username: mjackson
name: Michael Jackson
default shell: /bin/bash
home directory: /home/mjackson

sudo goes in front of command if you're trying to alter anything in the system
for example: passwd -d jsmith ✗
sudo passwd -d jsmith ✓

to delete a user including their root, their home directory
* sudo Tc. userdel mjackson

NEVER use nano /etc/passwd command OR /etc/shadow command

Your goal to privledge escalation is to get to the ROOT (Bigdee, Corl)

03/18 class
Principle of least privledge will be on this week's test!

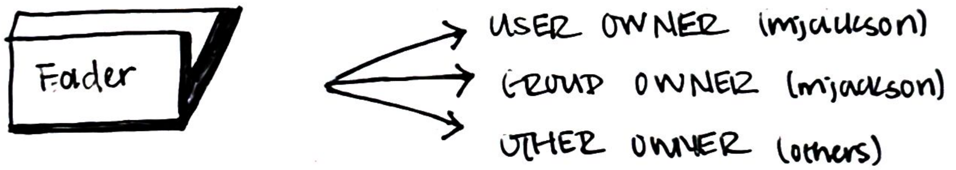
Linux Authorization:
1). Ownership
Ex: logged in as mjackson ✓ → /home/mjackson
whoami
pwd
touch song.txt (create a song file)
↑ mjackson is the OWNER of this file, NOT the admin
mkdir songs.txt
↑ mjackson is also the OWNER of this

Ownership DEFINES who has control of a file or directory
* might be on the test *

* if jjackson wants access, mjackson would have to grant it to her

Three Types of Ownership ★
1) User Ex: mjackson
2). Group mjackson
3). Others everyone else

When you create a file, linux will AUTOMATICALLY creates a user owner, group owner, and other owner.



USER OWNER (mjackson)
GROUP OWNER (mjackson)
OTHER OWNER (others)

2). Permission
DEFINES what can be done to a file or directory
1) Read (r)
2). Write (w)
3) Execute (x)

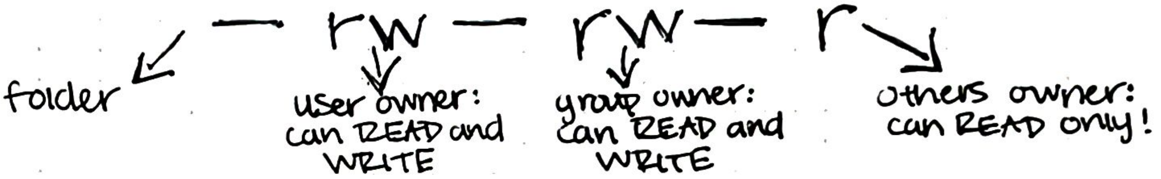
Every ownership folder has 3 types of permission

USER	GROUP	OTHERS
Read	Read	Read
Write	Write	Write
Execute	Execute	Execute

Command to check ownership and setting
1). whoami
2). groups
3). ls -l

COLOR OF FILES
blue —
white- text file
green — executable file

d at the beginning of line under ls- al is directory , -, is for file


- rw- rw- r
folder
user owner: can READ and WRITE
group owner: can READ and WRITE
others owner: can READ only!

File2: user 5
group 4
other 0
Permission: 540

File 3: user 4
group 4
other 4
Permission: 444

user	rwx	rw	r	wx	w	x	rx
group	rwx	rw	r	wx	w	x	rx
others	rwx	rw	r	wx	w	x	rx
	7	6	4	3	2	1	5

Command that changes permission on a file or folder (chown + chmod)

CHMOD

$$\underbrace{\text{Chmod}}_{\text{command}} \overset{\text{space}}{\wedge} \underbrace{u+x,\ g+x,\ o+x}_{\text{no space between}} \overset{\text{space}}{\wedge} \underbrace{\text{MyFile}}_{\text{name of file}}$$

Default permission for files: 666 / 664 (after umask)
directories: 777 / 775 (after umask)

umask by default is 002

rw-rw-rw ⟶ rw-rw-r
due to U-MASK

so rw-rw-rw
is $\overset{6}{\text{rw}} - \overset{6}{\text{rw}} - \overset{6}{\text{r}}$ subtracts 002

U-Mask changes default mask!

RWS or RWT → sticky bit, you cannot delete the "t", (temp folder)

→ allows only certain permission, user can run command, but limited access

↓ special execute command

command su IS on the test!

U-Mask: 0002
↑
SUI, GUID, Sticky Bit (stops you from deleting a folder)

* know the different between Passwd (user information) and shadow (pass)
* chown (changes <u>ownership</u>), chmod changes <u>permission</u>
* Principle of least privledge (Linux used this by accessing password, permission and ownership.