**Azuonye Daniel's Class Notes**

**Ethical Hacking and Cyber Security**  
**September 2024 — 6th Period**

**Welcome to your reference guide!**

Use **Ctrl-F** to search terms, commands, and definitions easily.  
Each section contains a short-form definition followed by more details if needed.  
Happy hacking!

**INDEX**

* Commands
* Networking
* File Permissions & chmod
* Process Management
* Package Management
* Terminology
* Logic Gates/Boolean
* Basic Linux Commands
* Computer Parts
* Keywords/Abbreviations
* Linux Basics: sudo, Navigation, File & Directory Management

**Commands**

1. **ls**:  
   Lists contents of a directory. For a detailed view (including hidden files), use ls -la.
2. **pwd**:  
   Prints your current location (working directory).

Also means (Present working Directory)

1. **cd**:  
   Changes the directory. Use cd .. to move up one level.
2. **mkdir**:  
   Creates a new directory (folder). Example: mkdir myfolder creates a folder named "myfolder."
3. **touch**:  
   Creates an empty file. Example: touch newfile.txt creates a file named "newfile.txt."
4. **rm**:  
   Removes a file. Use rm -r to delete a directory along with its contents.
5. **cp**:  
   Copies files or directories. Example: cp file1.txt /newfolder/ copies file1 to /newfolder/.
6. **mv**:  
   Moves or renames files. Example: mv oldname.txt newname.txt renames the file.
7. **cat**:  
   Shows the contents of a file. Example: cat file.txt displays the content of "file.txt."
8. **echo**:  
   Prints text to the terminal. Example: echo Hello World outputs "Hello World."
9. **man**:  
   Opens the manual for a command. Example: man ls displays the manual for the ls command.
10. **grep**:  
    Searches for a specific pattern of text within files. Example: grep 'text' file.txt looks for "text" inside "file.txt."
11. **find**:  
    Searches for files in a directory hierarchy. Example: find / -name filename looks for files named "filename."
12. **sudo**:  
    Runs a command with elevated privileges. Example: sudo apt-get update.
13. **df**:  
    Displays disk space usage. Use df -h for human-readable sizes.
14. **du**:  
    Displays the size of files and directories. Use du -h for human-readable sizes.
15. **chmod**:  
    Changes file permissions. Example: chmod 755 script.sh grants the owner full permissions and others read/execute permissions.
16. **chown**:  
    Changes file ownership. Example: chown user:group file.txt.
17. **ps**:  
    Displays currently running processes.
18. **kill**:  
    Terminates a process by its ID. Example: kill 1234.
19. **tar**:  
    Compresses or extracts files. Example: tar -cvf archive.tar directory/ creates an archive of the directory.
20. **zip**:  
    Compresses files into a .zip archive. Example: zip archive.zip file1 file2.
21. **unzip**:  
    Extracts files from a .zip archive. Example: unzip archive.zip.
22. **ssh**:  
    Connects to a remote machine via Secure Shell (SSH). Example: ssh user@hostname.

**Networking**

1. **ifconfig** or **ip a**:  
   Shows network interfaces and IP addresses.
2. **ping**:  
   Tests network connectivity. Example: ping google.com checks if you can reach Google's servers.
3. **traceroute**:  
   Traces the path packets take to a destination. Example: traceroute google.com.
4. **netstat**:  
   Displays network connections and open ports. Use netstat -tuln for a list of TCP/UDP connections.
5. **nmap**:  
   A tool for scanning open ports and network mapping. Example: nmap localhost scans your own machine.
6. **host**:  
   Finds the IP address of a domain name or the domain name of an IP address. Example: host google.com.
7. **wget**:  
   Downloads files from the internet. Example: wget http://example.com/file.zip.
8. **dig**:  
   Retrieves DNS information about a domain. Example: dig google.com.
9. **iwconfig**:  
   Configures wireless network interfaces on Linux.
10. **ss**:  
    Displays socket statistics.

**File Permissions & chmod Examples**

1. **chmod**:  
   Changes file permissions with numeric codes. Permissions are broken down as follows:
   * **r** = Read (4)
   * **w** = Write (2)
   * **x** = Execute (1)

**Examples:**

* + chmod 755 file.sh: Owner has full access (read, write, execute); group and others can only read and execute.
  + chmod 644 file.txt: Owner can read and write; group and others can only read.
  + chmod 777 folder: Everyone has full permissions (read, write, execute).

**Common combinations:**

* + **700**: Owner has full access; no access for group or others.
  + **600**: Owner can read and write; no access for group or others.
  + **555**: All users can read and execute, but no one can write.
  + **400**: Only the owner can read; no access for group or others.

1. **chown**:  
   Changes the ownership of a file. Example: sudo chown user:group file.txt changes the owner and group.
2. **ls -l**:  
   Displays file permissions in a detailed format.

**Process Management**

1. **top**:  
   Shows a list of system processes and their resource usage in real-time.
2. **ps**:  
   Displays a snapshot of running processes at the time.
3. **kill**:  
   Terminates a process using its ID. Example: kill 1234 stops the process with ID 1234.
4. **df**:  
   Shows how much disk space is being used. Use df -h to see human-readable sizes.
5. **du**:  
   Displays the size of directories. Example: du -h foldername shows the size of a directory.
6. **free**:  
   Shows memory (RAM and swap) usage.

**Package Management**

1. **sudo apt-get update**:  
   Updates the package list to fetch the latest versions available.
2. **sudo apt-get upgrade**:  
   Upgrades all installed packages to the newest version available.
3. **sudo apt-get install**:  
   Installs new software. Example: sudo apt-get install nmap installs the Nmap tool.
4. **sudo apt-get remove**:  
   Removes an installed software package.

**Terminology**

1. **Ethical Hacking**:  
   The legal practice of breaking into systems to identify weaknesses before malicious hackers do.
2. **Penetration Testing**:  
   Simulates a cyberattack to assess a system's security.
3. **Firewall**:  
   A network security system that controls incoming and outgoing network traffic based on pre-set security rules.
4. **Vulnerability**:  
   A weakness in a system that can be exploited by a threat actor.
5. **Exploit**:  
   A method or piece of code used to take advantage of a vulnerability.
6. **Patch**:  
   An update designed to fix security flaws and vulnerabilities in software.
7. **Malware**:  
   Malicious software designed to damage or disrupt systems (e.g., viruses, worms, ransomware).
8. **Zero-Day**:  
   A vulnerability exploited before the software vendor has issued a fix.
9. **Encryption**:  
   The process of converting data into a secure format to prevent unauthorized access.
10. **IP Address**:  
    A unique identifier assigned to a device on a network.
11. **Port Scanning**:  
    A technique used to probe a system's open ports to identify potential entry points.

**Logic Gates/Boolean**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **A** | **B** | **AND** | **OR** | **XAND** | **NOR** | **NAND** | | **0** | 1 | F | T | F | F | **T** | | **1** | 0 | F | T | F | F | **T** | | **0** | 0 | F | F | T | T | **T** | | **1** | 1 | T | T | T | F | **F** | |
| **Explanation of Gates:**   * **AND: True if both inputs are true.** * **OR: True if at least one input is true.** * **XAND (Exclusive AND): True if both inputs are the same.** * **NOR: True if neither input is true.** * **NAND: True if at least one input is false (opposite of AND).** |
|  |

**Basic Linux Commands**

1. **ls**: List contents of a directory.
2. **cd**: Change directory.
3. **pwd**: Print current working directory.
4. **mkdir/rmdir**: Make/remove directories.
5. **touch**: Creates an empty file.
6. **rm**: Removes files or directories.
7. **mv**: Renames or moves files and directories.
8. **cat**: Displays the content of files.
9. **nano**: Opens the nano text editor to edit files.
10. **cp**: Copies files and directories.
11. **find**: Searches for files and directories.
12. **echo**: Prints text to the terminal.
13. **chmod**: Changes file permissions.
14. **chown**: Changes file ownership.
15. **ps**: Lists currently running processes.
16. **top**: Displays real-time system process statistics.
17. **kill**: Terminates a process by its ID.
18. **df**: Displays disk usage information.
19. **du**: Shows disk usage of directories.
20. **free**: Shows memory usage statistics.
21. **sudo**: Runs commands with elevated privileges.

**Computer Parts**

1. **Motherboard**: The main circuit board of the computer, connecting all components.
2. **CPU (Central Processing Unit)**: The brain of the computer, executing instructions.
3. **GPU (Graphics Processing Unit)**: Handles rendering graphics, particularly useful for gaming or video editing.
4. **RAM (Random Access Memory)**: Temporary memory that stores data for currently running programs.
5. **SSD (Solid State Drive)**: Storage device with faster access speeds than traditional hard drives.
6. **Power Supply**: Provides power to all components in the computer.
7. **Case**: Enclosure that houses the computer components.
8. **Cooler**: Used to dissipate heat generated by the CPU and GPU.

**Keywords/Abbreviations**

1. **IP Address**: Internet Protocol, a unique address assigned to each device on a network.
2. **GUI**: Graphical User Interface, allows users to interact with a system through visual icons.
3. **DNS**: Domain Name System, translates domain names to IP addresses.
4. **OS**: Operating System, software that manages hardware and software resources on a computer.

**What is sudo and How to Use It**

* **sudo** stands for "superuser do." It is used to run commands with elevated privileges, typically as the root user, allowing you to perform tasks that require administrative rights.
* **Usage**:
  + sudo <command>: Executes the command with superuser privileges.
  + Example: sudo apt-get update will update package lists, which requires administrator privileges.
  + **Password Prompt**: When you use sudo, it asks for your user password, not the root password.

**Navigating Files and Directories**

* **Directory**: A folder that contains files or other directories. Think of it as an organizational structure for your files.
* **File**: A single item that contains data, such as a text document, an image, or a program.

**Basic Navigation Commands**

1. **pwd**: Prints the current working directory. This tells you where you are in the directory structure.
2. **ls**: Lists the contents of the current directory.
   * Use ls -la for detailed information, including hidden files.
3. **cd <directory>**: Changes the current directory to the specified one.
   * cd .. moves you up one directory level.
   * cd ~ takes you to your home directory.

**Creating, Removing, and Editing Files & Folders**

**Making Directories and Files**

1. **mkdir <directory\_name>**: Creates a new directory (folder).
   * Example: mkdir myfolder creates a directory named "myfolder."
2. **touch <file\_name>**: Creates an empty file.
   * Example: touch myfile.txt creates an empty file named "myfile.txt."

**Removing Directories and Files**

1. **rm <file\_name>**: Removes a file.
   * Example: rm myfile.txt deletes "myfile.txt."
2. **rm -r <directory\_name>**: Removes a directory and its contents recursively.
   * Example: rm -r myfolder deletes the directory "myfolder" and all files within it.

**Editing Files**

1. **nano <file\_name>**: Opens the file in the nano text editor for editing.
   * Example: nano myfile.txt opens "myfile.txt" for editing.
   * Use Ctrl + O to save changes and Ctrl + X to exit the editor.

**What Are Directories and Files?**

* **Files**: The basic units of data storage in a system. Each file contains a set of information, such as text, images, or code, and has a unique name in its directory.
* **Directories (Folders)**: Containers for organizing files and subdirectories. Directories themselves can be nested, creating a hierarchical structure.

**Understanding File Structure:**

* The Linux filesystem is organized in a tree-like structure.
* The **root** directory / is at the top, and all other files and directories branch off from it.
* Common directories include:
  + **/home**: Contains user-specific directories.
  + **/etc**: Holds system configuration files.
  + **/var**: Contains log files and other variable data.
  + **/bin**: Stores binary executable files (programs).

Learning resources for you to explore during downtime in class.  
  
  
Free -  
Formally Lynda.com now LinkedIn Learning: [Home - LinkedIn Learning - Info Guides at Winnipeg Public Library](https://guides.wpl.winnipeg.ca/linkedinlearning)

Paid or freemium-  
Coursera -[coursera.org/](https://www.coursera.org/)

Grade school CompSci - [rexk12.com](http://www.rexk12.com/)

[cybrary.it](http://www.cybrary.it/)

[offsec.com](http://www.offsec.com/)

[hackaday.com](http://hackaday.com/)

[udemy.com](http://www.udemy.com/)

[rapid7.com/blog/tag/metasploit/](http://www.rapid7.com/blog/tag/metasploit/)

[canadalearningcode.ca/grow-with-google/](http://www.canadalearningcode.ca/grow-with-google/)

[w3schools.com/python/default.asp](http://www.w3schools.com/python/default.asp)

[codecademy.com](http://www.codecademy.com/)

[ocw.mit.edu](http://ocw.mit.edu/)

**Linux Command Line Study Guide for Upcoming Spot Checks.**

I will be away tomorrow Sept 19th and I want you to prepare for in-class spot checks when I return (Friday, and Monday) where you'll demonstrate your Linux command-line skills using Debian 12 in VMware. Focus on the following key areas:

**Objectives**

* Navigate the Linux file system using basic commands.
* Perform CRUD (Create, Read, Update, Delete) operations on files and directories.
* Modify file permissions using chmod.
* Execute commands with superuser privileges using sudo.

**1. Navigating the File System**

**Commands to Practice:**

* pwd — Displays your current directory.
* ls — Lists files and directories.
  + Options:
    - ls -l (detailed view)
    - ls -a (shows hidden files)
* cd — Changes the current directory.
  + Examples:
    - cd /path/to/directory (absolute path)
    - cd Documents (relative path)
    - cd .. (move up one directory)
    - cd ~ (go to home directory)

**Tips:**

* Use man [command] to read the manual for any command.
* Google searches like "How to use the ls command in Linux" can provide quick tutorials.
* Ask ChatGPT for explanations or examples if you're unsure how a command works.

**2. CRUD Operations on Files and Directories**

**Create:**

* touch filename.txt — Creates an empty file.
* mkdir myfolder — Creates a new directory.

**Read:**

* cat filename.txt — Displays the contents of a file.
* less filename.txt — Allows you to scroll through a file.

**Update:**

* nano filename.txt — Opens a file in the Nano text editor.
* echo "Text" &gt;&gt; filename.txt — Appends text to a file.

**Delete:**

* rm filename.txt — Removes a file.
* rmdir myfolder — Removes an empty directory.
* rm -r myfolder — Removes a directory and its contents recursively.

**Tips:**

* Be cautious with rm commands; deleted files can't be easily recovered.
* For more examples, search "Basic file operations in Linux" on Google or ask ChatGPT.

**3. File Permissions and**chmod

**Understanding Permissions:**

* Use ls -l to view file permissions.
* Permissions are shown as -rwxrwxrwx, divided into User (u), Group (g), and Others (o).
  + r — Read (4)
  + w — Write (2)
  + x — Execute (1)

**Changing Permissions:**

* Symbolic method:
  + chmod u+x filename — Adds execute permission for the user.
  + chmod g-w filename — Removes write permission from the group.
* Numeric method:
  + chmod 755 filename — Sets permissions to rwxr-xr-x.

**Tips:**

* Practice changing permissions and observe the effects.
* Online resources like "Understanding chmod in Linux" can be very helpful.

**4. Using**sudo**for Superuser Privileges**

**Basics:**

* sudo [command] — Runs a command with superuser privileges.
* Example: sudo apt update — Updates package lists.

**Tips:**

* Use sudo carefully to avoid making unintended system changes.
* If you need clarification, ask ChatGPT or search "How to use sudo in Linux."

**Practice Exercises**

1. **Navigation and File Management:**
   * Create a directory named project and navigate into it.
   * Inside project, create a file called notes.txt.
   * Go back to your home directory.
2. **File Editing and Viewing:**
   * Add the text "Hello, Linux!" to notes.txt using echo.
   * View the contents using cat and less.
3. **Permission Changes:**
   * Make notes.txt executable.
   * Verify the permission changes with ls -l.
4. **Using**sudo**:**
   * Attempt to create a file in /root using sudo.
   * Update your system's package list with sudo apt update.

**Additional Resources**

* **Google Searches:**
  + Use specific queries like "How to change directory in Linux" for quick answers.
* **ChatGPT:**
  + Ask detailed questions for explanations or troubleshooting.
* **Manual Pages:**
  + Type man [command] in the terminal for the manual.
* **Online Tutorials:**

**-**

**Preparation Tips:**

* **Hands-On Practice:** The more you use these commands, the more comfortable you'll become.
* **Understand the 'Why':** Don't just memorize commands—understand what they do.
* **Ask for Help:** If you're stuck, use the resources above or reach out to me during class time and work periods.

danielazuonye@Daniel:~$ su -

Password: Azuonye

root@Daniel:~# usermod -aG sudo danielazuonye

root@Daniel:~# su - danielazuonye

danielazuonye@Daniel:~$ sudo apt update

[sudo] password for danielazuonye: Azuonye

Hit:1 http://deb.debian.org/debian bookworm InRelease



Hit:2 http://security.debian.org/debian-security bookworm-security InRelease

Hit:3 http://deb.debian.org/debian bookworm-updates InRelease

Reading package lists... Done

Building dependency tree... Done

Reading state information... Done

All packages are up to date.

**GitHub Commands**

Summary of Git Commands Used:

• Initialize Repository: bash - [git init]

• Configure Git: bash - [git config --global user.name "Your Name"] bash - [git config --global user.email "youremail@example.com"]

• Check Status: bash - [git status]

• Stage Files: bash - [git add filename] bash - [git add .]

• Commit Changes: bash - [git commit -m "message"]

• View Log: bash - [git log]

• Create and Switch Branch: bash - [git checkout -b branch-name]

• Switch Branch: bash - [git checkout branch-name]

• Merge Branches: bash - [git merge branch-name]

• Delete Branch: bash - [git branch -d branch-name]

• Undo Changes in Working Directory: o Using git restore (newer versions): bash - [git restore filename] o Using git checkout (older versions): bash - [git checkout -- filename]

• Remove Cached File: bash - [git rm --cached filename]

• Set Alias: bash - [git config --global alias.alias-name command]

• Tag a Commit: bash - [git tag -a v1.0 -m "message"]

• Reset Commits: bash - [git reset --soft HEAD~1] bash - [git reset --hard HEAD~1]

Top of Form

Bottom of Form