Report Laboratorio - Esercizio 3

Navigare nel Filesystem Linux e Impostazioni dei Permessi

Studente: Vincenzo Di Salvo
Data: 23 Febbraio 2026
Corso: Cyber Security Ethical Hacking - Build Week 3

Introduzione

Questo report documenta l'esecuzione completa dell'Esercizio 3, focalizzato sull'esplorazione del filesystem Linux e sulla gestione dei permessi dei file. L'obiettivo principale è acquisire familiarità con i comandi fondamentali di Linux per la navigazione del filesystem, la gestione dei dispositivi a blocchi, e la configurazione dei permessi di file e directory.

Obiettivi del Laboratorio

L'esercizio è suddiviso in tre parti principali:

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

Parte 1: Esplorare i Filesystem in Linux

Passo 1 - Accesso alla Riga di Comando

Per iniziare il laboratorio, ho effettuato l'accesso alla VM CyberOps Workstation con le seguenti credenziali:

- **Username:** analyst
- **Password:** cyberops

Ho quindi aperto l'emulatore di terminale facendo clic sull'icona del terminale situata nel Dock.

[Screenshot 1: Desktop VM CyberOps Workstation con terminale aperto]

Passo 2 - Visualizzazione dei Filesystem Montati

2.1 - Visualizzazione dei Dispositivi a Blocchi

Ho utilizzato il comando `lsblk` per visualizzare tutti i dispositivi a blocchi disponibili sul sistema:

[analyst@secOps ~]$ lsblk

[Screenshot 2: Output del comando lsblk]

Dispositivo	Dimensione	Tipo
sda	10G	disk
└─sda1	10G	part (/)
sdb	1G	disk
└─sdb1	1023M	part
sr0	1024M	rom

Table 1: Dispositivi a blocchi identificati dal sistema

Osservazioni:

- Il sistema ha tre dispositivi a blocchi installati: `sr0`, `sda`, e `sdb`
- `sda` contiene il filesystem radice (root) montato su `/`
- `sdb` contiene una partizione da 1GB non montata
- La convenzione `/dev/sdX` viene utilizzata per rappresentare gli hard disk in Linux

2.2 - Informazioni Dettagliate sui Filesystem

Ho eseguito il comando `mount` per ottenere informazioni dettagliate sui filesystem montati:

[analyst@secOps ~]$ mount

[Screenshot 3: Output completo del comando mount]

Per filtrare solo il filesystem radice, ho utilizzato `grep`:

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

Analisi:

- Il filesystem radice si trova su `/dev/sda1`
- Il punto di montaggio è `/` (root)
- Il tipo di filesystem è `ext4`
- Le opzioni di montaggio includono `rw` (read-write) e `relatime`

2.3 - Contenuto della Directory Radice

Ho navigato nella directory radice ed elencato il suo contenuto:

```
[analyst@secOps ~]$ cd /
[analyst@secOps /]$ ls -l
```

[Screenshot 4: Output di ls -l nella directory radice]

Domanda 1: *Qual è il significato dell'output? Dove sono fisicamente memorizzati i file elencati?*

Risposta: I file elencati sono le directory principali del filesystem Linux (bin, boot, dev, etc, home, lib, mnt, opt, proc, root, run, sbin, sys, tmp, usr, var). Questi file sono fisicamente memorizzati sulla partizione `/dev/sda1`, il primo dispositivo a blocchi del sistema.

Domanda 2: *Perché /dev/sdb1 non viene mostrato nell'output?*

Risposta: `/dev/sdb1` non appare nell'output perché non è attualmente montato sul sistema. Un dispositivo deve essere montato su una directory (punto di montaggio) per essere accessibile nel filesystem.

Passo 3 - Montaggio e Smontaggio Manuale dei Filesystem

3.1 - Verifica del Punto di Montaggio

Ho verificato l'esistenza della directory `second_drive` nella home dell'utente analyst:

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

[Screenshot 5: Directory second_drive vuota]

La directory risulta vuota, come previsto, poiché nessun dispositivo è ancora montato su di essa.

3.2 - Montaggio del Dispositivo

Ho montato la partizione `/dev/sdb1` sulla directory `second_drive`:

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

[Screenshot 6: Comando mount eseguito con successo]

Nessun output indica che il montaggio è avvenuto con successo.

3.3 - Verifica del Contenuto Post-Montaggio

Ho ricontrollato il contenuto della directory:

```
[analyst@secOps ~]$ ls -l second_drive/
total 20
```

drwx------ 2 root root 16384 Mar 3 10:59 lost+found
-rw-r--r-- 1 root root 183 Mar 3 15:42 myFile.txt

[Screenshot 7: Contenuto di second_drive dopo il montaggio]

Domanda 3: *Perché la directory non è più vuota? Dove sono fisicamente memorizzati i file elencati?*

Risposta: La directory non è più vuota perché ora punta al contenuto della partizione `/dev/sdb1` che è stata montata su di essa. I file sono fisicamente memorizzati sulla partizione `/dev/sdb1`, ma sono accessibili attraverso il punto di montaggio `~/second_drive/`.

3.4 - Verifica dello Stato di Montaggio

Ho verificato lo stato di montaggio con il comando `mount`:

[analyst@secOps ~]$ mount | grep /dev/sd
/dev/sda1 on / type ext4 (rw,relatime)
/dev/sdb1 on /home/analyst/second_drive type ext4 (rw,relatime)

[Screenshot 8: Verifica del filesystem montato]

3.5 - Smontaggio del Filesystem

Prima di smontare, mi sono assicurato di non trovarmi nella directory del punto di montaggio:

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

[Screenshot 9: Directory second_drive vuota dopo lo smontaggio]

Dopo lo smontaggio, la directory `second_drive` risulta nuovamente vuota, confermando che il filesystem è stato correttamente smontato.

Parte 2: Permessi dei File

Passo 1 - Visualizzazione e Modifica dei Permessi

1.1 - Navigazione nella Directory Scripts

Ho navigato nella directory degli script:

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

[Screenshot 10: Elenco dei file nella directory scripts]

1.2 - Analisi dei Permessi del File cyops.mn

Ho esaminato i permessi del file `cyops.mn`:

-rw-r--r-- 1 analyst analyst 2871 Apr 28 11:27 cyops.mn

Domanda 4: *Chi è il proprietario del file? E il gruppo?*

Risposta: Il proprietario del file è `analyst` e il gruppo è `analyst`.

Domanda 5: *I permessi per cyops.mn sono -rw-r--r--. Cosa significa?*

Risposta: La stringa dei permessi si scompone come segue:

Proprietario	Gruppo	Altri
rw- (6)	r-- (4)	r-- (4)
lettura + scrittura	solo lettura	solo lettura

Table 2: Decomposizione dei permessi del file cyops.mn

1.3 - Test dei Permessi sulla Directory /mnt

Ho tentato di creare un file nella directory `/mnt`:

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

[Screenshot 11: Errore di permesso negato]

Domanda 6: *Perché il file non è stato creato?*

Per rispondere, ho verificato i permessi della directory `/mnt`:

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

Risposta: Il file non è stato creato perché la directory `/mnt` appartiene all'utente `root` e il gruppo `root`. I permessi per "altri" utenti (tra cui `analyst`) sono `r-x` (lettura ed esecuzione), che non includono il permesso di scrittura (w). Per poter creare file in questa directory, sarebbe necessario:

1. Usare `sudo` per ottenere privilegi elevati
2. Modificare i permessi della directory con `chmod`
3. Cambiare la proprietà della directory con `chown`

1.4 - Modifica dei Permessi con chmod

Ho rimontato `/dev/sdb1` e navigato nella directory:

[analyst@secOps ~]$ sudo mount /dev/sdb1 ~/second_drive/
[analyst@secOps ~]$ cd ~/second_drive
[analyst@secOps second_drive]$ ls -l

[Screenshot 12: Contenuto iniziale di second_drive]

Permessi iniziali di `myFile.txt`: `-rw-r--r--` (644)

Ho modificato i permessi del file:

[analyst@secOps second_drive]$ sudo chmod 665 myFile.txt
[analyst@secOps second_drive]$ ls -l

[Screenshot 13: Permessi modificati del file]

Permessi dopo la modifica: `-rw-rw-r-x` (665)

Domanda 7: *I permessi sono cambiati? Quali sono i permessi di myFile.txt?*

Risposta: Sì, i permessi sono cambiati. I nuovi permessi `665` in ottale corrispondono a:

- Proprietario: `rw-` (6 = 110 binario = lettura + scrittura)
- Gruppo: `rw-` (6 = 110 binario = lettura + scrittura)
- Altri: `r-x` (5 = 101 binario = lettura + esecuzione)

Domanda 8: *Quale comando cambierebbe i permessi di myFile.txt a rwxrwxrwx?*

Risposta: Il comando sarebbe `chmod 777 myFile.txt`, dove 7 in ottale = 111 in binario = rwx (lettura + scrittura + esecuzione) per tutte e tre le categorie.

1.5 - Modifica della Proprietà con chown

Ho cambiato il proprietario del file da `root` ad `analyst`:

[analyst@secOps second_drive]$ sudo chown analyst myFile.txt
[analyst@secOps second_drive]$ ls -l

[Screenshot 14: Proprietà modificata del file]

Ho quindi testato la scrittura sul file:

[analyst@secOps second_drive]$ echo "test" >> myFile.txt
[analyst@secOps second_drive]$ cat myFile.txt

[Screenshot 15: Contenuto del file dopo l'aggiunta di testo]

Domanda 9: *L'operazione è riuscita? Spiega.*

Risposta: Sì, l'operazione è riuscita. Ora che `analyst` è il proprietario del file e i permessi del proprietario includono la scrittura (`rw-`), l'utente `analyst` può modificare il contenuto del file senza usare `sudo`.

Passo 2 - Directory e Permessi

2.1 - Confronto tra File e Directory

Ho navigato in `~/lab.support.files` ed esaminato i permessi:

[analyst@secOps ~]$ cd ~/lab.support.files/
[analyst@secOps lab.support.files]$ ls -l

[Screenshot 16: Elenco completo dei file con permessi]

Confronto tra `malware` (directory) e `mininet_services` (file):

- `malware`: `drwxr-xr-x`
- `mininet_services`: `-rwxr-xr-x`

Differenze principali:

- Il carattere 'd' all'inizio indica una directory
- Il bit di esecuzione (x) su una directory permette l'accesso (cd) alla directory
- Il bit di esecuzione (x) su un file permette l'esecuzione del file come programma

---

Parte 3: Link Simbolici e Altri Tipi di File Speciali

Passo 1 - Esame dei Tipi di File

1.1 - Tipi di File in Linux

Linux riconosce diversi tipi di file:

Simbolo	Tipo	Descrizione
-	File regolare	File di testo, binari, immagini
d	Directory	Cartelle
b	File a blocco	Dispositivi di storage
c	File a carattere	Terminali, dispositivi seriali
l	Link simbolico	Collegamenti a file/directory
p	Pipe	FIFO (First In First Out)
s	Socket	Comunicazione di rete

Table 3: Tipi di file in Linux

1.2 - Visualizzazione dei File nella Home

[analyst@secOps ~]$ ls -l

[Screenshot 17: Elenco dei file nella home directory]

1.3 - Esplorazione della Directory /dev

[analyst@secOps ~]$ ls -l /dev/

[Screenshot 18: Estratto della directory /dev con diversi tipi di file]

Esempi identificati:

- File a blocco: `brw-rw---- sda`, `brw-rw---- sdb`
- File a carattere: `crw-rw-rw- random`, `crw-rw-r-- rtc0`
- Link simbolici: `lrwxrwxrwx rtc -> rtc0`

1.4 - Creazione di Link Simbolici e Hard Link

Ho creato due file di test:

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

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

[Screenshot 19: Creazione dei file di test]

Ho quindi creato un link simbolico e un hard link:

[analyst@secOps ~]$ ln -s file1.txt file1_symbolic
[analyst@secOps ~]$ ln file2.txt file2_hard
[analyst@secOps ~]$ ls -l

[Screenshot 20: File con link simbolici e hard link]

Output:
lrwxrwxrwx 1 analyst analyst 9 file1_symbolic -> file1.txt
-rw-r--r-- 1 analyst analyst 9 file1.txt
-rw-r--r-- 2 analyst analyst 5 file2_hard
-rw-r--r-- 2 analyst analyst 5 file2.txt

Osservazioni chiave:

- `file1_symbolic` ha 'l' come primo carattere e mostra una freccia verso `file1.txt`
- `file2_hard` ha '2' nella quinta colonna, indicando 2 link all'inode
- `file2.txt` ha anch'esso '2' nella quinta colonna

1.5 - Test della Resilienza dei Link

Ho rinominato i file originali:

```
[analyst@secOps ~]$ mv file1.txt file1_new.txt
[analyst@secOps ~]$ mv file2.txt file2_new.txt
[analyst@secOps ~]$ cat file1_symbolic
cat: file1_symbolic: no such file or directory
[analyst@secOps ~]$ cat file2_hard
hard
```

[Screenshot 21: Test dei link dopo la ridenominazione]

Domanda 10: *Cosa pensi succederebbe a file2_hard se aprissi un editor di testo e cambiassi il testo in file2_new.txt?*

Risposta: Se modificassi il contenuto di `file2_new.txt`, anche `file2_hard` rifletterebbe immediatamente le stesse modifiche. Questo perché entrambi i file puntano allo stesso inode sull'hard disk. Un hard link non è semplicemente un riferimento al nome del file, ma un vero e proprio puntatore ai dati fisici. Quindi, qualsiasi modifica ai dati si rifletterebbe in tutti gli hard link che puntano a quell'inode.

Conclusioni e Riflessioni

Competenze Acquisite

Attraverso questo laboratorio ho acquisito competenze pratiche su:

1. **Gestione dei Filesystem**
    - Identificazione dei dispositivi a blocchi con `lsblk`
    - Montaggio e smontaggio manuale dei filesystem
    - Comprensione della struttura gerarchica di Linux
2. **Sistema dei Permessi**
    - Lettura e interpretazione delle stringhe di permessi
    - Modifica dei permessi con `chmod` (notazione ottale)
    - Cambio di proprietà con `chown`
    - Comprensione del significato del bit di esecuzione per directory
3. **Link e File Speciali**
    - Differenza tra link simbolici e hard link
    - Comportamento dei link dopo la ridenominazione dei file originali
    - Identificazione dei diversi tipi di file in Linux

Importanza per la Cybersecurity

La comprensione approfondita del sistema dei permessi di Linux è fondamentale per un analista di cybersecurity perché:

1. **Privilege Escalation:** Molti exploit mirano a sfruttare configurazioni errate dei permessi per ottenere accesso root
2. **Hardening del Sistema:** Una corretta configurazione dei permessi è essenziale per la sicurezza del sistema
3. **Analisi Forense:** Durante le investigazioni, è cruciale comprendere chi ha accesso a quali file e quando
4. **Incident Response:** Permessi errati possono essere indicatori di compromissione (IoC)

Problematiche Comuni

Come evidenziato nella riflessione finale del laboratorio, i permessi e la proprietà dei file sono spesso causa di problemi operativi. Un file con permessi o proprietà errati può:

- Bloccare l'esecuzione di servizi critici
- Impedire l'accesso a dati necessari
- Creare vulnerabilità di sicurezza
- Causare malfunzionamenti di applicazioni

Comandi Principali Utilizzati

Comando	Funzione
`lsblk`	Visualizza dispositivi a blocchi
`mount`	Monta filesystem / mostra filesystem montati
`umount`	Smonta filesystem
`ls -l`	Elenca file con dettagli e permessi
`ls -ld`	Mostra dettagli di una directory
`chmod`	Modifica permessi di file/directory
`chown`	Cambia proprietario di file/directory
`ln -s`	Crea link simbolico
`ln`	Crea hard link
`touch`	Crea file vuoto

Table 4: Comandi Linux utilizzati nel laboratorio

Applicazioni Pratiche

Le competenze acquisite in questo laboratorio sono direttamente applicabili a scenari reali di cybersecurity:

- **Configurazione di server sicuri:** Impostare permessi corretti per file di configurazione sensibili
- **Analisi di malware:** Comprendere come il malware può sfruttare permessi errati
- **Hardening di sistema:** Applicare il principio del least privilege
- **Forensics:** Analizzare chi ha modificato file critici e quando

---

Appendice: Screenshot Richiesti

Lista Completa degli Screenshot

1. Desktop VM CyberOps Workstation con terminale aperto
2. Output del comando lsblk
3. Output completo del comando mount
4. Output di ls -l nella directory radice
5. Directory second_drive vuota
6. Comando mount eseguito con successo
7. Contenuto di second_drive dopo il montaggio
8. Verifica del filesystem montato
9. Directory second_drive vuota dopo lo smontaggio
10. Elenco dei file nella directory scripts
11. Errore di permesso negato
12. Contenuto iniziale di second_drive
13. Permessi modificati del file
14. Proprietà modificata del file
15. Contenuto del file dopo l'aggiunta di testo
16. Elenco completo dei file con permessi
17. Elenco dei file nella home directory
18. Estratto della directory /dev con diversi tipi di file
19. Creazione dei file di test
20. File con link simbolici e hard link
21. Test dei link dopo la ridenominazione

Nota: Gli screenshot effettivi dovranno essere inseriti durante l'esecuzione pratica del laboratorio.

Riferimenti

- Linux File System Hierarchy: https://refspecs.linuxfoundation.org/FHS_3.0/fhs/index.html
- GNU Coreutils Documentation: https://www.gnu.org/software/coreutils/manual/
- Linux Man Pages: https://linux.die.net/man/