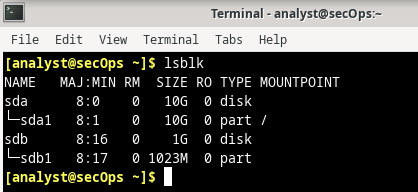
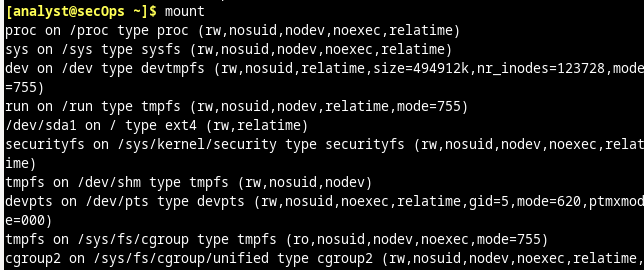
a. Use the lsblk command to display all block devices:  
  
b. Use the mount command to display more detailed information on the currently mounted filesystems in theCyberOps Workstation VM.

[analyst@secOps ~]$ mount

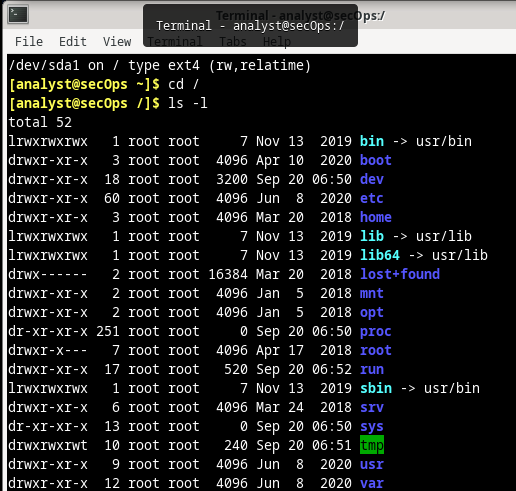


c. Run the mount command again, but this time, use the pipe | to send the output of mount to grep to filter the output and display only the root filesystem: [analyst@secOps ~]$ mount | grep sda1



d. Issue the following two commands below on the CyberOps Workstation VM:

[analyst@secOps ~]$ cd /

[analyst@secOps /]$ ls -l   


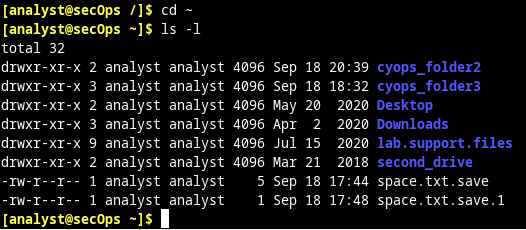
Questions: What is the meaning of the output? Where are the listed files physically stored?  
=> The first command changes the directory to the root directory. The root directory is the highest level of the filesystems. Because /dev/sda1 is mounted on the root directory (“/”), by listing the files in the root directory, the user is actually listing files physically stored in the root of the /dev/sda1 filesystem.

Why is /dev/sdb1 not shown in the output above?

Because /dev/sdb1 is not currently mounted.

a. Use the ls -l command to verify that the directory second\_drive is in the analyst's home directory.

[analyst@secOps /]$ cd ~

[analyst@secOps ~]$ ls –l  
  
  
b. Use ls -l again to list the contents of the newly created second\_drive directory.

[analyst@secOps ~]$ ls -l second\_drive/

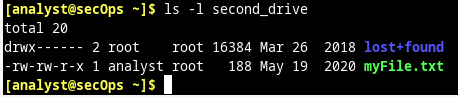
c. Use the mount command to mount /dev/sdb1 on the newly created second\_drive directory. The syntax of mount is: mount [options] <device to be mounted> <mounting point>.

[analyst@secOps ~]$ sudo mount /dev/sdb1 ~/second\_drive/



Now that the /dev/sdb1 has been mounted on /home/analyst/second\_drive, use ls -l to list the contents of the directory again.

[analyst@secOps ~]$ ls -l second\_drive/



Why is the directory no longer empty? Where are the listed files physically stored?  
=> After the mount, /home/analyst/second\_drive becomes the entry point to the filesystem physically stored in /dev/sdb1.

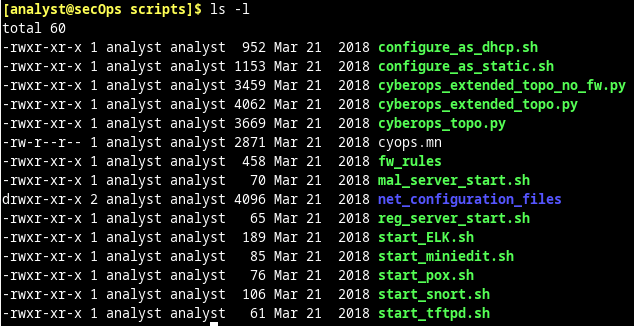
e. Issue the mount command with no options again to display detailed information about the /dev/sdb1 partition. As before, use the grep command to display only the /dev/sdX filesystems:

[analyst@secOps ~]$ mount | grep /dev/sd 

f. Unmounting filesystems is just as simple. Make sure you change the directory to something outside of the mounting point and use the umount command, as shown below:

[analyst@secOps ~]$ sudo umount /dev/sdb1  


a. Navigate to /home/analyst/lab.support.files/scripts/.

[analyst@secOps ~]$ cd lab.support.files/scripts/  
  


Consider the cyops.mn file as an example. Who is the owner of the file? How about the group?  
Owner: analyst; Group: analyst

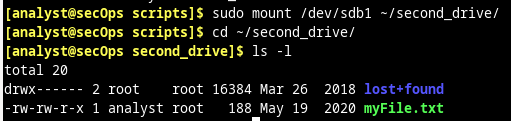
The permission for cyops.mn are –rw-r–r–. What does that mean?

The owner of the file (the analyst user) can read and write to the file but not execute it (-rw). Members of the analyst group other than the owner can only read the file (-r-), no execution or writing is allowed. All other users are not allowed to write or execute that file.



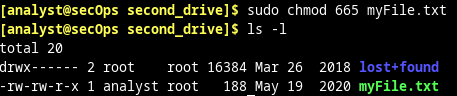
What can be done for the touch command shown above to be successful?  
The command can be executed as root (adding sudo before it) or the permissions of the /mnt directory can be modified.

d. The chmod command is used to change the permissions of a file or directory. As before, mount the /dev/sdb1 partition on the /home/analyst/second\_drive directory created earlier in this lab: [analyst@secOps ~]$ sudo mount /dev/sdb1 ~/second\_drive/



f. Use the chmod command to change the permissions of myFile.txt.

[analyst@secOps second\_drive]$ sudo chmod 665 myFile.txt

[analyst@secOps second\_drive]$ ls -l   
The chmod command takes permissions in the octal format. In that way, a breakdown of the 665 is as

follows:

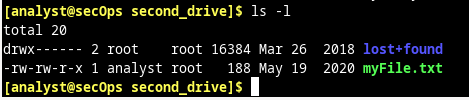
6 in octal is 110 in binary. Assuming each position of the permissions of a file can be 1 or 0, 110 means

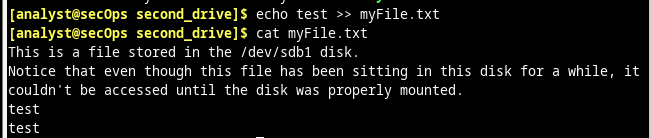
rw- (read=1, write=1 and execute=0).

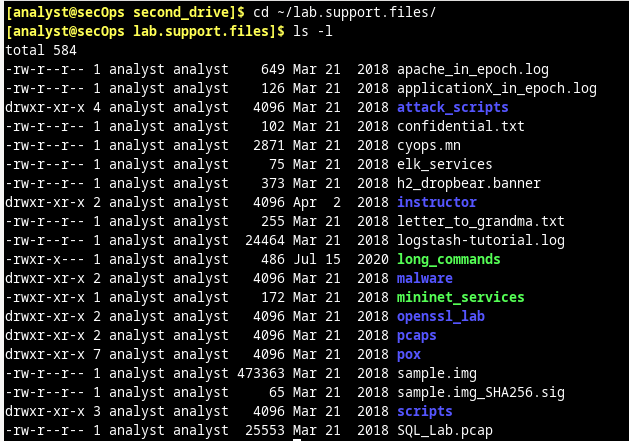
Therefore, the chmod 665 myFile.txt command changes the permissions to:

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

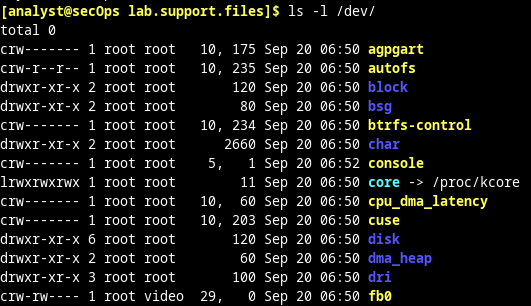
Group: rw- (6 in octal or 110 in binary)

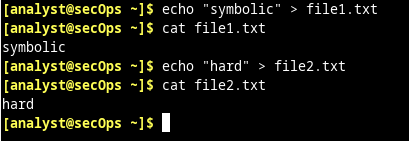
Other: r-x (5 in octal or 101 in binary) 



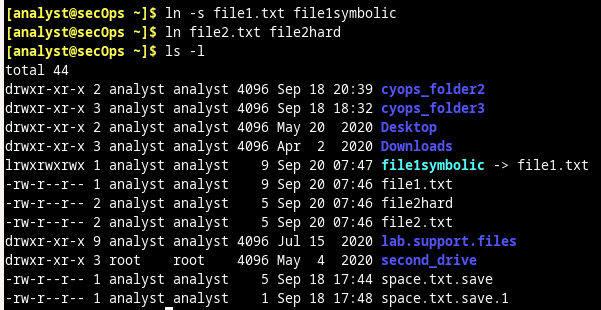


Step 1: Examine file types. a. Use the ls -l command to display the files in the /home/analyst folder. Notice the first characters of each line are either a “–“ indicating a file or a “d” indicating a directory. [analyst@secOps ~]$ ls –l  
[analyst@secOps ~]$ ls -l /dev/



  
d. Use ln –s to create a symbolic link to file1.txt, and ln to create a hard link to file2.txt:

[analyst@secOps ~]$ ln –s file1.txt file1symbolic

[analyst@secOps ~]$ ln file2.txt file2hard  


f. Change the names of the original files: file1.txt and file2.txt, and notice how it effects the linked files.

[analyst@secOps ~]$ mv file1.txt file1new.txt

[analyst@secOps ~]$ mv file2.txt file2new.txt  
