Report Linux Filesystem and Permission Settings

Parte 1: Esplorazione dei filesystem in Linux

- **Passo 1:** Accedere alla riga di comando avviando la VM CyberOps Workstation e aprendo una finestra del terminale.
- **Passo 2:** Visualizzare i filesystem attualmente montati utilizzando il comando lsblk, che elenca tutti i dispositivi a blocchi e le loro partizioni.
- **Passo 3:** Montare e smontare manualmente i filesystem. Il montaggio rende un filesystem accessibile al sistema operativo collegando una partizione fisica a una directory, nota come punto di montaggio.

Avviare la VM CyberOps Workstation e aprire una finestra del terminale.

Visualizzare i file system attualmente montati.

I filesystem devono essere *montati* prima di poter essere accessibili e utilizzati.
Montare un *filesystem* significa renderlo accessibile al sistema operativo. Montare un filesystem è il processo di collegamento della partizione fisica sul dispositivo a blocchi (disco rigido, unità SSD, chiavetta USB, ecc.) a una directory, tramite la quale è possibile accedere all'intero filesystem. Poiché la suddetta directory diventa la radice del filesystem appena montato, è anche nota come *punto di montaggio* .

a. Utilizzare il comando **lsblk** per visualizzare tutti i dispositivi a blocchi

```
[analyst@secOps /]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda      8:0    0   10G  0 disk
└─sda1   8:1    0   10G  0 part /
sdb      8:16   0    1G  0 disk
└─sdb1   8:17   0 1023M  0 part
sr0     11:0    1   57M  0 rom
[analyst@secOps /]$ mount
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
sys on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
dev on /dev type devtmpfs (rw,nosuid,relatime,size=371756k,nr_inodes=92939,mode=755)
run on /run type tmpfs (rw,nosuid,nodev,relatime,mode=755)
/dev/sda1 on / type ext4 (rw,relatime,data=ordered)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup2 on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=sy
stemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
bpf on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_
cls,net_prio)
```

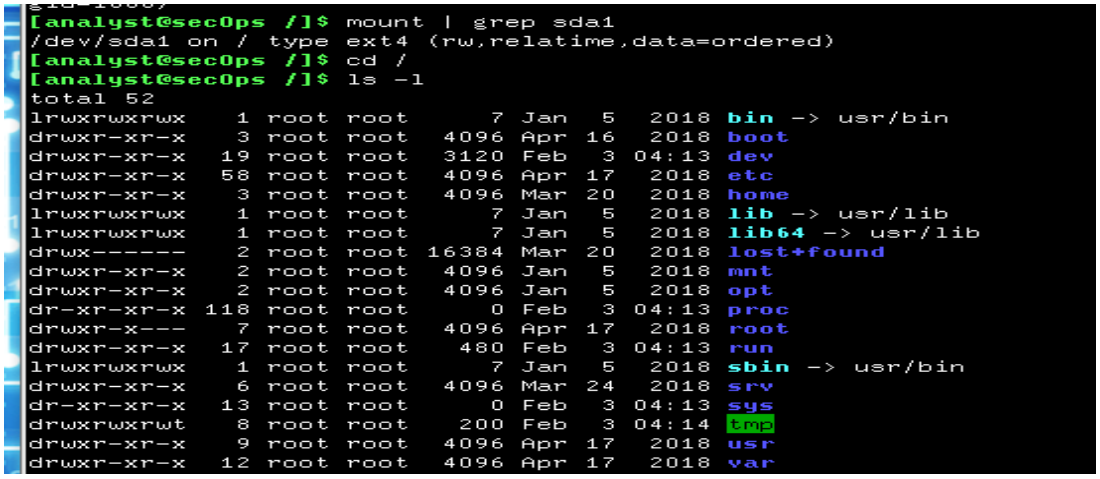
L'output mostra che la VM CyberOps Workstation ha tre dispositivi a blocchi installati: sr0, sda e sdb. L'output ad albero mostra anche le partizioni sotto sda e sdb. Convenzionalmente, /dev/sdX è usato da Linux per rappresentare i dischi rigidi, con il numero finale che rappresenta il numero di partizione all'interno di quel dispositivo. I computer con più dischi rigidi probabilmente visualizzeranno più dispositivi /dev/sdX. L'output implica che sda e sdb sono dischi rigidi, ognuno contenente una singola partizione. L'output mostra anche che sda è un disco da 10 GB mentre sdb ne ha 1 GB

b. Utilizzare il comando **mount** per visualizzare informazioni più dettagliate sui file system attualmente montati

Il filesystem root è dove è memorizzato il sistema operativo Linux stesso; tutti i programmi, gli strumenti, i file di configurazione sono memorizzati nel filesystem root per impostazione predefinita.

c. Eseguire nuovamente il comando **mount** con il pipe **|** per inviare l'output di mount a **grep** per filtrare l'output e visualizzare solo il file system root

Nell'output filtrato, mount ci mostra che il filesystem root si trova nella prima partizione del dispositivo a blocchi sda (/dev/sda1). Sappiamo che questo è il filesystem root a causa del punto di montaggio utilizzato: "/" (il simbolo della barra). L'output ci dice anche il tipo di formattazione utilizzato nella partizione, ext4 in questo caso. Le informazioni tra parentesi si riferiscono alle opzioni di montaggio della partizione

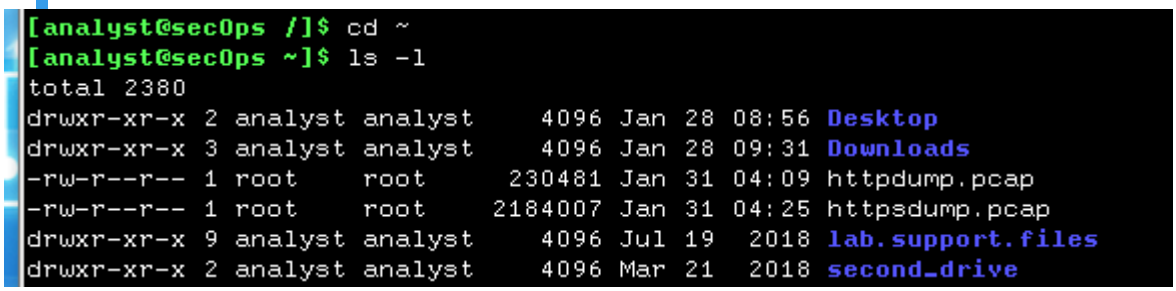


d. Emettere i comandi **cd /** e **ls -l**

Il primo comando cambia la directory nella directory root che è il livello più alto dei filesystem. Poiché /dev/sda1 è montato sulla directory root ("/"), elencando i file nella directory root, l'utente sta in realtà elencando i file fisicamente archiviati nella root del filesystem /dev/sda1.
/dev/sdb1 non viene visualizzato nell'output perché non è attualmente montato

Il comando **mount** può anche essere utilizzato per montare e smontare i filesystem. Come visto prima, la Workstation ha due dischi rigidi installati. Il primo è stato riconosciuto dal kernel come /dev/sda mentre il secondo è stato riconosciuto come /dev/sdb. Prima che un dispositivo a blocchi possa essere montato, deve avere un punto di montaggio.

1. Utilizzare i comandi **cd ~** e **ls -l** per verificare che la directory **second_drive** si trovi nella directory home dell'analista.



2. Utilizzare **ls -l** nuovamente per elencare il contenuto della directory second_drive appena creata. La directory è vuota.

```
[analyst@secOps ~]$ ls -l second_drive/
total 0
[analyst@secOps ~]$ sudo mount /dev/sdb1 ~/second_drive/
[sudo] password for analyst:
[analyst@secOps ~]$ ls -l second_drive/
total 20
drwx------ 2 root    root    16384 Mar 26  2018 lost+found
-rw-r--r-- 1 analyst analyst   183 Mar 26  2018 myFile.txt
[analyst@secOps ~]$ mount | grep /dev/sd
/dev/sda1 on / type ext4 (rw,relatime,data=ordered)
/dev/sdb1 on /home/analyst/second_drive type ext4 (rw,relatime,data=ordered)
[analyst@secOps ~]$ sudo umount /dev/sdb1
[analyst@secOps ~]$ ls -l second_drive/
total 0
```

3. Utilizzare il comando **mount** per montare **/dev/sdb1** sulla **second_drive** directory appena creata. La sintassi di mount è: **mount [opzioni] <dispositivo da montare> <punto di montaggio>.**

4. Ora che **/dev/sdb1** è stato montato su **/home/analyst/second_drive** , utilizzare **ls -l** per elencare nuovamente il contenuto della directory. Dopo il montaggio, /home/analyst/second_drive diventa il punto di accesso al file system fisicamente memorizzato in /dev/sdb1.

5. Inserire di nuovo il comando **mount** con il grep per visualizzare solo i filesystem **/dev/sdX**

6. Per smontare i filesystem, cambiare la directory in qualcosa al di fuori del punto di montaggio e usare il comando **umount**

Parte 2: Permessi dei file

Ogni file nei filesystem ha il suo set di permessi, portando sempre un set di definizioni su cosa gli utenti e i gruppi possono fare con il file.

Parte 2: Permessi dei file

a. **Passo 1:** Visualizzare e modificare i permessi dei file. In Linux, i permessi determinano chi può leggere, scrivere o eseguire un file. Utilizzando comandi come ls -l, è possibile visualizzare i permessi correnti, e con chmod è possibile modificarli.
b. **Passo 2:** Gestire i permessi delle directory. I permessi sulle directory influenzano la capacità degli utenti di accedere, modificare o eseguire file al loro interno

1. Andare su /home/analyst/lab.support.files/scripts/ e utilizzare il comando **ls -l** per visualizzare i permessi dei file

```
[analyst@secOps ~]$ cd lab.support.files/scripts/
[analyst@secOps scripts]$ ls -l
total 60
-rwxr-xr-x 1 analyst analyst  952 Mar 21  2018 configure_as_dhcp.sh
-rwxr-xr-x 1 analyst analyst 1153 Mar 21  2018 configure_as_static.sh
-rwxr-xr-x 1 analyst analyst 3459 Mar 21  2018 cyberops_extended_topo_no_fw.py
-rwxr-xr-x 1 analyst analyst 4062 Mar 21  2018 cyberops_extended_topo.py
-rwxr-xr-x 1 analyst analyst 3669 Mar 21  2018 cyberops_topo.py
-rw-r--r-- 1 analyst analyst 2871 Mar 21  2018 cyops.mn
-rwxr-xr-x 1 analyst analyst  458 Mar 21  2018 fw_rules
-rwxr-xr-x 1 analyst analyst   70 Mar 21  2018 mal_server_start.sh
drwxr-xr-x 2 analyst analyst 4096 Mar 21  2018 net_configuration_files
-rwxr-xr-x 1 analyst analyst   65 Mar 21  2018 reg_server_start.sh
-rwxr-xr-x 1 analyst analyst  189 Mar 21  2018 start_ELK.sh
-rwxr-xr-x 1 analyst analyst   85 Mar 21  2018 start_miniedit.sh
-rwxr-xr-x 1 analyst analyst   76 Mar 21  2018 start_pox.sh
-rwxr-xr-x 1 analyst analyst  106 Mar 21  2018 start_snort.sh
-rwxr-xr-x 1 analyst analyst   61 Mar 21  2018 start_tftpd.sh
```

2. Il comando **touch** permette la creazione rapida di un file di testo vuoto. Il file però non è stato creato perché il comando puo essere eseguito come root (aggiungendo sudo prima) o modificando i permessi nella directory /mnt.

```
[analyst@secOps scripts]$ touch /mnt/myNewFile.txt
touch: cannot touch '/mnt/myNewFile.txt': Permission denied
[analyst@secOps scripts]$ ls -ld /mnt
drwxr-xr-x 2 root root 4096 Jan  5  2018 /mnt
[analyst@secOps scripts]$ sudo mount /dev/sdb1 ~/second_drive/
[analyst@secOps scripts]$ cd ~/second_drive
[analyst@secOps second_drive]$ ls -l
total 20
drwx------ 2 root    root    16384 Mar 26  2018 lost+found
-rw-r--r-- 1 analyst analyst   183 Mar 26  2018 myFile.txt
```

3. Il comando **chmod** viene utilizzato per modificare i permessi di un file o di una directory. Montare la partizione **/dev/sdb1** sulla directory **/home/analyst/second_drive** creata in precedenza. Passare alla **second_drive** directory ed elencarne il contenuto

4. Utilizzare il comando **chmod** per modificare i permessi di **myFile.txt** (sono diventati -rw-rw-rx). Il comando **chmod** accetta i permessi in formato ottale. In questo modo, una ripartizione del 665 è la seguente: 6 in ottale è 110 in binario. Supponendo che ogni posizione dei permessi di un file possa essere 1 o 0, 110 significa rw- (lettura=1, scrittura=1 ed esecuzione=0). Pertanto, il comando **chmod 665 myFile.txt** modifica i permessi in:

Proprietario: rw- (6 in ottale o 110 in binario)
Gruppo: rw- (6 in ottale o 110 in binario)
Altro: rx (5 in ottale o 101 in binario)

5. Il comando **chown** è usato per cambiare la proprietà di un file o di una directory. Inserire il comando per rendere root il proprietario di **myFile.txt.**

```
[analyst@secOps second_drive]$ sudo chmod 665 myFile.txt
[analyst@secOps second_drive]$ ls -l
total 20
drwx------ 2 root    root    16384 Mar 26  2018 lost+found
-rw-rw-r-x 1 analyst analyst   183 Mar 26  2018 myFile.txt
[analyst@secOps second_drive]$ sudo chown analyst myFile.txt
[analyst@secOps second_drive]$ ls -l
total 20
drwx------ 2 root    root    16384 Mar 26  2018 lost+found
-rw-rw-r-x 1 analyst analyst   183 Mar 26  2018 myFile.txt
```

6. Ora che l'analista è il proprietario del file, provare ad aggiungere la parola "test" alla fine di **myFile.txt.** L'operazione è riuscita perché analyst è il proprietario del file e i permessi sono ancora impostati su 665 e quindi consentono al proprietario e agli utenti del gruppo di apportare modifiche.

```
[analyst@secOps second_drive]$ echo test >> myFile.txt
[analyst@secOps second_drive]$ cat myFile.txt
This is a file stored in the /dev/sdb1 disk.
Notice that even though this file has been sitting in this disk for a while, it couldn't
 accessed until the disk was properly mounted.
test
```

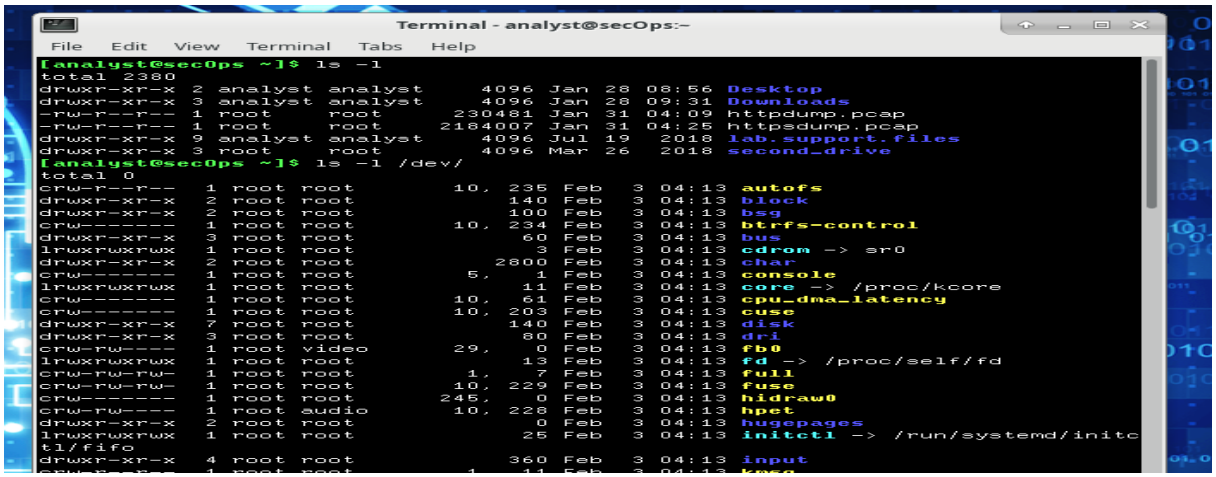
Collegamenti simbolici e altri tipi di file speciali

Parte 3: Link simbolici e altri tipi di file speciali

Passo 1: Esaminare i tipi di file. Oltre ai file regolari, Linux supporta link simbolici, che sono puntatori a un altro file o directory. Questi possono essere creati utilizzando il comando ln -s.

Esaminare i tipi di file.

a. Utilizzare il comando **ls –l** per visualizzare i file nella cartella /home/analyst e poi **ls –l /dev/** per produrre un elenco della directory /dev/

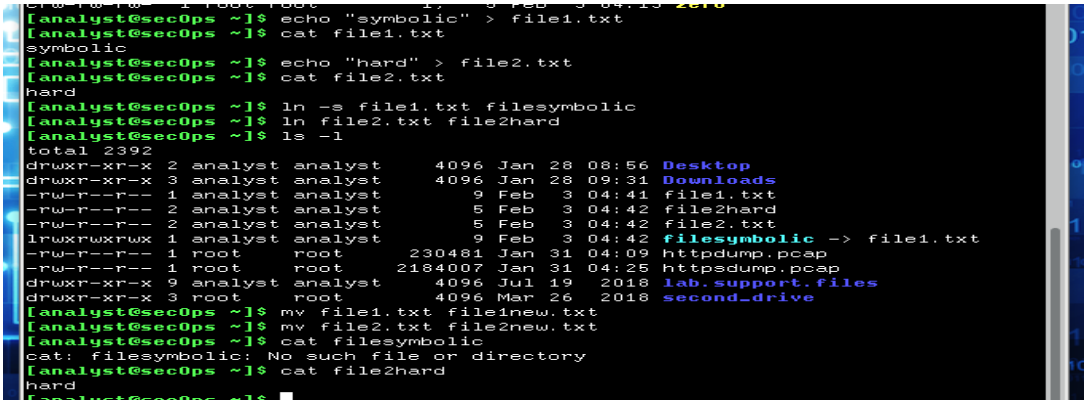


Ci sono due tipi di link in Linux: link simbolici e hard link. La differenza tra link simbolici e hard link è che un file di link simbolico punta al nome di un altro file e un file di hard link punta al contenuto di un altro file.

b. Utilizzare **ln –s** per creare un collegamento simbolico a file1.txt e **ln** per creare un collegamento hard a file2.txt. Utilizzare il comando **ls –l** ed esaminare l'elenco delle directory

c. Il file **filesymbolic** è un collegamento simbolico con una **l** all'inizio della riga e un puntatore -> a **file1.txt** . Il **file2hard** sembra essere un file normale e punta agli stessi attributi e alla stessa posizione del blocco del disco come **file2.txt**

d. Cambiare i nomi dei file originali: file1.txt e file2.txt, e notare l'effetto sui file collegati



file1symbolic è ora un collegamento simbolico non funzionante perché il nome del file a cui puntava **file1.txt** è cambiato, ma il file di collegamento fisso **file2hard** funziona ancora correttamente perché punta all'inode di **file2.txt** e non al suo nome, che ora è **file2new.txt**. Se si modificasse il testo a **file2hard.txt** si modificherebbe il contenuto anche a **file2new.txt.**