Users and Groups

Administration

- Why have users and groups?
- What is the difference between authentication and authorization?

User accounts

- Consist of:
  - USername
  - UID
  - Primary GID
  - GECOS
  - HOME
  - Shell
  - Encrypted Password (/etc/shadow)
- Other items can be seen in `/etc/passwd`

Group Accounts

- Consist of:
  - Group name
  - GID
  - List of members
- `cat /etc/group`

Primary group vs other Group membership

- WHen a file/dir is created, who owns it? What are its' permissions?
  - UID comes from process that created it
  - GID comes from process that created it
  - Permissions come from umask of process that created it.

Review

- Changing UID/GID and permissions
- How to change defaults associated with an account
- How are permissions applied?

Review

- Files we should make sure we know:
  - `/etc/shadow`
  - `/etc/passwd`
  - `/etc/group`
  - `/etc/skel`

Command review

Command	Command
passwd	chfn
chsh	umask
id	groups
who	finger

Command review

Command	~
adduser	addgroup
chmod	chgrp
chown	~

Permissions review (Files)

- read = r = 4 (view)
- write = w = 2 (modify)
- execute = x = 1 (run)
- Examples?

Permissions review (Directories)

- Same as for files except
  - r = view list of directory contents
  - w = modify list of directory
  - add, delete, rename files in dir
  - x = enter directory, or cd into dir

Permissions review

Mode layers:

- User (UID) - mode applied to processes with matching UID
- Group (GID) - mode applied to processes with matching GID
- Other (world)

Each mode can be set for r,w,x

Permissions

If a process wants to read, it will be allowed if the UID matches and user-read is set, or if GID matches and group-read is set, or if other read is set. Same for write and execute.

Special Permissions

In addition to other bits:

- 4 = setuid bit = s or S
- 2 = setGID bit = s or S
- 1 = sticky bit = t

SetUID

- Files
  - When the program is run, the process will have the UID of the file, instead of inheriting from parent process. Same for GID
- Directories:
  - No effect
- `ls -l /usr/bin/passwd`

SetGID

- Files
  - Same as SetUID described previously
- Directories
  - New contents of directory will have GID of directory instead of inheriting from creating process.

Sticky Bit

- Files
  - no effect
- Directories
  - only root and file owners can delete and rename files in the directory

---

Full file mode examples

- 2750?
- 6755?
- 1777?
- 0644?
- r-sr-sr-x?
- rwsr-Sr–?
- rwxrwxrwt?

Tidbit

Why the difference between `s` and `S` when setting SetUID or SetGID bits? If the file is NOT executable, you have `S`, otherwise `s`.

Umask

Each process has a mask that defines permission bits that should be removed from newly created files and directories.

- 0002
- 0027

Umask inherited from parent process. Processes can change their umask with a system call. Shell usually gets default umask from a shell init file.

KAHOOT

kahoot