Lab Project - 5

Objective: Linux Sudo Access

PRE-REQUISITES:

Oracle VirtualBox or VMWare, Ubuntu installed.

Lab 1: Introduction to sudo

Objective:

• Understand how sudo works and gain basic experience using it.

Tasks:

1.Check for sudo Installation:

o Verify that sudo is installed on your system.

bash

Copy code

which sudo

o If it is not installed, you can install it using the following command:

#On Ubuntu/Debian:

bash

Copy code

sudo apt install sudo

```
root@DESKTOP-5K616C3:~# apt install sudo
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
sudo is already the newest version (1.9.9-1ubuntu2.4).
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
```

#On CentOS/RHEL:

bash

Copy code

sudo yum install sudo

```
Updating Subscription Management repositories.
Last metadata expiration check: 0:04:20 ago on Thu 27 Feb 2025 09:45:53 PM UTC.
Package sudo-1.9.5p2-10.el9_3.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
root@rhel:~#
```

2. Verify sudo Access for Current User:

- Check whether the current user has sudo privileges by running a command that requires superuser permissions, such as:

sudo whoami

```
root@rhel:~# sudo whoami
root
root@rhel:~#
```

3. Execute Commands with sudo:

- Run a simple system command with sudo to confirm access. For example, try updating the system package list:

bash

Copy code

sudo apt update # Ubuntu/Debian

```
root@DESKTOP-5K616C3:~# sudo apt update
Ign:1 http://archive.ubuntu.com/ubuntu jammy InRelease
Ign:2 http://security.ubuntu.com/ubuntu jammy-security InRelease
```

sudo yum update # CentOS/RHEL

```
  Upgrading        : selinux-policy-targeted-38.1.45-3.el9_5.noarch
  Running scriptlet: selinux-policy-targeted-38.1.45-3.el9_5.noarch
  Upgrading        : kmod-28-10.el9.x86_64
  Upgrading        : libmount-2.37.4-20.el9.x86_64
  Upgrading        : glib2-2.68.4-14.el9_4.1.x86_64
  Upgrading        : polkit-libs-0.117-13.el9.x86_64
  Running scriptlet: container-selinux-3:2.232.1-1.el9.noarch
  Upgrading        : container-selinux-3:2.232.1-1.el9.noarch
  Running scriptlet: container-selinux-3:2.232.1-1.el9.noarch
```

4. Exit the sudo Session:

- After running the command, exit the root session by simply typing exit or waiting for the session timeout.

Lab 2: Configuring Sudo Access

Objective:

• Learn how to configure sudo access for specific users by editing the sudoers file.

Tasks:

1. Open the Sudoers File Safely:

 o Use visudo to edit the sudoers file, which is the correct and safest way to modify sudo permissions.

bash

Copy code

sudo visudo

```
## Sudoers allows particular users to run various commands as
## the root user, without needing the root password.
##
## Examples are provided at the bottom of the file for collections
## of related commands, which can then be delegated out to particula
## users or groups.
##
## This file must be edited with the 'visudo' command.
```

2. Grant Sudo Access to a User:

 o Add a new user to the sudoers file by adding the following line under the user section:

bash

Copy code

username ALL=(ALL) ALL

```
## Sudoers allows particular users to run various commands as
## the root user, without needing the root password.
##
## Examples are provided at the bottom of the file for collections
## of related commands, which can then be delegated out to particula
## users or groups.
##
## This file must be edited with the 'visudo' command.
```

- Replace username with the actual username you want to grant sudo access to.

```
                                  the user from other groups
  -h, --help                      display this help message and exit
  -l, --login NEW_LOGIN           new value of the login name
  -L, --lock                      lock the user account
  -m, --move-home                 move contents of the home directory to the
                                  new location (use only with -d)
  -o, --non-unique                allow using duplicate (non-unique) UID
  -p, --password PASSWORD         use encrypted password for the new password
  -R, --root CHROOT_DIR           directory to chroot into
  -P, --prefix PREFIX_DIR         prefix directory where are located the /etc/* f
  -s, --shell SHELL               new login shell for the user account
  -u, --uid UID                   new UID for the user account
  -U, --unlock                    unlock the user account
  -v, --add-subuids FIRST-LAST    add range of subordinate uids
  -V, --del-subuids FIRST-LAST    remove range of subordinate uids
  -w, --add-subgids FIRST-LAST    add range of subordinate gids
  -W, --del-subgids FIRST-LAST    remove range of subordinate gids
  -Z, --selinux-user SEUSER       new SELinux user mapping for the user account
```

3. Grant Sudo Access to a Group:

```
root@rhel:~# sudo groupadd mygroup
root@rhel:~#
```

o To grant sudo access to all members of a specific group (e.g., admin or sudo), you can add:

bash

Copy code

%groupname ALL=(ALL) ALL

```
                                  the user from other groups
 -h, --help                       display this help message and exit
 -l, --login NEW_LOGIN            new value of the login name
 -L, --lock                       lock the user account
 -m, --move-home                  move contents of the home directory to the
                                  new location (use only with -d)
 -o, --non-unique                 allow using duplicate (non-unique) UID
 -p, --password PASSWORD          use encrypted password for the new password
 -R, --root CHROOT_DIR            directory to chroot into
 -P, --prefix PREFIX_DIR          prefix directory where are located the /etc/* fi
 -s, --shell SHELL                new login shell for the user account
 -u, --uid UID                    new UID for the user account
 -U, --unlock                     unlock the user account
 -v, --add-subuids FIRST-LAST     add range of subordinate uids
 -V, --del-subuids FIRST-LAST     remove range of subordinate uids
 -w, --add-subgids FIRST-LAST     add range of subordinate gids
 -W, --del-subgids FIRST-LAST     remove range of subordinate gids
 -Z, --selinux-user SEUSER        new SELinux user mapping for the user account
```

- Replace groupname with the group you want to grant sudo access to.

```
root@rhel:~# sudo visudo
visudo: /etc/sudoers busy, try again later
root@rhel:~#
```

4. Apply the Changes:

- Save and exit the visudo editor (Ctrl+X, then Y to confirm, and Enter to save).

```
>_ Terminal
── 0 bash──────────────────────────────────────────────────────────
~
~
~
~
~
~
~
~
~
~
```

- The changes will take effect immediately.

```
~
~
~
~
:wq
```

5. Test the New User’s Sudo Access:

- Log in as the newly added user or use su to switch to that user:

bash

Copy code

su – username

```
root@rhel:~# su vinu
[vinu@rhel root]$
```

- Test sudo access by running:

bash

Copy code

sudo whoami

```
root@rhel:~# su vinu
[vinu@rhel root]$ sudo whoami

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.
```

Lab 3: Understanding and Configuring Sudo Permissions

Objective:

• Understand how to control specific sudo permissions (what commands a user can run with sudo).

Tasks:

1.Limit Sudo Access to Specific Commands:

oOpen the sudoers file and add a rule that only allows a user to run specific commands. For example:

bash

Copy code

username ALL=(ALL) /usr/bin/apt, /usr/bin/dpkg

```
# While you shouldn't normally run git as ro
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GI
#vinu ALL=(ALL) /usr/bin/apt, /usr/bin/dpkg
```

o This allows the user to run only apt and dpkg with sudo.

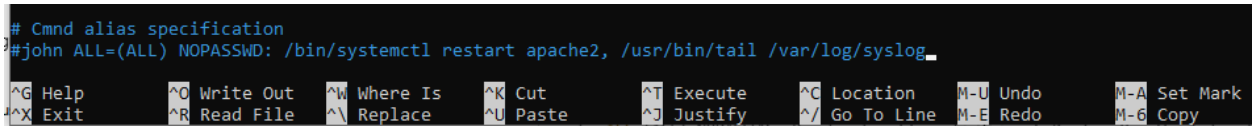
2. Set NOPASSWD for Certain Commands:

- You can configure sudo to not ask for a password for specific commands. Add the following line in the sudoers file:

bash

Copy code

```
username ALL=(ALL) NOPASSWD: /usr/bin/apt, /usr/bin/dpkg
```



- This allows the user to run apt and dpkg without entering a password.

3. Restrict Access to Only Certain Users:

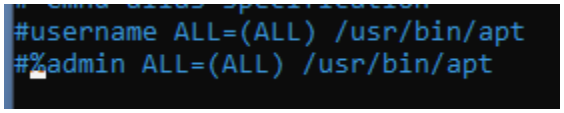
- You can also configure sudo to only allow certain users to execute certain commands. For example:

bash

Copy code

```
username ALL=(ALL) /usr/bin/apt

%admin ALL=(ALL) /usr/bin/apt
```



- This allows username and all users in the admin group to run apt.

```
#username ALL=(ALL) /usr/bin/apt
#%admin ALL=(ALL) /usr/bin/apt
```

4. Apply the Changes and Test:

- Save the sudoers file and exit.

- Test the restricted access by running only the permitted commands as the user.

```
visudo: /etc/sudoers.tmp unchanged
root@DESKTOP-5K616C3:~# apt install
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@DESKTOP-5K616C3:~#
```

Lab 4: Sudo Logs and Auditing

Objective:

• Learn how to view and manage sudo logs to track user activity.

Tasks:

1.Check Sudo Logs:

- By default, sudo logs all commands run to /var/log/auth.log (on Ubuntu/Debian) or /var/log/secure (on CentOS/RHEL).

```
root@rhel:~# cat /var/log/auth.log | grep 'sudo'
cat: /var/log/auth.log: No such file or directory
root@rhel:~#
```

  - View the sudo logs by running:

Copy code

sudo cat /var/log/auth.log # Ubuntu/Debian

```
 pam_unix(sudo:session): session opened for user root(uid=0) by (uid=0)
 pam_unix(sudo:session): session closed for user root
     root : TTY=pts/0 ; PWD=/root ; USER=root ; COMMAND=/usr/sbin/visud
 pam_unix(sudo:session): session opened for user root(uid=0) by (uid=0)
 pam_unix(sudo:session): session closed for user root
1341]: pam_unix(cron:session): session opened for user root(uid=0) by (
1341]: pam_unix(cron:session): session closed for user root
     root : TTY=pts/0 ; PWD=/root ; USER=root ; COMMAND=/usr/bin/cat /v
```

sudo cat /var/log/secure # CentOS/RHEL

```
Feb 27 22:48:09 rhel-9-1-6-20-2023 usermod[2080]: add 'rhel' to group 'wheel'
Feb 27 22:48:09 rhel-9-1-6-20-2023 usermod[2080]: add 'rhel' to shadow group 'wheel'
Feb 27 22:48:09 rhel-9-1-6-20-2023 passwd[2088]: pam_unix(passwd:chauthtok): password changed for rhel
Feb 27 22:48:42 rhel-9-1-6-20-2023 sshd[1790]: pam_unix(sshd:session): session closed for user root
Feb 27 22:49:49 rhel-9-1-6-20-2023 useradd[2457]: new group: name=vinu, GID=1015
Feb 27 22:49:49 rhel-9-1-6-20-2023 useradd[2457]: new user: name=vinu, UID=1014, GID=1015, home=/home/vin
u, shell=/bin/bash, from=/dev/pts/0
root@rhel:~#
```

2. Search for Sudo Commands:

o Use grep to search for sudo-related logs:

bash

Copy code

sudo grep 'sudo' /var/log/auth.log

```
 pam_unix(sudo:session): session opened for user root(uid=0) by (uid=0)
 pam_unix(sudo:session): session closed for user root
     root : TTY=pts/0 ; PWD=/root ; USER=root ; COMMAND=/usr/sbin/visud
 pam_unix(sudo:session): session opened for user root(uid=0) by (uid=0)
 pam_unix(sudo:session): session closed for user root
1341]: pam_unix(cron:session): session opened for user root(uid=0) by (
1341]: pam_unix(cron:session): session closed for user root
     root : TTY=pts/0 ; PWD=/root ; USER=root ; COMMAND=/usr/bin/cat /v
```

3.Configure Logging Level:

oYou can configure the logging level of sudo by modifying the sudoers file.

oAdd a line to the sudoers file to set the logging level (optional):

bash

Copy code

Defaults logfile="/var/log/sudo.log"

```
## rather than USERALIAS
# User_Alias ADMINS = jsmith, mikem
#Defaults logfile="/var/log/sudo.log"
```

4. View the Sudo Log File:

- You can now monitor the sudo log file to track all sudo commands used by different users:

bash

Copy code

sudo tail -f /var/log/sudo.log

bash Copy Edit

```bash
sudo tail -f /var/log/sudo.log
```

it will display the last few lines of the `sudo.log` file and continuously update as new `sudo` entries are logged.

If you meant to check whether `/var/log/sudo.log` exists and is a file, you can run:

bash Copy Edit

```bash
ls -l /var/log/sudo.log
```

Lab 5: Sudo Timeout and Tuning

Objective:

• Learn how to configure the sudo session timeout to improve security.

Tasks:

1. Set the sudo Timeout:

 o The sudo session timeout can be controlled by setting the timestamp_timeout parameter in the sudoers file.

 o To configure the timeout (in minutes), edit the sudoers file:

bash

Copy code

Defaults timestamp_timeout=10

```
# User_Alias ADMINS = jsmith, mikem
#Defaults logfile="/var/log/sudo.log"
#Defaults timestamp_timeout=XX
```

 o This means sudo will prompt for a password every 10 minutes.

2.Disable the Timeout:

- To disable the timeout entirely, set the timeout to 0:

bash

Copy code

Defaults timestamp_timeout=0

```
# User_Alias ADMINS = jsmith, mike
#Defaults logfile="/var/log/sudo.l
#Defaults timestamp_timeout=XX
#Defaults timestamp_timeout=0
```

3. Test the Timeout Configuration:

- Run a sudo command and wait for the specified timeout duration. After the timeout, sudo should prompt you for the password again.


```
# User_Alias ADMINS = jsmith, mike
#Defaults logfile="/var/log/sudo.l
#Defaults timestamp_timeout=XX
#Defaults timestamp_timeout=0
```

```
# User_Alias ADMINS = jsmith, mike
#Defaults logfile="/var/log/sudo.l
#Defaults timestamp_timeout=XX
#Defaults timestamp_timeout=0
```

4. Set a Negative Timeout:

- If you set timestamp_timeout to -1, sudo will ask for the password every time a sudo command is executed:

bash

Copy code

Defaults timestamp_timeout=-1

```
vinu@DESKTOP-5K616C3:~$ sudo visudo
/etc/sudoers:27:22: syntax error
timestamp_timeout to -1
                     ^~
What now?
```

Lab 6: Troubleshooting Sudo Issues

Objective:

• Learn how to troubleshoot common sudo access issues.

Tasks:

1.Check User's Group Membership:

- Verify that the user is part of the correct group (sudo or wheel) by running:

bash

Copy code

groups username

```
root@rhel:~/vinu# sudo usermod -aG vinu vinu
root@rhel:~/vinu#
```

2. Verify Permissions in the Sudoers File:

- Check for syntax errors in the sudoers file by running:

bash

Copy code

sudo visudo

```
root@rhel:~/vinu# sudo usermod -aG vinu vinu
root@rhel:~/vinu# sudo -l -U vinu
User vinu is not allowed to run sudo on rhel.
root@rhel:~/vinu#
```

- Ensure that there are no conflicting rules or misconfigurations.


```
root@rhel:~/vinu# sudo usermod -aG vinu vinu
root@rhel:~/vinu# sudo -l -U vinu
User vinu is not allowed to run sudo on rhel.
root@rhel:~/vinu#
```


3. Check Sudo Log for Errors:

- If a user cannot execute sudo, check the logs for any errors related to authentication.

```
root@rhel:~/vinu# sudo grep "sudo" /var/log/secure
```

```
── 0 bash──────────────────────────────────────────────────────────────────────
=/sbin/visudo
Feb 27 23:08:52 rhel-9-1-6-20-2023 sudo[2520]: pam_unix(sudo:session): session opened for user root
) by root(uid=0)
Feb 27 23:19:16 rhel-9-1-6-20-2023 sudo[2520]: pam_unix(sudo:session): session closed for user root
Feb 27 23:19:42 rhel-9-1-6-20-2023 sudo[2532]:    root : TTY=pts/0 ; PWD=/root/vinu ; USER=root ; C
=/sbin/usermod -aG sudo username
Feb 27 23:19:42 rhel-9-1-6-20-2023 sudo[2532]: pam_unix(sudo:session): session opened for user root
) by root(uid=0)
Feb 27 23:19:42 rhel-9-1-6-20-2023 sudo[2532]: pam_unix(sudo:session): session closed for user root
Feb 27 23:20:19 rhel-9-1-6-20-2023 sudo[2536]:    root : TTY=pts/0 ; PWD=/root/vinu ; USER=root ; C
=/sbin/usermod -aG sudo vinu
Feb 27 23:20:19 rhel-9-1-6-20-2023 sudo[2536]: pam_unix(sudo:session): session opened for user root
) by root(uid=0)
Feb 27 23:20:19 rhel-9-1-6-20-2023 sudo[2536]: pam_unix(sudo:session): session closed for user root
Feb 27 23:25:32 rhel-9-1-6-20-2023 sudo[2540]:    root : TTY=pts/0 ; PWD=/root/vinu ; USER=root ; C
=/sbin/usermod -aG vinu vinu
Feb 27 23:25:32 rhel-9-1-6-20-2023 sudo[2540]: pam_unix(sudo:session): session opened for user root
) by root(uid=0)
Feb 27 23:25:32 rhel-9-1-6-20-2023 sudo[2540]: pam_unix(sudo:session): session closed for user root
```

4. Test with Another User:

```
root@rhel:~/vinu# sudo whoami
root
root@rhel:~/vinu#
```

- If one user cannot use sudo, try using another user with sudo access to determine if the problem is user-specific.

```
root@rhel:~/vinu# sudo whoami
root
root@rhel:~/vinu#
```

su - another_user

```
root@DESKTOP-5K616C3:~# su - vinu
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.15.167.4-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro
```