Chapter 25

Network Information and Directory Services

If a large number of clients are mounting directories from a number of NFS servers, the same users may exist on multiple clients, but their user and group IDs may not be consistent. This inevitably leads to horrible problems with file permissions. Any setup with multiple clients in which the same users can exist on more than one client faces this kind of problem, unless the `/etc/passwd` and `/etc/group` files on all the clients are identical, or at least contain identical information for each particular user.

Additionally, in any such environment, maintaining a local set of users and local authentication on each machine is a huge administrative problem.

So, just as DHCP (see Chapter 20) solves the question of how to maintain local network configurations on multiple machines by centralizing the configuration of IP addresses, a solution is needed to the question of how to centralize user and group IDs and user authentication.

This chapter looks at two such solutions:

- The Network Information Service (NIS) that was pioneered by Sun Microsystems
- A more flexible and extensible system: openLDAP, an open source implementation of the Lightweight Directory Access Protocol

IN THIS CHAPTER

Using NIS for Authentication

NIS was originally developed by Sun and called "Yellow Pages," but the name was dropped after a legal conflict with British Telecom over their trademark on those words. However, most of the commands and filenames associated with NIS have the letters `yp` in them.

A NIS server provides a set of users, groups (and optionally other facilities) across the network. Clients that are set up with NIS client software talk to the NIS server to retrieve user and group information. When users log in, authentication is done against the information held on the server. This can guarantee that the same set of users and groups exist on each of the clients and, furthermore, that the user and group IDs will be synchronized across the network.

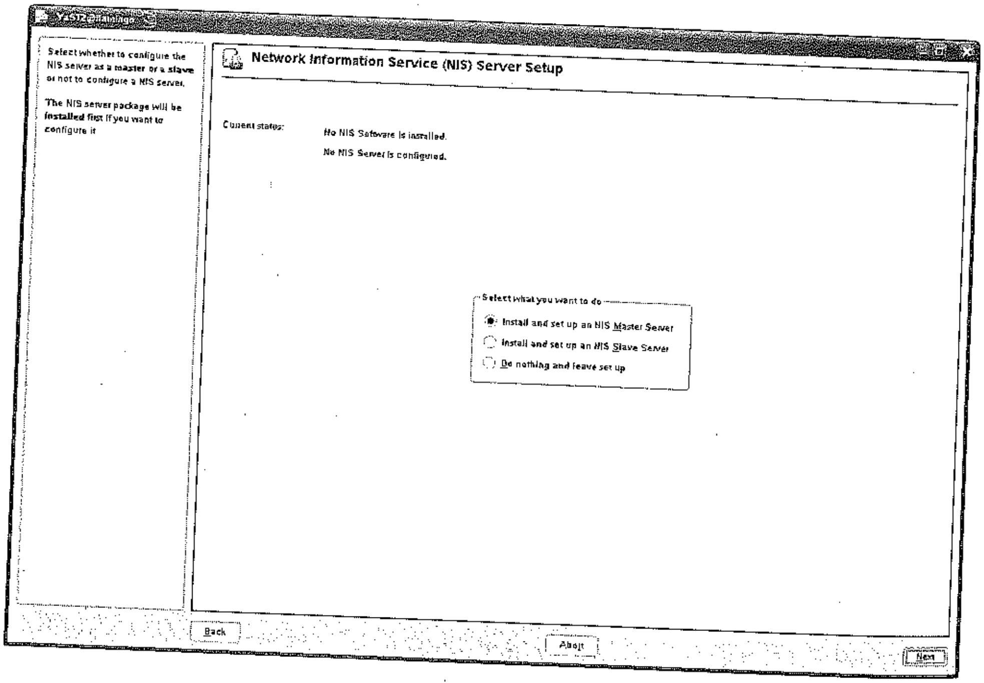
Setting Up a NIS Server Using YaST

NIS is widely used as a distributed authentication mechanism as it is freely available on almost all Unix and Unix-like systems. It is also easy to set up, and far less complex than LDAP.

You will not be surprised to find that YaST has a module for setting up a NIS server. To get a basic NIS configuration using this module is very straightforward.

FIGURE 25-1

The YaST module for NIS Server configuration



Select whether to configure the NIS server as a master or a slave or not to configure a NIS server.

The NIS server package will be **installed** first if you want to configure it

Network Information Service (NIS) Server Setup

Current status: No NIS Sofware is installed.
No NIS Server is configured.

Select what you want to do

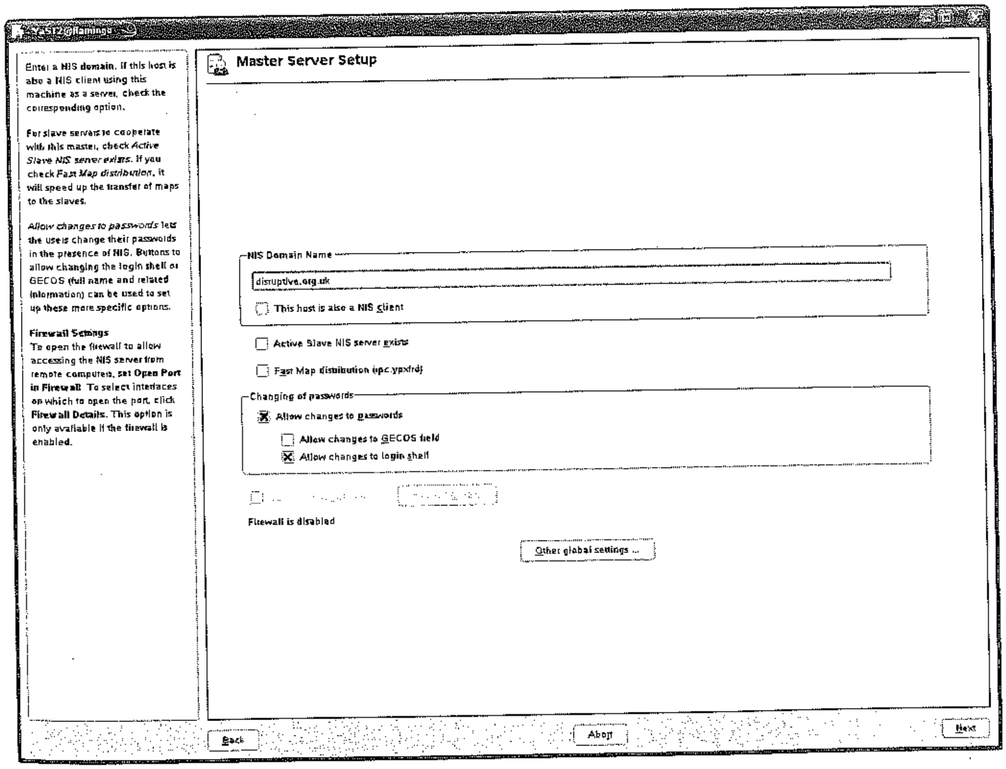
● Install and set up an NIS Master Server
○ Install and set up an NIS Slave Server
○ Do nothing and leave set up

You call the YaST NIS Server module from the YaST menus (Network Services) or from the command line by typing **yast2 nis_server**. Then follow these steps:

1. You will see a dialog like Figure 25-1. Choose Install and set up a NIS master server. If the NIS packages (`ypserv`, `yptools`, `ypbind`) are not already installed, YaST's package manager installs them for you.
2. Type the NIS domain name. (This is not necessarily the same as the DNS domain name, but very often, for reasons of simplicity, it may be.) As shown in Figure 25-2, you may choose to allow users to change their passwords and login shell.
3. The next screen (see Figure 25-3) defines the information set that the NIS server distributes to clients. If this server is to handle user logins, these so-called NIS maps (which contain and provide the necessary information) must at least include `group` and `passwd` (to export basic user information and authentication).

FIGURE 25-2

Specifying the NIS domain and capabilities in YaST



Enter a **NIS domain**. If this host is also a NIS client using this machine as a server, check the corresponding option.

For slave servers to cooperate with this master, check *Active Slave NIS server exists*. If you check *Fast Map distribution*, it will speed up the transfer of maps to the slaves.

Allow changes to passwords lets the users change their passwords in the presence of NIS. Buttons to allow changing the login shell or GECOS (full name and related information) can be used to set up these more specific options.

Firewall Settings
To open the firewall to allow accessing the NIS server from remote computers, set **Open Port in Firewall**. To select interfaces on which to open the port, click **Firewall Details**. This option is only available if the firewall is enabled.

Master Server Setup

NIS Domain Name: disruptive.org.uk

□ This host is also a NIS client

□ Active Slave NIS server exists

□ Fast Map distribution (rpc.ypxfrd)

Changing of passwords

☒ Allow changes to passwords

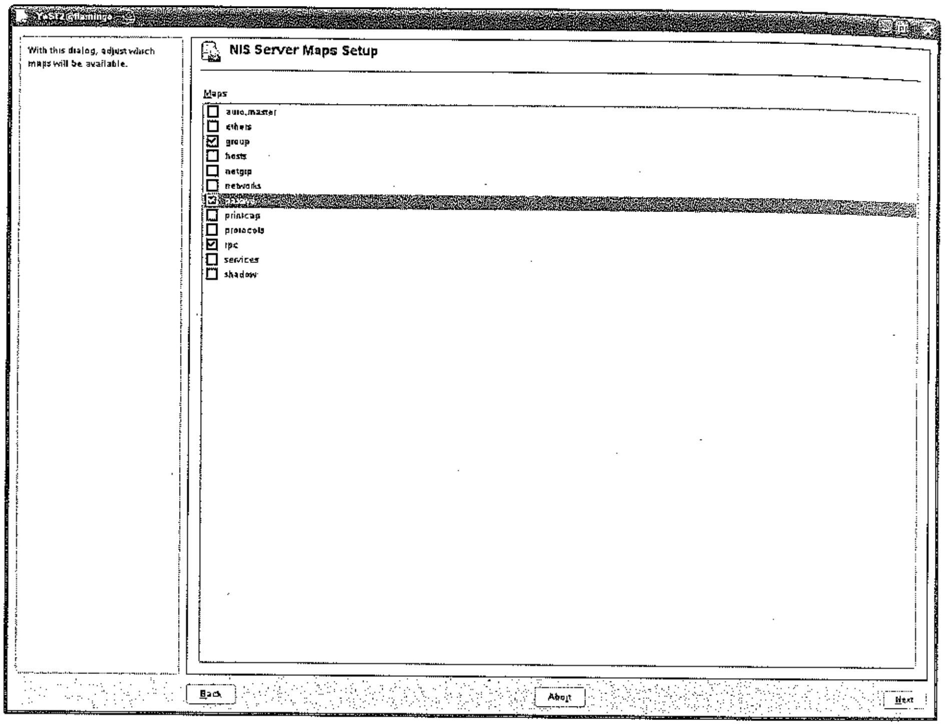
□ Allow changes to GECOS field

☒ Allow changes to login shell

Firewall is disabled

FIGURE 25-3

Selecting NIS maps to export



With this dialog, adjust which maps will be available.

NIS Server Maps Setup

Maps

- □ auto.master
- □ ethers
- ☑ group
- □ hosts
- □ netgrp
- □ networks
- ☑ passwd
- □ printcap
- □ protocols
- ☑ rpc
- □ services
- □ shadow

4. As shown in Figure 25-4, you select network masks that define the hosts that will be able to access the NIS server. Hosts can contact the NIS server for authentication information if the bitwise AND of a specified netmask and the host's IP address is equal to one of the values specified for the network in this dialog box. The netmask/network pair 0.0.0.0/0.0.0.0, therefore, enables any host to contact this NIS server. This is safe only if your network is not connected to the Internet or if you have a firewall between your network and the Internet that disables NIS and NFS RPC traffic. Click Finish to continue.

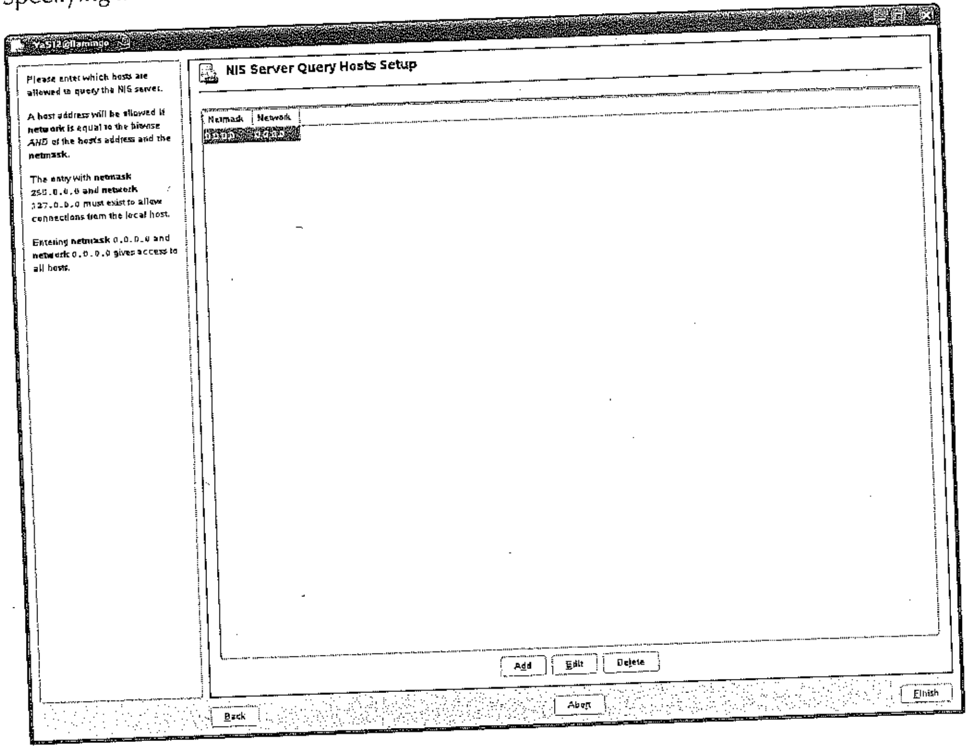
5. YaST now updates your system, starts the NIS server `ypserv`, and sets `ypserv` to run in its default runlevels. You are now ready to configure (and test) the NIS client as described later in this chapter, in the section "Configuring Clients for NIS."

Setting Up a NIS Server Manually

YaST's NIS server module makes it convenient and easy to set up a NIS server. However, it is always instructive to look at the changes that YaST has made in the last section and discover how to interact with the NIS server manually through the command line and configuration files.

FIGURE 25-4

Specifying IP addresses that can use this NIS server



Please enter which hosts are allowed to query the NIS server.

A host address will be allowed if **network** is equal to the bitwise *AND* of the hosts address and the **netmask**.

The entry with **netmask** 255.0.0.0 and **network** 127.0.0.0 must exist to allow connections from the local host.

Entering **netmask** 0.0.0.0 and **network** 0.0.0.0 gives access to all hosts.

The data that NIS actually exports is under /var/yp/. Databases for the NIS maps will be exported under /var/yp/<domain name>/.

There is a Makefile at /var/yp/Makefile. Typing make all in the directory /var/yp rebuilds the NIS maps. This Makefile has been rewritten by YaST on the basis of the NIS maps that we chose to export. If you look at the line in /var/yp/Makefile defining the action taken when you type make all, it contains exactly those maps:

```
all: group passwd rpc
```

The network information that was entered in the final stage of the YaST configuration is stored in the file /var/yp/securenets.

To set up a NIS server entirely from the command line, use the following steps.

1. To set the NIS domain name, use the following:

```
# ypdomainname disruptive.org.uk
```

rpm -i ypserv* suse\i586\ _

2. To go through a set of configuration steps to define, run the program /usr/lib/yp/ypinit (note that this is not in the path by default, so you need to call it with its full path):

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

3. At this point, we have to construct a list of the hosts which will run NIS servers. flamingo.disruptive.org.uk is in the list of NIS server hosts. Please continue to add the names for the other hosts, one per line. When you are done with the list, press Ctrl+D.

   ```
           next host to add:  flamingo.disruptive.org.uk
           next host to add:
   ```

4. You are asked to confirm:

   ```
   The current list of NIS servers looks like this:
   flamingo.disruptive.org.uk
   Is this correct?  [y/n: y]
   We need a few minutes to build the databases ...
   Building /var/yp/disruptive.org.uk/ypservers ...
   Running /var/yp/Makefile ...
   gmake[1]: Entering directory '/var/yp/disruptive.org.uk'
   Updating passwd.byname ...
   failed to send 'clear' to local ypserv: RPC: Program not
   registeredUpdating passwd.byuid ...
   failed to send 'clear' to local ypserv: RPC: Program not
   registeredUpdating group.byname ...
   failed to send 'clear' to local ypserv: RPC: Program not
   registeredUpdating group.bygid ...
   failed to send 'clear' to local ypserv: RPC: Program not
   registeredUpdating rpc.byname ...
   failed to send 'clear' to local ypserv: RPC: Program not
   registeredUpdating rpc.bynumber ...
   failed to send 'clear' to local ypserv: RPC: Program not
   registeredUpdating services.byname ...
   failed to send 'clear' to local ypserv: RPC: Program not
   registeredUpdating services.byservicename ...
   failed to send 'clear' to local ypserv: RPC: Program not
   registeredUpdating netid.byname ...
   failed to send 'clear' to local ypserv: RPC: Program not
   registeredgmake[1]: Leaving directory '/var/yp/disruptive.org.uk'
   flamingo.disruptive.org.uk has been set up as a NIS master server.
   ```

5. Now you can run ypinit -s flamingo.disruptive.org.uk on all slave servers.

   You received the errors RPC: Program not registered because ypserv was not running at that point. Now you can start ypserv:

   ```
   # rcypserv start
   ```

You can also set ypserv to run in its default runlevels:

```
# chkconfig ypserv on
```

The Makefile /var/yp/Makefile will have been created. You can edit the line starting with the target word all: to define which NIS maps you want rebuilt when you type make all.

The NIS server is now configured. In the next section, we look at how to set up a client to talk to the NIS server, whether you configured it manually or through YaST.

Configuring Clients for NIS

This section explains how to set up a NIS client for the server started in the previous section. As with setting up a NIS server, NIS clients can be configured using command-line utilities or by taking advantage of YaST's graphical administrative interface. The following two sections explain how to configure a NIS client using each of these methods.

Configuring a NIS Client Using YaST

YaST makes NIS client configuration almost trivial, using a single dialog box to collect information about the NIS domain that you want your client to use. To configure a system as a NIS client using YaST, do the following:

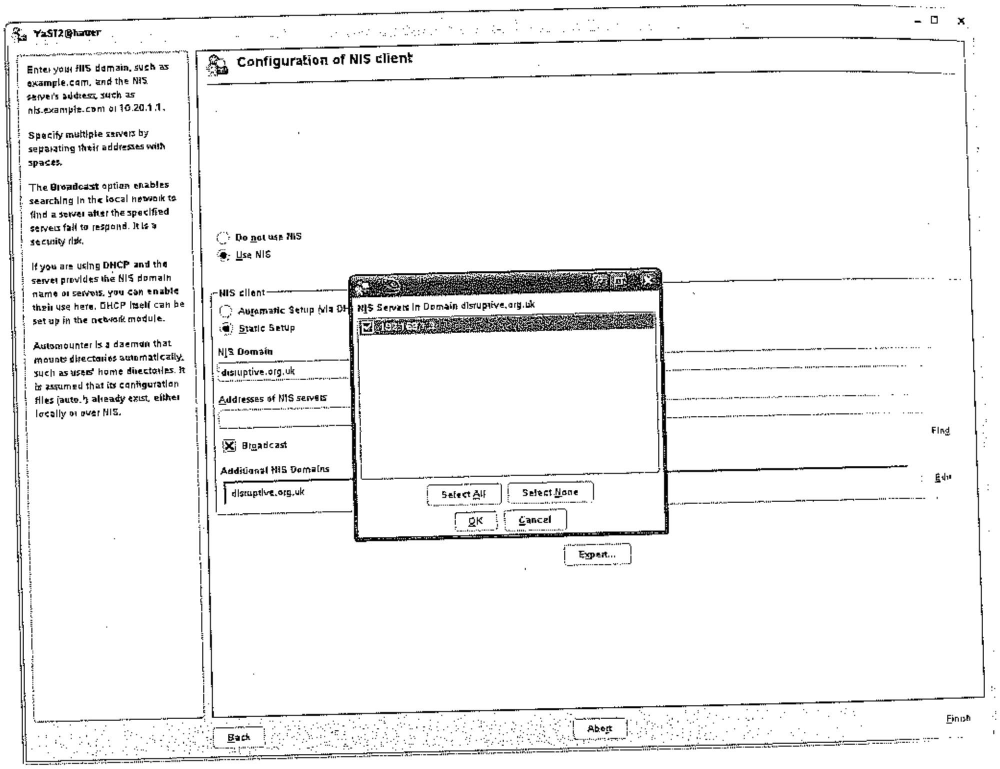
1. Start YaST's NIS client module. It is included in the Network Services Menu, or can be started from the command line with the command yast2 nis. You will see the screen shown in Figure 25-5.
2. Click the Use NIS option button, and enter the name of the NIS domain that you want the client system to contact for authentication information. If your client system uses DHCP to deliver NIS server information, click Automatic Setup (through DHCP). If you want the NIS Client to search the local network by broadcasting if a connection to the specified NFS server fails, click the Broadcast option button. To search for a NIS server on the local network for the specified domain, click Find to populate the Addresses of NIS servers field.
3. Click Finish. YaST modifies your system's configuration files to use NIS and exits after those modifications have been made.

Configuring a NIS Client Manually

As mentioned previously, it is interesting to understand exactly what's going on under the hood when configuring a NIS client. This section explains how to configure a system to be a NIS client without using graphical utilities.

FIGURE 25-5

Configuring a NIS client in YaST



Configuration of NIS client

Enter your NIS domain, such as example.com, and the NIS server's address, such as nis.example.com or 10.20.1.1.

Specify multiple servers by separating their addresses with spaces.

The **Broadcast** option enables searching in the local network to find a server after the specified servers fail to respond. It is a security risk.

If you are using **DHCP** and the server provides the NIS domain name or servers, you can enable their use here. DHCP itself can be set up in the network module.

Automounter is a daemon that mounts directories automatically, such as users' home directories. It is assumed that its configuration files (auto.*) already exist, either locally or over NIS.

○ Do not use NIS
● Use NIS

NIS client
○ Automatic Setup (via DH
● Static Setup

NIS Domain: disruptive.org.uk

Addresses of NIS servers:

☒ Broadcast

Additional NIS Domains: disruptive.org.uk

NIS Servers in Domain disruptive.org.uk

☑ [illegible]

To do some preconfiguration, log in as root or use the `su` command to become root on the client system and edit the `/etc/nsswitch.conf` file on the system you are using as a NIS client. Find the line that tells your system how to locate password entries and modify that line to look like the following:

```
passwd:     nis [NOTFOUND=return] files
```

This tells your system to look for password information in NIS and the lookup will fail if the appropriate information is not found.

Next, save a copy of your system's password file and then remove all entries in the existing password file for "normal users" (those with UIDs of 1000 or more). As the last line of the new, shorter password file, add the following:

```
+::::::
```

This tells NIS to append the contents of the password map (file) retrieved from the NIS server whenever password information is requested.

Note that the entries for any individual accounts (including your own) have been removed from the abbreviated password file. This enables you to do a fairly simple test to determine whether NIS is working. If you can log in using an account that is not present in the password file on your client system, but is present in the password file on your NIS server system, then NIS is working correctly.

To set up a NIS client, log in as root or use the `su` command to become root on the system you are using as a NIS client and do the following:

1. Make sure that the NIS client software package `ypbind` is installed on your Linux system.
2. Set the domain name of the NIS domain to which this new client will belong. This should be the same name as the domain name set earlier in this chapter. To set the NIS domain name (in this case, to the domain `foo.com`), issue a command such as the following:

   ```
   ypdomainname foo.com
   ```

3. Start the NIS client process using a command such as the following:

   ```
   rcypbind start
   ```

To verify that NIS is working correctly, use the `telnet` or `ssh` commands from the NIS client system to contact the client and attempt to log in as yourself. Remember that your password file entry is present in the password file on the NIS server, but not in the password file on the NIS client. If everything is working, set `ypbind` to run in its default runlevels using the command `chkconfig ypbind on`.

You should be able to log in successfully. Congratulations — you're running NIS! You should now modify your system's startup sequence to add the `/etc/init.d/ypserv` startup script.

NIS is a straightforward way of centralizing user and authentication information on a network. It is ideal for small and medium-sized networks and works well together with NFS, and also with automatic mounting of filesystems from NFS servers.

TIP **For additional information about NIS, see the NIS HOWTO at** `www.linux-nis.org/nis-howto/HOWTO/NIS-HOWTO.html`.

Working with LDAP in SUSE

As discussed in the previous section, one way to centrally manage your users and services is to use Network Information System (NIS). NIS was created by Sun to help Unix administrators manage their users without having to create user accounts locally on all machines.