Secure Programming Lab
Brute-Forcing Authentication/
Privilege Escalation

In today's lab you will learn how to perform a brute-force attack against the vulnerable login form on the Coffeeshop site using Burp Suite. Burp Suite is a powerful and widely used tool for web application security testing. It is used by penetration testers, security researchers, and developers to find vulnerabilities in web applications. Burp Suite provides a variety of tools for tasks like analysing HTTP requests, intercepting and modifying traffic, automating attacks (e.g., brute-forcing), and scanning for security issues, see Fig. 1. There are various tasks and questions along the way. Upload the answers, along with any code files you have changed to Brightspace.

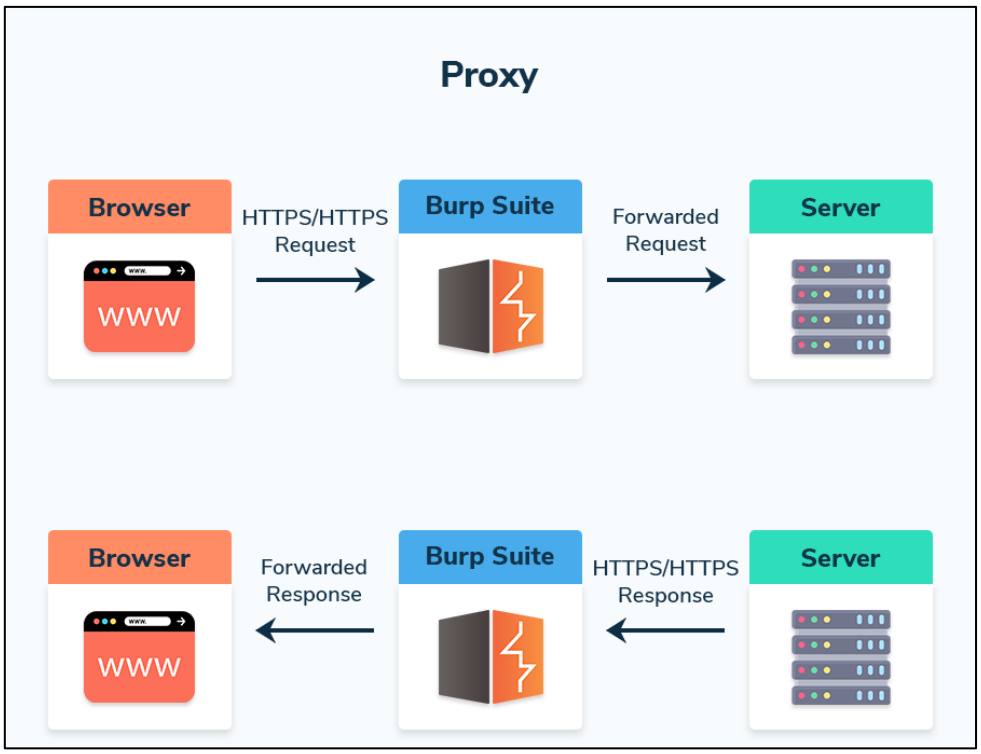

Figure 1: Burpsuite Proxy

Install Burpsuite (Physical Host)

1. Download the installer for your OS: https://portswigger.net/burp/communitydownload
2. Install Burpsuite. Follow the Installation Prompts:
    a. **Windows**: Double-click the .exe file and follow the on-screen prompts.

b. **macOS**: Open the downloaded .dmg file, then drag and drop the Burp Suite icon to your Applications folder.
c. **Linux**: Extract the .tar file and follow the instructions in the readme.txt or run Burp Suite using the terminal.

3. Launch Burpsuite. Keep all the defaults and click **Start Burp**.
4. Once Burp starts, you'll see the main interface with several tabs (Proxy, Intruder, Repeater, etc.), see Fig. 2. These are the various tools within Burp Suite.

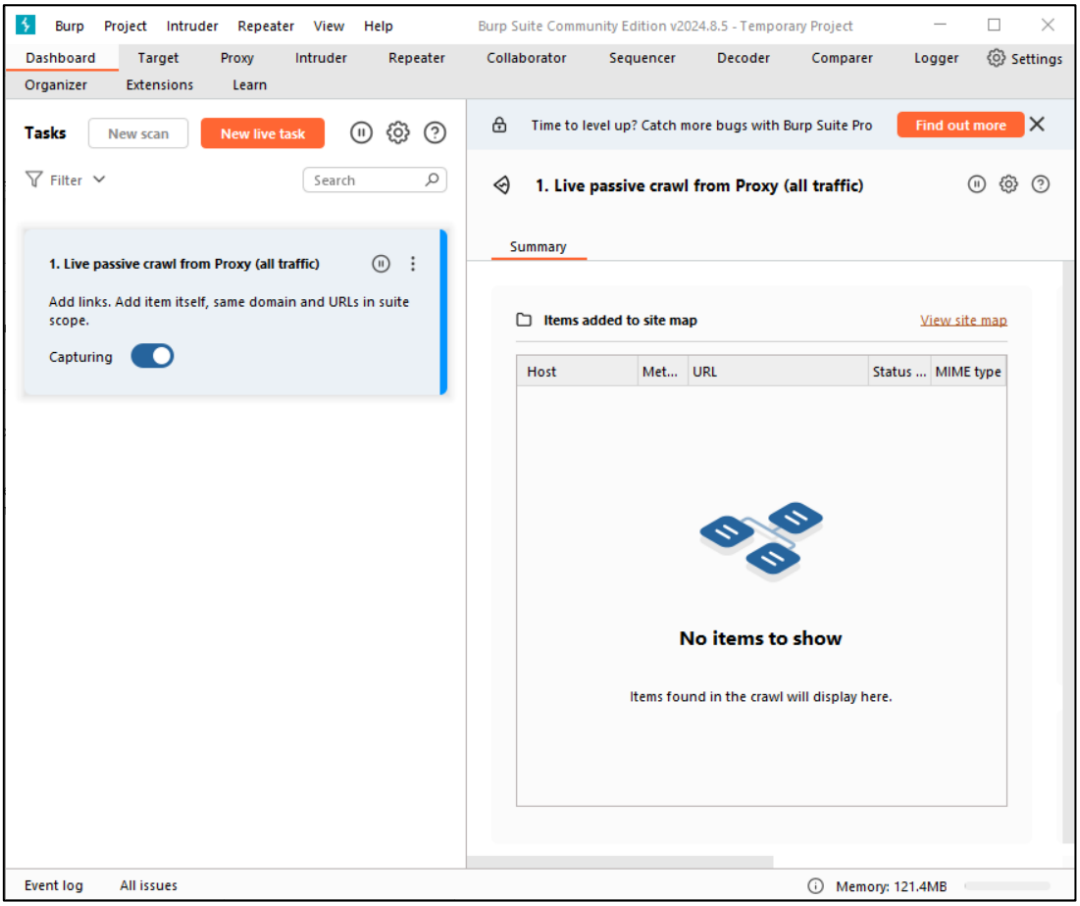

Figure 2: Burpsuite Dashboard GUI

Intercepting requests to the Coffeeshop site

We're going to use a tool in Burp called Proxy. This acts as the "man-in-the-middle" between the browser and the application. Once it is set, the Proxy interceptor will capture any traffic going from the browser to the application and back.

1. Click on the Proxy tab. Click on the **Open Browser** button. This will open up a new browser window. This browser is already configured to run through the proxy tool.
2. Make sure that **Intercept** is set to **off** (it should be by default).
3. Navigate to http://localhost:8080/account/login/
4. Type in the login credentials for Bob, but **DON'T** click submit.

5. Go back to the Burp proxy window and set **Intercept** to **on**.
6. Now click submit on the Coffeeshop login page.

You're Burpsuite/ Web Browser windows should look something like Figure 3:

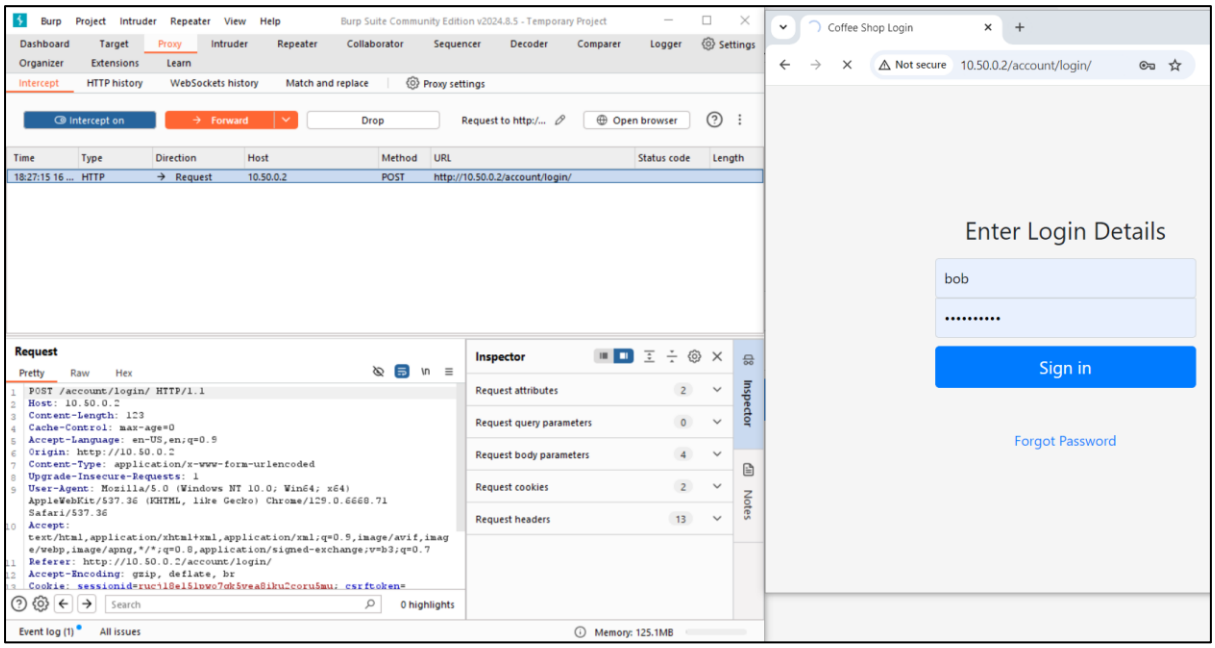

Figure 3: Bob's login credentials intercepted

Task 1

Identify the parameters in the request where the username and password are sent. What HTTP method is used (GET or POST)? Why is this important in the context of authentication security?

Brute-forcing the authentication page

1. In the Proxy tab, click **Forward** to send the request through. Turn **Interceptor off.**
2. In the **HTTP History** tab, locate the login request and right-click it, then select **Send to Intruder**.
3. Click on the Intruder tab. You should see something like Figure 4.
4. In the **Positions** sub-tab, click **Clear §** to clear all current payload markers (the positions Burp will attempt to modify).
5. Highlight the **username** and **password** parameter values in the intercepted request and click **Add §**. These are the fields Burp will attack.

Question 1

Why is it essential to correctly identify the parameters you want to brute-force?

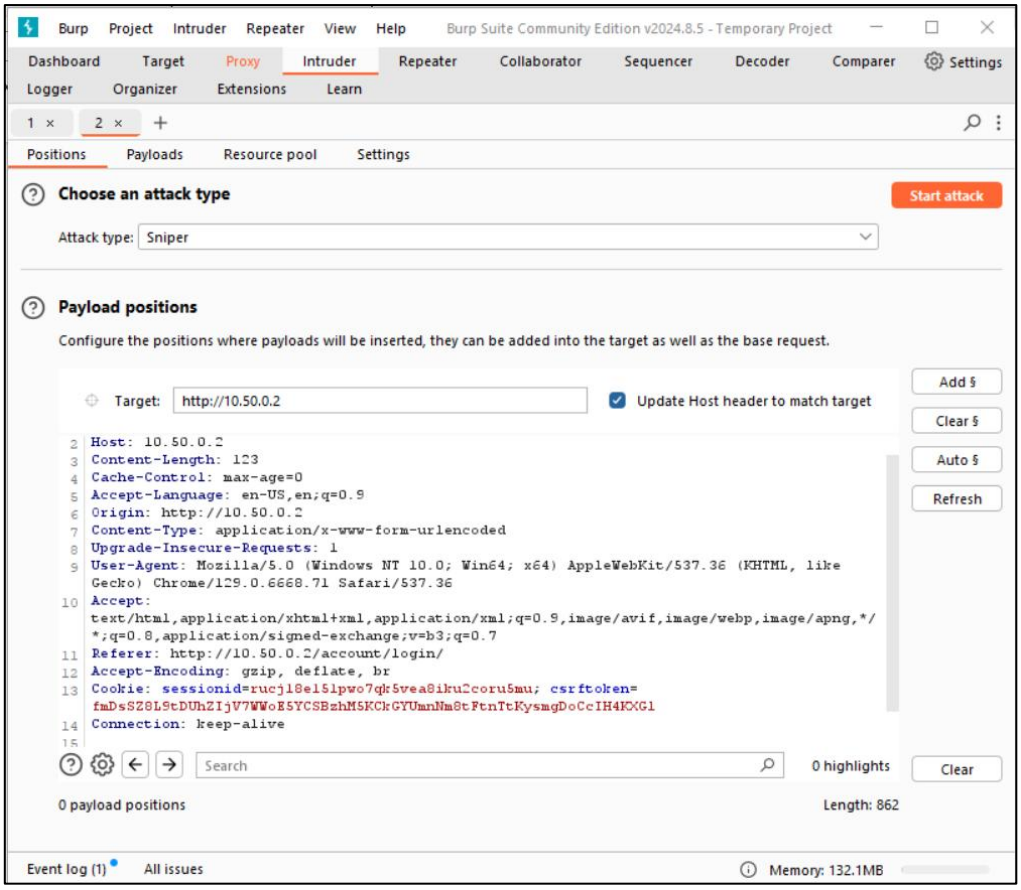

Figure 4: Burpsuite Intruder (Coffeeshop authentication page)

6. Set the **Attack Type** to **Cluster Bomb**. This allows an attack using multiple payload lists.
7. Click on the Payloads sub-tab. Select **Payload Set 1** from the payload set drop-down list (should be default). Leave **Payload Type** as **Simple List**.
8. Under **Payload Settings**, either paste the list of usernames from **username.txt** or load it using the **Load..** tab.
9. Repeat for **Payload Set 2**, this time adding the list of passwords from **passwords.txt**
10. Click on **Start Attack**. Note: you will get a warning about the responses being throttled – this is because we are using the Community Edition of Burpsuite so attacks will be slower.
11. When the attack is finished, study the responses to look for any behavior that may indicate a valid login. Review the length, status codes, and response times for each login attempt. Look for anomalies that may indicate a successful login.

Question 2

How did you identify the successful login attempt(s) from the Intruder results?

12. Right-click on the successful login attempt and choose **Show Response in Browser**. You will have to copy the URL into the Burpsuite browser. Click on **Show Response** if not redirected automatically.

Question 3

What page are you on the Coffeeshop website are you on now?

Mitigating Brute Force Authentication Attacks

1. **Implement Multi-Factor Authentication (MFA):**
    - Require additional factors such as one-time passwords (OTP), biometrics, or security tokens along with the user's password.
2. **Account Lockout Mechanisms:**
    - Lock accounts temporarily after a predefined number of failed login attempts (e.g., after 3-5 failed attempts). Implement mechanisms to avoid account lockout DoS attacks, such as using CAPTCHAs after failed logins.
3. **Rate Limiting for Login Attempts:**
    - Implement rate limiting by slowing down the response time after multiple failed attempts or restricting the number of login attempts allowed within a short time window.
4. **Use Strong Password Policies:**
    - Enforce strong password requirements (e.g., length, complexity) and encourage the use of password managers for users to create and store secure passwords.
5. **CAPTCHA on Login:**
    - Implement CAPTCHA mechanisms (e.g., reCAPTCHA) after a certain number of failed login attempts to ensure the user is human and prevent automated brute-force attempts.
6. **IP-based Restrictions or Blocklisting:**
    - Block IPs that exhibit suspicious activity, such as too many failed login attempts in a short period. This can be combined with geolocation-based access control.
7. **Session Timeouts and Inactivity Detection:**
    - Automatically log users out after a period of inactivity to minimize the risk if an attacker gains access to a session.