

Cyber 101: Linux and VMs



Linux

What will we be covering in this presentation?

- The basics of operating in and using Linux/the command line
 - File/directory navigation & management
 - Users, Groups, and Permissions
 - Working with a simple text editor
 - The Linux directory structure

So...what is Linux?

- An open source operating system based off of UNIX
- Developed by Linus Torvalds in 1991
- Several different distros of Linux exist...
 - Red Hat/Enterprise
 - Fedora
 - Ubuntu
 - Debian
 - CentOS



Uhhh excuse me I like Windows or Mac OS X. Why would I use Linux (or a command line interface)?

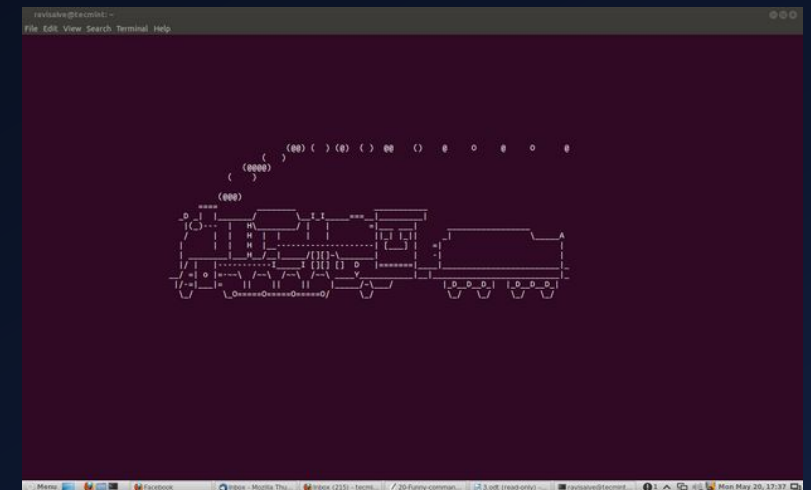
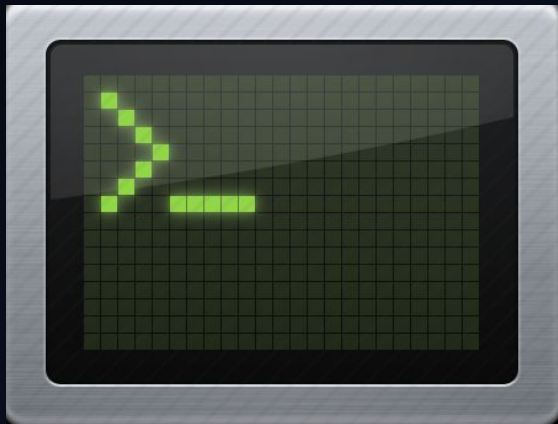
- Windows and Mac OS X have their merits for things like compatibility and familiarity, but...
 - Linux is FREE, so lots of businesses use it
 - It's an extremely efficient OS, and each distro gets a lot of revisions
 - In terms of specifications, it's very resource friendly!
- For command line interfaces, it's all about interfacing...
 - GUIs (Graphical User Interfaces) are nice-to-haves in the working world
 - It's also a nice skill to have!

Still not convinced on Linux???

- Linux will run on just about any computer
 - You can customize the installation to fit most hardware around today
- You can personally update Linux from the command line
- Linux is extremely customizable and is a great way for beginners to learn new skills and how to navigate a very widely used tool
 - Since it is very widely used in industry it is a great resume builder that companies will take note of
- Compilers for widely used languages (C, C++) are included and interpreters for Python, PHP, and more are included too

Okay so now I've got you hooked on Linux but now you're wondering what a command line interface is...

- A non-graphical user interface for a computer's operating system
- Basically, it lets you talk to your computer in a text-based environment.
- A lot of (if not all) OSs have something like this!
 - In Linux or Mac OS X, this is referred to as the 'terminal'
 - In Windows, this is called the 'Command Prompt'



Okay, so I can command prompt now...but now what?

- YOU CAN DO SO MANY AMAZING THINGS!!!
- What we'll be going over now are some useful commands that will prepare you for using your Linux environment!
- Don't be afraid to ask questions!

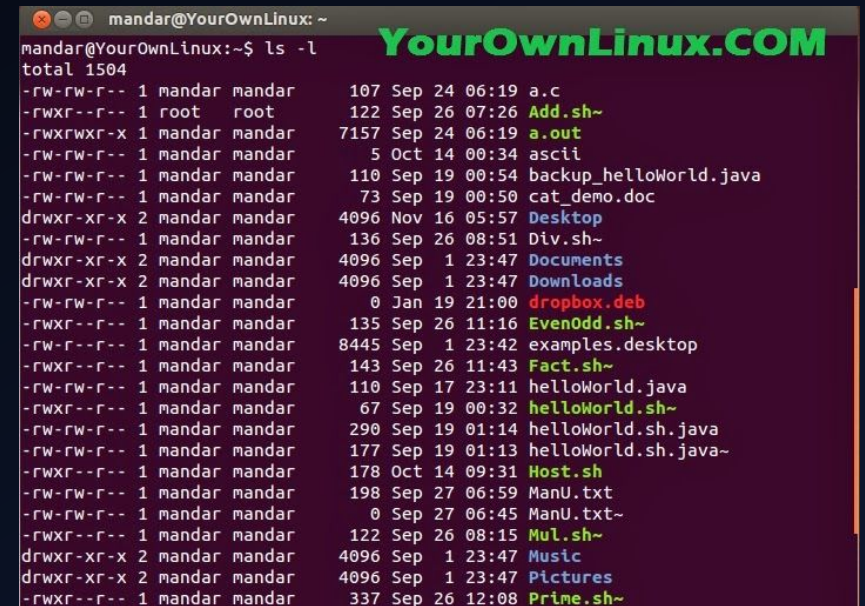


ONE OF THE MOST HELPFUL COMMANDS OF THEM ALL

- The 'man' command, this will be your go to command. Its short for manual
- Used to list and display command manual pages
 - Usage example 1 (UE1): 'man man' (yes, man has its own man page)
 - UE2: '**man ls**'
- EVERY COMMAND HAS THIS. If you forget how to use a command or want to know if it can do something, use this!

Moving on to directory navigation commands...

- The 'ls' command
- Stands for list directory
- Lists all of the current directory contents!
 - UE1: ls
 - UE2: ls -l
 - UE3: ls -la
- Wait a sec...what are those dash letter things at the end?
 - A quick talk about flags!!



```
mandar@YourOwnLinux: ~  
mandar@YourOwnLinux:~$ ls -l  
total 1504  
-rw-rw-r-- 1 mandar mandar 107 Sep 24 06:19 a.c  
-rwxr--r-- 1 root root 122 Sep 26 07:26 Add.sh~  
-rwxrwxr-x 1 mandar mandar 7157 Sep 24 06:19 a.out  
-rw-rw-r-- 1 mandar mandar 5 Oct 14 00:34 ascii  
-rw-rw-r-- 1 mandar mandar 110 Sep 19 00:54 backup_helloWorld.java  
-rw-rw-r-- 1 mandar mandar 73 Sep 19 00:50 cat_demo.doc  
drwxr-xr-x 2 mandar mandar 4096 Nov 16 05:57 Desktop  
-rw-rw-r-- 1 mandar mandar 136 Sep 26 08:51 Div.sh~  
drwxr-xr-x 2 mandar mandar 4096 Sep 1 23:47 Documents  
drwxr-xr-x 2 mandar mandar 4096 Sep 1 23:47 Downloads  
-rw-rw-r-- 1 mandar mandar 0 Jan 19 21:00 dropbox.deb  
-rwxr--r-- 1 mandar mandar 135 Sep 26 11:16 EvenOdd.sh~  
-rw-r--r-- 1 mandar mandar 8445 Sep 1 23:42 examples.desktop  
-rwxr--r-- 1 mandar mandar 143 Sep 26 11:43 Fact.sh~  
-rw-rw-r-- 1 mandar mandar 110 Sep 17 23:11 helloWorld.java  
-rwxr--r-- 1 mandar mandar 67 Sep 19 00:32 helloWorld.sh~  
-rw-rw-r-- 1 mandar mandar 290 Sep 19 01:14 helloWorld.sh.java  
-rw-rw-r-- 1 mandar mandar 177 Sep 19 01:13 helloWorld.sh.java~  
-rwxr--r-- 1 mandar mandar 178 Oct 14 09:31 Host.sh  
-rw-rw-r-- 1 mandar mandar 198 Sep 27 06:59 ManU.txt  
-rw-rw-r-- 1 mandar mandar 0 Sep 27 06:45 ManU.txt~  
-rwxr--r-- 1 mandar mandar 122 Sep 26 08:15 Mul.sh~  
drwxr-xr-x 2 mandar mandar 4096 Sep 1 23:47 Music  
drwxr-xr-x 2 mandar mandar 4096 Sep 1 23:47 Pictures  
-rwxr--r-- 1 mandar mandar 337 Sep 26 12:08 Prime.sh~
```

HERE COME THE COMMANDS!

- **cd** (stands for 'change directory'): lets you change directories
 - UE: **cd /home/<your username>**
 - Special characters: ~, ., .., /
- **pwd** (stands for 'print working directory'): lets you know where you are!
 - A lot more useful than it sounds!!
- **mkdir** (make directory): lets you make a directory (or 'folder')
 - UE1: **mkdir test_dir1**
 - UE2: **mkdir test_dir2 test_dir3**
- **touch**: a quick way to make an empty file
 - UE: **touch test_file1.txt test_file2.txt**

Moving, renaming, and removing

- **cp** (stands for copy!)
 - UE1: **cp test_file1.txt ~/test_dir1**
 - UE2: **cp test_file1.txt ..**
- **mv** (stands for move)
 - Like cp, but more like 'cut'
 - UE1: **mv test_file1.txt ~/test_dir2**
 - You can also rename files like this!
 - UE2: **mv test_file2.txt home_file.txt**
- **TAB COMPLETION: IT'S AWESOME**
- **rm** (stands for remove)
 - Used to remove both files AND directories!
 - UE1: **rm test_file1.txt**
 - UE2: **rm -r test_dir3**
 - UE3: **rm -rf test_dir2**
 - UE4 (DO NOT RUN THIS COMMAND)
 - **rm -rf test_dir1 /***
- Can anyone tell me what that asterisk is?

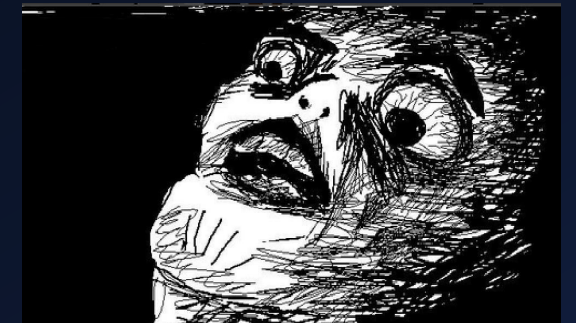


The Star Wildcard and the Pipe Operator

- The * is used to represent every string
 - You can use it to find, delete, etc. for files with a certain extensions
 - Ex: *.py / *.html / *.jpg / and much more
 - But BE CAREFUL!! The asterisk can also be used to DELETE all of your files so if you are using rm and * be sure your command is correct pressing enter!
 - You can use the -i flag when using rm for an **Interactive** way of deleting in order to prevent accidental deletion.
- The Pipe Operator (|) is used as a form of redirection
 - It will send the output from the first command to the next command after the pipe
 - *Command One | Command Two | Command Three*
 - *History | grep vim*

Output commands

- **echo**: writes arguments to standard out
 - UE1: **echo Hello everybody!**
 - This can also be used to let you know stuff that's in your directory!
 - UE2: **echo t***
- **cat** (no, not like the animal!): Prints out file contents
 - UE1: **cat test_file1.txt**
- **less**: like cat, but lets you scroll up and down
 - UE1: **less test_file1.txt**
 - UE2: **man ls | less**



Other Cool and Useful Commands

- `ctrl-l`: Clears the screen
- `ctrl-c`: Interrupts the current program running, most programs exit and give control back to the user
- `grep`: one of the methods used to search for files in the linux file system
- `whoami`: prints the username of whatever user you are logged in as out to the screen
- `find`: another method to search for files



Now then, permissions...

```
shum@sol:~$ ls -l
total 20
drwx----- 2 shum  staff  4096 Jan 16 22:04 Mail
drwx----- 3 shum  staff  4096 Jan 16 14:15 csc128
drwxr-xr-x  2 shum  staff  4096 Jan 13 16:42 public
drwxr-xr-x  2 shum  staff  4096 Jan 16 14:07 public_html
-rw-r--r--  1 shum  staff   628 Jan 15 20:04 verse
```

The diagram illustrates the components of the `ls -l` command output:

- file type**: Indicated by the first character of the permissions (e.g., `d` for directory, `-` for regular file).
- number of hard links**: The first number in the permissions field (e.g., `2`).
- user (owner) name**: The second name in the permissions field (e.g., `shum`).
- group name**: The second name in the permissions field (e.g., `staff`).
- size**: The file size in bytes (e.g., `4096`).
- date/time last modified**: The date and time the file was last modified (e.g., `Jan 16 22:04`).
- filename**: The name of the file (e.g., `Mail`).
- permissions**: The permissions are broken down into:
 - user permissions**: The first three characters of the permissions field (e.g., `drwx`).
 - group permissions**: The next three characters of the permissions field (e.g., `-----`).
 - other (everyone) permissions**: The last three characters of the permissions field (e.g., `-----`).

The permissions are further detailed as follows:

- r**: readable
- w**: writeable
- x**: executable

Useful permissions commands

- **sudo**: run a command as an administrative user
 - Some commands REQUIRE this. Other situations may relate back to permissions on the file/directory/user
 - UE1: **sudo su**
- **chmod**: lets you change the permissions on a file or directory
 - Must be run as ROOT if you are changing other's files
 - UE1: **sudo chmod 777 test_dir1** ← THIS IS A BAD IDEA
 - UE2: **sudo chmod 000 test_file1.txt** ← NO ONE CAN ACCESS THIS (except for admins)
 - 4 = read, 2 = write, 1 = execute
 - REGULAR FILES = 644, DIRECTORIES AND EXECUTABLE FILES = 755

VI / VIM (Side Note: VIM > EMACS)

- Comes with pretty much every distro of Linux
- Very simple to use at a basic level, but has a LOT of depth to it if you want to get super good at it.
- Open up your text file: **vim test_file1.txt**



The Basic Vim cheat sheet!

- To type in vim, enter **insert mode** by hitting the 'i' key
 - Exit out of this by hitting 'esc'
- To shift into copy/cut/paste mode, enter **visual mode** by hitting the 'v' key
 - While in visual, you can use the keyboard to select lines of text!
 - You can copy a line of text by hitting the 'y' key in visual mode
 - You can paste by hitting the 'p' key
 - You can also cut using the 'd' key
- To save and/or quit after editing, use the **command mode** (all commands start with ':')
 - :w = save
 - :q = quit
 - :wq = save and quit
 - :w! = force write
 - :q! = force quit

Now, let's talk about the Linux Directory Structure!

- If you do a `cd /` and an `ls` in there, you'll see a bunch of directories in there.
- Note: Directories are in blue, files are white!
- Each of these directories hold a lot of important information, and are organized in a certain way!
- So here's what you need to know...



Linux Directory Structure Goodies

- **home:** home directories for all users
 - **root:** home directory for the root user (DO NOT MESS WITH THIS)
- **bin:** Common single user linux commands/binaries
 - You'll notice lots of yellow files in here...what does that mean?
- **etc:** Configuration files + startup/shutdown scripts
- **var:** 'variable' files, or files that are expected to grow in size
 - Database files (var/lib), log files (var/log)
- **boot:** Boot loader files (kernel files)
- **usr:** User programs for 'second level' programs
 - **usr/bin:** usually where you look if the command you're running isn't inside of /bin
- **proc:** System process information

A few final-but important-commands

- Updating and installing on your system: **package managers**
 - Different on a lot of distros
 - CentOS uses **yum**, Ubuntu uses **apt**, Fedora uses **dnf**.
 - `<package manager> update`
 - `<package manager> install <package name>`
- Archive + Compression: **tar**
 - `tar czvf <name of file> <directory to compress>`
 - `tar xzvf <name of archive file>`
 - You may also run into **zip** or **bzip2** files – use your man pages!!!



Searching files and directories

- Looking for a particular named file/directory? Use the **find** command.
 - **find <directory to search> -name "<name of file/directory>"**
 - **find <directory to search> -type f -name "<name of file>"**
 - Find has lots of cool options associated with it-MAN!!!
- Need to look through your files to see if there's a certain word? Use the **grep** command! (I really like this command)
 - **grep -i "<string in file>" <file name>**
 - **grep -rni "<string in file>" <directory/to/search>**

SSH

- Secure **S**hell (SSH)
- Allows you to remotely log into another system/server
 - When you're logging in to GL you are SSH'ing into UMBC Server
- `ssh <username>@<destination>`
- `ssh -l <username> <destination>`

The background is a solid dark blue. On the left side, there are several parallel teal lines that start from the top and extend downwards, with some lines turning slightly to the right. On the bottom right, there are several parallel teal lines that start from the bottom and extend upwards and to the left.

Virtualization and VMs

So What is Virtualization??

- Virtualization is technology that allows you to create multiple simulated environments or dedicated resources from a single, physical hardware system
- Software called a hypervisor connects directly to that hardware and allows you to split 1 system into separate, distinct, and secure environments known as virtual machines (VMs). They take your physical resources and divide them up so that virtual environments can use them.



VMs

- A VM, or Virtual Machine, is an operating system that is installed on some software that mimics what a real environment of that operating system.
- Each VM is independent of each other but it is very possible to have VMs talk to each other.
- Because we have virtualization we can have the desktop environment of an operating system without actually having to have the hardware for that operating system.

Virtualization Software

There are 2 main softwares: VirtualBox and VMware. We recommend using VirtualBox but either one is fine.

The VMware logo, consisting of the word 'vmware' in a lowercase, sans-serif font. A registered trademark symbol (®) is located at the top right of the word.

Common Pitfalls - HyperV

This allows your computer to virtualization

For Windows 10: Boot into the BIOS, Under Security find System Security, Enable VTx if it is not already, save and exit, continue to boot normally, right click on the Windows button and select 'Apps and Features'. Select Turn Windows Features on or off. Select Hyper-V and click OK.

For Mac: Type `sysctl -a | grep machdep.cpu.features` and look for VMX (if it's there you are good)

For Linux: Power on the Machine and go into the BIOS, Open the Processor Submenu, Enable Intel Virtualization Technology (also known as Intel VT), Save and Exit

Booting into the BIOS

On startup:

- Windows: Hold F2
- Linux: Hold Alt and F4, F1, or Delete



Any Questions?

Lab

- First we want you to install VirtualBox
<https://www.virtualbox.org/wiki/Downloads>
- Next we'd like you to install Kali Linux and Ubuntu 18.04
<http://releases.ubuntu.com/18.04/>
<https://www.offensive-security.com/kali-linux-vm-vmware-virtualbox-image-download/>
For Ubuntu look for "ubuntu-18.04.1-desktop-amd64.iso"
For Kali, go to VirtualBox Images and get the 64-bit OVA.
- Once you have those installed go start either one and go to this site and begin working on Bandit:
 - <http://overthewire.org/wargames/bandit/bandit0.html>