Users, Groups, and Permissions in Linux

Agenda

- Creating and modifying users
- Creating and modifying groups
- File permissions

useradd Vs adduser

- adduser – will create a new user and request some information about the user
- useradd – will create a new user with defaults and NO PASSWORD

passwd

- The file /etc/passwd contains information about our users, but interestingly NOT their password
- passwd is a command that we can use to set or change the password of a user

usermod

- This is the PROPER way to modify a user.
- Technically you *can* (… and I sometimes do) edit /etc/passwd, /etc/shadow, /etc/group… with a text editor, but you really shouldn’t.

userdel

- If usermod modifies a user what do you thing userdel does?

https://www.marvel.com/characters/thanos/on-screen/profile

groups

- The groups command will list all the groups a user is a member of.
- If no user is specified it will assume the current user

groupadd Vs addgroup

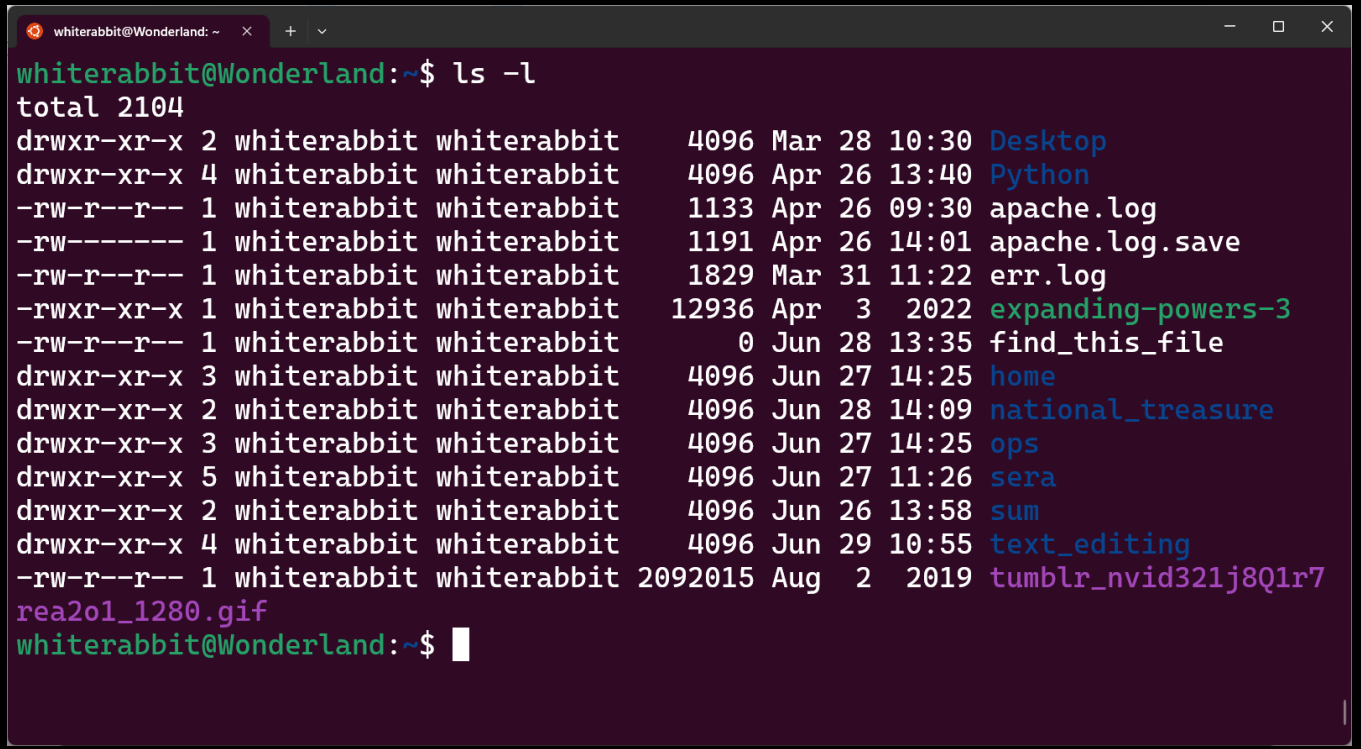
- Just like useradd and adduser addgroup will create a new group after requsting information and groupadd will simply create a new group.
- Fun fact: both adduser and addgroup really call useradd and groupadd in the end.

groupmod and groupdel

- Modify and delete groups just like their counterparts usermod and userdel modify and delete users.
- Again, you *can* edit the configuration files directly, but you really shouldn’t

chmod

- Change Mode – this is used to set file permissions
- r – read
- w – write
- x - execute



File Permissions Math

- -rwxrwxrwx
- Each permission is controlled by a single bit (1,0)
- Permissions are separated as Owner, Group, World
- 111 101 001 – what permissions are represented?
- 7 5 1

Using chmod

- We can specify the value for the permissions we want
  - chmod 777
    - Please try to avoid granting 777 permissions -_-
- We can add or remove permissions symbolically
  - chmod +rwx
    - Seriously avoid granting this much permission
- We should always keep in mind the Principle of least privilege

Using chmod

- User who owns it (u)
- File's group (g)
- Other users not in the file's group (o)
- all users (a)
- EXAMPLE chmod o-rw

setuid, setgid, and eww stickybits

- setuid – This will set the file to execute as the owner of the file
- setgid – This will set the file to execute as the group on the file
- sticky bit – If this permission is set it protects the file from deletion

But wait where are uid,gid, and the sticky bit stored?!?!

- Good catch! There is a secret 3rd set of bits before the standard permission where uid is 4 guid is 2 and the sticky bit is 1

I need information stat

- stat is a useful utility that shows us the current status of a given file.