Bryce Baxter

10/29/2025

CYSE 270

Assignment 7 – Storage Management

Part I

Steps 1-2

```
File  Edit  View  Terminal  Tabs  Help

┌──(root㉿kali.example.com)-[/home/student]
└─# sudo ls /dev/nvme*
/dev/nvme0  /dev/nvme0n1  /dev/nvme0n1p1  /dev/nvme0n1p14  /dev/nvme0n1p15

┌──(root㉿kali.example.com)-[/home/student]
└─# sudo fdisk -l
Disk /dev/nvme0n1: 24 GiB, 25769803776 bytes, 50331648 sectors
Disk model: Amazon Elastic Block Store
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: 71DFD449-F0E2-4249-B370-2373EC9CF964

Device           Start      End  Sectors  Size Type
/dev/nvme0n1p1  262144 50331614 50069471 23.9G Linux filesystem
/dev/nvme0n1p14   2048     8191     6144    3M BIOS boot
/dev/nvme0n1p15   8192   262143   253952  124M EFI System

Partition table entries are not in disk order.
```

Step 3

```
┌──(root㉿kali.example.com)-[/home/student]
└─# sudo parted -l
Model: Amazon Elastic Block Store (nvme)
Disk /dev/nvme0n1: 25.8GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
14      1049kB  4194kB  3146kB                     bios_grub
15      4194kB  134MB   130MB   fat16              boot, esp
 1      134MB   25.8GB  25.6GB  ext4
```

Part II

Step 1

```
┌──(root㉿kali.example.com)-[/home/student]
└─# sudo dd if=/dev/zero of=~/01294121.vdi bs=1M count=200
200+0 records in
200+0 records out
209715200 bytes (210 MB, 200 MiB) copied, 0.130017 s, 1.6 GB/s
```

Step 2

```
┌──(root㉿kali.example.com)-[/home/student]
└─# sudo losetup -fP ~/01294121.vdi
```

Step 3

```
File  Edit  View  Terminal  Tabs  Help

┌──(root㉿kali.example.com)-[/home/student]
└─# sudo ls /dev/nvme*
/dev/nvme0  /dev/nvme0n1  /dev/nvme0n1p1  /dev/nvme0n1p14  /dev/nvme0n1p15

┌──(root㉿kali.example.com)-[/home/student]
└─# sudo fdisk -l
Disk /dev/nvme0n1: 24 GiB, 25769803776 bytes, 50331648 sectors
Disk model: Amazon Elastic Block Store
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: 71DFD449-F0E2-4249-B370-2373EC9CF964

Device           Start      End  Sectors  Size Type
/dev/nvme0n1p1  262144 50331614 50069471 23.9G Linux filesystem
/dev/nvme0n1p14   2048     8191     6144    3M BIOS boot
/dev/nvme0n1p15   8192   262143   253952  124M EFI System

Partition table entries are not in disk order.


Disk /dev/loop0: 200 MiB, 209715200 bytes, 409600 sectors
```

```
┌──(root㉿kali.example.com)-[/home/student]
└─# sudo parted -l
Model: Amazon Elastic Block Store (nvme)
Disk /dev/nvme0n1: 25.8GB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start   End     Size    File system  Name  Flags
14      1049kB  4194kB  3146kB                     bios_grub
15      4194kB  134MB   130MB   fat16              boot, esp
 1      134MB   25.8GB  25.6GB  ext4
```

Part III

Steps 1-2

```
File  Edit  View  Terminal  Tabs  Help

┌──(root㉿kali.example.com)-[/home/student]
└─# sudo mkfs.ext4 /dev/loop0
mke2fs 1.47.2 (1-Jan-2025)
Discarding device blocks: done
Creating filesystem with 204800 1k blocks and 51200 inodes
Filesystem UUID: 7e921c70-1f7d-45df-8a57-694ebe473223
Superblock backups stored on blocks:
        8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done


┌──(root㉿kali.example.com)-[/home/student]
└─# sudo mkdir /cyse

┌──(root㉿kali.example.com)-[/home/student]
└─# sudo mount /dev/loop0 /cyse
```

Step 3

```
┌──(root㉿kali.example.com)-[/home/student]
└─# df -h
Filesystem       Size  Used Avail Use% Mounted on
udev             1.9G     0  1.9G   0% /dev
tmpfs            387M 1004K  386M   1% /run
/dev/nvme0n1p1    24G   14G  8.9G  61% /
tmpfs            1.9G  4.0K  1.9G   1% /dev/shm
tmpfs            5.0M     0  5.0M   0% /run/lock
tmpfs            1.9G  8.0K  1.9G   1% /tmp
tmpfs            1.0M     0  1.0M   0% /run/credentials/systemd-journald.service
/dev/nvme0n1p15  124M  286K  124M   1% /boot/efi
tmpfs            1.0M     0  1.0M   0% /run/credentials/systemd-resolved.service
tmpfs            1.0M     0  1.0M   0% /run/credentials/getty@tty1.service
tmpfs            387M  108K  387M   1% /run/user/1000
tmpfs            387M  100K  387M   1% /run/user/0
tmpfs            1.0M     0  1.0M   0% /run/credentials/serial-getty@ttyS0.servi
ce
/dev/loop0       182M   64K  168M   1% /cyse
```

Step 4

```
┌──(root㉿kali.example.com)-[/home/student]
└─# echo Bryce | sudo tee /cyse/01294121.txt
Bryce
```

Steps 5-6

```
┌──(root㉿kali.example.com)-[/home/student]
└─# sudo umount /cyse

┌──(root㉿kali.example.com)-[/home/student]
└─# ls /cyse
```

Part IIII

1. Explain the purpose of using the `sudo` command with `ls /dev/sd*` and `ls /dev/nvme*`. Why is administrator privilege required in this context? **The sudo command is used to execute a command with administrative privileges. When used with ls /dev/sd*, it ensures that the user has all permissions to list all blocked devices**
2. What is a loop device, and why do we use `losetup` to attach the virtual disk file as a loop device in this lab? **A loop device is a device that is used to make a file accessible. The losetup command can be used to associate a file with a loop device. This then causes the system to treat the virtual disk as a physical disk.**
3. Why do we format the virtual disk using `mkfs.ext4`? Explain what this command does and why we chose the `ext4` filesystem specifically. **The mkfs.ext4 command is used to create a ext4 filesystem on the virtual disk. The command initiates the data structures on the disk so that the operating system can store and retrieve files.**
4. After mounting the virtual disk to `/cyse`, what changes should you observe in the output of `df -h`? Explain how `df` helps verify that the disk is mounted correctly. **After mounting the disk to /cyse, df -h will show a different entry for the mounted filesystem. This will display the size, used and available space, and the mount point of the disk.**
5. Why is it important to unmount a directory (like `/cyse` in this lab) before detaching a virtual disk? What could happen if you detach a disk without unmounting it first? **It's important to unmount a directory before detaching a virtual disk to ensure that all data has been written on the existing disk and that the filesystem is in a consistent state. If you detach a disk without unmounting it first, there is a risk of data loss or corruption.**
6. After creating a file on the mounted virtual disk and then unmounting the disk, what do you expect to see when you check the contents of `/cyse`? Explain why this happens. **After you unmount the disk, the /cyse directory should be empty. This happens**

because unmounting the disk removes the filesystem from the directory tree and the directory is reverted into it's original state.

7. How does using a virtual disk file differ from using a physical disk partition on your system? What are some advantages and disadvantages of using virtual disks in cybersecurity labs? **A virtual disk file is a file on a existing filesystem that is treated as a block device while a physical disk partition is a designated section of a physical storage device. Some adavantages of using virtual disks are that files on virtual disks can be easily moved and copied and they can also be easily created and deleted at any time. The disadvantages are performance issues compared to physical partitions and storage management issues.**