The Linux NIS(YP)/NYS/NIS+ HOWTO

Thorsten Kukuk

v1.1.1, 18 November 2000

This document describes how to configure Linux as NIS(YP) or NIS+ client and how to install as NIS server.

Table of Contents

1. Introduction **1**
- 1.1. New Versions of this Document 1
- 1.2. Disclaimer 1
- 1.3. Feedback and Corrections 2
- 1.4. Acknowledgements 2

2. Glossary and General Information **3**
- 2.1. Glossary of Terms 3
- 2.2. Some General Information 4

3. NIS, NYS or NIS+ ? **5**
- 3.1. libc 4/5 with traditional NIS or NYS ? 5
- 3.2. glibc 2 and NIS/NIS+ 5
- 3.3. NIS or NIS+ ? 5

4. How it works **6**
- 4.1. How NIS works 6
- 4.2. How NIS+ works 6

5. The RPC Portmapper **7**

6. What do you need to set up NIS? **8**
- 6.1. Determine whether you are a Server, Slave or Client. 8
- 6.2. The Software 8

7. Setting Up the NIS Client **10**
- 7.1. The ypbind daemon 10
- 7.2. Setting up a NIS Client using Traditional NIS 11
- 7.3. Setting up a NIS Client using NYS 12
- 7.4. Setting up a NIS Client using glibc 2.x 12
- 7.5. The nsswitch.conf File 13
- 7.6. Shadow Passwords with NIS 14
  - 7.6.1. Linux 14
  - 7.6.2. Solaris 14
  - 7.6.3. PAM 14

8. What do you need to set up NIS+ ? **16**
- 8.1. The Software 16
- 8.2. Setting up a NIS+ client 16
- 8.3. NIS+, keylogin, login and PAM 17
- 8.4. The nsswitch.conf File 17

9. Setting up a NIS Server **19**
- 9.1. The Server Program ypserv 19
- 9.2. The Server Program yps 21
- 9.3. The Program rpc.ypxfrd 22
- 9.4. The Program rpc.yppasswdd 22

Table of Contents

10. Verifying the NIS/NYS Installation 23

11. Surviving a Reboot 25
- 11.1. NIS Init Script 25
- 11.2. NIS Domain Name 25
- 11.3. Distribution−specific Issues 25
  - 11.3.1. Caldera 2.x 25
  - 11.3.2. Debian 25
  - 11.3.3. Red Hat 6.x 25
  - 11.3.4. SuSE Linux 26

12. Common Problems and Troubleshooting NIS 27

13. Frequently Asked Questions 28

1. Introduction

More and more, Linux machines are installed as part of a network of computers. To simplify network administration, most networks (mostly Sun−based networks) run the Network Information Service. Linux machines can take full advantage of existing NIS service or provide NIS service themselves. Linux machines can also act as full NIS+ clients, this support is in beta stage.

This document tries to answer questions about setting up NIS(YP) and NIS+ on your Linux machine. Don't forget to read Section 5.

The NIS−Howto is edited and maintained by

```
        Thorsten Kukuk, <kukuk@suse.de>
```

The primary source of the information for the initial NIS−Howto was from:

```
Andrea Dell'Amico       <adellam@ZIA.ms.it>
Mitchum DSouza          <Mitch.DSouza@NetComm.IE>
Erwin Embsen            <erwin@nioz.nl>
Peter Eriksson          <peter@ifm.liu.se>
```

who we should thank for writing the first versions of this document.

1.1. New Versions of this Document

You can always view the latest version of this document on the World Wide Web via the URL http://www.suse.de/~kukuk/nis−howto/HOWTO/NIS−HOWTO.html.

New versions of this document will also be uploaded to various Linux WWW and FTP sites, including the LDP home page.

Links to translations of this document could be found at http://www.suse.de/~kukuk/nis−howto/.

1.2. Disclaimer

Although this document has been put together to the best of my knowledge it may, and probably does contain errors. Please read any README files that are bundled with any of the various pieces of software described in this document for more detailed and accurate information. I will attempt to keep this document as error free as possible.

1.3. Feedback and Corrections

If you have questions or comments about this document, please feel free to mail Thorsten Kukuk, at kukuk@suse.de. I welcome any suggestions or criticisms. If you find a mistake with this document, please let me know so I can correct it in the next version. Thanks.

Please do *not* mail me questions about special problems with your Linux Distribution! I don't know every Linux Distribution. But I will try to add every solution you send me.

1.4. Acknowledgements

We would like to thank all the people who have contributed (directly or indirectly) to this document. In alphabetical order:

```
Byron A Jeff            <byron@cc.gatech.edu>
Markus Rex              <msrex@suse.de>
Miquel van Smoorenburg  <miquels@cistron.nl>
Dan York                <dyork@lodestar2.com>
```

Theo de Raadt is responsible for the original yp−clients code. Swen Thuemmler ported the yp−clients code to Linux and also ported the yp−routines in libc (again based on Theo's work). Thorsten Kukuk has written the NIS(YP) and NIS+ routines for GNU libc 2.x from scratch.

2. Glossary and General Information

2.1. Glossary of Terms

In this document a lot of acronyms are used. Here are the most important acronyms and a brief explanation:

DBM

DataBase Management, a library of functions which maintain key−content pairs in a data base.

DLL

Dynamically Linked Library, a library linked to an executable program at run−time.

domainname

A name "key" that is used by NIS clients to be able to locate a suitable NIS server that serves that domainname key. Please note that this does not necessarily have anything at all to do with the DNS "domain" (machine name) of the machine(s).

FTP

File Transfer Protocol, a protocol used to transfer files between two computers.

libnsl

Name services library, a library of name service calls (getpwnam, getservbyname, etc...) on SVR4 Unixes. GNU libc uses this for the NIS (YP) and NIS+ functions.

libsocket

Socket services library, a library for the socket service calls (socket, bind, listen, etc...) on SVR4 Unixes.

NIS

Network Information Service, a service that provides information, that has to be known throughout the network, to all machines on the network. There is support for NIS in Linux's standard libc library, which in the following text is referred to as "traditional NIS".

NIS+

Network Information Service (Plus :−), essentially NIS on steroids. NIS+ is designed by Sun Microsystems Inc. as a replacement for NIS with better security and better handling of _large_ installations.

NYS

This is the name of a project and stands for NIS+, YP and Switch and is managed by Peter Eriksson <peter@ifm.liu.se>. It contains among other things a complete reimplementation of the NIS (= YP) code that uses the Name Services Switch functionality of the NYS library.

NSS

Name Service Switch. The /etc/nsswitch.conf file determines the order of lookups performed when a certain piece of information is requested.

RPC

Remote Procedure Call. RPC routines allow C programs to make procedure calls on other machines across the network. When people talk about RPC they most often mean the Sun RPC variant.

YP

Yellow Pages(tm), a registered trademark in the UK of British Telecom plc.

TCP−IP

Transmission Control Protocol/Internet Protocol. It is the data communication protocol most often used on Unix machines.

2.2. Some General Information

The next four lines are quoted from the Sun(tm) System & Network Administration Manual:

```
    "NIS was formerly known as Sun Yellow Pages (YP) but
     the name Yellow Pages(tm) is a registered trademark
     in the United Kingdom of British Telecom plc and may
     not be used without permission."
```

NIS stands for Network Information Service. Its purpose is to provide information, that has to be known throughout the network, to all machines on the network. Information likely to be distributed by NIS is:

- login names/passwords/home directories (/etc/passwd)
- group information (/etc/group)

If, for example, your password entry is recorded in the NIS passwd database, you will be able to login on all machines on the network which have the NIS client programs running.

Sun is a trademark of Sun Microsystems, Inc. licensed to SunSoft, Inc.

3. NIS, NYS or NIS+ ?

3.1. libc 4/5 with traditional NIS or NYS ?

The choice between "traditional NIS" or the NIS code in the NYS library is a choice between laziness and maturity vs. flexibility and love of adventure.

The "traditional NIS" code is in the standard C library and has been around longer and sometimes suffers from its age and slight inflexibility.

The NIS code in the NYS library requires you to recompile the libc library to include the NYS code into it (or maybe you can get a precompiled version of libc from someone who has already done it).

Another difference is that the traditional NIS code has some support for NIS Netgroups, which the NYS code doesn't. On the other hand the NYS code allows you to handle Shadow Passwords in a transparent way. The "traditonal NIS" code doesn't support Shadow Passwords over NIS.

3.2. glibc 2 and NIS/NIS+

Forgot all this if you use the new GNU C Library 2.x (aka libc6). It has real NSS (name switch service) support, which makes it very flexible, and contains support for the following NIS/NIS+ maps: aliases, ethers, group, hosts, netgroups, networks, protocols, publickey, passwd, rpc, services and shadow. The GNU C Library has no problems with shadow passwords over NIS.

3.3. NIS or NIS+ ?

The choice between NIS and NIS+ is easy – use NIS if you don't have to use NIS+ or have severe security needs. NIS+ is _much_ more problematic to administer (it's pretty easy to handle on the client side, but the server side is horrible). Another problem is that the support for NIS+ under Linux is still under developement – you need the latest glibc 2.1. There is an unsupported port of the glibc NIS+ support for libc5 as dropin replacement.

---

4. How it works

4.1. How NIS works

Within a network there must be at least one machine acting as a NIS server. You can have multiple NIS servers, each serving different NIS "domains" − or you can have cooperating NIS servers, where one is the master NIS server, and all the other are so−called slave NIS servers (for a certain NIS "domain", that is!) − or you can have a mix of them...

Slave servers only have copies of the NIS databases and receive these copies from the master NIS server whenever changes are made to the master's databases. Depending on the number of machines in your network and the reliability of your network, you might decide to install one or more slave servers. Whenever a NIS server goes down or is too slow in responding to requests, a NIS client connected to that server will try to find one that is up or faster.

NIS databases are in so−called DBM format, derived from ASCII databases. For example, the files `/etc/passwd` and `/etc/group` can be directly converted to DBM format using ASCII−to−DBM translation software ("makedbm", included with the server software). The master NIS server should have both, the ASCII databases and the DBM databases.

Slave servers will be notified of any change to the NIS maps, (via the "yppush" program), and automatically retrieve the necessary changes in order to synchronize their databases. NIS clients do not need to do this since they always talk to the NIS server to read the information stored in it's DBM databases.

Old ypbind versions do a broadcast to find a running NIS server. This is insecure, due the fact that anyone may install a NIS server and answer the broadcast queries. Newer Versions of ypbind (ypbind−3.3 or ypbind−mt) are able to get the server from a configuration file − thus no need to broadcast.

4.2. How NIS+ works

NIS+ is a new version of the network information nameservice from Sun. The biggest difference between NIS and NIS+ is that NIS+ has support for data encryption and authentication over secure RPC.

The naming model of NIS+ is based upon a tree structure. Each node in the tree corresponds to an NIS+ object, from which we have six types: directory, entry, group, link, table and private.

The NIS+ directory that forms the root of the NIS+ namespace is called the root directory. There are two special NIS+ directories: org_dir and groups_dir. The org_dir directory consists of all administration tables, such as passwd, hosts, and mail_aliases. The groups_dir directory consists of NIS+ group objects which are used for access control. The collection of org_dir, groups_dir and their parent directory is referred to as an NIS+ domain.

5. The RPC Portmapper

To run any of the software mentioned below you will need to run the program /usr/sbin/portmap. Some Linux distributions already have the code in the /sbin/init.d/ or /etc/rc.d/ files to start up this daemon. All you have to do is to activate it and reboot your Linux machine. Read your Linux Distribution Documentation how to do this.

The RPC portmapper (portmap(8)) is a server that converts RPC program numbers into TCP/IP (or UDP/IP) protocol port numbers. It must be running in order to make RPC calls (which is what the NIS/NIS+ client software does) to RPC servers (like a NIS or NIS+ server) on that machine. When an RPC server is started, it will tell portmap what port number it is listening to, and what RPC program numbers it is prepared to serve. When a client wishes to make an RPC call to a given program number, it will first contact portmap on the server machine to determine the port number where RPC packets should be sent.

Since RPC servers could be started by inetd(8), portmap should be running before inetd is started.

For secure RPC, the portmapper needs the Time service. Make sure, that the Time service is enabled in /etc/inetd.conf on all hosts:

```
#
# Time service is used for clock syncronization.
#
time    stream  tcp     nowait  root    internal
time    dgram   udp     wait    root    internal
```

IMPORTANT: Don't forget to restart inetd after changes on its configuration file !

6. What do you need to set up NIS?

6.1. Determine whether you are a Server, Slave or Client.

To answer this question you have to consider two cases:

1. Your machine is going to be part of a network with existing NIS servers
2. You do not have any NIS servers in the network yet

In the first case, you only need the client programs (ypbind, ypwhich, ypcat, yppoll, ypmatch). The most important program is ypbind. This program must be running at all times, which means, it should always appear in the list of processes. It is a daemon process and needs to be started from the system's startup file (eg. /etc/init.d/nis, /sbin/init.d/ypclient, /etc/rc.d/init.d/ypbind, /etc/rc.local). As soon as ypbind is running your system has become a NIS client.

In the second case, if you don't have NIS servers, then you will also need a NIS server program (usually called ypserv). Section 9 describes how to set up a NIS server on your Linux machine using the "ypserv" daemon.

6.2. The Software

The system library "/usr/lib/libc.a" (version 4.4.2 and better) or the shared library "/lib/libc.so.x" contain all necessary system calls to succesfully compile the NIS client and server software. For the GNU C Library 2 (glibc 2.x), you also need /lib/libnsl.so.1.

Some people reported that NIS only works with "/usr/lib/libc.a" version 4.5.21 and better so if you want to play it safe don't use older libc's. The NIS client software can be obtained from:

```
Site                 Directory                        File Name

ftp.kernel.org       /pub/linux/utils/net/NIS         yp-tools-2.4.tar.gz
ftp.kernel.org       /pub/linux/utils/net/NIS         ypbind-mt-1.7.tar.gz
ftp.kernel.org       /pub/linux/utils/net/NIS         ypbind-3.3.tar.gz
ftp.kernel.org       /pub/linux/utils/net/NIS         ypbind-3.3-glibc5.diff.gz
```

Once you obtained the software, please follow the instructions which come with the software. yp−clients 2.2 are for use with libc4 and libc5 until 5.4.20. libc 5.4.21 and glibc 2.x needs yp−tools 1.4.1 or later. The new yp−tools 2.4 should work with every Linux libc. Since there was a bug in the NIS code, you shouldn't use libc 5.4.21−5.4.35. Use libc 5.4.36 or later instead, or the most YP programs will not work. ypbind 3.3 will work with all libraries, too. If you use gcc 2.8.x or greater, egcs or glibc 2.x, you should add the ypbind−3.3−glibc5.diff patch to ypbind 3.3. If possible you should avoid the use of ypbind 3.3 for security reasons. ypbind−mt is a new, multithreaded daemon. It needs a Linux 2.2 kernel and glibc 2.1 or later.

7. Setting Up the NIS Client

7.1. The ypbind daemon

After you have succesfully compiled the software you are now ready to install it. A suitable place for the ypbind daemon is the directory /usr/sbin. Some people may tell you that you don't need ypbind on a system with NYS. This is wrong. ypwhich and ypcat need it always.

You must do this as root of course. The other binaries (ypwhich, ypcat, yppasswd, yppoll, ypmatch) should go in a directory accessible by all users, normally /usr/bin.

Newer ypbind versions have a configuration file called /etc/yp.conf. You can hardcode a NIS server there − for more info see the manual page for ypbind(8). You also need this file for NYS. An example:

```
ypserver 10.10.0.1
ypserver 10.0.100.8
ypserver 10.3.1.1
```

If the system cam resolv the hostnames without NIS, you may use the name, otherwise you have to use the IP address. ypbind 3.3 has a bug and will only use the last entry (ypserver 10.3.1.1 in the example). All other entries are ignored. ypbind−mt handle this correct and uses that one, which answerd at first.

It might be a good idea to test ypbind before incorporating it in the startup files. To test ypbind do the following:

- Make sure you have your YP−domain name set. If it is not set then issue the command:
  ```
              /bin/domainname nis.domain13;
  ```
  where `nis.domain` should be some string _NOT_ normally associated with the DNS−domain name of your machine! The reason for this is that it makes it a little harder for external crackers to retreive the password database from your NIS servers. If you don't know what the NIS domain name is on your network, ask your system/network administrator.
- Start up "/usr/sbin/portmap" if it is not already running.
- Create the directory "/var/yp" if it does not exist.
- Start up "/usr/sbin/ypbind"
- Use the command "rpcinfo −p localhost" to check if ypbind was able to register its service with the portmapper. The output should look like:
  ```
         program vers proto   port
          100000    2   tcp    111  portmapper
          100000    2   udp    111  portmapper
          100007    2   udp    637  ypbind
          100007    2   tcp    639  ypbind13;
  ```
  or
  ```
         program vers proto   port
          100000    2   tcp    111  portmapper
          100000    2   udp    111  portmapper
          100007    2   udp    758  ypbind
          100007    1   udp    758  ypbind
          100007    2   tcp    761  ypbind
  ```

```
        100007    1   tcp    761  ypbind13;
```

Depending on the ypbind version you are using.

* You may also run "rpcinfo −u localhost ypbind". This command should produce something like:

```
        program 100007 version 2 ready and waiting13;
```

or

```
        program 100007 version 1 ready and waiting
        program 100007 version 2 ready and waiting13;
```

The output depends on the ypbind version you have installed. Important is only the "version 2" message.

At this point you should be able to use NIS client programs like ypcat, etc... For example, "ypcat passwd.byname" will give you the entire NIS password database.

IMPORTANT: If you skipped the test procedure then make sure you have set the domain name, and created the directory

```
    /var/yp
```

This directory MUST exist for ypbind to start up succesfully.

To check if the domainname is set correct, use the /bin/ypdomainname from yp−tools 2.2. It uses the yp_get_default_domain() function which is more restrict. It doesn't allow for example the "(none)" domainname, which is the default under Linux and makes a lot of problems.

If the test worked you may now want to change your startupd files so that ypbind will be started at boot time and your system will act as a NIS client. Make sure that the domainname will be set before you start ypbind.

Well, that's it. Reboot the machine and watch the boot messages to see if ypbind is actually started.

7.2. Setting up a NIS Client using Traditional NIS

For host lookups you must set (or add) "nis" to the lookup order line in your /etc/host.conf file. Please read the manpage "resolv+.8" for more details.

Add the following line to /etc/passwd on your NIS clients:

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

You can also use the + and − characters to include/exclude or change users. If you want to exclude the user guest just add −guest to your /etc/passwd file. You want to use a different shell (e.g. ksh) for the user "linux"? No problem, just add "+linux::::::/bin/ksh" (without the quotes) to your /etc/passwd. Fields that you don't want to change have to be left empty. You could also use Netgroups for user control.

For example, to allow login−access only to miquels, dth and ed, and all members of the sysadmin netgroup, but to have the account data of all other users available use:

```
        +miquels:::::::
        +ed:::::::
        +dth:::::::
        +@sysadmins:::::::
        -ftp
        +:*::::::/etc/NoShell
```

Note that in Linux you can also override the password field, as we did in this example. We also remove the login "ftp", so it isn't known any longer, and anonymous ftp will not work.

The netgroup would look like

```
sysadmins (-,software,) (-,kukuk,)
```

IMPORTANT: The netgroup feature is implemented starting from libc 4.5.26. If you have a version of libc earlier than 4.5.26, every user in the NIS password database can access your linux machine if you run "ypbind" !

7.3. Setting up a NIS Client using NYS

All that is required is that the NIS configuration file (/etc/yp.conf) points to the correct server(s) for its information. Also, the Name Services Switch configuration file (/etc/nsswitch.conf) must be correctly set up.

You should install ypbind. It isn't needed by the libc, but the NIS(YP) tools need it.

If you wish to use the include/exclude user feature (+/−guest/+@admins), you have to use "passwd: compat" and "group: compat" in nsswitch.conf. Note that there is no "shadow: compat"! You have to use "shadow: files nis" in this case.

The NYS sources are part of the libc 5 sources. When run configure, say the first time "NO" to the "Values correct" question, then say "YES" to "Build a NYS libc from nys".

7.4. Setting up a NIS Client using glibc 2.x

The glibc uses "traditional NIS", so you need to start ypbind. The Name Services Switch configuration file (/etc/nsswitch.conf) must be correctly set up. If you use the compat mode for passwd, shadow or group, you have to add the "+" at the end of this files and you can use the include/exclude user feature. The configuration is excatly the same as under Solaris 2.x.

7.5. The nsswitch.conf File

The Network Services switch file /etc/nsswitch.conf determines the order of lookups performed when a certain piece of information is requested, just like the /etc/host.conf file which determines the way host lookups are performed. For example, the line

```
    hosts: files nis dns
```

specifies that host lookup functions should first look in the local /etc/hosts file, followed by a NIS lookup and finally through the domain name service (/etc/resolv.conf and named), at which point if no match is found an error is returned. This file must be readable for every user! You can find more information in the man−page nsswitch.5 or nsswitch.conf.5.

A good /etc/nsswitch.conf file for NIS is:

```
#
# /etc/nsswitch.conf
#
# An example Name Service Switch config file. This file should be
# sorted with the most-used services at the beginning.
#
# The entry '[NOTFOUND=return]' means that the search for an
# entry should stop if the search in the previous entry turned
# up nothing. Note that if the search failed due to some other reason
# (like no NIS server responding) then the search continues with the
# next entry.
#
# Legal entries are:
#
#       nisplus                 Use NIS+ (NIS version 3)
#       nis                     Use NIS (NIS version 2), also called YP
#       dns                     Use DNS (Domain Name Service)
#       files                   Use the local files
#       db                      Use the /var/db databases
#       [NOTFOUND=return]       Stop searching if not found so far
#

passwd:     compat
group:      compat
# For libc5, you must use shadow: files nis
shadow:     compat

passwd_compat: nis
group_compat: nis
shadow_compat: nis

hosts:      nis files dns

services:   nis [NOTFOUND=return] files
networks:   nis [NOTFOUND=return] files
protocols:  nis [NOTFOUND=return] files
rpc:        nis [NOTFOUND=return] files
ethers:     nis [NOTFOUND=return] files
```

```
netmasks:   nis [NOTFOUND=return] files
netgroup:   nis
bootparams: nis [NOTFOUND=return] files
publickey:  nis [NOTFOUND=return] files
automount:  files
aliases:    nis [NOTFOUND=return] files
```

passwd_compat, group_compat and shadow_compat are only supported by glibc 2.x. If there are no shadow rules in /etc/nsswitch.conf, glibc will use the passwd rule for lookups. There are some more lookup module for glibc like hesoid. For more information, read the glibc documentation.

7.6. Shadow Passwords with NIS

Shadow passwords over NIS are always a bad idea. You loose the security, which shadow gives you, and it is supported by only some few Linux C Libraries. A good way to avoid shadow passwords over NIS is, to put only the local system users in /etc/shadow. Remove the NIS user entries from the shadow database, and put the password back in passwd. So you can use shadow for the root login, and normal passwd for NIS user. This has the advantage that it will work with every NIS client.

7.6.1. Linux

The only Linux libc which supports shadow passwords over NIS, is the GNU C Library 2.x. Linux libc5 has no support for it. Linux libc5 compiled with NYS enabled has some code for it. But this code is badly broken in some cases and doesn't work with all correct shadow entries.

7.6.2. Solaris

Solaris does not support shadow passwords over NIS.

7.6.3. PAM

PAM does not support Shadow passwords over NIS, especially pam_pwdb/libpwdb. This is a big problem for RedHat 5.x users. If you have glibc and PAM, you need to change the /etc/pam.d/* entries. Replace all pam_pwdb rules through pam_unix_* modules. Due a bug in the pam_unix_auth.so module this will not always work.

An example /etc/pam.d/login file looks like:

```
#%PAM-1.0
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_unix.so
auth       required     /lib/security/pam_nologin.so
```

```
account    required     /lib/security/pam_unix.so
password   required     /lib/security/pam_unix.so
session    required     /lib/security/pam_unix.so
```

8. What do you need to set up NIS+ ?

8.1. The Software

The Linux NIS+ client code was developed for the GNU C library 2. There is also a port for Linux libc5, since most commercial Applications are linked against this library, and you cannot recompile them for using glibc. There are problems with libc5 and NIS+: static programs cannot be linked with it, and programs compiled with this library will not work with other libc5 versions.

You need to retrieve and compile the GNU C Library 2.1 for Intel based platforms, or GNU C Library 2.1.1 for 64bit platforms. As base System you need a glibc based Distribution like Debian, RedHat or SuSE Linux.

For every distribution, you need to recompile the gcc/g++ compiler, libstdc++ and ncures. For Redhat, you need to make a lot of changes of the PAM configuration. For SuSE Linux 6.0, you need to recompile the shadow package.

The NIS+ client software can be obtained from:

```
  Site             Directory                       File Name

  ftp.funet.fi     /pub/gnu/funet                  libc-*, glibc-crypt-*,
                                                   glibc-linuxthreads-*
  ftp.kernel.org   /pub/linux/utils/net/NIS+       nis-utils-1.3.tar.gz
```

You should also have a look at http://www.suse.de/~kukuk/nisplus/ for more information and the latest sources.

8.2. Setting up a NIS+ client

IMPORTANT: For setting up a NIS+ client read your Solaris NIS+ docs what to do on the server side! This document only describes what to do on the client side!

After installing the new libc and nis−tools, create the credentials for the new client on the NIS+ server. Make sure portmap is running. Then check if your Linux PC has the same time as the NIS+ Server. For secure RPC, you have only a small window from about 3 minutes, in which the credentials are valid. A good idea is to run xntpd on every host. After this, run

```
domainname nisplus.domain.
nisinit -c -H <NIS+ server>
```

to initialize the cold start file. Read the nisinit man page for more options. Make sure that the domainname will always be set after a reboot. If you don't know what the NIS+ domain name is on your network, ask your system/network administrator.

Now you should change your /etc/nsswitch.conf file. Make sure that the only service after publickey is nisplus ("publickey: nisplus"), and nothing else!

Then start keyserv and make sure, that it will always be started as first daemon after portmap at boot time. Run

```
keylogin -r
```

to store the root secretkey on your system. (I hope you have added the publickey for the new host on the NIS+ Server?).

"niscat passwd.org_dir" should now show you all entries in the passwd database.

8.3. NIS+, keylogin, login and PAM

When the user logs in, he need to set his secretkey to keyserv. This is done by calling "keylogin". The login from the shadow package will do this for the user, if it was compiled against glibc 2.1. For a PAM aware login, you have to change the /etc/pam.d/login file to use pam_unix2, not pwdb, which doesn't support NIS+. An example:

```
#%PAM-1.0
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_unix2.so       set_secrpc
auth       required     /lib/security/pam_nologin.so
account    required     /lib/security/pam_unix2.so
password   required     /lib/security/pam_unix2.so
session    required     /lib/security/pam_unix2.so
```

8.4. The nsswitch.conf File

The Network Services switch file /etc/nsswitch.conf determines the order of lookups performed when a certain piece of information is requested, just like the /etc/host.conf file which determines the way host lookups are performed. For example, the line

```
    hosts: files nisplus dns
```

specifies that host lookup functions should first look in the local /etc/hosts file, followed by a NIS+ lookup and finally through the domain name service (/etc/resolv.conf and named), at which point if no match is found an error is returned.

A good /etc/nsswitch.conf file for NIS+ is:

```
#
# /etc/nsswitch.conf
#
# An example Name Service Switch config file. This file should be
# sorted with the most-used services at the beginning.
#
# The entry '[NOTFOUND=return]' means that the search for an
# entry should stop if the search in the previous entry turned
# up nothing. Note that if the search failed due to some other reason
# (like no NIS server responding) then the search continues with the
# next entry.
#
# Legal entries are:
#
#       nisplus                 Use NIS+ (NIS version 3)
#       nis                     Use NIS (NIS version 2), also called YP
#       dns                     Use DNS (Domain Name Service)
#       files                   Use the local files
#       db                      Use the /var/db databases
#       [NOTFOUND=return]       Stop searching if not found so far
#

passwd:     compat
group:      compat
shadow:     compat

passwd_compat: nisplus
group_compat:  nisplus
shadow_compat: nisplus

hosts:      nisplus files dns

services:   nisplus [NOTFOUND=return] files
networks:   nisplus [NOTFOUND=return] files
protocols:  nisplus [NOTFOUND=return] files
rpc:        nisplus [NOTFOUND=return] files
ethers:     nisplus [NOTFOUND=return] files
netmasks:   nisplus [NOTFOUND=return] files
netgroup:   nisplus
bootparams: nisplus [NOTFOUND=return] files
publickey:  nisplus
automount:  files
aliases:    nisplus [NOTFOUND=return] files
```

9. Setting up a NIS Server

9.1. The Server Program ypserv

This document only describes how to set up the "ypserv" NIS server.

The NIS server software can be found on:

```
Site               Directory                     File Name

ftp.kernel.org     /pub/linux/utils/net/NIS      ypserv-1.3.11.tar.gz
```

You could also look at http://www.suse.de/~kukuk/nis/ for more information.

The server setup is the same for both traditional NIS and NYS.

Compile the software to generate the `ypserv` and `makedbm` programs. You can configure ypserv to use the securenets file or the tcp_wrappers. The tcp_wrapper is much more flexible, but a lot of people have big problems with it. And some configuration files for tcp_wrappers may cause a memory leak. If you have problems with ypserv compiled for tcp_wrapper, recompile it using the securenets file. ypserv --version tells you, which version you have.

If you run your server as master, determine what files you require to be available via NIS and then add or remove the appropriate entries to the "all" rule in `/var/yp/Makefile`. You always should look at the Makefile and edit the Options at the beginning of the file.

There was one big change between ypserv 1.1 and ypserv 1.2. Since version 1.2, the file handles are cached. This means you have to call makedbm always with the -c option if you create new maps. Make sure, you are using the new `/var/yp/Makefile` from ypserv 1.2 or later, or add the -c flag to makedbm in the Makefile. If you don't do that, ypserv will continue to use the old maps, and not the updated one.

Now edit `/var/yp/securenets` and `/etc/ypserv.conf`. For more information, read the ypserv(8) and ypserv.conf(5) manual pages.

Make sure the portmapper (portmap(8)) is running, and start the server `ypserv`. The command

```
    % rpcinfo -u localhost ypserv
```

should output something like

```
    program 100004 version 1 ready and waiting
    program 100004 version 2 ready and waiting
```

The "version 1" line could be missing, depending on the ypserv version and configuration you are using. It is only necessary if you have old SunOS 4.x clients.

Now generate the NIS (YP) database. On the master, run

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

On a slave make sure that `ypwhich -m` works. This means, that your slave must be configured as NIS client before you could run

```
    % /usr/lib/yp/ypinit -s masterhost
```

to install the host as NIS slave.

That's it, your server is up and running.

If you have bigger problems, you could start `ypserv` and `ypbind` in debug mode on different xterms. The debug output should show you what goes wrong.

If you need to update a map, run `make` in the `/var/yp` directory on the NIS master. This will update a map if the source file is newer, and push the files to the slave servers. Please don't use `ypinit` for updating a map.

You might want to edit root's crontab *on the slave* server and add the following lines:

```
      20 *    * * *    /usr/lib/yp/ypxfr_1perhour
      40 6    * * *    /usr/lib/yp/ypxfr_1perday
      55 6,18 * * *    /usr/lib/yp/ypxfr_2perday
```

This will ensure that most NIS maps are kept up−to−date, even if an update is missed because the slave was down at the time the update was done on the master.

You can add a slave at every time later. At first, make sure that the new slave server has permissions to contact the NIS master. Then run

```
    % /usr/lib/yp/ypinit -s masterhost
```

on the new slave. On the master server, add the new slave server name to `/var/yp/ypservers` and run `make` in `/var/yp` to update the map.

If you want to restrict access for users to your NIS server, you'll have to setup the NIS server as a client as well by running ypbind and adding the plus−entries to /etc/passwd _halfway_ the password file. The library functions will ignore all normal entries after the first NIS entry, and will get the rest of the info through NIS. This way the NIS access rules are maintained. An example:

```
root:x:0:0:root:/root:/bin/bash
daemon:*:1:1:daemon:/usr/sbin:
bin:*:2:2:bin:/bin:
sys:*:3:3:sys:/dev:
sync:*:4:100:sync:/bin:/bin/sync
games:*:5:100:games:/usr/games:
man:*:6:100:man:/var/catman:
lp:*:7:7:lp:/var/spool/lpd:
mail:*:8:8:mail:/var/spool/mail:
news:*:9:9:news:/var/spool/news:
uucp:*:10:50:uucp:/var/spool/uucp:
nobody:*:65534:65534:noone at all,,,,:/dev/null:
+miquels:::::
+:*:::::/etc/NoShell
[ All normal users AFTER this line! ]
tester:*:299:10:Just a test account:/tmp:
miquels:1234567890123:101:10:Miquel van Smoorenburg:/home/miquels:/bin/zsh
```

Thus the user "tester" will exist, but have a shell of /etc/NoShell. miquels will have normal access.

Alternatively, you could edit the `/var/yp/Makefile` file and set NIS to use another source password file. On large systems the NIS password and group files are usually stored in `/etc/yp/`. If you do this the normal tools to administrate the password file such as `passwd`, `chfn`, `adduser` will not work anymore and you need special homemade tools for this.

However, `yppasswd`, `ypchsh` and `ypchfn` will work of course.

9.2. The Server Program yps

To set up the "yps" NIS server please refer to the previous paragraph. The "yps" server setup is similar, _but_ not exactly the same so beware if you try to apply the "ypserv" instructions to "yps"! "yps" is not supported by any author, and contains some security leaks. You really shouldn't use it !

The "yps" NIS server software can be found on:

```
Site                  Directory                  File Name

ftp.lysator.liu.se    /pub/NYS/servers           yps−0.21.tar.gz
ftp.kernel.org        /pub/linux/utils/net/NIS   yps−0.21.tar.gz
```

9.3. The Program rpc.ypxfrd

rpc.ypxfrd is used for speed up the transfer of very large NIS maps from a NIS master to NIS slave servers. If a NIS slave server receives a message that there is a new map, it will start ypxfr for transfering the new map. ypxfr will read the contents of a map from the master server using the yp_all() function. This process can take several minutes when there are very large maps which have to store by the database library.

The rpc.ypxfrd server speeds up the transfer process by allowing NIS slave servers to simply copy the master server's map files rather than building their own from scratch. rpc.ypxfrd uses an RPC−based file transfer protocol, so that there is no need for building a new map.

rpc.ypxfrd can be started by inetd. But since it starts very slow, it should be started with ypserv. You need to start rpc.ypxfrd only on the NIS master server.

9.4. The Program rpc.yppasswdd

Whenever users change their passwords, the NIS password database and probably other NIS databases, which depend on the NIS password database, should be updated. The program "rpc.yppasswdd" is a server that handles password changes and makes sure that the NIS information will be updated accordingly. rpc.yppasswdd is now integrated in ypserv. You don't need the older, separate yppasswd−0.9.tar.gz or yppasswd−0.10.tar.gz, and you shouldn't use them any longer. The rpc.yppasswdd in ypserv 1.3.2 has full shadow support. yppasswd is now part of yp−tools−2.2.tar.gz.

You need to start rpc.yppasswdd only on the NIS master server. By default, users are not allowed to change their full name or the login shell. You can allow this with the −e chfn or −e chsh option.

If your passwd and shadow files are not in another directory then /etc, you need to add the −D option. For example, if you have put all source files in /etc/yp and wish to allow the user to change his shell, you need to start rpc.yppasswdd with the following parameters:

```
rpc.yppasswdd −D /etc/yp −e chsh
```

or

```
rpc.yppasswdd −s /etc/yp/shadow −p /etc/yp/passwd −e chsh
```

There is nothing more to do. You just need to make sure, that `rpc.yppasswdd` uses the same files as `/var/yp/Makefile`. Errors will be logged using syslog.

---

10. Verifying the NIS/NYS Installation

If everything is fine (as it should be), you should be able to verify your installation with a few simple commands. Assuming, for example, your passwd file is being supplied by NIS, the command

```
    % ypcat passwd
```

should give you the contents of your NIS passwd file. The command

```
    % ypmatch userid passwd
```

(where userid is the login name of an arbitrary user) should give you the user's entry in the NIS passwd file. The "ypcat" and "ypmatch" programs should be included with your distribution of traditional NIS or NYS.

If a user cannot log in, run the following program on the client:

```
#include <stdio.h>
#include <pwd.h>
#include <sys/types.h>

int
main(int argc, char *argv[])
{
  struct passwd *pwd;

  if(argc != 2)
    {
      fprintf(stderr,"Usage: getwpnam username\n");
      exit(1);
    }

  pwd=getpwnam(argv[1]);

  if(pwd != NULL)
    {
      printf("name.....: [%s]\n",pwd->pw_name);
      printf("password.: [%s]\n",pwd->pw_passwd);
      printf("user id..: [%d]\n", pwd->pw_uid);
      printf("group id.: [%d]\n",pwd->pw_gid);
      printf("gecos....: [%s]\n",pwd->pw_gecos);
      printf("directory: [%s]\n",pwd->pw_dir);
      printf("shell....: [%s]\n",pwd->pw_shell);
    }
  else
    fprintf(stderr,"User \"%s\" not found!\n",argv[1]);

  exit(0);
}
```

Running this program with the username as parameter will print all the information the getpwnam function gives back for this user. This should show you which entry is incorrect. The most common problem is, that the password field is overwritten with a "*".

GNU C Library 2.1 (glibc 2.1) comes with a tool called getent. Use this program instead the above on such a system. You could try:

```
   getent passwd
```

or

```
   getent passwd login
```

11. Surviving a Reboot

Once you have NIS correctly configured on the server and client, you do need to be sure that the configuration will survive a reboot.

There are two separate issues to check: the existence of an init script and the correct storage of the NIS domain name.

11.1. NIS Init Script

In your version of Linux, you need to check your directory of init scripts, typically `/etc/init.d`, `/etc/rc.d/init.d` or `/sbin/init.d` to be sure there is a startup script there for NIS. Usually this file is called `ypbind` or `ypclient`.

11.2. NIS Domain Name

Perhaps the greatest issue that some people have with NIS is ensuring that the NIS domain name is available after a reboot. According to Solaris 2.x, the NIS domain name should be entered as a single line in:

```
   /etc/defaultdomain
```

However, most Linux distributions does not seem to use this file.

11.3. Distribution−specific Issues

At this time, the following information is known about how various Linux distributions handle the storage of the NIS domainname.

11.3.1. Caldera 2.*x*

Caldera uses the file `/etc/nis.conf` which has the same format as the normal `/etc/yp.conf`.

11.3.2. Debian

Debian appears to follow Sun's usage of `/etc/defaultdomain`.

11.3.3. Red Hat 6.*x*

Create or modify the variable **NISDOMAIN** in the file `/etc/sysconfig/network`.

11.3.4. SuSE Linux

Modify the variable **YP_DOMAINNAME** in `/etc/rc.config` and then run the command **SuSEconfig**.

12. Common Problems and Troubleshooting NIS

Here are some common problems reported by various users:

1. The libraries for 4.5.19 are broken. NIS won't work with it.
2. If you upgrade the libraries from 4.5.19 to 4.5.24 then the su command breaks. You need to get the su command from the slackware 1.2.0 distribution. Incidentally that's where you can get the updated libraries.
3. When a NIS server goes down and comes up again ypbind starts complaining with messages like:
   ```
   yp_match: clnt_call:
   RPC: Unable to receive; errno = Connection refused
   ```
   and logins are refused for those who are registered in the NIS database. Try to login as root and kill ypbind and start it up again. An update to ypbind 3.3 or higher should also help.
4. After upgrading the libc to a version greater then 5.4.20, the YP tools will not work any longer. You need yp−tools 1.2 or later for libc >= 5.4.21 and glibc 2.x. For earlier libc version you need yp−clients 2.2. yp−tools 2.x should work for all libraries.
5. In libc 5.4.21 − 5.4.35 yp_maplist is broken, you need 5.4.36 or later, or some YP programs like ypwhich will segfault.
6. libc 5 with traditional NIS doesn't support shadow passwords over NIS. You need libc5 + NYS or glibc 2.x.
7. ypcat shadow doesn't show the shadow map. This is correct, the name of the shadow map is shadow.byname, not shadow.
8. Solaris doesn't use always privileged ports. So don't use password mangling if you have a Solaris client.

---

13. Frequently Asked Questions

Most of your questions should be answered by now. If there are still questions unanswered you might want to post a message to

```
comp.os.linux.networking
```