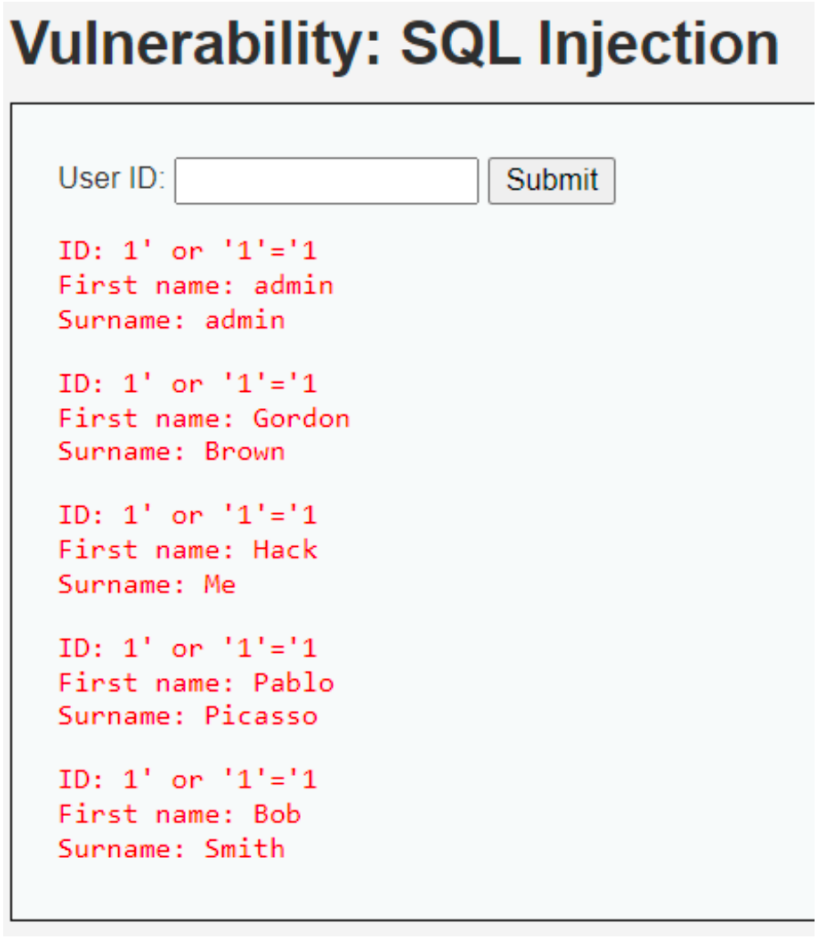
CMPE - 279 Assignment - 3

Student Name 1: Adarsh Patil
Student ID: 014749228

Student Name 2: Saurabh Dake
Student ID: 015970058

Q1. Describe the SQLi attack you used, how did you cause the user table to be dumped? What was the input string you used?

The application's text box serves as the input field when the security level is low. The user's ID must be entered into this text field. Given that it is a text box, we can carry out a SQL Injection attack by adding malicious input there and attacking the vulnerable piece of code. The string being used in this instance is **1' or '1' = '1'**. The input entered takes the place of the actual user ID that must be entered, where the second condition is always true that is the condition after OR which 1 is always equal to 1. As a result, the database will return and display all of its data as seen below.

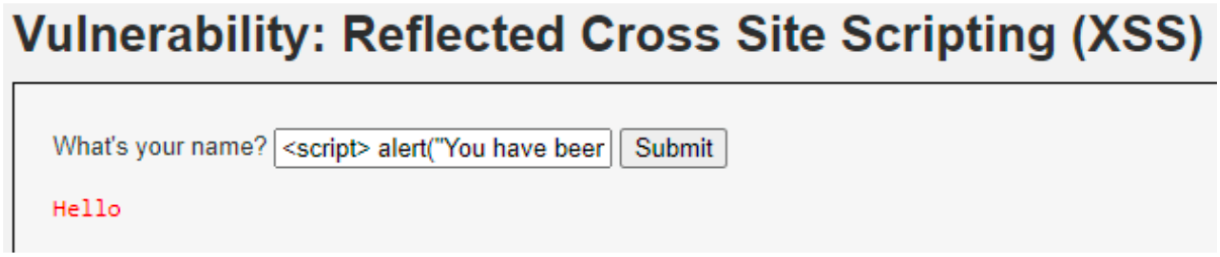


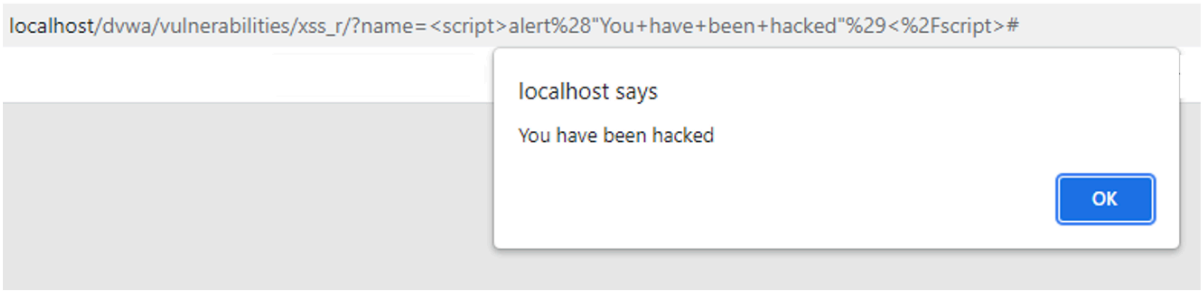
Q2. If you switch the security level in DVWA to “Medium”, does the SQLi attack still work?

The SQL injection attack that is used above would not work when the security level is changed to ‘Medium’ in DVWA. The reason being the input field is changed to a Dropdown list from the input text field. Due to the dropdown list, there is an option to select the User ID from the list. So, the vulnerability is been protected by changing the input field from Text Box to Dropdown List.

Q3. Describe the reflected XSS attack you used, how did it work?

The XSS attack is done here using the script tag in the given input text field. We will be writing some JavaScript code in between the script tag. **<script> alert("You have been hacked") </script>** is the input string that is being used here. Here the string entered will be executed by the browser and returned by the web application immediately instead of being stored like in the case of a stored XSS.





Q4. If you switch the security level in DVWA to “Medium”, does the XSS attack still work?

On switching the security level in DVWA to “Medium”, the above XSS attack does not work as before. After the security level is changed, the source code replaces the <script> tag with “” (empty). So in case of the same attacks, the content between the <script> tag gets printed.