

Buildweek_III_Linux_Permissions

Consegna

Obiettivi

In questo laboratorio, prenderai familiarità con i filesystem Linux.

- Parte 1 Esplorare i Filesystem in Linux
- Parte 2 Permessi dei File
- Parte 3 Link Simbolici e Altri Tipi di File Speciali

Risorse Richieste

- VM CyberOps Workstation

Svolgimento

Qual è il significato dell'output? Dove sono fisicamente memorizzati i file elencati? Perché `/dev/sdb1` non viene mostrato nell'output sopra?

Il comando mostra tutti gli elementi presenti all'interno della directory root nella quale ci siamo spostati precedentemente tramite `cd /`. L'output non mostra `/dev/sdb1` in quanto rappresenta una **partizione fisica** rilevata dal kernel, ma di default non viene mostrata in `ls /` perché non è automaticamente montata in un punto della gerarchia del filesystem.

In altre parole `/dev/sdb1` è solo un file speciale che identifica la partizione (presente nella directory `/dev`).

Finché non viene montato, i suoi contenuti non sono accessibili nella gerarchia `/`; solo dopo il mount, la partizione apparirà come directory con i suoi file all'interno del punto di mount scelto.

```
[analyst@secOps ~]$ cd /
[analyst@secOps /]$ ls -l
total 52
lrwxrwxrwx   1 root root      7 May  3 15:26 bin -> usr/bin
drwxr-xr-x   3 root root 4096 Jun 18 19:07 boot
drwxr-xr-x  20 root root 3920 Oct  1 03:53 dev
drwxr-xr-x  73 root root 4096 Jun 19 04:45 etc
drwxr-xr-x   3 root root 4096 Mar 20  2018 home
lrwxrwxrwx   1 root root      7 May  3 15:26 lib -> usr/lib
lrwxrwxrwx   1 root root      7 May  3 15:26 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   3 root root 4096 Jun 17 15:07 opt
dr-xr-xr-x 212 root root    0 Oct  1 03:53 proc
drwxr-xr-x   9 root root 4096 Sep 23 07:58 root
drwxr-xr-x  22 root root  580 Oct  1 03:53 run
lrwxrwxrwx   1 root root      7 May  3 15:26/sbin -> usr/bin
drwxr-xr-x   6 root root 4096 Mar 24  2018 srv
dr-xr-xr-x  13 root root    0 Oct  1 03:53 sys
drwxrwxrwt  11 root root  260 Oct  1 03:56 tmp
drwxr-xr-x  10 root root 4096 Jun 19 03:15 usr
drwxr-xr-x  12 root root 4096 Jun 19 04:45 var
[analyst@secOps /]$
```

Perché la directory non è più vuota? Dove sono fisicamente memorizzati i file elencati?

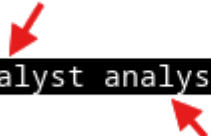
La directory non è più vuota in quanto ha "accolto" i file che erano memorizzati all'interno di `/dev/sdb1`. Una volta montato `sdb1` all'interno della cartella `second_drive` questi file diventano visibili e vi si può normalmente operare.

```
[analyst@secOps ~]$ 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
```

Considera il file `cyops.mn` come esempio. Chi è il proprietario del file? E il gruppo?

Il proprietario del file è `analyst` ed il gruppo che lo gestisce è anch'esso chiamato `analyst`.

```
-rw-r--r-- 1 analyst analyst 2871 Mar 21  2018 cyops.mn
```



I permessi per cyops.mn sono -rw-r-r-. Cosa significa?

I permessi indicati nel file significano che il proprietario ha i permessi di lettura e scrittura "rw", il gruppo analyst e chi ne fa parte hanno soltanto il permesso di lettura "r", tutti gli altri utenti hanno anch'essi solamente il permesso di lettura "r".

Perché il file non è stato creato? Elenca i permessi, la proprietà e il contenuto della directory /mnt e spiega cosa è successo. Con l'aggiunta dell'opzione -d, elenca i permessi della directory genitore. Registra la risposta nelle righe sottostanti.

Non è possibile crearvi un file all'interno con i permessi dello user analyst in quanto la cartella mnt è gestita da root e tutti gli altri utenti non hanno i permessi di scrittura ma solamente di esecuzione "x".

```
drwxr-xr-x  2 root root 4096 Jan  5  2018 mnt
```

Cosa si può fare affinché il comando touch mostrato sopra abbia successo?

Perché il comando precedente abbia successo è possibile concedere temporaneamente i permessi root all'utente analyst tramite sudo.

Eseguendo il comando `sudo touch /mnt/myNewFile.txt` ed inserendo la relativa password sarà possibile ottenere un upgrade dei privilegi che ci permetterà di scrivere all'interno della cartella.

Quali sono i permessi del file myFile.txt?

I permessi del file appena creato sono di lettura e scrittura per il proprietario del file e di sola lettura per tutti gli altri utenti.

Il file eredita proprietario e gruppi dalla cartella madre nella quale è stato creato, in questo caso root.

```
-rw-r--r-- 1 root root 0 Oct  1 04:38 myNewFile.txt
```

Usa il comando `chmod 665 /mnt/myNewFile.txt`. I permessi sono cambiati? Quali sono i permessi di myFile.txt?

I permessi sono variati solo per il gruppo root e gli altri users. E' stato attribuito il permesso di lettura e **scrittura** al gruppo root; per gli altri users è stato aggiunto il permesso di **esecuzione** lasciando invariato quello di lettura. Il proprietario del file mantiene in questo caso i permessi precedenti di lettura e scrittura.

```
-rw-rw-r-x 1 root root 0 Oct  1 04:38 myNewFile.txt
```

Quale comando cambierebbe i permessi di myFile.txt a rwxrwxrwx, garantendo a qualsiasi utente nel sistema pieno accesso al file?

Il comando da utilizzare per garantire a tutti indiscriminatamente i permessi di lettura, scrittura ed esecuzione è il seguente: `sudo chmod 777 myNewFile.txt`.

Ora che analyst è il proprietario del file, prova ad accodare la parola 'test' alla fine di myFile.txt. L'operazione è riuscita? Spiega.

L'operazione è riuscita in quanto, avendo dato l'ownership del file ad analyst, l'utente non ha più la necessità di richiedere i privilegi root per svolgere l'operazione ed è quindi stato possibile scrivere all'interno di myNewFile.txt.

```
-rw-rw-r-x 1 analyst root 0 Oct  1 04:38 myNewFile.txt
[analyst@secOps mnt]$ echo test >> myNewFile.txt
[analyst@secOps mnt]$ cat myNewFile.txt
test
```

Qual è la differenza tra la parte iniziale della riga di malware e la riga di mininet_services?

L'unica differenza sostanziale è rappresentata dalla presenza di una "d" all'inizio di malware; questa lettera indica semplicemente che siamo dinanzi ad una directory e non ad un file "-".

```
drwxr-xr-x 2 analyst analyst 4096 May 25 13:01 malware
-rwxr-xr-x 1 analyst analyst 172 Jul 25 16:27 mininet_services
```

Cosa pensi succederebbe a `file2hard` se aprissi un editor di testo e cambiassi il testo in `file2new.txt`?

Se si apre `file2new.txt` con un editor di testo e ne si modifica il contenuto, i blocchi di dati su disco associati all'inode vengono aggiornati.

Poiché `file2hard` è un hard link che punta allo stesso inode, esso continuerà a leggere e scrivere sugli stessi blocchi fisici. In altre parole:

- Qualsiasi modifica fatta a `file2new.txt` si rifletterà automaticamente anche in `file2hard`.
- Non ci sarà duplicazione di file: entrambi i nomi (`file2hard` e `file2new.txt`) sono solo "etichette diverse" per lo stesso contenuto.