Introduction to Unix

The Foundation for Cybersecurity

UMASS CYBERSEC CLUB

Disclaimer!

This might feel overwhelming... But that’s normal!

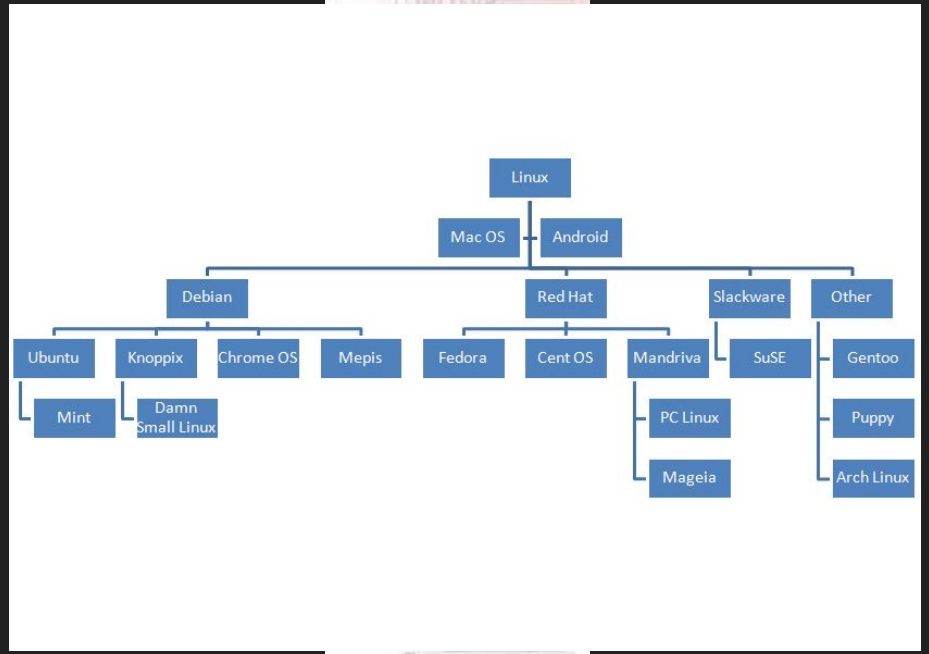
Take 5 minutes. Introduce yourself to the people around you and discuss the following question:

What is Unix? If you don’t know, what have you heard about it?

UMASS CYBERSEC CLUB

What is Unix?

- Operating System first developed in the 1960s
  - Many different versions called **distributions**: macOS, GNU/Linux



UMASS CYBERSEC CLUB

Why it Matters

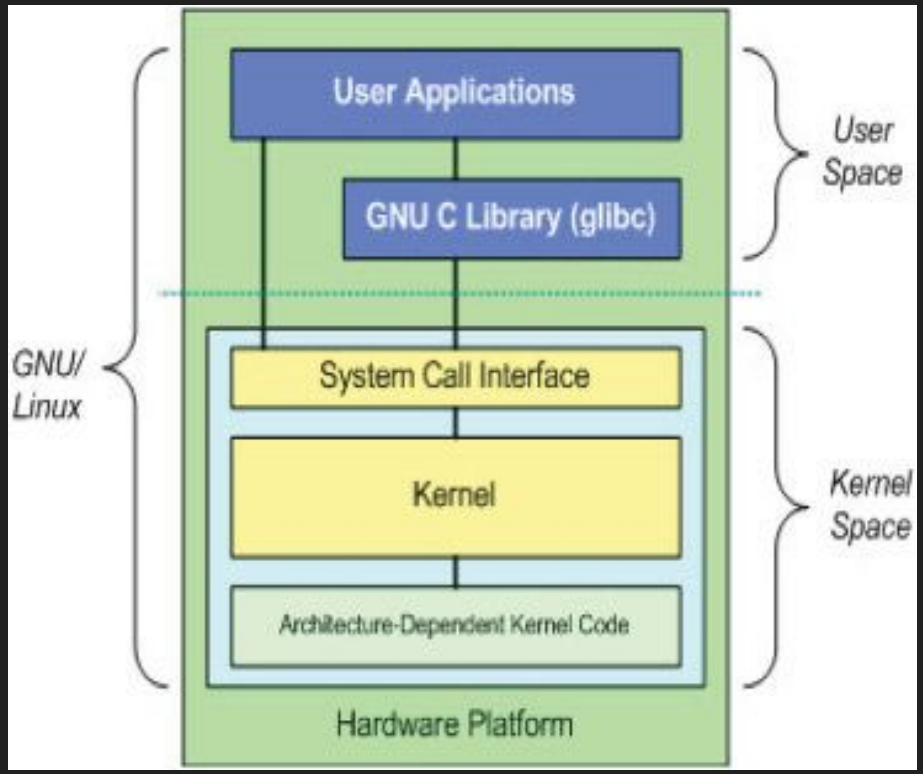
Even if you don't go into Cybersec, Unix is used everyday by SWEs and IT professionals to utilize tools like git (github), Docker, and to build things in different environments

Unix-like OS used in millions of mobile devices (Android and iOS), macOS, servers, etc.

UMASS CYBERSEC CLUB

Unix as an Operating System

- **Kernel**: The core of the unix operating system meaning it has full control over your whole system
- Analogy: Think of UNIX as a factory and the kernel is the big boss. It has control over everything!
- Fun example: https://www.leagueoflegends.com/en-us/news/dev/dev-null-anti-cheat-kernel-driver/

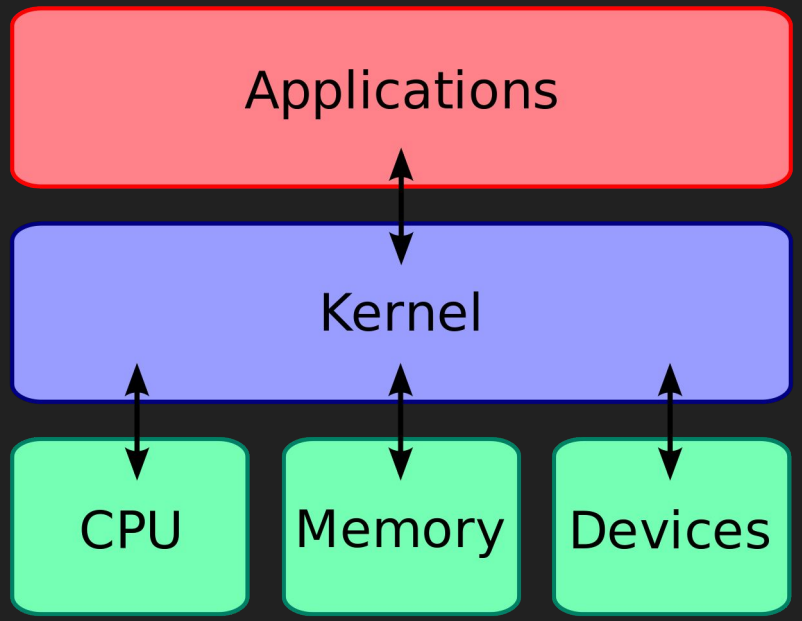


UMASS CYBERSEC CLUB

Unix as an Operating System

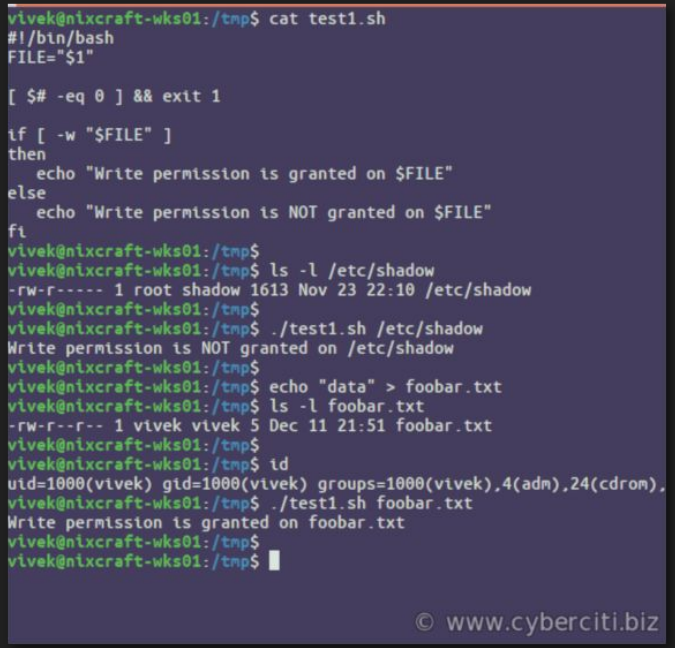
What does the Kernel do?

- First thing loaded
- Uses processes which are executables with a unique ID (PID) and memory
  - Chrome/Firefox
- Manages Input/Output (I/O) from programs/processes
- Manage memory and hardware accessories



The Shell

- The shell lets users run things on their computer through commands
  - Can also communicate with the kernel
  - Sometimes called command line interface(CLI)
- Analogy: Think of the shell as a floor manager. It interacts with the big boss(kernel) and controls the workers(processes)



UMASS CYBERSEC CLUB

The Shell

- The default shell in ubuntu is called **bash**, which is also a command you can run
  - In your shell, you should have a $ next to your username meaning you're a non-root user
  - Bash allows you to also write scripts to automate certain tasks

```
#!/bin/bash

i=0

while [[ $i -lt 11 ]]
do
        if [[ "$i" == '9' ]]
        then
                echo "Number $i!"
                ((i++))
                continue
        fi
        echo $i
        ((i++))
done
```

UMASS CYBERSEC CLUB

Open up your terminal and type the following commands into the shell. What did each command do? Discuss this with the people around you.

ls ← (L and S)
whoami
apt search python

UMASS CYBERSEC CLUB

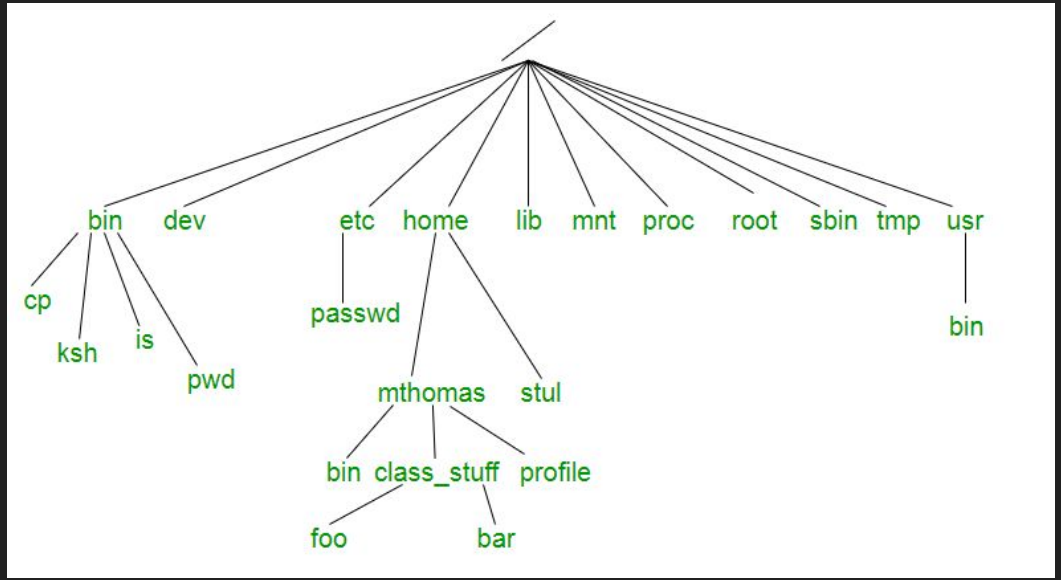
Files

- Everything in Unix is considered a **file**
- **File:**
  - Readable (text), Binary (executable), compressed archives, etc..
  - If you want to make a file executable run **chmod +x file_name**
  - Directories
    - /bin/
    - /etc/
    - /home/[user-name] ← if you open your terminal you are here!

UMASS CYBERSEC CLUB

File System

- Files and directories are ordered in a tree
- The bottom directory, called root (not to be confused with root user) can be found using **cd /**
- The file system contains all the users, commands, and system configurations stored as files



UMASS CYBERSEC CLUB

Open up your terminal and type the following commands into the shell. What did each command do? Discuss this with the people around you.

```
cd /
cat /etc/passwd
ls /usr | grep “hexdump”
file /usr/bin/whoami
```

UMASS CYBERSEC CLUB

Package Managers

- Package managers keep track of all the applications, libraries, and dependences you install on your Operating System
- You can see all the installed packages by typing **apt list** in your terminal
- Sometimes your installs might fail due to out of date dependencies, in that case run **sudo apt update**
- To install a new application you'll have to do **sudo apt install [package-name]**
- You can and might have to install applications by other means:
  - Running bash script installers **./name_of_installer_script.sh**
  - Building from **git** repositories
  - Copy paste commands from the official installation guides

UMASS CYBERSEC CLUB

Users and Groups

- Each user has permissions on what they can read, write, and execute
  - Type **ls -la** in your console and you can see the various permissions each file requires in your current directory
- Groups allow you to give multiple users defined permissions more easily
  - The **sudo** group is the Windows equivalent of giving a user Administrator permissions
- The root user is that can do anything
  - Think Administrator in Windows!
- You can execute a command as root by typing **sudo** before a command
  - You must supply a password and have valid permissions

UMASS CYBERSEC CLUB

Text Editors - The Big Three

- Sometimes you'll have to edit files whether it be for code editing, writing notes, or even more
- There are many options but most people will pick one of three base text editors:
  - **nano**
    - This should come pre-installed on your Kali machine so you can type **nano file.txt** in your terminal to open and/or create a file called **file.txt** and allow type and edit it
    - **CRTL+X** is the hotkey to close a file and type **y** followed by **enter** to save it to **file.txt**
  - **Vim (or Neovim)**
    - To install neovim we can run **sudo apt install vim**
    - Type in **vim file.txt** to open a file in the vim text editor
    - To write text type **i** then to exit vim hit ESCAPE then type **:wq**
  - **emacs**
    - To install emacs we can run **sudo apt install emacs**
    - We can use **emacs file.txt** to open a file in the emacs text editor
    - We can write text to files by typing
      - Saving is CRTL+X followed by CRTL+S
      - Quitting is CRTL+X followed by CRTL+C

UMASS CYBERSEC CLUB

But.... You could just use anything

- **Visual Studio Code**
  - This is the average code editor that you have probably used in many of your classes already
  - You can install it by following this guide https://code.visualstudio.com/docs/setup/linux#_debian-and-ubuntu-based-distributions

Piping, Redirection, and More!

- Sometimes you might want to chain commands together to get filter outputs of command or even pass it directly to another
- **Pipe:** The “|” operator (no quotes) will send the output of the command on the left to the command on the right
    - So if we run ls | base64 the output will be a base64 encoded output of the ls command
- **Right Redirection:** The “>” operator will send the output of the command on the left to the file name on the right
    - So if we run ls > output.txt the output of ls will be in the file **output.txt**
- If we want to write multiple commands on one line we can separate them with semicolons
    - So we can do ls; whoami which will run output both of the commands to our terminal

UMASS CYBERSEC CLUB

Connecting To Other Machines - Secure Shell

- Sometimes you'll have to connect to remote Linux machines whether it be for classes at UMass (Edlab) or personal servers
- To connect to remote servers most machines have an application called Secure Shell (SSH)
- To connect to a remote server using ssh you can use:

ssh user_name@server_hostname

Using other Commands

- For starting basic commands you can use the following Cheatsheet
- For most other commands you can Google or ask in the server about what commands you use in a particular context
  - You might phrase a Google query like: “Linux command to view hex of files”
- If you have specific questions about a command you can type **man command-name**

UMASS CYBERSEC CLUB

Hivestorm 2024

- Same Organizers as the cyberdefense competition (CCDC)
- Teams of 4 (+2 Alts)
- Virtual competition open to all students with no team limit per school - Great way to get experience!
- Auditing a simulated corporate network for security issues (like CCDC, but no active threats)
- Applications & More Info released today! (Due Sept 27)
- Competition held Wednesday, October 16

UMASS CYBERSEC CLUB

https://www.hivestorm.org/event.html - Info
https://forms.gle/BViUNekAobFckLNh7 - Signup

We have 5+ challenges on our Training Platform.
https://training.umasscybersec.org

Stick around and try them out with the people around you.
All flags are UMASS{text}

UMASS CYBERSEC CLUB

Extra Stuff (May help with 5th challenge)

procfs

- Special file system that stores process information in files under /proc
- OS will read and write from these files to keep them updated
- Access info about a specific PID by going to /proc/<PID>
    - Different files/directories contain specific information
        - The fd directory contains file descriptors
        - The environ file contains ENV variables at start of process
        - The root directory links to root dir for process (generally /)
        - The status file stores a lot of general information
        - The maps file stores memory maps (stack, heap, dynamic libs)