ASSIGNMENT 3

Team Members: Sherin Thlakulathil Elias (014635504)
Anagha Sethuraman (015220517)

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

We used SQL injection attack to dump all the data from the user table. SQL injection attack is done by inserting or manipulating SQL query using input data from the client. This type of attack helps to retrieve sensitive and confidential data from the database.

Initially, the application's security level is set to 'low' and the input string 1' or '1' = '1 is passed. This will change the query as below:

*select * from users where user_id=1' or '1' = '1*

Since we have passed the or condition 1=1 in the SQL query, this will always be true and returns all the data from the users table. Another option is to append the input string 'UNION SELECT user, password FROM users' which will give similar results.

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

When the security level is changed from "Low" to "Medium", the input type for entering user id is changed to a dropdown where all the user ids will be populated, and the user can select one from them. In this case, SQL injection attack can be done by modifying the input option values by inspecting the element as shown below.

<option value ="1 or 1=1 UNION SELECT user, password FROM users#> 1 </option>

When this value is selected it will modify the SQL query and return all the data from the users table.

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

Reflected XSS attack occurs when a web application reflects a malicious script onto the victim's browser.The link that activates the script sends a request to a website with a vulnerability that lets malicious programs run there.
The reflected XSS attack we used is as follows:

With "Low" security and input string.

<script>alert("Your system is Hacked!")</script> . This displays the message "Your system is Hacked!" in the alert box.

Similarly, when a URL with injected code is sent, session cookies can be obtained.

http://localhost/dvwa/vulnerabilities/xss_r/?name=<script>alert(document.cookie)</script>

Also specified document.cookie as the window location in the script tag to reroute the victim to the configured web server.

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

On switching the security level to “Medium”, the <script> command failed to work. This was because the source code specifically looked for the presence of “<script>” and if a match was found, the request was ignored. However, JavaScript is not case-sensitive and we were able to successfully acquire the victim's session cookie by using the below command :

<SCRIPT>alert(document.cookie)</SCRIPT>