Basic Commands for Linux!

Please Take Notes

Overview

- Get basic knowledge of Linux commands
- Adapted towards competition environment
- Options included are mostly useful for competition

More Things

- Anything highlighted in GREEN is something that you have to INSTALL
- General topics are highlighted in PURPLE
- There are many many many programs out there but we will only be covering basics
- There are some that you actually have to install, but they are closely related with the basic ones

Command line

- Where you execute commands in Linux
- Best way to do cyber things in Linux
- Need to know how it operates before moving on to anything else

What are commands?

- Words you enter into the command line that execute programs
- Can be low level programs (cd, ls) or high level (quake2, supertux)
- Commands are typed using lower case

Basics

man

- displays a guide/manual on how to use a command

- USAGE: `man [command]`
- EXAMPLE: “`man ls`” will display a guide on how to use the LIST (`ls`) command

clear

- clears the screen!
- USAGE: clear
- SHORTCUTS: CTRL+L
- Your screen will be very full after typing a few commands. If it gets to much, clear it off!

sudo

- allows you to execute commands as root **OR** logging in as root
- USAGE: `sudo [command]` or `sudo [user]`
- `sudo stands for SUPER USER DO`
- `You will never get an “access denied” error is you use “sudo” in front of a command or user - but BE CAREFUL!`

su

- substitutes user identity; allows you to log into any account
- USAGE: su [user]

- NOTES:
  - no argument will log in as root, but have to use `sudo su` in order to do this

exit

- exits the current shell session
- USAGE: `exit`

- NOTE: if you are logged in as root, you will have to `exit` twice to close the terminal

exit programs

- if you wanna exit a program, **most of the time** you can type **CTRL+C** and it will exit the operation. sometimes, however, you cannot do this and you will have to close the terminal.

Navigation/Basic File Operations

ls

- lists the contents of the current directory
- USAGE: `ls [options]`
- USEFUL OPTIONS:
  - -a = show hidden files (strongly recommend using!)
  - -l = long listing format
  - -R = recursively
  - -t = sort by modification time
  - -u = sort by last access time

cd

- changes directories
- USAGE: `cd [directory]`
- NOTES:
  - "`cd ../`" will back you up one directory level
  - keep repeating "`../`" to go back multiple directories; 3 times will go back 3 directories
  - "`cd`" alone will take you back to your home directory

tabs

- not a command, just something important
- when you are executing a command with an argument that is a file, a directory or some other object in a collection, type the argument, press tab, and will display all options starting with those characters.

- EXAMPLE: if you have `myfolder/`, can do "`cd myf[PRESSES TAB]`" will complete the path IF `myfolder/` is the only object in that directory starting with "`myf`"

copy and paste

- also just something a bit useful, if you want to copy something:
  - highlight it
  - press middle mouse button
- BUT: will immediately paste to where your text cursor is currently at as soon as you press middle mouse button

previous commands

- up and down arrow keys to go to forward and back in history
- Sounds weird - definitely try this several times to fully understand.

mv

- moves a file to different location/overwrite
- USAGE: `mv [files] [location]`

cp

- copies a file to another location
- USAGE: `cp [files] [location]`

rm

- removes a file
- USAGE: `rm [options] [file]`
- USEFUL OPTIONS:
  - -f = force
  - -r = recursively for files
- NOTES: this is a DANGEROUS command when in the hands of someone who doesn't understand what is going on.
- Try several times in practice

mkdir/rmdir

- creates/removes a directory
- USAGE:
  - `mkdir [folder]`
  - `rmdir [folder]`
- NOTE: if the directory is full, will not remove; in this case is better to use `rm -rf [directory]`

touch

- creates a blank, empty file
- USAGE: touch [files]

Useful System Commands/More File Operations

cat

- displays contents of a file
- USAGE: `cat [file]`

less

- displays contents but allows you to scroll through with arrow keys
- USAGE: `less [file]`
- NOTE: press ‘q’ to quit

grep

- searches for a string in a file(s)
- USAGE:
  - grep [str] [pattern] [file]
  - [command] | grep [str]
- NOTES: don't worry about patterns (regex); that comes later

piping

- taking the output of one command and applying it to another
- reduces amount of things u type in
- EXAMPLES:
  - less | grep “walnuts”
  - ps aux | grep “inetd”
  - ls -R | grep “Steve Aoki - No Beef.mp3”

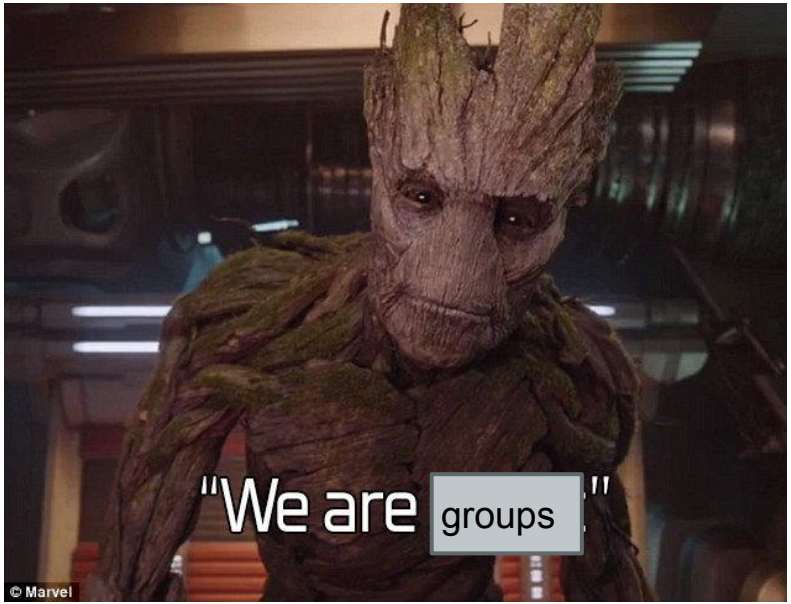
useradd/userdel and groupadd/groupdel

- to add or delete users
- USAGE:
  - `userdel [username]`
  - `useradd [username]`

- **groupadd/groupdel**: add or deletes groups, same usage as `userdel/useradd`

groups

- shows all of the groups a user is part of
- USAGE: `groups [user]`



passwd

- sets a password for an account
- USAGE:`passwd [options] [user]`
- USEFUL OPTIONS:
  - -d = deletes the password for the user
- NOTES: really only useful if you've just used userdel/useradd or are modifying root password

ifconfig

- displays network information like IP address, subnet mask, etc.
- USAGE: ifconfig [options]

ftp/sftp

- used to log into an FTP server
- USAGE: `ftp [host]`

sftp: uses SSH protocol, same usage but much more secure

tar

- used to create and unzip unix archives
- USAGE: `tar [options] [files]`
- USES:
  - unzip .tar.gz = `tar -xzvf très_shady.tar.gz`
  - unzip .tar.bz2 = `tar -xjvf très_shady.tar.bz2`

vi/vim/visudo

vi:

- text editor, really simple

vim:

- modified version of vi, better. have to download

visudo:

- like vi but **only** used to edit sudo file(s); don't have to but is safer thing to do **in general**

nano

- better than vi, a little more streamlined
- USAGE: `nano [file]`

pico: nano based off of this, nano has more options tho

emacs

- a bit complicated but much better than most other command line text editors, u can learn it if u want
- DOWNLOAD: http://www.gnu.org/software/emacs/

sysv-rc-conf (no more chkconfig)

- configure and schedule startup processes (edits `rc[0-6].d`)
- USAGE: `sysv-rc-conf [options] [command]`
- USEFUL OPTIONS:
  - `--list [service]` = list all services, can place service name after to search for that service
  - -o = organize:
    - a = sort alphabetically
    - p = sort by level priority

sysv-rc-conf continued

- runlevels - different levels of operation in system
  - 0 = halt (shutdown)
  - **1 = single user mode**
  - 2 = graphical multi-user with networking
  - 3-5 = unused
  - **6 = reboot**
- different for other operating systems

crontab

- schedule cron tasks
- USAGE: `crontab [options]`
- USEFUL OPTIONS:
  - -l = lists all jobs
  - -e = allows you to edit

ps/top

- prints process information
- USAGE: `ps [options]`
- EXAMPLES:
  - `ps aux` - prints SNAPSHOT of all processes

top:

- much better organization for processes, shows memory, CPU usage, etc. AND dynamically updates
- USAGE: `top`
- NOTE: has navigation controls u have to learn but pretty easy to learn

htop

- like top but with pretty colors
- just type `htop`

chmod

- changes permissions on a file
- USAGE: `chmod [code] [file]`
- EXAMPLE:chmod u=rwx,g=rx,o=r myfile
- The letters **u**, **g**, and **o** stand for "**user**", "**group**", and "**other**".
- The equals sign ("**=**") means "set the permissions exactly like this"
- The letters "**r**", "**w**", and "**x**" stand for "read", "write", and "execute", respectively
- “myfile” is the file name
- The commas separate the different classes of permissions
- There are no spaces in between them
- So, **u=rwx** means file owner can read/write/execute file
- **g=rx** means group members of the file owner can read/execute file
- **o=r** means all other users can read file

chmod

- You can also shorten your command using octal numbers
- USAGE: `chmod [code] [file]`
- EXAMPLE: chmod code of **751** is represented below:

	OWNER 4+2+1=7	GROUP 4+1= 5	OTHERS (programs, rd users, etc) 1=1
[r]EAD = 4	Yes	Yes	No
[w]RITE = 2	Yes	No	No
E[x]ECUTE = 1	Yes	Yes	Yes

More Useful Things

history

- shows command history
- USAGE: `history [options]`
- USEFUL OPTIONS:
  - -c = clears history
- NOTE: can just go to `.bash_history` in home directory

alias/unalias

- defines a string that, when entered in command line, will execute commands it is set to
- USAGE: `alias [name]='[stuff]'`

unalias: `unalias [alias]` will delete the alias

function/unset

- better than aliases, can easily put multiple commands
- USAGE: `function name() { [stuff]; [morestuff]; }`
- to delete, do: `unset [function name]`

killall

- kills a process by name
- USAGE: `killall [process name]`

ping:

- tests for network connection
- USAGE: `ping [url/host]`

export

- updates the value of an environment variable
- USAGE: `export [var]=[value]`
- EXAMPLES:
  - `export PATH=$PATH:/usr/local/bin`
- bin - for binaries
- if you have an executable in a directory listen in PATH, it will run like regular command

Package Management

Package Management: Overview

- online repositories, can download to your computer through command line
- Debian distros (ex: Ubuntu, GNU) - apt-get (using dpkg package manager)
- Redhat/Fedora/CentOS/Mint/Yellow Dog Linux - yum (using RPM package manager)
- NOTE: `.deb` != `.rpm`
- `dpkg` & `rpm` are **package managers**, `yum` and `apt-get` are **management tools**

dpkg

- packages in `.deb` files
- package manager - allows files to be interpreted, keeps track of all packages
- USAGE: `dpkg [options]`
- USEFUL OPTIONS:
  - -l = list all packages currently installed

apt-get

- used with dpkg, allows you to download specific package
- USAGE: `apt-get [command] [package]`
- USEFUL COMMANDS:
  - install - installs a package
  - update - gets packages for system updates
  - upgrade - actually updates
  - autoremove - removes, but also unneeded things

apt-get continued

- USEFUL OPTIONS:
  - --purge = use when executing autoremove, deletes configuration files for these packages
  - soooo many others, would take way too long to explain all of them

apt-cache

- gets information from repository; best use is for searching
- USAGE: `apt-cache [command] [package]`
- USEFUL COMMANDS:
  - search - searches for that packge
  - unmet - shows unmet dependencies
  - pkgnames - lists all packages installed on system, can use with `grep` to search

rpm

- Red Hat Package Manager, same concept as dpkg but uses `.rpm` files
- USAGE: `rpm [options] [package]`
- USEFUL OPTIONS:
  - -ivh = to install a downloaded `.rpm`
  - -q = checks if package is installed
  - -Uvh = upgrades an existing `rpm`
  - -e = erase a package

yum

- same concept as apt-get, just different syntax
- USAGE: yum [command] [package]
- USEFUL COMMANDS:
  - install - installs a package
  - update - updates system
  - remove/erase - removes a package
  - search - searches for a package
  - deplist - lists dependencies for a package

wget

- downloads a file from a url
- USAGE: `wget [url]`
- easier for most to go to website and download but
  - 1) if website is slow, wget bypasses loading of content and stuff
  - 2) some packages require wget to be used

Miscellaneous Commands

cal

- displays a calendar
- USAGE: `cal`
- NOTES: useful for when you forget what day it is

dc

- desktop calculator
- USAGE: dc
- NOTES:
  - uses weird syntax; # # [operator]
    - ex: + 4 2 would equal 6
  - type ‘p’ to calculate
  - tbh just use a regular calculator

eog

- opens up an image
- USAGE: `eog [file]`

Still confused?

- Don’t worry, you will someday. It just takes practice
- You are learning a NEW language.

Some learning resources

- http://tldp.org/LDP/abs/html/basic.html
- http://ss64.com/bash/
- http://askubuntu.com/
- http://help.ubuntu.com/