

Report: Gestione dei Permessi di Lettura, Scrittura ed Esecuzione in Linux

Titolo: Gestione dei Permessi in Linux: Esercizio Pratico e Analisi dei Risultati

Introduzione: Questo esercizio ha avuto come obiettivo la configurazione e la gestione dei permessi di lettura, scrittura ed esecuzione per una directory e un file in un ambiente Linux. La corretta gestione dei permessi è fondamentale per la sicurezza e la protezione dei dati in un sistema operativo basato su Linux, consentendo di controllare chi può accedere, modificare o eseguire file e directory.

Scelta del File/Directory: Per questo esercizio, è stata creata una directory denominata `Cartella_Esercizi` e al suo interno un file chiamato `doc.txt`. La scelta di queste entità è stata fatta per simulare uno scenario comune in cui si desidera proteggere un'area di lavoro (la directory) e un documento specifico al suo interno.

Screenshot della Creazione del File o della Directory:

```
(kali㉿kali)-[~]  
$ mkdir Cartella_Esercizi  
  
(kali㉿kali)-[~]  
$ cd Cartella_Esercizi  
  
(kali㉿kali)-[~/Cartella_Esercizi]  
$ touch doc.txt
```

Questi comandi hanno permesso di creare la struttura di base per l'esercizio.

Screenshot della Verifica dei Permessi Attuali (Iniziale): *(Fare riferimento a "permessi.png" e "permessi cartella.png" - prima del chmod)*

```
(kali㉿kali)-[~/Cartella_Esercizi]  
$ chmod 740 .  
  
(kali㉿kali)-[~/Cartella_Esercizi]  
$ ls -l .  
total 0  
-rw-rw-r-- 1 kali kali 0 Jun  3 08:05 doc.txt  
  
(kali㉿kali)-[~/Cartella_Esercizi]  
$ ls -ld .  
drwxr-xr-x 2 kali kali 4096 Jun  3 08:05 .
```

```
(kali㉿kali)-[~/Cartella_Esercizi]  
$ chmod 640 doc.txt  
  
(kali㉿kali)-[~/Cartella_Esercizi]  
$ ls -l doc.txt  
-rw-r----- 1 kali kali 0 Jun  3 08:05 doc.txt
```

Prima delle modifiche, i permessi di default erano:

- **doc.txt:** `-rw-rw-r--` (664) - Proprietario e gruppo avevano permessi di lettura e scrittura; altri avevano solo lettura.
- **Cartella_Esercizi:** `drwxrwxr-x` (775) - Proprietario e gruppo avevano tutti i permessi (lettura, scrittura, esecuzione); altri avevano permessi di lettura ed esecuzione.

Motivazione delle Scelte dei Permessi Configurati e Screenshot della Modifica dei Permessi:

1. Permessi per la Directory Cartella_Esercizi:

```
(kali㉿kali)-[~/Cartella_Esercizi]
$ chmod 740 .

(kali㉿kali)-[~/Cartella_Esercizi]
$ ls -l .
total 0
-rw-rw-r-- 1 kali kali 0 Jun  3 08:05 doc.txt

(kali㉿kali)-[~/Cartella_Esercizi]
$ ls -ld .
drwxr----- 2 kali kali 4096 Jun  3 08:05 .
```

Proprietario (U-User: kali) - rwx (7): È stato concesso il permesso completo (lettura, scrittura, esecuzione). Questo è essenziale per consentire al proprietario di:

- r (leggere): visualizzare il contenuto della directory (es. listare i file).
 - w (scrivere): creare, eliminare e rinominare file all'interno della directory.
 - x (eseguire): accedere e navigare all'interno della directory.
- **Gruppo (G-Group: kali) - r-- (4):** È stato concesso solo il permesso di lettura. La motivazione è permettere ai membri dello stesso gruppo (se ce ne fossero altri) di visualizzare i nomi dei file nella directory, ma non di modificarli, crearne di nuovi o accedervi. L'assenza del permesso di esecuzione (x) impedisce comunque l'ingresso nella directory, rendendo la lettura dei nomi dei file inaccessibile tramite ls se non si ha accesso alla directory stessa.
 - **Altri (O-Others) - --- (0):** Non è stato concesso alcun permesso. La motivazione è garantire la massima riservatezza e sicurezza, impedendo a qualsiasi altro utente sul sistema (che non sia il proprietario o parte del gruppo) di accedere, visualizzare o manipolare il contenuto della directory.

2. Permessi per il File doc.txt:

- **Proprietario (U-User: kali) - rw- (6):** Sono stati concessi i permessi di lettura e scrittura. Questo permette al proprietario di:

- r (leggere): visualizzare il contenuto del file.

- w (scrivere): modificare o eliminare il contenuto del file.

```
(kali@kali)-[~/Cartella_Esercizi]
$ chmod 640 doc.txt

(kali@kali)-[~/Cartella_Esercizi]
$ ls -l doc.txt
-rw-r----- 1 kali kali 0 Jun  3 08:05 doc.txt
```

- Non è stato concesso il permesso di esecuzione (-), poiché doc.txt è un file di testo e non uno script o un programma eseguibile.

- **Gruppo (G-Group: kali) - r-- (4):** È stato concesso solo il permesso di lettura. La motivazione è consentire ai membri dello stesso gruppo di leggere il contenuto del documento, ma di non poterlo modificare o eliminare.
- **Altri (O-Others) - --- (0):** Non è stato concesso alcun permesso. Questo assicura che il contenuto del documento sia completamente privato e non accessibile a utenti esterni.

Screenshot della Creazione Nuovo Utente per Prova:

È stato creato un nuovo utente Stefano per testare i permessi come "altri" (o come utente non proprietario/non nel gruppo proprietario) rispetto ai file e directory creati da kali.

```
(kali@kali)-[~/Cartella_Esercizi]
$ sudo adduser Stefano
[sudo] password for kali:
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for Stefano
Enter the new value, or press ENTER for the default
  Full Name []:
   Room Number []:
   Work Phone []:
   Home Phone []:
    Other []:
Is the information correct? [Y/n] y

(kali@kali)-[~/Cartella_Esercizi]
$ su Stefano
Password:
```

Analisi dei Risultati Ottenuti Durante i Test dei Permessi:

Test come Proprietario (kali):

- **Creazione di doc2.txt nella directory:**
L'operazione touch doc2.txt è riuscita. Questo conferma che il proprietario (kali) ha il permesso di scrittura (w) sulla directory Cartella_Esercizi, consentendogli di creare nuovi file al suo interno.

```
(kali@kali)-[~/Cartella_Esercizi]
$ touch doc2.txt

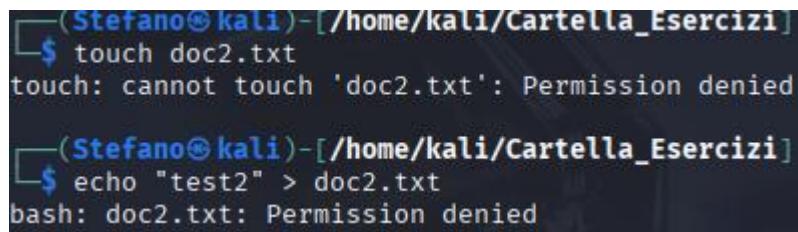
(kali@kali)-[~/Cartella_Esercizi]
$ echo "Test1" > doc2.txt

(kali@kali)-[~/Cartella_Esercizi]
$ cat doc2.txt
Test1
```

- **Scrittura nel file doc.txt:** Il comando `echo "Test1" > doc.txt` è stato eseguito con successo e il contenuto è stato scritto nel file.
- **Lettura del file doc.txt:** Il comando `cat doc.txt` ha mostrato correttamente "Test1". Questi test dimostrano che il proprietario ha pieno controllo (lettura e scrittura) sul file `doc.txt` e sulla directory `Cartella_Esercizi`, come previsto dai permessi 740 per la directory e 640 per il file.

Test come Altro Utente (Stefano):

- **Tentativo di creazione di doc2.txt nella directory:** Il comando `touch doc2.txt` ha generato un errore: `Permission denied`. Questo è il risultato atteso. L'utente Stefano non è il proprietario di `Cartella_Esercizi` e non appartiene al gruppo `kali` con permessi di scrittura, né rientra nella categoria "altri" che ha zero permessi. Di conseguenza, non ha il permesso di scrittura (w) sulla directory, impedendogli di creare nuovi file.
- **Tentativo di scrittura in doc2.txt (anche se non esiste):** Il comando `echo "test2" > doc2.txt` ha anche generato `Permission denied`. Anche se `doc2.txt` non è stato creato, il tentativo di reindirizzare l'output in un file in una directory senza permessi di scrittura causa questo errore. Similmente, se avesse tentato di scrivere in `doc.txt`, avrebbe ottenuto lo stesso errore in quanto Stefano non ha permessi di scrittura su quel file.



```
(Stefano@kali)-[/home/kali/Cartella_Esercizi]
$ touch doc2.txt
touch: cannot touch 'doc2.txt': Permission denied

(Stefano@kali)-[/home/kali/Cartella_Esercizi]
$ echo "test2" > doc2.txt
bash: doc2.txt: Permission denied
```

Questi test confermano che i permessi impostati per la directory e il file funzionano come previsto, impedendo accessi non autorizzati e scritture da parte di utenti non autorizzati.

Conclusioni: Questo esercizio ha illustrato efficacemente l'importanza e la funzionalità dei permessi di lettura, scrittura ed esecuzione in Linux. Abbiamo dimostrato come, attraverso l'uso del comando `chmod` con notazione ottale, sia possibile definire granularmente i diritti di accesso per proprietario, gruppo e altri utenti. I test condotti hanno chiaramente evidenziato che le modifiche ai permessi sono state applicate correttamente, garantendo che solo gli utenti autorizzati possano interagire con i file e le directory secondo le regole stabilite, proteggendo così l'integrità e la confidenzialità dei dati. La gestione accurata dei permessi è una colonna portante della sicurezza in ambiente Linux.