

**UNIVERSITY INSTITUTE OFENGINEERING**

**Department of Computer Science & Engineering**

**Subject Name:** Web and Mobile Security Lab

**Subject Code:** 20CSP-338

**Submitted to: Submitted by:**

Er. Himanshi Name: Sahil Kaundal

UID: 21BCS8197

Section: 616

Group: A

**INDEX**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Ex. No** | **List of Experiments** | **Conduct (MM: 12)** | **Viva**  **(MM: 10)** | **Record (MM: 8)** | **Total**  **(MM: 30)** | **Remarks/Signature** |
| 1.1 | Open any website on computer system and identify http packet on monitoring tool like Wireshark. |  |  |  |  |  |
| 1.2 | Design a method to simulate the html injection and cross site scripting to exploit the attackers. |  |  |  |  |  |
| 1.3 | Implementation of Cross site request forgery (XSRF) attack. |  |  |  |  |  |
| 2.1 | Implementation of Design methods to break authentication schemes (SQL Injection attack). |  |  |  |  |  |
| 2.2 | Program to generate message digest for the given message using the SHA/MD5 algorithm and verify the integrity of message. |  |  |  |  |  |
| 2.3 | Perform Penetration testing on a web application to gather information about the system (Foot Printing) |  |  |  |  |  |
| 2.4 | Implementation of Session hijacking attack on http-enabled website. |  |  |  |  |  |

**Experiment 7**

**Student Name:** Sahil Kaundal **UID:** 21BCS8197

**Branch:** BE CSE (Lateral Entry) **Section/Group:** 616/A

**Semester:** 5th **Date of Performance:** 02/11/2022

**Subject Name:** WMS Lab **Subject Code:** 20CSP-338

1. **Aim/Overview of the practical:**

Implementation of Session hijacking attack on http-enabled website.

1. **Task to be done/ Which logistics used:**

To Implementation of Session hijacking attack on http-enabled website.

**3. Apparatus / Simulator Used:**

* Windows 7 & above version.
* Google Chrome
* OWASP ZAP
* JHijack - a numeric session hijacking tool

**Session Hijacking:**

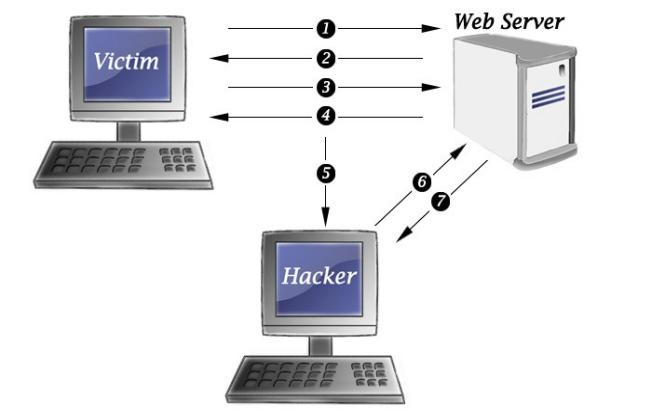
The Session Hijacking attack consists of the exploitation of the web session control mechanism, which is normally managed for a session token. Because http communication uses many different TCP connections, the web server needs a method to recognize every user’s connections. The most useful method depends on a token that the Web Server sends to the client browser after a successful client authentication. A session token is normally composed of a string of variable width and it could be used in different ways, like in the URL, in the header of the http requisition as a cookie, in other parts of the header of the http request, or yet in the body of the http requisition.

The Session Hijacking attack compromises the session token by stealing or predicting a valid session token to gain unauthorized access to the Web Server.

The session token could be compromised in different ways; the most common are:

* Predictable session token;
* Session Sniffing;
* Client-side attacks (XSS, malicious JavaScript Codes, Trojans, etc);
* [Man-in-the-middle attack](https://owasp.org/www-community/attacks/Man-in-the-middle_attack)
* [Man-in-the-browser attack](https://owasp.org/www-community/attacks/Man-in-the-browser_attack)
* Sniffer and Session Hijacking

**How Session Hijacking Works**



Session hijacking takes place when the trust of two host, services or accounts is compromised by an attacker who is known as the Man-In-The-Middle (MiTM)

Session hijacking can happen in multiple ways. There are web-based hijacks, wireless AP hijacks, also known as an evil twin attack, and TCP session-based hijacks. The principle is the same in all attacks and that is to attack the lower layers on the OSI model than the actual session is occurring. As an example, in a TCP attack, the idea is to let Layers 5 to 7 establish trust and then take the Layer 4 socket.

There are five key methods of Session hijacking: **Session Fixation**. **Session Side Jacking**. **Cross-Site Scripting**.

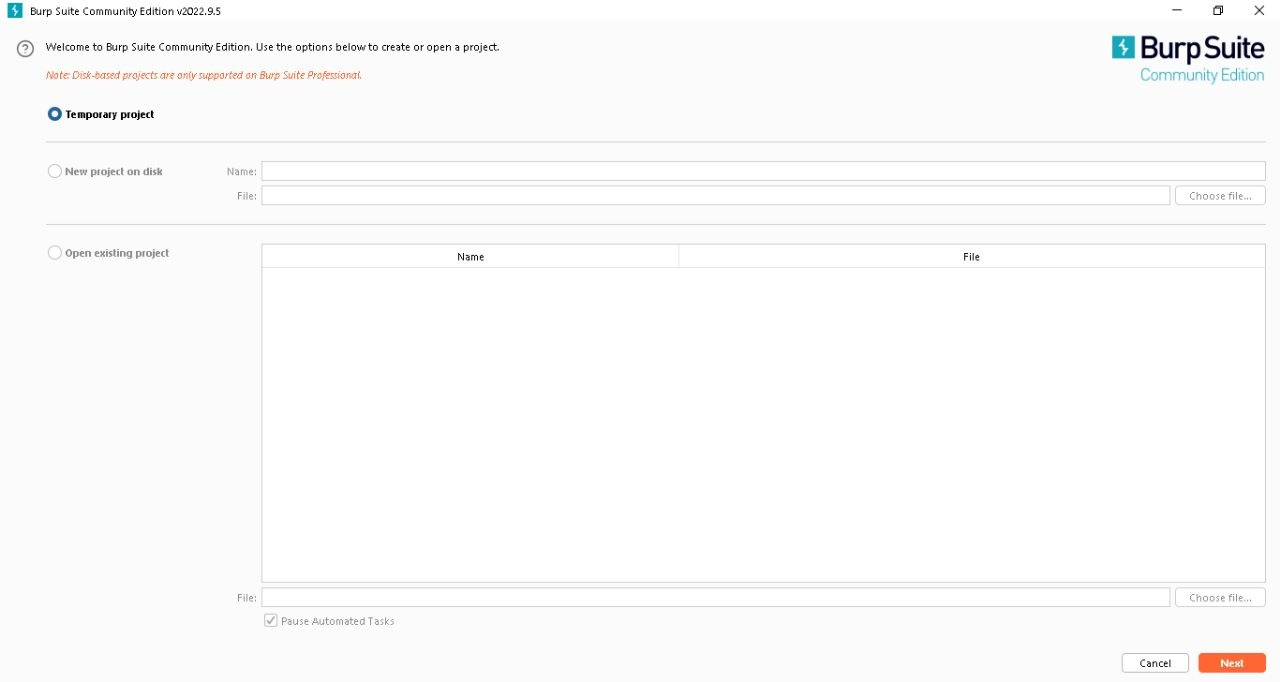
**4. Program/ Steps/ Method:**

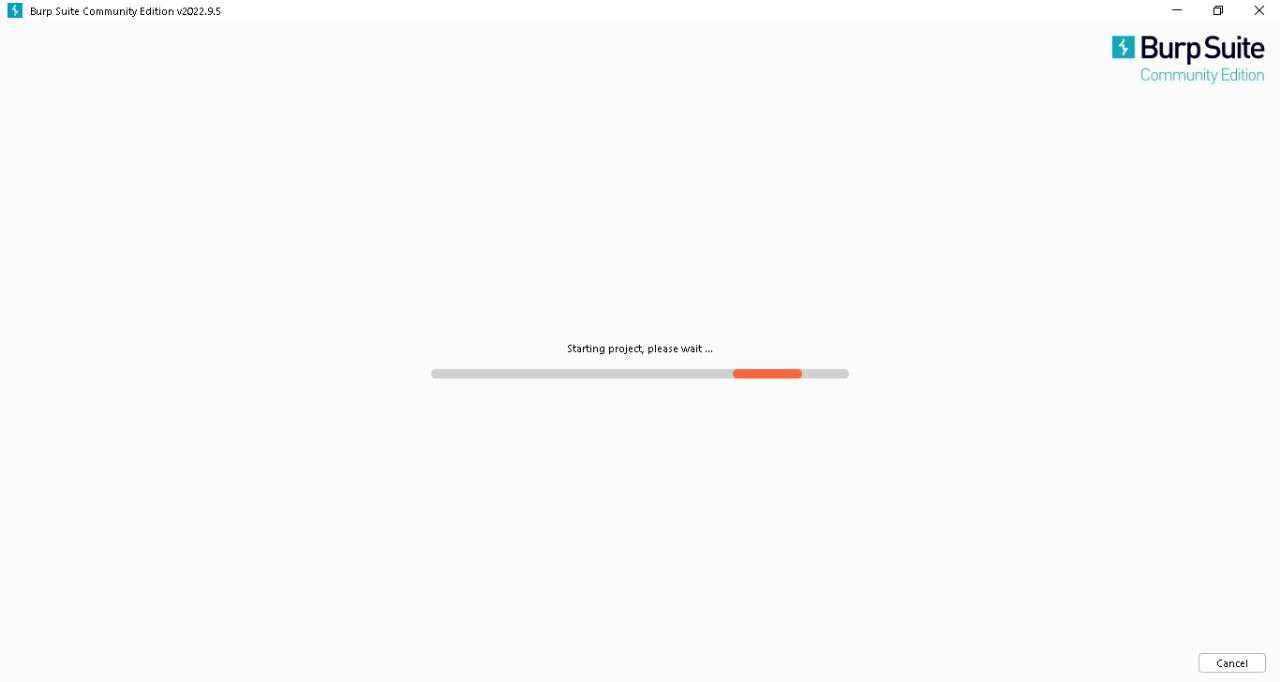
1. Login to the website as the victim and reach any page offering a secure function requiring authentication.
2. Delete from the cookie jar all the cookies which satisfy any of the following conditions.
   * in case there is no HSTS adoption: the Secure attribute is set.
   * in case there is partial HSTS adoption: the Secure attribute is set or the Domain attribute is not set.
3. Save a snapshot of the cookie jar.
4. Trigger the secure function identified at step 1.
5. Observe whether the operation at step 4 has been performed successfully. If so, the attack was successful.
6. Clear the cookie jar, login as the attacker and reach the page at step 1.
7. Write in the cookie jar, one by one, the cookies saved at step 3.
8. Trigger again the secure function identified at step 1.
9. Clear the cookie jar and login again as the victim.
10. Observe whether the operation at step 8 has been performed successfully in the victim’s account. If so, the attack was successful; otherwise, the site is secure against session hijacking.

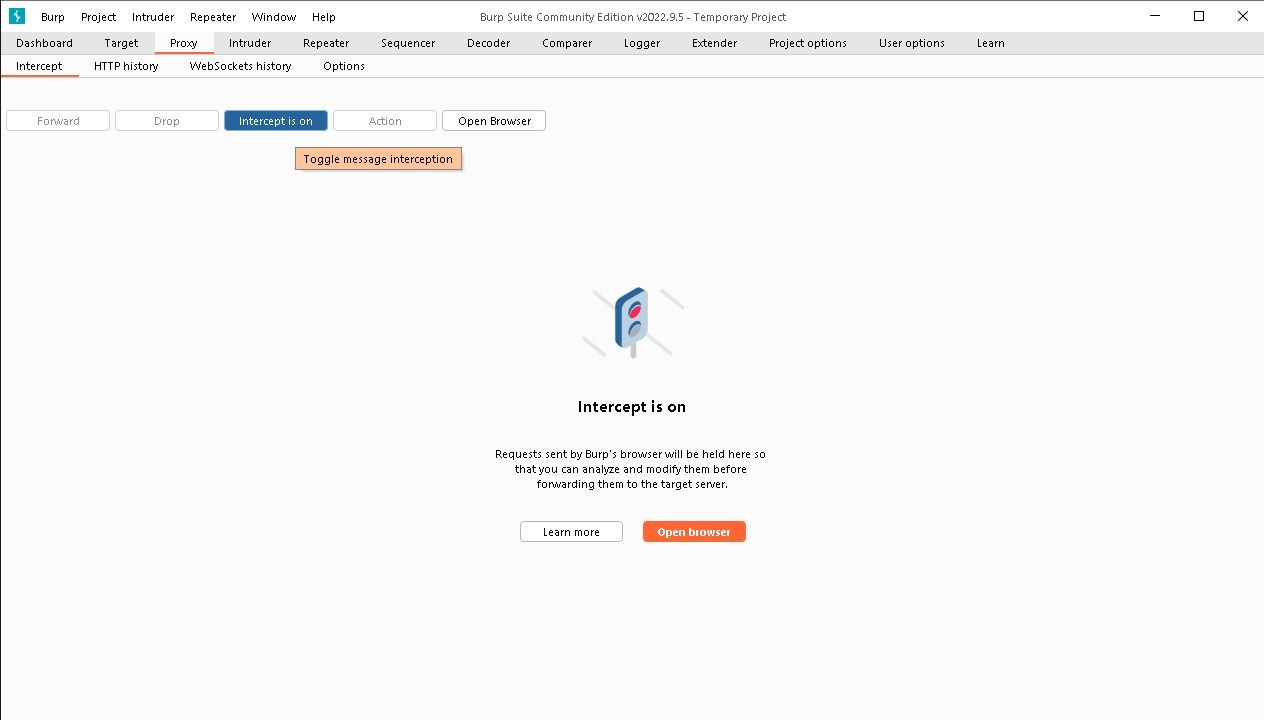
#### Session Sniffing:

1. Capture a valid token session called “Session ID” using sniffer.
2. Use the valid token session to gain unauthorized access to the Web Server.

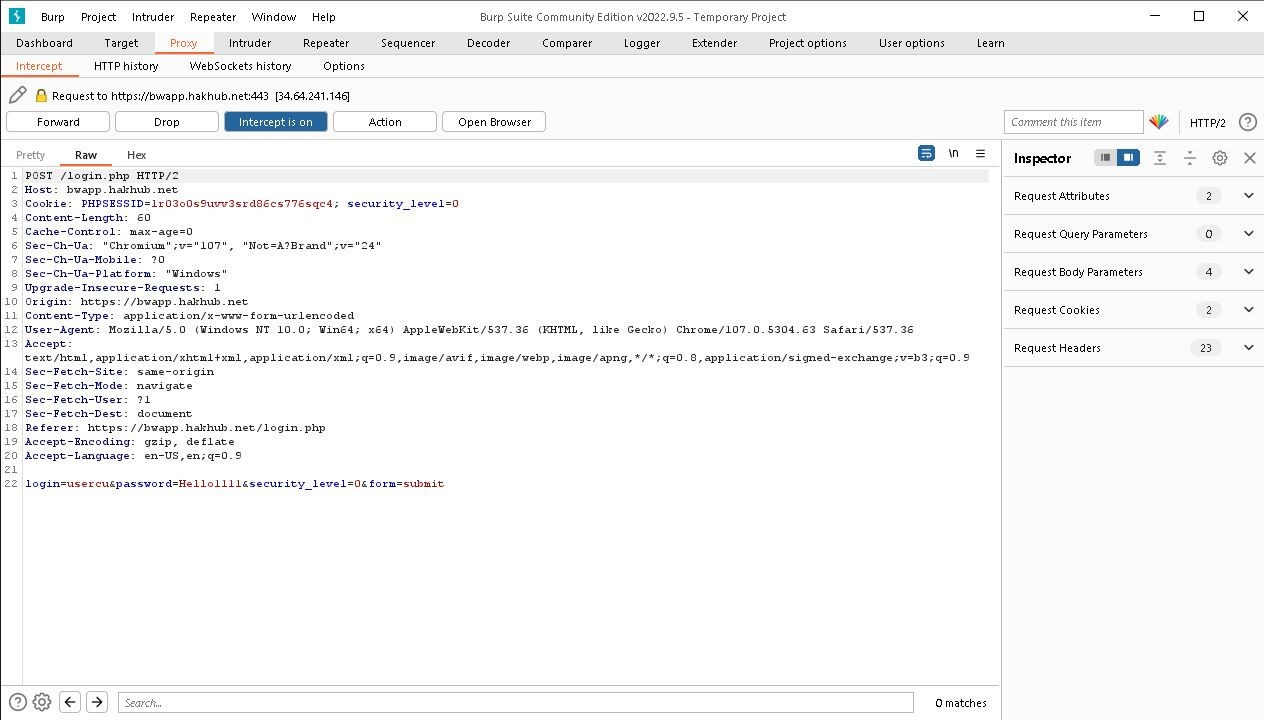
**5. DBMS Script/Result/Output/Writing Summary:**

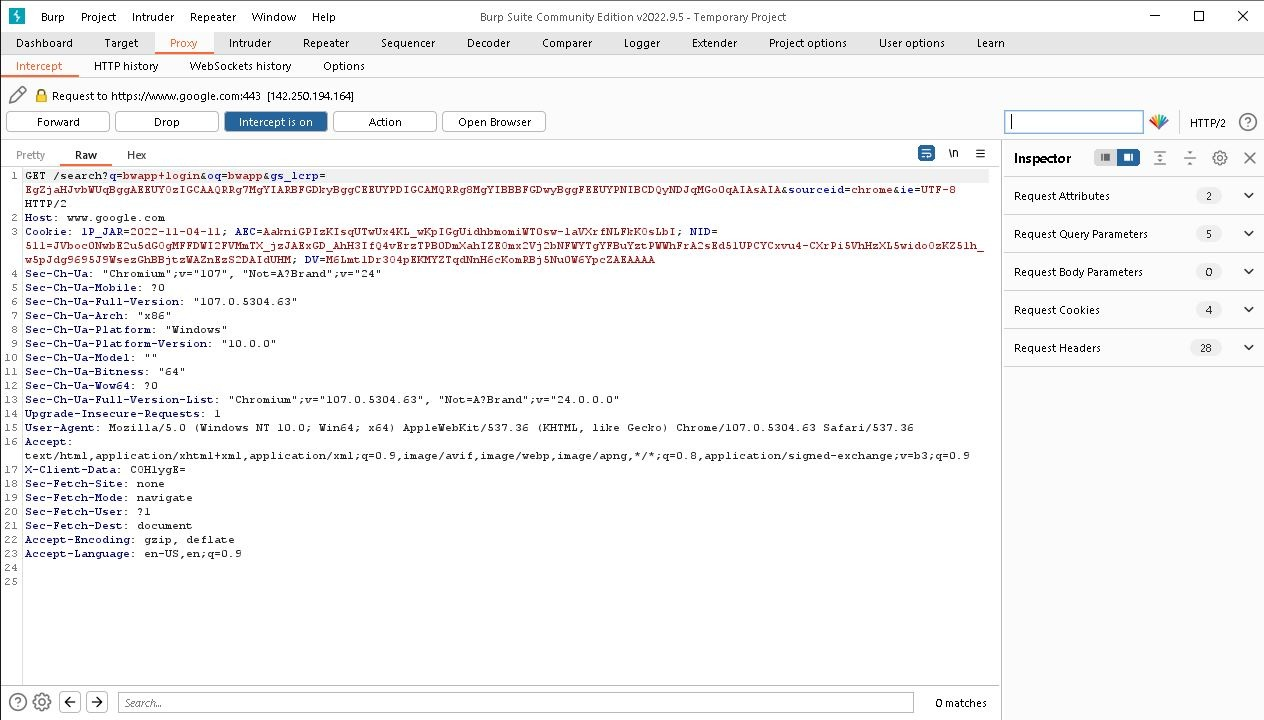


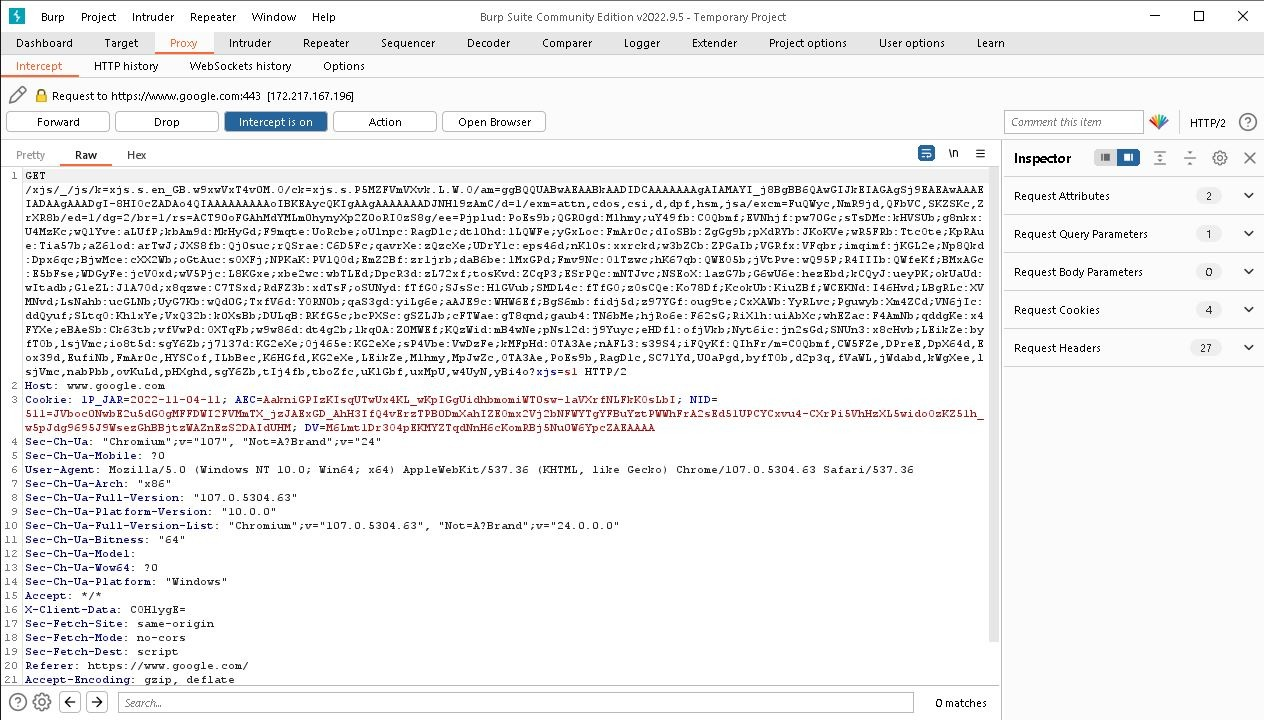


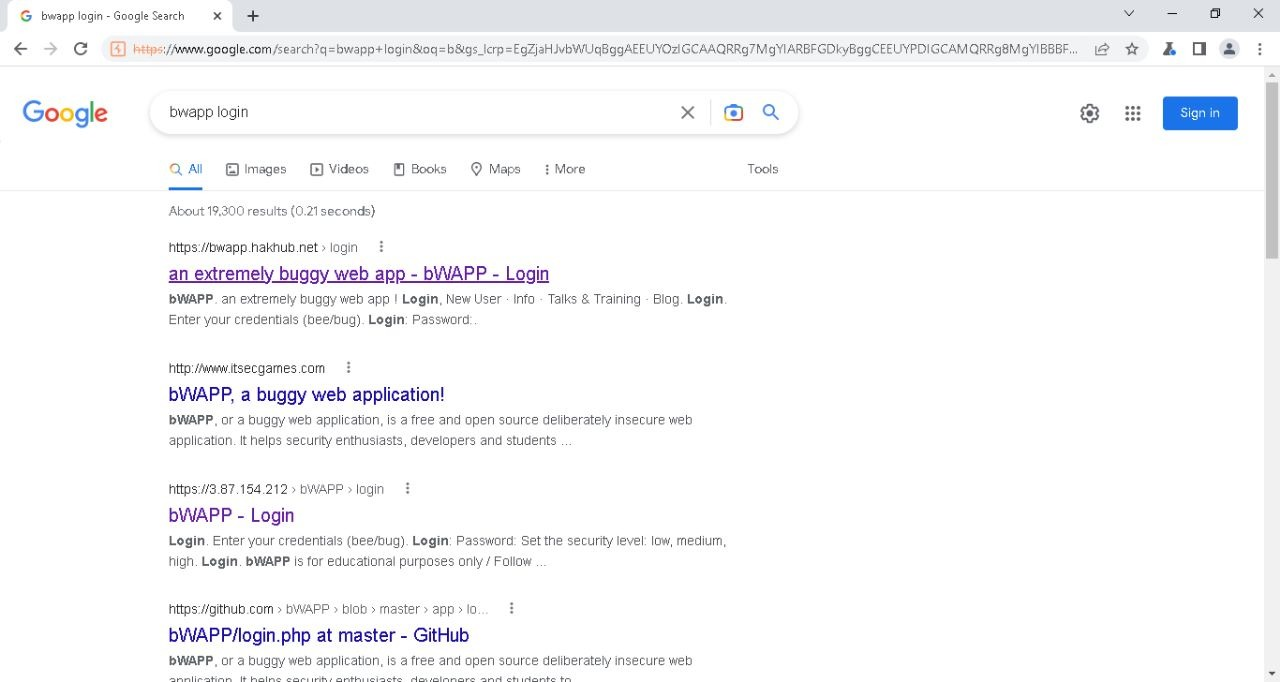


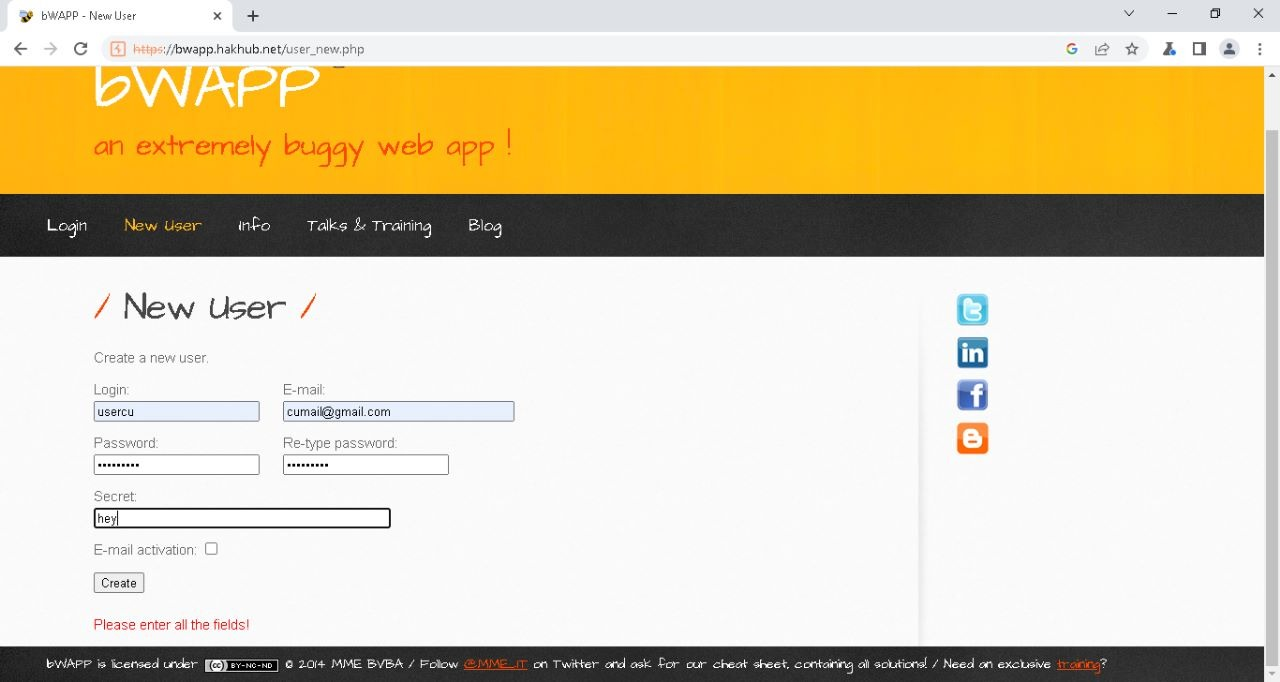


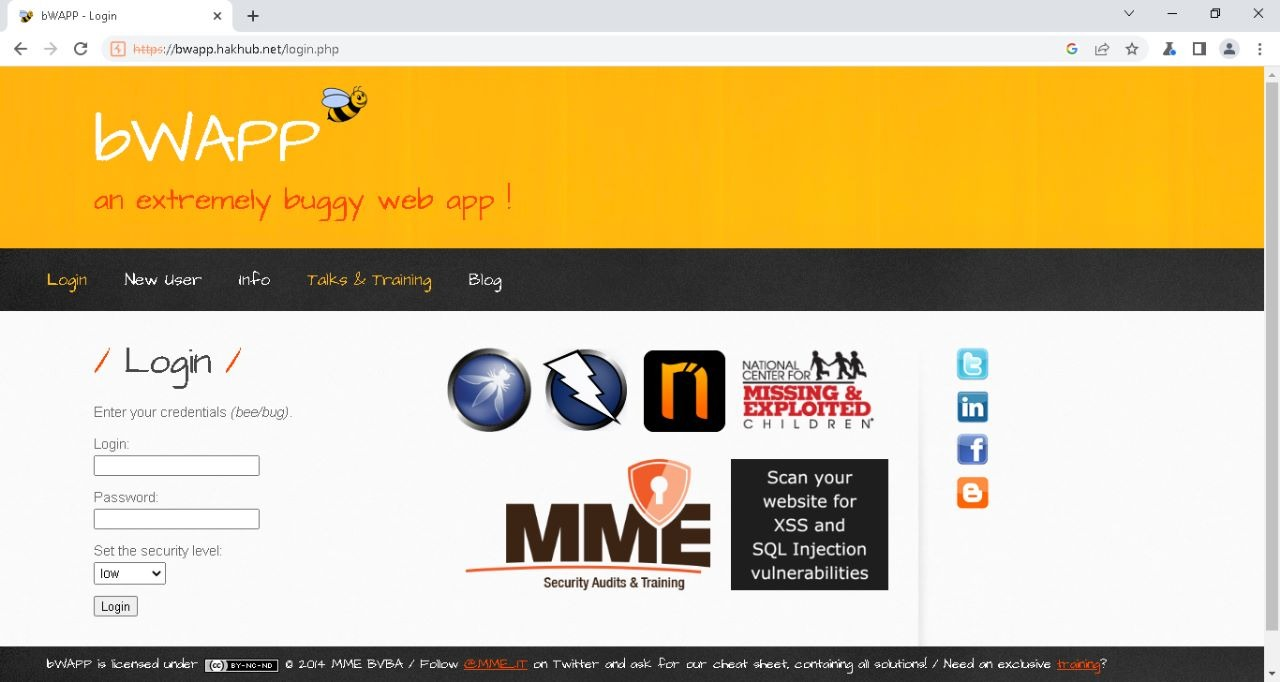


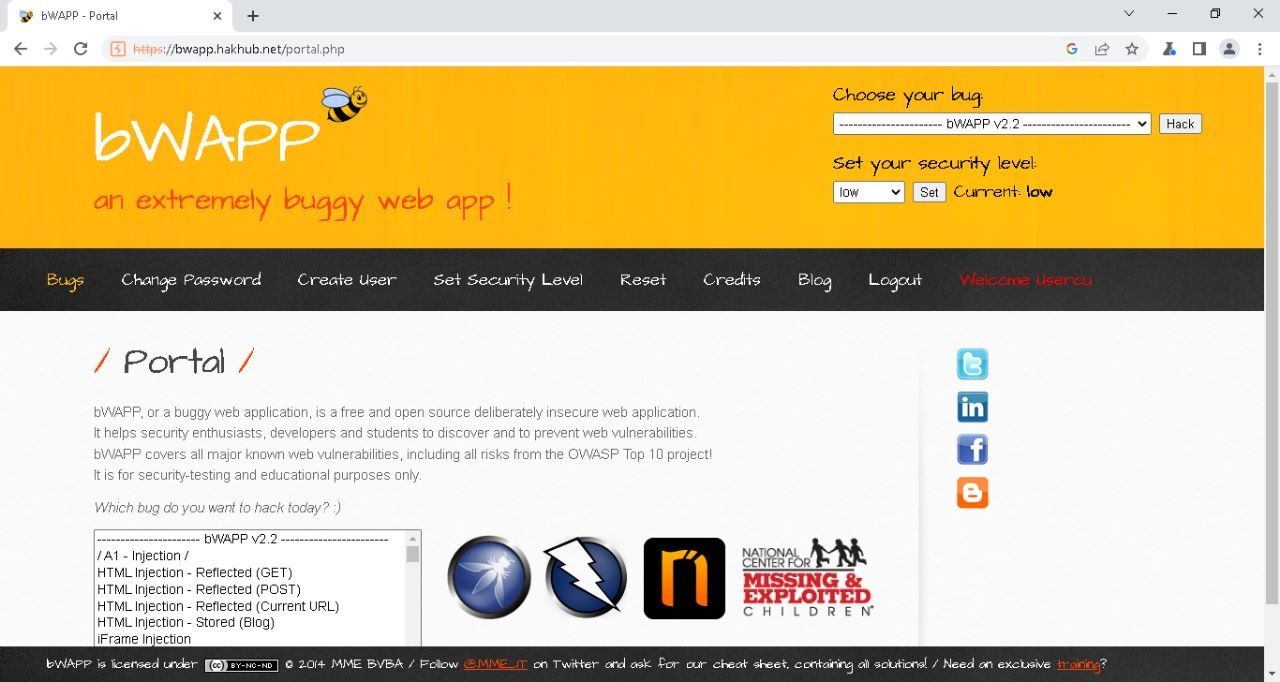


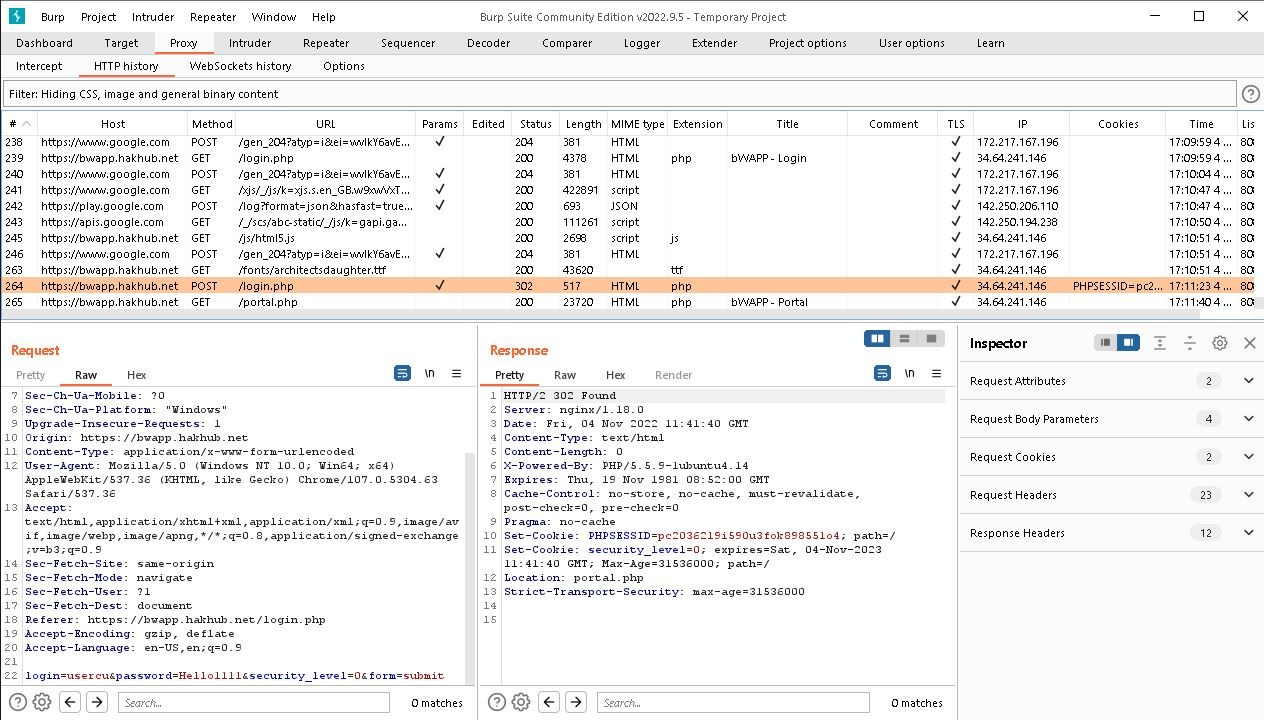












**Learning outcomes (What I have learnt):**

In the above experiment we have learnt that using session hijacking attack how the token session can be manipulated.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |