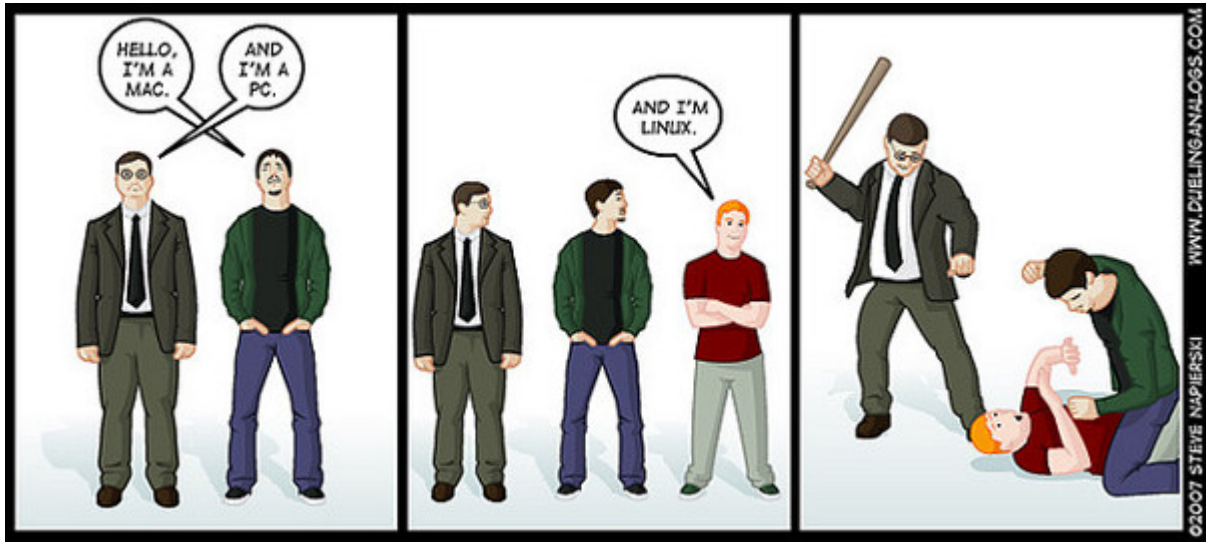
Lec-03-2
Users and Groups

Dr Faisal Hasan and Dr. Hymie Latif
Computing and Information Technology
College of Enterprise and Development
Otago Polytechnic
Dunedin, New Zealand

Bachelor of Information Technology
IN616 – Operating Systems Concepts
Semester 1, 2020

Schedule

- Searching Recap
- Users in Linux
  - Creating and managing users
- Groups in Linux
  - Creating and managing groups



Notices

- **Week 5: Skills Based Assessment**
  - In-class assessment
  - Covers everything we have done/will do in weeks 1,2,3,4
  - This is a hard assessment, start preparing!

- **Cisco NetAcademy:** netacad.com/login
  - Login using student email and password
  - **Online course: Linux Essentials**
  - Everyone should be enrolled (ask me if there are problems)

TOPIC:

RECAP: Searching

`grep` VS `find`

- `find`
  - Primarily for **file system entries**
  - Search for: file names, directory names
- `grep`
  - Primarily for **file content**
  - It can be used for anything (if using pipes)
  - Search for: file content
  - Search for: file names, directory names

`find`: Syntax

```
find <location> <options> <expression>
find /etc –maxdepth 1 –name “host”
```

- location:

  `/etc`
- options:

  `-maxdepth 1`
- expression:

  `-name “host”`

`find`: Examples

`find`

Find all files and directories (anything) in current working directory

`find ~ –name “*.jpg”`

Find anything ending in .jpg in home directory

`find ~/images -not –name “*.jpg”`

Find anything not ending in .jpg in images directory

`find /etc –name “login*” ! –name “*.html”`

Find anything starting with login and not ending with .html

`find –name “kittens*” –type f`

Find files starting with kitten

`find –type d –name “css”`

Find directory called css

`find /etc /var/log –type f –name “*.log”`

Find files ending with .log in /etc and /var/log

grep: Syntax

```
grep <options> <expression> <file/s>
grep –rin “andrew” ~/studentlist.txt
```

- expression:

  “andrew”

- options:

  -rin

- file:

  ~/studentlist.txt

`grep`: Examples

`grep “GNU” GPL-3`

Find GNU in file named GPL-3

`grep –i “GNU” GPL-3`

Find case-insensitive GNU (e.g., gnu, Gnu etc.)

`grep –in “GNU” GPL-3`

Same as above, but print line numbers

`grep –n “^GNU” GPL-3`

Find GNU only at start of the line

`grep –rn “and$” /etc`

Find and only at end of line reclusively in /etc directory

`grep “t[wo]” GPL-3`

Find “to” or “tw”

`grep “[0-9][0-9]” GPL-3`

Find two digits in a row

`grep` and piping

- `ls /etc`
  - List all files in `/etc`
- `grep “host*”`
  - Find the string “`host*`” followed by anything
  - But there is no file to search?!

```
ls /etc | grep “host*”
```

Command 1 **Command 2**

Pipe

Class-03-1: More Resources

- **Digital ocean tutorial : find**
  - https://www.digitalocean.com/community/tutorials/how-to-use-find-and-locate-to-search-for-files-on-a-linux-vps
- **Digital ocean tutorial : grep**
  - https://www.digitalocean.com/community/tutorials/using-grep-regular-expressions-to-search-for-text-patterns-in-linux
- **Digital ocean tutorial : regular expressions**
  - https://www.digitalocean.com/community/tutorials/an-introduction-to-regular-expressions

- **Make sure you finish Lab-03-1**
- **And understand the fundamentals**

TOPIC:

Basic Linux Commands

Basic File/Directory Commands

- **`mkdir`**
  - Creates a directory
- **`mv`**
  - Renames, or moves a file/directory
- **`cp`**
  - Copies a file/directory
- **`Files: rm`**
- **`Directories: rm -R (or rmdir` but has to be empty)**
  - Remove a directory (recursively – all subdirectories)

Basic Directory Commands: Examples

- **mkdir**

```
mkdir images
mkdir -p /home/user/images/vacation
```

- **mv**

```
mv script.sh /media/user/backup/script-bu.sh
mv -R ~/images/* /media/user/backup/images/
```

  - Needs –R for recursive

- **cp**

```
cp ~/images/1.jpg /media/user/backup/images/1.jpg
cp -R ~/images/* /media/user/backup/images/
```

  - Needs –R for recursive

Basic File Commands

- **`cat`**
  - Views a file
  - Or adds two files together (con**cat**enate)
- **`tail`**
  - Prints last 10 lines of a file
- **`head`**
  - Prints first 10 lines of a file
- **`more`**
  - Prints one page (terminal size) of file at a time
  - Press space to see *more*, press q to exit
- **`less`**
  - Better than more! Just has extended capability (backward and forward scrolling)
  - Faster because less does not read entire file before displaying

Basic File Commands: Examples

- **cat**

```
cat script.sh
cat script1.sh script2.sh
```

- **tail**

```
tail /var/log/auth.log
```

- **head**

```
head /var/log/auth.log
head myverylongscript.sh
```

- **more**

```
more myverylongscript.sh
```

- **less**

```
less /var/log/syslog
```

Writing Files

- **Creating files: with touch**

```
touch filename
touch myscript.sh
```

  - Creates file (if non-existent) or
  - Updates timestamp of existing file

- **Creating files: with redirection**

```
<command> > filename
<command> > output.txt
ls /etc > output.txt
```

  - Redirects output into a file
  - Prints filenames in /etc then saves filenames to output.txt

Modifying Files

- **Modifying files: with redirection**

```
<command> >> filename
<command> >> output.txt
ls -l /etc >> output.txt
```

  - Appends output into a file (in this case output.txt)
  - The file usually exists, but doesn't have to

- **Combining files: with concatenation**

```
cat filename1 filename2
cat streamA.txt streamB.txt
```

  - Combines files, two or more (prints to screen by default)

```
cat streamA.txt streamB.txt streamC.txt > linux.txt
```

  - Can also use redirection!

TOPIC:

Linux and Users

Linux and Users

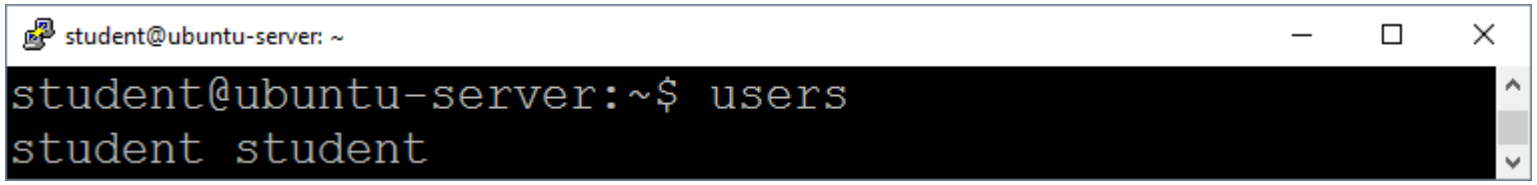
- Users have:
  - Login name
  - User identifier (UID)
  - Primary group membership (GID)
    - Users can only have one primary group
    - But can be members of one or more secondary groups
  - Full name
  - Home directory
  - Default shell

Logged in Users

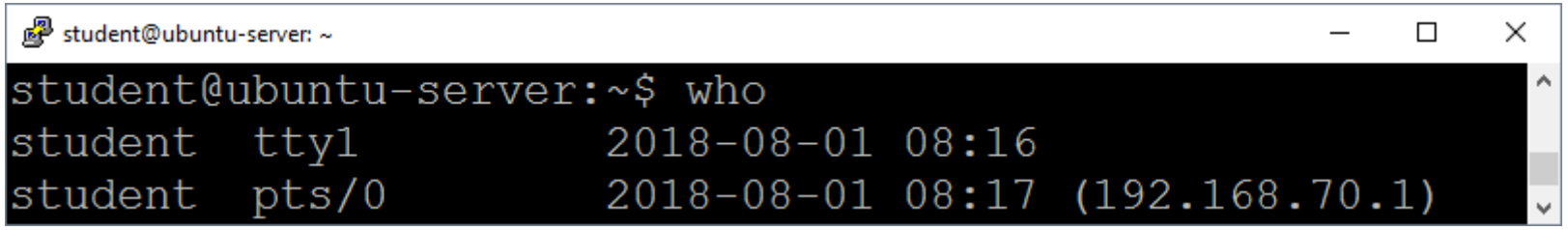
- **`users`**
  - Simply list any logged in users
- **`who`**
  - List logged in users, their terminal and login tine
- **`w`**
  - List logged in users, their terminal, login time, and system resource usage

Logged in Users: Examples

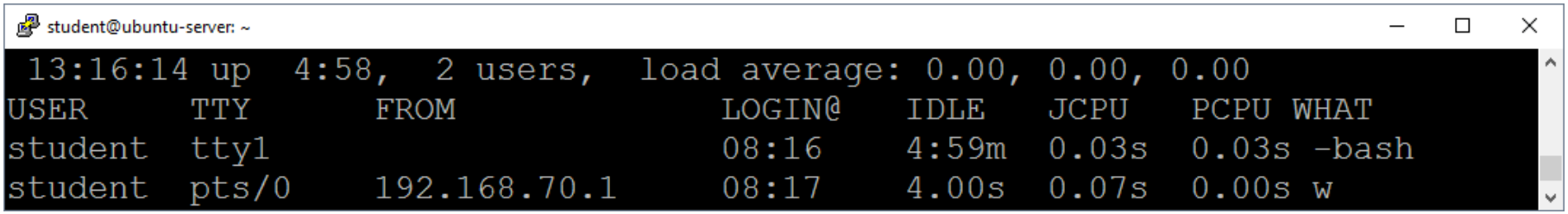
- **`users`**



- **`who`**



- **`w`**



All Users

- The **`users`**, **`who`** and **`w`** only list logged in users

- The password file contains **all** users
- Location: **`/etc/passwd`**

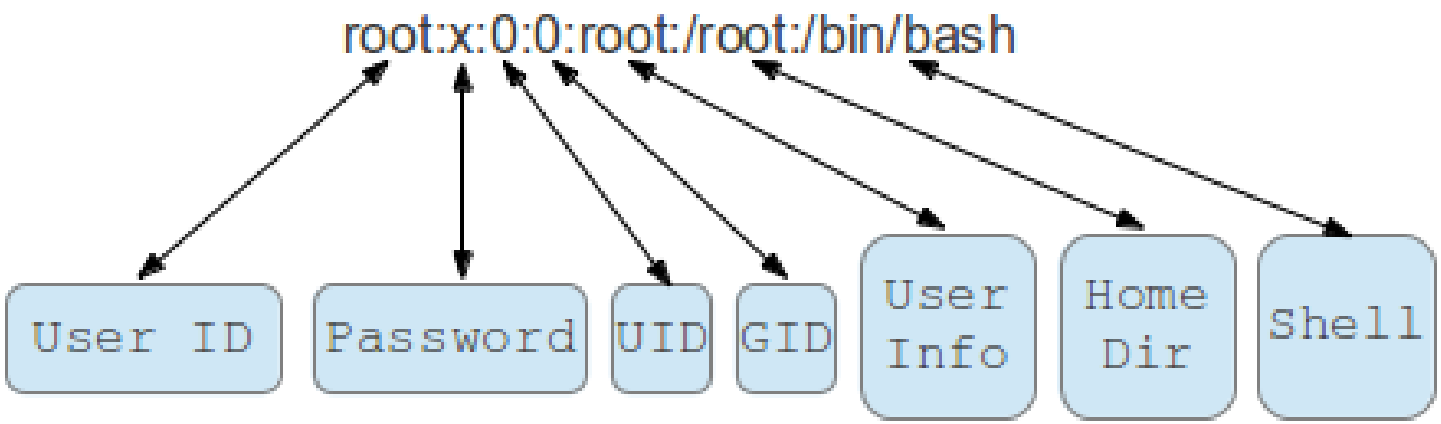
- This file contains configuration information
  - username
  - UID
  - GID
  - User info (properties such as full name)
  - Home directory
  - Default shell

The /etc/passwd file

```
user@ubuntu:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
messagebus:x:106:110::/var/run/dbus:/bin/false
uuidd:x:107:111::/run/uuidd:/bin/false
user:x:1000:1000:user,,,:/home/user:/bin/bash
user@ubuntu:~$ _
```

Linux Passwords

`cat /etc/passwd`



`student:x:1000:1000:student,,,:/home/student:/bin/bash`

`manager:x:1001:1001::/home/manager:/bin/bash`

All Users: Passwords

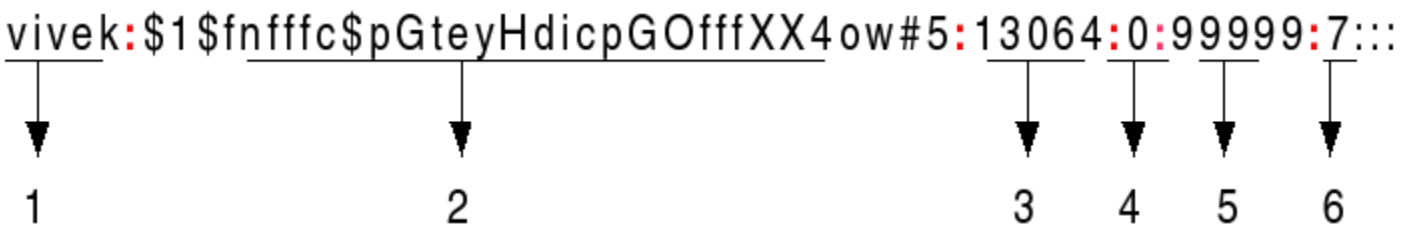
- Ironically, the password file contains no passwords

- Passwords are stored in the shadow file
- Location: **`/etc/shadow`**

- This file only shows users with a password
  - And has a different structure

Linux Passwords

`cat /etc/shadow`



1. Username
2. Hashed password
3. Last password change date
4. Minimum days for password change
5. Maximum days for password change
6. Warning in days for password change
7. Inactive until disabled (after password expires)
8. Account expired date

Managing Users

- **`useradd`**
  - Adds users
  - You must provide all user information (e.g., home directory)
- **`usermod`**
  - Manages configuration for an existing user
  - Great for managing a user's primary group membership
- **`userdel`**
  - Deletes a user
- **`passwd`**
  - Changes a user's password
- **`adduser (PLEASE DO NOT USE THIS ONE!)`**
  - Wizard tool for user creation
  - Not a default tool, not available on all distributions (← Dr Frantz says bad!)

Lab-03-2 – Start

- **<u>TOPICS:</u>**
- **Determine logged in users**
- **Review password file**
- **Create accounts for these two suspicious hobbits!**

TOPIC:

Linux and Creating Users

Creating Users

- We can add a new user with **useradd**

  useradd <options> <username>

- You can run command with no options (please don't!):

  useradd frodo

- Best practice: Specify configuration during creation

  useradd –d /home/frodo –m –s /bin/bash frodo

  useradd –d /home/samwise –m –s /bin/zsh samwise

  -d is the home dir

  -m is create the home dir

  -s is set the default shell

- Note: no password set using this method

Creating Users: Setting Passwords

- We must give a new user a password
- This can be achieved using the **`passwd`** command

```
passwd <options> <username>
```

- You can run the command with no username:

```
passwd
```

  - This will change the password for current user

- You can run the command with username specified:

```
passwd frodo
```

  - This will change the password for the specified user (frodo)

Creating Users: Setting Passwords

- Watch the prompts!

```
student@ubuntu-server: ~
student@ubuntu-server:~$ sudo passwd saruman
[sudo] password for student:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

- What is happening here?
  - sudo asks for the logged in user's password (student)
  - Then asks for new password for user (Saruman)

Creating Users: Secondary Groups

- Every user is associated with a primary group
  - username = **`frodo`**
  - primary group= **`frodo`**

- The primary group is created by default
  - Why: group permissions (we will talk about this next class)

- Our first command did not specify secondary groups
- The following commands use –G to add these

```
useradd –d /home/frodo –m –G sudo frodo
useradd –d /home/frodo –m –G sudo,fellowship frodo
```

TOPIC:

Linux and Groups

Linux and Groups

- Groups are used to organise users

- Groups:
  - a collection of user accounts
- Why:
  - Ease administration
  - Security measure

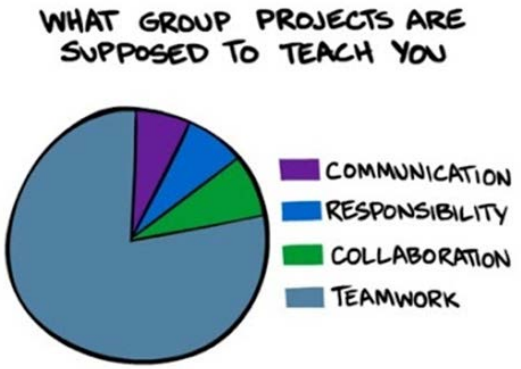
Linux and Groups

- Every user is associated with a **primary group**
  - username = **`frodo`**
  - primary group= **`frodo`**
- Users can also have **secondary groups**
  - Could have **no** secondary groups
  - Could have **one** secondary group
  - Could have **many** secondary groups
- How many is many?

  `65,536` **(2^{16})**

  Linux kernel > 2.6.3

  (unsigned short integer)



Managing Groups: Utilities

`groupadd`

- Adds a group

`groupmod`

- Modify a group name
- Modify a group ID
- This can be dangerous! Use with caution

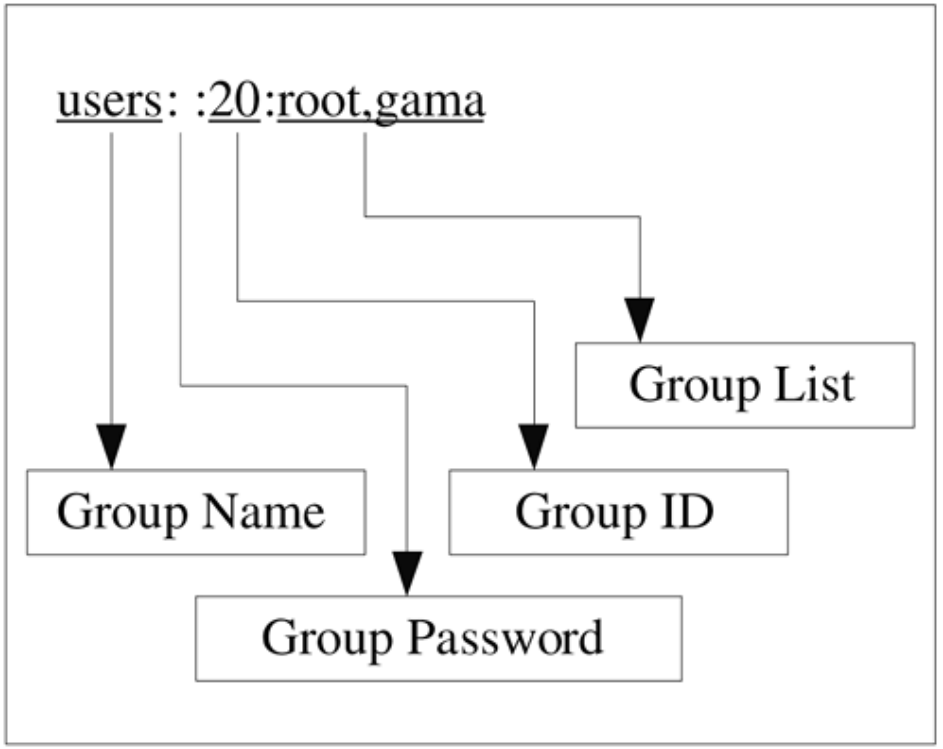
`groupdel`

- Deletes a group

The /etc/groups file

cat /etc/groups

```
user@opstudent-host:~$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,user
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:user
```



Primary VS Secondary Groups

- **Primary groups**
  - Primary groups **own** during file creation
  - By default, primary group is the **`<username>`** group
  - Why? **Exclusive ownership** of newly created files
  - A user will only ever have **one primary group**
  - View primary group in **`/etc/passwd`**

- **Secondary groups**
  - Secondary groups are used to **handle permissions**
  - Users can have **multiple secondary groups**
  - Members of a group seen in **`/etc/group`**

Add Users to Secondary Groups

- Not recommended to change primary group
  - Unless you have a reason and know what you are doing

- Add users to a **secondary group**

```
usermod -a -G fellowship frodo
usermod -a -G fellowship samwise
```

-a append user to another group

-G a list of groups

NOTE: Adding to a group needs a logout and login

```
usermod -G sudo,fellowship,theonering frodo
usermod -g fellowship frodo (Danger Zone!)
```

Primary VS Secondary: Groups

- **Overview of group membership; commands:**
  - **`id`** → Shows UIDs and GIDs for the logged in user and group membership
  - **`groups`** → Shows group names for the logged in user
  - **`members`** → Shows members for groups (not on default install)

- **Changing group associations/membership**
  - **`usermod -g groupname username`**
    - Changes primary group for user
  - **`usermod -G groupname username`**
    - Overwrite secondary group membership
    - Use **`-a`** to append to current group memberships

Lab-03-2 – Finish

- **TOPICS:**
- **Creating groups**
- **Managing groups**

Class-03-2 – Homework

- **Check/Use Cisco NetAcademy**
- **Make sure you are up-to-date with labs**
- **Start preparing for Skills Based Assessment in week 5**