Relazione sull'Esercizio: Gestione dei Permessi in Linux

A cura di Iris Canole

1. Introduzione e Obiettivo dell'Esercizio

La gestione dei permessi nei sistemi operativi basati su kernel Linux rappresenta uno dei pilastri fondamentali della sicurezza informatica. Una corretta e granulare configurazione dei privilegi di accesso—lettura (`r`), scrittura (`w`) ed esecuzione (`x`)—è essenziale per proteggere l'integrità dei dati, garantire la riservatezza delle informazioni e prevenire accessi o modifiche non autorizzate che potrebbero compromettere la stabilità e la sicurezza dell'intero sistema.

Ogni file e directory possiede un set di permessi specifici per il proprietario, il gruppo di appartenenza e tutti gli altri utenti, e la loro gestione consapevole è una competenza imprescindibile.

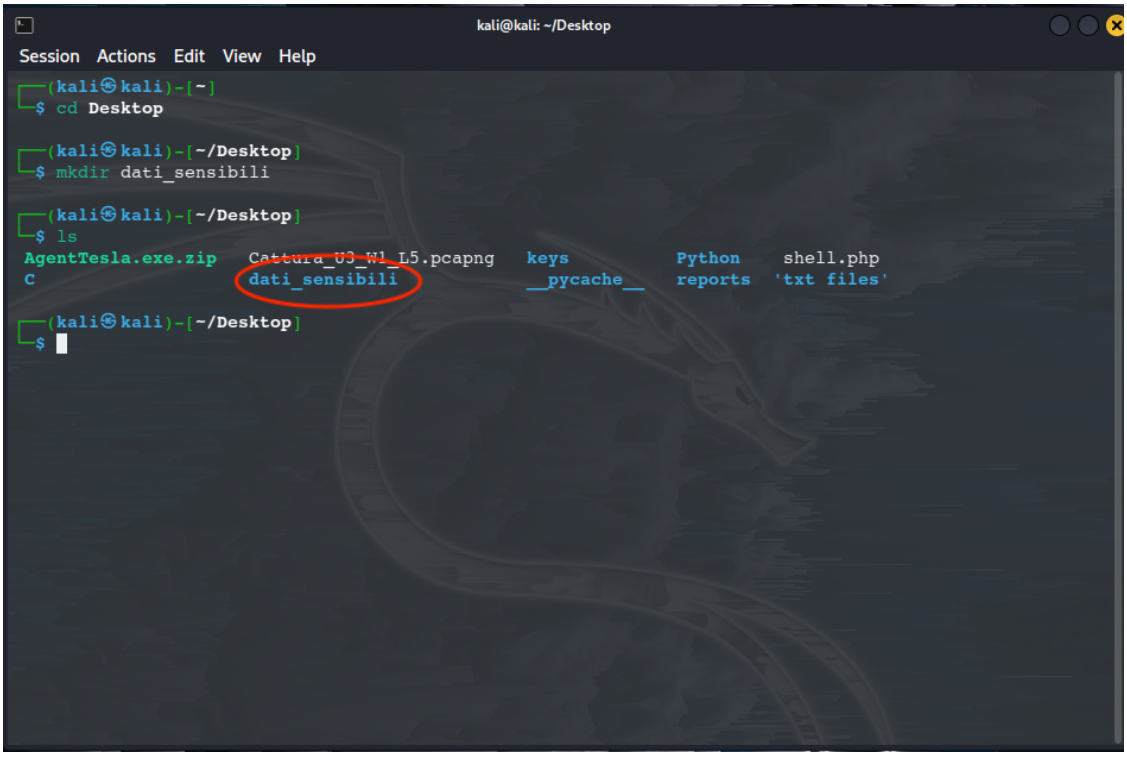
L'obiettivo di questo esercizio è quello di configurare e gestire in modo pratico i permessi per una risorsa (file o directory) su un sistema Linux. L'attività prevede di documentare ogni passaggio tecnico, motivando in modo analitico le scelte di configurazione adottate in relazione a un ipotetico scenario di sicurezza.
Questa relazione documenta, nella sezione seguente, lo svolgimento pratico delle operazioni eseguite a terminale.

2. Svolgimento Pratico dell'Esercizio

In questa sezione vengono presentati, in ordine cronologico, i passaggi tecnici eseguiti per portare a termine l'esercizio. Ogni fase è corredata da una breve descrizione che documenta l'operazione svolta.

2.1. Creazione della Risorsa (File/Directory)

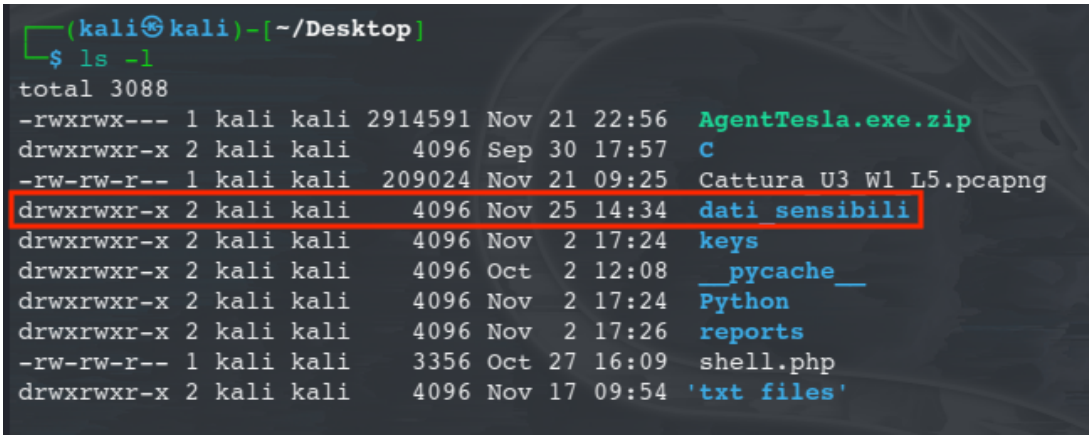
L'esercizio ha inizio con la creazione di una risorsa su cui operare. È stata creata una directory chiamata dati_sensibili per simulare un contenitore destinato a ospitare informazioni riservate, accessibili esclusivamente a un utente specifico.



2.2. Verifica dei Permessi Iniziali

Prima di procedere con qualsiasi modifica, è cruciale verificare i permessi che il sistema operativo assegna di default alla nuova risorsa. Questa analisi preliminare permette di comprendere lo stato di partenza e di pianificare le modifiche necessarie per raggiungere il livello di sicurezza desiderato.

Utilizzando il comando `ls -l`, è stato possibile ispezionare i permessi iniziali. I permessi di default per una nuova directory sono `drwxrwxr-x`.



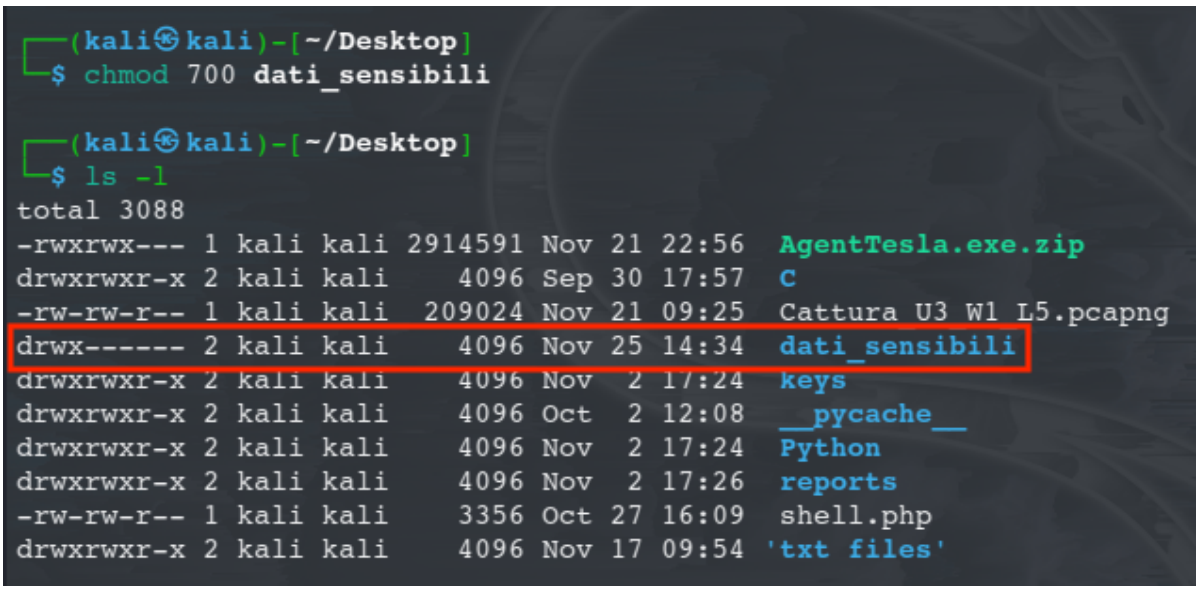
Con questi permessi, il **proprietario** ha pieni poteri (lettura, scrittura, esecuzione), il **gruppo** può entrare nella directory (`cd`), listarne il contenuto (`ls`), può creare, rinominare ed eliminare i file e può accedere e attraversare la directory, mentre **qualsiasi altro utente** sul sistema che **non** è il proprietario e **non** appartiene al gruppo proprietario della directory può elencare i contenuti della

directory e accedere e attraversare la directory, ma non creare, eliminare o rinominare file al suo interno, azione che richiederebbe il permesso di scrittura (w).

2.3. Modifica dei Permessi

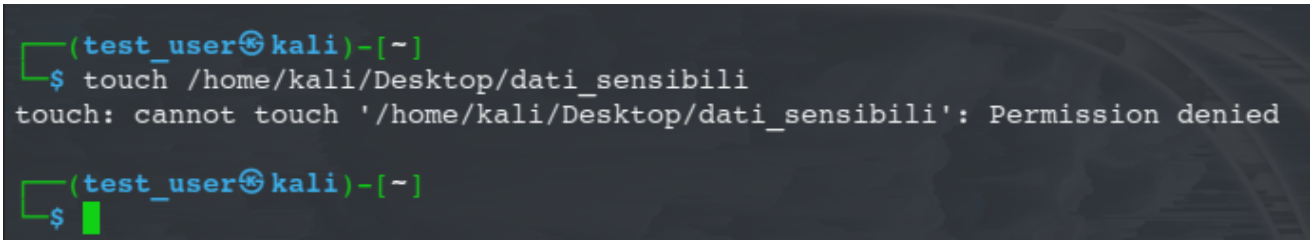
I permessi di default non erano adeguati allo scenario ipotizzato, in quanto consentivano a qualsiasi utente del sistema di accedere e leggere il contenuto della directory dati_sensibili. Per questo motivo, è stato necessario restringere drasticamente l'accesso.

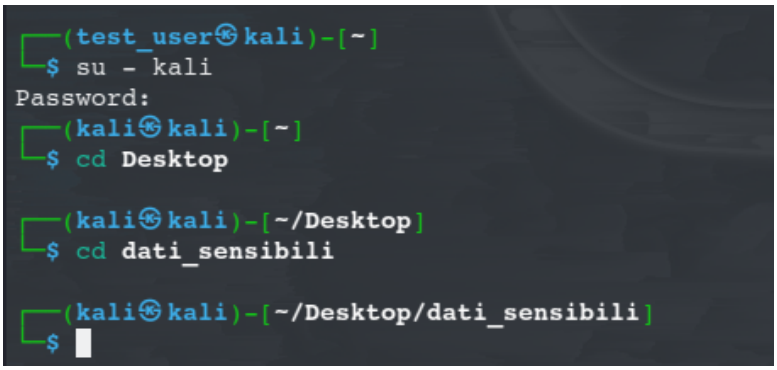
Con il comando `chmod 700 dati_sensibili` sono stati rimossi tutti i permessi per il gruppo e per gli altri utenti, lasciando l'accesso completo (`rwx`) esclusivamente al proprietario. La notazione ottale `700` si traduce in `rwx` per il proprietario, e **nessun permesso** (`---`) per gruppo e altri. La successiva verifica con `ls -l` conferma l'avvenuta modifica.



2.4. Test dei Nuovi Permessi

Per verificare l'efficacia delle nuove regole, è stato condotto un test pratico. È stato effettuato un tentativo di accesso alla directory dati_sensibili da un utente diverso dal proprietario.
L'output del terminale, che mostra un errore "Permission denied", conferma il successo della configurazione. Il sistema operativo ha correttamente impedito l'accesso all'utente non autorizzato, agendo in conformità con i permessi restrittivi impostati.
Il completamento di questa fase pratica ci permette ora di passare a un'analisi più approfondita delle decisioni strategiche che hanno guidato l'esercizio.





3. Relazione e Analisi delle Scelte

Questa sezione analizza il ragionamento alla base delle configurazioni di sicurezza implementate.

3.1. Motivazione delle Scelte sui Permessi

La scelta di impostare i permessi a `700 (rwx------)` è stata guidata dal **principio del privilegio minimo** (Principle of Least Privilege), un concetto cardine della sicurezza informatica.

Secondo questo principio, a un'entità (utente, processo, etc.) devono essere concessi solo i permessi strettamente necessari per svolgere le proprie funzioni legittime, e nient'altro.

Applicando questo principio al nostro scenario:

- **Proprietario (`rwx`):** L'utente proprietario necessita del pieno controllo sulla directory per poter creare, leggere, modificare ed eliminare i file contenuti al suo interno. Il permesso di esecuzione (`x`) è indispensabile per poter accedere (`cd`) alla directory stessa.
- **Gruppo (`---`):** Poiché la directory contiene dati sensibili di pertinenza esclusiva del proprietario, non vi è alcuna ragione per cui il gruppo di appartenenza debba avere visibilità o accesso. Rimuovere tutti i permessi al gruppo previene la condivisione involontaria di informazioni.
- **Altri (`---`):** A maggior ragione, nessun altro utente del sistema deve poter accedere, neanche in sola lettura. Negare tutti i permessi a questa categoria è il passo fondamentale per isolare la risorsa e garantirne la confidenzialità.

Questa configurazione garantisce la massima riservatezza per i file contenuti nella directory

`dati_sensibili`.

3.2. Analisi dei Risultati Ottenuti

I risultati del test descritto al punto 2.4 hanno confermato che il comportamento del sistema è stato esattamente quello previsto e desiderato. Il tentativo di accesso da parte di un utente non proprietario si è concluso con un messaggio di errore inequivocabile: `Permission denied`.

L'errore è la conseguenza della rimozione del permesso di esecuzione (`x`) per la categoria "altri". In Linux, i permessi su una directory hanno un significato specifico:

- `r` (read) consente di listare i nomi dei file contenuti
- `x` (execute) concede il permesso di *attraversare* l'inode della directory per accedere ai dati dei file al suo interno. Il permesso x agisce quindi come un "cancello" di accesso. Senza di esso, il sistema non può nemmeno accedere al blocco dati della directory per leggerne il contenuto, rendendo di fatto inutile il permesso r per un utente non autorizzato.

La mancanza del flag `x` è stata quindi la causa diretta che ha impedito all'utente di entrare nella cartella, bloccando sul nascere qualsiasi tentativo di interazione.

4. Conclusioni

L'esercizio ha permesso di consolidare le conoscenze pratiche e teoriche sulla gestione dei permessi nel sistema operativo Linux. Attraverso la creazione, la modifica e la verifica dei privilegi di accesso, è stato possibile applicare concretamente concetti di sicurezza fondamentali come il principio del privilegio minimo, osservandone gli effetti diretti sul comportamento del sistema.

Si ribadisce l'importanza critica di questa competenza: una gestione dei permessi attenta e consapevole è una delle prime e più efficaci linee di difesa contro accessi non autorizzati, data breach e compromissioni del sistema. Per qualunque professionista che operi nel settore IT e, in particolare, nella sicurezza informatica, la padronanza di questi strumenti è un requisito non negoziabile.

In conclusione, l'obiettivo dell'esercizio è stato raggiunto con successo, avendo dimostrato la capacità di manipolare e verificare i permessi in modo consapevole, documentato e motivato da solidi principi di sicurezza.