Roaming profiles with NFS and NIS

The following HOWTO describes the configuration of roaming profiles using Slackware Linux. Roaming profiles come in handy in setups like SOHO networks, schools, town halls or more generally any network where many users have to work on many different desktop clients.

- All the users' data and preferences are stored centrally on the server.
- Authentication is also centralized, so a new user is only created once on the server.
- Any user can connect to any desktop client and immediately retrieve his files, desktop preferences, personal wallpaper, bookmarks, etc.

Prerequisites

Centralized authentication requires all machines - server and desktop clients - to be perfectly synchronized. Take a look at the NTP HOWTO to see how this can be done.

Configure the server

Define static ports for NFS and NIS

We want to use NFS and NIS with a firewall, so the first thing we do is assign static ports to their respective services. The step below is not strictly necessary, but in order to do things properly, we will note down all chosen ports in /etc/services, at the end of the file:

```plaintext
... # Static ports for NIS
ypserv 834/tcp  # NIS server
ypserv 834/udp  # NIS server

# Static ports for NFS
statd 32765/tcp  # NFS statd (in)
statd 32765/udp  # NFS statd (in)
statd 32766/tcp  # NFS statd (out)
statd 32766/udp  # NFS statd (out)
mountd 32767/tcp  # NFS mountd
mountd 32767/udp  # NFS mountd
lockd 32768/tcp  # NFS lockd
lockd 32768/udp  # NFS lockd
rquotad 32769/tcp  # NFS mountd
rquotad 32769/udp  # NFS mountd
```

Don't forget to open these ports in the firewall. This is a more common mistake than you would think.

Edit /etc/rc.d/rc.rpc to assign static ports for statd:
Last but not least, create `/etc/sysctl.conf` to assign ports to `lockd`:

```bash
fs.nfs.nlm_tcpport=32768
fs.nfs.nlm_udpport=32768
```

This last option will only be effective after rebooting the server.

**Share users' home directories with NFS**

Edit `/etc/exports` to define your NFS shares. In the example below, `/home` is shared for the whole 192.168.2.0/255.255.255.0 subnet:

```bash
/home 192.168.2.0/24(rw,async,no_subtree_check)
```

Activate the shares:

```bash
# exportfs -a
# chmod 0755 /etc/rc.d/rc.rpc
# chmod 0755 /etc/rc.d/rc.nfsd
```

Reboot the server and check if all services are activated and use the right ports:

```bash
# pmap_dump
  100000  2  tcp  111  portmapper
  100000  2  udp  111  portmapper
  100024  1  udp  32765  status
  100024  1  tcp  32765  status
```
Setup centralized authentication with NIS

Slackware64 has a bug in yptools, that can be corrected like this:

```
# cd /usr/lib
# ln -s /usr/lib64/yp .
```

This bug has been fixed in Slackware64 14.0

Define a NIS domain by creating and/or editing /etc/defaultdomain. Finding an appropriate domain name is a mere convention. In the example below, this is my company’s NIS server (“Montpezat” is a small town in South France):

```
microlinux.montpezat
```

Edit /etc/yp.conf so the server becomes its own NIS client:

```
domain microlinux.montpezat server localhost
```

Uncomment the relevant lines in /etc/rc.d/rc.yp, the NIS startup script, and define the use of static ports for ypserv:

```
if [ -r /etc/defaultdomain ]; then
    nisdomainname `cat /etc/defaultdomain`
fi

if [ -x /usr/sbin/ypserv ]; then
```
echo "Starting NIS server:  /usr/sbin/ypserv"
/usr/sbin/ypserv --port 834
fi

if [ -x /usr/sbin/rpc.yppasswdd ]; then
  echo "Starting NIS master password server:  /usr/sbin/rpc.yppasswdd"
  /usr/sbin/rpc.yppasswdd
fi

if [ -d /var/yp ]; then
  echo "Starting NIS services:  /usr/sbin/ypbind -broadcast"
  /usr/sbin/ypbind -broadcast
fi

On a default Slackware installation, this script is already activated. Note that it needs /etc/rc.d/rc.rpc to be started previously.

# /etc/rc.d/rc.yp start

Proceed to a few adjustments in /var/yp/Makefile:

- the value of MINGID must be altered from 500 to 1.
- values for MERGE_PASSWD and MERGE_GROUP flip from true to false.
- the publickey option has to be deactivated.

Here's how the file is edited accordingly:

...
MINGID=1
...
MERGE_PASSWD=false
MERGE_GROUP=false
...

all: passwd group hosts rpc services netid protocols netgrp mail \
    shadow # publickey networks ethers bootparams printcap \
    # amd.home auto.master auto.home auto.local passwd.adjunct \
    # timezone locale netmasks
...

Initialize the NIS domain:

# /usr/lib/yp/ypinit -m

From now on, adding one or more new users can be done as usual with the adduser or useradd commands, but there's an additional step involved. For new users to be taken into account, issue the following commands:

# cd /var/yp
# make
Configure the client(s)

Mount the shared user home directories

On the client side, we will cleanup /home, which will be our mountpoint for the servers' NFS shares. A vanilla Slackware install contains a /home/ftp directory, which we will get rid of:

```bash
# rmdir /home/ftp
```

Start /etc/rc.d/rc.rpc and try to mount the shares manually. In the example below, nestor is my company's server. Replace this value with either your server's hostname or IP address:

```bash
# chmod 0755 /etc/rc.d/rc.rpc
# /etc/rc.d/rc.rpc start
# mount -t nfs nestor:/home /home
# ls /home
# umount /home
```

Once this step has been successful, define a permanent NFS mount in /etc/fstab. Again, replace nestor by your servers' hostname or IP address:

```bash
... 
nestor:/home /home nfs rsize=8192,wsize=8192,timeo=14,intr 0 0
```

Configure the client to use the NIS server for authentication

As noted above, Slackware64 versions prior to 14.0 contain a bug in yptools, which can be fixed like this:

```bash
# cd /usr/lib
# ln -s /usr/lib64/yp .
```

Define the NIS domain like you did on the server, by creating and/or editing /etc/defaultdomain:

```bash
microlinux.montpezat
```

Define the NIS server by editing /etc/yp.conf:

```bash
domain microlinux.montpezat server nestor
```

Modify /etc/nsswitch.conf so it looks like this:

```bash
passwd:     files nis
shadow:     files nis
group:      files nis
hosts:      files dns nis
networks:   files
```
services:       files 
protocols:     files 
rpc:           files 
ethers:       files 
netmasks:     files 
netgroup:     files 
bootparams:   files 
automount:    files 
aliases:      files 

To startup NIS on the client, uncomment the relevant lines in the `/etc/rc.d/rc.yp` startup script:

```bash
...
if [ -r /etc/defaultdomain ]; then
    nisdomainname `cat /etc/defaultdomain`
fi
...
if [ -d /var/yp ]; then
    echo "Starting NIS services: /usr/sbin/ypbind -broadcast"
    /usr/sbin/ypbind -broadcast
fi
...
```

On a vanilla install, there's no activation (`chmod 0755 /etc/rc.d/rc.yp`) to be done, since the script is already activated.

Reboot the client and check if users defined on the server can connect.

Default the clients to `init 3` until all testing and debugging is done. Once everything works fine, switch to `init 4`.

**Sources**

- Originally written by Niki Kovacs

howtos, nis, nfs, roaming profiles, centralized authentication, author kikinovak

---

From: https://docs.slackware.com/ - SlackDocs

Permanent link: https://docs.slackware.com/howtos:network_services:roaming_profiles

Last update: 2012/10/09 06:55 (UTC)