# Overview

## Goals

NFS is a powerful and flexible way to share files between a server and client computers. NFS is a perfect file sharing method to mount user's home directories on centralized storage. In network infrastructure, this makes it much easier to backup and protect user files and programs.

In this lab, you will install and configure nfs server on your Debian system. This will allow your CentOS client computer to mount a network share that is located on the Debain server. Previous to NFSv4, NFS was only capable of using host authentication (meaning that it allowed specified host computers access to the share). NFSv4 can use idmap and kerberos to configure NFS with user authentication and this significantly improves the security model for NFS. However, in our lab we will be configuring NFS to share to host computers only.

## Deliverables

* Lab Report Including the following
  + All the completed information sections throughout the lab
  + Description of the lab steps completed and the results
  + Any errors you encountered and the steps you took to resolve them
  + This includes noting no errors were found
  + Answers to the questions posed at the end of this lab
* Contents of the following files:
  + /etc/default/nfs-common
  + /etc/exports

## Information

|  |  |  |  |
| --- | --- | --- | --- |
| Description | Formula | Used in Lab | Answer |
| Assigned Subnet Number | *n* | - |  |
| Your Last Name | *<last\_name>* | - |  |
| Debian VM IP | 172.16.*n*.10 | **DEBIAN\_IP** |  |
| CentOS VM IP | 172.16.*n*.50 | **CENTOS\_IP** |  |
| PI IP (optional) | 172.16.n.20 | **PI\_IP** |  |
| DNS Forwarder | - | 172.16.1.254 | 172.16.1.254 |
| Subzone | *<last\_name>*.hades.lab | **SUBZONE** |  |

# Stage 1: NFS Server Installation & Configuration

## Getting Started

1. Load Debian VM
2. Verify the VM’s IP Address matches the IP address you calculated in the Information section above
   1. Refer to Lab 1 for information on viewing and modifying your IP address

## Initial Installation and Configuration (Debian)

1. Install the “nfs-kernel-server” package
   1. aptitude -R install nfs-kernel-server nfs-common portmap
2. Create a bind mount and make it persistent through reboots
   1. NFSv4 relies on all shared directories to be under a single location. In Linux, you don't really have to have them under that directory, you can simply create a mount link or bind between the directories. This is similar to a permanent shortcut.
   2. As the root user on your Debian server, type

# mkdir -p /export/home

# mkdir -p /home/shared

# mount --bind /home/shared /export/home

* 1. Edit the /etc/fstab
     1. Add the following line to make the bind mount automatic when the computer reboots

/home/shared /export/home none ro,bind 0 0

1. Edit /etc/default/nfs-kernel-server
   1. Debian NFS uses 3 files
      1. /etc/default/nfs-kernel-server – Holds main server configuration information
      2. /etc/default/nfs-common – Contains configuration information for other services used by NFS
      3. /etc/exports – Contains the actual shared file configuration for each directory you want to share to your clients
2. Edit /etc/default/nfs-common
   1. Configure NFS to use idmapd to map a connected computer host to a local user account. If we were going to configure NFS to use Kerberos and LDAP, this would enables user-based authentication and is considered more secure than traditional NFS host-based authentication

# Do you want to start the idmapd daemon? It is only needed for NFSv4.

NEED\_IDMAPD=yes

1. Edit /etc/exports
   1. Configure the NFS exports to use Kerberos authenticated logon

# 'fsid=0' designates this path as the nfs4 root  
# 'crossmnt' is necessary to properly expose the paths  
# 'no\_subtree\_check' is specified to get rid of warning messages  
# about the default value changing. This is the default value

/export **CENTOS\_IP**(rw,sync,fsid=0,no\_subtree\_check,crossmnt)

/export/home **CENTOS\_IP**(rw,sync,no\_subtree\_check)

1. Edit /etc/idmapd.conf
   1. IDMAPD is the program that maps the server userid with the clients id. We are going to use our handy LDAP to match the user ids to make thing work smoothly.
   2. Make sure both lines in the idmapd.conf file are setup for your domain and translation method

[General]

Verbosity = 0  
Domain = **SUBZONE**

[Translation]  
Method = nsswitch

[Mapping]  
Nobody-User = nobody  
Nobody-Group = nobody

1. Restart the NFS server
   1. /etc/init.d/nfs-kernel-server restart
2. Check your logs for any error messages

# Stage 2: NFS Client Configuration

## Mount the NFS share on the client (CentOS)

1. In order to now make use of our newly created network home drive, we need to manually mount the NFS share from our CentOS client
   1. Configure your shared directory under your home directory  
      As the root user on you CentOS vm, type:

# mkdir -p /home/shared

* 1. As root, type the following command in a terminal on your CentOS client to mount the new NFS share:

# mount -t nfs4 -o proto=tcp,port=2049 debian.**SUBZONE:**/home /home/shared

1. Now that we can successfully mount the home directory on our client, we need to make the mount persistent through reboots. As we did with the bind mount on our Debian, let's modify the fstab
   1. In the fstab, we will set the auto option to have it connect on restart  
      The \_netdev option will allow it to not fail if the resource is not available. This is handy when starting up to not have the system hang.
   2. Edit /etc/fstab

# mount our NFSv4 home directory on boot

debian.**SUBZONE**:/home /home/shared nfs4 defaults 0 0

# Bonus: Pi NFS Configurtion (+2 pts)

## Mount the NFS share to the Pi client

Using the instructions for the CentOS client as a guideline:

1. Manually mount the Pi to the NFS share,
2. Make the mount persistent.

To get the full 2 points bonus clearly document the steps involved in doing the manual mount and making the mount persistent. Show the mount works. Use a documenting format consistent with the way the labs are written.

# Questions

1. In this lab, we configured NFS without using Kerberos for computer authentication. Why is it more secure to use Kerberos with NFS for client authentication?
2. With NFS, why is it necessary to use the idmapd service?
3. NFSv4 uses the pseudo file system under the /export folder to share files from a server. This creates an additional step to make the bind mounts to setup the location. Explain why you think this is or is not a good idea?