

Modifica dei Permessi su Kali

Esercizio di oggi:

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

1. Screenshot della Creazione del File o della Directory:
 - Fornisci uno screenshot che mostri i comandi utilizzati per creare il file o la directory.
2. Screenshot della Verifica dei Permessi Attuali:
 - Fornisci uno screenshot che mostri i comandi `ls -l` e l'output prima della modifica dei permessi.
3. Screenshot della Modifica dei Permessi:
 - Fornisci uno screenshot che mostri i comandi `chmod` utilizzati e l'output successivo con `ls -l`.
4. Screenshot del Test dei Permessi:
 - Fornisci uno screenshot che mostri i tentativi di scrivere nel file o di creare un nuovo file nella directory, insieme ai comandi e agli output.
5. Relazione:
 - Scrivi una relazione spiegando le scelte fatte riguardo ai permessi configurati.

La relazione deve includere:

 - La motivazione delle scelte fatte per i permessi di lettura, scrittura ed esecuzione.
 - Un'analisi dei risultati ottenuti durante i test dei permessi.

Creazione del File o della Directory

Nel primo passaggio ho creato una directory chiamata 'esercizio_permessi' con il comando:

bash:

```
mkdir esercizio_permessi
```

Questo comando serve per creare una nuova directory vuota nel percorso corrente.

La directory appena creata eredita i permessi di default impostati per l'utente.

```
(kali㉿kali)-[~]
└─$ mkdir esercizio_permessi

(kali㉿kali)-[~]
└─$ ls -l
total 84
drwxr-xr-x 3 kali kali 4096 May 30 03:19 Desktop
drwxr-xr-x 2 kali kali 4096 May 7 04:18 Documents
drwxr-xr-x 3 kali kali 4096 May 7 05:18 Downloads
drwxr-xr-x 2 kali kali 4096 Jun 3 08:29 esercizio_permessi
-rw-rw-r-- 1 kali kali 244 May 22 09:10 hashes.txt
drwxr-xr-x 2 kali kali 4096 May 7 04:18 Music
-rw-rw-r-- 1 kali kali 641 May 7 04:37 packages.microsoft.gpg
-rw-rw-r-- 1 kali kali 35 May 22 10:21 pass.txt
drwxr-xr-x 2 kali kali 4096 Jun 3 08:19 Pictures
-rw-rw-r-- 1 kali kali 30900 May 26 09:12 polimorfi.exe
drwxr-xr-x 2 kali kali 4096 May 7 04:18 Public
drwxr-xr-x 2 kali kali 4096 May 7 04:18 Templates
-rw-rw-r-- 1 kali kali 39 May 22 05:32 user.txt
drwxr-xr-x 2 kali kali 4096 May 7 04:18 Videos
-rw-rw-r-- 1 kali kali 0 May 7 04:26 vscode.deb
```

Verifica dei Permessi Attuali

A questo punto verifichiamo i permessi della cartella appena creata con il comando `ls -l` così che ci vengono elencati i vari file e le directory con i relativi permessi.

L'output mostrava che la directory esercizio_permessi inizialmente aveva i permessi:

```
drwxrwxr-x 2 kali kali 4096 Jun 3 08:29 esercizio_permessi
```

drwxr-xr-x significa che:

- Il **proprietario** ha permessi di lettura, scrittura ed esecuzione (rwx(read)(write)(execution))
- Il **gruppo** e **altri utenti** hanno solo lettura ed esecuzione (r-x)

Modifica dei Permessi

Successivamente, ho modificato i permessi della directory usando:

chmod 600 esercizio_permessi

```
(kali@kali)-[~]  
$ chmod 600 esercizio_permessi
```

Questo comando rimuove **tutti i permessi di gruppo e altri utenti**, lasciando al proprietario solo lettura e scrittura (rw-----). L'obiettivo era **limitare completamente l'accesso agli altri utenti**, anche alla semplice visualizzazione del contenuto della directory.

Dopo la modifica, l'output di ls -l mostrava:

drw----- esercizio_permessi

```
(kali@kali)-[~]  
$ ls -l  
total 84  
drwxr-xr-x 3 kali kali 4096 May 30 03:19 Desktop  
drwxr-xr-x 2 kali kali 4096 May 7 04:18 Documents  
drwxr-xr-x 3 kali kali 4096 May 7 05:18 Downloads  
drw----- 2 kali kali 4096 Jun 3 08:29 esercizio_permessi  
-rw-rw-r-- 1 kali kali 244 May 22 09:10 hashes.txt  
drwxr-xr-x 2 kali kali 4096 May 7 04:18 Music  
-rw-rw-r-- 1 kali kali 641 May 7 04:37 packages.microsoft.gpg  
-rw-rw-r-- 1 kali kali 35 May 22 10:21 pass.txt  
drwxr-xr-x 2 kali kali 4096 Jun 3 08:29 Pictures  
-rw-rw-r-- 1 kali kali 30900 May 26 09:12 polimorfi.exe  
drwxr-xr-x 2 kali kali 4096 May 7 04:18 Public  
drwxr-xr-x 2 kali kali 4096 May 7 04:18 Templates  
-rw-rw-r-- 1 kali kali 39 May 22 05:32 user.txt  
drwxr-xr-x 2 kali kali 4096 May 7 04:18 Videos  
-rw-rw-r-- 1 kali kali 0 May 7 04:26 vscode.deb
```

Questo è il blocco dei **permessi del file o directory**. È composto da **10 caratteri**:

- **1° carattere**: tipo di file
 - d → significa **directory**
 - Se fosse - sarebbe un file normale.

- **2°-4° carattere**: permessi del **proprietario (user)**
 - r → read (lettura)
 - w → write (scrittura)
 - - → no execute (non eseguibile / non può entrare nella directory)

In questo caso: rw- → il proprietario può leggere e scrivere, **ma non eseguire/entrare** nella directory (inusuale per una directory!).

- **5°-7° carattere**: permessi del **gruppo**
 - --- → il gruppo **non ha nessun permesso**
- **8°-10° carattere**: permessi di **altri (others)**
 - --- → gli altri **non hanno nessun permesso**

In sintesi:

- È una **directory**
- Solo il **proprietario** può leggerla e scriverla
- Ma il proprietario **non può accedervi** perché **manca il permesso di esecuzione (x)**
- Tutti gli altri (gruppo e altri utenti) **non possono fare nulla**

Successivamente, per poter testare nuovamente la scrittura al suo interno, ho dovuto **ripristinare i permessi completi**:

chmod 777 esercizio_permessi

Così ho concesso **tutti i permessi a tutti gli utenti** (lettura, scrittura ed esecuzione).

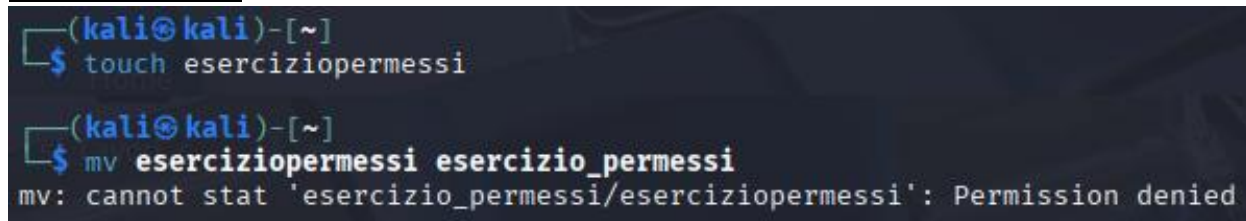
Test dei Permessi

Per testare il comportamento dei permessi, ho provato a creare un file all'interno della directory con:

touch eserciziopermessi – ho poi provato a spostare il file creato all'interno della directory.

Durante il primo tentativo (quando i permessi erano 600), ho ottenuto l'errore:

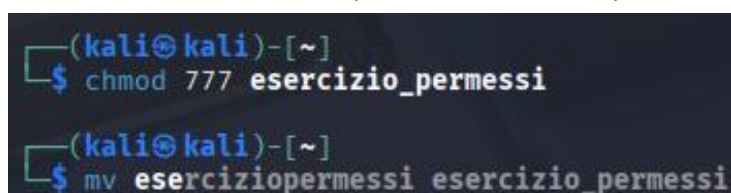
Permission denied.



```
(kali@kali)-[~]  
$ touch eserciziopermessi  
  
(kali@kali)-[~]  
$ mv eserciziopermessi esercizio_permessi  
mv: cannot stat 'esercizio_permessi/eserciziopermessi': Permission denied
```

Questo è esattamente il comportamento previsto, dato che la directory non aveva il bit di esecuzione (x), fondamentale per accedere a una directory in Linux.

Non ci resta che dare allora i permessi alla directory con il comando **chmod 777 esercizio_permessi** :



```
(kali@kali)-[~]  
$ chmod 777 esercizio_permessi  
  
(kali@kali)-[~]  
$ mv eserciziopermessi esercizio_permessi
```

Dopo aver ripristinato i permessi con chmod 777, il comando touch ha funzionato correttamente, e il file è stato creato.

Verifichiamo infine, l'output ottenuto con quest'ultimo comando inserito:

Analisi dell'output (immagine finale)

```
(kali@kali)-[~]
$ ls -l
total 84
drwxr-xr-x 3 kali kali 4096 May 30 03:19 Desktop
drwxr-xr-x 2 kali kali 4096 May 7 04:18 Documents
drwxr-xr-x 3 kali kali 4096 May 7 05:18 Downloads
drwxrwxrwx 2 kali kali 4096 Jun 3 08:34 esercizio_permessi
-rw-rw-r-- 1 kali kali 244 May 22 09:10 hashes.txt
drwxr-xr-x 2 kali kali 4096 May 7 04:18 Music
-rw-rw-r-- 1 kali kali 641 May 7 04:37 packages.microsoft.gpg
-rw-rw-r-- 1 kali kali 35 May 22 10:21 pass.txt
drwxr-xr-x 2 kali kali 4096 Jun 3 08:33 Pictures
-rw-rw-r-- 1 kali kali 30900 May 26 09:12 polimorfi.exe
drwxr-xr-x 2 kali kali 4096 May 7 04:18 Public
drwxr-xr-x 2 kali kali 4096 May 7 04:18 Templates
-rw-rw-r-- 1 kali kali 39 May 22 05:32 user.txt
drwxr-xr-x 2 kali kali 4096 May 7 04:18 Videos
-rw-rw-r-- 1 kali kali 0 May 7 04:26 vscode.deb
```

Nell'immagine vediamo questa riga in particolare:

drwxrwxrwx 2 kali kali 4096 Jun 3 08:34
esercizio_permessi

Questa è la stringa dei permessi e significa:

- **d** = è una **directory**
- **rw**x = il **proprietario** (kali) ha tutti i permessi: lettura (r), scrittura (w), esecuzione (x)
- **rw**x = il **gruppo** (kali) ha tutti i permessi
- **rw**x = **tutti gli altri utenti** (others) hanno tutti i permessi

Questa configurazione (777) è molto permissiva: chiunque può entrare nella directory, leggere/modificare/cancellare i file.

Riepilogo sul chmod 777

Il comando chmod 777 esercizio_permessi ha cambiato i permessi in modo da rendere la directory:

- Accessibile da **chiunque**
- Modificabile da **chiunque**
- Visualizzabile da **chiunque**

Analisi finale e Motivazione delle Scelte

Permessi iniziali (600)

- Obiettivo: **Bloccare completamente l'accesso agli altri utenti**
- Permessi assegnati: rw-----
- Motivazione: questo scenario simula una directory privata in cui solo il proprietario può leggere o modificare i contenuti, ma **nemmeno accedervi** se manca il permesso di esecuzione.

Test con touch fallito

- Comportamento atteso: impossibile accedere alla directory per scrivere file
- Dimostra come il **bit di esecuzione** su una directory sia essenziale

Permessi finali (777)

- Obiettivo: **Consentire la scrittura per completare il test**
 - Permessi assegnati: rwxrwxrwx
 - Motivazione: ho assegnato i permessi massimi temporaneamente per verificare il successo dell'operazione touch.
-

Conclusione

Questo esercizio ha fornito un'opportunità concreta per comprendere a fondo il funzionamento e l'impatto dei permessi nel sistema operativo Linux.

Attraverso la creazione e la modifica di directory e file, si è potuto osservare come ogni combinazione di permessi (r, w, x) influenzi direttamente la possibilità di accedere, modificare o semplicemente visualizzare contenuti all'interno del file system.

Un punto chiave emerso è l'importanza del permesso di **esecuzione (x) nelle directory**: senza di esso, anche se si ha la lettura, **non è possibile entrare nella directory o interagire con i file al suo interno**, come dimostrato dal messaggio "Permission denied" durante il tentativo di scrittura.

L'esercizio ha inoltre mostrato come i permessi siano un **mezzo fondamentale per la sicurezza**: un semplice `chmod 600` può impedire a chiunque (inclusi altri utenti) di accedere a dati riservati, mentre un `chmod 777`, pur sbloccando ogni restrizione, rappresenta un rischio in ambienti multiutente o di produzione.

Infine, l'esperienza pratica con `chmod` ha evidenziato la **flessibilità del controllo degli accessi in Linux**, permettendo una gestione dinamica e puntuale, adattabile alle esigenze di ogni singolo contesto operativo.

In sintesi, lavorare consapevolmente con i permessi non è solo una questione tecnica, ma anche una **buona pratica di sicurezza** che ogni utente Linux dovrebbe padroneggiare.