FILESYSTEM AND STORAGE DEVICE MANAGEMENT

If you are coming from a Windows environment, the way that Linux represents and manages storage devices will look rather different to you. You've already seen that the filesystem has no physical representation of the drive, like the C:, D:, or E: system in Windows, but rather has a file tree structure with / at the top, or root, of it. We first look how additional drives and other storage devices are mounted upon that filesystem, leading up to the / (root) directory. ***Mounting in this context simply means attaching drives or disks to the filesystem to make them accessible to the operating system (OS).*** For you as a hacker, it's necessary to understand the file and storage device management system, both on your own system and, often, the system of your target. Hackers commonly use external media to load data, hacking tools, or even their OS. Once you're on your target system, you need to understand what you're working with, where to find confidential or other critical files, how to mount a drive to the target, and whether and where you can put those files on your system. We begin with the directory known as /dev, which you've probably already noticed in the directory structure: dev is short for device, and every device in Linux is represented by its own file within the /dev directory. Let's start out by working with /dev.

THE DEVICE DIRECTORY /DEV:

Linux has a special directory that contains files representing each attached device: the appropriately named /dev directory. As your first introduction, navigate to the /dev directory and then perform a long listing on it.

```
Kali> cd /dev
```

The devices are displayed in alphabetical order by default. You may recognize some of the devices, such a cdrom and cpu, but others have rather cryptic names. Each device on your system is represented by a file in the /dev directory, including devices you've probably never used or even realized existed. On the off chance you do, there is a device file waiting to be used for it.

If you scroll down this screen a bit, you should see more listings of devices. Of particular interest are the devices sda1, sda2, sda3, sdb, and sdb1, which are the hard drive and its partitions and a USB flash drive and its partitions.

HOW LINUX REPRESENTS STOREAGE DEVICES:

Linux uses logical labels for drives that are then mounted on the filesystem. These logical labels will vary depending on where the drives are mounted, meaning the same hard drive might have different labels at different times, depending on where and when it's mounted.

Originally, Linux represented floppy drives (remember those?) as fd0 and hard drives as hda. You will still occasionally see these drive representations on legacy Linux systems, but today most floppy drives are gone (thank goodness). Even so, old legacy hard drives that used an IDE or EIDE interface are still represented in the form hda. ***Newer Serial ATA (SATA) interface drives and Small Computer System Interface (SCSI) hard drives are represented as sda***. ***Drives are sometimes split up into sections known as partitions, which are represented in the labeling system with numbers***. When systems have more than one hard drive, Linux simply names them serially by incrementing the last letter in

alphabetical order, so the first drive is sda, and the second drive is sdb, the third drive is sdc, and so on. The serial letter after sd is often referred to as the major number.

DEVICE NAMING SYSTEM:

Table:

DEVICE	FILE DESCRTIPTION
sda	First SATA hard drive
sdb	Second SATA hard drive
sdc	Third SATA hard drive
Sdd	Fourth SATA hard drive

DRIVE PARTITION:

Some drives can be split into partitions in order to manage and separate information. For instance, you may want to separate your hard drive so that your swap file, home directory, and / directory are all on separate partitions—you might want to do this for a number of reasons, including to share resources and to relax the default permissions. Linux labels each partition with a minor number that comes after the drive designation. This way, the first partition on the first SATA drive would be sda1. The second partition would then be sda2, the third sda3, and so on.

TABLE:

PARTITION	DESCRIPTION
sda1	The first partition (1) on the first (a) SATA drive
sda2	The second (2) partition on the first (a) drive
sda3	The third (3) partition on the first (a) drive
sda4	The fourth (4) partition on the first (a) drive

At times, you may want to view the partitions on your Linux system to see which ones you have and how much capacity is available in each. You can do this by using the fdisk utility. Using the -l switch with fdisk lists all the partitions of all the drives

```
Kali> fdisk -l
```

The devices sda1, sda2, and sda5 are listed in the first stanza. These three devices make up the virtual disk from my virtual machine, which is a X GB drive with three partitions, including the swap partition (sda5), which acts like virtual RAM—similar to page files in Windows—when RAM capacity is exceeded.

Note that fdisk indicates that it is an HPFS/NTFS/ExFAT filesystem type. These file types—***High Performance File System (HPFS), New Technology File System (NTFS), and Extended File Allocation Table (exFAT)—***are not native to Linux systems but rather to macOS and Windows systems. It's worth being able to recognize file types native to different systems when you investigate. The filesystem might indicate what kind of machine the drive was formatted on, which can be valuable information. Kali is able to utilize USB flash drives created on many different operating systems. On top of this, the way files are stored and managed is different in Linux, too. New versions of Windows use an NTFS filesystem, whereas older Windows systems use File Allocation Table (FAT) systems.

Linux uses a number of different types of filesystems, but the most common are ext2, ext3, and ext4. These are all iterations of the ext (or extended) filesystem, with ext4 being the latest.

CHARACTER AND BLOCK DEVICES:

Something else to note about the naming of device files in the /dev directory is that the first position contains either c or b. You can see this in Listing 101 at the start of most of the entries, and it looks something like this:

```
---------------------------------------------------------------
crw------- 1 root root 10,175 May 16 12:44 agpgart
---------------------------------------------------------------
```

These letters represent the two ways that devices transfer data in and out. ***The c stands for character***, and these devices are known, as you might expect, as ***character devices***. ***External devices that interact with the system by sending and receiving data character by character, such as mice or keyboards, are character devices***. ***The b stands for the second type: block devices. They communicate in blocks of data (multiple bytes at a time) and include devices like hard drives and DVD drives. These devices require higher speed data throughput and therefore send and receive data in blocks (many characters or bytes at a time).*** Once you know whether a device is a character or block device, you can easily get more information about it.

LIST BLOCK DEVICES AND INFO WITH LSBLK:

The Linux command lsblk, short for list block, lists some basic information about each block device listed in /dev. The result is similar to the output from fdisk -l, but it will also display devices with multiple partitions in a kind of tree, showing each device with its partitions as branches, and does not require root privileges to run.

```
Kali> lsblk
```

The output includes the floppy drive as fd0 and DVD drive as sr0, even though neither is on my system—this is simply a holdover from legacy systems. ***We can also see information on the mount point of the drive—this is the position at which the drive was attached to the filesystem. Note that the hard drive sda1 is mounted at / and the flash drive is mounted at /media***.

MOUNTING AND UNMOUNTING:

Most modern operating systems, including most new versions of Linux, automount storage devices when they're attached, meaning the new flash drive or hard drive is automatically attached to the filesystem.

A storage device must be first physically connected to the filesystem and then logically attached to the filesystem in order for the data to be made available to the operating system. In other words, even if the device is physically attached to the system, it is not necessarily logically attached and available to the operating system. The term mount is a legacy from the early days of computing when storage tapes (before hard drives) had to be physically mounted to the computer system—think of those big computers with spinning tape drives you might have seen old scifi movies.

As mentioned, the point in the directory tree where devices are attached is known as the mount point. The two main mount points in Linux are /mnt and /media. As a general rule, internal hard drives are mounted at /mnt, and external USB devices such as flash drives and external USB hard drives are mounted at /media, though technically any directory can be used.

MOUNTING STORAGE DEVICE:

In some versions of Linux, you need to mount a drive manually in order to access its content, so this is a skill worth learning. To mount a drive on the filesystem, use the ***mount*** command. The mount point for the device should be an empty directory; if you mount a device on a directory that has subdirectories and files, the mounted device will cover the contents of the directory, making them invisible and unavailable.

```
Kali> mount  /dev/sdb1  /mnt
```

```
Kali> mount  /dev/sdc1  /media
```

The filesystems that are mounted on a system are kept in a file at `/etc/fstab` (short for filesystem table), which is read by the system at every bootup.

UNMOUNTING WITH UMOUNT:

If you're coming from a Mac or Windows background, you've probably unmounted a drive without knowing it. Before you remove a flash drive from your system, you "eject" it to keep from causing damage to the files stored on the device. Eject is just another word for unmount. Similar to the mount command, you can unmount a second hard drive by entering the umount command followed by the file entry of the device in the /dev directory, such as /dev/sdb. Note that the command is not spelled unmount but rather umount (no n).

```
Kali> umount  /dev/sdb1
```

You cannot unmount a device that is busy, so if the system is reading or writing to the device, you will just receive an error.

MONITORING FILE SYSTEM:

The command df (for disk free) will provide us with basic information on any hard disks or mounted devices, such as CD, DVD, and flash drives, including how much space is being used and how much is available. Without any options, df defaults to the first drive on your system (in this case, sda). If you want to check a different drive, simply follow the df command with the drive representation you want to check (for example, df sdb).

```
Kali> df
```

The first line of output here shows category headers, and then we get the information. The disk space is given in 1KB blocks.

```
Kali> du <path of file or directory>
```

du command, short for disk usage, is used to estimate file space usage.

The du command can be used to track the files and directories which are consuming excessive amount of space on hard disk drive.

CHECKING FOR ERROR:

The fsck command (short for filesystem check) checks the filesystem for errors and repairs the damage, if possible, or else puts the bad area into a bad blocks table to mark it as bad. To run the fsck command, you need to specify the filesystem type (the default is ext2) and the device file to check. It's important to note that you must unmount the drive before running a filesystem check. If you fail to unmount the mounted device, you will receive the error message.

```
Kali> umount  /dev/sdb1
```

```
Kali> fsck -p /dev/sdb1
```

```
-p          to repair automatically any problems with the device.
```