Navigating the Linux Filesystem and Permission Settings

PARTE 1)

Lanciamo la workstation, apriamo il prompt e lanciamo il comando lsblk per vedere le unità montate

```
[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 1024M  0 rom
[analyst@secOps ~]$
```

Usando il comando mount, possiamo vedere informazioni più specifiche:

```
[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=500780k,nr_inodes=125195,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=systemd)
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/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
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/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime,pagesize=2M)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=37,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=10755)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
tmpfs on /tmp type tmpfs (rw,nosuid,nodev)
mqueue on /dev/mqueue type mqueue (rw,relatime)
configfs on /sys/kernel/config type configfs (rw,relatime)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=101288k,mode=700,uid=1000,gid=1000)
```

Ora usiamo il comando mount | grep sda1 per avere come output solo il root filesystem:

```
[analyst@secOps ~]$ mount | grep sda1
/dev/sda1 on / type ext4 (rw,relatime,data=ordered)
```

Lanciamo ora i comandi cd / e ls -l :

```
[analyst@secOps ~]$ cd /
[analyst@secOps /]$ ls -l
total 52
lrwxrwxrwx   1 root root     7 Jan  5  2018 bin -> usr/bin
drwxr-xr-x   3 root root  4096 Apr 16  2018 boot
drwxr-xr-x  19 root root  3140 Oct 28 04:29 dev
drwxr-xr-x  58 root root  4096 Oct 23 05:00 etc
drwxr-xr-x   3 root root  4096 Mar 20  2018 home
lrwxrwxrwx   1 root root     7 Jan  5  2018 lib -> usr/lib
lrwxrwxrwx   1 root root     7 Jan  5  2018 lib64 -> usr/lib
drwx------   2 root root 16384 Mar 20  2018 lost+found
drwxr-xr-x   2 root root  4096 Jan  5  2018 mnt
drwxr-xr-x   2 root root  4096 Jan  5  2018 opt
dr-xr-xr-x 120 root root     0 Oct 28 04:29 proc
drwxr-x---   7 root root  4096 Oct 23 05:30 root
drwxr-xr-x  17 root root   480 Oct 28 04:29 run
lrwxrwxrwx   1 root root     7 Jan  5  2018 sbin -> usr/bin
drwxr-xr-x   6 root root  4096 Mar 24  2018 srv
dr-xr-xr-x  13 root root     0 Oct 28 04:29 sys
drwxrwxrwt   8 root root   200 Oct 28 04:29 tmp
drwxr-xr-x   9 root root  4096 Apr 17  2018 usr
drwxr-xr-x  12 root root  4096 Apr 17  2018 var
```

Il primo comando ci porta alla cartella principale del sistema(root), il secondo ci mostra tutti i file e cartelle contenuti in esso, con i vari permessi.

/dev/sdb1 non è montato.

Andiamo a verificare che il secondo drive si trovi nella cartella dell'analyst:

```
[analyst@secOps ~]$ cd ~
[analyst@secOps ~]$ ls -l
total 14244
drwxr-xr-x 2 analyst analyst    4096 Mar 22  2018 Desktop
drwxr-xr-x 3 analyst analyst    4096 Mar 22  2018 Downloads
-rw-r--r-- 1 root    root    8426196 Oct 25 04:49 httpdump.pcap
-rw-r--r-- 1 root    root    6132163 Oct 25 21:43 httpsdump.pcap
drwxr-xr-x 9 analyst analyst    4096 Jul 19  2018 lab.support.files
-rw-r--r-- 1 root    root         33 Oct 23 05:28 my_tftp_data
drwxr-xr-x 2 analyst analyst    4096 Mar 21  2018 second_drive
```

Verifichiamo che la cartella è vuota:

```
[analyst@secOps ~]$ ls -l second_drive/
total 0
[analyst@secOps ~]$
```

Ora usiamo il comando sudo mount /dev/sdb1 ~/second_drive/ per caricare /dev/sdb1 nella cartella second_drive e verifichiamone il contenuto:

```
[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
```

Ora la cartella contiene l'accesso ai dati del filesystem che sono fisicamente presenti in /dev/sdb1.

Usando mount | grep /dev/sd andiamo a recuperare informazioni sulla partizione /dev/sdb1:

```
[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)
```

Con il comando sudo umount /dev/sdb1 invece smontiamo l'unità:

```
[analyst@secOps ~]$ sudo umount /dev/sdb1
[sudo] password for analyst:
[analyst@secOps ~]$ ls -l second_drive/
total 0
[analyst@secOps ~]$
```

PARTE 2)

Navighiamo nella cartella /home/analyst/lab.support.files/scripts/ e con ls -l vediamo tutti i file e cartelle con i vari permessi:

```
[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
```

Prendendo come esempio cyops.mn, possiamo vedere che il proprietario e il gruppo sono analyst, e che ci sono permessi di lettura e scrittura per l'utente (analyst in questo caso), mentre per gruppo e altri utenti i permessi sono solamente di lettura.

Proviamo a creare un file nella cartella mnt con il comando touch:

```
[analyst@secOps scripts]$ touch /mnt/myNewFile.txt
touch: cannot touch '/mnt/myNewFile.txt': Permission denied
```

Riceviamo un errore, proviamo quindi col comando ls -ld /mnt a verificare i permessi della parent directory:

```
[analyst@secOps scripts]$ ls -ld /mnt
drwxr-xr-x 2 root root 4096 Jan  5  2018 /mnt
[analyst@secOps scripts]$
```

Possiamo vedere che i permessi di scrittura sono solamente per l'utente root, per creare il file quindi dovremmo usare lo stesso comando usando sudo.

Andiamo ora a ricaricare dev/sdb1 nella cartella second_drive:

```
[analyst@secOps scripts]$ sudo mount /dev/sdb1 ~/second_drive/
[sudo] password for analyst:
[analyst@secOps scripts]$
```

Andiamo a visualizzare il contenuto della cartella:

```
[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
```

I permessi per il file myFile.txt sono lettura e scrittura per l'utente analyst, e solo di lettura per gruppo e altri utenti.

Usiamo ora sudo chmod 665 myFile.txt per cambiare i permessi al file:

```
[analyst@secOps second_drive]$ sudo chmod 665 myFile.txt
[sudo] password for analyst:
[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
```

Ora il file ha permessi di lettura e e scrittura per utente e gruppo, mentre ha permessi di lettura ed esecuzione per tutti gli altri utenti.

Usando il comando sudo chmod 777 myFile.txt tutti gli utenti avrebbero pieno accesso al file.

Cambiamo ora la proprietà del file con il comando chown, usando sudo chown root 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 root analyst   183 Mar 26  2018 myFile.txt
```

Andiamo ad usare il comando echo test >> myFile.txt e stampiamo il risultato:

```
[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
test
```

Dopo il contenuto del testo, siamo riusciti ad aggiungere "test" in fondo al file, questo perché il gruppo analyst ha permessi di scrittura e lettura.

Torniamo nella cartella cd ~/lab.support.files/ e visualizziamo tutti i file contenuti:

```
[analyst@secOps second_drive]$ cd ~/lab.support.files/
[analyst@secOps lab.support.files]$ ls -l
total 580
-rw-r--r-- 1 analyst analyst    649 Mar 21  2018 apache_in_epoch.log
-rw-r--r-- 1 analyst analyst    126 Mar 21  2018 applicationX_in_epoch.log
drwxr-xr-x 4 analyst analyst   4096 Mar 21  2018 attack_scripts
-rw-r--r-- 1 analyst analyst    102 Mar 21  2018 confidential.txt
-rw-r--r-- 1 analyst analyst   2871 Mar 21  2018 cyops.mn
-rw-r--r-- 1 analyst analyst     75 Mar 21  2018 elk_services
-rw-r--r-- 1 analyst analyst    373 Mar 21  2018 h2_dropbear.banner
drwxr-xr-x 2 analyst analyst   4096 Apr  2  2018 instructor
-rw-r--r-- 1 analyst analyst    255 Mar 21  2018 letter_to_grandma.txt
-rw-r--r-- 1 analyst analyst  24464 Mar 21  2018 logstash-tutorial.log
drwxr-xr-x 2 analyst analyst   4096 Mar 21  2018 malware
-rwxr-xr-x 1 analyst analyst    172 Mar 21  2018 mininet_services
drwxr-xr-x 2 analyst analyst   4096 Mar 21  2018 openssl_lab
drwxr-xr-x 2 analyst analyst   4096 Mar 21  2018 pcaps
drwxr-xr-x 7 analyst analyst   4096 Mar 21  2018 pox
-rw-r--r-- 1 analyst analyst 473363 Mar 21  2018 sample.img
-rw-r--r-- 1 analyst analyst     65 Mar 21  2018 sample.img_SHA256.sig
drwxr-xr-x 3 analyst analyst   4096 Mar 21  2018 scripts
-rw-r--r-- 1 analyst analyst  25553 Mar 21  2018 SQL_Lab.pcap
[analyst@secOps lab.support.files]$
```

Se all'inizio dei permessi vediamo "d", ciò significa che quella determinata riga si riferisce a una directory.

Torniamo nella cartella /home/analyst ed esaminiamo i tipi di file anche qui:

```
[analyst@secOps lab.support.files]$ cd /home/analyst
[analyst@secOps ~]$ ls -l
total 14244
drwxr-xr-x 2 analyst analyst    4096 Mar 22  2018 Desktop
drwxr-xr-x 3 analyst analyst    4096 Mar 22  2018 Downloads
-rw-r--r-- 1 root    root    8426196 Oct 25 04:49 httpdump.pcap
-rw-r--r-- 1 root    root    6132163 Oct 25 21:43 httpsdump.pcap
drwxr-xr-x 9 analyst analyst    4096 Jul 19  2018 lab.support.files
-rw-r--r-- 1 root    root         33 Oct 23 05:28 my_tftp_data
drwxr-xr-x 3 root    root       4096 Mar 26  2018 second_drive
```

Facciamo ora una lista della cartella /dev:

```
crw-rw-rw-  1 root tty          5,    2 Oct 28 06:01 ptmx
drwxr-xr-x  2 root root               0 Oct 28 04:29 pts
crw-rw-rw-  1 root root         1,    8 Oct 28 04:29 random
crw-rw-r--  1 root rfkill      10,   52 Oct 28 04:29 rfkill
lrwxrwxrwx  1 root root               4 Oct 28 04:29 rtc -> rtc0
crw-rw----  1 root audio      250,    0 Oct 28 04:29 rtc0
brw-rw----  1 root disk         8,    0 Oct 28 04:29 sda
brw-rw----  1 root disk         8,    1 Oct 28 04:29 sda1
```

Evidenziamo questa sezione, dove possiamo notare la prima lettera "c" "l" e "b":

- L = un collegamento che punta a un altro file o directory
- C = dispositivo a caratteri, un dispositivo che gestisce un carattere alla volta (es. tastiera)
- B = dispositivo a blocchi, un dispositivo che gestisce dati in blocchi di dati fissi, come ad esempio SSD

Andiamo a creare due file:

```
[analyst@secOps ~]$ echo "symbolic" > file1.txt
[analyst@secOps ~]$ cat file1.txt
symbolic
[analyst@secOps ~]$ echo "hard" > file2.txt
[analyst@secOps ~]$ cat file2.txt
hard
```

Con ln –s file1.txt file1symbolic andiamo a creare un symbolic link a file1.txt
Con ln file2.txt file2hard andiamo a creare un hard link a file2.txt
Il symbolic link punta al file, mentre l'hard link punta direttamente ai dati.

```
[analyst@secOps ~]$ ln -s file1.txt file1symbolic
[analyst@secOps ~]$ ln file2.txt file2hard
[analyst@secOps ~]$ ls -l
total 14256
drwxr-xr-x 2 analyst analyst    4096 Mar 22  2018 Desktop
drwxr-xr-x 3 analyst analyst    4096 Mar 22  2018 Downloads
lrwxrwxrwx 1 analyst analyst       9 Oct 28 06:20 file1symbolic -> file1.txt
-rw-r--r-- 1 analyst analyst       9 Oct 28 06:16 file1.txt
-rw-r--r-- 2 analyst analyst       5 Oct 28 06:16 file2hard
-rw-r--r-- 2 analyst analyst       5 Oct 28 06:16 file2.txt
-rw-r--r-- 1 root    root    8426196 Oct 25 04:49 httpdump.pcap
-rw-r--r-- 1 root    root    6132163 Oct 25 21:43 httpsdump.pcap
drwxr-xr-x 9 analyst analyst    4096 Jul 19  2018 lab.support.files
-rw-r--r-- 1 root    root         33 Oct 23 05:28 my_tftp_data
drwxr-xr-x 3 root    root       4096 Mar 26  2018 second_drive
```

Proviamo ora a modificare il nome dei file:

```
[analyst@secOps ~]$ mv file1.txt file1new.txt
[analyst@secOps ~]$ mv file2.txt file2new.txt
[analyst@secOps ~]$ cat file1symbolic
cat: file1symbolic: No such file or directory
[analyst@secOps ~]$ cat file2hard
hard
```

Possiamo vedere che il symbolic link non funziona più, perché punta al nome di un file che è stato però modificato, mentre l'hard link funziona ancora perché punta al contenuto del file stesso.
Se modificassimo il contenuto del file con hard link, modificheremmo anche il contenuto del collegamento hard link, visto che puntano allo stesso punto della memoria.