Experiment No. 5

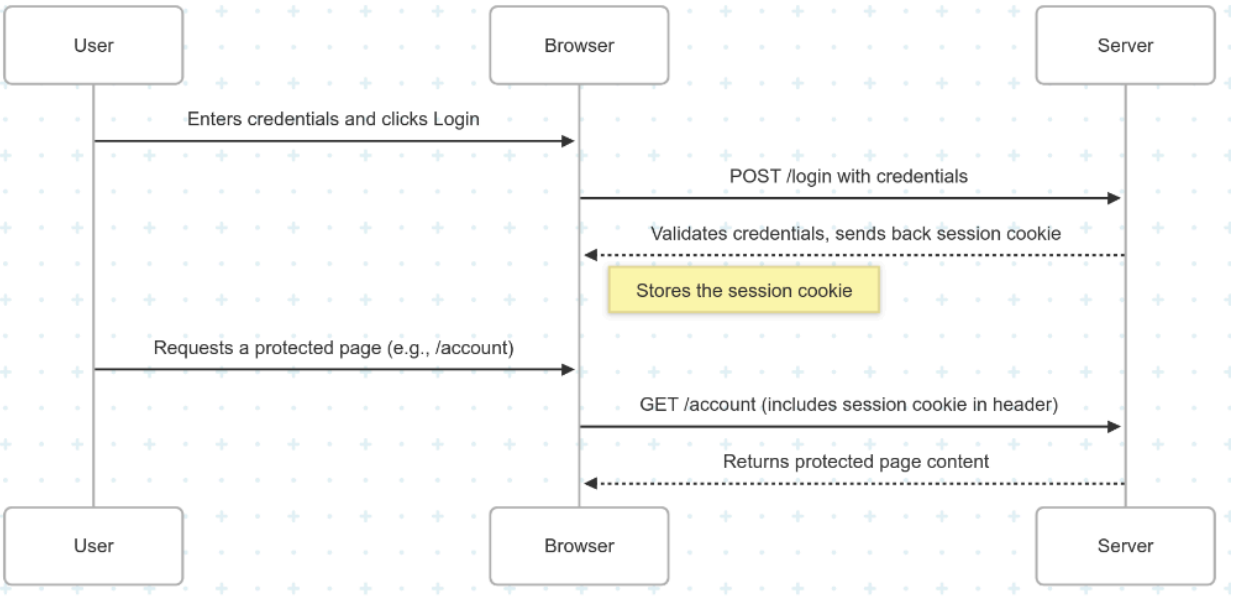
Aim: Attacking session and authentication in Mutillidae using brute force and cookie manipulation.

Learning Objective: To attack session and authentication in Mutillidae using brute force and cookie manipulation.

Theory:

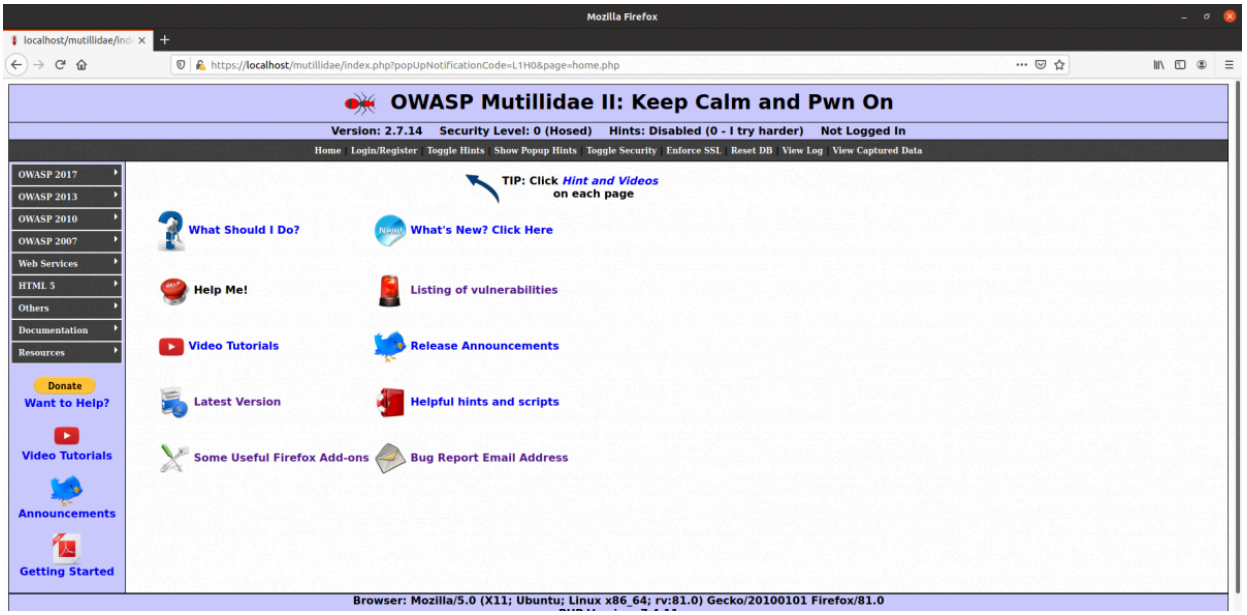
Authentication and **session management** are the cornerstones of web application security. Authentication is the process of verifying a user's identity (e.g., with a username and password), while session management is how the application keeps track of that user across multiple requests after they have logged in. Flaws in either can lead to a complete compromise of user accounts and data.

- **Brute-Force Attacks** This is an attack against authentication where an attacker systematically tries to guess a password. The attack uses a predefined list of common passwords (a "dictionary") or tries all possible character combinations. Applications are vulnerable to this if they don't have proper defenses. Key defenses include:
    - **Rate Limiting:** Limiting the number of login attempts from a single IP address in a given time frame.
    - **Account Lockout:** Temporarily or permanently locking an account after a certain number of failed login attempts.
- **Session Cookies and Manipulation** After a user logs in, the server gives their browser a **session cookie**. This cookie, containing a unique identifier, is sent back to the server with every subsequent request, so the server knows who is making the request. An attack on session management involves an attacker stealing or manipulating this cookie. If a cookie contains predictable information or sensitive data that the client can change (like a privilege level), an attacker can modify it to hijack sessions or escalate their privileges.

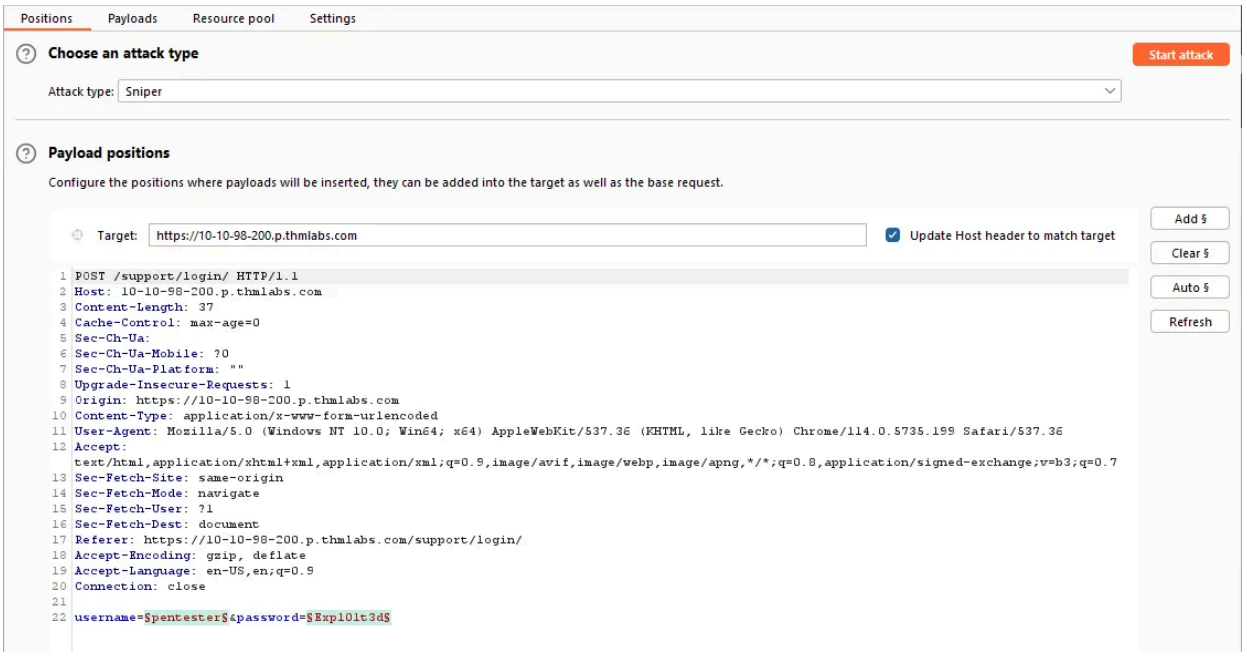


This experiment uses the **OWASP Mutillidae II** application and **Burp Suite**. You must have Mutillidae running (e.g., via OWASP Broken Web Apps or SamuraiWTF) and your browser configured to proxy traffic through Burp Suite.

1. **Set Up the Environment and Target** Start your Mutillidae instance and ensure your browser's traffic is being captured by Burp Suite's Proxy. Navigate to the login page in Mutillidae (index.php?page=login.php).



2. **Perform a Brute-Force Attack with Burp Intruder** In your browser, attempt to log in with any random username and password (e.g., test:test). In Burp Suite, go to the "Proxy" > "HTTP history" tab and find this POST request to login.php. Right-click the request and select "Send to Intruder".
3. **Configure and Run the Attack** Go to the "Intruder" tab. In the "Positions" sub-tab, clear the default payload markers and add one only on the password parameter's value. Switch to the "Payloads" tab, select a "Simple list" payload type, and add a small list of common passwords (e.g., password, 123456, admin, mutillidae). Click "Start attack".



4. **Analyze Brute-Force Results** In the Intruder attack window, look for a response that is different from the others. A successful login will typically have a different "Length" or a 302 redirect "Status" code, indicating success. Note the password that worked.
5. **Analyze and Manipulate Cookies** Log in to Mutillidae with the credentials you found. Navigate around the application and observe the cookies being set in Burp Suite's Proxy history or your browser's developer tools. Look for any interesting cookies. Mutillidae may set a cookie that looks like showhints=1.
6. **Escalate Privileges via Cookie Manipulation** In Burp Suite, go to the "Proxy" > "Intercept" tab and turn Intercept on. Refresh the page in your browser to capture a new request. Find the Cookie: header and modify a value (e.g., if you saw a cookie like isAdmin=0, change it to isAdmin=1). Click "Forward" to send the modified request to the server and observe if your access level or the page content changes.

Here are the example commands and techniques for the procedure.

1. Burp Intruder for Brute-Force

- **Target Request:** POST /mutillidae/index.php?page=login.php
- **Intruder Position:** Set the payload marker around the value of the password parameter. e.g., username=admin&password=§password§&login-php-submit-button=Login
- **Payloads:** Use a simple list of potential passwords.

- **Analysis:** Sort the results by the "Length" column to find the outlier response that indicates a successful login.

2. Burp Proxy for Cookie Manipulation

- **Enable Intercept:** In the "Proxy" > "Intercept" tab, ensure "Intercept is on".
- **Capture Request:** Refresh a page in the application after logging in.
- **Modify Cookie:** In the "Raw" view of the intercepted request, find the Cookie: header. Manually edit the value of a cookie (e.g., from showhints=1 to showhints=0) and forward the request.

Learning Outcome: Upon completing this experiment, you will understand how weaknesses in authentication schemes and session management can be exploited through brute-forcing and client-side cookie manipulation. You will also gain practical experience using security tools like Burp Suite to automate password guessing and modify session data to bypass security controls in a safe, controlled lab environment.

Conclusion:

……

……

……

Name:

Class: BE-CSE

Roll No.:

For Faculty Use

Correction Parameters	Formative Assessment [40%]	Timely completion of Practical [40%]	Attendance / Learning Attitude [20%]	
Marks Obtained				