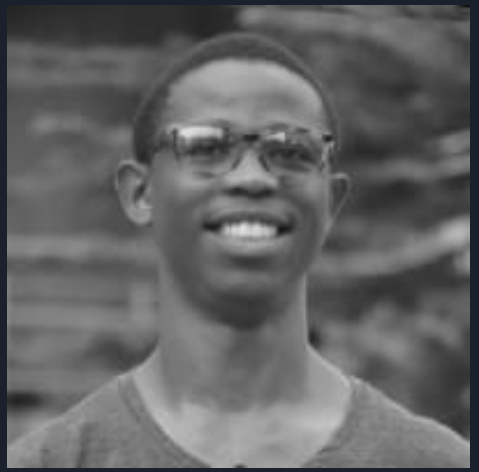
Introduction to Linux Masterclass

25th March 2024 - 28th March 2024

Nemuel Wainaina

> whoami



Nemuel Wainaina

- Software Engineer
- Security Researcher
- Technical Instructor (1.7k+ students)

Github : https://github.com/nemzyxt

LinkedIn : https://linkedin.com/in/nemuel-wainaina

Twitter : https://twitter.com/nemuelwainaina

Medium : https://medium.com/@nemzyxt

> curriculum

Day 1: Getting Started
- Brief Linux history
- Why we should learn it
- Environment setup
- The Linux filesystem

Day 2: Command-line basics 1
- Navigating around the system
- Finding stuff
- File & directory operations
- Text editing
- Getting system information

Day 3: Command-line basics 2
- Managing software
- Networking operations
- Managing processes
- Managing users
- Managing permissions
- A note on Bash scripting

Day 4: Kali Linux
- Brief intro to Kali Linux
- Demo: Cracking a WPA key
- Q&A Session

Day 1: Getting Started

Nemuel Wainaina

25th March 2024

> history of Linux

- Created by Linus Torvalds in 1991
- Released as a free and open-source project
- Drew attention from developers all around the world who started to contribute

- Development still in progress: https://github.com/torvalds/linux
- With the rise in popularity, came its adoption into various kinds of systems used by people

> where it is used

- Desktop systems eg. Ubuntu, Fedora, Kali Linux
- Servers eg. CentOS, Ubuntu Server
- Mobile devices eg. Android
- Embedded systems eg. Smart TVs, Automobile Infotainment systems
- Networking equipment eg. Routers, Switches

> so what then is this thing, ‘Linux’?

- Linux is an operating system kernel
- An operating system serves as an interface between computer hardware and the users, facilitating interaction and doing resource management
- The kernel is to the operating system what the heart is to the human body
- It handles things like device management, process management, memory management, etc.

> comes Linux distributions ...

- Just as the human heart can't do much good on it's own, the Linux kernel is useful when integrated with other things eg. a nice Graphical User Interface
- Different companies (and individuals even) have built complete Operating Systems on top of the Linux kernel eg. Canonical has built Ubuntu, Google has built Android, Offensive Security has built Kali Linux
- All these different operating systems are what we call Linux distributions (or Linux distros)
- List of Linux distros: https://distrowatch.com

> the perks of Linux

- Free and open-source (who doesn’t want free things ?)
- Stability and reliability
- Performance
- Community support
- Security
- etc.

> installing Linux (or any operating system)

2 ways:

a) On bare metal
b) As a virtual machine

- For starters, the second method is preferred
    1. Download and install virtualization software eg. Virtualbox
    2. Get the OS installation file (iso file)
    3. Install it on the virtualization software and start using it

> practical

Virtualbox:

- https://www.virtualbox.org/wiki/Downloads

Kali Linux (what we will be using):

- https://www.kali.org/get-kali/#kali-virtual-machines

> the Linux filesystem

/ - root directory of the entire filesystem hierarchy
/bin/ - essential user command binaries
/sbin/ - system binaries
/boot/ - static files of the bootloader (**DON'T DELETE ANYTHING HERE**)
/mount/ - serves as a mount point for removable media devices
/dev/ - used by the kernel to manage hardware devices
/root/ - home directory for the root user
/home/ - user home directories eg. /home/nemuel for the user 'nemuel'
/tmp/ - temporary files
/var/ - variable files eg. logs
/usr/ - multi-user utilities and applications

NOTE: Different distributions may have other different directories

> some differences from Windows

- Directory structure is unified in Linux
- Forward slashes instead of backslashes (Windows: C:\Users\Nemuel, Linux: /home/nemuel)
- Case sensitivity eg. music is not the same as Music
- File extensions
- Command-line adoption (historical context, target audience)
- File permissions enforcement

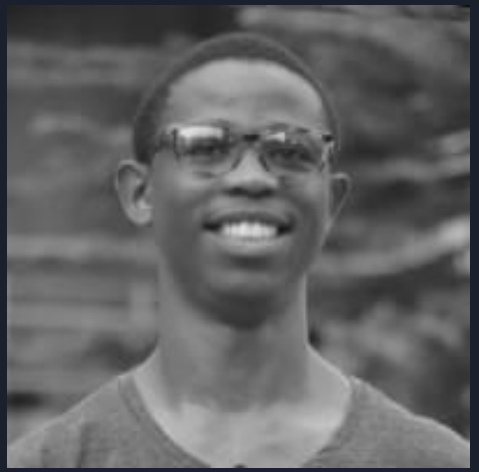
Q&A

Day 2: Command-line Basics (part 1)

Nemuel Wainaina

26th March 2024

> whoami



Nemuel Wainaina

- Software Engineer
- Security Researcher
- Technical Instructor (1.7k+ students)

Github : https://github.com/nemzyxt

LinkedIn : https://linkedin.com/in/nemuel-wainaina

Twitter : https://twitter.com/nemuelwainaina

Medium : https://medium.com/@nemzyxt

> different types of UIs (User Interfaces)

- **UI:** a way for a user to interact with a software application or digital device.

- This could take various forms:
  - GUI (Graphical User Interface)
  - MDI (Menu Driven Interface)
  - CLI (Command Line Interface)

> so why learn the command-line?

- Efficiency: You can accomplish more in fewer steps
- More flexibility and control
- Automation (through scripting)
- Remote management
- Career opportunities: Sys admin, DevOps, Software dev, Cybersec, Data analysis, etc.

> command-line foundations ...

$ man <command_name> :

- Gives us the manual pages for the command (documentation)

On switches and command-line arguments:

- **Command-line arguments** are pieces of information required to complete the intended task
  - eg. ***cp file1.txt destination/directory/***
  - ‘file1.txt’ and ‘destination/directory/’ are required by cp command
- **Switches** are additional modifiers or options that can be added to a command to either change its behavior or provide additional functionality
  - eg. ***ls -a*** : ‘-a ‘ extends the ls command to list even hidden files
  - Short-form: -a (single hyphen), long-form: --all (2 hyphens)

> some basic commands

File system navigation:

- **pwd :** Print the current working directory
- **ls :** List the contents of the current directory
- **cd :** Change directory

Finding stuff:

- **locate :** Search for files and directories in a pre-built database
- **whereis :** Get the location of a binary file (executable)
- **find :** The most flexible and powerful command for finding files
- **grep :** A filter to search for keywords

Text editing:

- **nano :** basic command-line text editor

> some basic commands (cont'd)

File & directory operations :

- **mkdir :** create a new empty directory
- **rmdir :** delete an empty directory
- **touch :** create an empty file
- **file :** view the type of file
- **rm :** delete/remove files (and directories)
- **cp :** copy files and directories
- **mv :** move or rename files and directories
- **cat :** concatenate or display the content of files
- **more, less :** view file content page by page
- **head :** display the beginning of a file
- **tail :** display the end of a file

> some basic commands (cont'd)

Getting system information:

- **uname :** get system information

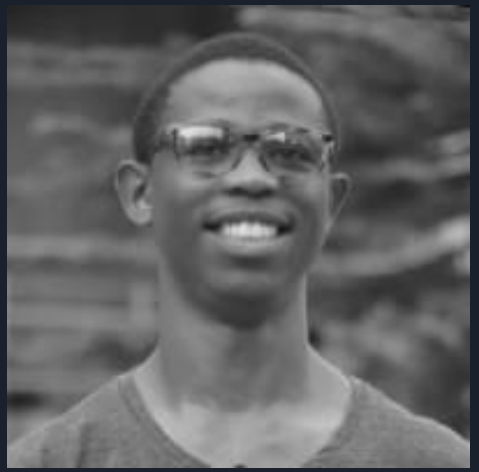
Q&A

Day 3: Command-line Basics (part 2)

Nemuel Wainaina

27th March 2024

> whoami



Nemuel Wainaina

- Software Engineer
- Security Researcher
- Technical Instructor (1.7k+ students)

Github : https://github.com/nemzyxt

LinkedIn : https://linkedin.com/in/nemuel-wainaina

Twitter : https://twitter.com/nemuelwainaina

Medium : https://medium.com/@nemzyxt

> more commands ...

Managing users:

- **whoami :** display the current username
- **who :** display ***information*** about users who are currently logged in
- **users :** display a ***list*** of currently logged on users
- **adduser, useradd :** create a new user
- **passwd :** change user password
- **sudo <commands> :** execute the command(s) with root privileges

> more commands ...

Software/package management:

- **apt-get / yum / dnf :** install, update, or remove software packages
- **apt-get install :** install a software package
- **apt-get update :** update the local package index
- **apt-get upgrade :** upgrade installed packages to their latest versions
- **apt-get dist-upgrade :** upgrade installed packages (intelligently)
- **apt-get remove :** remove the package but retain config files
- **apt-get purge :** remove both the package and config files
- **apt-cache search :** search for package name in online package database
- **apt-get do-release-upgrade :** upgrade to newer distro release

> more commands ...

Networking operations:

- **ping :** check network connectivity
- **ifconfig :** display and configure network interfaces
- **netstat :** display network connections, routing tables and interface statistics
- **ssh :** securely connect to remote machines
- **scp :** securely copy files between machines
- **wget :** download files from the internet

> managing processes

Managing processes:

- **ps :** display information about running processes
- **top :** display dynamic real-time information about running processes
- **kill :** send a signal to a process, default being the TERM (terminate) signal
- **killall :** kill processes by name
- **pgrep :** look up processes by name and other attributes
- **pkill :** send signal to processes based on certain criteria

> managing permissions

- Permissions dictate who can read, write or execute files & directories on the system
- They are divided into 3 sets: **User** (U), **Group** (G), **Others** (O)

- There are 3 primary types of permissions:
  - **Read (r) :** allows the user to view the contents of the file or directory
  - **Write (w) :** allows the user to modify the contents of the file or directory
  - **Execute (x) :** allows the user to execute the file if it is a program or script

- The underlying system only understands 1s and 0s language …
  - r w x => 1 1 1 => 4 2 1 Therefore: 4 + 2 + 1 = 7 (all permissions granted)

> more on permissions

Examples :

- rw- : 4 + 2 = 6
- r-x : 4 + 1 = 5
- **---** : 0 + 0 + 0 = 0

To change file permissions:

- **chmod :** change file or directory permissions

> a note on bash scripting

- **Bash (Bourne Again Shell) :** a command-line interpreter (shell) for Linux-based systems. It's the default shell even on MacOS!

- **Bash scripting :** involves writing scripts that contain a series of commands to be executed by the Bash shell. These scripts can be used to automate repetitive tasks, perform system administration tasks, etc.

- To create one, create a new file, with a **'.sh'** extension. Add some commands to it. Give it **execute (x)** permissions. Run it with:
  - ***./scriptname.sh***

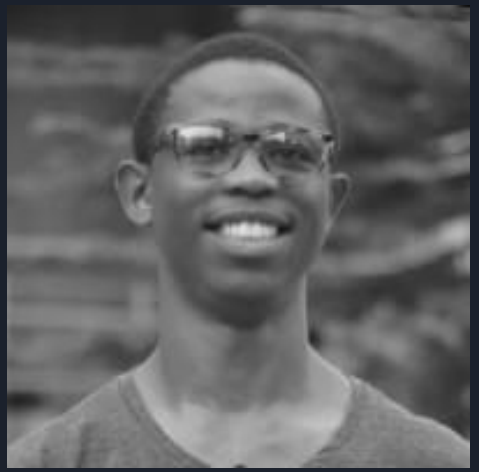
Q&A

Day 4: Kali Linux (basics)

Nemuel Wainaina

28th March 2024

> whoami



Nemuel Wainaina

- Software Engineer
- Security Researcher
- Technical Instructor (1.7k+ students)

Github : https://github.com/nemzyxt

LinkedIn : https://linkedin.com/in/nemuel-wainaina

Twitter : https://twitter.com/nemuelwainaina

Medium : https://medium.com/@nemzyxt

> a bit on Kali Linux ...

- It is a Debian-based distribution designed for use by cybersecurity practitioners on things like pentesting and digital forensics.
- Comes preinstalled with 600+ tools for this work.
- Some of the benefits of using it for cybersecurity are:
  - tools are preconfigured, so no need to install them separately
  - nice organization of tools by the purpose they serve eg. exploitation, information gathering, etc.
  - cross-platform compatibility, meaning it can be installed and used across a variety of hardware platforms and architectures.

- To download Kali Linux: https://www.kali.org/get-kali/#kali-platforms

> demo: cracking a WPA key

- Challenge link: https://cybertalents.com/challenges/machines/wpa-crack

- Download the **cap** file (handshake capture): Remember the ‘wget’ command?
- Look through the ‘Wireless Attack’ tools and choose the right one (aircrack-ng)
- Go through its usage (manual pages or even help menu)
- Choose a wordlist to use (under ‘Password Attacks’ -> ‘wordlists’) eg. rockyou.txt
- Execute the command to start cracking …
    - ***aircrack-ng -w /path/to/rockyou.txt /path/to/handshake.cap***

> next steps

- Practise, practise, practise!
- Learn Bash scripting
- Learn version control with git
- Practice using Kali Linux (ethically!)
- Share your newly acquired knowledge & skills with others

- You can also now consider building your own cool Linux distro in future 😀🥳

Q&A