we can use the mail server, we need to turn the services on and then restart the service. Recall that there are three protocols that need to be activated for a full system.

To do this, we issue the commands:

```
$ chkconfig  sendmail  on
$ chkconfig  pop3  on
$ chkconfig  imap  on
$ xinetd
```

Remember that you can check to see if the services are on by running the chkconfig - - list command first.

10.12.4.1.8 Restarting Sendmail

After we have made the changes to the sendmail files (.cf and .cw), we need to restart the server. Issue the command:

```
$ /etc/rc.d/init.d/sendmail  restart        or
$ service  sendmail  restart
```

10.12.4.2 Postfix Mail Application

The second mail server application that will be reviewed is **postfix**. The installation of postfix is covered in Appendix E1, as it must be downloaded as source code and then complied. Later versions of Red Hat and Fedora now include Postfix in the installation package. This is not a difficult process and may be easily performed if the basic directions are followed.

10.12.4.2.1 Postfix Configuration

After the source code has been compiled, there are two files that need to be configured, **/etc/postfix/main.cf** and **/etc/postfix/master.cf**.

10.12.4.2.1.1 main.cf File Modifications

Four modifications need to be implemented to the **/etc/postfix/mail.cf** file. The first action required is to determine the fully qualified hostname, that is, it contains both the system name and the domain name. As an example, this can be determined by issuing the command:

$ hostname
L49.ourlab.com

1. INTERNET HOST AND DOMAIN NAMES
   #myhostname = host.domain.tld
   Change to:
   myhostname = {hostname}.ourlab.com
2. INTERNET HOST AND DOMAIN NAMES
   #mydomain = domain.tld
   Change to:
   mydomain = ourlab.com
3. DELIVERY TO MAILBOX
   #mail_spool_directory = /var/spool/mail
   Change to:
   mail_spool_directory = /var/spool/mail
4. NETWORK ADDRESS
   #mynetworks = 168.100.189.0/28, 127.0.0.0/8
   #mynetworks = $config_directory/mynetworks
   #mynetworks = hash:/etc/postfix/network_table
   Add at the bottom of this list:
   mynetworks = 192.168.102.0/24, 127.0.0.0/8

Save the file and exit.

Point one is used if the hostname command returns both the host and domain names. If it does not, then the point two must be used, otherwise it is not required as Postfix is capable of extracting the two names. It will not hurt to properly set the mydomain name value, and will insure proper operation. Point three is normally only required if the mail_spool_directory is different from the standard configuration.

Everything should now be operational, but for the short term, we want to test the operation. To do this, we will make a temporary modification to the **/etc/postfix/master.cf** file. Edit the line and fine the following section (near the top of the file):

```
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (50)
# ==========================================================================

smtp      inet  n       -       n       -       -       smtpd -v
. . .
```

Note that there are two entries for smtp, the one shown (which is immediately below the comments) and one at the end of the list that specifies ***smtp unix***, do not modify the line containing “unix”.

The system will now produce a very verbose log of each mail sent. This is for testing only and we do not want to leave it on permanentaly. When we have completed our testing, and know that postfix is operating properly, the “ -v” will be deleted.

For a minimal system that works, that is all that is required for the configuration.

10.12.4.2 Postfix Activation

Before the mail server may be fully configured, the DNS must be properly configured. Go back to section 10.12.3.1.5, DNS Server Requirements to set up the appropriate zone files.

Next we need to activate the service. But before we can activate, we must direct **chkconfig** to add the service. We first issue the command:

```
# chkconfig – –add postfix
```

Now we will get the proper response when we issue the command:

```
# chkconfig – – list | grep post
postfix          0:off 1:off 2:off 3:off 4:off 5:off 6:off
```

Then issue the commands:

```
$ chkconfig postfix on
$ chkconfig – – list | grep post
postfix          0:off 1:off 2:on 3:on 4:on 5:on 6:off
$ xinetd
```

After the file has been modified, the postfix must be restarted. Issue the command:

```
$ service postfix restart
```

By default on a normal system, **Sendmail** is active, allowing the delivery of mail on the localhost. This is required to allow messages that are created by the system to be sent to the logged in user. An example might be a **cron** job that sends its results to the user, or security applications that mail reports to the administrator. Postfix will replace this function, so Sendmail must be deactivated. Issue the command:

```
# chkconfig sendmail off
```

10.12.4.3 POP and IMAP

With the upgrade of Fedora, **qpopper** is no longer utilized A newer application is used to provide the same functions, this being **dovecot**. This package provides both pop3 and imap4rev1. It requires minimal memory, and is considered “fail safe”. When using IMAP, it supports extensions for SORT, THREAD, and IDLE. Additionally, security through TLS / SSL supported, including IPv6.

If dovecot is not included, then it may be installed using yum. First verify that dovecot is installed:

```
# chkconfig – –list | grep dove
```

dovecot 0:off 1:off 2:on 3:on 4:on 5:on 6:off

If dovecot is not turned on, then issue the **chkconfig dovecot on** command. After which, verify that pop is operational. Issue the command:

 # chkconfig – –list | grep pop
pop-before-smtp 0:off 1:off 2:on 3:on 4:on 5:on 6:off

Make sure that the **xinetd** command is issued to insure that the server table is re-read.

10.12.5 Client Mail Applications

There are a number of client mail programs that are able to function for the user. The main difference being the ease of use and what the user likes.

The **mail** program was originally designed as a MUA that resided on the mail server, and the user logged on to the mail server via a telnet session. It has subsequently been upgraded so that the user can configure **mail** to collect their mail from a remote mail server.

10.12.5.1 Sending Mail the Hard Way on your Mail Server

There are definitely easier ways of sending mail, but here we want to show a little more of what happens.

10.12.5.2 Create Mail Example

First we want to create a new user – yourself (if you have not done this previously in a lab for Chapter 3). Add the username {first-initial}{last-name} (jdoe for John Doe). Set the passwd with the same name.

Next we need to create a file to send. Just to keep things simple, lets change to your home directory. Create a file called **firstmail** with the following (nano firstmail).

This is generated by {your name}

10.12.5.3 Sending Mail Example

Now issue the command:

$ /usr/lib/sendmail {your-userid}<firstmail

You have just sent the message as ***root***, now logout and log back in under your username. Before we properly read the mail, lets look where the file resides.

10.12.5.4 Reading Mail Example

Change to the directory **/var/spool/mail** and do a listing. You should find a file there with your username. Display this file. You should find something like the following:

From root {day date time}
Return-path: <root>
Received: (from root@localhost)
by localhost.localdomain {dotted number} id {8 digit hex number}
for {user-id}; {day date time}
Date: {day date time}

From: root<root@localhost.localdomain>
Message-Id: <date.ID#@localhost.localdomain>

This is generated by {your name}

Now we want to read the file "officially". Logout as ***root*** and back in as your userid (jdoe). You should have a notice that you have new mail.

Issue the command:
$ **mail**

You will have something like the following message:
Mail version 8.1 6/6/93 Type ? for help
"/var/spool/mail/{your id}": 1 message 1 new
> N 1 root@localhost.local {day date time}
&

Type in a **1** to read the mail, you will get:
Message 1:
From root {day date time}
Date: {day date time}
From: root <root@localhost.localdomain

{your message}
&

Now type **q** to quit and you get
Saved 1 message in mbox

Since you are at your home directory, do a listing and you will note a file called **mbox**. Display the file **mbox**. You will see the same message with the added line:
Status: RO
meaning the file is read only.

Go back to **/var/spool/mail** and list your file. You will now note that it is empty. Your message has been transferred to the **mbox** file.

10.12.6 Using the MUA Mail Program – needs to be verified

The CLI **mail** program is a very sophisticated mailer, although it requires commands to make it function. When used in the direct command mode ($mail), commands are always issued at the "**&**" line header.

When starting **mail**, the default operation is the read mode. When there is incoming mail, a header will be displayed for each message. Typing the message number will display the mail.

After reading a message, you can delete or reply to the message. To delete, type a **d** at the **&**; to reply, type an **r** and you will automatically go into the text entry mode.

For a start, lets send a mail to our own username (make sure your username exists on the system). We issue the command:

```
$ mail  dennis
Subject:   Test Mail
This is a test mail to myself.
.
Cc:{Hit Enter}
$
```

Now, change to the **/var/spool/mail** directory and look at the contents of your file (in my case "dennis"). You will now see the message that you sent.

Next we want to look at the postfix log file. Change to the **/var/log** directory, and if using postfix, display the contents of the maillog file. Every message that is transmitted is now logged with a multitude of information. Remember that when setting up postfix, we specified that we wanted verbose logs.

OK, the above was a valid mail message, next we want to send one to an invalid user. Issue the following:

```
$ mail   noonehere
Subject:   Bad Mail
This mail better not work.
.
Cc:{Hit Enter}
$
```

We know this is a bogus mail, and in fact we should have received a message to our logged in username telling us this. But lets again go back to the **/var/log/maillog** file. Now if we display the contents, and do some searching, we will observe the following line (somewhere):

```
...
>>> CHECKING RECIPIENT MAPS <<<
Jun 24 12:24:55 longgrain postfix/smtpd[3322]: ctable_locate: leave existing entry key
    noonehere@longgrain.dearroz.net
Jun 24 12:24:55 longgrain postfix/smtpd[3322]: maps_find: recipient_canonical_maps:
    noonehere@longgrain.dearroz.net: not found
Jun 24 12:24:55 longgrain postfix/smtpd[3322]: maps_find: recipient_canonical_maps:
    noonehere: not found
...
```

This is great. We have transmitted a bogus message and the system told as much.

To create a new message, start mail with the command:

mail   username@somesystem.com

Make sure that the user-id is another user on a different system.

You will be put into the text entry mode. You will first be required to enter a Subject, then you can enter the message text. After entering the text message, terminate the message by entering a period (**.**) on a new line. This will bring up the **(Cc)** prompt. At this point you can enter another name or hit enter to exit back to the CLI prompt.

OK, lets see what has happened. Our own system is the mail server, so we are waiting to forward our mail to another system. Change to **/var/spool/mqueue** and do a listing using the **ls –l** command. We see two files with the file names of the type:

df{message-id}

qf{message-id}

note the owner of each file.

Listing the **df-file**, we observe:

{test message}

Listing the **qf-file**, we observe:

[message header}

Many more options are available, check out the **man mail** page for details.

Now that the system is up and operational, we want to remove the verbose output. Go back to the **/etc/postfix/master.cf** file and remove the modification that was previously made.

```
smtp      inet  n       -       n       -       -       smtpd -v
. . .
```

Lets look at a diagram of how email messages flow across the Internet.

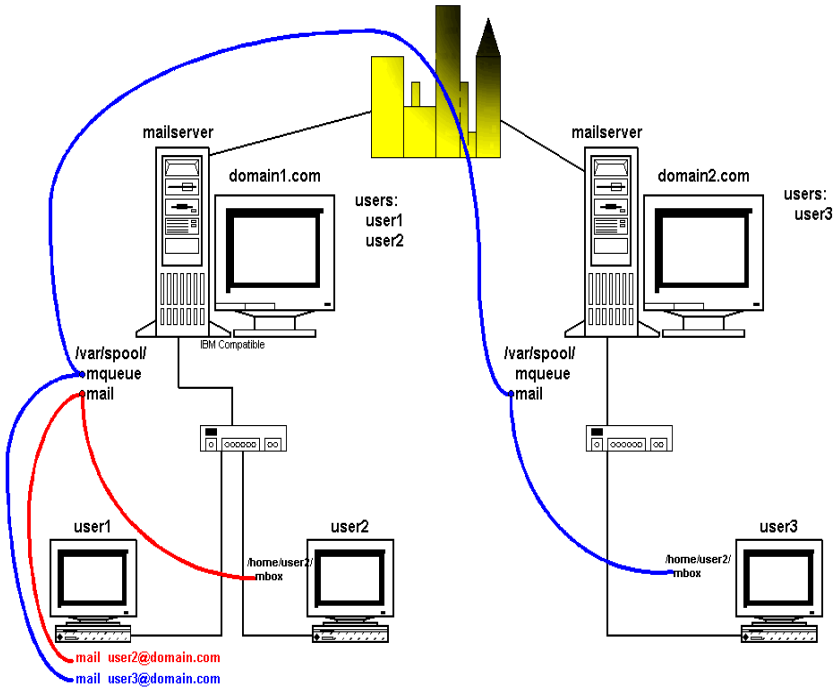


Figure 10-5: Mail Flow

10.12.7 Using Thunderbird for your Mail Program (must confirm)

Start X Windows and open a terminal window. Enter the command:

thunderbird & (this starts Thunderbird in the background) or

select thunderbird from the start – Internet menu

On the first time you must set up the mailer configuration. The following screens need to be set.

The first screen assumes that no data will be collected from another application.

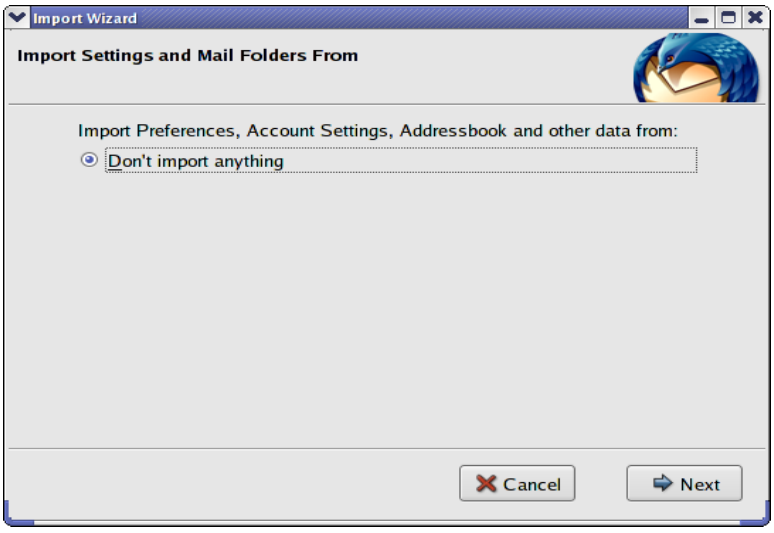


Figure 10-6: Thunderbird Import Settings

Next, we need to specify that we are setting up an email account.

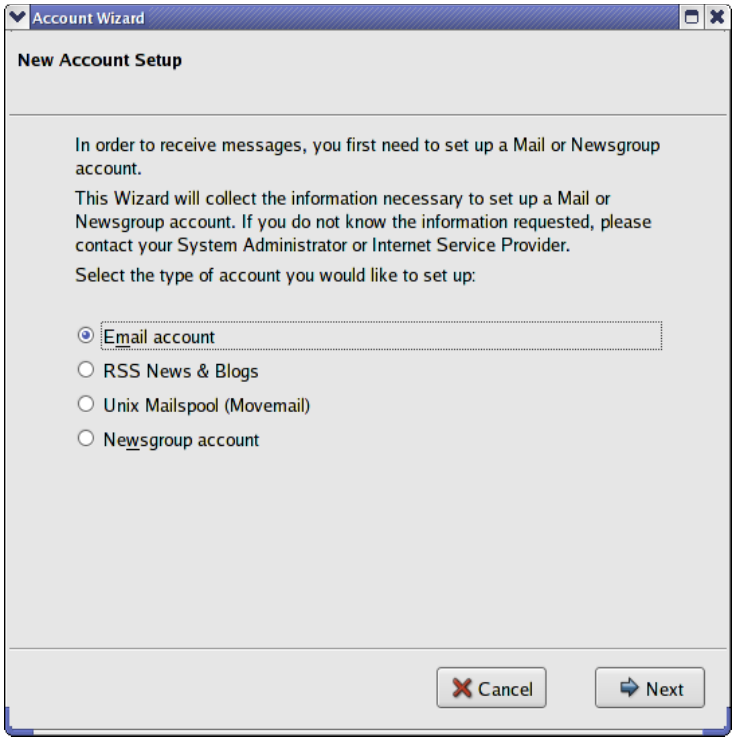


Figure 10-7: New Account Setup

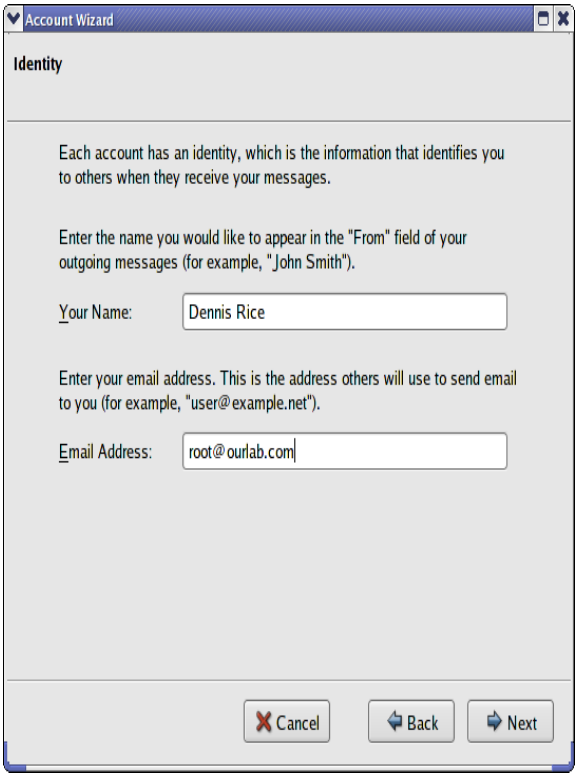


Figure 10-8: Account Identity

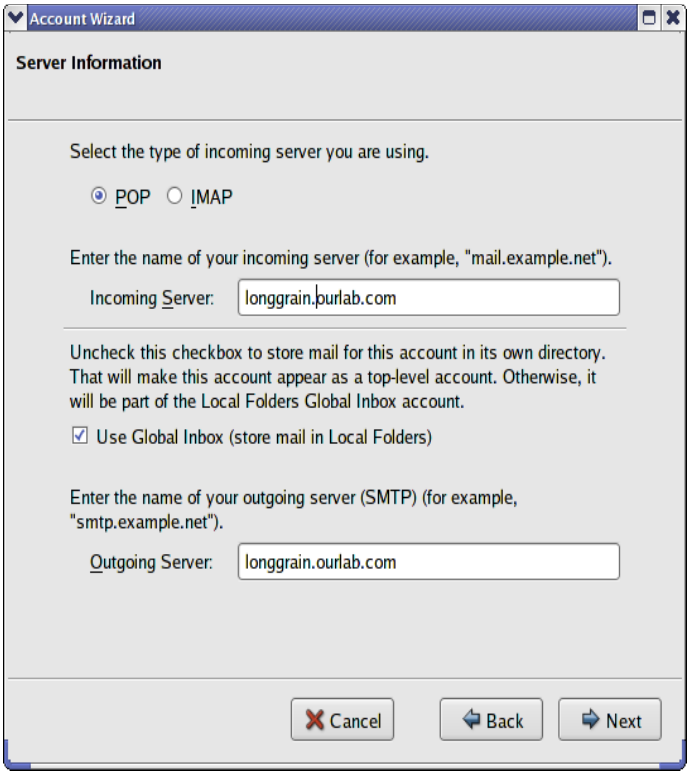


Figure 10-9: Server Identification

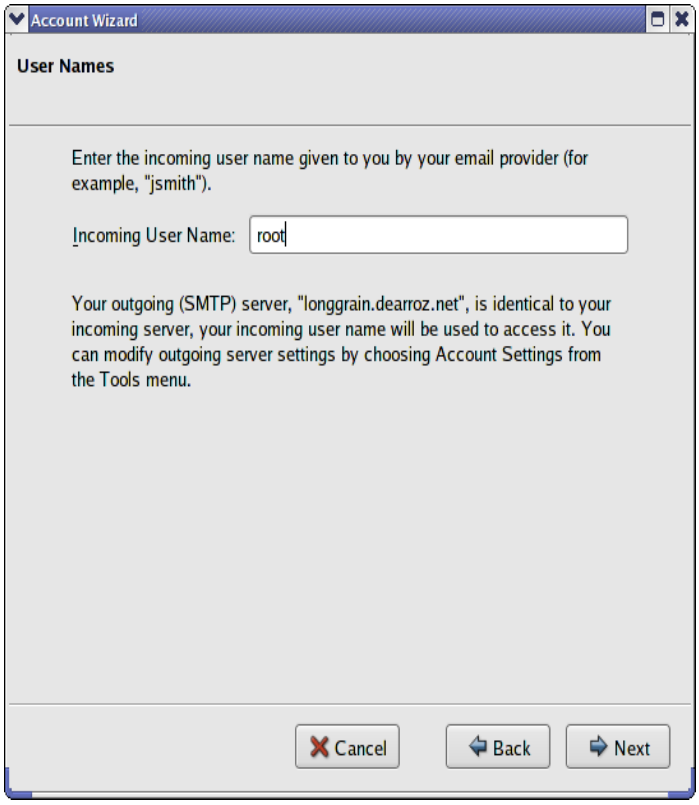


Figure 10-10: User Name

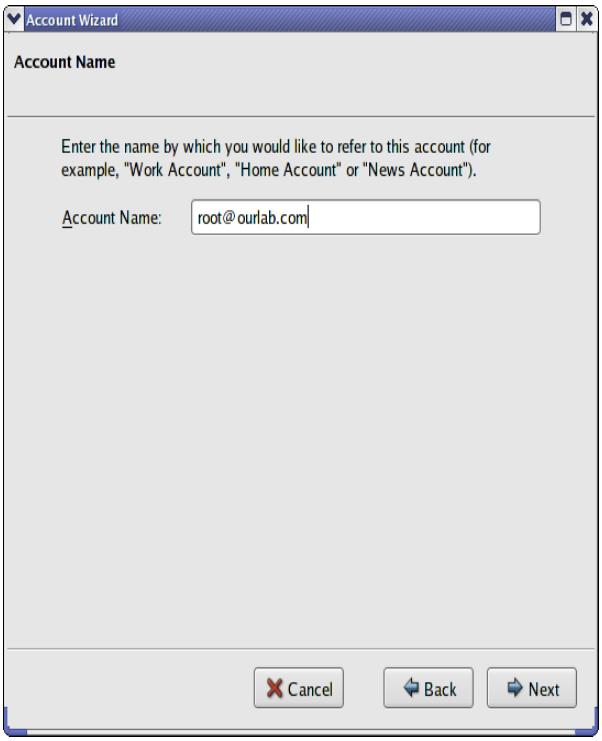


Figure 10-11: Account Name

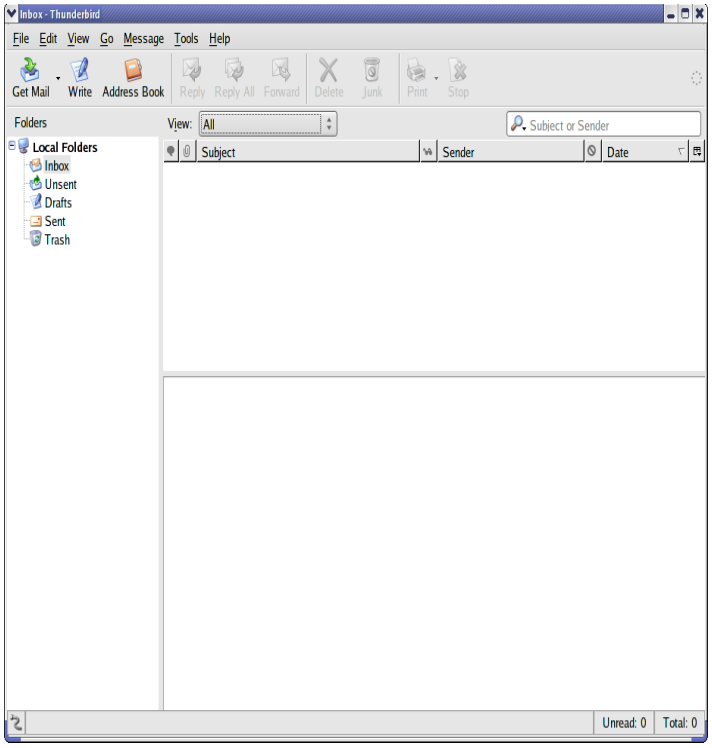


Figure 10-12: Thunderbird Screen

Thunderbird has now been configured for our server. To test out the connectivity, send a mail to yourself and then collect the mail.

SENDMAIL NOTES

3 parts

- Mail User Agent (MUA)
  - Mail client – lets user compose and read mail messages
    - Netscape, pine, mutt, modzila
- Mail Transfer Agent (MTA)
  - Transfers mail between mail servers, does not send or receive mail from user
    - sendmail, qmail, procmail
- Local Delivery Agent (LDA) or Message Delivery Agent (MDA)
  - Retreives mail from an MTA and stores it in the user's spool mailbox (/var/spool/mail/username)
    - procmail, fetchmail

Mail Protocols

- SMTP – Simple Mail Transfer Protocol
  - Used by MTA to send and receive mail between mail servers
- POP3 – Post Office Protocol (version 3)
  - Used by LDA to retrieve messages from MTA and store them on the user's host, then delete them from the MTA
- IMAP4 – Internet Message Access Protocol (version 4)
  - Used by LDA to read mail on an MTA, allowing the mail to be retained on the server

MIME – Multipurpose Internet Mail Extensions
SMTP designed for ASCII text, does not support other binary file types. MIME encodes binary files (sound, graphics) as attachments

10.13 MySQL Database Server[11]

MySQL is a basic and powerful relational database program. It is commonly included with the book distribution or with common commercial versions, it is available off the Internet for free.[12] It is important to understand the basic concept of what a SQL database engine provides, as it is the foundation of how a business maintains its operational data. The database is a very powerful tool, allowing management to generate various reports that display appropriate information by which the progress of the business may be managed.

Two issues will be addressed in this section. First will discuss the installation and basic operation of MySQL, second a very simple primer will be show of a simple database and its basic operation. The second portion is to allow one to understand the function of a database and how it might be used. It is not intended to be anything near a comprehensive discussion of the SQL language, this is a totally different topic which requires considerable additional study. Here we only provide a very simple understanding of the power of an SQL database.

MYSQL is available from a multitude of sources, including **http://www.mysql.com** or from http://www.redhat.com . Two separate packages must be downloaded – the server and client. Two versions for each package are available, compilable and rpm binary. Here we will assume that you obtain the rpm binary version for the rest of the process.

10.13.1 Required Files for MySQL

Linux includes four files to support the installation of MySQL. They are:

1. mysql-3.23.36-1.i386.rpm
2. mysqlclient9-3.23.22-4.i386.rpm
3. mysql-devel-3.23.36-1.i386.rpm
4. mysql-server-3.23.36-1.i386.rpm

Number 3 is not required, but we will install it because the process is easier.

10.13.2 Installation of MySQL

Installation of both the server and client packages is simple and straightforward. If you did a full installation, the following should not be necessary.

Issue the command:

rpm –ivh mysql*.rpm

you will observe:

Preparing . . . . ###### . . .
1: mysql ###### . . .

[11] Red Hat Linux 7 Server, Mohammed J. Kabir

[12] MySQL is free with some limitations, most simply you cannot make a profit off of it. Do not sell it, package it with a product that is sold, or charge to administer it. You can charge to write scripts to create and maintain databases that use MySQL. The fee is very low, and goes to support the improvement of MySQL.

2: mysql-devel ###### . . .
3: mysql-server ###### . . .
4: mysqlclient9 ###### . . .

This installs MySQL in the following directories:
MySQL client operational program:
/usr/lib/mysql
MySQL server program:
/usr/share/mysql/mysql.server
Administration:
/usr/bin/mysqladmin
Configuration:
/usr/bin/mysql_config
Documentation:
/usr/share/doc/mysql-{version}
Database Data:
/var/lib/mysql/{dbname}/datafiles

Back again, lets do the installation the easy way. Issue the command:
$ yum –y install mysql*

10.13.3 Adding Perl CGI to MySQL

To complete the installation, assuming that we wish to utilize the perl programming language to enhance screen presentations, we need to add the capability to create CGI scripts for the Perl programming language. The same may also be done for the Python program. Issue the command, allowing the use of perl scripts, enter the following:

perl –MCPAN –e shell

and answer the questions:

Ready for manual configuration [yes]
Build and cache directory [/root/.cpan]
Cache size [10M]
Cache scanning [at start]
Policy on prerequisites [follow]
gzip program [/bin/gzip]
tar program [/bin/tar]
unzip program [/usr/bin/unzip]
make program [/usr/bin/make]
lynx program [/usr/bin/lynx]®
ncftpget program [/usr/bin/ncftpget]
ftp program [/usr/bin/ftp]
pager program [/usr/bin/less]
shell [/bin/bash]
Parameters for "perl Makefile.PL" [ ]
Parameters for "make" [ ]
Parameters for "make install" [ ]
Timeout for inactivity [0]
ftp-proxy [ ]
http-proxy [ ]

no-proxy [ ]

The above is for Red Hat 7.2, later versions have a more comprehensive list of questions, but provide the same functionality. Perl is an excellent means to improve the look and operation of MySQL, particularly from a web page.

10.13.4 Activating MySQL

Now that MySQL is installed, it must be activated. Following the standard process, the following commands are issued:

```
# chkconfig --list | grep mysql
   mysqld          0:off  1:off  2:off  3:off  4:off  5:off  6:off
# chkconfig mysqld on
# xinetd
```

Now we need to process a series of commands to start MySQL.

 # /etc/rc.d/init.d/mysqld restart or
 # service mysqld restart

and we will observe the shutdown (fails) and restart (OK) responses. MySQL will automatically start on a new bootup.

10.13.5 Testing MySQL Operation

Now test MySQL to make sure it is working:

$ /usr/bin/mysqladmin ping

and you will observe:

mysql is alive

Note that the mysqladmin ping command must be issued prior to creating a database, as shown above. If it is issued after the initial user is created (below), then the command must be modified to request a password, by using the command:

$ mysqladmin ping –p

Next issue the command to observe that it is a process:

$ ps aux look at the bottom of the list for lines containing "mysql".

Finally we need to set up a set of passwords for MySQL.

The first command has a long output (with respect to the screen), and we desire to observe it, so issue the command:

$ /usr/bin/mysql_install_db | less

Next, issue the command to create a password for the root administrator:

$ /usr/bin/mysqladmin –u root password 'password'

The password must be enclosed in the single quotes. The first "password" is a line command that specifies that the word is to be encrypted.

Naturally, in the normal operation you would put in your own private "password", in this case, so we can work through any problems, use the actual word 'password'. Make sure you put it into either single or double quotes.

One very important word of caution, DO NOT FORGET THE ROOT PASSWORD! You will be sorry if you do. Suggest that you create a file in your home directory to save the MySQL password. (We will get around this a little later on.)

10.13.6 Starting MySQL on a Client Host

Now that the server function is up and running, we need to start the client process. For now, you are logged on as root, so issue the command:

$ mysql –u root –p

at the password prompt, type **"your-password"**

You will observe an opening message, then the prompt

Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 3 to server version 3.22.15 .
Type 'help' for help.
mysql >

An option not shown in the welcome prompt is to end some commands, with "**\G**". When displaying information from a **select** command, the "\G" will provide a long listing vertically rather than the typical horizontal format.

10.13.6.1 Quitting MySQL

As a quick test, enter **quit** (or exit) at the prompt to exit mysql.

All commands issued to mysql must either end with a "**;**", "**\g**", "\G" or for the command to be forwarded to the server.

10.13.7 Looking at the Internal Base Database

Now that MySQL is installed, we will describe a little bit of the operation of the application. It is not necessary to really understand the administration of a database, but it is necessary to understand what is happening by those that use it. Lets start mysql again, and enter your password. After you have logged in, at the **mysql>** prompt, enter:

```
>show databases;          < remember to end with the semicolon
Databases
Mysql
1 row in set (0.00 sec)

>use mysql;
>show tables;
Tables_in_mysql
colums_priv
db
func
host
tables_priv
user
```

6 rows in set (0.00 sec)

If you should enter a command incorrectly, and end up with the "->" prompt, then enter a " ; " to return to the "mysql>" prompt.

>show columns from user;

Field	*Type*	*Null*	*Key*	*Default*	*Extra*
Host					
User	*values as noted*				
…					
Alter_priv					

17 rows in set (0.01 sec)

>select host, user, password from user;

host	*User*	*Password*
localhost	*root*	*encrypted*
{hostname}.ourlab.com	*root*	
localhost		
{hostname}.ourlab.com		

4 rows in set (0.00 sec)

Hostnames should differ on your system.

Now lets add you as an individual user to the database. When we insert data into the database table, we need to specify the entry in one of two formats – every field in the table, or by specifying the specific fields to be updated. Recall that the user table has 17 fields, so we must specify a value for each. We want these to be:

Host	'localhost'
User	'{your-first-name}
Password	'devry'
Select_priv	'Y'
Insert_priv	'Y'
Update_priv	'Y'
Delete_priv	'Y'
Create_priv	'Y'
Drop_priv	'Y'
Reload_priv	'Y'
Shutdown_priv	'Y'
Process_priv	'Y'
File_priv	'Y'
Grant_priv	'Y'
References_priv	'Y'
Index_priv	'Y'
Alter_priv	'Y'

We therefore need to issue the command:

insert into user values ('localhost', '{your-name}', PASSWORD ('your-password'), 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y', 'Y','Y','Y','Y','Y','Y','Y','Y');

Remember to place single quotes around "your-password".

The password function encrypts your password within the table using the RD4 encryption process. This is not the greatest encryption process (it is the one that Microsoft uses), but it does work. Now go back and again display the host, user, and password contents of the user (use the up arrow so you do not have to retype the full command).

To enter the data for just a limited number of fields, use the command:

insert into user (host, user, password) values ('localhost', '{your-name}', PASSWORD ('your-password'));

The above command is a painful process to enter, but all of the "Y"'s are required to provide all privileges to your username. What do you do if you type an error. You do not need to enter each line all over again, you can use the arrow keys to repeat the commands and make corrections.

To enter another user, such as a classmate, using the up arrow to repeat the insert command, use the back-arrow to move the cursor back through the insert command – delete your name and insert a classmate's name – hit ENTER. Again go back and look at the users. Since you added two users with the same password we want to look at what the encrypted passwords look like. Under mysql, they have the same encrypted password – not as secure as Linux, but at least the password is not obvious.

It is obvious that this is quite painful to utilize. That is the penalty that you must pay to utilize a SQL. By using other programs such as Perl or Python, one can create GUI screens that significantly simplify the process of data entry or reading.

Our system for MySQL is now set up and running. Our next section will be on setting up a simple database. Further enhancements will be in additional sections to develop Perl scripts.

Least it be said, we will now provide a few very simple examples of running MySQL, this is not intended to be anything near to an education in operating a SQL relational database system. That is another full term course (at least). Your position is to maintain the server and to understand the basics of its operation, not the administration of the database data entry or query.

10.13.8 Accessing the SQL Server

After the server system has been installed, it should start automatically on bootup. Check the **/etc/rc.d/init.d** directory for the **mysql** file. If you need to start or stop the mysql server, issue the appropriate command:

10.13.9 Database Design

Two examples for setting up a database are given. The first will be very simple, the second slightly more complex.

Next we need to specify how a database table is structured. Visualize a database table layout of columns and rows, where each row is a unique set of information, but each column contains like information. This can be shown as:

Record 1	Field1	Field2	Field3 . . .
Record 2	Field1	Field2	Field3 . . .
Record 3	Field1	Field2	Field3 . . .

The values inserted into a specified field are all alike in the type of information that they contain. For example, Field1 might contain the last name of a person, Field2 might contain the first name, Field3 the individual's birthdate.

10.13.9.1 Simple Database Example

A database is a collection of tables, where each table is a collection of information that has a similar format. Each table consists of records made up of multiple fields. Below we create a simple database table in design. Later we will create an actual table. Typically each field is of a specified length, hence each record is also of a constant length.
Lets create an example database table:

Content	Field Name	Length	Type
Last Name	lstname	30	Character
First Name	fstname	30	Character
Major	major	5	Character
SSN	ssn	10	Character
Date of Birth	dob	8	Date

Here we have a database table that has five fields, with each specifying the type of field and its length. When creating a table, make sure the individual fields are long enough to hold the desired data, otherwise the entered information will be truncated.

Now each record will appear something like:

:lstname (30):fstname (30):major(5):ssn (10):dob(8) or more realistically:

```
Doe++++++++++++++++++++++++++++John++++++++++++++++++++++++++++TCOM+123-45-6789+19711025
\------------------lstname----------------------/\---------------------fstname-----------------/\ major /\-----ssn-------/\---dob---/
```

The “+” signs represent a blank space in the field.

Each individual that you enter into the database will have a record in the above format. There is a little more to the format, but it is basically what we have above. Every record must be unique! Otherwise we will have a confusion as to which record we are looking at when we do a search.

10.13.9.2 A Little More Complicated MySQL Example

One of the most obvious requirements for a database is the need for an address book. We will create a simple database that we can later add features to for our common use.

First we need to discuss what a database is. Our definition is:

> A database is a collection of one or more tables of like information formatted into common groups that can be sorted and retrieved in a manner that is convenient to the user.

Each Record contains a number of fields, which collectively are unique, and that for a specific field, similar information for each field is maintained.

Now we need to define what type of information is maintained in our database that we might wish to store. We will create a basic system, which may be enhanced for one’s personal use. From the basic definitions above, we need to add some additional rules:

1. A field may be one of the following (others are available, but this is a start):

a. Character
b. Numeric
c. Integer
d. Date
e. Memo

2. Each field has a fixed length, which must be long enough to store all anticipated information. If the field is not long enough, the extra data will be truncated.

The basic table that we desire to set up is as follows:

Field	Function	Name	Type	Length
1.	Last Name	lstname	Char	20
2.	First Name	fstname	Char	20
3.	Address 1	add1	Char	80
4.	Address 2	add2	Char	80
5.	City	city	Char	30
6.	State	state	Char	3
7.	Zip Code	zip	Char	10
8.	Phone (h)	hphone	Char	12
9.	Phone (w)	wphone	Char	12
10.	Email	email	Char	40
11.	Key	id	Num	–

We now have our basic list of fields. In this example all are character except for the "key" field, which will insure that all records are unique. Even though the fields of zip and phone are basically numeric values, we generally make them character based so we can put in a hyphen or have leading zeros, making them sortable.

Now lets create our database using MySQL. After the client MySQL has been started, issue the command:

create database address; *(remember to put in the ";")*

Query OK, 1 row affected . . .

Thus we have our basic database established, but we need to create a table of our fields that we wish to use.

Even though we have created the database, the system does not know that we want to use it (we might want to use another). Issue the command:

```
use address;
create table addlist (
lstname     char(20)    not null,
fstname     char(20)    not null,          terminate each line with
add1        char (80),                     a comma
add2        char (80),
city        char (30),
state       char (3),
zip         char (10),
hphone      char (12),
wphone      char (12),
email       char (40),
primary key (lstname, fstname) ) ;          do not forget the ;
```

This list tells us that:

lstname Is a character field 20 characters in length, and it can not be null (empty)

fstname Is a character field 20 characters in length, but it may be empty

primary key (lstname, fstname)
This specifies that the indexing keys for the database is composed of two fields.

Now we have our database table fully set up. Our next step is to add data to it. Lets see how MySQL displays our table. Issue the command:

describe addlist;

and we get:

Field	***Type***	***Null***	***Key***	***Default***	***Extra***
lstname	***char(20)***		***PRI***		
fstname	***char(20)***		***PRI***		
add1	***char(80)***	***Yes***		***Null***	
add2	***char(80)***	***Yes***		***Null***	
city	***char(30)***	***Yes***		***Null***	
state	***char(3)***	***Yes***		***Null***	
zip	***char(10)***	***Yes***		***Null***	
hphone	***char(12)***	***Yes***		***Null***	
wphone	***char(12)***	***Yes***		***Null***	
email	***char(40)***	***Yes***		***Null***	

Now lets add some data to our table, issue the command:

insert into addlist (lstname, fstname, add1, city, state, zip, wphone) values (“Ourbiz”, “Corp”, “1 Second Place”, “Our City”, “NW”, “99999”, “800-555-1234”);

Notice that all of the values have been enclosed in quotation marks – that is because they are character type fields. The quote marks may be either single (') or double (“) We do not have to enter all of the field data for a specified record, but we must specify which fields we wish to enter data into. When we created the table, recall that we specified two fields as being “not null”, data must be entered for these two fields, otherwise an error will be displayed to the user.

This is a very difficult process – to say the least. Remember that we are at this time entering data under the Command Line Mode. Later, you can add scripts to the process that can provide a simple screen data entry format and automatically creates the data insert command. (This is beyond the scope of this book.)

Now lets list all of the the contents of our table (our one record anyways). Issue the command:

select * from addlist;

This displays all records, we will get:

```
+-------------------+--------------------+
+-------------------------------------------------------------+
+-------------------------------------------------------------+
+------------------------------+-----+----------+------------+
+------------+-----------------------------------------+
|  lstname           |  fstname           +
|  add1                                                      |
|  add2                                                      |
|  city                        |state|  zip     |  hphone    |
|  wphone    |  email                                |
+--------------------+--------------------+
+-------------------------------------------------------------+
+-------------------------------------------------------------+
+------------------------------+-----+----------+------------+
+------------+-----------------------------------+--------+
|  Ourbiz            |  Corp               |
 1 Second Place                                              |
 NULL                                                        |
 Our City                       | NW  | 99999    | NULL       |
 972-929-6777| NULL                                    |
```

Now lets add an additional record:

insert into addlist (fstname, lstname, hphone, email) values (“Doe”, “Jane”, “817-555-9876”, msjane@staywarm.com.il) ;

Then to view just selected fields, issue the command:

select fstname, lstname, city, state, zip, hphone, email, id from addlist ;

Our results are:

```
+---------+---------+-----------+-------+--------+-------------+
+---------------+
| fstname | lstname | city       | state | zip    |  hphone     |
email                          +
+---------+---------+-----------+-------+--------+-------------+
---------------------------------+
|  Corp   |  Ourbiz   | Our City | NW    | 99999  |972-555-1234 |
NULL                           |
|  Jane   | Doe       |  NULL    | NU    | NULL   |817-555-9876 |
msjane@staywarm.com            |
+---------+----------+----------+-------+--------+-------------+
---------------------------------+
```

In the above, the output is formated to make it easy to read, but this is commonly not the case, as the record is commonly longer than the screen.

Note that fields which are blank have the term “NULL”.

The above is a little hard to read, and MySQL provides another means to display the data in a vertical format. Issue the command:

select fstname, lstname, city, state, zip, hphone, email from addlist \G

Our output (truncated) would look like the following:

```
*************************************************************************
```

fstname Corp
lstname Ourbiz

city Our City
state NW
zip 99999
hphone 972-555-1234
email Null
.....Record 2.....
fstname Jane
lstname Doe
city Null
state NU
zip Null
email msjane@staywarm.com.il
.....

10.13.10 Recovering Lost Root Password

Previously, it was noted that when creating the MySQL root password, that you must not forget it, or else. Well, there is a magic method to recover from that problem.

OK, you've lost the root password, and we previously said you would be in real trouble, well, we told a small fib. It can be recovered and we the following is how to do it. Perform the following steps.

1. Stop MySQL
   From the administrator's prompt:
   # service mysqld stop
2. Start MySQL in the safe mode, setting it to not read the “grant” tables that contains all of the MySQL database passwords.
   # mysqld_safe --skip-grant-tables &
   [5555] *(this is the service number – yours will differ)*
   Starting mysqld daemon with databases from /var/lib/mysql
3. Issue the **mysqladmin** command to reset the administrator's password. As an example, we will set it to the word “mine”:
   # mysqladmin -u root flush-privileges password “mine”
   The privileges for the root administrator has now been reset.
4. Restart the MySQL daemon:
   # service mysqld restart
   Stopping MySQL: 001234 10:41:35 mysqld ended
   [OK]
   Starting MySQL: [OK]
   [1] + Done safe_mysqld --skip-grant-tables

The administrator's password has now been reset and the system can now be accesses.

10.13.11 MySQL Database Backup

One must always protect their data. The best approach is to make a backup of the database. To perform this, issue the following command:

$ **mysqldump - - add-drop-table -u [username] -p '[password]' [database] > [backup_file]**

As an example, one might backup the address database as:

```
$ mysqldump - - add-drop-table -u myname -p 'password'
      address > addressbkup.sql
```

You have now created a backup database.

Now that you have a backup, you need to be able to restore the database if something goes wrong. Issue the command:

```
$ mysql --u [username] -p '[password]' [database] <
      [backup_file]
```

To restore our previous example:

```
$ mysql –u myname -p 'password' address < addressbkup.sql
```

Remember that if your database has been lost, then the main “mysql” database has probably also been lost. So restore it too.

10.14 Print Server [13]

It is not practical to provide a printer to every user, so it is often found that one printer with either a network Ethernet port or an Ethernet to Parallel port converter is often used. The problem is that the administrator does not have total control over this method. An alternative method is to install a Print Server that provides a larger spooler (queuing directory) and control over who is allowed to utilize it. There are two methods by which a server may be set up, LPRng and CUPS.

10.14.1 Print Server Configuration using LPRng [14]

The original method of configuring a printer, either via serial or parallel connection, was to configure the **/etc/printcap** file. This file is typically configured by using the **printtool** application, but may also be modified manually for special attributes.

An example of a printcap file might be:

```
ljet4|lp|ps|Postscript|600dpi 20MB memory|Rm 315|local|LPT1:\
:lp=/dev/lp0:rw:\
:sd=/var/spool/lpd/ljet4:mx#0:pl#72:pw#85:\
lf=/var/log/lpd-errs:if=/usr/local/cap/ljet4:
```

Explaining each line:

- Line 1: Each printer may be assigned multiple different names, each pointing to the same physical device (the same name may not be used on two different printers). Each name is separated by a pipe character (|). In this example, various names have been used to describe the type, resolution, and location of the printer (nice idea). The line is terminated with the “ :\ ” characters. The colon specifies the end of a field, and the forward slash specifies that the next field is located on the next line.
- lp: Specifies the device driver for the printer.
- rw: Printer is read-write capable, thus the printer can issue status commands to the Linux OS.

[13] Hello Linux, Clyde Boom, Lancom Technologies
[14] Mark Allen, www.comptechdoc.org/os/linux

sd: Specifies the print spool directory.
mx: Specifies the maximum size of a print job, in this case the size is unlimited.
pl: Specifies the page length – 72 lines.
pw: Specifies the page width – 85 characters.
lf: Specifies the log file if an error occurs.
if: Specifies the input filter if one is required for the printer.

Additional parameters that may be configured include:
br: Used to specify the bit rate of a serial port.
sh: A Boolean value to specify the suppression of a header.
rp: An alternative to lp, specifies a network printer.
rm: An alternative to lp, specifies a printer connected to a remote host.
rs: Specifies that only designated account users may use the printer.
rg: Specifies that only designated group users may use the printer.

10.14.2 Print Services using CUPS

Configuring the system to utilize the CUPS configurator for an attached printer. Before proceeding, backup the following files in the **/etc/cups** directory for your protection in case of an error:

cupsd.conf □ **cupsd.conf.org**
client.conf □ **client.conf.org**

10.14.2.1 Print Server Configuration using CUPS

From an XTerm window, perform the following:

1. On the print server, log the IP Address for use on the client systems.
2. Edit the **/etc/cups/cupsd.conf** file.
   a. Search for the line **</BrowseAddress>**, find the sixth occurance until you locate the following line:
      # BrowseAddress @IF(name)
   b. Add a blank line immediately below the above specified line. You may want to enter a comment line (a line starting with a # character) to document what you are doing.
   c. Add the following line directly below, left justified:
      BrowseAddress 192.168.102.255
      You will normally use your own network address. The last octet of 255 is the broadcast address. This is the address that announcements will be broadcast to from the server. Here we are announcing to all other hosts on the network that this system is the print server.
   c. Add a blank line immediately below the new line just entered.
   d. Search for the following lines:
      <Location/>
      Order Deny,Allow
      Allow From 127.0.0.1
      <Location/>

e. To allow all hosts on the network access, add the following line below the localhost address (127.0.0.1):

Allow From 192.168.102.*

If only one other host were to be permitted access, then the following line would be added – as an example, assume that the desired host has an IP Address of 192.168.102.201 .

Allow From 192.168.102.201

The above set of statements allow the clients to access the printer. Additional, each Client is able to see the CUPS opeing screen, the Printers screen (displays printers that exist) and the Jobs screen (displays jobs that are in queue).

f. The next set of changes allow a user that is logged in to the print server to observe their print jobs without authenticating (logging in as the administrator). To pause or cancel their own jobs, the user must authenticate.
Search the file for the following lines:

<Location /jobs/>
AuthType Basic
AuthClass User
Order Deny,Allow
Deny From All
Allow From 127.0.0.1
</Location>

To allow the user access, add the following statement at the end of the block (just above the </Location> line):

Allow From 192.168.102.201

If it is desired to allow full access without authentication, change the AuthType to:

AuthType None

The AuthClass User permits a user to pause or cancel their job after authentication. If the setting is:

AuthClass System

then a user must be a member of a groupname called **sys** in order to modify the job status.
In the above statement blocks, the statements generally have the following meaning:

- **Order Deny,Allow** Specifies the order of the statements in the block.
- **Deny From All** Initially deneys access from all other hosts on the network.
- **Allow From 127.0.0.1** Allows access from the **localhost**, that is the print server.
- **Allow From 192.168.102.201**

Allows access from the single host.

g. Admin – 19-39

Save the modifications and exit. This completes the configuration of the Print Server using the CUPS configurator.

10.14.2.2 Client Configuration using CUPS

The next task is to configure the Client host to direct print jobs to the print server. Perform the following tasks:

1. Edit the **/etc/hosts** file. Add the following line at the end of the file:
   192.168.102.XXX ps.ourlab.com pserver
   Where XXX is the last octet of the print server.
2. Save the file and exit.
3. Test the connection by performing a network connection test:
   ping pserver

   You should observe a response to the pings. Terminate the ping with a ^C.
4. Assuming that the Client is also running CUPS, edit the **/etc/cups/cupsd.conf** file. Search for the line:
   /BrowserPoll

   Add a blank line below this line and then add:

10.15 FAX Server

10.16 Commands Used in this Chapter

cat	Display one or more files, or create a new file from direct keyboard input
cd	Change Directory
chgrp	Change the group of a file or directory
chkconfig	Check the operational or modify the status of a service
chmod	Change the permissions of a file or directory
chown	Change the owner of a file or directory
control-panel	A utility for configuring various attributes of the system
copy (Cisco)	Utility within a Cisco Router to copy a configuration
cp	Copy a file or directory
dig	Utility to determine the IP address of a remote server from DNS
exportfs	Exports NFS configuration to the etab file
ftp	File Transfer Protocol utility
groupadd	Create a new group
host	Utility to determine the IP address of a remote server from DNS
htpasswd	Password utility for http web access
ln	Link utility
ls	List directory contents
m4	Utility for configuring sendmail
mail	User utility for sending and receiving mail
mkdir	Creates a new directory
mount	Mounts a removable device such as a CDROM
mysql	Utility to operate the MySQL database
mysqladmin	Utility to administer the MySQL database

mysqldump	Utility to backup a MySQL database
nano	Basic text editor
nslookup	Utility to determine the IP address of a remote server from DNS
ntsysv	Utility to check the operation or modify the status of a service
passwd	Utility to modify the password of a user
perl	Interpreter programming language
pico	Basic text editor
Pring Configurator	GUI utility to configure a printer from X Windows
printconf	Utility to configure a printer from the command line
printtool	Utility to configure the printcap file
ps	Displays active processes
rpcinfo	Displays RPC Information
rpm	Installation package manager
sendmail	Utility for sending mail or one of the Internet mail server daemons
service	Utility for starting, stopping, or restarting an Internet service
serviceconf	Utility to check the operation or modify the status of a server
smbclient	Utility to test the operation of a samba server
smbpasswd	Utility to modify the password for samba
testparm	Utility to check the configuration of the smb.conf file
touch	Utility to create a file of zero length or contents
umount	Utility to unmount a removable drive such as a CDROM
useradd	Create a new user on the system
usermod	Utility to modify a users attributes
vi	Basic text editor
xinetd	Utility to re-read the Internet services status

10.17 Chapter Review Questions

1. You require a directory on a Unix / Linux server to shared with other Unix / Linux hosts. What server function is required? b
   a. nfs
   b. samba
   c. ssh
   d. telnet
2. What is the Internet Protocol for an ICMP message? a
   a. 1
   b. 6
   c. 8
   d. 17

3. The preferred GUI interface for turning on and activating a server service is what? a
   a. ntsysv
   b. serviceconf
   c. service
   d. server
4. The Internet Protocol is specified in which OSI Layer? a
   a. Data Link Layer
   b. Network Layer
   c. Physical Layer
   d. Transport Layer
5. The Internet Service is specified in which OSI Layer? b
   a. Data Link Layer
   b. Network Layer
   c. Session Layer
   d. Transport Layer
7. What is the Internet Service Port for SSH? b
   a. 1
   b. 22
   c. 23
   d. 80
8. A socket comprises what entities? b
   a. IP Address
   b. IP Address and Service Port
   c. MAC and IP Address
   d. MAC and Service
9. In order to turn a service on, what command is issued? b
   a. chkconfig service
   b. chkconfig service on
   c. chkconfig – –list | service
   d. service on
10. Which mail application provides mail service between the mail server and the client? b
    a. IMAP
    b. MTA
    c. MTU
    d. POP
11. What is the Internet Service port for the DNS Named Service? D
    a. 1
    b. 21
    c. 25
    d. 53
12. What Run Level is the CLI mode? B
    a. 2
    b. 3
    c. 4
    d. 5

13. In which directory are system level web pages maintained? D
    a. /home/html
    b. /home/username/public_html
    c. /var/html
    d. /var/www/html
14. To access a user's personal web page, what URL is used? d
    a. /URL
    b. /URL/index.html
    c. /URL/username
    d. //URL/~username
15. What is the default web page file name? a
    a. /var/www/html/index.html
    b. /var/www/html/index.txt
    c. /var/www/page.html
    d. /var/wwwweb.txt
16. Which server function provides a conversion between a URL and an IP address? b
    a. DHCP
    b. DNS
    c. HTTP
    d. SNMP
17. What is the DNS file that specifies the zone files? a
    a. /etc/named.conf
    b. /etc/dns.conf
    c. /etc/zone.conf
    d. /var/named/named.conf
18. What is the file that specifies the local reverse zone? a
    a. /etc/0.0.127.in-addr.arpa.zone
    b. /etc/named/0.0.127.in-addr.arpa.zone
    c. /var/named/127.arpa.zone
    d. /var/named/0.0.127.in-addr.arpa.zone
19. What is the syntax of a DNS zone file's address record? b
    a. hostname A IP-Address
    b. hostname IN A IP-Address
    c. IP-Address IN A hostname
    d. IP-Address A hostname
20. Which DNS file provides pointers to the DNS Root servers? C
    a. /var/named/named.ca
    b. /var/named/chroot/var/named/cache.na
    c. /var/named/chroot/var/named/named.ca
    d. /var/named/chroot/var/named/named.root
21. Which file contains the DNS IP-Address for a client? B
    a. /etc/resolve.conf
    b. /etc/resolv.conf
    c. /etc/named/resolv.conf
    d. /var/named/resolv.conf

22. What file must be created for DHCP that specifies the IP-Address range? A
    a. /etc/dhcpd.conf
    b. /etc/dhcp/dhcpd.conf
    c. /etc/dhcpd/dhcpd.leases
    d. /var/dhcpd/dhcpd.leases
23. Which server application provides mail service between mail servers? B
    a. MTA
    b. MTU
    c. SMTP
    d. SNMP
24. Which mail protocol provides for automatically downloading of mail from the server? B
    a. IMAP
    b. POP
    c. SMTP
    d. SNMP
25. Which mail protocol retains the mail on the server? A
    a. IMAP
    b. MTU
    c. POP
    d. SMTP
26. In which directory does a mail server retain undelivered mail? D
    a. /etc/spool/mail
    b. /home/username
    c. /var/mail
    d. /var/spool/mail
27. In order for mail servers to work, what other server must be properly configured? B
    a. DHCP
    b. DNS
    c. FTP
    d. Telnet
28. What is a simple client application for a client to send and receive mail? B
    a. fetchmail
    b. mail
    c. qumail
    d. postfix
29. What is a simple database application that comes with Linux? C
    a. Access
    b. Dbase
    c. MySQL
    d. Oracle

30. What is the protocol used to provide web pages? C
    a. dhcp
    b. ftp
    c. http
    d. snmp
31. To configure apache, which file must be modified? C
    a. /etc/httpd/conf/apache.conf
    b. /etc/httpd/apache.conf
    c. /etc/httpd/conf/httpd.conf
    d. /etc/webd.conf
32. What is the top of the DNS node structure called? D
    a. Com
    b. Dot
    c. Node
    d. Root Node
33. To activate a service, what command is issued? A
    a. /etc/rc.d/init.d/service start
    b. /etc/rc.d/init.d/service stop start
    c. /service/init.d restart
    d. /service/init.d start
34. What command is equivalent to /etc/rc.d/init.d/server? D
    a. /etc/init.d/server-name
    b. server
    c. service restart
    d. service servername
35. What is the path / filename to check if there is a problem activating a service? B
    a. /var/log/filename
    b. /var/log/messages
    c. /var/messages/log
    d. /var/spool/log
36. To share a directory between Unix / Linux systems, what path file must be modified? C
    a. /etc/etab
    b. /etc/exportfs
    c. /etc/exports
    d. /etc/mtab
37. What is the application to test MySQL operation? A
    a. mysqladmin
    b. perl –MCPAN –e shell
    c. ps aux
    d. service mysql
38. What is the MySQL command to list available databases? C
    a. list databases
    b. list tables
    c. show databases
    d. show tables

39. What file is used to configure a LPRng print server? D
    a. /etc/client.conf
    b. /etc/cupsd.conf
    c. /etc/cups/cupsd.conf
    d. /etc/printcap
40. What file is used to configure a cups print server? B
    a. /etc/cups.conf
    b. /etc/cups/cupsd.conf
    c. /etc/hosts
    d. /etc/printcap
41. What is the Internet Service port for smtp? C
    a. 1
    b. 6
    c. 25
    d. 110
42. To list the run level status for a specific service, what command is issued? c
    a. chkconfig – –level n service-name
    b. chkconfig – –list
    c. chkconfig – –list | grep service-name
    d. chkconfig service-name
43. Which server provides shared directory content between Unix and Linux systems? B
    a. FTP
    b. NFS
    c. Samba
    d. SMTP
45. What command must be issued to set up a shared directory between Unix / Linux systems? B
    a. exportfs
    b. exportfs –av
    c. exports
    d. exports –av
46. Which server shares a directory of Unix / Linux system to a MS Windows system? C
    a. HTTP
    b. NFS
    c. samba
    d. SMTP
47. Administratively, how should a samba server be set up for user access? B
    a. administrator
    b. group
    c. individual
    d. user

48. What is the command to modify a file's group ownership? B
    a. adduser
    b. chgrp
    c. chmod
    d. usermod
49. What file is modified for the samba server (version 7 and later)? B
    a. /etc/samba.conf
    b. /etc/samba/smb.conf
    c. /etc/smb.conf
    d. /etc/smb/samba.conf
50. What is the command to modify a file's permissions? C
    a. chfile
    b. chgrp
    c. chmod
    d. chperm
51. Which server provides for the transfer of files between systems? A
    a. FTP
    b. MAIL
    c. SMTP
    d. SNMP
52. Which server provides remote booting of an operating system? D
    a. DHCP
    b. FTP
    c. HTTP
    d. TFTP
53. Within the global section of smb.conf, what attribute must be modified to specify a group of users? C
    a. account
    b. netbios name
    c. workgroup
    d. users
54. What file maintains the MS Windows encrypted password for samba (version 7 and later)? C
    a. /etc/passwd
    b. /etc/samba/passwd
    c. /etc/samba/smbpasswd
    d. /etc/smbpasswd
55. What is the command to create the samba password file? B
    a. cat mksmbpasswd.sh > /etc/samba/smbpasswd
    b. cat passwd | mksmbpasswd.sh > /etc/samba/smbpasswd
    c. cat passwd | mksmbpasswd > /etc/samba/smbpasswd
    d. cat passwd > smbpasswd
56. What is the command to test a samba configuration? D
    a. test samba.conf
    b. test smb.conf
    c. testparm samba.conf
    d. testparm smb.conf

57. What privileges does an anonymous user have when using FTP? C
    a. Administrative rights
    b. Directory creation
    c. Download files only
    d. Upload and download files
58. What privileges does a guest user have when using FTP? D
    a. Administrative rights
    b. Download files only
    c. HTTP access
    d. Upload and download files
59. What is the command to list the status of a service? c
    a. chkconfig
    b. chkconfig server-name
    c. chkconfig - - list
    d. service server-name
60. You wish to search for a specific service to determine if it is active. What command is issued? a
    a. chkconfig - - list | grep service-name
    b. chkconfig - - service-name
    c. grep server-name
    d. service server-name | grep
61. After activating a service, what additional command is required to re-read the services file? d
    a. inetd
    b. service server-name restart
    c. service server-name start
    d. xinetd
62. What protocol is used to support Samba? c
    a. nmb
    b. samba
    c. smb
    d. win
63. For Samba, what must the permissions be for the shared directory? b
    a. r w x r w S - - -
    b. r w x r w s - - -
    c. r w x r w x - - -
    d. r w x r w s r w t
64. What Internet Service provides unsecured remote access? d
    a. FTP
    b. HTTP
    c. Secure Shell
    d. Telnet

65. For non-real users to access an FTP server, what is the login name? a
    a. anonymous
    b. guest
    c. real
    d. user
66. What type of secured FTP user is allowed full access to the file system? c
    a. anonymous
    b. guest
    c. real
    d. root
67. What protocol does TFTP utilize? d
    a. IP
    b. MAC
    c. TCP
    d. UDP
68. What is the protocol used for web browsing? b
    a. FTP
    b. HTTP
    c. SMTP
    d. SNMP
69. For a personal web page, in what directory are the html files maintained? b
    a. ~/
    b. ~/public_html
    c. /var/html/public_html
    d. /var/www/html/public_html
70. What is the configuration file for Apache? d
    a. /etc/httpd.conf
    b. /etc/httpd/httpd.conf
    c. /etc/httpd/conf/http.conf
    d. /etc/httpd/conf/httpd.conf
71. What does the acronym DNS stand for? d
    a. Domain Name Saber
    b. Domain Name Selection
    c. Domain Name Server
    d. Domain Name System
72. What is the relationship between a URL and IP Address called when using DNS? a
    a. Bind
    b. Name
    c. Relate
    d. Resolve
73. What is the business primary DNS called? b
    a. Local
    b. Master
    c. Primary
    d. Slave

74. What is the business backup DNS called? c
    a. Backup
    b. Secondary
    c. Slave
    d. Remote
75. What is the daemon name for DNS? d
    a. bind
    b. dnsd
    c. name
    d. named
76. For a Fedora Core system, in what directory are the zone files located? d
    a. /etc/named/
    b. /etc/named/chroot/var/named
    c. /var/named/
    d. /var/named/chroot/var/named/
77. The file that specifies an Internet Service URL to IP Address is known as what? b
    a. Cache File
    b. Forward Domain Zone File
    c. Localhost Forward Zone File
    d. Reverse Domain Zone File
78. The file that specifies an Internet Service IP Address to URL is known as what? d
    a. Cache File
    b. Localhost Forward Zone File
    c. Localhost Reverse Zone File
    d. Reverse Domain Zone File
79. Within a DNS zone file, SOA stands for what? a
    a. Statement of Authority
    b. Statement of Authorization
    c. System of Authority
    d. System of Authorization
80. Within a DNS zone file, NS stands for what? b
    a. Name Selector
    b. Name Server
    c. Naming Service
    d. Notation Selector
81. Within a DNS zone file, MX stands for what? b
    a. Mail Access Unit
    b. Mail Exchange Server
    c. Mail Exchange User
    d. Mail Transport Unit
82. Within a DNS zone file, IN stands for what? c
    a. Interface
    b. Intermod
    c. Internet
    d. Intersection

83. Within a DNS zone file, A stands for what? a
    a. Address
    b. Alpha
    c. Adductor
    d. Attachment
84. Within a DNS zone file, what value must be updated in addition to the updated information? d
    a. Expire value
    b. IP Address
    c. Refresh value
    d. Serial Number
85. What is the daemon name of the DHCP service? c
    a. addressd
    b. dhcp
    c. dhcpd
    d. ipad
86. Within a DHCP configuration file, what specifies the minimum duration time that a host retains an IP Address? a
    a. default-lease-time
    b. max-lease-time
    d. minimum-lease-time
    d. option lease-time
87. For DHCP, which file specifies the configuration? d
    a. /etc/dhcp.conf
    b. /etc/dhcp/dhcpd.conf
    c. /etc/dhcpd.cfg
    d. /etc/dhcpd.conf
88. Within a dhcp configuration file, which line specifies the address range assignment? c
    a. max-lease-time
    b. option
    c. range
    d. subnet
89. For dhcp, what file must be just created? d
    d. /etc/dhcp.conf
    b. /etc/dhcp.leases
    c. /etc/dhcp/dhcpd.leases
    d. /etc/dhcpd.leases
90. For email, a user uses what type of application to communicate with a mail server? d
    a. Mail App
    b. Mail Retrieval Agent
    c. Mail Transport Agent
    d. Mail User Agent

91.
    a.
    b.
    c.
    d.
x.
    a.
    b.
    c.
    d.
x.
    a.
    b.
    c.
    d.
x.
    a.
    b.
    c.
    d.
x.
    a.
    b.
    c.
    d.
x.
    a.
    b.
    c.
    d.
x.
    a.
    b.
    c.
    d.

Chapter Index