BW3 - Esercizio 3

Analisi dello Storage e Gestione dei Collegamenti nel File System Linux

Autore: Cybereagles

Sintesi

Il presente report documenta un'attività di analisi e amministrazione del *file system* all'interno di un ambiente Linux. L'obiettivo è esplorare gli strumenti diagnostici nativi per l'identificazione dell'hardware di archiviazione e comprendere a fondo le logiche di collegamento dei file. L'indagine mappa i dispositivi a blocchi e le partizioni, esaminando nel dettaglio la gestione dei permessi, le operazioni di montaggio e le differenze strutturali tra i collegamenti fisici (**Hard Link**) e i collegamenti simbolici (**Symbolic Link**).

Scopo del test e analisi dello scenario

Scenario e Obiettivi

L'attività si è svolta all'interno di un ambiente Linux controllato (utente *analyst@secOps*). L'obiettivo principale è fornire una panoramica chiara dell'interazione tra la rappresentazione logica dei dati (i nomi dei file) e la loro effettiva allocazione fisica sul disco (gli inode), analizzando parallelamente la struttura dei permessi di sistema.

- VM: Macchina locale (Linux SecOps) con focus sull'analisi del partizionamento e dei permessi.

Strumenti

- **lsblk**: Strumento da riga di comando utilizzato per elencare i dispositivi a blocchi, mostrando come è suddiviso il disco rigido, i dispositivi non montati, le dimensioni, i tipi e i relativi *mountpoints*.
- **mount**: Utility impiegata per visualizzare o montare i *file system* e le partizioni, indicando la modalità di accesso (es. *rw* per Lettura-Scrittura) e opzioni di ottimizzazione come *relatime*.
- **ls**: Strumento essenziale per l'elencazione dei file e l'ispezione della Gerarchia del Filesystem (FHS), utilizzato per analizzare permessi, proprietari e tipo di oggetti (directory, file regolari, link simbolici).

Svolgimento

Fase 1: Mappatura dei dispositivi a blocchi

Nella prima fase, l'indagine si è concentrata sulla mappatura dei dispositivi a blocchi e delle partizioni. È stato utilizzato il comando `$ lsblk` per visualizzare la topologia dei dischi fisici. Si è osservato che il disco ***sda*** è suddiviso nella partizione operativa ***sda1***, montata sulla directory ***root* (/)**, mentre il disco secondario ***sdb*** contiene la partizione ***sdb1*** non ancora montata dal sistema.

```
$ lsblk
```

```
[analyst@secOps ~]$ lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
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
```

Figura 1 Output del comando lsblk che mostra la gerarchia ad albero dei dischi sda e sdb, evidenziando le partizioni e i relativi mountpoints.

Fase 2: Analisi dei punti di montaggio e opzioni del file system

Successivamente, si è proceduto ad analizzare le partizioni montate e le relative opzioni. Tramite l'output filtrato, è stato confermato che il *file system* radice si trova in ***/dev/sda1*** ed è formattato in ***ext4***. Le opzioni di montaggio rilevate includono *rw* (Lettura e Scrittura) e *relatime* (Relative Access Time), un parametro che riduce le scritture non necessarie sul disco per l'aggiornamento dei *timestamp*, ottimizzandone la reattività.

```
$ mount | grep sda1
```

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

Figura 2 Dettaglio del comando mount filtrato per la partizione sda1, indicante il tipo ext4 e le opzioni di montaggio rw e relatime.

Fase 3: Esplorazione della Gerarchia del Filesystem e verifica Permessi

L'analisi si è spostata all'interno della directory radice per esaminare lo scheletro del sistema operativo. Si è notato che un tentativo di creazione di un nuovo file nella **directory *./mnt*** è fallito restituendo un errore "Permission denied". Questo ha confermato che l'utente corrente non disponeva dei permessi di scrittura (*w*) su tale directory, protetta e di proprietà dell'utente ***root***.

```
$ ls -l
```

```
$ touch /mnt/myNewFile.txt
```

```
$ ls -ld /mnt
```

```
[analyst@secOps ~]$ cd /
[analyst@secOps /]$ ls -l
total 52
lrwxrwxrwx   1 root root     7 May  3  2025 bin -> usr/bin
drwxr-xr-x   3 root root  4096 Jun 18  2025 boot
drwxr-xr-x  20 root root  3920 Feb 23 08:41 dev
drwxr-xr-x  73 root root  4096 Jun 19  2025 etc
drwxr-xr-x   3 root root  4096 Mar 20  2018 home
lrwxrwxrwx   1 root root     7 May  3  2025 lib -> usr/lib
lrwxrwxrwx   1 root root     7 May  3  2025 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  2025 opt
dr-xr-xr-x 202 root root     0 Feb 23 08:41 proc
drwxr-x---   8 root root  4096 Jun 18  2025 root
drwxr-xr-x  22 root root   600 Feb 23 08:51 run
lrwxrwxrwx   1 root root     7 May  3  2025 sbin -> usr/bin
drwxr-xr-x   6 root root  4096 Mar 24  2018 srv
dr-xr-xr-x  13 root root     0 Feb 23 08:41 sys
drwxrwxrwt  11 root root   260 Feb 23 09:03 tmp
drwxr-xr-x  10 root root  4096 Jun 19  2025 usr
drwxr-xr-x  12 root root  4096 Jun 19  2025 var
[analyst@secOps scripts]$ ls -ld /mnt
drwxr-xr-x 2 root root 4096 Jan  5  2018 /mnt
[analyst@secOps scripts]$ touch /mnt/myNewFile.txt
touch: cannot touch '/mnt/myNewFile.txt': Permission denied
```

Figura 3 Output dei comandi ls -l e ls -ld nella root e tentata creazione del file in /mnt con esito "Permission denied".

Fase 4: Montaggio del secondo disco e manipolazione dei file

Per rendere accessibili i dati sul secondo disco, la **partizione *./dev/sdb1*** è stata montata manualmente. È stata poi eseguita la modifica dei permessi di un file specifico (*myFile.txt*) utilizzando il formato numerico ottale tramite **l'utility chmod**. I permessi sono stati aggiornati a ***665* (*-rw-rw-r-x*)**, consentendo l'accesso in scrittura anche al gruppo proprietario. Successivamente, l'esito dell'operazione è stato testato con successo scrivendo all'interno del file.

```
$ sudo mount /dev/sdb1 ~/second_drive/
```

```
$ sudo chmod 665 myFile.txt
```

```
$ echo test >> myFile.txt$ cat myFile.txt
```

```
[analyst@secOps scripts]$ sudo mount /dev/sdb1 ~/second_drive/
[sudo] password for analyst:
Sorry, try again.
[sudo] password for analyst:
[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
[analyst@secOps second_drive]$ sudo chmod 665 myFile.txt
```

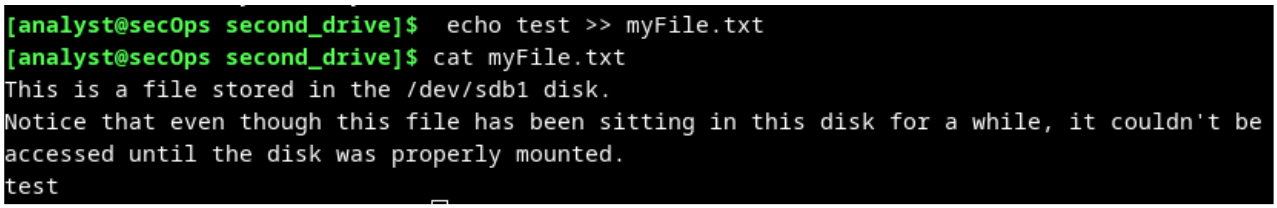

Figura 4 Esecuzione del montaggio, elevazione dei permessi con chmod 665 e inserimento confermato del testo all'interno del file.

Fase 5: Gestione avanzata dei collegamenti (Hard Link e Symbolic Link)

L'ultima fase ha esaminato sperimentalmente le differenze strutturali tra i tipi di collegamento file. Sono stati creati un collegamento simbolico e un collegamento fisico. È stato dimostrato che l'**Hard Link** opera direttamente a livello di ***inode***, puntando ai dati grezzi condivisi e garantendo totale resilienza alle modifiche del nome del file originale. Al contrario, il **Link Simbolico** si è rivelato come una semplice scorciatoia intrinsecamente vulnerabile alla rottura qualora il file sorgente venga spostato o rinominato.

```
$ ln –s file1.txt file1symbolic
```

```
$ ln file2.txt file2hard
```

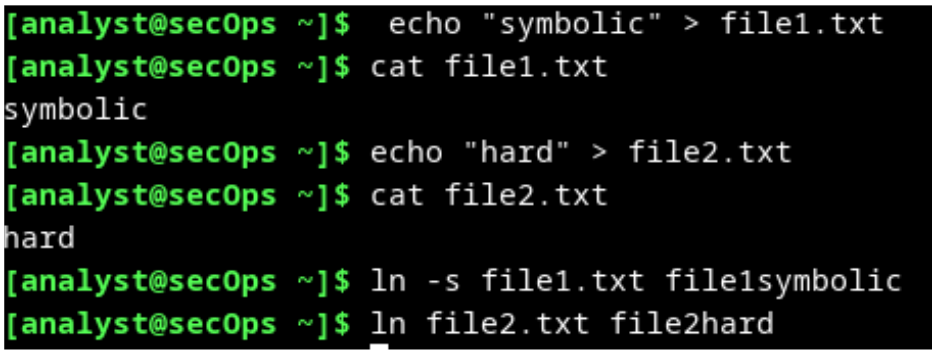

Figura 5 Creazione dei link e dimostrazione pratica della resilienza dell'Hard Link rispetto all'interruzione del Link Simbolico.

Conclusioni

L'attività pratica ha permesso di consolidare le competenze relative alla gestione dello storage in ambiente Linux. L'ispezione tramite **lsblk** e **mount** si è dimostrata uno step essenziale per distinguere tra partizioni operative e dispositivi collegati ma non montati. Inoltre, si è validato che gli **Hard Link** operano in modo resiliente tramite inode, a differenza dei **Link Simbolici** che risultano suscettibili alle rinomine. La solida comprensione di questi meccanismi architetturali risulta un requisito fondamentale per operare in sicurezza nell'organizzazione dei dati e nelle procedure di backup.

Si raccomandano le seguenti azioni di mitigazione:

- **Principio del Least Privilege sui File di Sistema**: Assicurarsi che le directory critiche di sistema mantengano i permessi restrittivi nativi (es. *drwxr-xr-x*), utilizzando l'elevazione dei privilegi tramite sudo esclusivamente quando strettamente necessario, per prevenire manomissioni.
- **Adozione Strategica dei Collegamenti Fisici**: Privilegiare l'implementazione di Hard Link rispetto ai Link Simbolici per la referenziazione di dati critici condivisi, sfruttando la loro indipendenza dal nome del file per prevenire la rottura accidentale dei collegamenti operativi.

- **Monitoraggio dei Dispositivi Orfani**: Verificare metodicamente l'infrastruttura di archiviazione per individuare dischi fisici connessi al sistema ma privi di punti di montaggio (come osservato per la partizione *sdb1*), al fine di governare attivamente l'hardware ed evitare fughe di dati.