Managing Users: Creating an Account

- append a line in `/etc/passwd`, use new UID
- if a new group ID is used, append a line in `/etc/group`
- (Linux/Solaris) append a line in `/etc/shadow`, password field = ,,*"
- create the home directory of the user
- change owner and group of the home directory
- change protection bits of the home directory
- set the first password of the user with the `passwd` command

Managing Users: Disabling/Removing an Account

- set the corresponding password field in `/etc/shadow` to ,,*"
- change protection bits of the home directory to `---------`
- do a backup of the home directory
- recursively delete the contents of the home directory
- remove entry from `/etc/passwd`

Managing Users: useradd/userdel

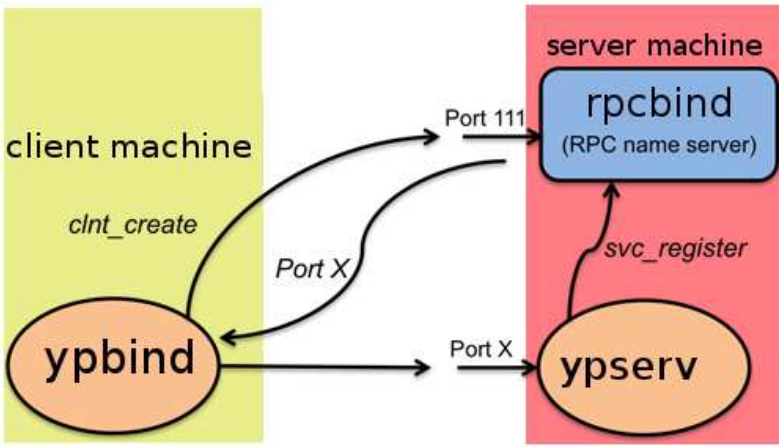
tools (not standardized)

`adduser`/`useradd` and `rmuser`/`deluser`/`userdel` commands

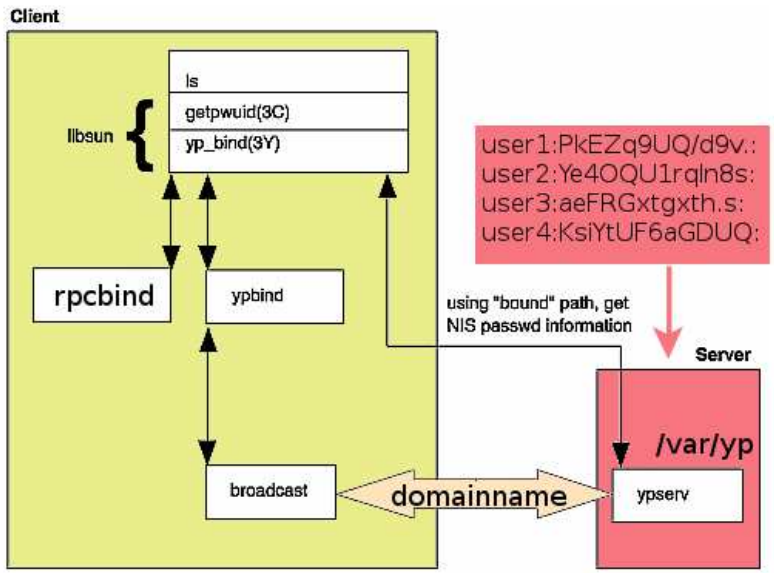
The steps above are especially useful

- if tools like `adduser` are missing
- for shell scripts creating many accounts

Network–wide Usermanagement: NIS (uses RPC)



NIS (binding client to server)



Network–wide Usermanagement: NIS (1)

NIS = network information service

invented by Sun as an RPC application ≈ 1988

need `portmap` (FreeBSD: `rpcbind`) service

consists of

- server: distributes user account information `ypserv`
- client: asks for correct authentication `ypbind`

common identity string: the *YP–Domainname* (see domainname(1))

`ypinit` sets up a NIS server from `/etc/master.passwd`

Network–wide Usermanagement: NIS (2)

server: start `ypserv`

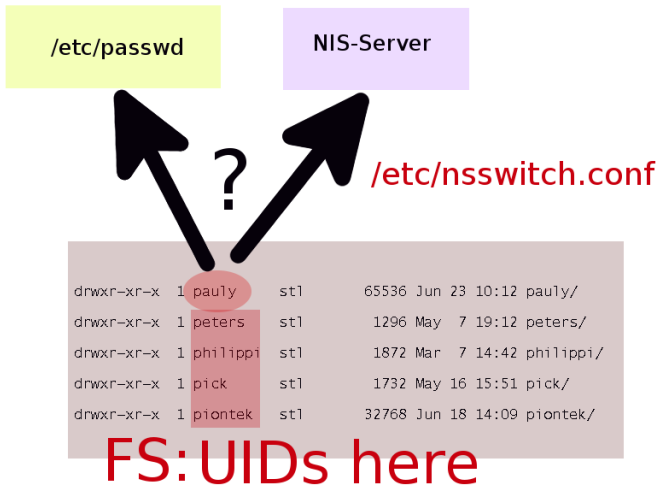
NIS *maps* under `/var/yp`

control access through

- `/var/yp/securenets` (FreeBSD/Linux)
- `/var/yp/ypserv.acl` (OpenBSD)

update `/etc/master.passwd` ⇝`make` in `/var/yp`

Problem: where do the usernames come from?



Network–wide Usermanagement: NIS (3)

client: start `ypbind`, domain name is command-line arg

two ways to refer to NIS-entries:

- `/etc/nsswitch.conf` include `nis` keyword
- `/etc/master.passwd` include `+:*::::::::` entry

`passwd` command ⇝local password file ⇝NIS server

same goes for `group`, `hosts`, `services`, ...

`root` account **locally** (for network problems, server shutdown etc.)

Network–wide Usermanagement: NIS (4)

commands

`ypwhich` prints the NIS server name

`ypmatch username passwd` prints the passwd entry of username

`ypcat passwd` prints the passwd map

more centralisation : group, services, hosts, ...

Network–wide Usermanagement: LDAP overview

LDAP=lightweight directory access protocol

concept used with *Active Directory* within Windows

openldap: user management / AD emulation / integration

- server side `slapd`
  - AD = special case of LDAP data
  - tedious configuration work
  - maybe SSL configuration
- client side
  - PAM
  - `nss_ldap`
  - `ldap.conf`

Network–wide Usermanagement: LDAP client

- configure LDAP server to be contacted (ldap port 389, ldaps 636)

  `/usr/local/etc/ldap.conf`
  `/usr/local/etc/openldap/ldap.conf`
  `-> host stl-s-proj2.htw-saarland.de stl-s-proj1.htw-saarland.de`

- simple LDAP query

  `ldapsearch -x -b "ou=organizational_unit"`

- install package `nss_ldap`
  (enables `ldap` keyword in `/etc/nsswitch.conf`)

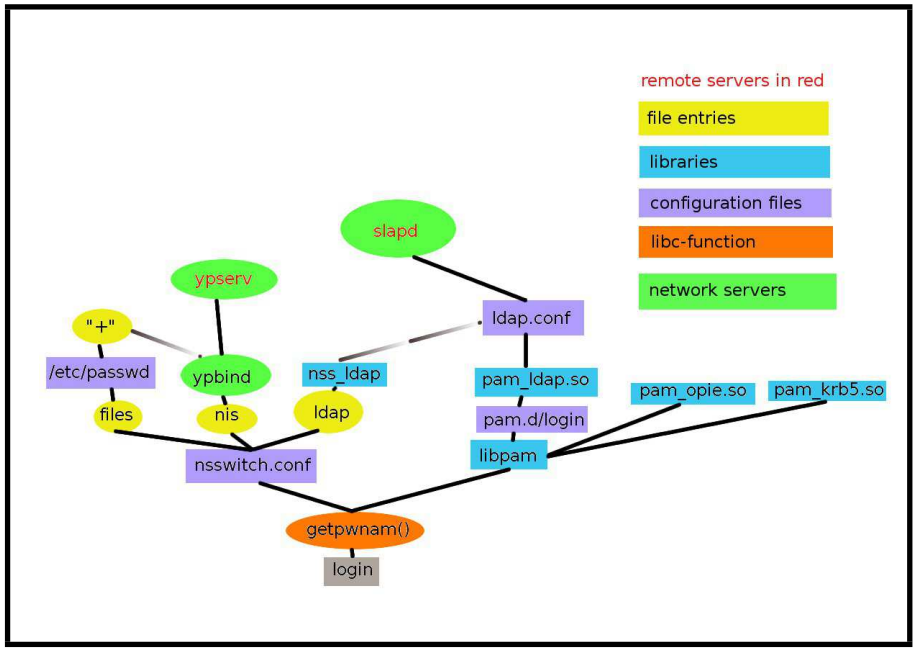
- install package `pam_ldap` (enables `ldap` keyword in `/etc/pam.d` files)

PAM: Mixing Authentication Methods

- different auth for different users

- different auth for different services

- extensible mechanism for new auth methods



Pluggable Authentication Module (1)

variety of authentication methods

- smartcards
- Kerberos
- one–time–passwords (OPIE)
- . . . (what next?)

configurable *modules* needed $\rightsquigarrow$**PAM**

Pluggable Authentication Module (2)

directory

`/etc/pam.d`

config files with sections

auth authentication functions

account account management functions

session session handling functions

password password management functions

entries (example):

```
auth            sufficient      pam_opie.so
```


Managing Users: More Commands

password-related commands for users and admins

- `vipw` (root)
- `chpass` change password entries (root)
- `chsh` change shell (root/user)
- `chfn` change real name (root/user)
- `passwd` change password (root/user)
- `pw` swiss army knife to change password entries (FreeBSD)

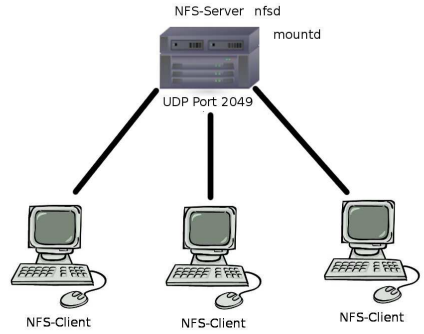
Network File System: NFS (1)

distribute file system

example: `/home`

implementation: RPC

handbook section 27.3



Network File System: NFS (2)

server host

- needs servers
  - `mountd` handles mount requests (`exports` file)
  - `nfsd` handles data requests at port 2049/udp
  - `portmap` or `rpcbind` to handle RPC
- needs configuration
  - services above must be started at boot time
  - which filesystems are exported to other hosts `/etc/exports`

example entry

```
/home -maproot=bin: 134.96.216.81
```

NIS/NFS security

no security issues since several years

summary of configuration issues

- NIS: separate NIS `passwd` map from local `/etc/passwd`
- NIS: control client access to NIS server
- NFS: no exports to the world
- NFS: map root to a non–root account
- NFS: firewalling the NFS–port

more details

`http://www.securityfocus.com/infocus/1387`

Alternatives

OpenAFS (Andrew File System) `http://www.openafs.org/`

influenced NFSv4

CIFS / SMB `http://www.samba.org/`

Special Feature: amd

Automount Daemon

can mount the network device, whenever a file is accessed

for example, if the user logs in

⇝no permanent connection to NFS server needed

Limiting Users

- don't interfere with needs of other users
- don't interfere with system processes

Limiting Users: Per-Process Limits (1)

```
$ ulimit -a
core file size          (blocks, -c) unlimited
data seg size           (kbytes, -d) 524288
file size               (blocks, -f) unlimited
max locked memory       (kbytes, -l) unlimited
max memory size         (kbytes, -m) unlimited
open files                      (-n) 3117
pipe size            (512 bytes, -p) 1
stack size              (kbytes, -s) 65536
cpu time               (seconds, -t) unlimited
max user processes              (-u) 1558
virtual memory          (kbytes, -v) unlimited
```


Limiting Users: Per-Process Limits (2)

there are three limits:

- kernel limit (=absolute system limit), often in kernel header file
- hard limit (may only be lowered by user), set by
  - system admin in global login script `/etc/profile`, or
  - sysctl kernel variable, or
  - system–specific files (FreeBSD: `/etc/login.conf`)
  - user via `ulimit`
- soft limit (may be lowered/raised by user), $\leq$ hard limit (use `ulimit -S`)

Limiting Users: Disk Quotas

- cannot be enforced on process level
- is a filesystem property
- must be enabled in kernel
- must be set when mounting a filesystem (see below)
- command `quota -v` lists disk usage
- command `edquota -u user` sets user limit

Note: quotas slow down writing to disk

6. File System

Drives and Capacity

as of 2014

Drive	Bandwidth (read)	Capacity	EUR/GB
hard disk drive	1.6 GB/s	60 GB...4 TB	0.06...0.20
solid state drive	2.7 GB/s	120 GB...2 TB	0.70...0.85
secure digital memory card	150 MB/s	4 GB...128 GB	0.68...0.85
USB memory stick	60 to 90 MB/s	4 GB...256 GB	0.69...2.00
digital versatile disk	61.7 MB/s (16x)	4.7 GB (1s, 1l)	0.69...2.00

http://en.wikipedia.org/wiki/Hard_disk_drive
http://www.intel.com/content/www/us/en/
solid-state-drives/solid-state-drives-ssd.html
http://www.tomshardware.com/charts/-usb-3.0-card-reader-charts-2014/
-01-Compact-Flash-Sequential-Read-MB-s,3542.html
http://www.tomshardware.com/reviews/DVD-Burner,2447-8.html