

Relazione sui Permessi Configurati per il File `permessi.txt`

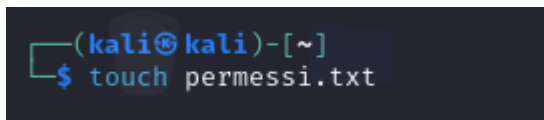
1. Introduzione

Il presente documento descrive il processo di configurazione dei permessi sul file `permessi.txt`, creato con il comando `touch`. Successivamente, abbiamo analizzato i permessi predefiniti assegnati al file e li abbiamo modificati utilizzando il comando `chmod`. L'obiettivo era limitare i permessi di scrittura per l'utente e verificare il comportamento del file attraverso un editor di testo.

2. Creazione del File e Analisi Iniziale dei Permessi

Il file `permessi.txt` è stato creato con il comando:

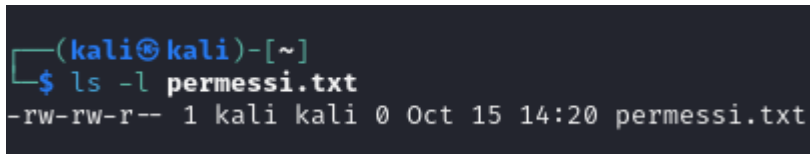
```
touch permessi.txt
```



```
(kali@kali)-[~]  
$ touch permessi.txt
```

Dopo la creazione, i permessi sono stati visualizzati con il seguente comando:

```
ls -l permessi.txt
```



```
(kali@kali)-[~]  
$ ls -l permessi.txt  
-rw-rw-r-- 1 kali kali 0 Oct 15 14:20 permessi.txt
```

L'output mostrava qualcosa di simile a:

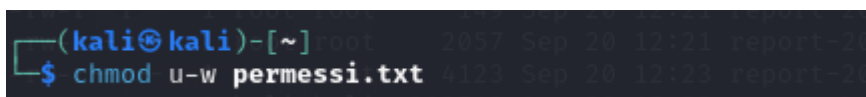
Questo indica che:

- L'utente ha permessi di lettura e scrittura (`rw-`).
- Il gruppo e gli altri utenti hanno solo permessi di lettura (`r--`).

3. Modifica dei Permessi: Rimozione del Permesso di Scrittura per l'Utente

Per ridurre i permessi dell'utente, abbiamo deciso di rimuovere il permesso di scrittura. Questa operazione è stata eseguita con il seguente comando:

```
chmod u-w permessi.txt
```



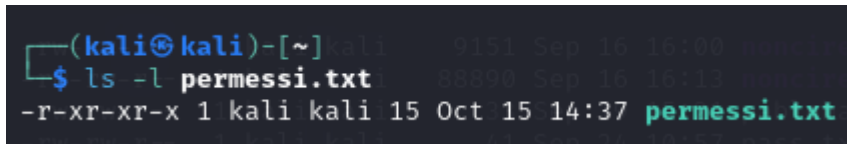
```
(kali@kali)-[~]  
$ chmod u-w permessi.txt
```

Il comando `chmod` consente di modificare i permessi di un file o di una directory. In questo caso:

- `u` indica che stiamo operando sui permessi dell'utente (owner).
- `-w` rimuove il permesso di scrittura.

Dopo l'esecuzione del comando, i permessi del file sono stati nuovamente verificati con:

```
ls -l permessi.txt
```



```
(kali㉿kali)-[~]
$ ls -l permessi.txt
-r-xr-xr-x 1 kali kali 15 Oct 15 14:37 permessi.txt
```

4. Verifica del Comportamento: Test con un Editor di Testo

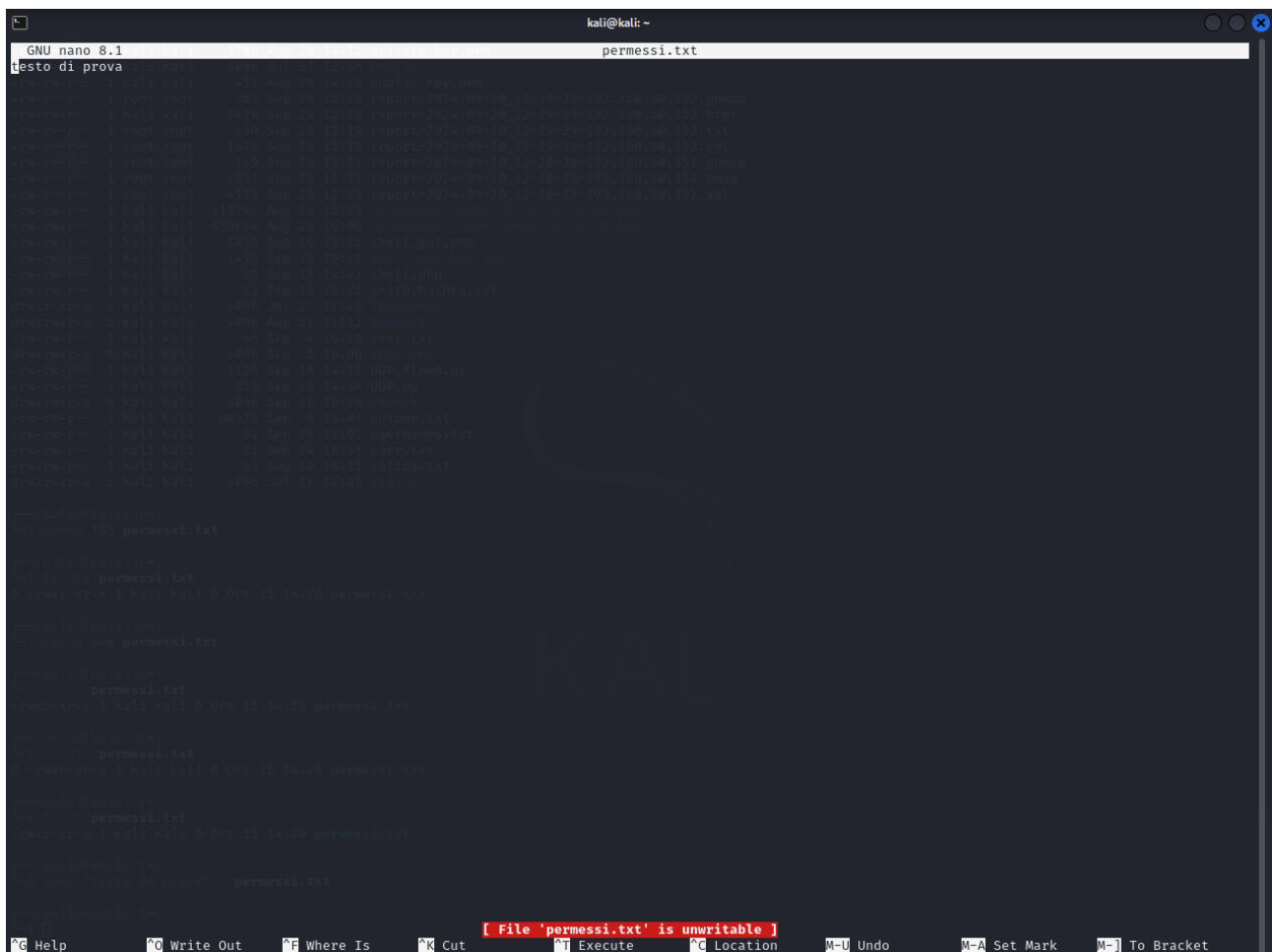
Dopo la modifica dei permessi, è stato effettuato un test per verificare che l'utente non fosse in grado di scrivere nel file. Abbiamo aperto il file con l'editor `nano` utilizzando il seguente comando:

```
nano permessi.txt
```



```
(kali㉿kali)-[~]
$ nano permessi.txt
```

L'editor ha confermato che il file non era scrivibile, mostrando un messaggio come "File non scrivibile" o impedendo il salvataggio delle modifiche senza prima confermare. Questo test ha confermato che la modifica dei permessi è stata applicata correttamente.



```
GNU nano 8.1 permessi.txt
testo di prova

[ File 'permessi.txt' is unwritable ]
PG Help  ^O Write Out  ^F Where Is  ^K Cut  ^T Execute  ^C Location  M-U Undo  M-A Set Mark  M-J To Bracket
```

5. Motivazioni delle Scelte sui Permessi

La scelta di rimuovere il permesso di scrittura per l'utente è stata fatta per dimostrare come i permessi di un file possono essere gestiti in Linux per prevenire modifiche indesiderate. In ambienti multiutente o di produzione, è comune limitare l'accesso in scrittura ai file critici per prevenire errori o manomissioni accidentali.

- **Lettura:** È stato mantenuto il permesso di lettura per consentire all'utente di visualizzare il contenuto del file.
- **Scrittura:** La rimozione del permesso di scrittura ha garantito che il file non potesse essere modificato dall'utente. Questa è una misura di sicurezza importante in contesti dove la protezione dell'integrità dei file è fondamentale.
- **Esecuzione:** Il permesso di esecuzione non è stato assegnato, poiché il file `permessi.txt` non è un file eseguibile (es. uno script o un programma binario).

6. Analisi dei Risultati

I test eseguiti hanno confermato che:

- Il file non era più modificabile dall'utente dopo la rimozione del permesso di scrittura.
- L'utente poteva comunque visualizzare il contenuto del file, garantendo un accesso limitato ma utile.
- L'editor di testo `nano` ha correttamente impedito la modifica del file, offrendo un feedback chiaro all'utente sullo stato dei permessi.

7. Conclusione

La configurazione dei permessi sul file `permessi.txt` ha dimostrato come i sistemi basati su Linux gestiscono in modo efficiente l'accesso ai file tramite il meccanismo di permessi. La modifica dei permessi ha fornito un controllo preciso sull'accessibilità, consentendo la lettura ma prevenendo modifiche non autorizzate. Questo tipo di gestione è fondamentale per mantenere la sicurezza e l'integrità dei dati in un ambiente condiviso o di produzione.