

Gestione di File e Permessi in Linux: Un Esempio Pratico

1. Creazione della Directory e del File

Il primo passo consiste nel creare una directory (o cartella) e, al suo interno, un nuovo file di testo.

- **Creare la directory:** Per prima cosa, creiamo una directory chiamata S10L2. Utilizziamo il comando `mkdir` (make directory).

```
(kali@kali)-[~]  
$ mkdir S10L2
```

- **Entrare nella directory:** Spostiamoci all'interno della directory appena creata con il comando `cd` (change directory).
- **Creare il file:** Ora, all'interno di S10L2, creiamo un file vuoto chiamato `esempio.txt` usando il comando `touch`.

```
(kali@kali)-[~/S10L2]  
$ touch esempio.txt
```

2. Analisi dei Permessi Iniziali con `ls -l`

Per visualizzare i dettagli del nostro file, inclusi i suoi permessi, usiamo il comando `ls -l`.

```
(kali@kali)-[~/S10L2]  
$ ls -l  
total 0  
-rw-rw-r-- 1 kali kali 0 Sep 16 08:29 esempio.txt
```

La prima parte di questa riga, `-rw-rw-r--`, è fondamentale perché descrive chi può fare cosa con il file.

- **Prima tripletta (rw-): Permessi del Proprietario (User)**
 - `r` (read): Il proprietario può leggere il contenuto del file.
 - `w` (write): Il proprietario può scrivere e modificare il file.
 - `-` (nessun permesso): Il proprietario non ha il permesso di esecuzione.
- **Seconda tripletta (rw-): Permessi del Gruppo (Group)**
 - `r` (read): Gli utenti appartenenti allo stesso gruppo del file possono leggerlo.

- w (write): Gli utenti del gruppo possono modificarlo.
- - (nessun permesso)
- **Terza tripletta (r--): Permessi degli Altri (Others)**
 - r (read): Tutti gli altri utenti sul sistema possono solo leggere il file.
 - - (nessun permesso)
 - - (nessun permesso)

In breve, con i permessi di default -rw-rw-r--, sia il proprietario che il gruppo possono leggere e scrivere il file, mentre tutti gli altri possono solo leggerlo.

3. Modifica dei Permessi del File

Utilizzo **chmod** per modificare i permessi del file:

- Aggiunta permessi di esecuzione all'utente corrente

```
(kali㉿kali)-[~/S10L2]
$ chmod u+x esempio.txt

(kali㉿kali)-[~/S10L2]
$ ls -l
total 0
-rwxrw-r-- 1 kali kali 0 Sep 16 08:29 esempio.txt
```

Notiamo il cambiamento nei permessi del file

- Rimozione permessi di scrittura per gli utenti del gruppo

```
(kali㉿kali)-[~/S10L2]
$ chmod g-w esempio.txt

(kali㉿kali)-[~/S10L2]
$ ls -l
total 0
-rwxr--r-- 1 kali kali 0 Sep 16 08:29 esempio.txt
```

- Aggiunta permessi di esecuzione per gli utenti del gruppo

```
(kali㉿kali)-[~/S10L2]
$ chmod g+x esempio.txt
```

```
(kali㉿kali)-[~/S10L2]  
$ ls -l  
total 0  
-rwxr-xr-- 1 kali kali 0 Sep 16 08:29 esempio.txt
```

4. Analisi dei Nuovi Permessi

Analizziamo il nuovo significato di `-rwxr-xr--`:

- **Proprietario (rwx)**: Ora ha i permessi di **lettura, scrittura ed esecuzione**. Può leggere, modificare ed eseguire il file.
- **Gruppo (r-x)**: Ora ha i permessi di **lettura ed esecuzione**, ma non di scrittura. I membri del gruppo possono leggere ed eseguire lo script, ma non possono più modificarlo.
- **Altri (r--)**: I permessi sono rimasti invariati. Tutti gli altri utenti possono solo **leggere** il file.

Con una serie di comandi `chmod`, ho cambiato radicalmente le regole di accesso al file, rendendolo uno script eseguibile dall'utente primario e dal gruppo e, allo stesso tempo, proteggendolo da modifiche indesiderate da parte degli altri utenti.