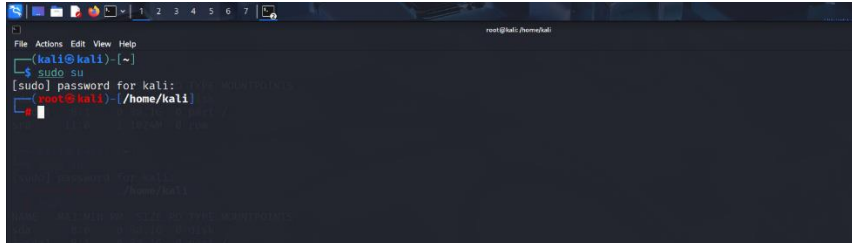


Navigating the Linux Filesystem and Permission Settings.

Part 1: Exploring Filesystems in Linux:

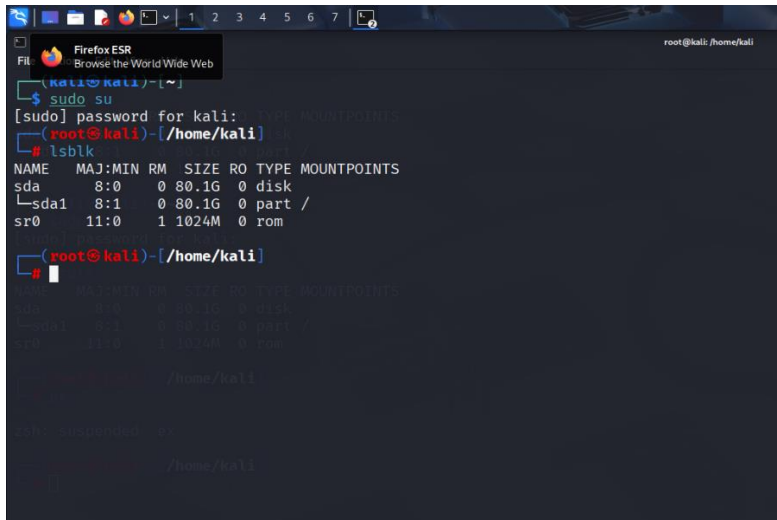
Step 1: Access the command line and launch the VM Workstation and open a terminal.



```
(kali@kali)-[~]
└─$ sudo su
[sudo] password for kali:
root@kali: /home/kali
```

Step 2: Display the filesystems currently mounted: A filesystem must be mounted before use. Mounting attaches a storage device's partition to a directory, making its contents accessible to the operating system. This directory, which becomes the root of the attached filesystem.

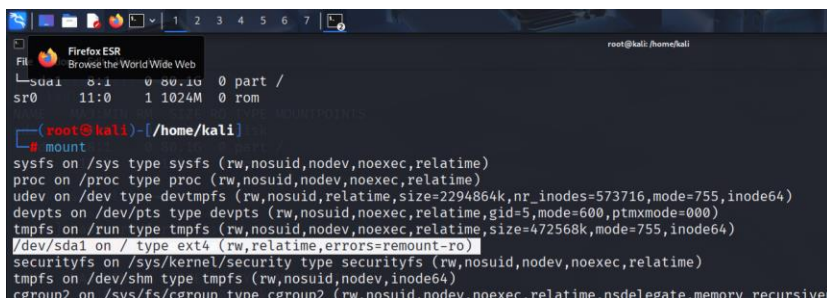
- i. Use the **lsblk** command to display all block devices:



```
root@kali: /home/kali
└─$ lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda          8:0    0 80.1G  0 disk
└─sda1       8:1    0 80.1G  0 part /
sr0         11:0    1 1024M  0 rom
```

The output above shows that the Workstation has two block devices installed: sr0 and sda.

- ii. Use the **mount** command to display more detailed information on the currently mounted filesystems in the Workstation.



```
root@kali: /home/kali
└─$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=2294864k,nr_inodes=573716,mode=755,inode64)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=600,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,nodev,noexec,relatime,size=472568k,mode=755,inode64)
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev,inode64)
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_recursiveprot)
```

Focusing on the root filesystem, the filesystem stored in **/dev/sda1**. The root filesystem is where the Linux operating system itself is stored; all the programs, tools, configuration files are stored in root filesystem by default.

- iii. Run the **mount** command again with the pipe **|** to send the output of mount to **grep** to filter the output and display only the root filesystem:

```
(root@kali)-[/home/kali]
# mount | grep sda1
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)

(root@kali)-[/home/kali]
#
```

The **mount** output indicates that the root filesystem resides on the first partition of the **sda** device (**/dev/sda1**). We identify it as the root filesystem by its mount point **/**. The output also specifies the partition's filesystem type, **ext4** here, and lists its mount options in parentheses.

- iv. Issue the following two commands below on the **Security Workstation VM**:

Step 3: Manually mounting and unmounting filesystems.

- i. Use the **ls -l** command to verify that the directory **second_drive** is in the analyst's home directory.

Note: If the **second_drive** directory does not already exist, create it using **mkdir second_drive**.

```
(root@kali)-[/home/kali]
# cd ~

(root@kali)-[~]
# ls -l
total 760
drwxr-xr-x 2 root root 4096 Aug 29 13:00 NewFolder
-rw-r--r-- 1 root root 91 May 4 2025 pass.txt
-rw-r--r-- 1 root root 252928 Aug 8 2025 Payload.exe
-rw-r--r-- 1 root root 673 Feb 10 2025 practice.txt
-rw-r--r-- 1 root root 3745 Dec 17 07:17 special.txt
-rw-r--r-- 1 root root 252928 Aug 30 02:27 testpayload1.exe
-rw-r--r-- 1 root root 252928 Aug 29 13:01 testpayload.exe

(root@kali)-[~]
# mkdir second_drive

(root@kali)-[~]
# ls -l
total 764
drwxr-xr-x 2 root root 4096 Aug 29 13:00 NewFolder
-rw-r--r-- 1 root root 91 May 4 2025 pass.txt
-rw-r--r-- 1 root root 252928 Aug 8 2025 Payload.exe
-rw-r--r-- 1 root root 673 Feb 10 2025 practice.txt
drwxr-xr-x 2 root root 4096 Feb 14 04:35 second_drive
-rw-r--r-- 1 root root 3745 Dec 17 07:17 special.txt
-rw-r--r-- 1 root root 252928 Aug 30 02:27 testpayload1.exe
-rw-r--r-- 1 root root 252928 Aug 29 13:01 testpayload.exe

(root@kali)-[~]
#
```

- ii. Use `ls -l` again to list the contents of the newly created `second_drive` directory.

```
(root@kali)~  
# ls -l second_drive  
total 0  
  
(root@kali)~  
#
```

Note that the directory is empty.

- iii. Use the `mount` command to attach `/dev/sda1` to the newly created `second_drive` directory.

```
(root@kali)~/home/kali  
# mount /dev/sda1 ~/second_drive  
  
(root@kali)~/home/kali  
#
```

No output is provided which means the mounting process was successful.

- iv. Run the `mount` command again without options to view the detailed information for `/dev/sda1`. As before, pipe the output through `grep` to show only the `/dev/sd` entries.

```
(root@kali)~  
# mount | grep /dev/sd  
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)  
/dev/sda1 on /root/second_drive type ext4 (rw,relatime,errors=remount-ro)  
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)  
  
(root@kali)~  
#
```

Part 2: File Permissions

Linux filesystems include built-in permissions that control how users can view, modify, navigate, and execute files. Each file has its own set of permissions defining what actions are allowed for specific users and groups.

Step 1: Visualize and Change the File Permissions.

- Use the `ls -l` command to display file permissions.

```
root@kali: /home/kali
[sudo] password for kali:
root@kali: /home/kali
# ls -l
total 3320960
-rw-r--r-- 1 root root 0 Apr 20 2019 2018-12-19-MyDoom-infection-traffic.pcap
-rw-r--r-- 1 root root 204725 Aug 11 2024 2018-12-19-MyDoom-infection-traffic.pcap.zip
drwxr-xr-x 2 root root 4096 Aug 11 2024 2018-12-19-MyDoom-zip-attachments-and-extracted-EXE-files
-rw-r--r-- 1 root root 171343 Aug 11 2024 2018-12-19-MyDoom-zip-attachments-and-extracted-EXE-files.zip
-rw-r--r-- 1 root root 0 Sep 5 20:50 2025-09-05-Xloader-infection-traffic.pcap
-rw-r--r-- 1 root root 16958872 Sep 18 22:33 2025-09-05-Xloader-infection-traffic.pcap.zip
drwxrwxr-x 2 kali kali 4096 Apr 12 2025 Abuja
drwxr-xr-x 11 root root 4096 Mar 22 2025 AdvPhishing
drwxr-xr-x 8 root root 4096 Aug 29 07:00 ALEAPP
drwxr-xr-x 3 root root 4096 Feb 8 17:40 Analyzing-a-MySQL-Database-Attack
drwxr-xr-x 11 root root 4096 May 8 2025 AndroRAT
drwxrwxr-x 2 kali kali 4096 Apr 22 2025 armitage-tmp
drwxr-xr-x 5 root root 4096 Mar 22 2025 CyberPhish
drwxr-xr-x 2 kali kali 4096 Aug 29 19:41 Desktop
drwxr-xr-x 2 kali kali 4096 May 5 2025 Documents
drwxr-xr-x 2 kali kali 4096 Dec 10 08:54 Downloads
drwxrwxr-x 4 kali kali 4096 Feb 19 2025 enum4linux-ng
-rw-r--r-- 1 root root 1908226 Aug 5 2025 get-pip.py
-rw-r--r-- 1 root root 1908226 May 2 2025 get-pip.py.1
-rw-r--r-- 1 root root 1908226 May 2 2025 get-pip.py.2
drwxr-xr-x 3 root root 4096 May 2 2025 ghost_eye
drwxr-xr-x 15 root root 4096 Dec 23 15:56 gobuster
-rw-r--r-- 1 root root 3460 Oct 6 23:27 index.html
drwxr-xr-x 3 root root 4096 Feb 8 17:37 indicators
drwxr-xr-x 2 kali kali 4096 Feb 3 2025 Music
-rw-r--r-- 1 root root 165 Jul 26 2025 my_pw_hashes.txt
-rw-r--r-- 1 root root 9307097 Sep 21 13:43 ngrok-v3-stable-linux-amd64.tgz
-rw-r--r-- 1 root root 672 Sep 22 07:07 passkey2.txt
-rw-r--r-- 1 root root 136 May 13 2025 passkey.txt
```

- Use the `chmod` command to change the permissions of `passkey2.txt`.

```
root@kali: /home/kali
# sudo chmod 665 passkey2.txt
```

```
drwxr-xr-x 3 root root 4096 Feb 8 17:37 indicators
drwxr-xr-x 2 kali kali 4096 Feb 3 2025 Music
-rw-r--r-- 1 root root 165 Jul 26 2025 my_pw_hashes.txt
-rw-r--r-- 1 root root 9307097 Sep 21 13:43 ngrok-v3-stable-linux-amd64.tgz
-rw-rw-r-x 1 root root 672 Sep 22 07:07 passkey2.txt
-rw-r--r-- 1 root root 136 May 13 2025 passkey.txt
-rw-r--r-- 1 root root 31 May 13 2025 passwords.txt
-rw-r--r-- 1 root root 28 May 13 2025 password.txt
drwxr-xr-x 2 kali kali 4096 Feb 3 2025 Pictures
drwxr-xr-x 2 kali kali 4096 Feb 3 2025 Public
-rw-r--r-- 1 root root 24932352 Dec 20 2015 python-3.4.4.msi
drwxr-xr-x 8 root root 4096 May 4 2025 Responder
```

Note that the permission was changed from `-rw-r--r--` to `-rw-rw-r-x`

The `chmod` command takes permissions in the octal format. In that way, a breakdown of the 665 is as follows:

6 in octal is 110 in binary. Assuming each position of the permissions of a file can be 1 or 0, 110 means `rw-` (read=1, write=1 and execute=0).

Therefore, the `chmod 665 passkey2.txt` command changes the permissions to:

Owner: rw- (6 in octal or 110 in binary)

Group: rw- (6 in octal or 110 in binary)

Other: r-x (5 in octal or 101 in binary)

- c. The **chown** command is used to change ownership of a file or directory. Issue the command to make root the owner of the **passkey2.txt**

```
(root@kali)-[/home/kali]
# chown kali passkey2.txt

(root@kali)-[/home/kali]
#
```

Note, the ownership is being changed from **root** to **kali**

```
-rw-r--r-- 1 root root      165 Jul 26  2025 my_pw_hashes.txt
-rw-r--r-- 1 root root 9307097 Sep 21 13:43 ngrok-v3-stable-linux-amd64.tgz
-rw-rw-r-x 1 kali root      672 Sep 22 07:07 passkey2.txt
-rw-r--r-- 1 root root     136 May 13  2025 passkey.txt
-rw-r--r-- 1 root root       31 May 13  2025 passwords.txt
```

