User & Group Administration

David Morgan

Users

- useradd/userdel
- /home/<user>
- /etc/passwd is the user database
- /etc/shadow has passwords (relocated from passwd)
- /etc/group
- whoami
- su / sudo / SUID – process UID exception
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
- every process runs “as” a certain user
  - whose ID is embedded in the process
  - command shell assumes the ID given at login
  - processes you launch then inherit it from your shell
- user id thus implicit in all session activities
  - helps determine access to resources
- users can be grouped

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
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)

/etc/passwd entries hold user information

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

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

/etc/group entries hold group information

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

offical name  password (not used)  GID  member list
/etc/shadow entries hold ancillary user information

craig:$1$2YL52jhL$11992:60:75:3:14:12417:134550548

- user name
- encrypted password
- last password change (11/1/02)
- days thereafter before change permitted
- days thereafter when change required (password expires)
- login warning pre-expiry leadtime days
- post-expiry inactivity interval before account locked
- auto-disablement deadline (12/31/03)
- reserved

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 2004-2015
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

Ways to add users

- do everything by hand
- let account management utilities do most of it
  - useradd
  - passwd
Adding users in 2 steps

- use useradd
- then set password with passwd

[root@EMACH1 /root]# useradd charlie
[root@EMACH1 /root]# passwd charlie
Changing password for user charlie
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
[root@EMACH1 /root]# su charlie
[charlie@EMACH1 /root]$ cd
[charlie@EMACH1 charlie]$ pwd
/home/charlie
[charlie@EMACH1 charlie]$ ls -a
.
.. .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

Now find out what happened!

- step 1
- step 2
- become charlie
- enter his home directory
- identify home directory
- directory is populated
- charlie’s in the list alright
Ways to remove users

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

Deleting users

```bash
[root@EMACH1 /root]# userdel -r charlie
[root@EMACH1 /root]# su charlie
su: user charlie does not exist
[root@EMACH1 /root]# ls -a /home/charlie
ls: /home/charlie: No such file or directory
[root@EMACH1 /root]# cat /etc/passwd | grep charlie
[root@EMACH1 /root]#
```

doesn’t live here anymore

home directory who??
gone. really!
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>

Disabling a user’s login ability

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

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
Adding users in batch mode

```bash
#!/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
```

Security drawback of chpasswd

- uses a file of cleartext passwords
- when finished destroy or remove that file
Groups

- **purpose**
  - let a set of users share identical access to a file by extending common permissions to them

- **mechanism**
  - files have a group affiliation
  - users have group memberships
  - access to a file can be extended to members of its group separate from that extended to others

There are groups

Groups are defined in /etc/group

file /etc/group

```
administrators:x:542:socrates,roy
teachers:x:543:plato
students:x:544:aristotle
```
Composing a group
– by assigning group(s) to a user

- 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.

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

Their groups
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)

The group

administrators:x:542:socrates,roy
 teachers:x:543:plato
 students:x:544:aristotle

The members

Permissions for groups

- rWXRr-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
### 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 a student)

### 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…

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

- Craig Smith’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.
Windows* has comparable features

*ntfs filesystem

file-specific permissions per user or group

user accounts

groups

© David Morgan 2004-2015