

# User & Group Administration

David Morgan

© David Morgan 2004

## Users

- useradd/userdel
- /home/<user>
- /etc/passwd is the user database
- /etc/shadow has passwords (relocated from passwd)
- /etc/group
- whoami
- su

© David Morgan 2004

# Users

- system usage demands a user identification
  - supplied at login... no login, no usage
- system keeps a list of user accounts
  - may or may not correspond to human users
- user id is implicit in all session activities
  - helps determine access to resources
- users can be grouped

© David Morgan 2004

# The files of record

- /etc/passwd – holds list of recognized users
- /etc/shadow – holds their passwords
- /etc/group – holds list of recognized groups,  
names of member users for each

© David Morgan 2004

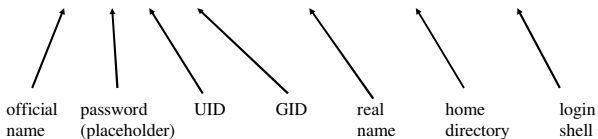
## User information

- official name (“login id” or “username”)
- password
- official number (or UID)
- group number for a “primary group” (GID)
- real name (and other real-world info)
- home directory
- program to run when user logs in (login shell)

© David Morgan 2004

## /etc/passwd entries hold user information

craig:x:507:507:Craig Smith:/home/craig:/bin/bash



© David Morgan 2004

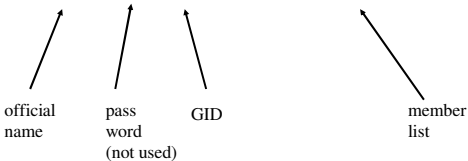
## Group information

- group name
- password
- official number (or GID)
- member list

© David Morgan 2004

/etc/group entries hold  
group information

children:x:522:hansel, pinochio,gretel,heidi



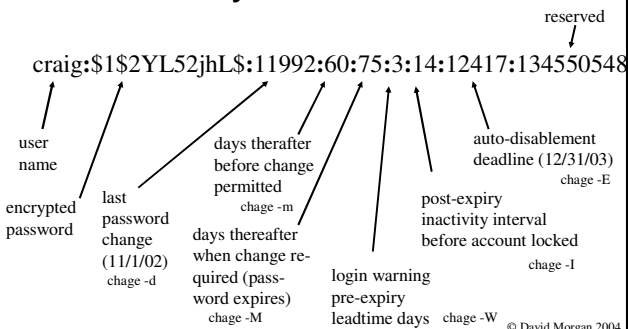
© David Morgan 2004

## Ancillary user information

- password aging items
  - time since last password change
  - number of days before password can be changed
  - number of days after which password must be changed
  - days before password expiry to give warning at login
  - days after password expiry to expire account
  - deadline at which to auto-disable account

© David Morgan 2004

## /etc/shadow entries hold ancillary user information



© David Morgan 2004

## Use chage to view...

```
[root@EMACH1 /root]# chage -l craig
Minimum:      60
Maximum:      75
Warning:       3
Inactive:     14
Last Change:  Nov 01, 2002
Password Expires:  Jan 15, 2003 ← last change + maximum
Password Inactive: Jan 29, 2003 ← ... + inactive
Account Expires:  Dec 31, 2003
```

© David Morgan 2004

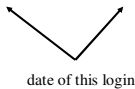
## ...or to modify

<u>Item modified</u>	<u>chage option used</u>
Minimum	-m
Maximum	-M
Warning	-W
Inactive	-I
Last Change	-d
Account Expires	-E

© David Morgan 2004

## Login during warning period

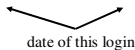
```
EMACH1 login: craig
Password:
Warning: your password will expire in 3 days
Last login: Sat Jan 11 16:03:31 on tty2
[craig@EMACH1 craig]$ date
Sat Jan 11 16:04:37 PST 2003
```



© David Morgan 2004

## Login after password expiry

```
EMACH1 login: craig
Password:
Your password has expired; please change it!
Changing password for craig
(current) UNIX password:
New UNIX password:
Retype new UNIX password:
Last login: Sat Jan 11 16:04:34 on tty2
[craig@EMACH1 craig]$
[craig@EMACH1 craig]$ date
Thu Jan 16 16:00:34 PST 2003
```



© David Morgan 2004

## New values thereafter

```
[root@EMACH1 /root]# chage -l craig
Minimum:    60
Maximum:    75
Warning:    3
Inactive:   14
Last Change: Jan 17, 2003 ← new change date reflected
Password Expires: Apr 02, 2003 → deadlines advanced accordingly
Password Inactive: Apr 16, 2003 →
Account Expires: Dec 31, 2003
```

© David Morgan 2004

## Editing the files of record safely

- plain editors invite error introduction and multiuser conflicts
- `/etc/passwd` – use `usermod` or `vipw`
- `/etc/shadow` – use `passwd`, `chage`, `usermod`
- `/etc/group` – use `groupmod` and `usermod`, or `vigr`

© David Morgan 2004

## Adding users – actions involved

- record added to `/etc/passwd`
- record added to `/etc/shadow`
- record added to `/etc/group`
- create user home directory `/home/<username>`
- copy default startup files to home directory
- set permissions on new files and directories
- set password
- customize user info with, e.g., `usermod` or `chage`

© David Morgan 2004

## Ways to add users

- do everything by hand
- let account management utilities to most of it
  - `useradd`
  - `passwd`

© David Morgan 2004

## Adding users in 2 steps

- use useradd
- then set password with passwd

© David Morgan 2004

## Adding users in 2 steps

```
[root@EMACH1 /root]# useradd charlie ← step 1
[root@EMACH1 /root]# passwd charlie ← step 2
Changing password for user charlie
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
Now find out what happened!
↓
[root@EMACH1 /root]# su charlie ← become charlie
[charlie@EMACH1 /root]$ cd ← enter his home directory
[charlie@EMACH1 charlie]$ pwd ← identify home directory
/home/charlie
[charlie@EMACH1 charlie]$ ls -a ← directory is populated
. .Xdefaults .bash_profile .kde .screenrc
.. .bash_logout .bashrc .kderc Desktop
[charlie@EMACH1 charlie]$ cat /etc/passwd | grep charlie
charlie:x:531:539::/home/charlie:/bin/bash ← charlie's in the list alright
```

© David Morgan 2004

## Ways to remove users

- do everything by hand
- let account management utilities to most of it
  - userdel -r

© David Morgan 2004

## Deleting users

```
[root@EMACH1 /root]# userdel -r charlie
[root@EMACH1 /root]# su charlie
su: user charlie does not exist ← doesn't live here anymore
[root@EMACH1 /root]# ls -a /home/charlie
ls: /home/charlie: No such file or directory ← home directory who??
[root@EMACH1 /root]# cat /etc/passwd | grep charlie
[root@EMACH1 /root]# ← gone. really!
```

© David Morgan 2004

## Disabling login without removing user

- replace shell
- substitute a “do nothing” program instead of /bin/bash
- /bin/false does nothing, returns immediately

```
usermod -s /bin/false <username>
```

© David Morgan 2004

## Disabling a user's login ability

```
[root@EMACH1 /root]# su charlie ← login as charlie works, gets a prompt
[charlie@EMACH1 /root]$ exit
exit
[root@EMACH1 /root]# usermod -s /bin/false charlie ← /bin/false returns,
                                                    does nothing
[root@EMACH1 /root]# su charlie ← login as charlie “works,” but reverts
[root@EMACH1 /root]# cat /etc/passwd | grep charlie ← right back to root's prompt
charlie:x:531:539::/home/charlie:/bin/false
[root@EMACH1 /root]# usermod -s /bin/bash charlie
[root@EMACH1 /root]# cat /etc/passwd | grep charlie
charlie:x:531:539::/home/charlie:/bin/bash
[root@EMACH1 /root]# su charlie ← bash shell is back, login as charlie
[charlie@EMACH1 /root]$ ← gets a user prompt again
```

© David Morgan 2004

## Adding users in batch mode

Set up a source file listing users in the form username:password

e.g., file “userinfo”

```
able:apple  
baker:banana  
charlie:cantelope
```

© David Morgan 2004

## Assigning passwords in batch mode with chpasswd command

man chpasswd:

“chpasswd reads a file of user name and password pairs from standard input and uses this information to update a group of existing users. ...

*[but] The named user must exist.”*

Solution: make the named users exist first, with a script that “useradds” them by looping through the list, then feed the list to chpasswd

© David Morgan 2004

## Adding users in batch mode

```
#!/bin/bash
i=1
while read LINE
do
    user=`echo $LINE | cut -f 1 -d :`
    useradd $user
    let i=$i+1
done < userinfo

cat userinfo | chpasswd
```

© David Morgan 2004

## Groups

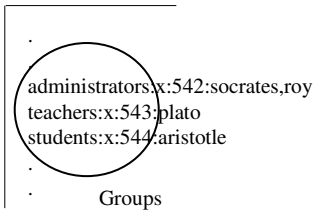
- Purpose
  - Let a set of users share files by extending common permissions to them
- Mechanism
  - Files have a group affiliation
  - Users have group memberships
  - Separate access to a file can be extended to members of its group

© David Morgan 2004

# There are groups

Groups are defined in /etc/group

file /etc/group



© David Morgan 2004

# Adding users to a group

- Use usermod

usermod -G employees,salesmen willie

- usermod does not specify a group's users
- usermod specifies a user's groups
- group's users per cumulative usermods

usermod -G employees,salesmen(willie)

usermod -G salesmen(joe)



willie and joe  
become members of salesmen,  
there are probably others

© David Morgan 2004

## Files have a group affiliation

Files' group affiliations are shown by the `ls -l` command:

```
[root@EMACH1 schools]# ls -l
total 12
-rw-r--r-- 1 root  students 121 Dec  8 17:15 assignments
-rw-rw---- 1 root  teachers 119 Dec  8 17:13 grades
-rw-r----- 1 root  administ 95 Dec  8 17:10 salaries
```

Files

Their groups

© David Morgan 2004

## Users have group memberships

Users' memberships appear in the file that defines the groups, (`/etc/group`) not the one that defines the users (`/etc/passwd`)

file `/etc/group`

The group

administrators:x:542:socrates,roy

teachers:x:543:plato

students:x:544:aristotle

The members

© David Morgan 2004

## Permissions for groups

**-rwxr-x---**

- **File type** (file, directory, device,...)
- Accesses granted to **file's associated User**
- Accesses granted to members of **file's Group**
- Accesses granted to all **Other users**

© David Morgan 2004

## Who can read what?

```
[root@EMACH1 schools]# ls -l
total 12
-rw-r--r-- 1 root  students  121 Dec  8 17:15 assignments
-rw-rw---- 1 root  teachers  119 Dec  8 17:13 grades
-rw-r----- 1 root  administ  95 Dec  8 17:10 salaries
```

socrates (an administrator) can read:  
salaries (because he's an administrator)  
assignments (because anybody can)

plato (a teacher) can read:  
grades (because he's a teacher)  
assignments (because anybody can)

aristotle (a student) can read:  
assignments (because he's student)

© David Morgan 2004

# Users have group memberships

Users' group memberships appear in the file that defines the groups, (/etc/group) not the one that defines the users (/etc/passwd)...

Plus one additional, default membership...

```
craig:x:507:507:Craig Smith:/home/craig:/bin/bash
```

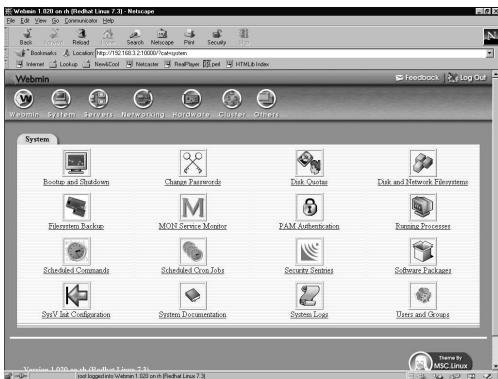


craig's "default group," by group number

... in /etc/passwd, the so-called default group. User holds membership in this one, plus those found in /etc/group.

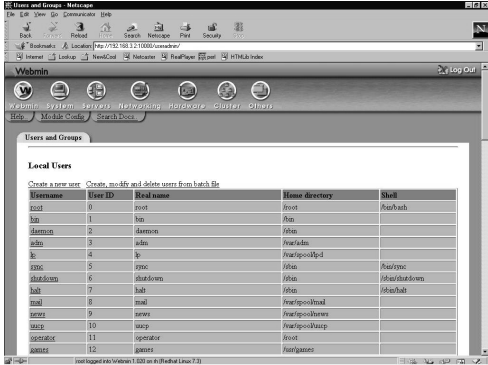
© David Morgan 2004

# Webmin



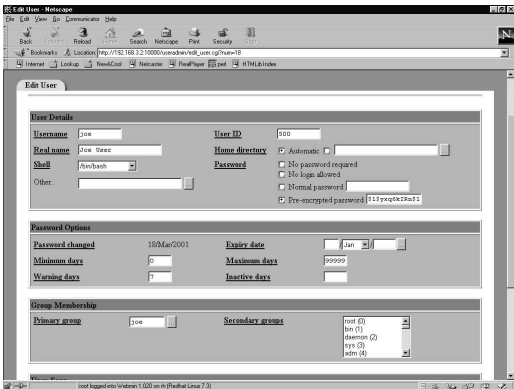
© David Morgan 2004

# Webmin



© David Morgan 2004

# Webmin



© David Morgan 2004