NFS

Probably the most common network sharing service on internal networks is NFS. It is fairly simple and allows for granular control of who can access shares and with what permissions the share is accessed.

Service Name:
    nfs

Package:
    nfs (Acronym for Network File System, as opposed to your local file system)

Config:
    /etc/exports

Access Control:
    iptables
    SELinux
    Built in access control

Log Files:
    /var/log/messages

NFS is not as simple as FTP, by default. There are no default shares provided when you install the software. Something that IS simple is that you do not install the software because it is already installed as part of the "base" group.

To create an nfs share, create a directory and reference it in /etc/exports and specify who gets access to it.

```
# mkdir /share
# vim /etc/exports
```

With the following content:

```
/share          192.168.0.0/255.255.255.0(ro,sync)
```

Of course, to make a share writeable, switch the "ro" with "rw."

A note on share permissions:

If your users will be mounting this NFS share as root you must include no_root_squash as an option. no_root_squash is a setting that allows nfs clients to connect as root. Without this setting, the root user on clients that connect has the permissions of the user 'nfsnobody', uid 65534 and will not be able to write to the share, even if it has the rw option.

Any changes to this file should be followed by this command to re-apply the /etc/exports file:

```
# exportfs -r
```

Start the service persistently (meaning it will return when the machine starts).

```
# service nfs start
# chkconfig nfs on
```

Here is a quick command to see what is being shared from your system:

```
# showmount -e
```

Advanced configuration of NFS

A common mistake is NFS version confusion.

By default NFS supports versions 2, 3 and 4.

If you wish to disable support for NFS versions 2 and 3 edit /etc/sysconfig/nfs as follows:

```
RPCNFSDARGS="-N 2 -N 3"
```

After restarting NFS using the command service nfs restart, verify the versions which your NFS server supports using the command cat /proc/fs/nfsd/versions

Naturally, disabling NFSv4 and only allowing NFSv2 and NFSv3 connections requires adjusting the RPCNFSDARGS accordingly.

Create firewall rules to allow connectivity over the network to your NFS server:

NFSv2 and NFSv3 only

Define static ports for the following in /etc/sysconfig/nfs:

```
RQUOTAD_PORT
LOCKD_TCPPORT
LOCKD_UDPPORT
MOUNTD_PORT
STATD_PORT
```

It is recommended to use high port numbers which are not currently in use and no make them sequential so that it is easier to troubleshoot.

For example:

```
RQUOTAD_PORT=30000
LOCKD_TCPPORT=30001
LOCKD_UDPPORT=30001
MOUNTD_PORT=30002
STATD_PORT=30003
```

Now add the necessary firewall rules using the commands:

```
# iptables -I INPUT -p tcp -m multiport --dports 30000,30001,30002,30003,2049,111 -j
ACCEPT
# iptables -I INPUT -p udp --dport 111 -j ACCEPT
# service iptables save
```

NFSv4 only

NFSv4 is much easier to secure with a firewall because it only uses 2049/tcp and no longer needs the helper services used with previous versions.

```
# iptables -I INPUT -p tcp --dport 2049 -j ACCEPT
# service iptables save
```

Define your NFS shares in /etc/exports

Syntax:

/directory/to/be/shared IP_ADDRESS/IP_NETWORK(share options) DIFFERENT_IP_ADDRESS/IP_NETWORK(share options)

NFSv3 Example #1:

```
/data/public 192.168.122.0/24(sync,ro,root_squash)
```

/data/public is shared for all hosts on the 192.168.122.0/24 network with the options:

sync specifies that all writes are immediately committed to the NFS server by NFS clients
ro defines the share as read-only
root_squash the user root on the NFS client is root on the local filesystem but not the NFS share

NFSv3 Example #2:

```
/data/HR 192.168.122.0/24(sync,ro,root_squash) 192.168.123.0/
24(sync,rw,no_root_squash)
```

!!! NOTE !!!

The above is all on one line

Similar to above. the /data/HR share is made available to clients across 2 IP networks, with the 192.168.123.0/24 network having read-write access and their root ability extends into the share.

To view NFS shares using NFSv2 or NFSv3, use the command:

```
$ showmount -e ip_of_nfs_server
```

NFSv4 Example

With NFSv4, clients mount 1 share, that is the / of the NFS server. This is not the root filesystem of the NFS server but a pseudo-root filesystem which you configure. All other shares are mount bound to the pseudo root so they appear relative to it.

Given that we want to share the following directories on the NFS server using NFSv4:

```
/home/public
```

```
/data/ISO
/usr/share/doc
```

Create the directory to act as the pseudo-root:

```
# mkdir /nfs
```

Create the mount points for the directories relative to the pseudo-root /nfs

```
# mkdir -p /nfs/{public,ISO,doc}
```

Now persistently bind mount those directories to their respective mountpoints in /etc/fstab:

```
/home/public        /nfs/public    none   bind    0 0
/data/ISO           /nfs/ISO       none   bind    0 0
/usr/share/doc      /nfs/doc       none   bind    0 0
```

Reprocess /etc/fstab by running mount -a

Edit /etc/exports to define your NFS shares:

```
/nfs *(fsid=0,rw,root_squash,sync)
/nfs/public *(rw,root_squash,sync)
/nfs/ISO *(rw,root_squash,sync)
/nfs/doc *(rw,root_squash,sync)
```

/nfs is defined as the pseudo-root of the NFS server as it is configured with the option fsid=0

The shares are configured in the same way (each share may be configured differently).

Each share is read-write for any IP client with root permissions being extended to the NFS share. Naturally this isn't very safe!

In order for the client to access all the shares (should they have permissions to do so), they would execute the following command:

```
# mount  -t  nfs4  IP_ADDRESS_OF_NFS_SERVER:/destination  /mountpoint
```

For example:

```
# mount 192.168.122.2:/ /data
```

The above command mounts all the NFS servers on the client’s /data directory.

To mount this persistently add the entries to /etc/fstab accordingly. For example

```
192.168.122.2:/         /data   nfs4 defaults  0  0
```

Then reprocess /etc/fstab by using mount -a

Lab activity

Create a directory called /foo/nfs_data and copy /etc/*-release to it. Then define a NFSv4 share called called nfs_data with the /data directory set as the pseudo root. Allow all IP addresses to connect to it with read only access.

Make sure that your server is protected by a firewalling allowing NFS access