

Lab 7: Configuring A Linux File Server

Theory:

The Network File System (NFS) feature provides a means of sharing Linux file systems and directories with other Linux and UNIX computers on a network. Though designed primarily for sharing files between UNIX/Linux systems, there are also Windows versions of NFS that you can use to access an NFS server as well.

With NFS on the server side, you choose which directory you want to share and **export** it. On the client side, a computer can mount the exported directory. Once mounted, the exported directory, as well as all files and directories below that point in the server's file system, appear as though they exist in the client's file system.

On your NFS server, you can choose which computers have access to the exported shared directory, exclude files and subdirectories of that directory that you don't want shared, and define permissions that individual users have to access the shared directory structure.

On the NFS client, the administrator can choose where the shared directory is connected into the client computer's file system, choose if the shared directory is mounted read/write or read-only, and decide if the shared directory is mounted manually or automatically.

Setting Up the NFS Server

To use Linux as an NFS file server you need to set up the `/etc/exports` file. Once that file is configured, you need to make sure that the NFS service is set to start when you boot your system.

Editing the `/etc/exports` File

The `/etc/exports` file contains definitions of the directories you share. To make changes to the `/etc/exports` file, open it in any text editor as the root user. The following is the type of information you need to add to this file:

- ▶ Directory -- The name of the directory you want to share.
- ▶ Computers -- The names of the computers that have access to the shared directory.
- ▶ Options -- Option that identify how the shared directory can be used.

By default, the `/etc/exports` file is empty, so no directories are available for sharing. You can choose any directory (including its subdirectories) that you want to share and define how it is shared.

The following is an example of an `/etc/exports` file:

```
/mydata sales99(rw,squash_uids=0-99) market15(rw)
/usr/project *.yourdomainhere.com(ro)
/home 10.0.0.0/255.0.0.0(rw)
```

The three lines in this example `/etc/exports` file represent three directories that you are making available from your NFS file server: `/mydata`, `/usr/project`, and `/home`. In each case, the host computers that can have access to the share directory are identified either by host name (such as `sales99` or `market15`), domain name (such as `yourdomainhere.com`), or IP address.

Unless a fully-qualified domain name is given (such as `sales99.yourdomainhere.com`), the host is assumed to belong to the local domain. An asterisk (*) can be used to match all hosts in a particular domain. For example, `*.yourdomainhere.com` would match `abc.yourdomain.com`, `def.yourdomain.com`, and so on.

IP addresses can be used individually (such as `10.0.0.10`) or to allow access to all host computers in a network. For example, `10.0.0.0/255.0.0.0` allows access to all computers whose IP address begins with 10 to have access to the resource.

Along with each host or network you allow to access your resource, you can identify a set of **options**. You add options to each host or network by inserting the options between a pair of parentheses after the host or network name. Separate multiple options with commas. In the previous example, the options `(rw)` and `(ro)` were used to indicate that the share was available with **read/write** access or **read-only** access, respectively.

Starting the NFS Service

To start the NFS service so that your shared directories from `/etc/exports` are made available to the network, you can run the `nfs` run-level script. To do this, as the root user, type the following:

```
/etc/init.d/nfs start
```

Note: The `nfs` script does not start the NFS daemons unless you have a valid `/etc/exports` file configured with at least one shared directory.

If NFS starts successfully, you should set the `nfs` start-up script to run automatically when Linux boots. To do this, you need to create a link to the `nfs` script to your default run level directory.

To determine your run level, type the following:

```
runlevel
```

The output of this `runlevel` command tells you your current run level. The run level is probably 3 or 5.

There should be an `nfs` script in the start-up directory for that level: either `/etc/rc.d/rc3.d` or `/etc/rc.d/rc5.d`, respectively. The name of the script should begin with a capital "S" and be named something like `S60nfs`. If there is no such script, create it by typing one of the two following commands as root user:

```
ln -s /etc/init.d/nfs /etc/rc.d/rc3.d/S60nfs
ln -s /etc/init.d/nfs /etc/rc.d/rc5.d/S60nfs
```

Once the links are created, the NFS server should start each time you reboot Linux.

Setting Up the NFS client

To access a shared directory from an NFS server, a client Linux system uses the `mount` command. The `mount` command connects the remote directory to a local **mount point**, much the same way the `mount` command mounts local hard disks, partitions, CD-ROM devices, and floppy disk drives.

Before you mount a remote directory, you should create a mount point directory. For example, you could create a directory as follows:

```
mkdir /usr/projectX
```

Once you create the directory, you use the `mount` command, as the `root` user, to mount the remote directory onto your local filesystem. For example, the following is an example of a `mount` command for mounting the `/usr/project` from the host computer called `sales99`:

```
mount sales99:/usr/project /usr/projectX
```

If the `mount` is successful, you can see the contents of the `/usr/project` directory on `sales99` by typing `ls /usr/projectX` locally. In this case, the `mount` is temporary. When the system reboots, the `mount` goes away. To make the `mount` shown above occur every time your system starts up, you could add the following entry to your `/etc/fstab` file:

```
sales99:/usr/project /usr/projectX nfs bg,rw 0 0
```

In this example, the `/usr/project` directory from the `sales99` host computer is set to mount on the `/usr/projectX` directory on your local file system *at boot time*. The remote directory is identified as a **network file system** (`nfs`). The directory is mounted with *read/write* access (`rw`). If, when your system tries to mount the remote directory, it is not immediately available, subsequent attempts to mount the resource are done in the background -- without hanging your system waiting for the mount command to complete. (Use the `bg` option if the resource is not critical for starting your system. If for some reason the remote directory is not available, the `bg` option lets your system continue the boot process while it continues to try to connect to the remote directory.)

To Do:

Work in pairs. Each of you will set up an NFS server and connect to your partner's NFS server as a client.

Note: You may have to disable your firewall to get NFS to work. *This is normal.* NFS is designed to be used on a LAN, not on the Internet!

```
/etc/init.d/iptables stop
```

1. To set up your NFS server:

- (a) Edit the `/etc/exports` file. If it exists and has a non-zero size, make a backup first.

Add the following line:

```
/home 192.168.93.0/255.255.255.0(rw) 127.0.0.1(rw)
```

This will set up the `/home` directory for read/write access to all hosts with the class C IP address `192.168.93.x`. The *loopback address* will allow you to test your network share yourself.

- (b) Start the NFS server.
- (c) Notify the NFS server of your share using `exportfs -ra`
- (d) Verify that the exported directory has been registered:

```
cat /proc/fs/nfs/exports
```

2. To set up your NFS client:

- (a) Make the following mount point:

```
/mnt/shared
```

This is where we will be accessing the network share.

- (b) Mount the network share:

```
mount server-address:/home /mnt/shared
```

- (c) Change to the shared directory and test your access rights. Ask your partner to give you write permission to a specific directory.

- (d) Disconnect from the network share. First, make sure that you are not in the mounted directory. Then, use a `umount` command, either:

```
umount server-address:/home (using the share name)  
or umount /mnt/shared (using the mount point)
```

3. To use NFS at boot up:

- (a) One person should add your partner's share to your `/etc/fstab` file. Make sure to make a backup first:

```
cd /etc  
cp fstab fstab.old
```

Meanwhile, the other person should set up the NFS server to run when the system boots. Use `S20nfs` as the link name.

Reboot both machines -- the server first and then the client -- and verify that the network share was mounted correctly. Restore your old `fstab` file.

- (b) Repeat part (a), but switch who sets up the client and who sets up the server.

4. Create another mount point and mount `/home/cnet9565` from Mike Boldin's server, `mothra`, IP address `192.168.93.253`.

Change to your mount point and make a directory using your full name -- for example,

```
mkdir james_bond
```

In this directory, create a file called `lab7_summary.txt` that contains the following information:

- ▶ your server's hostname
 - ▶ your IP address
 - ▶ the mount point that you used to mount `mothra` : `/home/cnet9565`
 - ▶ your lab partner's name
5. Unmount `mothra:/home/cnet9565` and notify Mike Boldin that you have completed the lab.