8 Network Filesystem einrichten

Um Clients ganze Verzeichnisse von Servern zum Lesen oder Lesen und Schreiben zur Verfügung zu stellen, benutzt man im Unix-Umfeld und generell in heterogenen Umgebungen gern ein spezielles Dateisystem, das *Network File System*, kurz NFS. Vor Samba (siehe Kapitel 9) war dies die einzige Möglichkeit, Windows-Clients Verzeichnisse auf Linux-Servern anzubieten.

Im weiteren Verlauf dieses Buchs lernen Sie im Kapitel 10, stabile und kostengünstige Linux-Arbeitsplätze statt absturzgefährdeter und teurer Windows-PCs zu nutzen. Dabei ist es erforderlich, den Linux-Clients Verzeichnisse auf Festplatten von Linux-Servern zum Lesen und Schreiben und von CD-ROMs/DVDs nur zum Lesen zur Verfügung zu stellen.

Im Kapitel 10 können Sie lesen, wie Sie *Thin Clients* ohne Festplatte einrichten, die sogar ihr gesamtes Linux-Dateisystem per Network File System von einem Linux-Server beziehen.

Wenn Sie mehrere Linux-Server in Ihrem Netz einsetzen, z.B. zur Lastverteilung, oder mit Linux-Clients arbeiten, dann stehen Sie vor dem Problem, dass Sie auf jedem dieser Rechner eine eigenständige Benutzerverwaltung benötigen. Einfacher ist es, wenn Sie alle Benutzer nur einmal auf einem zentralen Anmelde-Server anlegen müssen, von dem die anderen Rechner dann die Anmelde-Daten beziehen. Im Windows-Bereich würden Sie dies mit Anmelde-Servern für Arbeitsgruppen bzw. Domänen erledigen (siehe Kapitel 9).

Eine Lösung für dieses Problem ist NIS, der *Network Information Service*. Dieser Dienst war früher unter dem Namen *YP* (*YellowPages*) zu finden. Aus rechtlichen Gründen darf dieser Name nicht mehr benutzt werden, trotzdem tragen viele der Programmkomponenten und Variablen immer noch YP im Namen.

Eine NIS-Installation nutzt meist NFS, da die Benutzer immer das gleiche Home-Verzeichnis erwarten, egal auf welchem Rechner sie sich anmelden. Daher mountet man in der Regel das Homeverzeichnis vom Anmelde-Server per NFS.

Open Source-NFS-Server und -Clients für Windows-Abarten sind den Autoren bisher nicht bekannt. Daher ist NFS hier nur für Linux-Server und Linux-Clients beschrieben. Stabile lizenzpflichtige NFS-Server und -Clients für Windows gibt es u.a. von Hummingbird.

Um NFS im Linux-Umfeld benutzen zu können, muss man den Linux-Server und den Linux-Client vorbereiten:

Nach dem Einrichten von NFS auf dem Server

- müssen Sie bestimmen, welche Verzeichnisse der Server welchen Clients für welche Zugriffe zur Verfügung stellen soll und
- dann auf den Clients diese Verzeichnisse jeweils in den lokalen Verzeichnisbaum einhängen.

8.1 Einsatzfelder für NFS

NFS brauchen Sie immer dann, wenn sich Linux-Rechner untereinander Laufwerke – dazu gehören auch CD-ROM-Laufwerke – gegenseitig zur Verfügung stellen. Zwar könnten Sie hierzu auch Samba (siehe Kapitel 9) verwenden, generell ist aber der Zugriff per NFS deutlich stabiler als der per Samba.

Da man auf NFS-Dateisysteme schon beim Booten zugreifen kann, lassen sich so große Teile des Filesystems von einem fernen Rechner beziehen.

Der Dateizugriff per NFS ist für Clients vollständig transparent und funktioniert mit sehr unterschiedlichen Serverstrukturen.

8.2 NFS-Server installieren und konfigurieren

Wie viele andere Distributionen auch, installiert SuSE in der Voreinstellung einen NFS-Server.

Den NFS-Server gibt es prinzipiell in zwei Varianten, einmal als Kernel-NFS, andererseits als Userspace-NFS:

Das Kernel-NFS ist direkt im Betriebssystem-Kern verankert und damit deutlich performanter, setzt aber einen entsprechend kompilierten Kernel voraus. Da SuSE die Standard-Kernel mit Kernel-NFS konfiguriert hat, installiert sie standardmäßig kein Userspace-NFS.

Das Userspace-NFS erfordert keinerlei Veränderungen am Kernel, lässt sich also leicht auch nachträglich installieren.

Vom Funktionsumfang her sind beide Versionen identisch. Sie können sogar beide Versionen nebeneinander installieren; welche Version Sie dann starten, legen Sie über Variablen in der Konfigurationsdatei von YaST fest.

8.2.1 Kernel NFS

Falls Sie auf Ihrem System bisher keinerlei NFS-Server installiert haben, so sollten Sie nun die Pakete `nfsutils` und `portmap` der Serie `n` installieren oder die Dateien `nfsutils.rpm` und `portmap.rpm` aus dem Verzeichnis `n1` laden.

Sollten Sie einen eigenen Kernel erstellen, achten Sie bitte darauf, in der Konfigurationsdatei für den Kernel die folgenden Schalter zu aktivieren:

- `CONFIG_NFS_FS` und
- `CONFIG_NFSD`.

8.2.2 User Space NFS

Sollten Sie aus irgendeinem Grund doch das Userspace-NFS nutzen wollen, so müssen Sie das Paket `nfs-server` aus der Serie `n` bzw. die Datei `nfs-server.rpm` aus dem Verzeichnis `n2` nachinstallieren.

Der weitere Teil dieses Kapitels bezieht sich auf Kernel-NFS, das keine weiteren Installationsschritte erfordert.

8.2.3 Der Portmapper

Um NFS nutzen zu können, benötigt man einen Dämon als Service-Vermittler für Client/Server Dienste, die mit *Remote Procedure Calls* (Fern-Aufrufe für Prozeduren) arbeiten, den *RPC-Portmapper*.

Bei einem derartigen Dienst kann ein Client über ein zugehöriges Serverprogramm Prozeduren auf dem Server ausführen. Zu jeder der Prozeduren gehört eine eindeutige Programm-Nummer. Der Portmapper ordnet diesen Programmnummern Ports zu. Wenn Sie die aktuelle Zuordnung mit dem Befehl

```
rpcinfo -p
```

abrufen, erhalten Sie eine Tabelle mit folgendem Aufbau:

```
boss:~ # rpcinfo -p
   Program Vers Proto   Port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100021    1   udp   1032  nlockmgr
    100021    3   udp   1032  nlockmgr
```

```
    100021    4   udp   1032  nlockmgr
    100024    1   udp    964  status
    100024    1   tcp    966  status
    100005    1   udp   1033  mountd
    100005    1   tcp   1058  mountd
    100005    2   udp   1033  mountd
    100005    2   tcp   1058  mountd
    100005    3   udp   1033  mountd
    100005    3   tcp   1058  mountd
```

In der ersten Spalte dieser Tabelle sehen Sie jeweils die Programmnummern für die RPC-Calls, in der vierten Spalte die zugeordneten Ports. Die fünfte Spalte beschreibt die zugeordnete Funktion.

8.2.4 Start des NFS-Servers

Um einen NFS-Server zu aktivieren, muss man den Portmapper und dann den Server in dieser Reihenfolge starten:

Zuerst ruft man den Portmapper über das Startscript

```
rcportmap start
```

auf und danach den eigentlichen Server mit

```
rcnfsserver start
```

Das Bootscript aktiviert diese beiden Programme automatisch, wenn in YaST unter *Administration des Systems* • *Konfigurationsdatei verändern* die folgenden Einstellungen vorhanden sind:

- *START_PORTMAP = yes*
- *NFS_SERVER = yes*
- *USE_KERNEL_NFSD_NUMBER = 4*

Damit ist der NFS-Server einsatzbereit, auch wenn er bisher noch keinerlei Verzeichnisse exportiert.

Im nächsten Schritt müssen Sie dem Server mitteilen, welche Verzeichnisse er an welche Clients exportieren soll.

8.3 Verzeichnisse exportieren

Wenn Sie einen funktionsfähigen NFS-Server eingerichtet haben, müssen Sie noch Verzeichnisse freigeben.

Damit der Server weiß, welche Verzeichnisse er exportieren soll, braucht man diese Verzeichnisse nur in die Datei `/etc/exports` einzutragen. Diese nach der Standardinstallation leere Datei können Sie z.B. folgendermaßen tabellarisch einrichten:

```
# Beispieldatei  /etc/exports
# Zeilen, die mit dem Zeichen # beginnen werden ignoriert
#
/home   *.lokales-netz.de(rw)  www.linuxbu.ch(ro)
/cdrom  (ro)
```

Diese tabellenartige Darstellung in der Form

```
/pfad/zum/verzeichnis   Rechnername(n)(option1,option2,...)
```

gibt drei Daten an:

- Pfad zum Verzeichnis (siehe 8.3.1),
- Rechner, die zugreifen dürfen (siehe 8.3.2) und
- Optionen (siehe 8.33)

Für jedes Verzeichnis können Sie mehrere Rechner/Domains mit den zugehörigen Optionen angeben. Im vorliegenden Beispiel dürfen alle Rechner der Domain `lokales-netz.de` lesend und schreibend auf `/home` zugreifen, der Rechner `www.linuxbu.ch` nur lesend.

Wenn Sie die `/etc/exports` verändert haben, müssen Sie den NFS-Server neu starten, damit er diese Veränderungen registriert. Dazu geben Sie ein:

```
rcnfsserver restart
```

8.3.1 Pfad zum Verzeichnis

Die Angaben des obigen Beispiels exportieren zwei Verzeichnisse, das gesamte Homeverzeichnis mit den Benutzerdaten und das CD-ROM-Laufwerk.

Die Pfadangabe dürfen Sie nicht weglassen, da sonst die Freigabe sinnlos ist. Alle weiteren Angaben dürfen entfallen.

8.3.2 Welche Rechner dürfen zugreifen?

Die zweite Angabe hinter dem Verzeichnisnamen beschränkt die Rechner, die auf diese Freigabe zugreifen dürfen.

Auf das Homeverzeichnis sollen nur Rechner aus dem lokalen Netz zugreifen dürfen. Da die entsprechende Angabe für das CD-ROM-Laufwerk fehlt, dürfen hier alle Rechner, also auch beliebige Rechner aus dem Internet, zugreifen.

Die Rechner, die auf das Verzeichnis zugreifen dürfen, können Sie auf folgende Arten angeben:

1. Einem einzelnen Rechner erlauben Sie den Zugriff, indem Sie seinen Namen oder seine IP angeben.
2. Einer Gruppe von Rechnern können Sie den Zugriff erlauben, indem Sie Rechnernamen angeben, welche die Joker (Wildcards) "*" oder "?" enthalten. Im Beispiel erlauben Sie u.a. dem Rechner *rosine.lokales-netz.de* den Zugriff, da dieser Name der Angabe **.lokales-netz.de* entspricht. Das Wildcardzeichen "*" steht für eine beliebige Zeichenfolge, also auch für *rosine*.
3. Sie können einen IP-Bereich angeben, indem Sie eine IP-Adresse und eine zugehörige Netzwerkmaske angeben. Mit 192.168.1.0/255.255.255.0 (oder auch 192.168.1.0/24) erlauben Sie allen Rechnern, deren IP in den ersten drei Werten 192.168.1. lautet, den Zugriff.
4. Sie erlauben allen Rechnern den Zugriff, indem Sie in dieser Spalte keine Angabe machen, oder ein "*" als Jokerzeichen eintragen.

8.3.3 Optionen

Die dritte Angabe beinhaltet Optionen, hier im Beispiel für Zugriffsrechte.

Die wichtigsten Optionen sind:

Befehl	*Erläuterung*
`rw`	*Read-Write* gibt den Clients Lese- und Schreibrechte für das Verzeichnis.
`ro`	*Read-Only* ist die Voreinstellung, bei der Clients nicht in das Verzeichnis hineinschreiben dürfen.
`root_squash`	Voreinstellung, die privilegierte Zugriffe des Super-Users *root* unterbindet. *Root*- Zugriffe führt der Server nur mit den Rechten des Benutzers *nobody* aus.

Befehl	*Erläuterung*
`no_root_squash`	Das Gegenteil zu obiger Option. Der Super-User *root* kann vom Client aus mit seinen vollen Rechten auf die Dateien auf dem Server zugreifen.
`all_squash`	Der Server führt alle Zugriffe vom Client nur mit den Rechten des Users *nobody* aus.
`noaccess`	Verbietet den Clients den Zugriff auf Unterverzeichnisse; damit kann man einzelne Unterverzeichnisse eines freigegebenen Verzeichnisses sperren.

Tabelle 8.1: Wichtige Optionen für Zugriffssteuerung

Eine vollständige Liste aller Optionen finden Sie in der Manpage von `exports`.

Die Optionen notiert man innerhalb runder Klammern. Mehrere Optionen trennt man durch Kommata ohne Leerzeichen. Zulässig wäre z.B. die Angabe

```
/cdrom      *.lokales-netz.de  (ro,no_root_squash)
```

Hier darf der Superuser mit seinen Rechten nur lesend auf das CD-ROM-Laufwerk zugreifen.

8.4 Netzwerk-Verzeichnisse einbinden

Ein Netzwerkverzeichnis, das auf irgendeinem Rechner freigegeben ist, können Anwender, genauso wie CD-ROM-Laufwerke, mit dem Befehl `mount` in ihr lokales Dateisystem einbinden (mounten), wenn sie über die notwendigen Zugriffsrechte verfügen.

8.4.1 NFS-Zugriff auf linuxbuch

Um Ihnen das Testen zu erleichtern, haben die Autoren ein Verzeichnis auf `linuxbuch.debacher.net` exportiert und für alle Rechner freigegeben; die zugehörige Datei `/etc/exports` hat folgenden Inhalt:

```
# See exports(5) for a description.
# This file contains a list of all directories
# exported to other computers.
# It is used by rpc.nfsd and rpc.mountd.
/usr/local/ftp/pub      *(ro)
```

Auf dieses Verzeichnis können Sie auch mit anonymem FTP (Kapitel 5) zugreifen.

Wenn Sie mit dem Internet verbunden sind, können Sie dieses Verzeichnis in Ihr lokales Filesystem einbinden, indem Sie als root folgenden Befehl eingeben:

```
mount   -t nfs  linuxbuch.debacher.net:/usr/local/ftp/pub
/mnt
```

Anschließend können Sie mit den üblichen Linux-Befehlen zum Anzeigen von Inhaltsverzeichnissen bzw. zum Kopieren von Dateien auf das Verzeichnis `/mnt` zugreifen. Alle Zugriffe auf das Verzeichnis `/mnt` gehen dann auf den Server zu diesem Buch.

Wollen Sie das Verzeichnis nach Ihren Experimenten wieder freigeben, bevor Sie die Internet-Verbindung abbauen, geben Sie ein:

```
umount   /mnt
```

Mit dem Befehl `showmount` kann man abfragen, welche Verzeichnisse ein Rechner per NFS anbietet. Dazu gibt man ein:

```
/usr/sbin/showmount   -e   linuxbuch.debacher.net
```

Der Rechner gibt dann Folgendes aus:

```
root@boss:~ > showmount   -e   linuxbuch.debacher.net
Export list for linuxbuch.debacher.net:
/usr/local/ftp/pub *
```

Auf das Verzeichnis `/usr/local/ftp/pub` können Sie also von jedem Rechner aus zugreifen.

8.4.2 Der Befehl mount

Ein NFS-Client muss wissen, welches Dateisystem er beziehen möchte und an welcher Stelle er es in sein lokales Dateisystem einbinden will. Für diese Festlegungen dient der Befehl `mount`.

Sie kennen aus dem vorangegangenen Abschnitt

```
mount  -t  nfs   linuxbuch.debacher.net:/usr/local/ftp/pub
  ➥ /mnt
```

und vom Einhängen eines CD-ROM-Laufwerks:

```
mount  -t   iso9660   /dev/cdrom   /cdrom
```

Der Mount-Befehl erwartet also Quelle, Ziel und den Typ des Dateisystems (Parameter `-t`):

Der erste Parameter nennt die Quelle, also was in das Dateisystem eingebunden werden soll, in den Beispielen ein Verzeichnis eines anderen Rechners oder ein CD-ROM-Laufwerk. Zwischen dem Rechnernamen und dem Verzeichnis steht immer ein Doppelpunkt; beim CD-ROM-Laufwerk auf dem gleichen Linux-System geben Sie ein Gerät, hier `/dev/cdrom` an, bei einem CD-ROM-Laufwerk auf einem anderen Linux-System den Rechnernamen und die Gerätebezeichnung, hier `linuxbuch.debacher.net:/dev/cdrom`.

Der zweite Parameter gibt an, über welches Verzeichnis die Ressource eingebunden werden soll, den so genannten *Mountpoint*. Die Angabe ist beliebig, das Verzeichnis muss nur existieren und leer sein. Die SuSE-Distribution legt standardmäßig für diesen Zweck die Verzeichnisse `/cdrom` und `/mnt` an. Nach erfolgreichem Mounten finden Sie die eingebundenen Daten in dem vorher leeren Verzeichnis.

Mit dem Parameter `-t` (Typ) können Sie u.a. die folgenden Dateisysteme angeben:

Typ des Dateisystems	*Bedeutung*
`nfs`	Network File System
`iso9660`	Dateisystem auf CD-ROM
`vfat`	Windows-Dateisystem
`ext2`	Linux-Dateisystem
`proc`	Pseudo-Dateisystem

Tabelle 8.2: Dateisysteme

8.4.3 Verzeichnisse permanent in das System einhängen

Nach den bisherigen Beschreibungen darf nur der Super-User root irgendwelche Ressourcen mounten. Praktikabler ist, allen Benutzern das Einhängen (Mounten) von CDs und Disketten zu erlauben. Andere Ressourcen will man schon beim Booten ohne manuellen Eingriff ins System einbinden.

Für dieses permanente Einbinden von Dateisystemen ist die Datei `/etc/fstab` zuständig, über die man auch Festplattenpartitionen einbindet. Bei einer Standardinstallation erzeugt YaST eine Datei in der folgenden Art:

```
/dev/hda5       swap            swap    defaults        0   0
/dev/hda6       /               ext2    defaults        1   1
/dev/hda7       /tmp            ext2    defaults        1   2
/dev/hda8       /var            ext2    defaults        1   2
/dev/hda2       /boot           ext2    defaults        1   2
/dev/hda9       /home           ext2    defaults        1   2

/dev/hdd        /media/cdrom    auto    ro,noauto,user,exec 0

/dev/fd0        /media/floppy   auto    noauto,user     0   0

proc            /proc           proc    defaults        0   0
# End of YaST-generated fstab lines
```

Die Spalten entsprechen den Parametern des Mount-Befehls.

- In der ersten Spalte steht die Datenquelle bzw. das jeweilige Gerät. Eine Angabe wie `/dev/hda5` bezeichnet die Partition *Fünf* der ersten DIE-Festplatte (siehe Kapitel 2, Festplatten vorbereiten). Das Gerät `/dev/hdd` bezeichnet hier ein IDE CD-ROM-Laufwerk und `/dev/fd0` das erste Diskettenlaufwerk.
- In der zweiten Spalte stehen die Einhänge-Ordner (Mountpoints), über die Sie die jeweiligen Geräte im System ansprechen können.
- Die dritte Spalte gibt die Dateisysteme an. Neu gegenüber dem Mount-Befehl ist hier die Angabe `auto`. Bei Einträgen mit diesem Dateityp versucht das System selbst, das Dateisystem zu ermitteln. Das ist bei Wechsel-Datenträgern wie Disketten und CDs sinnvoll. In der vierten Spalte folgen die Optionen, wieder durch Kommata getrennt ohne Leerzeichen. Interessant sind hier die Optionen `noauto` und `user`. Mit der Option `noauto` verhindern Sie, dass die entsprechende Zeile schon beim Hochfahren des Systems aktiviert wird. Das wäre für Wechselmedien nicht sinnvoll. Mit der Option `user` erlauben Sie allen Usern, dieses Dateisystem zu mounten. Die Option `exec` erlaubt zusätzlich das Ausführen von Programmen im Dateisystem. In der oben dargestellten Konfiguration können Sie also keine Programme von einer Diskette aus starten.
- Die Spalten fünf und sechs steuern das Sichern bzw. Überprüfen von Dateisystemen.
- Bei ext2-Partitionen sollte in der fünften Spalte eine `1` stehen, ansonsten eine `0`. Wenn in der fünften Spalte eine `1` steht, dann sollte in der sechsten Spalte eine `2` stehen, außer beim Wurzelverzeichnis, das kennzeichnen Sie mit einer `1`. Die `0` gibt an, dass der Dämon das entsprechende Verzeichnis beim Mounten nicht testen soll. Das Wurzelverzeichnis testet er vorrangig, alle anderen Verzeichnisse später.

Um ein Verzeichnis per NFS automatisch zu beziehen, können Sie in die Datei `/etc/fstab` eine weitere Zeile aufnehmen:

```
/dev/hda5       swap            swap      defaults       0   0
/dev/hda6       /               ext2      defaults       1   1
/dev/hda7       /tmp            ext2      defaults       1   2
/dev/hda8       /var            ext2      defaults       1   2
/dev/hda2       /boot           ext2      defaults       1   2
/dev/hda9       /home           ext2      defaults       1   2

/dev/hdd        /media/cdrom    auto      ro,noauto,user,exec 0

/dev/fd0        /media/floppy   auto      noauto,user    0   0

proc            /proc           proc      defaults       0   0
# End of YaST-generated fstab lines

boss.lokales-netz.de:/cdrom
                    ➥ /mnt     nfs   ro                 0   0
```

8.5 NFS-Probleme aufspüren und beheben

Sind auf einem Server notwendige Dämonen nicht aktiviert oder fehlen gewünschte Freigaben, erleben Anwender dies als Fehler beim Mounten von Verzeichnissen. Wenn Sie auf dem Server Root-Rechte besitzen, können Sie den Status der Server-Programme überprüfen.

```
rcportmap status
```

Sie sollten ein einfaches `OK` als Antwort erhalten.

Testen Sie danach, ob auch der NFS-Server läuft, mit

```
rcnfsserver status
```

Sie sollten hier die Meldung `NFS server up` erhalten.

Sollte einer der Dienste nicht aktiv sein, so überprüfen Sie die Einstellungen in YaST und starten die Dienste per Hand.

Sollte bis hierher alles korrekt aussehen, so fehlt es an der Freigabe, eventuell wurde der NFS-Server nach Änderungen nicht neu gestartet. Ob eine Freigabe auf Ihrem Rechner aktiv ist, können Sie jederzeit testen mit

```
/usr/sbin/showmount   -e
```

Wollen Sie einen fremden Rechner untersuchen, so hängen Sie wie oben beschreiben den Rechnernamen als Parameter an den Befehl an:

```
/usr/sbin/showmount   -e  linuxbuch.debacher.net
```

Falls die Freigabe nur für bestimmte Rechner gilt, lohnt sich auch ein Blick in die Datei `/var/log/messages` des freigebenden Rechners. Diese protokolliert alle Mount-Versuche und auch den Grund für eine eventuelle Ablehnung.

8.6 NIS

Der *Network Information Service* NIS benötigt einen NIS-Server, der die Benutzerdaten für seine NIS-Domain verwaltet. Zu dieser NIS-Domain können beliebig viele NIS-Clients gehören. In größeren Domainen kann es sinnvoll sein, zusätzlich Slave-Server einzusetzen, die beim Ausfall des Hauptservers dessen Aufgabe übernehmen können. Auf Slave-Server soll hier nicht weiter eingegangen werden.
In den Beispieldateien dieses Kapitels heißt die NIS-Domain `lokales-netz`. Die Bezeichnung können Sie recht frei wählen, es muss keine offizielle DNS-Domain sein.

Neben NIS gibt es noch eine aktuellere Implementierung NIS+. NIS+ überträgt die Benutzerdaten verschlüsselt übers Netz. Der Vorteil von NIS+ besteht in höherer Sicherheit, dafür ist die Konfiguration deutlich aufwändiger. Der folgende Text beschreibt NIS.

8.7 NIS Server-Installation

Auf dem Anmelde-Server müssen die Pakete *ypserv*, *ypbind* und *yp-tools* installiert sein, die Sie bei SuSE in der Serie *n1* finden.

Nach der Installation der Pakete müssen Sie unter *YaST • Administration des Systems • Konfigurationsdatei verändern* einige Werte einstellen.

Zuerst geben Sie einen Domainnamen an, dessen Namen Sie später auch auf den Clients angeben.

```
YP_DOMAINNAME="lokales-netz"
```

Weiter müssen Sie verhindern, dass YaST die Konfigurationsdatei für Clients erzeugt.

```
CREATE_YP_CONF="no"
```

Außerdem müssen Sie die notwendigen Serverdienste starten lassen.

```
START_YPSERV="yes"
```

Danach können Sie den NIS-Server durch einen Reboot aktivieren, oder an der Konsole eingeben:

```
domainname lokales-netz
rcypserv start
```

Nun müssen Sie noch erreichen, dass der NIS-Server die Daten aus den Benutzerdateien

- `/etc/passwd`
- `/etc/shadow`
- `/etc/group`
- …

bekommt. Dazu dient ein Aufruf des Programmes `make`. Dem Programm geben Sie über den Schalter `-C /var/yp` das Verzeichnis an, mit dem es arbeiten soll. Der Schalter `-s` (silent) unterbindet Ausgaben.

```
make  -s  -C  /var/yp
```

Dies übersetzt die Benutzerdaten in die Dateien für NIS. Sie finden die erzeugten Dateien im Verzeichnis `/var/yp/lokales-netz/`. Die Dateien liegen in einem speziellen Datenbank-Format vor, das schneller auswertbar ist als eine einfache Textdatei.

Da NIS leider nichts über Änderungen in den Benutzerdateien erfährt, müssen Sie diesen Befehl regelmäßig aufrufen, im einfachsten Fall über einen Cronjob. Ergänzen Sie die Crontab um die folgenden Zeile:

```
*/15  *  *  *  *  make  -s  -C  /var/yp
```

Damit sind neue Benutzer und geänderte Passworte spätestens nach 15 Minuten in der gesamten NIS-Domain bekannt.

Welche Daten der NIS-Server verteilen darf, legen Sie mit der Datei `/var/yp/Makefile` fest, die vom make-Aufruf ausgewertet wird. Sie können hier mit

```
MINUID=100
MINGID=100
```

festlegen, dass er nur Benutzer bzw. Gruppen ab der genannten ID exportiert.

Die Hauptrisiken von NIS ergeben sich aus den Zeilen

```
MERGE_PASSWD=true
MERGE_GROUP=true
```

NIS kann nämlich nicht mit Shadow-Passwörtern umgehen und fügt daher die Daten aus den Dateien `/etc/passwd` und `/etc/shadow` wieder zu einer Datei zusammen, zumindest für den Export.

Welche Export-Dateien NIS anlegt, bestimmt die Zeile

```
all:  passwd  group  rpc  services  netid
```

8.8 NIS Client-Installation

Für den Client müssen Sie die Pakete *ypbind* und *yp-tools* installieren. Danach stellen Sie in *YaST • Administration des Systems • Konfigurationsdatei verändern* einige Parameter ein

```
YP_DOMAINNAME="lokales-netz"
```

Der Client muss wissen, wie er den NIS-Server findet. Am sichersten ist es, hier die IP-Adresse des Servers anzugeben.

```
YP_SERVERS="192.168.1.2"
```

Weiter müssen Sie erreichen, dass YaST die Konfigurationsdatei für Clients erzeugt.

```
CREATE_YP_CONF="yes"
```

Und natürlich müssen Sie das Client-Programm starten.

```
START_YPBIND="yes"
```

Beim Beenden von YaST verändert SuSEconfig auf dem Client die Dateien `/etc/passwd` und `/etc/group`, indem es eine Zeile

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

an die Datei anhängt. Die Datei `/etc/passwd` sieht dann z.B. folgendermaßen aus (Auszug, Ende der Datei).

```
.....
pop:x:67:100:pop admin:/var/lib/pop:/bin/false
perforce:x:68:60:perfoce admin:/var/lib/perforce:/bin/false
sapdb:x:69:61:SAPDB demo user:/var/opt/sapdb:/bin/false
nobody:x:65534:65534:nobody:/var/lib/nobody:/bin/bash
debacher:x:500:100:Uwe Debacher:/localhome/debacher/:/bin/bash
+::::::
```

Sie sehen hier in der Beispieldatei die von SuSE vorgegebenen Systembenutzer wie *sapdb* und *nobody*, sowie einen lokalen Benutzer *debacher*. Die Daten aller weiteren Benutzer bekommt der Rechner über den NIS-Server.

Nach dem Beenden von YaST aktivieren Sie den Client, indem Sie entweder den Rechner rebooten, oder an der Konsole eingeben

```
domainname lokales-netz
rcypbind start
```

Beim Start versucht das NIS-Client-Programm, Kontakt zu einem NIS-Server zu bekommen, und gibt eine entsprechende Meldung aus.

8.9 Die Home-Verzeichnisse

Im Prinzip kann sich nach dem Start des NIS-Servers ein Benutzer auf jedem Rechner anmelden, auf dem der NIS-Client läuft.

Wenn Sie das gleich ausprobieren, dann werden Sie feststellen, dass ein Login mit falschen Daten abgelehnt wird, Sie aber mit richtigen Eingaben sofort wieder im Anmeldebildschirm landen, da die Benutzer auf dem Client bisher keine Home-Verzeichnisse besitzen. Statt auf jedem Client für jeden Benutzer ein Home-Verzeichnis anzulegen, mounten Sie besser die Home-Verzeichnisse vom Anmelde-Server.

Im einfachsten Fall exportieren Sie auf dem Anmelde-Server das komplette Home-Verzeichnis und mounten dies dann auf den Client-Rechnern entsprechend.

Zum Exportieren müssen Sie auf dem NIS-Server folgende Zeile in Ihre Datei `/etc/exports` aufnehmen.

```
/home   *.lokales-netz.de(rw)
```

Damit erlauben Sie, dass jeder Rechner aus der Domain `lokales-netz.de` dieses Verzeichnis zum Lesen und Schreiben mounten darf. Falls Sicherheit keine so große Rolle spielt, könnten Sie im einfachsten Fall auch schreiben

```
/home   *(rw)
```

Falls Sie höhere Sicherheitsansprüche besitzen, können Sie auch gezielt nur einzelnen Rechnern das Mounten erlauben.

```
/home   rosine.lokales-netz.de(rw)
  ➥ zitrone.lokales-netz.de(rw)
```

Damit steht dem genannten Client-Rechner dieses Verzeichnis mit allen darin befindlichen Home-Verzeichnissen zur Verfügung.

Auf den Client-Rechnern können Sie dieses Verzeichnis generell ganz mounten, indem Sie die Datei `/etc/fstab` um eine Zeile erweitern.

```
192.168.1.2:/home      /home   nfs     defaults 0 0
```

Damit mounten Sie das Verzeichnis `/home` des NIS-Servers in das Verzeichnis `/home` auf dem Client. Da Sie das Verzeichnis des Servers nur in ein leeres Verzeichnis auf dem lokalen Rechner mounten können, dürfen die Home-Verzeichnisse eventueller lokaler Benutzer nicht in `/home` liegen. Legen Sie für diesen Fall ein Verzeichnis `/localhome` für die Homeverzeichnisse der lokalen Benutzer an.

Damit sollten sich auch Benutzer, die nur auf dem Server angelegt sind, am Client anmelden und am Client arbeiten können. Viel Spaß bei der Arbeit in der NIS-Domain.

8.10 NIS Feintuning

Mit den bisherigen Beschreibungen arbeitet das NIS-System bereits einwandfrei. Für die praktische Arbeit und vor allem die System-Sicherheit gibt es aber noch ein paar Optimierungsmöglichkeiten.

8.10.1 Passwort-Änderungen

Interessant wird es, wenn ein Benutzer beim Arbeiten auf einem Client-Rechner sein zentrales NIS-Passwort ändern möchte. Das dafür übliche Programm `passwd` greift nur auf die lokalen Dateien zu und bricht mit einer Fehlermeldung ab.

Um den Benutzern das Ändern ihres Passworts im gesamten Netzwerk zu ermöglichen, muss ein weiterer Dienst, der Passwortdämon `YPPASSWDD`, gestartet werden mit:

```
START_YPPASSWDD = yes
```

Das zweite `d` im Befehl gibt an, dass es sich um den Dämon handelt und es ist darauf zu achten, diesen Befehl nicht mit dem `yppasswd` auf dem Client zu verwechseln.

Um diesen Dienst ohne Neustart aktivieren zu können, starten Sie den Dämon per Hand.

```
rcyppasswdd  start
```

Nun kann ein Benutzer sein Passwort ändern, indem er auf dem Client-Rechner das Programm `yppasswd` aufruft.

Wenn Sie ein versehentliches Benutzen des alten Programmes `passwd` vermeiden wollen, dann sollten Sie dieses durch einen Link auf `yppasswd` ersetzen.

```
cd /usr/bin
mv passwd passwd.orig
ln -s yppasswd passwd
```

NIS-Benutzer rufen einfach `passwd` auf, die lokalen Benutzer können dann ihr lokales Passwort immer noch durch einen Aufruf von `passwd.orig` ändern.

8.10.2 Vertrauenswürdige Rechner

Wollen Sie den NIS-Zugriff auf bestimmte Rechner beschränken, so können Sie in die Datei `/var/yp/securenets` die Einschränkungen eintragen. Voreingestellt erlaubt dort am Ende die Zeile

```
0.0.0.0                 0.0.0.0
```

allen Rechnern den Zugriff. Angeben müssen Sie hier als erste Zahl eine Netzmaske und als zweite Zahl eine IP-Adresse.

Mit

```
255.255.255.0                 192.168.1.0
```

erlauben Sie nur Rechnern aus Ihrem lokalen Netz den Zugriff auf den NIS-Server. Sie müssen dann natürlich die ursprüngliche Zeile entfernen. Sowie die Client-IP nämlich eine der Regeln erfüllt, darf der Rechner zugreifen.

8.10.3 Vertrauen in die Benutzer

Durchaus nützliche Tools von NIS-Systemen bergen gewisse Risiken.
Mit dem NIS-Programm `ypcat` können Sie bzw. Ihre Benutzer eine *Mapdatei* lesen.

```
ypcat  passwd
```

zeigt Benutzern die komplette Passwort-Datei an. Einen bestimmten Datensatz können Sie dann abrufen.

```
ypcat  debacher  passwd
```

würde also den Datensatz für den Benutzer *debacher* liefern.

Neuere Systeme bieten den Befehl `getent` mit den gleichen Funktionen.

In den Datensätzen tauchen zwar nur die verschlüsselten Passwörter auf, das ist aber trotzdem riskant. Passwortdateien lassen sich mit einer gewissen Chance knacken, indem man ein großes Wörterbuch benutzt, jedes Wort verschlüsselt und dann mit den verschlüsselten Passwörtern vergleicht. Auf nahezu jedem System lässt sich ein großer Teil der Passwörter so knacken.

Sie sollten den normalen Benutzern die Zugriffsrechte auf diese Dateien wegnehmen, indem Sie die Dateirechte auf 500 ändern.

```
chmod  500  /usr/bin/ypcat
```

Zusammen mit Samba (siehe Kapitel 9) gibt Ihnen NIS die Möglichkeit, auch in größeren und heterogenen Netzen mit nur einem einzigen Anmelde-Server zu arbeiten. Nur auf diesem Server müssen Sie Ihre Benutzerdaten pflegen und verwalten. Dieser Server sollte dann aber über genügend Plattenkapazität für die Homeverzeichnisse verfügen.