

## Day 35

# Exploitation Analyst

### User Management and PAM:

#### Sudo Access:

##### What is sudo?

sudo (short for “superuser do”) is a Linux command that lets a permitted user run programs with the security privileges of another user, typically the root user. It allows users to perform administrative tasks without needing to log in as root.

##### Why is it required?

- **Security:** Instead of sharing the root password, users get limited administrative access.
- **Accountability:** Commands run via sudo are logged, helping track who did what.
- **Convenience:** Users can run specific commands with elevated privileges without switching users.

##### Disadvantages

- **Misconfiguration risks:** Incorrect sudoers file settings can give excessive privileges.
- **Potential abuse:** If a sudo user’s account is compromised, attacker gets root-level access.
- **Complexity:** Managing fine-grained sudo permissions requires care and knowledge.

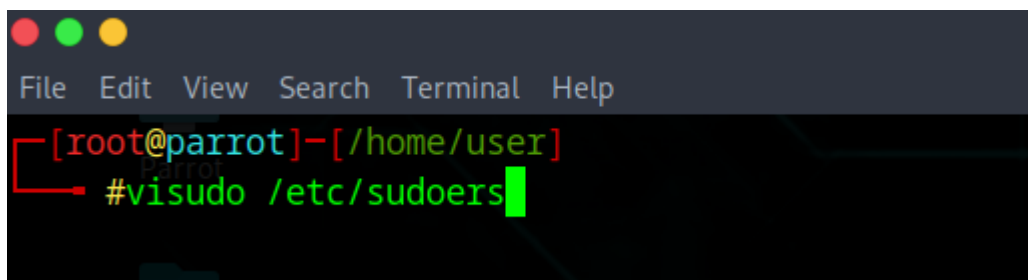
##### Which file allows sudo users?

- The `/etc/sudoers` file controls who can use sudo and what commands they can run.
- It should always be edited with visudo to prevent syntax errors.

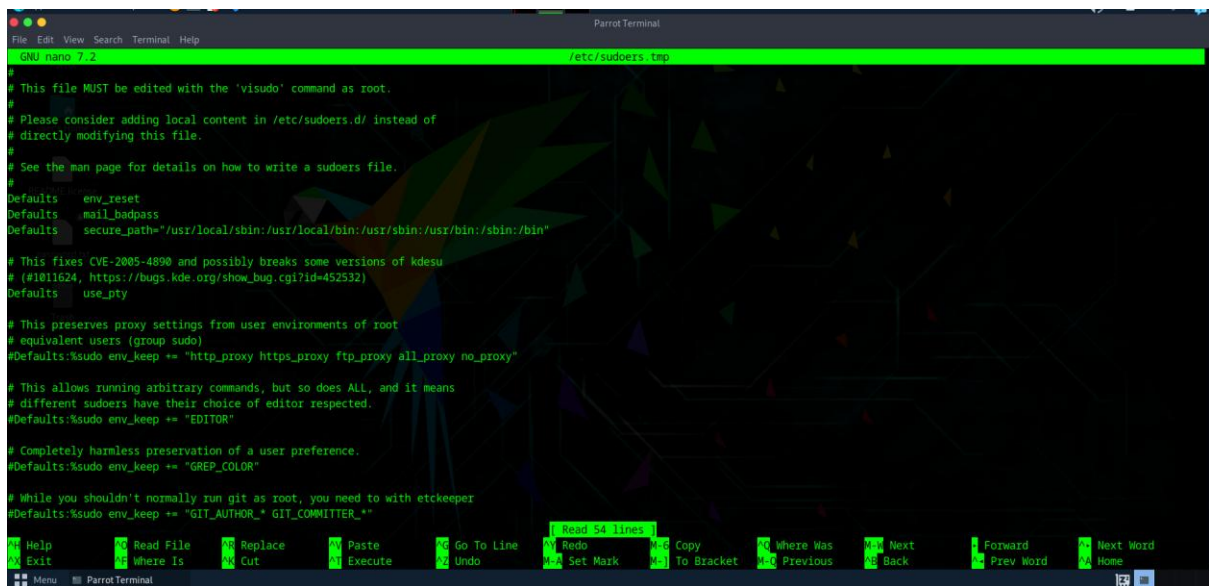
#### Setting Sudo Access rules:

Steps:

Open the `/etc/sudoers` with the visudo editor:

A screenshot of a terminal window with a dark background. At the top, there are three colored window control buttons (red, green, yellow) and a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The terminal prompt shows the user is root@parrot in the directory /home/user. The command '#visudo /etc/sudoers' has been entered, and a green cursor is at the end of the line.

Following screen will appear:



```
GNU nano 2.2.2 /etc/sudoers.tmp
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults    env_reset
Defaults    mail_badpass
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

# This fixes CVE-2005-4890 and possibly breaks some versions of kdesu
# (#1011624, https://bugs.kde.org/show_bug.cgi?id=452532)
Defaults    use_pty

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

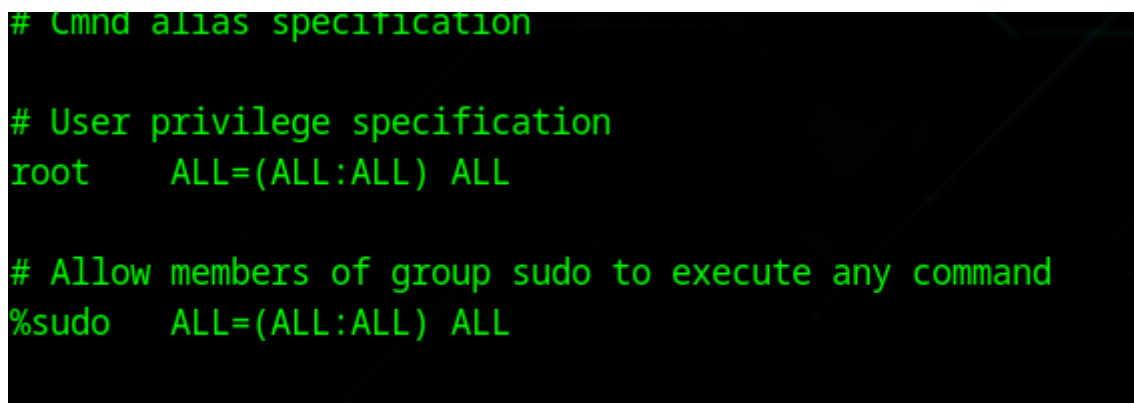
# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

[Read 54 lines]
Help      Read File  Replace  Paste  Go To Line  Redo  Copy  Where Was  Next  Forward  Next Word
Exit      Where Is  Cut      Execute Undo  Set Mark  To Bracket Previous Back Prev Word Home
```

Scroll down and reach this section:



```
# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL
```

This line in /etc/sudoers:

```
root    ALL=(ALL:ALL) ALL
```

means that the **root user** has full administrative privileges on the system. Breaking it down:

- root — the username this rule applies to.
- ALL (first) — root can run commands from any host (useful in networked setups).
- (ALL:ALL) — root can run commands as any user and any group.
- ALL (last) — root can run any command.

What can we do here?

To give user aditya permission to restart the apache2 service using sudo without giving full root access, you can add a rule in the sudoers file like this:

```
aditya ALL=NOPASSWD: /bin/systemctl restart apache2
```

```
# User privilege specification
root    ALL=(ALL:ALL) ALL

aditya  ALL=NOPASSWD: /bin/systemctl restart apache2
█
```

Save the file and exit.