REPORT S10-L2: Gestione Permessi Linux (chmod)

Studente: Vincenzo Zarola
Corso: Cybersecurity Specialist
Oggetto: Configurazione e hardening dei permessi su file script in ambiente Linux.

1. Obiettivo dell'Esercitazione

L'obiettivo dell'attività è acquisire competenza nella **gestione** dei **permessi** (Access Control) su sistemi Linux. L'esercizio prevede la creazione di un file, l'analisi dei permessi predefiniti e la successiva applicazione di una politica di sicurezza **restrittiva** per garantire l'integrità e la riservatezza del file.

---

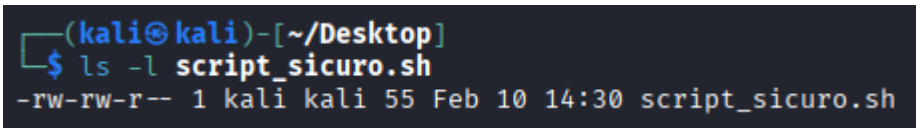
2. Procedura Esecutiva

Fase 1: Creazione della Risorsa

È stato creato un nuovo file script denominato **script_sicuro.sh** contenente un semplice comando di output. Per la creazione è stato utilizzato il comando **echo** con reindirizzamento dell'output (**>**).

Fase 2: Verifica dei Permessi Predefiniti

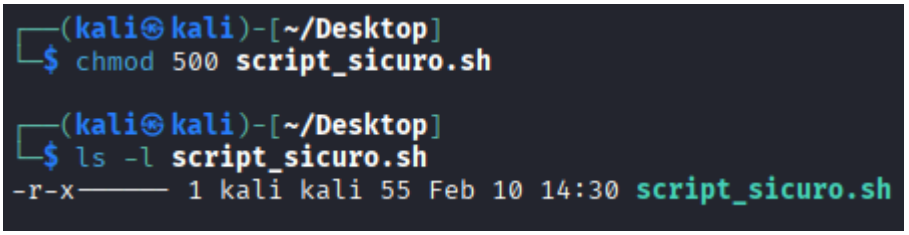
Prima di applicare le modifiche, è stato verificato lo stato iniziale dei permessi tramite il comando **ls -l**. Dallo screenshot sottostante si nota che i permessi di default assegnati dal sistema erano **-rw-rw-r--** (**664**). Questa configurazione è insicura per il nostro scenario, in quanto permette la modifica del file (scrittura) sia all'utente che al gruppo.



Fase 3: Modifica dei Permessi

Per mettere in sicurezza il file, è stato utilizzato il comando **chmod 500**. L'obiettivo era rendere il file **eseguibile** ma **immutabile** (non modificabile) anche per il proprietario stesso, e totalmente inaccessibile agli altri utenti.

- **Comando:** *chmod 500 script_sicuro.sh*
- **Risultato:** `-r-x------`

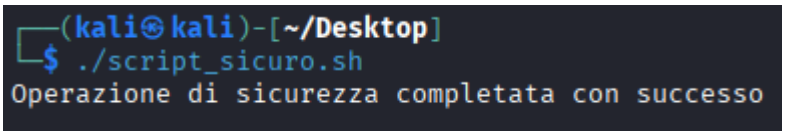


---

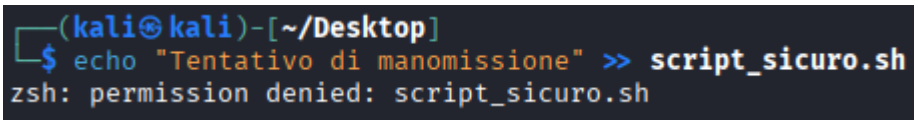
3. Test di Funzionalità e Sicurezza

Per validare la configurazione, sono stati eseguiti due test distinti:

Test A: Verifica Esecuzione (Successo) È stato tentato l'avvio dello script tramite **./script_sicuro.sh**. Poiché il bit di esecuzione (**x**) è stato impostato per il proprietario, l'operazione è andata a buon fine.



Test B: Verifica Integrità/Scrittura (Blocco) È stato tentato di modificare il contenuto del file aggiungendo una riga di testo (append **>>**). Come previsto, il sistema ha bloccato l'operazione restituendo l'errore **Permission denied**. Questo conferma che la **protezione** contro le modifiche accidentali o malevole è **attiva**.



---

4. Conclusioni

La scelta di impostare i permessi a **500 (r-x------)** risponde a precisi requisiti di sicurezza:

1. **Motivazione dei Permessi Utente (User: 5 / r-x):**
    - **Lettura (r):** Necessaria affinché l'interprete (bash) possa leggere le istruzioni contenute nel file.
    - **Esecuzione (x):** Necessaria per lanciare lo script come programma.
    - **Scrittura (w) NEGATA:** Ho rimosso volontariamente il permesso di scrittura. Questo impedisce che uno script critico venga alterato per errore o sovrascritto, garantendone l'**Integrità**.
2. **Motivazione per Gruppo e Altri (Group/Others: 0 / ---):**
    - Tutti i permessi sono stati revocati per garantire la massima **Riservatezza**. Nessun altro utente del sistema può vedere il contenuto dello script o eseguirlo.

Conclusione: I test effettuati confermano che la configurazione è efficace: lo script è utilizzabile per il suo scopo (esecuzione) ma protetto da alterazioni non autorizzate.