# Report: Manipolazione Permessi File System

**Data:** 10 Febbraio 2026

**Sistema Operativo:** Kali Linux

**Utente:** ***kali***

## 1) Introduzione

Nei sistemi operativi **Unix-like**, la sicurezza del dato poggia sul sistema di controllo degli accessi basato sui permessi del file system. Ogni risorsa è associata a un proprietario e a un gruppo, e l'accesso è regolato da una triade di privilegi: lettura (**r**), scrittura (**w**) ed esecuzione (**x**). Comprendere e saper manipolare questi parametri tramite il comando ***chmod*** è una competenza fondamentale per un analista di sicurezza, poiché configurazioni errate rappresentano uno dei vettori di attacco più comuni per l'escalation dei privilegi o la compromissione dell'integrità dei dati.

## 2) Obiettivo

Creazione di un file e gestione dei permessi di accesso (Read, Write, Execute).

## 3) Analisi delle Operazioni

### Fase 1: Creazione e stato Iniziale

Il file ***forza_napoli.txt*** è stato creato inizialmente con l'editor di testo ***nano***. Dallo stato dei permessi mostrato nel terzo screenshot, il file aveva originariamente un'impostazione di (`-rw-rw-r--`):

- **User/Owner:** Lettura e Scrittura.
- **Group:** Lettura e Scrittura.
- **Others:** Solo Lettura.

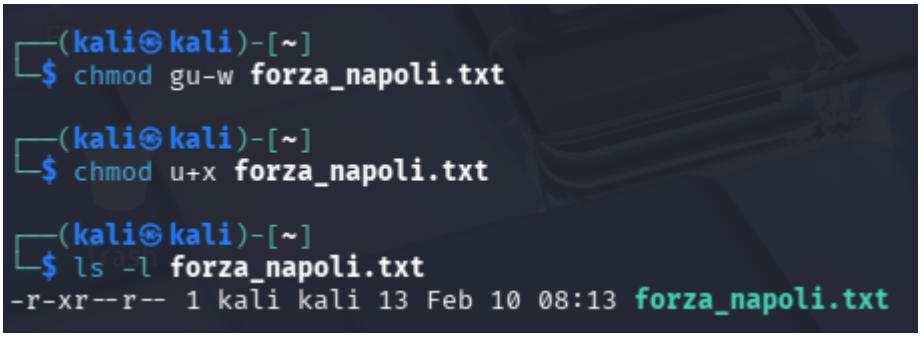

### Fase 2: Modifica dei Permessi (chmod)

Nello screenshot, sono stati eseguiti due comandi cruciali:

1. ***chmod gu-w forza_napoli.txt***: Hai rimosso il permesso di **scrittura** (w) sia all'utente proprietario (u) che al gruppo (g).
2. ***chmod u+x forza_napoli.txt***: Hai aggiunto il permesso di **esecuzione** (x) solo per l'utente proprietario.

Il risultato finale (verificato con `ls -l`) è diventato **`-r-xr--r--`**:

- **User:** Sola lettura ed esecuzione (niente scrittura).
- **Group/Others:** Sola lettura.




**Fase 3: Test di Sicurezza (Access Control)**

Nell'ultimo screenshot, hai tentato di sovrascrivere il file con il comando:

- ***`echo "forza juve" > forza_napoli.txt`***

Il sistema ha risposto correttamente con **`zsh: permission denied`**. Questo avviene perché, nonostante tu sia il proprietario del file, hai esplicitamente rimosso il tuo permesso di scrittura nel passaggio precedente.


## 4) Conclusione

L'analisi condotta conferma che il meccanismo di **Access Control** del kernel Linux ha operato in conformità con le policy definite dall'utente. Attraverso l'uso del comando ***`chmod`***, è stato possibile alterare la maschera dei permessi del file, passando da una configurazione di default a una restrittiva.

In ottica di **Cyber Security**, questa attività evidenzia l'importanza del **Hardening del File System**: limitare i permessi di scrittura ai soli file strettamente necessari riduce drasticamente la superficie di attacco, impedendo a eventuali processi compromessi o utenti non autorizzati di iniettare codice malevolo o modificare configurazioni critiche.