Back to UNIX-Usermanagement: Concept of Groups

each user belongs to *exactly one* principal group (⇝`/etc/passwd`)

the group ID and name defined in `/etc/group`

users may belong to additional groups

```
$ id theobald
uid=55177(theobald) gid=1111(stl)
groups=1111(stl), 1113(stlnagios),60001(cuda)
```

corresponding entries in `/etc/group`

```
cuda:*:60001:dweber,bohr,theobald
```


Managing Users: Remarks

- password file protection: file locking, command `vipw`
- different users *should have* different UIDs.
- network wide identities with NIS, NIS+, SMB, LDAP ...

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

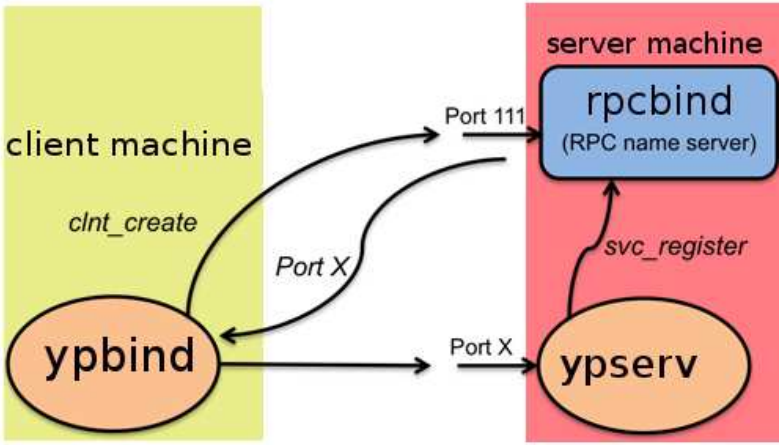
tools (not standardized)

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

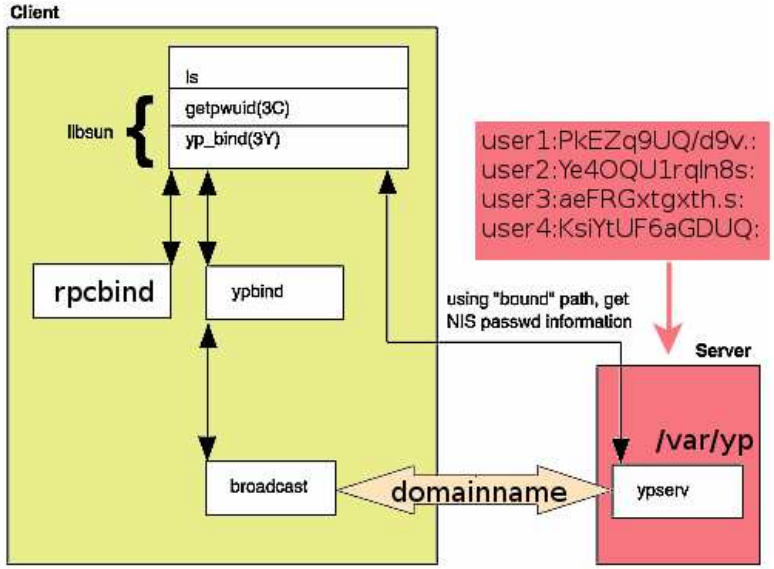
The steps above are especially useful

- 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`

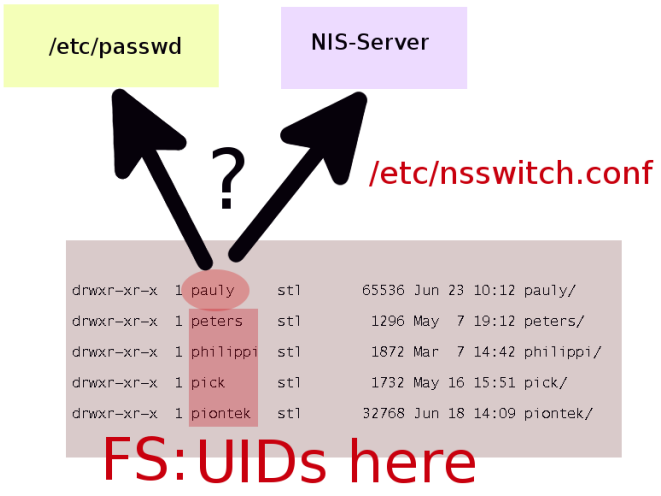
NIS *maps* under `/var/yp`

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

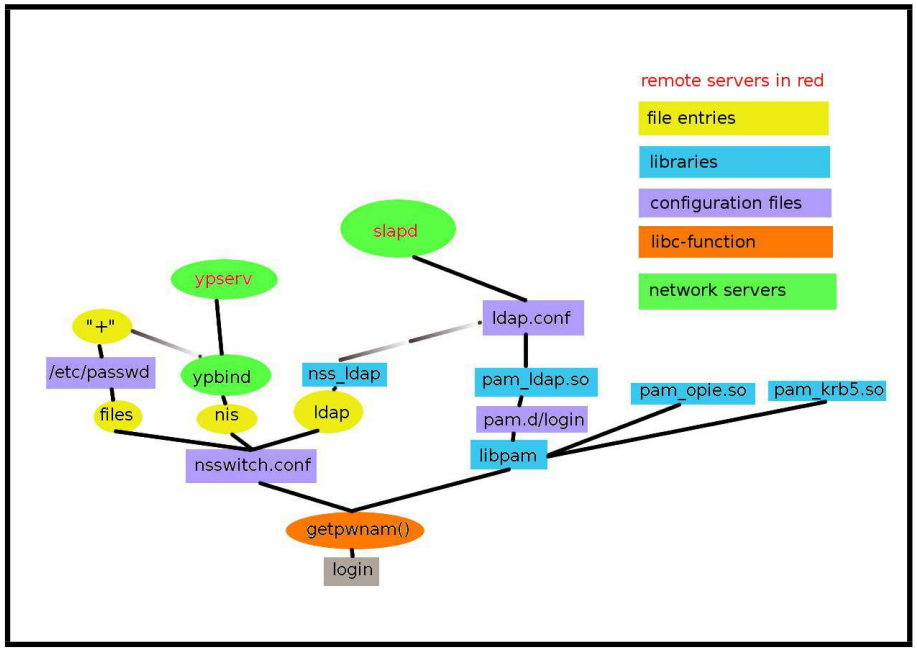
- configure LDAP server to be contacted (ldap port 389, ldaps 636)

  `/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)