Lab 3: Authentication Vulnerabilities (password-based)

Username enumeration via different responses

Part 1: Description and Objective

Objective

The primary objective of this lab is to examine common vulnerabilities that occur in password-based login mechanisms. Specifically, this section focuses on exploiting these mechanisms to gain unauthorized access.

Lab Challenge: Username enumeration via different responses

- **Description:** According to the lab details, the target system contains a vulnerability allowing for username enumeration and password brute-force attacks. The system possesses an account with a predictable username and password, which can be found in the following provided wordlists:
    - Candidate usernames
    - Candidate passwords
- **Requirements:** To successfully complete this lab, the following steps must be performed:
    1. Enumerate a valid username using the different responses from the server.
    2. Brute-force the identified user's password.
    3. Access the user account page.

Part 2: Solution Overview

Based on the provided solution guide, the attack vector involves using **Burp Suite Intruder** to automate the enumeration process. The strategy is divided into two main phases:

Phase 1: Username Enumeration

- Intercept the POST /login request and send it to **Burp Intruder**.

- Set the attack type to **Sniper** and configure the payload position on the username parameter.
- Use the "Candidate usernames" list as the payload.
- **Analysis:** Analyze the **Length** column or response body. A valid username will trigger a different response (e.g., a specific error message like "Incorrect password" instead of "Invalid username") compared to invalid usernames.

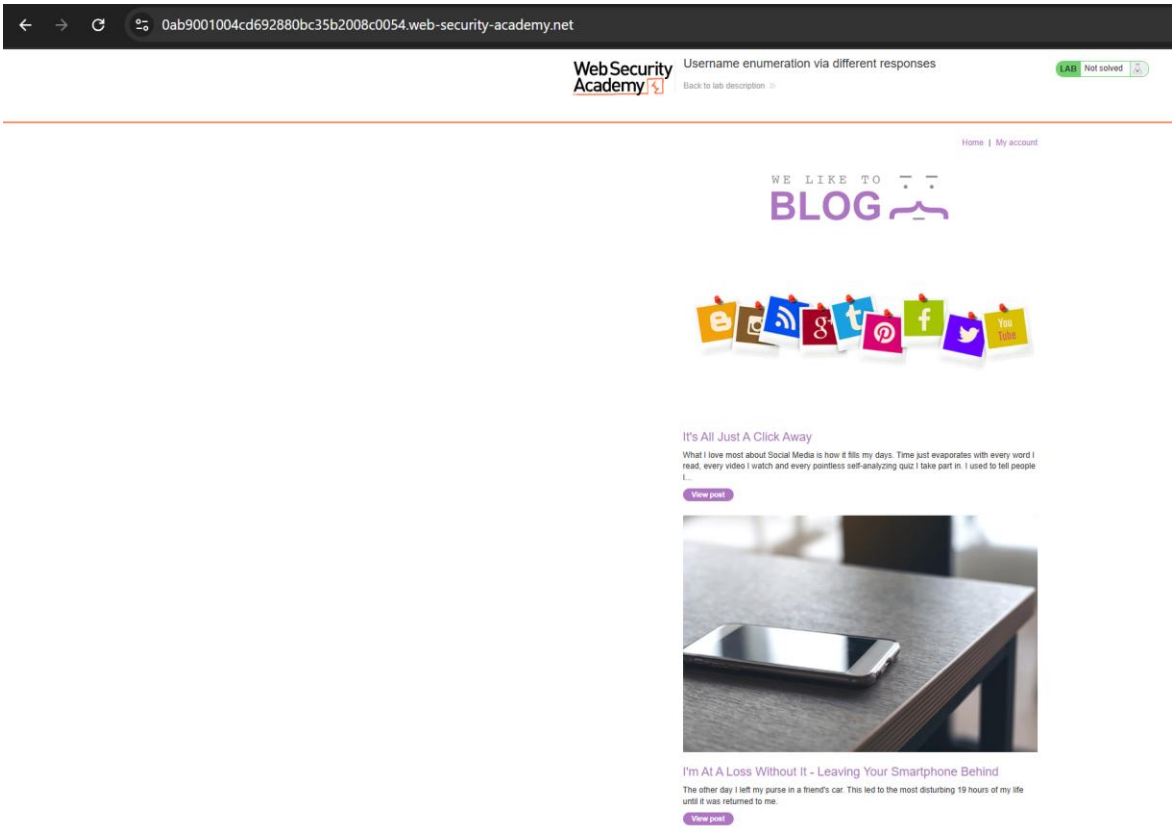
Phase 2: Password Brute-forcing

- Once the valid username is identified, update the payload position to the password parameter.
- Use the "Candidate passwords" list as the payload.
- **Analysis:** Look for a **302** status code (redirection), which indicates a successful login, whereas failed attempts usually return a **200** status code.

Part 3: Step-by-Step Implementation

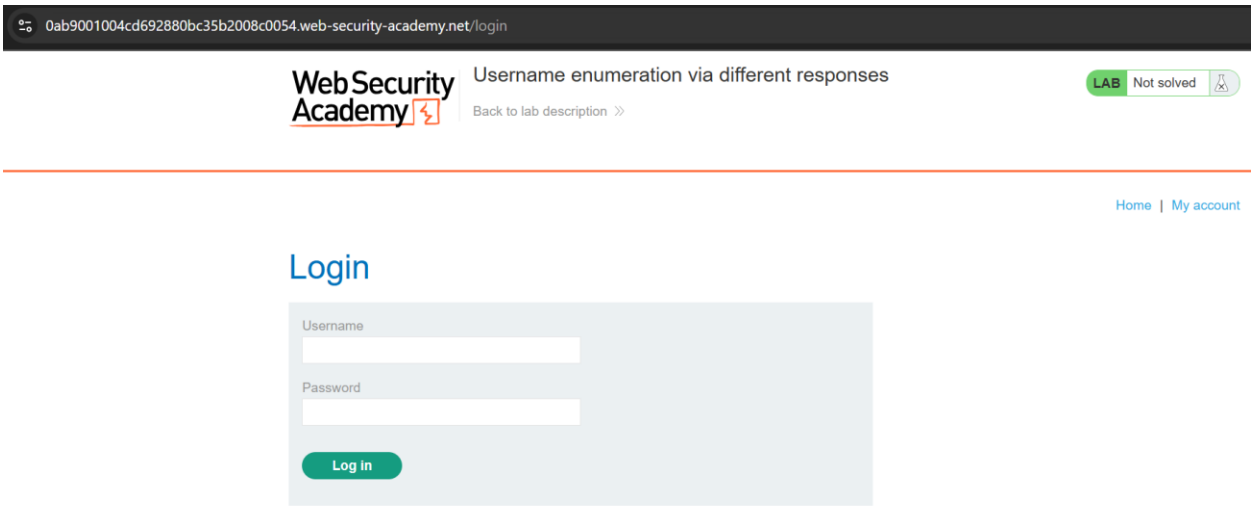
Step 1: Access the Lab

- **Action:** Navigate to the PortSwigger Web Security Academy page for the lab "Username enumeration via different responses".
- **Execution:** Click the orange button labeled **"ACCESS THE LAB"** to initiate the virtual lab environment.



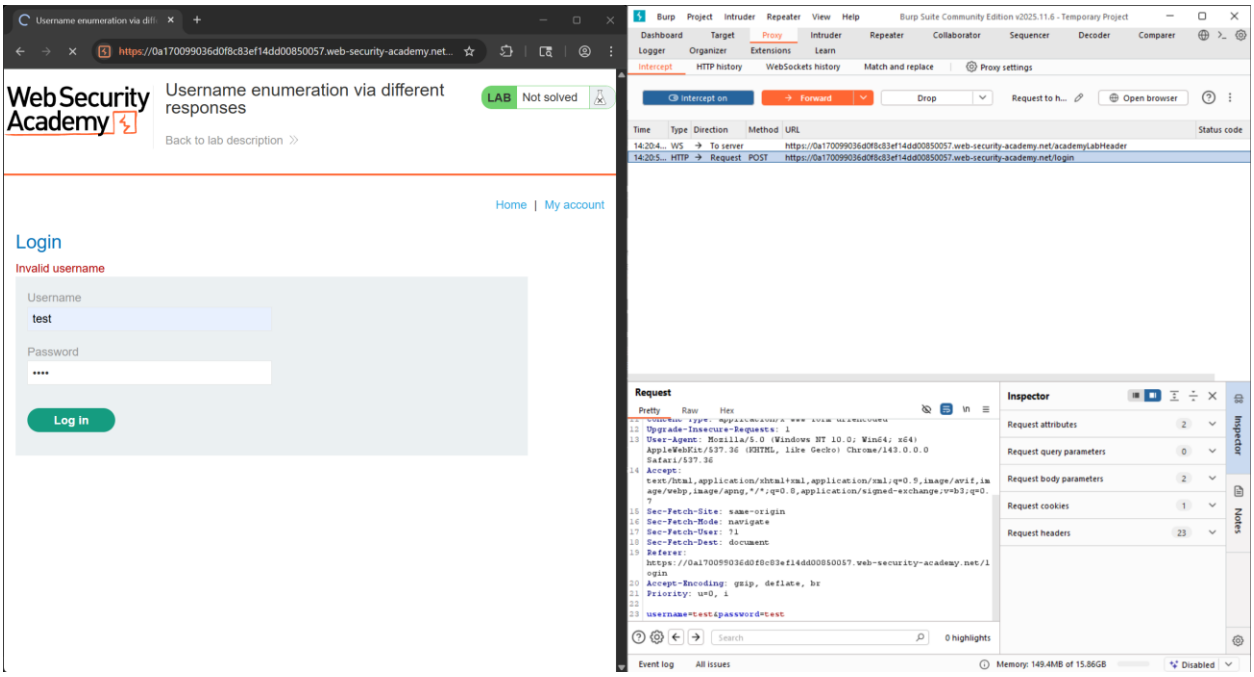
Step 2: Navigate to the Login Page

- **Action:** From the lab homepage, locate the navigation menu.
- **Observation:** The lab environment loads the "We Like to Blog" homepage, and the **"My account"** link is visible in the top-right corner of the interface.
- **Execution:** Click on **"My account"** to proceed to the login screen.



Step 3: Intercept the Login Request

- **Action:** Capture the HTTP request generated when attempting to log in.
- Execution:
    1. Ensure **Burp Suite** is running and the browser traffic is being proxied through it.
    2. On the login page (from Step 2), enter an **invalid** username and password (e.g., user / 123) and click the **"Log in"** button.
    3. Open Burp Suite and navigate to the **Proxy > HTTP history** tab.
    4. Locate the POST /login request in the history list. Highlight this request to view its details.



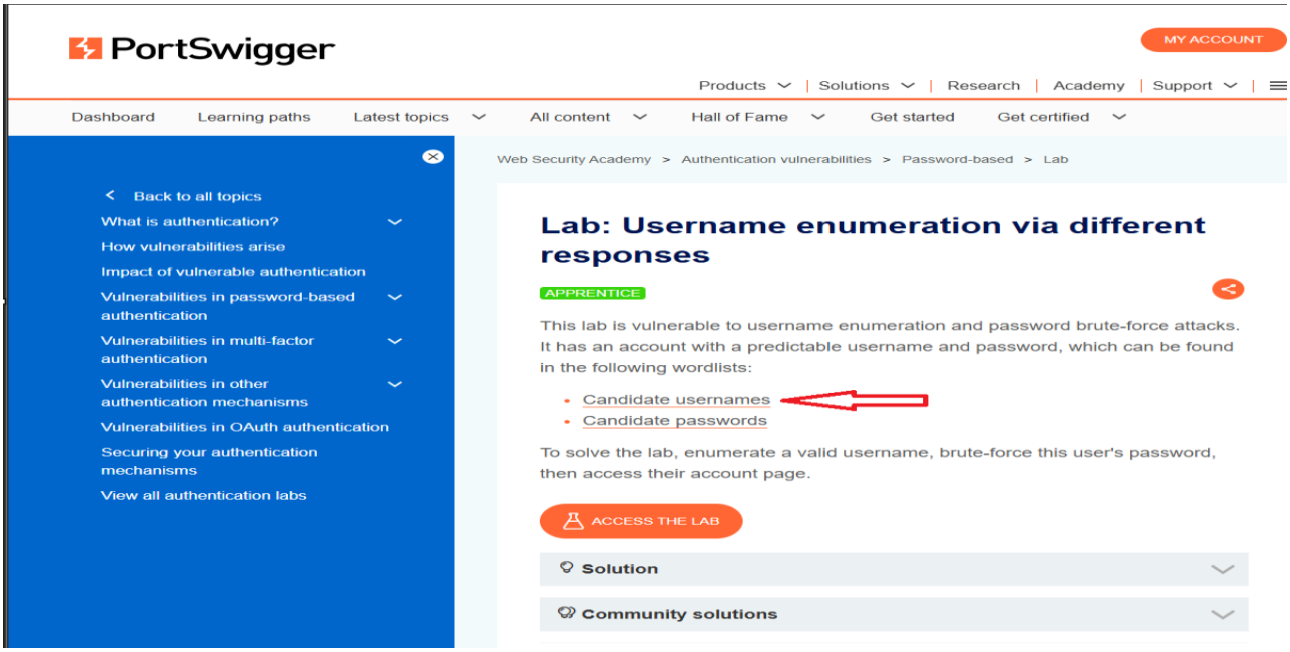
Step 4: Configure the Username Enumeration Attack

- **Action:** Send the intercepted request to Burp Intruder to set up the payload positions.
- Execution:
    1. In the intercept window (from Step 3), right-click on the request data and select **"Send to Intruder"** (or press Ctrl + I).
    2. Navigate to the **Intruder** tab, then the **Positions** sub-tab.
    3. Ensure the **Attack type** is set to **"Sniper"**.
    4. Click the **"Clear §"** button to remove the default markers.

5. Highlight the value of the username parameter (e.g., test) and click the **"Add §"** button to define it as the payload position.
6. **Verification:** The parameter should look like: username=§test§&password=test.

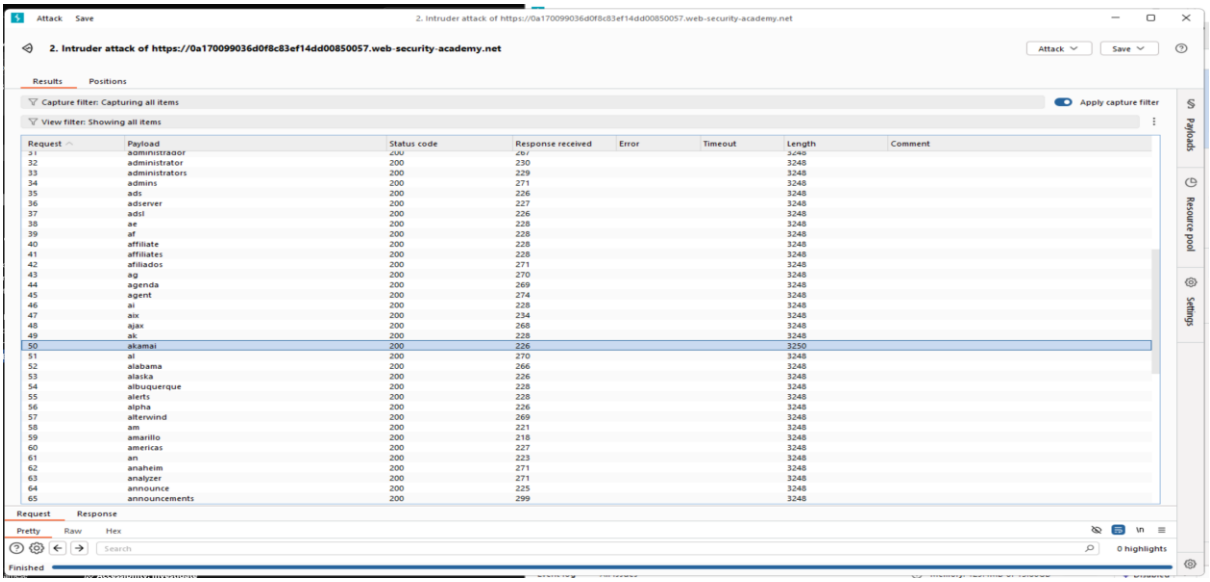
Step 5: Launch the Username Enumeration Attack

- **Action:** Load the list of candidate usernames into the payload configuration and execute the attack.
- Execution:
    1. In the **Intruder** tab, ensure the **Payloads** side panel (or tab) is open.
    2. Check that **Payload type** is set to **"Simple list"**.
    3. Copy the list of **Candidate Usernames** from the PortSwigger lab description page.



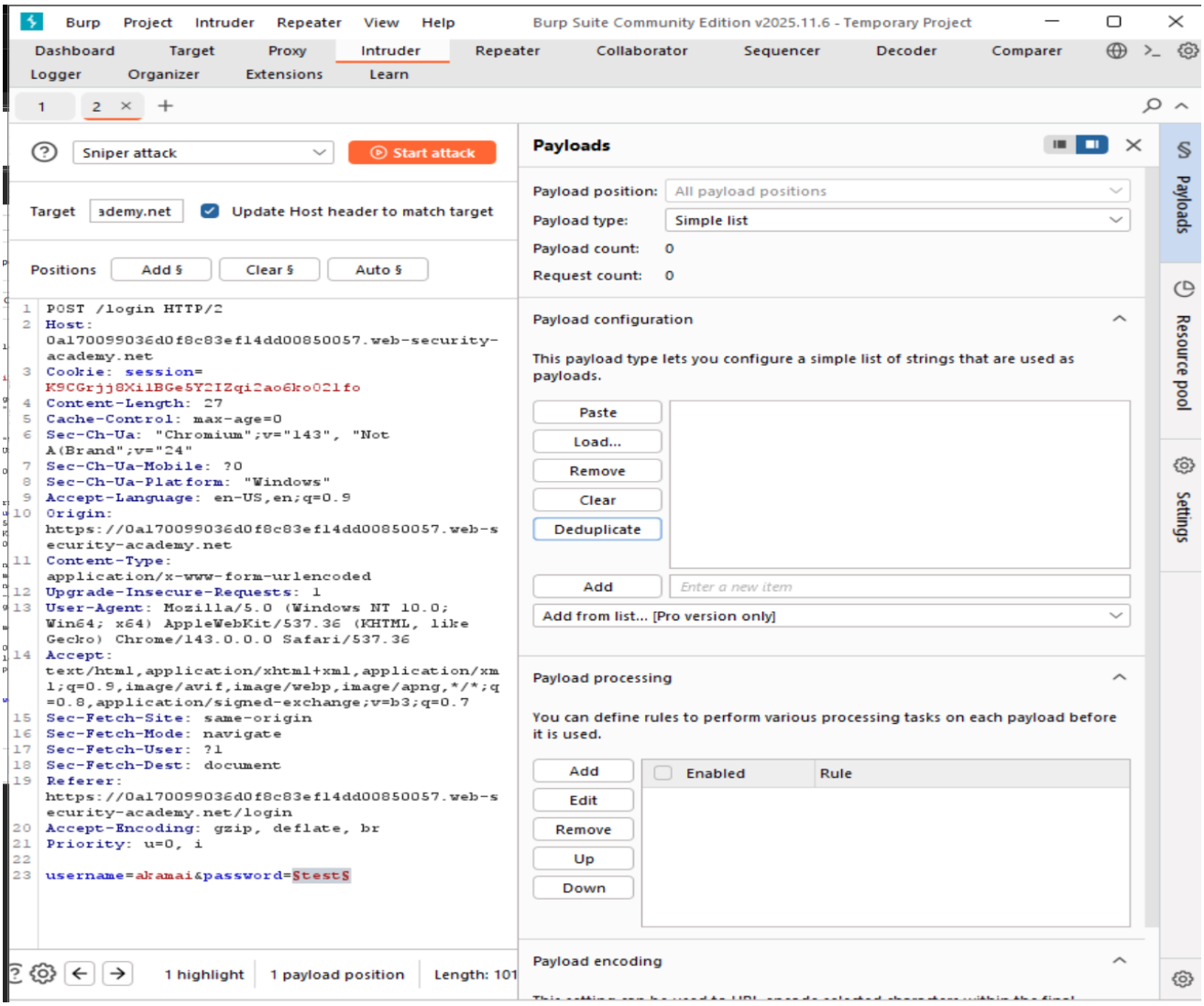
4. In the **Payload configuration** section, click **"Paste"** to load the usernames into the list.
5. Click the **"Start attack"** button (orange button in the top bar).
6. **Analysis:** The attack window will open. Wait for it to finish, then click on the **"Length"** column header to sort the results.
7. **Observation:** Identify the single request that has a different length compared to the others. This specific request corresponds to the valid username.

- **Conclusion:** This discrepancy indicates that akamai is the valid username. The slight difference in length typically corresponds to a different error message (e.g., "Incorrect password" instead of "Invalid username").



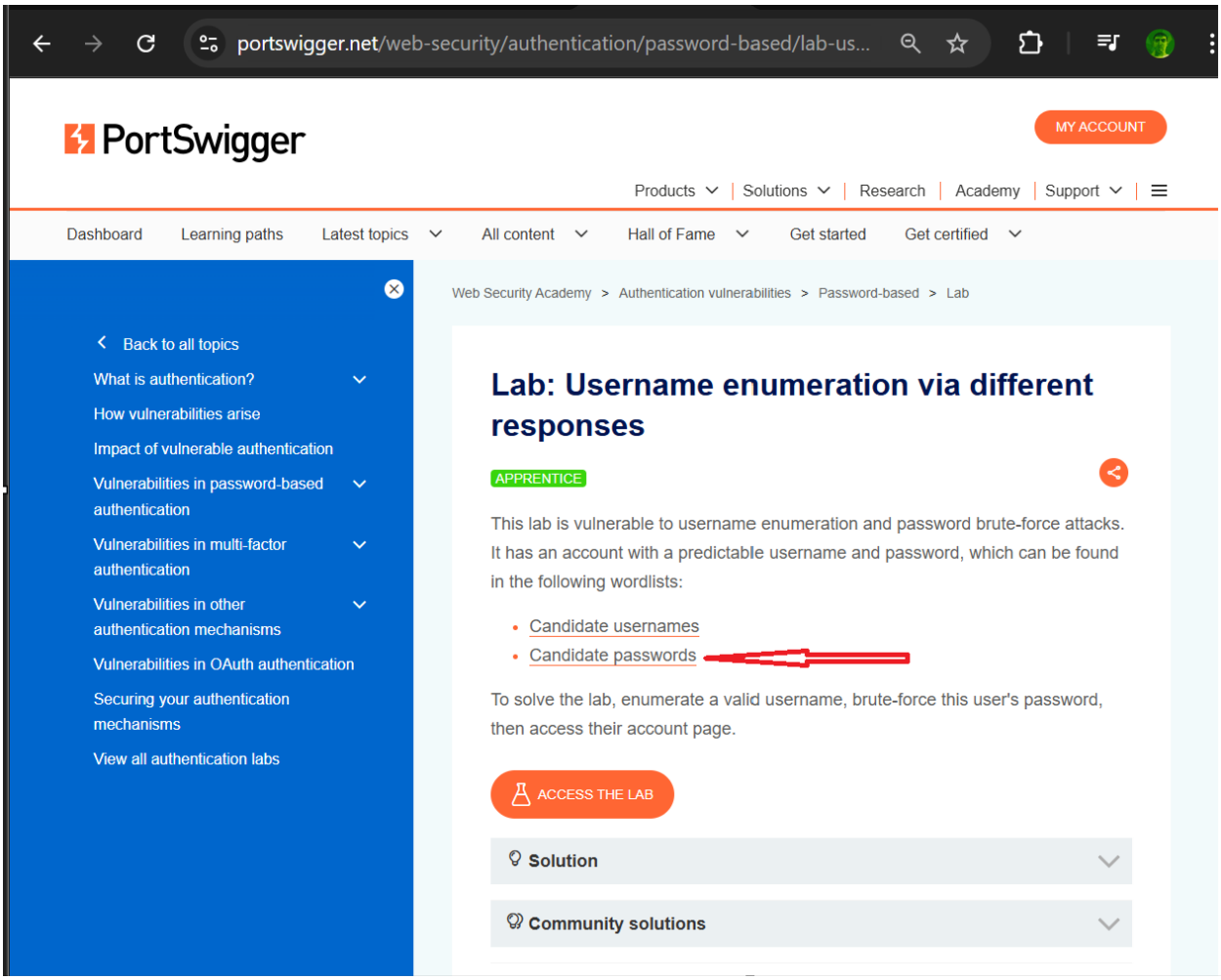
Step 6: Configure the Password Brute-Force Attack

- **Action:** Now that the valid username (akamai) is known, configure Burp Intruder to brute-force the password for this specific account.
- Execution:
    1. Go back to the **Intruder > Positions** tab.
    2. Click **"Clear §"** to remove the previous markers.
    3. Manually change the username parameter value to the valid username found: username=akamai.
    4. Highlight the value of the password parameter (currently test or similar) and click **"Add §"** to set the new payload position.
    5. **Verification:** The parameters should look like: username=akamai&password=§test§
    6. Ensure the **Attack type** is still set to **"Sniper"**.



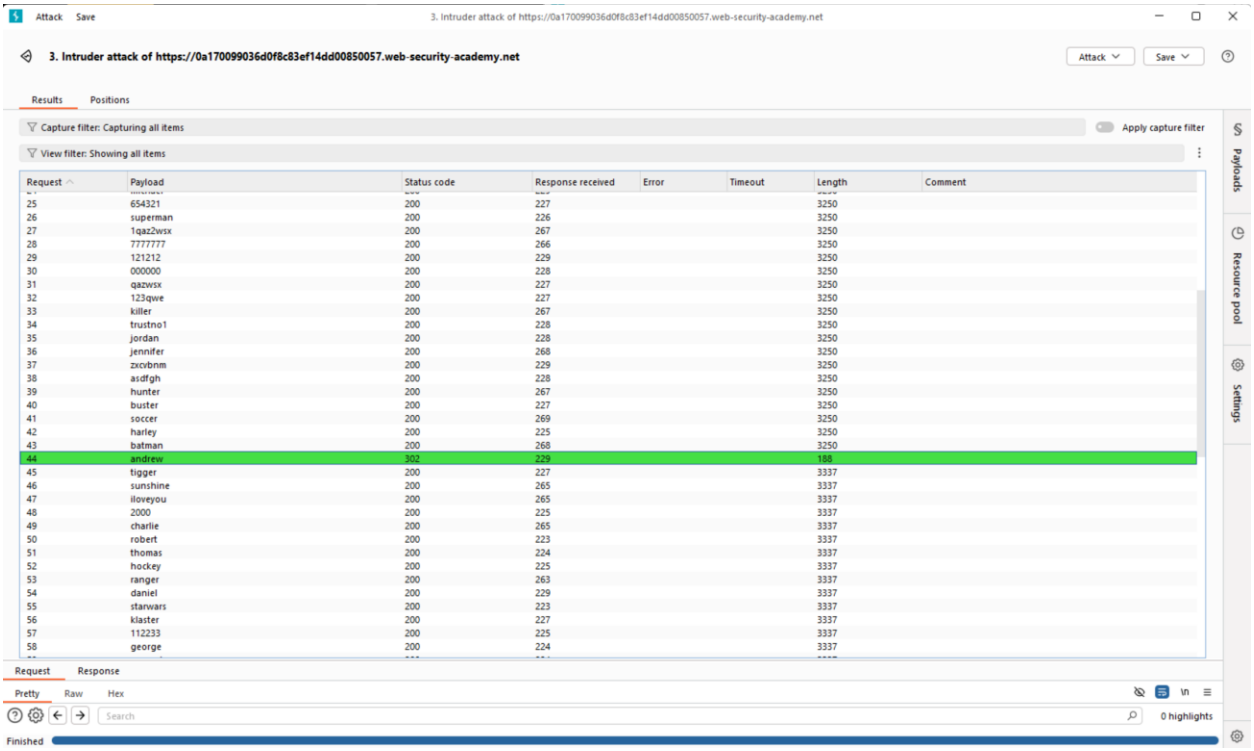
Step 7: Launch the Password Attack

- **Action:** Load the list of candidate passwords and execute the brute-force attack to find the correct credential.
- Execution:
    1. Navigate to the **Payloads** side panel (or tab).
    2. Ensure **Payload type** is set to **"Simple list"**.
    3. **Clear** any existing payloads (from the username list used previously).
    4. Copy the list of **Candidate Passwords** from the lab description page.



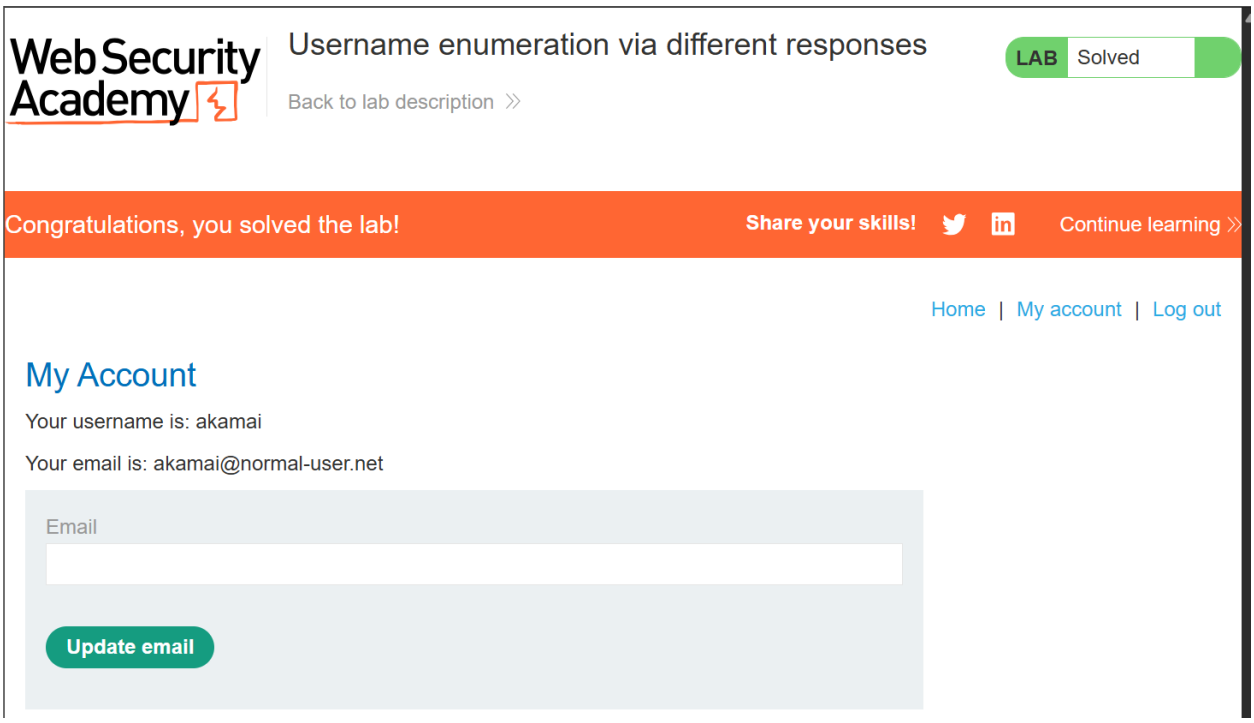
5. Paste the passwords into the **Payload configuration** list.
6. Click **"Start attack"**.
7. **Analysis:** Wait for the attack to complete. Sort the results by the **"Status"** code column.
8. **Observation:** Look for a request that returns a **302** status code. A 302 code indicates a successful redirection (login), whereas failed attempts return 200.

- **Conclusion:** A 302 status code typically indicates a successful authentication followed by a redirection to the user's dashboard. Therefore, the valid password for the user akamai is identified as andrew.



Step 8: Final Verification (Login)

- **Action:** Verify the discovered credentials by manually logging into the application to solve the lab.
- Execution:
    1. Return to the browser and navigate to the login page.
    2. Enter the identified username: **akamai**.
    3. Enter the identified password: **andrew**.
    4. Click the **"Log in"** button.
    5. **Result:** The application should log you in successfully and redirect you to the "My Account" page. The lab status should update to "Solved".



Username enumeration via response timing

Part 1: Description and Objective

Objective

The goal of this challenge is to enumerate a valid username based on the time the server takes to respond, rather than the content of the response message. Following that, the password must be brute-forced to access the account.

Vulnerability

The application is vulnerable because it performs time-consuming tasks (like password hashing) only when a valid username is provided. This creates a timing discrepancy (side-channel) that can be measured.

Security Mechanism

The lab implements IP-based brute-force protection. To bypass this, we must spoof our IP address using the X-Forwarded-For header for each request.

Part 2: Solution Overview

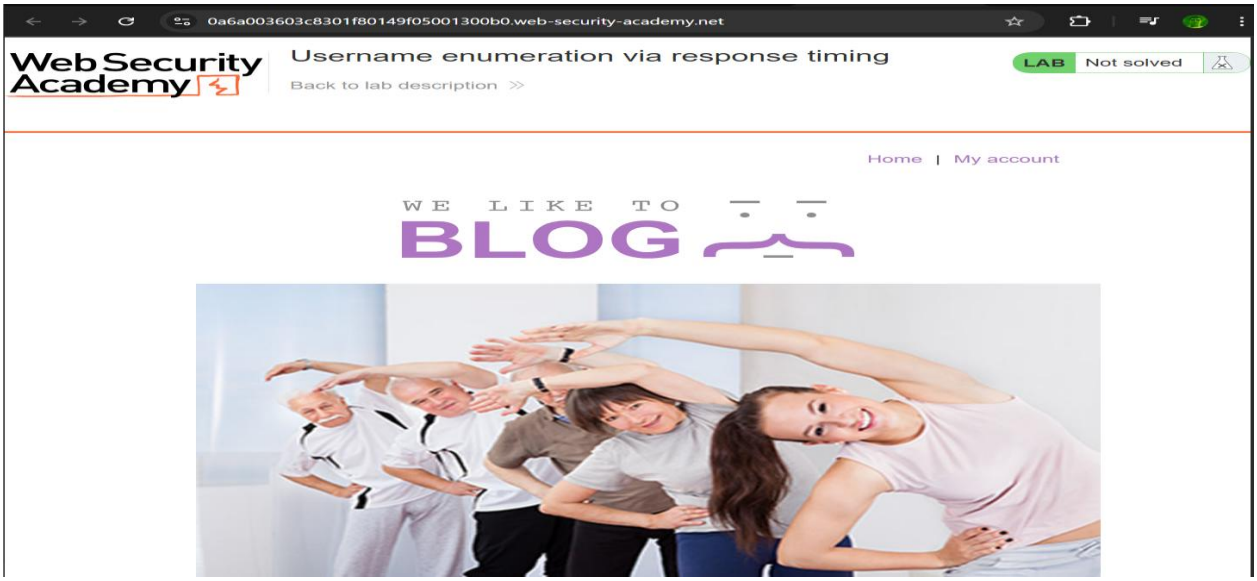
Based on the provided solution analysis, the exploitation process involves the following key phases:

1. **Bypass IP Protection:** Modify the request to include the X-Forwarded-For header. This header allows the simulation of requests coming from different IP addresses.
2. Username Enumeration (Timing Attack):
    a. Send a POST /login request with a **very long password** (e.g., 100 characters) to maximize the processing time for valid users (due to hashing).
    b. Use **Pitchfork** attack type:
        i. **Payload 1 (IP):** Iterates through numbers to fake different IPs.
        ii. **Payload 2 (Username):** Iterates through the candidate usernames.
    c. **Analysis:** Monitor the "Response received" or "Response completed" columns. The valid username will result in a significantly longer response time.
3. **Password Brute-force:** Once the username is known, perform a second Pitchfork attack using the found username and the list of candidate passwords, looking for a 302 status code.

Part 3: Step-by-Step Implementation

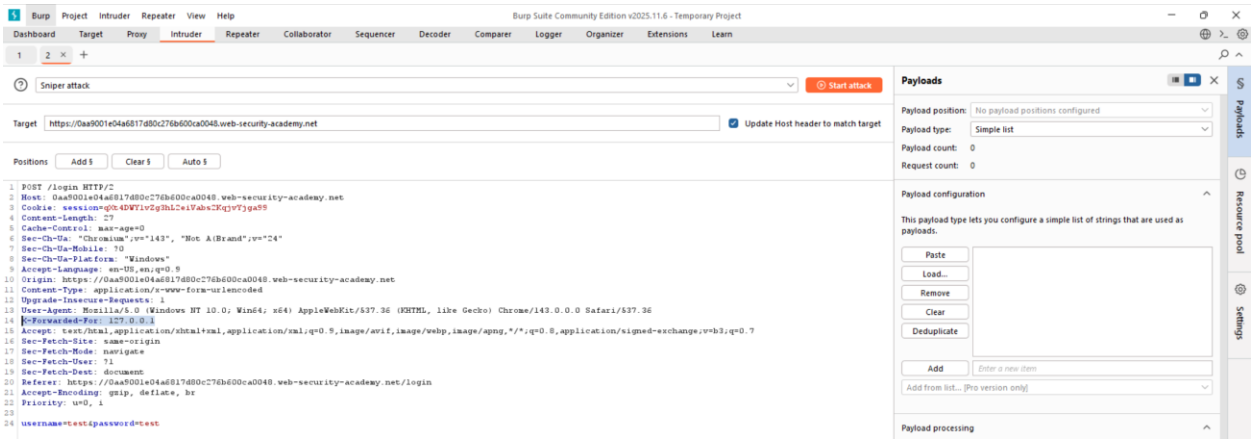
Step 1: Access the Lab

- **Action:** Navigate to the lab "Username enumeration via response timing" on PortSwigger Academy.
- **Execution:** Click the **"ACCESS THE LAB"** button to start the instance.



Step 2: Intercept and Modify the Request

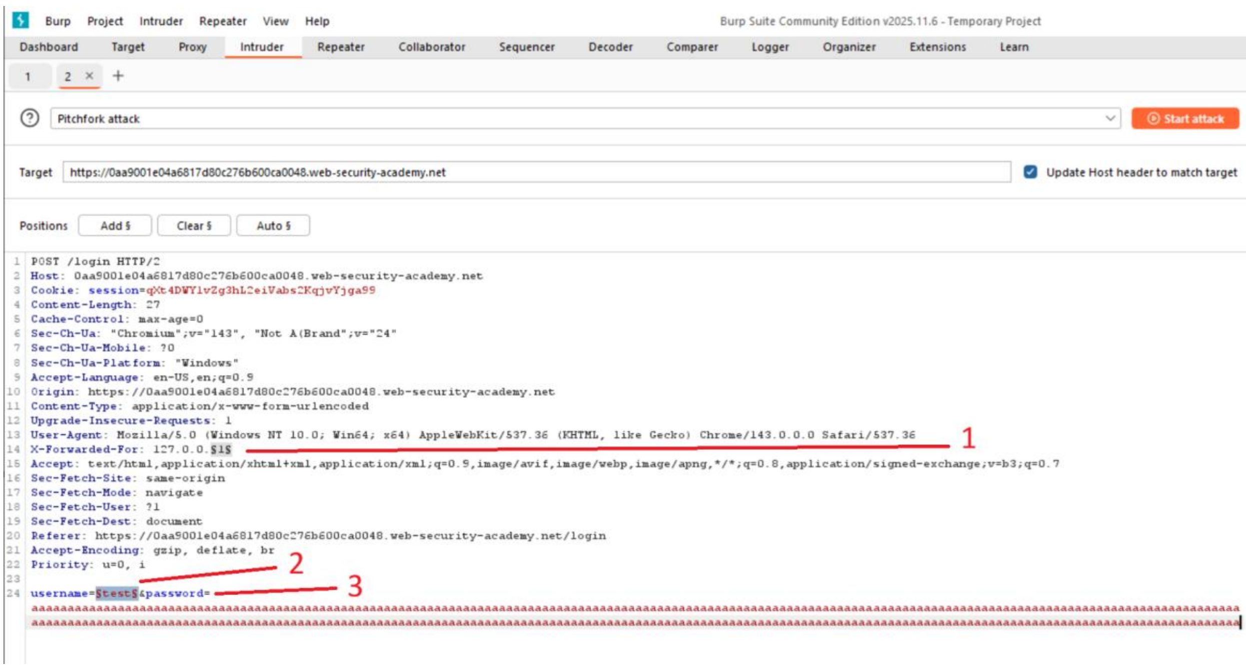
- **Action:** Capture a login request and manually add the IP spoofing header.
- Execution:
    1. Navigate to the "My account" page.
    2. Enter an invalid username and password (e.g., test / 123) and click **"Log in"**.
    3. In **Burp Suite Proxy**, intercept the POST /login request.
    4. **Modification:** Manually add the following header to the request (under the User-Agent or Cookie line): X-Forwarded-For: 127.0.0.1
    5. Right-click the modified request and select **"Send to Intruder"** to prepare for the attack configuration.



Step 3: Configure the Pitchfork Attack

- **Action:** Configure Burp Intruder to use the **Pitchfork** attack type. This allows the attack to iterate through a list of IP addresses and a list of usernames simultaneously. Additionally, set a long password to induce a time delay for valid usernames.
- Execution:
    1. **Select Attack Type:** In the **Intruder > Positions** tab, change the **Attack type** dropdown menu from "Sniper" to **"Pitchfork"**.
    2. Set Payload Markers:
        - **Payload 1 (IP Spoofing):** Highlight the last number of the IP address in the X-Forwarded-For header (e.g., the 1 in 127.0.0.1) and click **"Add §"**.
            - *Result:* X-Forwarded-For: 127.0.0.§1§
        - **Payload 2 (Username):** Highlight the value of the username parameter (e.g., test) and click **"Add §"**.

    - *Result:* username=§test§
3. **Set Static Long Password:** Manually replace the password parameter value with a very long string (at least 100 characters) to maximize the hashing delay.
    - Result: password=aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa... (ensure no § markers are on the password).



Step 4: Configure Attack Payloads

Action: Define the payload sources for both the IP spoofing (numbers) and the username enumeration (wordlist).

Execution:

1. Navigate to the **Intruder > Payloads** tab.

2. Configure Payload Set 1 (IP Spoofing):

    Select **Payload set: 1** from the dropdown menu.

    Change **Payload type** to **"Numbers"**.

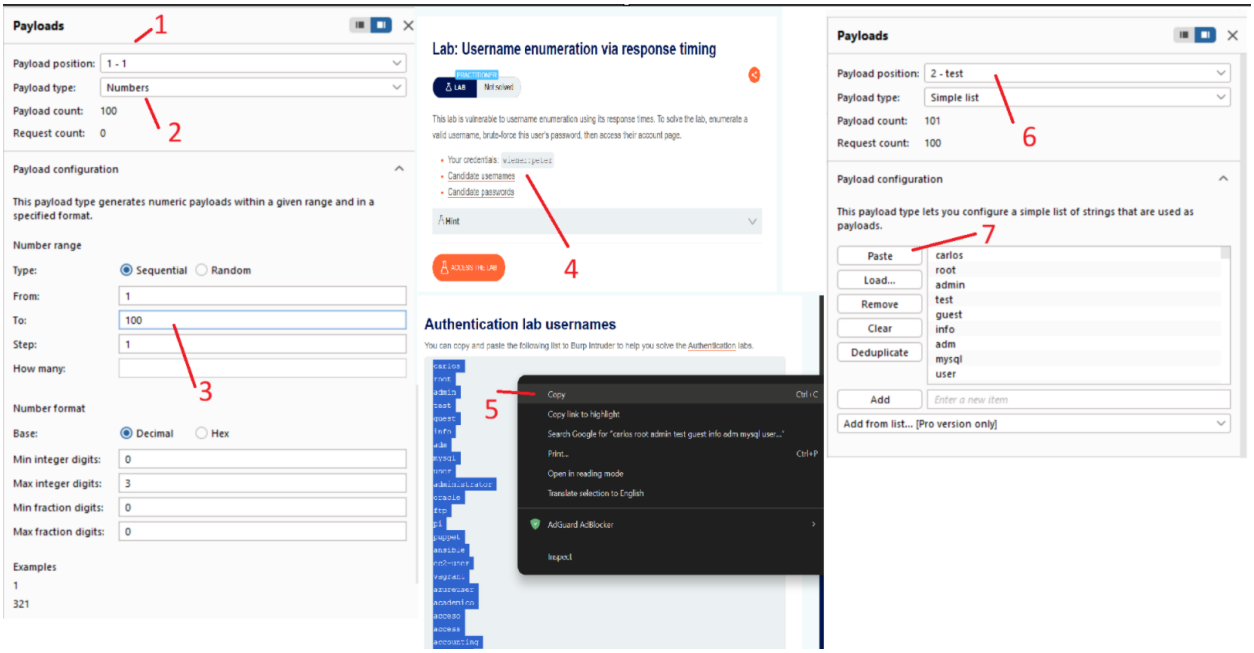
Set **From: 1**, **To: 100**, and **Step: 1**. This will generate numbers from 1 to 100 to create unique fake IPs (e.g., 127.0.0.1, 127.0.0.2, etc.).

3. Configure Payload Set 2 (Usernames):

Select **Payload set: 2** from the dropdown menu.

Ensure **Payload type** is set to **"Simple list"**.

Paste the list of **Candidate Usernames** provided in the lab description.



Step 5: Launch the Attack and Analyze Response Timing

- **Action:** Execute the Pitchfork attack and identify the valid username by analyzing the server's response time.
- Execution:

1. Click the **"Start attack"** button in Burp Intruder.

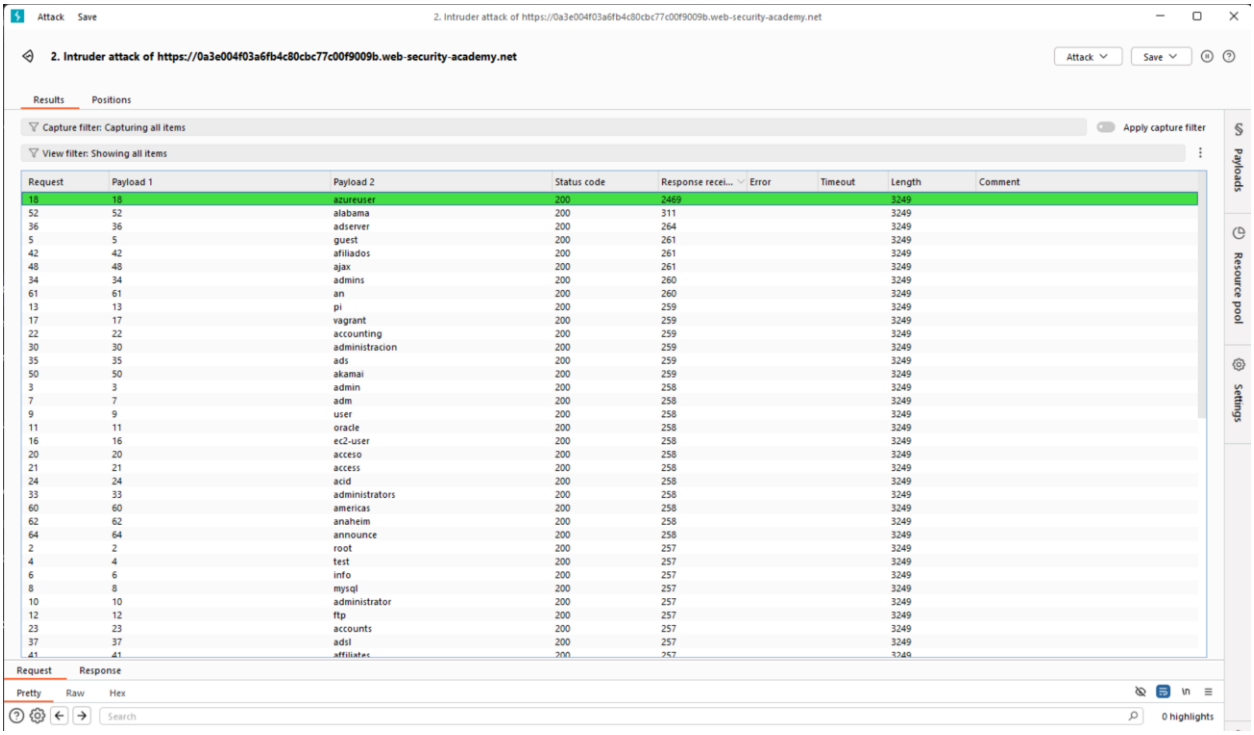
2. Enable Timing Columns: In the attack results window, go to the **"Columns"** menu (or right-click the header row) and ensure **"Response received"** and **"Response completed"** are checked.

3. Analyze Results: Wait for the attack to finish. Click on the **"Response received"** column header to sort the results by time.

4. Observation: Identify the username associated with the request that took the longest time to process.

Expected Behavior: Invalid usernames are rejected quickly. The valid username forces the server to hash the long password, causing a noticeable delay (e.g., significantly higher milliseconds compared to others).

5. **Result:** The request for the username **azureuser** took **2,469 milliseconds** to process, while all other requests completed in approximately **260 milliseconds**.



Step 6: Configure the Password Brute-Force Attack

Action: Now that the username is identified, configure a second Pitchfork attack to brute-force the password while continuing to spoof the IP address.

Execution:

1. Return to the **Intruder > Positions** tab.

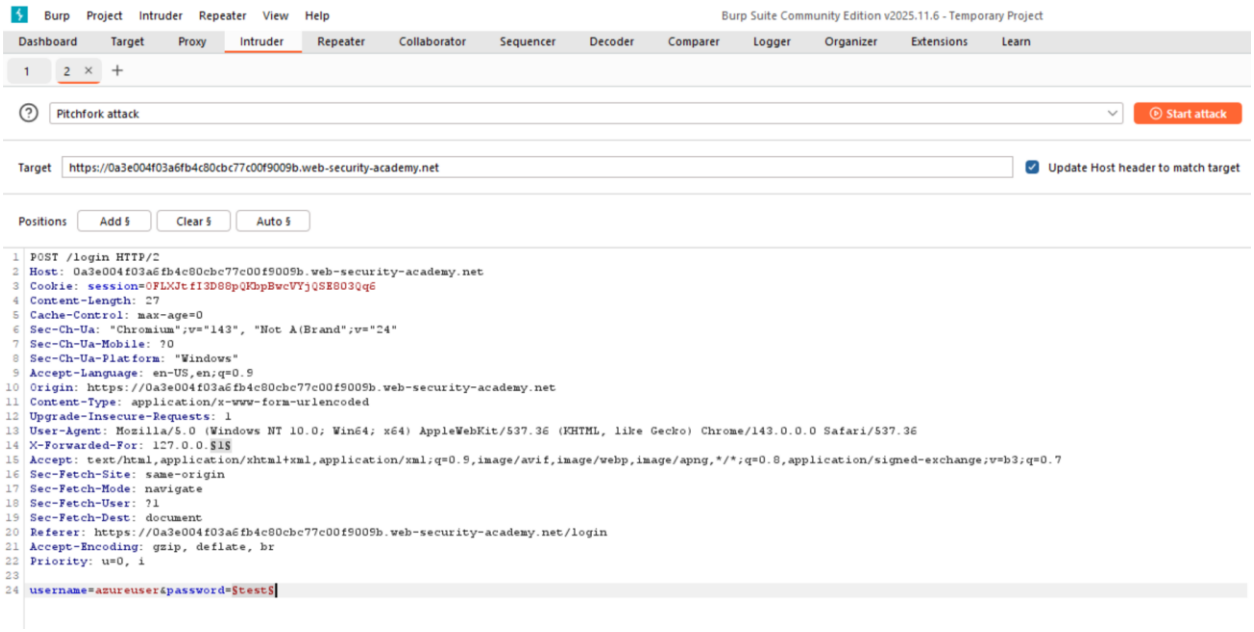
2. **Update Username:** Replace the username parameter payload §test§ with the valid username found: username=azureuser.

3. **Update Password:** Replace the long password string with the password parameter payload marker.

Result: password=§test§

Maintain IP Spoofing: Keep the payload marker on the X-Forwarded-For header unchanged (X-Forwarded-For: 127.0.0.§1§).

Attack Type: Ensure the attack type remains **"Pitchfork"**.



Step 7: Launch the Password Brute-Force Attack

Action: configure the payloads to iterate through IP addresses and candidate passwords simultaneously, then execute the attack to find the correct credential.

Execution:

1. Navigate to the **Intruder > Payloads** tab.

2. **Payload Set 1 (IP Spoofing):** Keep the settings from the previous step (**Numbers**, 1 to 100, Step 1).

3. Payload Set 2 (Passwords):

Select **Payload set: 2**.

Important: Click **"Clear"** to remove the old username list.

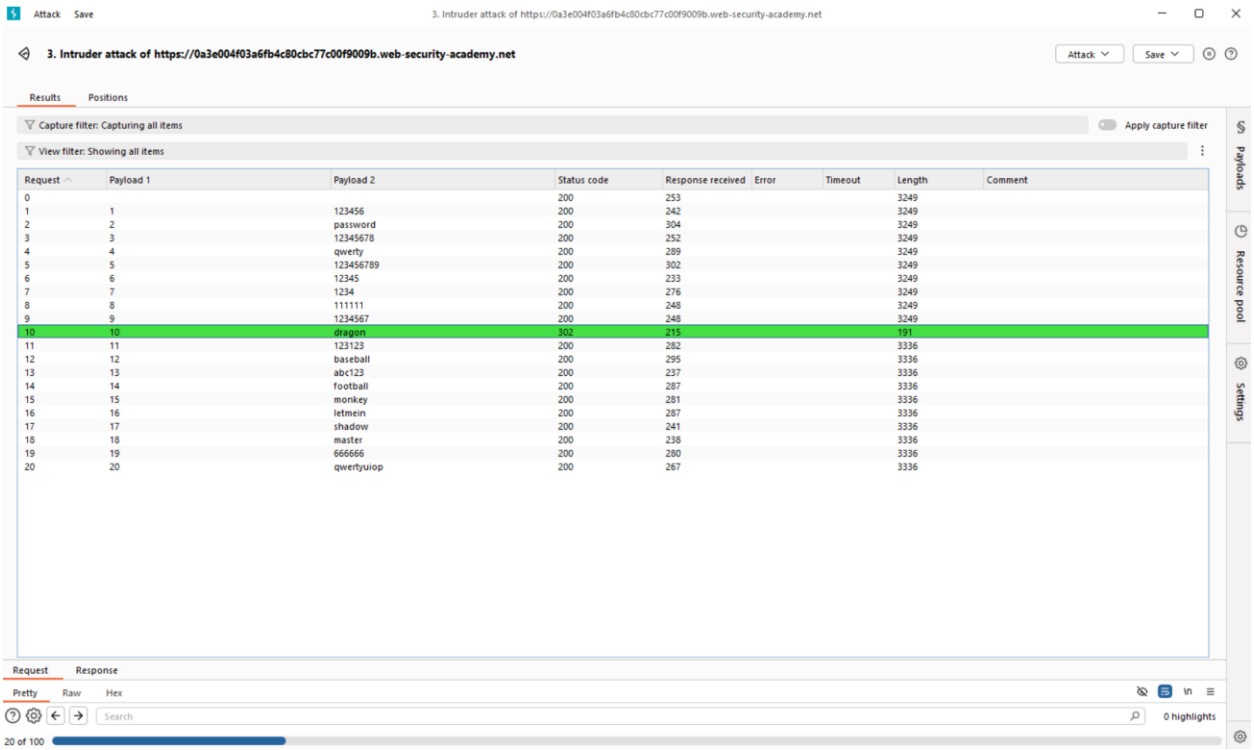
Paste the list of **Candidate Passwords** (from the lab description) into the list.

4. Click **"Start attack"**.

5. **Analysis:** Wait for the attack to finish. Click on the **"Status"** column header to sort the results.

6. Observation: Identify the single request that returns a **302** status code (Found/Redirect). This indicates a successful login.

7. **Conclusion:** A 302 status code indicates a successful login and redirection. Therefore, the valid credentials are **Username: azureuser** and **Password: dragon**.



Step 8: Final Verification (Login)

Action: Verify the discovered credentials by manually logging into the application to solve the lab.

Execution:

1. Return to the browser and the lab login page.

2. Enter the username: **azureuser**.

3. Enter the password: **dragon**.

4. Click the **"Log in"** button.

5. **Result:** The application should log you in successfully. The lab banner status should change to **"Solved"**.