

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 ; smith` ✗
`sudo passwd -d ; smith` ✓

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

NEVER use `nano /etc/passwd` command

Your goal to privilege escalation is to get to the ROOT (Bigdee, Cori)

03/18 CLASS

Principle of least privilege will be on this week's test!

Linux Authorization:

1. Ownership

Ex: logged in as mjackson → /home/mjackson

✓

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

`whoami`
`pwd`

`touch song.txt` (create a song file)

↑
mjackson is the OWNER of this file,
NOT the admin

`mkdir songs.txt`

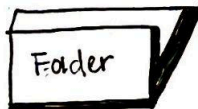
↑
mjackson is also the OWNER of this

* if mjackson 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
Read
Write
Execute

GROUP
Read
Write
Execute

OTHERS
Read
Write
Execute

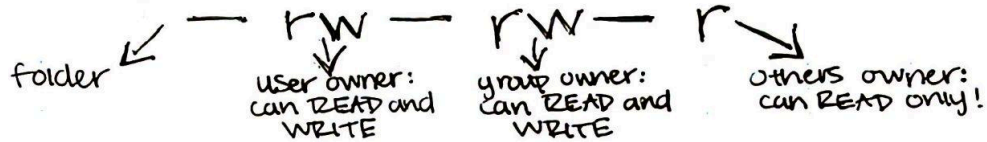
Command to check ownership and setting

- 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



File 2: 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

