ComS 252 Homework 10: NFS and NIS

Group assignment (check syllabus for group penalty details)

Due November 28, 2023

1 Objectives

In this assignment, you will configure client and server machines to share files and users using NFS and NIS.

2 Build the virtual machines

1. Download the ISO file to initialize the virtual machines for homework.
2. In VirtualBox, create **two** new virtual machines for this assignment. The default disk size (a few GB) should be sufficient. You will need to run both machines at the same time, so adjust the memory settings accordingly.
3. For each VM, set the ISO file as the optical disk, and boot up the VM.
4. Select "Build Hw10c virtual machine" or "Build Hw10s virtual machine"
5. You are encouraged to take a snapshot of each VM after installation completes.
6. At first boot, each VM initializes itself by fetching and running a script from the server. The script will, among other things, create a user account with your ISU username. Users `alice`, `bob`, and `chuck` will have accounts and files set up on machine `Hw10s`. All user accounts will initially have passwords that are the account name, followed by "`pw`".
7. When each VM shuts down after initialization, you are again encouraged to take a snapshot.

3 Set up network interfaces

3.1 Using VirtualBox (HIGHLY recommended)

You will set up a network between the two VMs, using the topology shown in Figure 1.

1. Under the settings for each VM, under "Network", you should already have Adapter 1 enabled and attached to a "NAT" network. Now, enable Adapter 2, attach it to an "Internal Network". The name of the internal network should be the same for both VMs.
2. Boot up each VM, login, and use `nmcli` to give Adapter 2 a private static IP address on the same subnet, as shown in Table 1. If this subnet happens to be one that your host machine uses, you will need to choose a different subnet for this assignment, and update the static IP addresses on both machines. This will be referred to as the "internal network subnet" later.
3. Restart both VMs, and make sure they are running at the same time. In each VM, make sure you can `ping` an Internet address (try `google.com` or `8.8.8.8`) and each static address in Table 1. Once the VMs can send packets to each other and to the Internet, you are ready to configure the server and client.

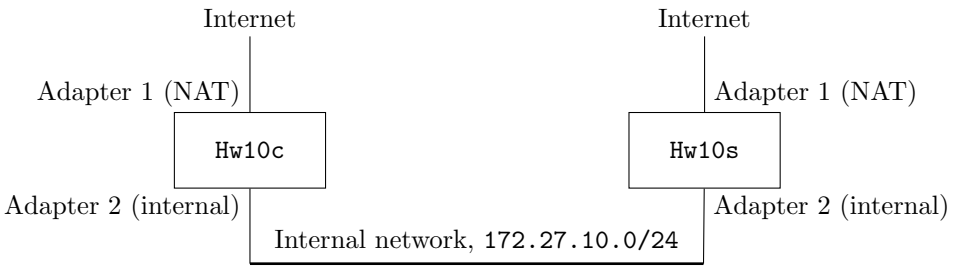

Figure 1: Network topology

Hostname	IP address
Hw10c	172.27.10.50
Hw10s	172.27.10.10

Table 1: IP addresses to use for Adapter 2 on each VM

3.2 Using UTM (all hope abandon, ye who enter here)

1. Use a single network adapter on each VM. It will likely need to be *shared*, so the VMs can connect to the Internet and to each other. This means your *host* machine will see all packets exchanged by the VMs, and thus your host machine could cause interference.

2. Use `nmcli` to determine the IP address assigned to each VM, and the subnet, which will be referred to as the "internal network subnet" later. These are *dynamic* and may change, which could cause problems later.

3. While both machines are running at the same time, make sure you can `ping` an Internet address (try `google.com` or `8.8.8.8`) and the IP addresses of each VM. Once the VMs can send packets to each other and to the Internet, you are ready to configure the server and client.

3.3 Set up `/etc/hosts`

On each VM, add entries to `/etc/hosts` for both the client and server machines. Use the VM's username as the domain name, and use `Hw10s` and `Hw10c` as the machine names. To test, make sure you can transfer packets using

```
ping Hw10c.username
ping Hw10s.username
```

on both VMs, where `username` is replaced with the appropriate username.

4 Sharing files (1)

4.1 Server

1. Configure the server VM so that the `nfs-server` service starts at boot time, and exports the directory `/export` to any IP addresses on the internal network subnet.

2. Configure the server VM so that the `firewalld` service *does not* start at boot time. You will set up the firewall, allowing the appropriate connections through, as the last step.

3. Reboot the server machine.

4.2 Client

1. Test by mounting the server's `/export` directory "by hand", using

   ```
   mount Hw10s.username:/export /mnt
   ```

   where `username` is replaced with the appropriate username. Check that the files in `/mnt` match those in `/export` on the server. You may notice that the file owners do not match; we will fix that in part 5. Unmount once that works.

2. Configure the automounter service, `autofs`, to start at boot time.

3. Create directory `/shares` (as root), and set this as a family of mount points for various network shares, managed by the automounter.

4. Configure the automounter to mount `Hw10s.username:/export` to directory `/shares/server`.

5. Reboot the client and test.

5 Network identity

5.1 Server

Configure the server VM as an NIS "master" server for domain "`cs252`". There are no other NIS servers. The server should start automatically every time the server VM boots. You will need to do the following.

1. Run service `nis-domainname`.

2. Run service `ypserv`.

3. Edit `/etc/sysconfig/network` (see lecture notes).

4. Edit `/var/yp/securenets` (see lecture notes).

5. Run once, as root: `/usr/lib64/yp/ypinit -m`.

5.2 Client

Configure the client VM as an NIS client for domain "`cs252`", whenever the machine boots. You will need to do the following.

1. Run service `ypbind`.

2. Edit `/etc/yp.conf` (see lecture notes).

3. Use utility `authselect` to enable NIS authentication.

The utility `ypcat` is useful for debugging NIS. When this is working, files in `/shares/server` should have proper owners and groups, and users `alice`, `bob`, and `chuck` should be able to login on the client VM. However, when these users login, they will not have any home directory; we will fix that next.

After you have connected the client VM to NIS, you will **not** want to have the client VM running unless the server VM is running; otherwise the client will spend lots of time trying and failing to connect to the NIS server.

6 Sharing files (2)

6.1 Server

Configure the server VM to export the directory `/home`, using NFS, to any clients on the internal network subnet. This change should persist across reboots.

6.2 Client

Configure the client VM to automount the users' home directories from the server, as follows.

1. Set `/home` as a family of mount points to be managed by the automounter.
2. Configure the automounter to mount `Hw10s.username:/home/user` to directory `/home/user`, for every user. You **must** use wildcards in the map file for `/home`; you are not allowed to create a separate entry for each user.
3. Tell SELinux that home directories are now over NFS, by running (as root):

```
setsebool -P use_nfs_home_dirs 1
```

Reboot and test. Once this works, users `alice`, `bob`, and `chuck` should be able to login on the client VM and access their home directories, without getting a permission error at login.

7 Secure the server VM

On the server, turn the firewall back on. Configure the firewall to permanently allow the NFS and rpc-bind services through the firewall. You will also need to permanently allow UDP connections for NIS; you will need to allow this port number through "by hand" (read the `man` page for `firewall-cmd` to see how to allow port numbers through). Some hints:

- Look at `/etc/services` to find an unused port number below 1000, to use for `ypserv`. You may use a port number reserved for another service, as long as that service is not needed on the VM.
- Read the man page for `ypserv` to see how to set its port number. (By default, it gets a random port number.)
- You can add a line of the form

  ```
  YPSERV_ARGS="arg1 arg2"
  ```

  to `/etc/sysconfig/network` to send arguments `arg1 arg2` to `ypserv` when it is started.
- `rpcinfo -p` is your friend.

Be sure these changes to the server persist across reboots. Reboot the *client* and make sure everything still works.

8 Submitting your work

On each VM, from your user account, run "`sudo Turnin`" to submit your work. Make sure both virtual machines are running when you submit, as the submission script will check that the VMs can exchange packets.

As usual, submission requires Internet access (and VPN access, from off campus), as this will collect and upload your work to the homework server. Feedback on your submission is collected in a text file, that you can view later using "`cat submit.log`" or "`less submit.log`".

To shutdown the VMs cleanly, run "`poweroff`".