

Pratica S10/L2

Introduzione: l'obiettivo di questo esercizio è stato gestire i permessi in un ambiente Linux, creando una directory, modificarne i permessi e verificare che le nuove impostazioni funzionassero correttamente per un utente non proprietario.

1. Creazione della directory

Il primo passo è stato creare un directory dedicata all'esercizio. Ho deciso di chiamarla *esercizio_permessi* per chiarezza.

```
(kali㉿kali)-[~]  
$ mkdir /home/kali/esercizio_permessi
```

Successivamente, ho utilizzato il comando `ls -ld` per verificare i permessi di default assegnati dal sistema alla nuova directory.

```
(kali㉿kali)-[~]  
$ ls -ld /home/kali/esercizio_permessi  
drwxrwxr-x 2 kali kali 4096 Sep 16 09:51 /home/kali/esercizio_permessi
```

L'output che ho ricevuto indica che questa directory ha permessi di lettura, scrittura ed esecuzione per ogni utente.

2. Modifica dei permessi

Per l'esercizio, ho deciso di modificare i permessi della directory in modo che fosse completamente controllabile solo dal proprietario, mentre gli altri utenti potevano semplicemente leggerne il contenuto.

```
(kali㉿kali)-[~]  
$ chmod 744 /home/kali/esercizio_permessi
```

Questo ha impostato i permessi su 744 che corrispondono a:

- 7 (rwx): Controllo totale per il proprietario.
- 4 (r--): Sola lettura per il gruppo.
- 4 (r--): Sola lettura per gli altri.

```
(kali㉿kali)-[~]  
$ ls -ld esercizio_permessi  
drwxr--r-- 2 kali kali 4096 Sep 16 09:51 esercizio_permessi
```

3. Test dei permessi con un utente non proprietario

Per dimostrare l'efficacia delle modifiche, ho creato un nuovo utente (*utente_test*) con il comando *adduser* e ho tentato di eseguire un'azione che richiedesse il permesso di scrittura. Dopodiché ho eseguito un login con l'utente appena creato e ho provato a creare un file all'interno della directory.

```
(kali㉿kali)-[~/esercizio_permessi]
$ su utente_test
Password:
(kali㉿kali)-[~/esercizio_permessi]
$ touch test.txt
touch: cannot touch 'test.txt': Permission denied
```

Il sistema ha risposto con il messaggio "*Permission denied*". Questo ha confermato che la configurazione dei permessi ha funzionato correttamente, bloccando l'azione di scrittura per l'utente non proprietario.

4. Conclusioni

L'esercizio ha dimostrato con successo come *chmod* possa essere utilizzato per modificare i permessi predefiniti di una directory. La configurazione *chmod 744* si è rivelata efficace nel limitare l'accesso in scrittura solo al proprietario, garantendo l'integrità dei dati nella directory e rafforzando la sicurezza del sistema.