**E-Portfolio Linux Activity: Demonstrating Basic Linux Skills**

**1.1. Log in to the Horizon VMware:**

* Open VMware Horizon Client on your local machine.
* Enter the server address or hostname provided by your ITCS (Information Technology and Computer Services) department (e.g., horizon8.cet.ecu.edu).
* Log in using your ECU credentials.

**1.2. Launch Kali Linux:**

* Locate the Kali Linux virtual machine (VM) within the VMware Horizon Client interface (e.g., MIS 4123).
* Double-click on the Kali Linux VM to start it.

**1.3. Log in to Kali Linux:**

* Wait for the Kali Linux VM to load.
* Use the default login credentials (username: kali, password: kali) to log in.

**Preliminary Task 2: Create Student Folders**

**2.1. Open the Terminal (the black box immediately to the right of the Firefox browser icon):**

**Press Ctrl + Alt + T** to open a terminal window. You'll use this terminal to execute commands.

**2.2. Create Your Student Folder:**

2.2.1 Start a Root Shell Session by entering the following command and providing your user password when prompted. Root Shell Sessions should be used sparingly because some actions can’t be undone. This task is included to give exposure to the root shell (notice it is in **red** to highlight that you aren’t in the user shell):

**sudo -i**

This command will start a new shell session as the root user. You won't need to use **sudo** before commands within this session because you are already logged in as root.

A computer screen shot of a computer

Description automatically generated

Figure 1. Root Shell Session.

A screenshot of a computer

Description automatically generated

Figure 2. User Shell Session.

***SAMPLE Artifact for* E-Portfolio Linux Activity (blended with some additional instructions/troubleshooting tips from me.): Demonstrating Basic Linux Skills. You will write something similar (your own words so that your artifact is distinct and personalized from other classmates).**

**Scenario: XYZ Financial Services, Inc.**

**Objective:** The objective of this e-portfolio activity is to assess my basic Linux skills in the context of cybersecurity and create a portfolio that showcases my proficiency. This not only allows me to demonstrate my capabilities but also fosters curiosity about the intersections between Linux and cybersecurity.

**Preparation:**

To begin, I set up a Kali Linux environment, a crucial step in mastering Linux. I organized my work by creating the following structure within my personal folder:

* **Documentation:** A place for textual documentation and notes, akin to maintaining a logbook during cybersecurity investigations.
* **Scripts:** A dedicated space for my Linux shell scripts, reminiscent of scripting in cybersecurity for automating tasks.
* **Screenshots:** A repository for screenshots, analogous to capturing evidence in cybersecurity incidents.

**Task 1: User Management**

**Objective:** Create a new user account, set a password, and assign it to a specific group.

**Steps:**

1. **Create a new user account with the name "cyberstudent" and set the password to "secure123."**

cd /home/your\_username/Scripts

nano create\_user.sh

This will open a new, empty file named "create\_user.sh" in the nano text editor.

A screenshot of a computer

Description automatically generated

Figure 3. Error message when the file already exists.

Copy and paste the following command (and comments) into the "create\_user.sh" file (select, right-click, copy AND THEN in the nano text editor, right-click and choose Paste Selection. ***Troubleshooting:*** *Do not copy and paste when the document is in the Canvas browser as it will give you unexplained errors. Open the file as a regular word document and then copy all the text in gray below. You can always type these commands and comments into the text editor. However, copying and pasting tried-and-true commands will limit your errors.* *Make sure that there is a # in front of each line that is a comment. If you have white text in the nano text editor, it symbolizes a command. Anything in white that is not a command will give you errors. You should only have one line below that is white (the sudo command line). The rest of the lines are comments and shouldn’t be displayed as white.*

**sudo useradd -m -d /home/cyberstudent -p $(openssl passwd -1 secure123) cyberstudent**

**Why This Matters:**

1. Create a "cybersecurity" group if it doesn't exist.

Open a text editor on your Linux system. **Make sure you are in the /home/your-username/Scripts folder (if you are not there, type cd /home/your\_username/Scripts before entering nano create\_group.sh command).** You can use the built-in text editor called "nano" if it's available by entering the following command in your terminal:

nano create\_group.sh

This will open a new, empty file named "create\_group.sh" in the nano text editor.

#!/bin/bash

# Create the "cybersecurity" group if it doesn't exist.

sudo groupadd cybersecurity

**Command Explanation:**

* + **groupadd**: This command is used to create a new group.
  + **cybersecurity**: This is the name of the group we are creating.

**Why This Matters:**

* + Groups are used to organize users and control their access to resources. In cybersecurity, organizing users into appropriate groups helps manage access permissions effectively. It's analogous to categorizing personnel into different roles within a secure facility.

1. Add this user to the “cybersecurity” group.

**Make sure you are in the /home/your-username/Scripts folder (if you are not there, type cd /home/your\_username/Scripts before entering nano create\_group.sh command).** Then, open a text editor on your Linux system. You can use the built-in text editor called "nano" if it's available by entering the following command in your terminal:

nano add\_to\_group.sh

This will open a new, empty file named "add\_to\_group.sh" in the nano text editor.

sudo usermod -aG cybersecurity cyberstudent

**Command Explanation:**

* + usermod: This command is used to modify user account properties.
  + -aG cybersecurity: This option adds the user to the "cybersecurity" group.
  + cyberstudent: This is the username of the user we want to modify.

**Task 2: File Operations**

**Objective:** Perform basic file operations in the Linux command line.

**Steps:**

1. Create a new text file named "important.txt" in your home directory.

**Make sure you are in the /home/your-username/Scripts folder (if you are not there, type cd /home/your\_username/Scripts before entering nano create\_group.sh command).** Then, open a text editor on your Linux system. You can use the built-in text editor called "nano" if it's available by entering the following command in your terminal:

nano create\_important\_file.sh

This will open a new, empty file named "create\_important\_file.sh" in the nano text editor.

#!/bin/bash

# This script creates a new text file named "important.txt" in the home directory.

# Create "important.txt" in the home directory.

touch ~/important.txt

# Save this script as "create\_important\_file.sh" in /home/your\_username/Scripts.

# To save, use the following command in the text editor:

# **Ctrl + O, then accept the filename by pressing “Enter” since you are already in the folder you want to save it in. After pressing “Enter,” you should see “xxx lines saved” or some similar message.**

**# Exit nano by pressing Ctrl + X.**

**Command Explanation:**

* + touch: This command is used to create a new file. In this case, we are creating a file named "important.txt."
  + ~/important.txt: The ~/ represents your home directory, so the file is created there.

**Why This Matters:**

* + Understanding how to create files is fundamental for working with data and configurations in Linux. It's analogous to creating a new document in a word processor; it's the first step in managing data.

1. Copy "important.txt" to a subdirectory called "confidential."

**Make sure you are in the /home/your-username/Scripts folder (if you are not there, type cd /home/your\_username/Scripts before entering nano create\_group.sh command).** Then, open a text editor on your Linux system. You can use the built-in text editor called "nano" if it's available by entering the following command in your terminal:

nano copy\_important\_file.sh

This will open a new, empty file named "copy\_important\_file.sh" in the nano text editor.

#!/bin/bash

# This script copies "important.txt" to a subdirectory called "confidential."

# Create the "confidential" directory if it doesn't exist.

mkdir -p ~/confidential

# Copy "important.txt" to the "confidential" directory.

cp ~/important.txt ~/confidential/

# Save this script as "copy\_important\_file.sh" in /home/your\_username/Scripts.

# To save, use the following command in the text editor:

# **Ctrl + O, then accept the filename by pressing “Enter” since you are already in the folder you want to save it in. After pressing “Enter,” you should see “xxx lines saved” or some similar message.**

**# Exit nano by pressing Ctrl + X.**

**Command Explanation:**

* + **mkdir**: This command is used to create a new directory or folder. Here, we are creating a directory named "confidential."
  + **cp**: The **cp** command is used for copying files or directories.
  + **~/important.txt**: This is the source file we want to copy.
  + **~/confidential/**: This is the destination directory where we want to copy the file.

**Why This Matters:**

* + Creating directories and copying files are essential file management tasks in Linux. It's akin to organizing your files into folders on your computer. In cybersecurity, proper file organization ensures data integrity and security.

1. Rename "important.txt" to "classified.txt."

**Make sure you are in the /home/your-username/Scripts folder (if you are not there, type cd /home/your\_username/Scripts before entering nano create\_group.sh command).** Then, open a text editor on your Linux system. You can use the built-in text editor called "nano" if it's available by entering the following command in your terminal:

nano rename\_file.sh

This will open a new, empty file named "rename\_file.sh" in the nano text editor.

#!/bin/bash

# This script renames "important.txt" to "classified.txt."

# Rename "important.txt" to "classified.txt."

mv ~/important.txt ~/classified.txt

# Save this script as "rename\_file.sh" in /home/your\_username/Scripts.

# To save, use the following command cin the text editor:

# **Ctrl + O, then accept the filename by pressing “Enter” since you are already in the folder you want to save it in. After pressing “Enter,” you should see “xxx lines saved” or some similar message.**

**# Exit nano by pressing Ctrl + X.**

**Command Explanation:**

* + mv: This command is used to rename or move files or directories.
  + ~/important.txt: This is the current name of the file we want to rename.
  + ~/classified.txt: This is the new name we want to assign to the file.

**Why This Matters:**

* + Renaming files is a common operation when working with data. In cybersecurity, this can be important for maintaining confidentiality and managing sensitive information. It's like changing the label on a file to indicate its new status.

**File Operations - Why This Matters Reflection:**

File operations are the lifeblood of data management, much like managing sensitive documents in a secure file cabinet. As I performed these operations in Linux, I couldn't help but draw parallels to cloud storage and synchronization tools in the cybersecurity toolkit. Curiosity led me to contemplate how these fundamental Linux file operations relate to the broader cybersecurity ecosystem, especially when it comes to secure data handling.

**Task 3: Permissions and Ownership**

**Objective:** Manage file permissions and ownership.

**Steps:**

1. Create a file named "topsecret.txt" in your home directory.

**Make sure you are in the /home/your-username/Scripts folder (if you are not there, type cd /home/your\_username/Scripts before entering nano create\_group.sh command).** Then, open a text editor on your Linux system. You can use the built-in text editor called "nano" if it's available by entering the following command in your terminal:

nano create\_topsecret\_file.sh

This will open a new, empty file named "create\_topsecret\_file.sh" in the nano text editor.

#!/bin/bash

# This script creates a new file named "topsecret.txt" in the top directory.

# Create "topsecret.txt" in the home directory.

touch ~/topsecret.txt

# Save this script as "create\_topsecret\_file.sh" in /home/your\_username/Scripts.

# To save, use the following command in the text editor:

# **Ctrl + O, then accept the filename by pressing “Enter” since you are already in the folder you want to save it in. After pressing “Enter,” you should see “xxx lines saved” or some similar message.**

**# Exit nano by pressing Ctrl + X.**

**Command Explanation:**

* + **touch**: This command is used to create a new file. In this case, it creates a file named "topsecret.txt."
  + **~/topsecret.txt**: The **~/** represents your home directory, so the file is created there.

**Why This Matters:**

* + Creating files is a fundamental task in data management. It's analogous to creating a new document or file on your computer. In cybersecurity, it's essential to know how to create and secure files to protect sensitive information.

1. Change the ownership of "topsecret.txt" to the "root" user.

**Make sure you are in the /home/your-username/Scripts folder (if you are not there, type cd /home/your\_username/Scripts before entering nano create\_group.sh command).** Then, open a text editor on your Linux system. You can use the built-in text editor called "nano" if it's available by entering the following command in your terminal:

nano change\_ownership.sh

This will open a new, empty file named "change\_ownership.sh" in the nano text editor.

#!/bin/bash

# This script changes the ownership of "topsecret.txt" to the "root" user.

# Change ownership to the "root" user.

sudo chown root ~/topsecret.txt

# Save this script as "change\_ownership.sh" in /home/your\_username/Scripts.

# To save, use the following command in the text editor:

# **Ctrl + O, then accept the filename by pressing “Enter” since you are already in the folder you want to save it in. After pressing “Enter,” you should see “xxx lines saved” or some similar message.**

**# Exit nano by pressing Ctrl + X.**

**Command Explanation:**

* + **sudo**: This command is used to execute the following command with superuser privileges, which are necessary to change ownership.
  + **chown**: This command is used to change the ownership of files or directories.
  + **root**: This specifies the new owner (in this case, the "root" user).
  + **~/topsecret.txt**: This is the file whose ownership is being changed.

**Why This Matters:**

* + Changing ownership is a key aspect of controlling access to files. In cybersecurity, managing file ownership ensures that only authorized users have specific privileges. It's akin to transferring ownership of a classified document to a security officer.

1. Deny all permissions to the group and others for "topsecret.txt."

**Make sure you are in the /home/your-username/Scripts folder (if you are not there, type cd /home/your\_username/Scripts before entering nano create\_group.sh command).** Then, open a text editor on your Linux system. You can use the built-in text editor called "nano" if it's available by entering the following command in your terminal:

nano deny\_permissions.sh

This will open a new, empty file named "deny\_permissions.sh" in the nano text editor.

#!/bin/bash

# This script denies all permissions to the group and others for "topsecret.txt."

# Remove read, write, and execute permissions for group and others.

sudo chmod go-rwx ~/topsecret.txt

# Save this script as "deny\_permissions.sh" in /home/your\_username/Scripts.

# To save, use the following command in the text editor:

# **Ctrl + O, then accept the filename by pressing “Enter” since you are already in the folder you want to save it in. After pressing “Enter,” you should see “xxx lines saved” or some similar message.**

**# Exit nano by pressing Ctrl + X.**

**Command Explanation:**

* + **chmod**: This command is used to change file permissions.
  + **go-rwx**: This specifies the permissions to be set. In this case, "go" represents the group and others, and "rwx" signifies that read, write, and execute permissions are being removed.
  + **~/topsecret.txt**: This is the file for which permissions are being modified.

**Why This Matters:**

* + Managing file permissions is crucial for controlling who can access, read, modify, or execute a file. In cybersecurity, fine-grained control over permissions is akin to setting up security protocols to restrict access to classified information. Understanding these concepts is vital for safeguarding sensitive files and directories.

**Permissions and Ownership - Why This Matters Reflection:**

File permissions and ownership are the guardians of data integrity, reminiscent of controlling access to a classified document. In cybersecurity, we often deal with access control lists (ACLs) and permissions. I found myself intrigued by how Linux's approach compared to Windows ACLs. Exploring these connections piqued my curiosity, especially regarding strategies for securing sensitive files and directories.

Understanding Linux file permissions and ownership not only enhances your Linux skills but also provides the foundation for data security, a critical component of cybersecurity. It's like mastering the protocols and controls to secure a vault containing classified documents, ensuring that only authorized personnel have access.

**Task 4: Process Management**

**Objective:** Manage processes and understand system resource usage.

**Steps:**

1. Start a background process named "myprocess" that runs the "top" command.

**Make sure you are in the /home/your-username/Scripts folder (if you are not there, type cd /home/your\_username/Scripts before entering nano create\_group.sh command).** Then, open a text editor on your Linux system. You can use the built-in text editor called "nano" if it's available by entering the following command in your terminal:

nano start\_myprocess.sh

This will open a new, empty file named "start\_myprocess.sh" in the nano text editor.

#!/bin/bash

# This script starts a background process named "myprocess" running the "top" command.

# Start "myprocess" as a background process.

top > /dev/null 2>&1 &

# Save this script as "start\_myprocess.sh" in /home/your\_username/Scripts.

# To save, use the following command in the text editor:

# **Ctrl + O, then accept the filename by pressing “Enter” since you are already in the folder you want to save it in. After pressing “Enter,” you should see “xxx lines saved” or some similar message.**

**# Exit nano by pressing Ctrl + X.**

**Command Explanation:**

* + **top**: This is a command-line tool for monitoring system processes and resource usage.
  + **> /dev/null**: This part redirects the standard output of the **top** command to **/dev/null**, effectively discarding the output.
  + **2>&1**: This redirects the standard error (file descriptor 2) to the same location as the standard output (file descriptor 1), which is **/dev/null**.
  + **&**: This symbol at the end of the command makes it run in the background.

**Why This Matters:**

* + Starting background processes is a common practice in Linux, especially when you want to run tasks without tying up your terminal. In cybersecurity, background processes are similar to running security tools and monitoring agents that continuously observe system behavior.

1. Find and display the process ID (PID) of "myprocess."

**Make sure you are in the /home/your-username/Scripts folder (if you are not there, type cd /home/your\_username/Scripts before entering nano create\_group.sh command).** Then, open a text editor on your Linux system. You can use the built-in text editor called "nano" if it's available by entering the following command in your terminal:

nano find\_myprocess\_pid.sh

This will open a new, empty file named "find\_myprocess\_pid.sh" in the nano text editor.

#!/bin/bash

# This script finds and displays the process ID (PID) of "myprocess."

# Find the PID of "myprocess."

pgrep -x top

# Save this script as "find\_myprocess\_pid.sh" in /home/your\_username/Scripts.

# To save, use the following command in the text editor:

# **Ctrl + O, then accept the filename by pressing “Enter” since you are already in the folder you want to save it in. After pressing “Enter,” you should see “xxx lines saved” or some similar message.**

**# Exit nano by pressing Ctrl + X.**

**Command Explanation:**

* + **pgrep**: This command is used to search for processes by their names.
  + **-x top**: The **-x** option ensures an exact match for the process name "top."

**Why This Matters:**

* + Identifying and tracking processes is essential for system management and troubleshooting. In cybersecurity, understanding process management helps in monitoring for suspicious activities and potential threats.

1. Terminate the "myprocess" background process.

**Make sure you are in the /home/your-username/Scripts folder (if you are not there, type cd /home/your\_username/Scripts before entering nano create\_group.sh command).** Then, open a text editor on your Linux system. You can use the built-in text editor called "nano" if it's available by entering the following command in your terminal:

nano terminate\_myprocess.sh

This will open a new, empty file named "terminate\_myprocess.sh" in the nano text editor.

#!/bin/bash

# This script terminates the "myprocess" background process.

# Terminate "myprocess."

pkill -x top

# Save this script as "terminate\_myprocess.sh" in /home/your\_username/Scripts.

# To save, use the following command in the text editor:

# **Ctrl + O, then accept the filename by pressing “Enter” since you are already in the folder you want to save it in. After pressing “Enter,” you should see “xxx lines saved” or some similar message.**

**# Exit nano by pressing Ctrl + X.**

**Command Explanation:**

* + **pkill**: This command is used to signal processes based on their names.
  + **-x top**: The **-x** option specifies that we want to target the "top" process.

**Why This Matters:**

* + Stopping or terminating processes is crucial for system stability and security. In cybersecurity, terminating malicious processes or unauthorized activities is a critical aspect of incident response and threat mitigation.

**Process Management - Why This Matters Reflection:**

Process management is akin to overseeing a dynamic workforce. In cybersecurity, we use intrusion detection systems (IDS) and endpoint detection and response (EDR) tools to monitor processes. As I managed processes in Linux, I pondered the similarities with managing security tools in a cybersecurity operation center. Curiosity drove me to explore how these skills could be leveraged in real-world security scenarios.

Understanding Linux process management not only enhances your Linux skills but also provides a foundation for managing and monitoring processes, a vital aspect of cybersecurity. It's like learning how to manage a team of security analysts who monitor and respond to threats in a security operation center, ensuring the overall security of a system or network.

**Task 5: System Information**

**Objective:** Retrieve system information and hardware details.

**Steps:**

1. Display the system hostname.

**Make sure you are in the /home/your-username/Scripts folder (if you are not there, type cd /home/your\_username/Scripts before entering nano create\_group.sh command).** Then, open a text editor on your Linux system. You can use the built-in text editor called "nano" if it's available by entering the following command in your terminal:

nano display\_hostname.sh

This will open a new, empty file named "display\_hostname.sh" in the nano text editor.

#!/bin/bash

# This script displays the system hostname.

# Display the system hostname.

hostname

# Save this script as "display\_hostname.sh" in /home/your\_username/Scripts.

# To save, use the following command in the text editor:

# **Ctrl + O, then accept the filename by pressing “Enter” since you are already in the folder you want to save it in. After pressing “Enter,” you should see “xxx lines saved” or some similar message.**

**# Exit nano by pressing Ctrl + X.**

**Command Explanation:**

* + **hostname**: This command is used to display the system's hostname, which is its unique identifier on a network.

**Why This Matters:**

* + Knowing the system's hostname is crucial for network administration and system identification. In cybersecurity, understanding system identification is akin to recognizing specific devices on a network, which is essential for security monitoring and incident response.

1. List the CPU information.

**Make sure you are in the /home/your-username/Scripts folder (if you are not there, type cd /home/your\_username/Scripts before entering nano create\_group.sh command).** Then, open a text editor on your Linux system. You can use the built-in text editor called "nano" if it's available by entering the following command in your terminal:

nano list\_cpu\_info.sh

This will open a new, empty file named "list\_cpu\_info.sh" in the nano text editor.

lscpu

**Running the Scripts that you Built for each of these Tasks.**

**Task 1: User Management**

**Step 1: Create a new user account with the name "cyberstudent" and set the password to "secure123." Save this as a script.**

1. Build the script as described in the previous instructions.
2. Open a Terminal on your Linux system.
3. Navigate to the Scripts folder where you saved the script:

cd /home/your\_username/Scripts

Replace **your\_username** with your actual username.

1. Make the script executable using **sudo**:

sudo chmod +x create\_user.sh

1. You may be prompted to enter your password since **sudo** requires administrative privileges.
2. Run the script by entering:

./create\_user.sh

1. Follow any on-screen instructions or prompts provided by the script.

**Step 2a: Create the cybersecurity group and then add the user to the "cybersecurity" group. Save this as a script.**

1. Build the script as described in the previous instructions.
2. Open a Terminal on your Linux system.
3. Navigate to the Scripts folder where you saved the script:

cd /home/your\_username/Scripts

Replace **your\_username** with your actual username.

1. Make the script executable using **sudo**:

sudo chmod +x create\_group.sh

1. You may be prompted to enter your password since **sudo** requires administrative privileges.
2. Run the script by entering:

./create\_group.sh

1. Follow any on-screen instructions or prompts provided by the script.

**Step 2b: Add this user to the "cybersecurity" group. Save this as a script.**

1. Build the script as described in the previous instructions.
2. Open a Terminal on your Linux system.
3. Navigate to the Scripts folder where you saved the script:

cd /home/your\_username/Scripts

Replace **your\_username** with your actual username.

1. Make the script executable using **sudo**:

sudo chmod +x add\_to\_group.sh

1. You may be prompted to enter your password since **sudo** requires administrative privileges.
2. Run the script by entering:

./add\_to\_group.sh

1. Follow any on-screen instructions or prompts provided by the script.

**Task 2: File Operations**

**Step 1: Create a new text file named "important.txt" in your home directory. Save this as a script.**

1. Build the script as described in the previous instructions.
2. Open a Terminal on your Linux system.
3. Navigate to the Scripts folder where you saved the script:

cd /home/your\_username/Scripts

Replace **your\_username** with your actual username.

1. Make the script executable using **sudo**:

sudo chmod +x create\_important\_file.sh

1. You may be prompted to enter your password since **sudo** requires administrative privileges.
2. Run the script by entering:

./create\_important\_file.sh

**Step 2: Copy "important.txt" to a subdirectory called "confidential." Save this as a script.**

1. Build the script as described in the previous instructions.
2. Open a Terminal on your Linux system.
3. Navigate to the Scripts folder where you saved the script:

cd /home/your\_username/Scripts

Replace **your\_username** with your actual username.

1. Make the script executable using **sudo**:

sudo chmod +x copy\_important\_file.sh

1. You may be prompted to enter your password since **sudo** requires administrative privileges.
2. Run the script by entering:

./copy\_important\_file.sh

**Step 3: Rename "important.txt" to "classified.txt." Save this as a script.**

1. Build the script as described in the previous instructions.
2. Open a Terminal on your Linux system.
3. Navigate to the Scripts folder where you saved the script:

cd /home/your\_username/Scripts

Replace **your\_username** with your actual username.

1. Make the script executable using **sudo**:

sudo chmod +x rename\_file.sh

1. You may be prompted to enter your password since **sudo** requires administrative privileges.
2. Run the script by entering:

./rename\_file.sh

**Task 3: Permissions and Ownership**

**Step 1: Create a file named "topsecret.txt" in your top directory. Save this as a script.**

1. Build the script as described in the previous instructions.
2. Open a Terminal on your Linux system.
3. Navigate to the Scripts folder where you saved the script:

cd /home/your\_username/Scripts

Replace **your\_username** with your actual username.

1. Make the script executable using **sudo**:

sudo chmod +x create\_topsecret\_file.sh

1. You may be prompted to enter your password since **sudo** requires administrative privileges.
2. Run the script by entering:

./create\_topsecret\_file.sh

**Step 2: Change the ownership of "topsecret.txt" to the "root" user. Save this as a script.**

1. Build the script as described in the previous instructions.
2. Open a Terminal on your Linux system. Log into the Root User account.
3. Navigate to the Scripts folder where you saved the script:

cd /home/your\_username/Scripts

Replace **your\_username** with your actual username.

1. Make the script executable using **sudo**:

sudo chmod +x change\_ownership.sh

1. You may be prompted to enter your password since **sudo** requires administrative privileges.
2. Run the script by entering:

./change\_ownership.sh

**Step 3: Deny all permissions to the group and others for "topsecret.txt." Save this as a script.**

1. Build the script as described in the previous instructions.
2. Open a Terminal on your Linux system.
3. Navigate to the Scripts folder where you saved the script:

cd /home/your\_username/Scripts

Replace **your\_username** with your actual username.

1. Make the script executable using **sudo**:

sudo chmod +x deny\_permissions.sh

1. You may be prompted to enter your password since **sudo** requires administrative privileges.
2. Run the script by entering:

./deny\_permissions.sh

**Task 4: Process Management**

**Step 1: Start a background process named "myprocess" that runs the "top" command. Save this as a script.**

1. Build the script as described in the previous instructions.
2. Open a Terminal on your Linux system.
3. Navigate to the Scripts folder where you saved the script:

cd /home/your\_username/Scripts

Replace **your\_username** with your actual username.

1. Make the script executable using **sudo**:

sudo chmod +x start\_myprocess.sh

1. You may be prompted to enter your password since **sudo** requires administrative privileges.
2. Run the script by entering:

./start\_myprocess.sh

**Step 2: Find and display the process ID (PID) of "myprocess." Save this as a script.**

1. Build the script as described in the previous instructions.
2. Open a Terminal on your Linux system.
3. Navigate to the Scripts folder where you saved the script:

cd /home/your\_username/Scripts

Replace **your\_username** with your actual username.

1. Make the script executable using **sudo**:

sudo chmod +x find\_myprocess\_pid.sh

1. You may be prompted to enter your password since **sudo** requires administrative privileges.
2. Run the script by entering:

./find\_myprocess\_pid.sh

**Step 3: Terminate the "myprocess" background process. Save this as a script.**

1. Build the script as described in the previous instructions.
2. Open a Terminal on your Linux system.
3. Navigate to the Scripts folder where you saved the script:

cd /home/your\_username/Scripts

Replace **your\_username** with your actual username.

1. Make the script executable using **sudo**:

sudo chmod +x terminate\_myprocess.sh

1. You may be prompted to enter your password since **sudo** requires administrative privileges.
2. Run the script by entering:

./terminate\_myprocess.sh

**Task 5: System Information**

**Step 1: Display the system hostname. Save this as a script.**

1. Build the script as described in the previous instructions.
2. Open a Terminal on your Linux system.
3. Navigate to the Scripts folder where you saved the script:

cd /home/your\_username/Scripts

Replace **your\_username** with your actual username.

1. Make the script executable using **sudo**:

sudo chmod +x display\_hostname.sh

1. You may be prompted to enter your password since **sudo** requires administrative privileges.
2. Run the script by entering:

./display\_hostname.sh

**Step 2: List the CPU information. Save this as a script.**

1. Build the script as described in the previous instructions.
2. Open a Terminal on your Linux system.
3. Navigate to the Scripts folder where you saved the script:

cd /home/your\_username/Scripts

Replace **your\_username** with your actual username.

1. Make the script executable using **sudo**:

sudo chmod +x list\_cpu\_info.sh

1. You may be prompted to enter your password since **sudo** requires administrative privileges.
2. Run the script by entering:

./list\_cpu\_info.sh

**Step 3: List the available disk space. Save this as a script.**

1. Build the script as described in the previous instructions.
2. Open a Terminal on your Linux system.
3. Navigate to the Scripts folder where you saved the script:

cd /home/your\_username/Scripts

Replace **your\_username** with your actual username.

1. Make the script executable using **sudo**:

sudo chmod +x list\_disk\_space.sh

1. You may be prompted to enter your password since **sudo** requires administrative privileges.
2. Run the script by entering:

./list\_disk\_space.sh

Finally, come up with a few other commands that you can incorporate into your artifact that helps you distinguish your work from others in the class. Between these additional commands, however you layout your reflection statements, and your actual reflection will make it clear that this is your work and reflection rather than a “forced, required” class activity. Again, make it clear you are interested in these skills and how they relate to your work and how it builds on your previous tech knowledge.

**More commands/options that add more power to your commands.**

* 1. **ls -R** If you want to see the entire directory, you can use the **ls -R** command (see screenshot below; your file structure could be different).

