

S10/L2

Gestione dei Permessi in Linux

1. Introduzione

In questa relazione vengono descritti i passaggi eseguiti per configurare e testare i permessi di lettura, scrittura ed esecuzione su un file e una directory in un sistema Linux. L'obiettivo è dimostrare come creare elementi di filesystem, modificarne i permessi e verificarne il corretto funzionamento attraverso test pratici.

2. Creazione del File e della Directory

Sono stati creati un file e una directory con i seguenti comandi:

- **touch esempio.txt**
- **mkdir esempio_dir**

Nello specifico, il comando touch esempio.txt ha generato un nuovo file vuoto chiamato "esempio.txt", mentre il comando mkdir esempio_dir ha creato una directory chiamata "esempio_dir".

```
(kali@kali2023)-[~]  
$ touch esempio.txt  
  
(kali@kali2023)-[~]  
$ mkdir esempio_dir  
  
(kali@kali2023)-[~]  
$
```

3. Verifica dei Permessi Attuali

Per controllare i permessi di default assegnati al file e alla directory, è stato eseguito:

- **ls -l esempio.txt**
- **ls -l esempio_dir**

L'output di questi comandi mostra i permessi iniziali prima di qualsiasi modifica.

```
(kali@kali2023)-[~]  
$ ls -l esempio.txt  
-rw-r--r-- 1 kali kali 18 Jun  3 14:15 esempio.txt
```

```
(kali@kali2023)-[~]  
$ ls -l esempio_dir  
total 0  
-rw-rw-r-- 1 kali kali 0 Jun  3 14:15 nuovo_file.txt
```

4. Modifica dei Permessi

Le modifiche apportate ai permessi sono state le seguenti:

- Per il file “esempio.txt”:
 - Comando: **chmod u=rw,g=r,o=r esempio.txt**
 - Risultato: l’utente (owner) ha permesso di lettura e scrittura (rw-), mentre il gruppo e gli altri possono solo leggere (r--).
- Per la directory “esempio_dir”:
 - Comando: **chmod u=rwx,g=rx,o=rx esempio_dir**
 - Risultato: l’utente ha permessi di lettura, scrittura ed esecuzione (rwx), mentre gruppo e altri possono leggere ed eseguire (r-x).

```
(kali㉿kali2023)-[~]  
$ chmod u=rw,g=r,o=r esempio.txt
```

```
(kali㉿kali2023)-[~]  
$ chmod u=rwx,g=rx,o=rx esempio_dir
```

Dopo l’esecuzione dei comandi, è stata eseguita nuovamente la verifica con `ls -l` per confermare la correzione dei permessi.

```
(kali㉿kali2023)-[~]  
$ ls -l esempio.txt  
-rw-r--r-- 1 kali kali 18 Jun  3 14:15 esempio.txt
```

```
$ ls -l esempio_dir  
total 0  
-rw-rw-r-- 1 kali kali 0 Jun  3 14:15 nuovo_file.txt
```

5. Test dei Permessi

I permessi sono stati testati in modo pratico:

- Scrittura nel file: **echo "Test dei permessi" > esempio.txt**
- Creazione di un file nella directory: **touch esempio_dir/nuovo_file.txt**
- Verifica: **ls -l esempio_dir/nuovo_file.txt**

```
(kali㉿kali2023)-[~]
$ echoecho "Test dei permessi" > esempio.txt

(kali㉿kali2023)-[~]
$ touchmpio_dir/nuovo_file.txt
ls -l esempio_dir/nuovo_file.txt

-rw-rw-r-- 1 kali kali 0 Jun  3 14:15 esempio_dir/nuovo_file.txt
```

```
(kali㉿kali2023)-[~]
$ ls -l esempio_dir/nuovo_file.txt
-rw-rw-r-- 1 kali kali 0 Jun  3 14:15 esempio_dir/nuovo_file.txt
```

```
(kali㉿kali2023)-[~]
$ chmocat esempio.txt
Test dei permessi
```

Questi test hanno permesso di verificare se l'utente poteva effettivamente scrivere nel file e creare un file all'interno della directory, mentre utenti diversi (non illustrati) avrebbero dovuto trovarsi bloccati dalle limitazioni impostate.

6. Motivazione delle Scelte dei Permessi

Per il file “esempio.txt”, è stata scelta la configurazione **u=rw,g=r,o=r** per i seguenti motivi:

- L'utente proprietario deve poter leggere e modificare il contenuto del file.
- Il gruppo e gli altri utenti devono poter solo leggere, evitando modifiche accidentali o non autorizzate.

Questa impostazione è utile in contesti collaborativi dove è necessario condividere informazioni in sola lettura con più utenti, ma mantenere la possibilità di scrittura riservata al proprietario.

Per la directory “esempio_dir”, si è optato per **u=rwx,g=rx,o=rx**:

- Il proprietario può creare, cancellare o rinominare file al suo interno (permesso di scrittura), oltre che leggere ed eseguire.
- Il gruppo e gli altri utenti possono scorrere la directory (r) e accedervi (x), ma non possono creare o eliminare file.

Questa scelta garantisce che i file contenuti nella directory siano visibili e utilizzabili da tutti, ma che solo il proprietario ne possa modificare la struttura.

7. Analisi dei Risultati

Durante i test, è stato verificato che:

- Il comando **echo "Test dei permessi" => esempio.txt** ha correttamente sovrascritto il contenuto del file, dimostrando che l'utente proprietario disponeva dei permessi di scrittura.
- Il comando **touch esempio_dir/nuovo_file.txt** ha creato con successo un nuovo file all'interno della directory, confermando i permessi di scrittura per l'utente.

Tutti i risultati ottenuti corrispondono alle aspettative: i permessi assegnati hanno correttamente limitato o consentito le operazioni desiderate.

8. Conclusioni

L'esercizio ha permesso di comprendere il funzionamento dei permessi di Linux, imparando a modificare le autorizzazioni di file e directory e a verificare il loro effetto tramite test pratici. Le scelte effettuate sono coerenti con buone pratiche di sicurezza, consentendo al proprietario di mantenere il controllo sulle modifiche, pur offrendo visibilità in sola lettura agli altri utenti.