

## Lab 5: NIS

In this lab, you will go through the steps required to setup your system both as an NIS client and as an NIS server. As an NIS client, you will be able to use the administrative data files (such as `/etc/passwd` or `/etc/hosts`) which are being served by an NIS server. As an NIS server, you will be able share your system's administrative files with others. If you perform this lab outside of class, you will have to perform the NIS server setup first, since there will be no NIS server available to you on the network. However, you will need to be familiar with the NIS client setup, so review the NIS Client Configuration first. Perform the indicated steps, and write the answers to questions in the appropriate spaces on this hand-out. Turn-in one completed handout per team by the beginning of class next week. Be sure to include the names of everyone on your team.

### Configure the System

Bring the system up single-user and perform the initial network configuration.

- Step 1. Boot your system into single user mode. To accomplish this, reboot the system, and interrupt the Red Hat splash screen at the beginning of the boot process with Control-X. Then type **linux single** at the **boot:** prompt.
- Step 2. Configure your system with the IP address 10.0.0.N/24, where N is your system number. Don't forget about the files `/etc/hosts` and `/etc/sysconfig/network`. Refer to Lab 1 if necessary – by now, you should be able to reliably configure your network in just a few minutes, and understand the relationship between the various configuration files and commands.
- Step 3. Bring up the **lo** and **eth0** interfaces.

### NIS Client Configuration

Manually configure the system to be an NIS client, using the services provided by an existing NIS server.

- Step 4. The first step in setting up your system to use NIS services is to set your NIS domain name. A system can only be in a single NIS domain at a time, and once set, it the NIS domain is generally not changed. There is no way to be in one NIS domain to use one set of files and another to use another set of files. I have already configured an NIS server system for you using the NIS domain name **cis68c2**. This server is supplying an `/etc/hosts` and an `/etc/passwd` database, containing additional information for your system to use. Set your NIS domain name with the command below (remember, the NIS domain has no relation to a DNS domain – they are entirely separate):

```
# domainname cis68c2
```

- Step 5. NIS uses RPC services; therefore the **portmap** daemon must be running. If it is not running, start it using the Red Hat run script:

```
# /etc/init.d/portmap start
```

- Step 6. The **ybind** daemon is responsible for sending NIS queries to the NIS server to which it is bound. To establish a binding, it will broadcast a request to find an NIS server, and once bound, it will be able to submit queries for data served by the NIS server. Start the **ybind** daemon to ready your system to use NIS:

```
# ybind
```

### Verify ybind Functionality

Verify that the NIS client daemon `ybind` is functioning correctly

- Step 7. Verify using **ps** that **ybind** is running. If not, or if there is an error in any of the verification steps in this section, the system was not configured correctly and one or more steps above were probably skipped. This must be resolved before proceeding.

**Q1.** How many **ybind** processes are running? \_\_\_\_\_

**Q2.** What other daemon do you know about that also runs as multiple processes or threads? \_\_\_\_\_

Step 8. The **rpcinfo** command is a diagnostic tool that returns information about RPC services – it can be used to determine if an RPC server is responsive. Use the command below to test if the **ypbind** RCP service is functioning. If it is, **rpcinfo** will report back the RPC service port number for that service.

```
$ rpcinfo -u localhost ypbind
```

**Q3.** What did **rpcinfo** report? \_\_\_\_\_

**Q4.** Look up the **program number** in **/etc/rpc**. What RCP service does that number correspond to? \_\_\_\_\_

Step 9. The **ypwhich** command will output the name of the NIS server to which your system is bound. Run **ypwhich** to determine your NIS server.

**Q5.** Which NIS server is your system bound to? \_\_\_\_\_

### NIS Database Selection

*Configure the **/etc/nsswitch.conf** service switch file to allow the system to obtain NIS served data.*

Step 10. Traditionally, UNIX programs found key administrative data by searching files such as **/etc/passwd**, **/etc/hosts**, **/etc/group**, and so forth. Programmers use core functions in UNIX library code such as **gethostbyname** and **gethostbyaddr** to obtain information about a particular host or IP address, and **getpwent**, and **getpwuid** to obtain password database information. With the advent of DNS, NIS, NIS+, etc., these core functions were modified to allow retrieval of data from the newer mechanisms as well. The name services switch file **/etc/nsswitch.conf** tells these functions which mechanism to use to find the data. The available mechanisms are **files**, **nis**, **dns**, **nisplus**, etc. Configure the **hosts** and **passwd** entries in the **/etc/nsswitch.conf** to have the system find information by first looking in a local file (**/etc/hosts**, **/etc/passwd**), and if not found there, to use NIS. Look for the lines beginning with **hosts** and **passwd**, and make the entries look like:

```
passwd:      files nis
hosts:      files nis
```

Step 11. Your system should now be configured to find hostnames first from **/etc/hosts**, and if not found there, it will use NIS. All the hosts in the NIS server's **/etc/hosts** file are now available to your system as if they were in your system's own **/etc/hosts** file. If you want to define a new host name, it can be added locally to your system's **/etc/hosts** file, or you can have the administrator add it to the NIS server's **/etc/hosts** file to make it available to all systems in the NIS domain. Verify that programs now use NIS for the **hosts** database - ping a host for which there is no entry in your system's **/etc/hosts** file.

### Examining an NIS-served database

*Use NIS utilities to examine the contents of the NIS served databases.*

Step 12. Once NIS appears to be working, you can examine the NIS databases being served. The **ypcat** program is analogous to the **cat** program - it dumps the database specified on the command line. Use **ypcat** to dump the **hosts** database being served by the NIS server for your domain:

```
$ ypcat hosts
```

**Q6.** Describe the output: \_\_\_\_\_

**Q7.** Where is the file from which this data originated? \_\_\_\_\_

**Q8.** What command line would you use to output only the entry for **host21**? \_\_\_\_\_

Step 13. Dumping an entire database can be expensive since the entire database is sent across the network, and databases such as **hosts** or **passwd** can be very large. It is better to use the **ypmatch** program to lookup a single record from a database (rather than **ypcat** piped to **grep**). Use **ypmatch** to obtain the entry for **host11** from the **hosts** database. The syntax is similar to **grep**'s (REs are not supported however): **ypmatch** *item yppmap*. Try:

```
$ ypmatch host11 hosts
```

**Q9.** Using **ypmatch**, what is the IP address of the host **inet-gw**? \_\_\_\_\_

**Q10.** What is the real user name of the **guest** account? \_\_\_\_\_

### Boot-time Initialization

*Configure the system to use NIS services from a given domain when entering multi-user mode.*

Step 14. To have your system startup NIS when the system boots, in Red Hat Linux, there must be a symbolic link in your system's startup script directory **/etc/rc3.d** (or **/etc/rc5.d** for the graphical startup) to the actual startup script **/etc/init.d/ypbind**. You can use Red Hat's **chkconfig** command, or use the command below to create the symbolic link.

```
# ( cd /etc/rc3.d && ln -s ../init.d/ypbind s74ypbind )
```

Step 15. Finally, to activate **ypbind** at startup, set the **NISDOMAIN** shell variable in Red Hat's networking configuration file **/etc/sysconfig/network**. The Red Hat networking startup script **/etc/init.d/ypbind** uses this value; only if the **NISDOMAIN** variable is set will the script configure the NIS domain name with the value of the variable and start **ypbind** each time the system enters run level 3 (or 5). Add the line below to the **/etc/sysconfig/network** file.

```
NISDOMAIN=cis68c2
```

Step 16. Reboot now into multi-user mode - your system should function as an NIS client the same as when you manually configured NIS in the previous steps.

Step 17. The NIS server for the domain **cis68c2** is serving its **passwd** database and these entries are available to your system. You can login to your own system using any account defined on the NIS server. Try to login to your own system using the **guest** account which is defined in the NIS **passwd** database. The password is the same as the password we are using this quarter for the root and student accounts.

### NIS Server Setup

*Setup your system as an NIS server, so that other clients can use your system's administrative files.*

Step 18. Bring the system to single-user mode. Be sure the **portmap**, **ypserv** and **ypbind** services are not running.

Step 19. Bring up the **lo** and **eth0** interfaces.

Step 20. Configuration of an NIS server must occur from the **/var/yp** directory. Change to the **/var/yp** directory.

Step 21. Every NIS server must have a unique NIS domain name. Decide on a simple domain name (*not* **cis68c2**) and set the domain name as you did previously using the **domainname** command. Be sure the domain name is set, and that other systems are not using it!

**Q11.** How do you verify that the NIS domain name is set? \_\_\_\_\_

Step 22. The file **Makefile** in the **/var/yp** directory contains a set of instructions telling the **make** utility how to build the DBM databases used by NIS from the original flat ASCII files such as **/etc/passwd** or **/etc/hosts**. The **Makefile** should be modified so that it will only build the databases that your system will serve. Search for the **all target** by editing the file with **vi** and typing **/^all** and hit enter. The **all target** lists the DBM databases that will be built when you run **make** later (do not run it now!). Comment out the original **all** line using the **#** character and create a new **all** line *above* the commented out line (do this to avoid a bug in **make** where a trailing backslash on a commented line is not ignored). The lines should look like the lines below, in the same order:

```
all: hosts passwd
#all: passwd group hosts rpc services netid protocols \
```

Step 23. Start **portmap** as you did previously.

Step 24. Start the NIS server daemon **ypserv** (you can start it manually, or use the **/etc/init.d** script). Starting this now will prevent certain harmless error messages when running the **ypinit** command below. If you do not have the **ypserv** daemon installed on your system, install the **ypserv** package from the Red Hat 7.2 installer disc (disc one).

Step 25. Start the NIS client daemon **ypbind** – your system can be a client of its own services.

Step 26. The NIS database now needs to be initialized. This is done with the **ypinit** command, using the **-m** option (**-m** means *master*). You run this command only *once*, which primes the system by creating the initial DBM database(s). The **ypinit** command will ask you for a list of hosts to add as (slave) NIS servers, and will automatically add your host as the first choice. Do not add any additional servers. Run the command and respond to **ypinit**'s prompts – read the prompts carefully! You can ignore the error about **gethostbyname** being temporarily unavailable.

```
# /usr/lib/yp/ypinit -m
```

**Q12.** Which files were updated? \_\_\_\_\_

Step 27. You should now have a new sub-directory **/var/yp/domainname** where *domainname* is the name of the NIS domain you set previously. Verify that the files **hosts.byname**, **hosts.byaddr**, **passwd.byname**, **passwd.byuid** are in this new directory. These are the DBM databases used by NIS for better performance, rather than the flat ASCII files (**/etc/hosts**, etc.). This means that any changes to the server's ASCII files need to be forced into the DBM files.

Step 28. Verify that the file **/var/yp/ypservers** contains only your hostname (note: there is also a binary DBM version of this file in new sub-directory mentioned above). These were created by **ypinit**.

Step 29. Verify that **/etc/nsswitch.conf** is configured to use NIS for the **passwd** and **hosts** databases.

### Verify NIS Functionality

*Verify the NIS domain and that the appropriate daemons are working correctly.*

Step 30. Verify your system's NIS domain name binding with **ypwhich**.

Step 31. Use **ypcat** or **ypmatch** to dump the **hosts** or **passwd** database to verify that the databases are being served.

Step 32. Once your system can use its own NIS databases, other properly configured NIS clients are also eligible to use them. You can ask someone on the network to setup their system to use your server by setting their NIS domain name to your server's NIS domain, and restarting **ypbind** with the command **/etc/init.d/ypbind restart**.

### Making New Entries Available through NIS

*Add new entries in /etc/hosts, etc. and rebuild the DBM databases to make the data available to NIS clients.*

Step 33. The process for adding new entries into the NIS-served files is simple. New entries are added to the flat ASCII files, and then the DBM databases are rebuilt. Add a new entry to your **/etc/hosts** file.

Step 34. From **/var/yp**, run **make**. This will update the DBM database files.

```
# make
```

Step 35. Now the new entry should be available to all NIS clients bound to your NIS server. Verify that an NIS client bound to your NIS domain sees the new entry.

### Advanced Configuration (optional)

*Explore the additional configuration options for NIS.*

Step 36. The file **/var/yp/securenets** adds a little more security for NIS. Configure **securenets** so that only hosts on your network can access NIS. Do this by replacing the catchall access line of 0.0.0.0 with a line that gives access to only your network (see the line below). Note: you will not be able to verify this setting unless you configure some other system to be on another network and setup a route so that you can verify access for the 10.0.0.0 net but no access for other networks.

```
255.255.255.0 10.0.0.0
```

Step 37. The configuration file **/etc/ypserv.conf** is used for advanced settings of the NIS server daemon **ypserv** – review the **ypserv.conf** man page to learn how about advanced configuration of an NIS server.

- Step 38. The configuration file **/etc/yp.conf** is used for advanced settings of an NIS client daemon **ypbind** – review the **yp.conf** man page and learn about how NIS clients can be configured.
- Step 39. Extra Credit: Learn and document the steps to configure an additional NIS slave server.
- Step 40. Extra Credit: Configure your system to be an NIS server for passwd and automount information, and an NFS server so that any user can login to any system in your NIS domain with their own account such that their home directory (served by your NFS server) is automounted to the user's system.