

# XSS & SQL Injection

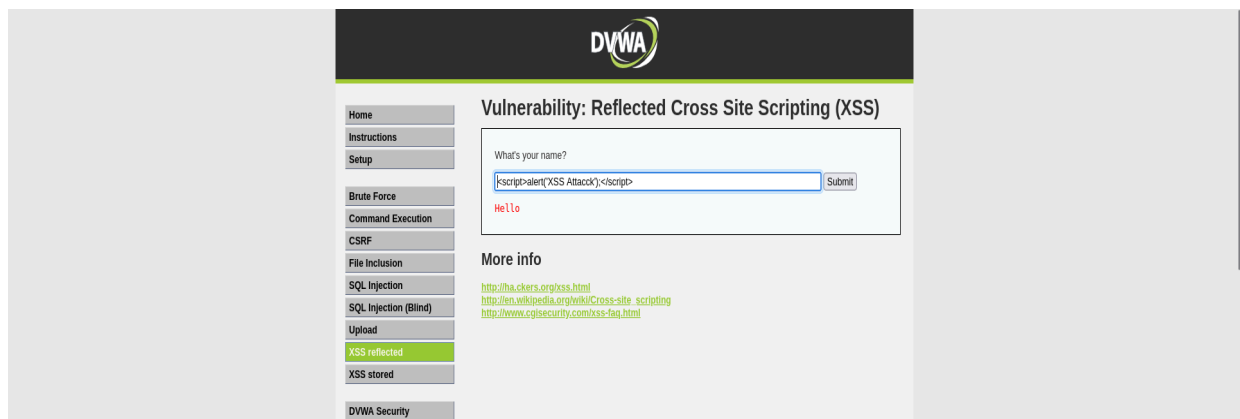
## Obiettivo

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

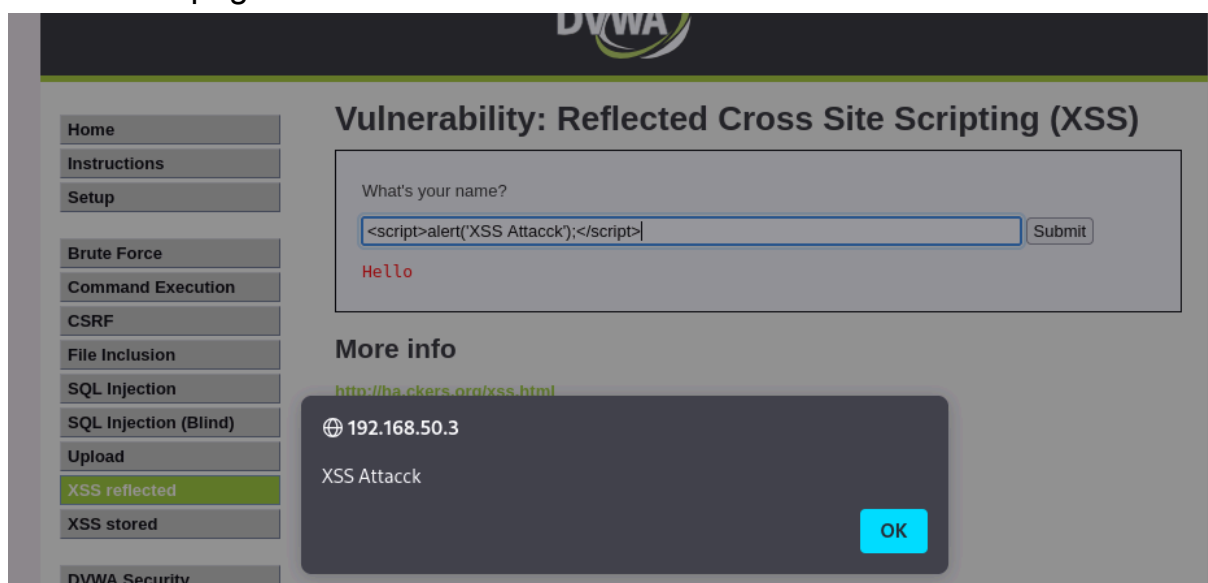
## XSS Reflected

### Security Low

Andiamo ora ad analizzare nello specifico ciò che concerne un tipo di attacco XSS Reflected. Come possiamo vedere negli screen sottostanti un attaccante potrebbe andare ad inserire del codice JavaScript all'interno del campo di ricerca (In questo caso `<script> alert('XSS Attack');</script>`)



Il server prende l'input dell'utente e lo inserisce direttamente nel contenuto HTML della pagina dei risultati.



## Security Medium

Per quanto riguarda il caso in cui la security impostata su Medium come possiamo vedere viene effettuata una 'sanificazione' dell'input (`echo 'Hello' . str_replace('<script>', '', $_GET['name']);`);

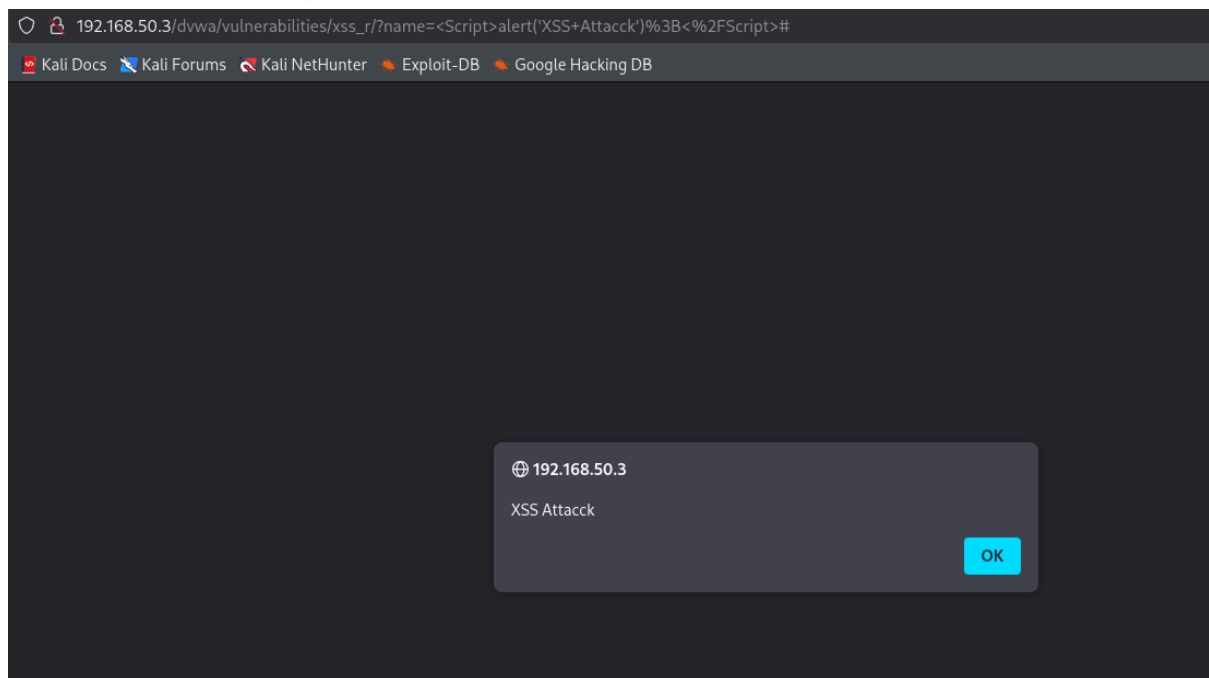
Medium Reflected XSS Source

```
<?php
if(!array_key_exists('name', $_GET) || $_GET['name'] == NULL || $_GET['name'] == ''){
    $isempty = true;
} else {
    echo "<pre>";
    echo 'Hello' . str_replace('<script>', '', $_GET['name']);
    echo "</pre>";
}
?>
```

Ma questo tipo di sanificazione non è abbastanza in quanto essendo i tag di javascript casesensitive è possibile utilizzare lo stesso comando modificandolo per ottenere lo stesso risultato come mostrato in figura. Quindi modificando il tag con `<Script>alert('testo');</Script>`

The screenshot shows the DVWA web application interface. At the top, the DVWA logo is visible. On the left, there is a sidebar menu with various security challenges: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, SQL Injection, SQL Injection (Blind), Upload, XSS reflected (highlighted in green), XSS stored, DVWA Security, PHP Info, About, and Logout. The main content area is titled "Vulnerability: Reflected Cross Site Scripting (XSS)". It contains a form with the label "What's your name?" and a text input field containing the payload `<Script>alert('XSS Attacck');</Script>`. A "Submit" button is next to the input field. Below the form, there is a "More info" section with three links: <http://hackers.org/xss.html>, [http://en.wikipedia.org/wiki/Cross-site\\_scripting](http://en.wikipedia.org/wiki/Cross-site_scripting), and <http://www.cgisecurity.com/xss-faq.html>. At the bottom left, the user information is displayed: "Username: admin", "Security Level: medium", and "PHPIDS: disabled". At the bottom right, there are two buttons: "View Source" and "View Help".

Otteniamo come aspettato



## SQL Injection

Un attacco di tipo SQL injection SQLi permette ad un utente non autorizzato di prendere il controllo sui comandi SQL utilizzati da un'applicazione Web. La query è progettata per estrarre i nomi delle tabelle dal database dvwa e visualizzarli sull'interfaccia dell'applicazione, sfruttando una vulnerabilità SQL Injection di tipo **UNION**.

**' UNION SELECT table\_name,null FROM information\_schema.tables  
WHERE table\_schema = 'dvwa' -- -**

Affinché **UNION** funzioni, il numero e il tipo di colonne della query iniettata devono corrispondere al numero e al tipo di colonne della query originale

- : È il simbolo standard di commento in SQL che dice al server di ignorare tutto ciò che segue fino alla fine della riga.
- : A volte viene aggiunto per assicurare che, se l'applicazione aggiunge automaticamente una virgoletta singola o un punto e virgola dopo l'input, il commento sia comunque valido.

Come possiamo vedere dall'esecuzione della query ci vengono restituiti i nomi delle tabelle del database DVWA.

The screenshot shows the DVWA (Damn Vulnerable Web Application) interface. On the left is a sidebar with navigation links: Home, Instructions, Setup, Brute Force, Command Execution, CSRF, File Inclusion, and SQL Injection (which is highlighted in green). The main content area is titled "Vulnerability: SQL Injection". It contains a "User ID:" label, an input field, and a "Submit" button. Below the input field, the results of a successful SQL injection are displayed in red text. The first result shows the query: `ID: ' UNION SELECT table_name,null FROM information_schema.tables WHERE table_schema = 'dvwa' -- -`, with the output: `First name: guestbook` and `Surname:`. The second result shows the query: `ID: ' UNION SELECT table_name,null FROM information_schema.tables WHERE table_schema = 'dvwa' -- -`, with the output: `First name: users` and `Surname:`.

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

### Vulnerability: SQL Injection

User ID:

```
ID: ' UNION SELECT table_name,null FROM information_schema.tables WHERE table_schema = 'dvwa' -- -
First name: guestbook
Surname:

ID: ' UNION SELECT table_name,null FROM information_schema.tables WHERE table_schema = 'dvwa' -- -
First name: users
Surname:
```