

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.

0ab9001004cd692880bc35b2008c0054.web-security-academy.net

WebSecurity Academy Username enumeration via different responses LAB Not solved


Home | My account

WE LIKE TO BLOG



It's All Just A Click Away
What I love most about Social Media is how it fills my days. Time just evaporates with every word I read, every video I watch and every pointless self-analyzing quiz I take part in. I used to tell people I...

[View post](#)



I'm At A Loss Without It - Leaving Your Smartphone Behind
The other day I left my purse in a friend's car. This led to the most disturbing 19 hours of my life until it was returned to me.

[View post](#)

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.

0ab9001004cd692880bc35b2008c0054.web-security-academy.net/login

WebSecurity Academy Username enumeration via different responses LAB Not solved

Home | My account

Login

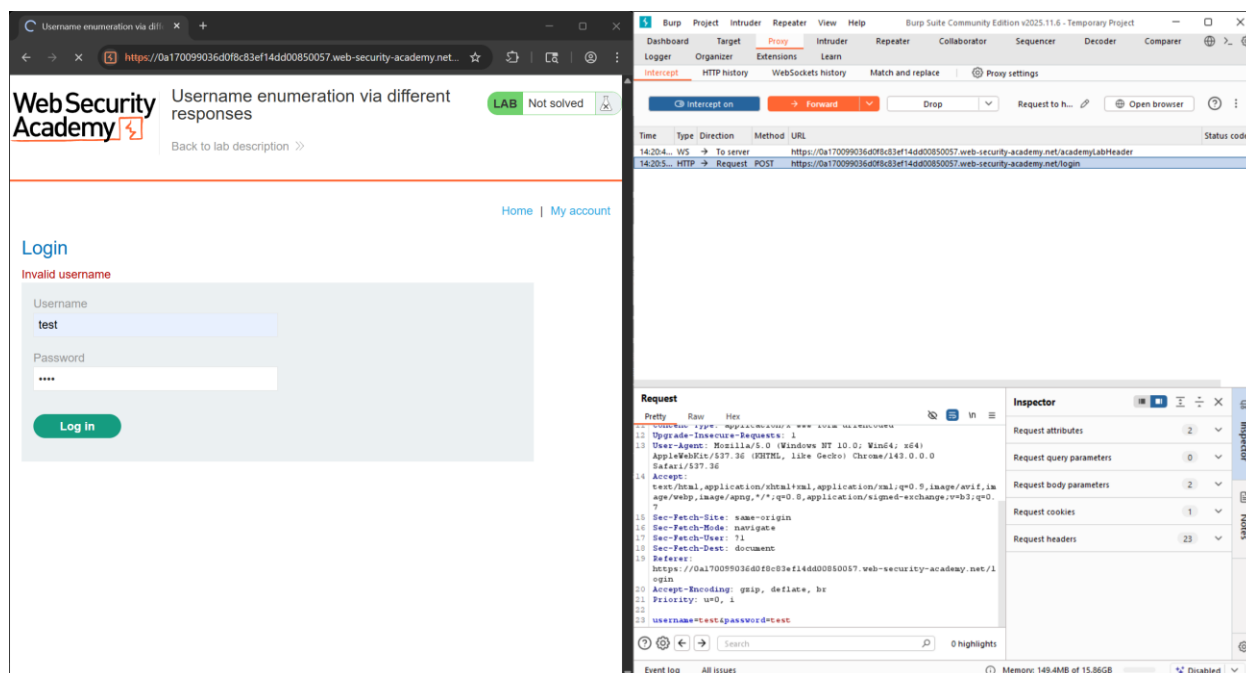
Username

Password

[Log in](#)

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.

The screenshot shows the PortSwigger Web Security Academy interface. On the left is a blue sidebar with a 'Back to all topics' link and a list of authentication-related topics. The main content area is titled 'Lab: Username enumeration via different responses' and is labeled 'APPRENTICE'. It describes a lab vulnerable to username enumeration and password brute-force attacks. A red arrow points to the 'Candidate usernames' link in the wordlists section. Below the description is an 'ACCESS THE LAB' button and two expandable sections for 'Solution' and 'Community solutions'.

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.

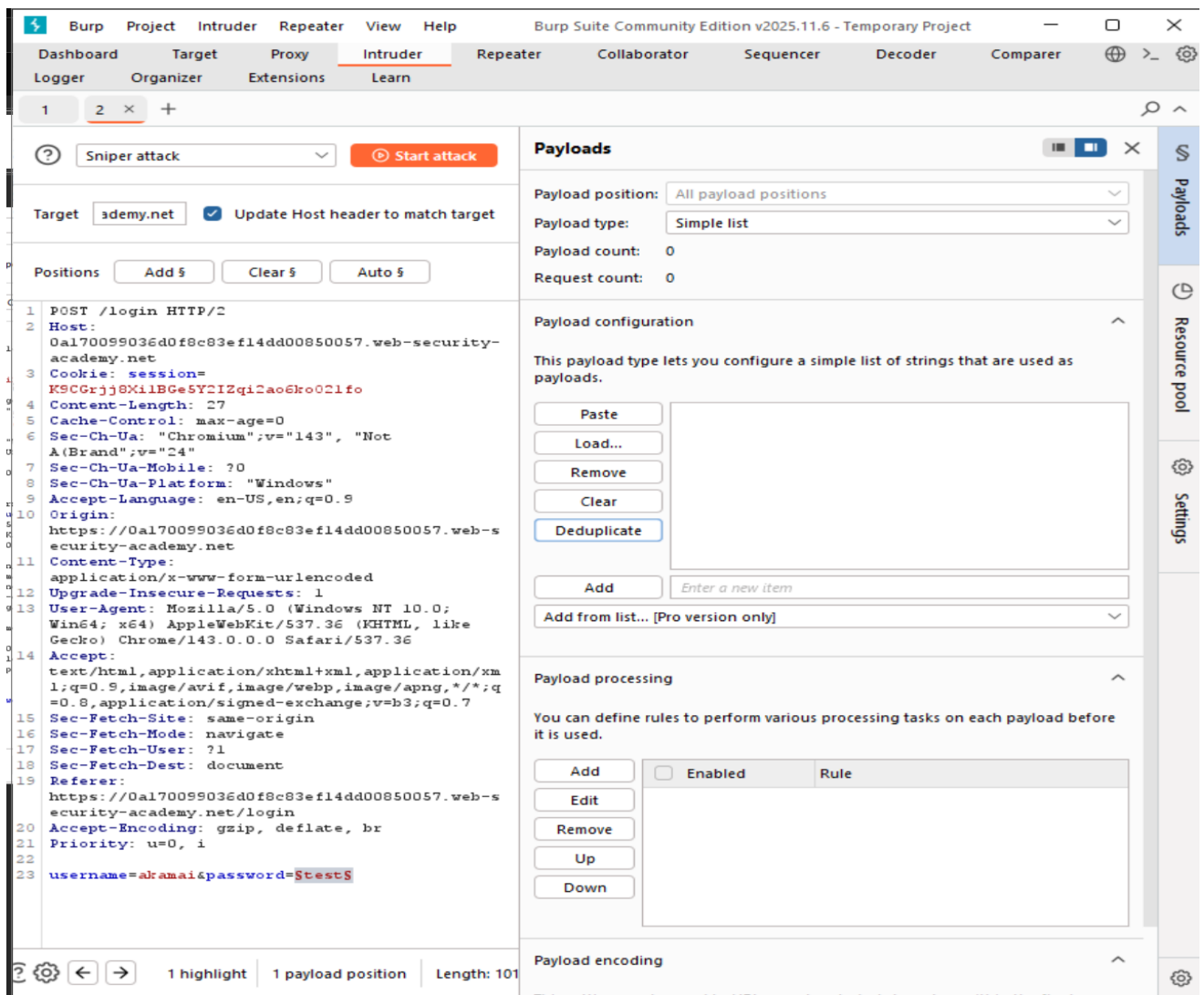
- **Conclusion:** This discrepancy indicates that akamai is the valid username. The slight difference in length typically corresponds to a different error message (e.g., "Incorrect password" instead of "Invalid username").

The screenshot shows the Burp Suite Intruder tool interface. The main table displays a list of payloads and their corresponding status codes and response lengths. The 'akamai' payload is highlighted, showing a status code of 200 and a response length of 3248.

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
31	admin	200	230			3248	
32	admin	200	230			3248	
33	admin	200	230			3248	
34	admin	200	230			3248	
35	admin	200	230			3248	
36	admin	200	230			3248	
37	admin	200	230			3248	
38	admin	200	230			3248	
39	admin	200	230			3248	
40	admin	200	230			3248	
41	admin	200	230			3248	
42	admin	200	230			3248	
43	admin	200	230			3248	
44	admin	200	230			3248	
45	admin	200	230			3248	
46	admin	200	230			3248	
47	admin	200	230			3248	
48	admin	200	230			3248	
49	admin	200	230			3248	
50	akamai	200	230			3248	
51	al	200	270			3248	
52	alabama	200	266			3248	
53	alaska	200	236			3248	
54	albuquerque	200	238			3248	
55	alerts	200	238			3248	
56	alpha	200	236			3248	
57	alterwind	200	269			3248	
58	am	200	221			3248	
59	amarillo	200	218			3248	
60	americas	200	227			3248	
61	an	200	223			3248	
62	anaheim	200	271			3248	
63	analyzer	200	271			3248	
64	announcements	200	225			3248	
65	announcements	200	299			3248	

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.

The screenshot shows the PortSwigger Web Security Academy interface. The top navigation bar includes the PortSwigger logo, a 'MY ACCOUNT' button, and links for Products, Solutions, Research, Academy, and Support. Below this is a secondary navigation bar with links for Dashboard, Learning paths, Latest topics, All content, Hall of Fame, Get started, and Get certified. A left sidebar contains a list of topics under 'Authentication vulnerabilities', including 'What is authentication?', 'How vulnerabilities arise', 'Impact of vulnerable authentication', 'Vulnerabilities in password-based authentication', 'Vulnerabilities in multi-factor authentication', 'Vulnerabilities in other authentication mechanisms', 'Vulnerabilities in OAuth authentication', 'Securing your authentication mechanisms', and 'View all authentication labs'. The main content area displays the lab title 'Lab: Username enumeration via different responses' with an 'APPRENTICE' badge. The lab description states: 'This lab is vulnerable to username enumeration and password brute-force attacks. It has an account with a predictable username and password, which can be found in the following wordlists:'. Two bullet points follow: 'Candidate usernames' and 'Candidate passwords', with a red line drawn under the second one. The instructions continue: 'To solve the lab, enumerate a valid username, brute-force this user's password, then access their account page.' Below the text is an orange button labeled 'ACCESS THE LAB'. At the bottom, there are two expandable sections: 'Solution' and 'Community solutions'.

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.

Attack Save 3. Intruder attack of https://0a170099036d0f8c3ef14dd00850057.web-security-academy.net

3. Intruder attack of https://0a170099036d0f8c3ef14dd00850057.web-security-academy.net Attack Save

Results Positions

Capture filter: Capturing all items Apply capture filter

View filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
25	654321	200	227			3250	
26	superman	200	226			3250	
27	1qaz2wsx	200	267			3250	
28	7777777	200	266			3250	
29	121212	200	229			3250	
30	000000	200	228			3250	
31	qazwsx	200	227			3250	
32	123qwe	200	227			3250	
33	killer	200	267			3250	
34	trustno1	200	228			3250	
35	jordan	200	228			3250	
36	jennifer	200	268			3250	
37	zxcvbnm	200	229			3250	
38	asdfgh	200	228			3250	
39	hunter	200	267			3250	
40	butler	200	227			3250	
41	soccer	200	269			3250	
42	harley	200	225			3250	
43	batman	200	268			3250	
44	andrew	302	229			188	
45	tigger	200	227			3337	
46	sunshine	200	265			3337	
47	iloveyou	200	265			3337	
48	2000	200	225			3337	
49	charlie	200	265			3337	
50	robert	200	223			3337	
51	thomas	200	224			3337	
52	hockey	200	225			3337	
53	ranger	200	263			3337	
54	daniel	200	229			3337	
55	starwars	200	223			3337	
56	klaster	200	227			3337	
57	112233	200	225			3337	
58	george	200	224			3337	

Request Response


Pretty Raw Hex

Search 0 highlights

Finished

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".



WebSecurity Academy 

Username enumeration via different responses

LAB Solved

[Back to lab description >>](#)

Congratulations, you solved the lab!

Share your skills!   Continue learning >>

[Home](#) | [My account](#) | [Log out](#)

My Account

Your username is: akamai

Your email is: akamai@normal-user.net

Email

Update email

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

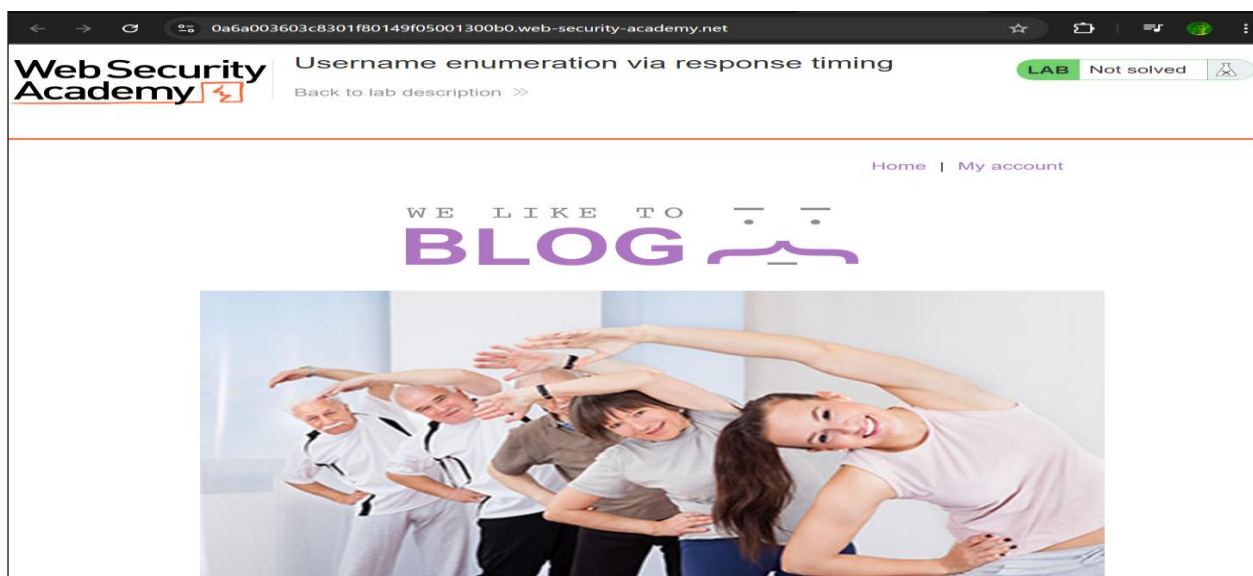
Based on the provided solution analysis, the exploitation process involves the following key phases:

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

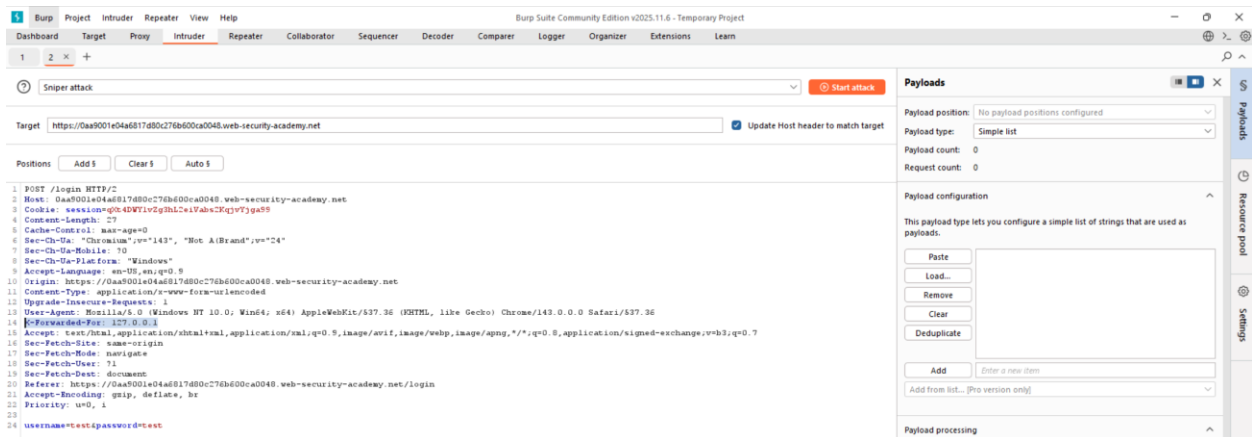
Step 1: Access the Lab

- **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.

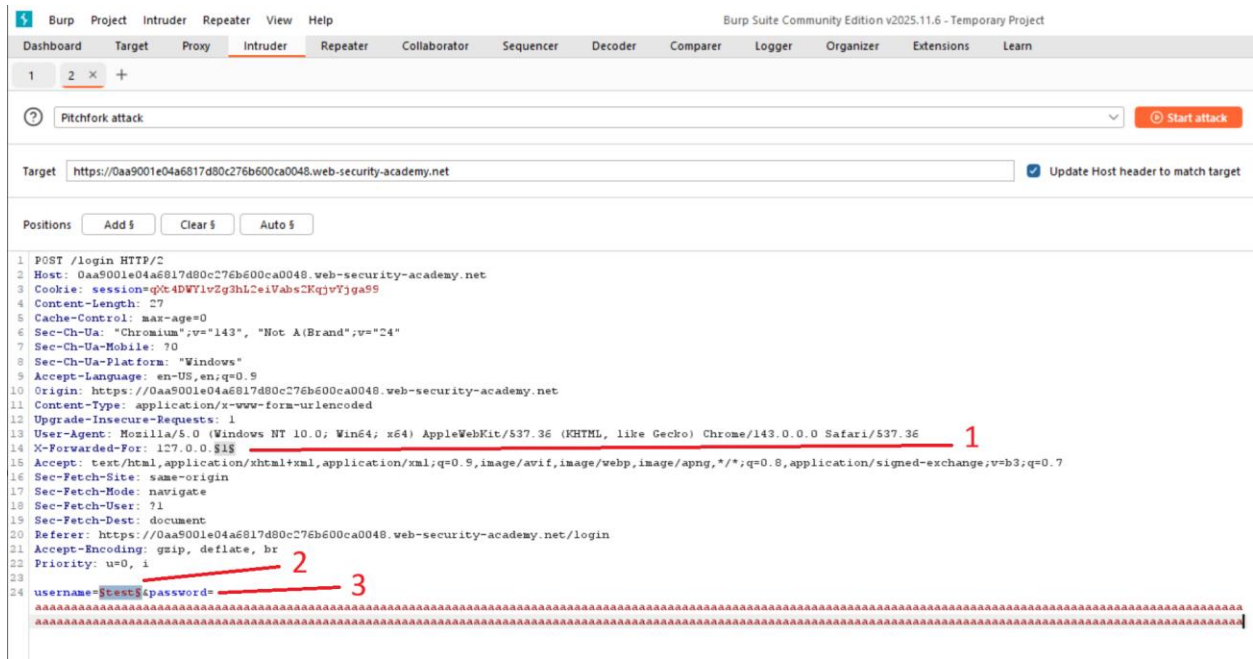


Step 3: Configure the Pitchfork Attack

- **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:

(ensure no § markers are on the password).



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

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

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

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

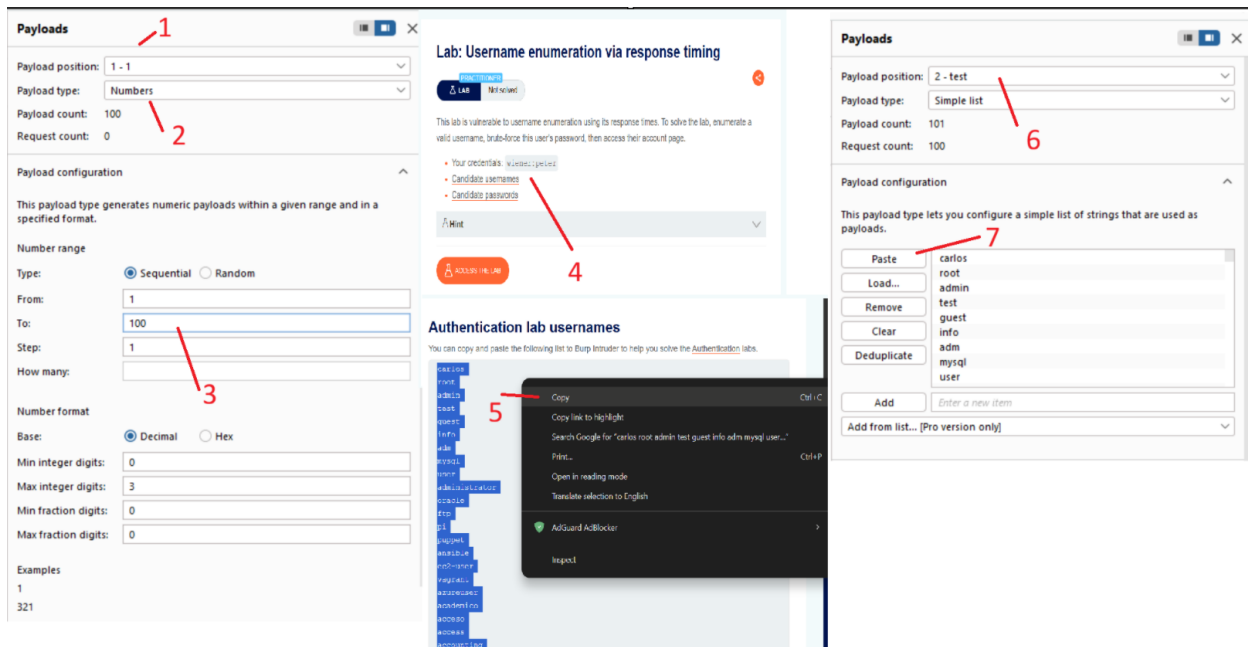
Set **From: 1, To: 100**, and **Step: 1**. This will generate numbers from 1 to 100 to create unique fake IPs (e.g., 127.0.0.1, 127.0.0.2, etc.).

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.

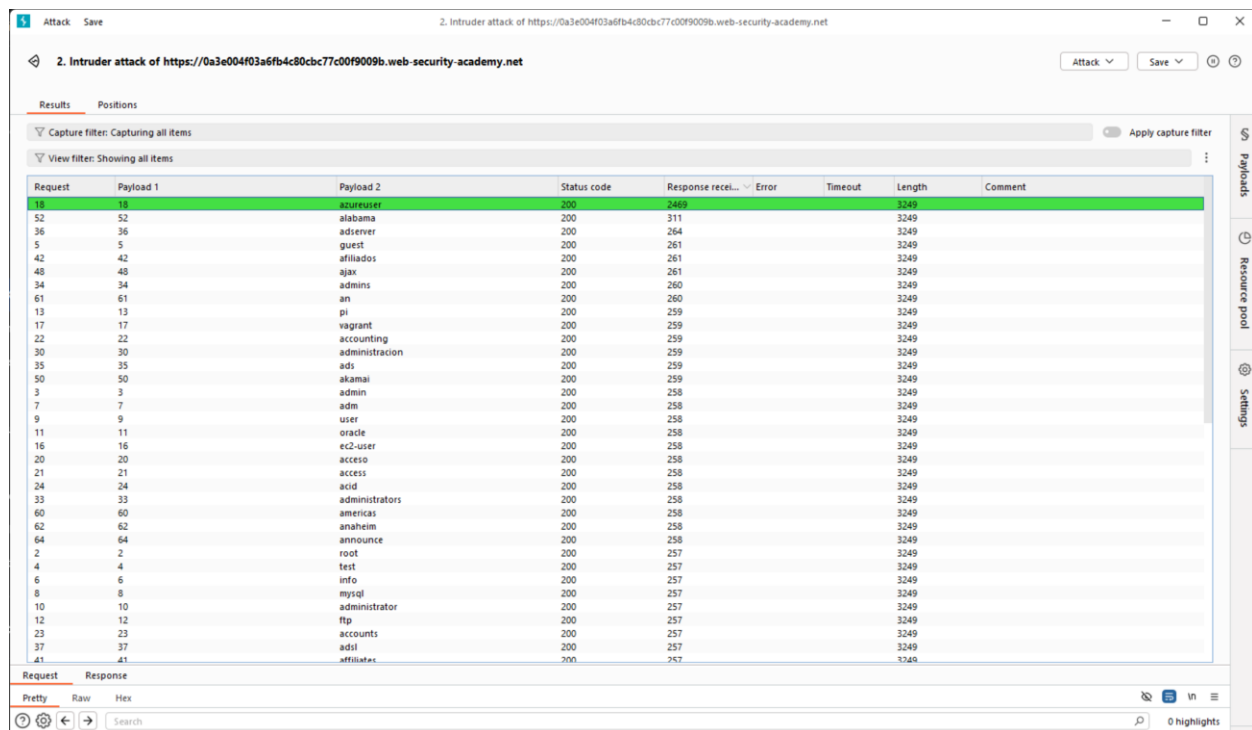
2. Enable Timing Columns: In the attack results window, go to the **"Columns"** menu (or right-click the header row) and ensure **"Response received"** and **"Response completed"** are checked.

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**.



The screenshot shows the Burp Suite interface with the 'Results' tab selected. The table displays the results of an intruder attack on the URL https://0a3e004f03a6fb4c80bc77c00f9009b.web-security-academy.net. The table has columns for Request, Payload 1, Payload 2, Status code, Response received, Error, Timeout, Length, and Comment. The request with the longest response time (2469 ms) is highlighted in green, corresponding to the payload 'azureuser'.

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
18	18	azureuser	200	2469			3249	
52	52	alabama	200	311			3249	
36	36	adserver	200	264			3249	
5	5	guest	200	261			3249	
42	42	afiliados	200	261			3249	
48	48	ajax	200	261			3249	
34	34	admins	200	260			3249	
61	61	an	200	260			3249	
13	13	pi	200	259			3249	
17	17	vagrant	200	259			3249	
22	22	accounting	200	259			3249	
30	30	administracion	200	259			3249	
35	35	ads	200	259			3249	
50	50	akamai	200	259			3249	
3	3	admin	200	258			3249	
7	7	adm	200	258			3249	
9	9	user	200	258			3249	
11	11	oracle	200	258			3249	
16	16	ec2-user	200	258			3249	
20	20	access	200	258			3249	
21	21	access	200	258			3249	
24	24	acid	200	258			3249	
33	33	administrators	200	258			3249	
60	60	americas	200	258			3249	
62	62	anaheim	200	258			3249	
64	64	announce	200	258			3249	
2	2	root	200	257			3249	
4	4	test	200	257			3249	
6	6	info	200	257			3249	
8	8	mysql	200	257			3249	
10	10	administrator	200	257			3249	
12	12	ftp	200	257			3249	
23	23	accounts	200	257			3249	
37	37	adsl	200	257			3249	
41	41	affiliates	200	257			3249	

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.

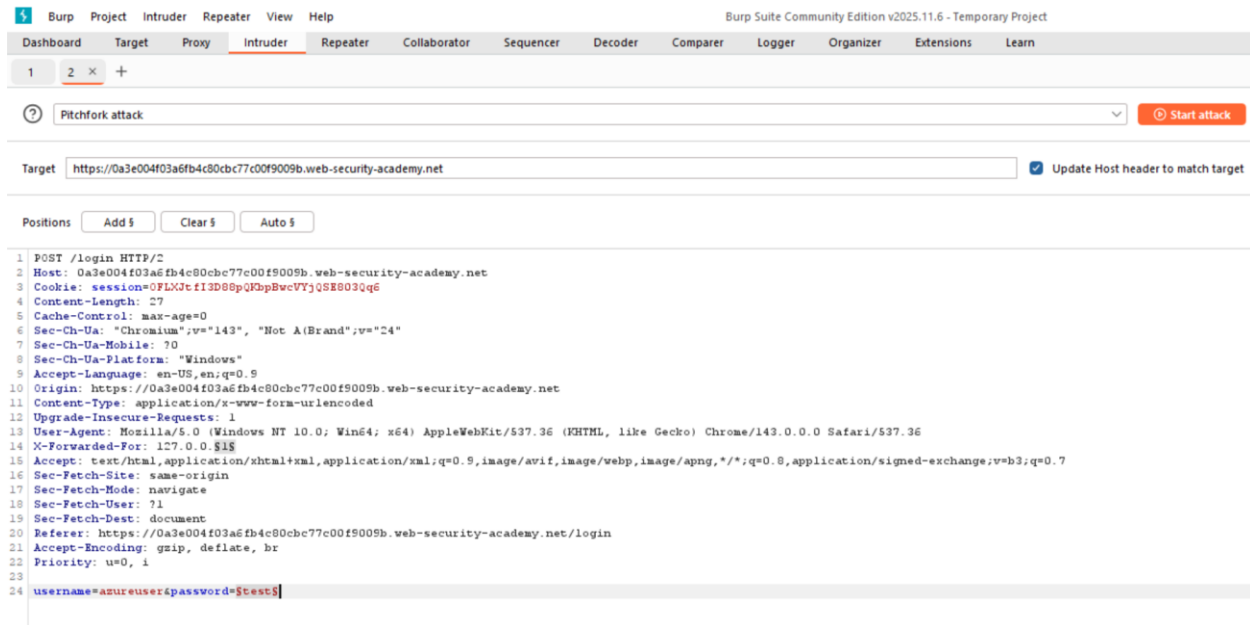
Execution:

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.

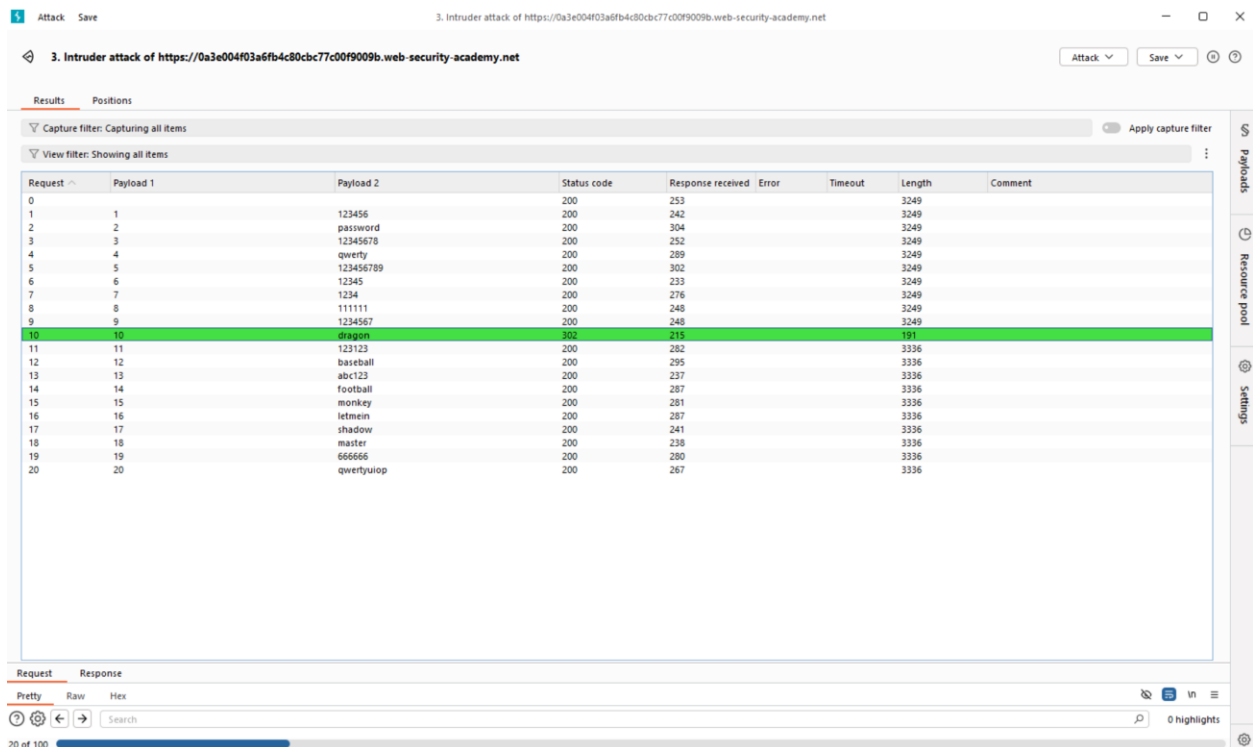
Paste the list of **Candidate Passwords** (from the lab description) into the list.

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**.



The screenshot shows the Burp Suite interface with an intruder attack titled "3. Intruder attack of https://0a3e004f03a6fb4c80cbc77c00f9009b.web-security-academy.net". The "Results" tab is active, displaying a table of captured requests. The table has columns for Request, Payload 1, Payload 2, Status code, Response received, Error, Timeout, Length, and Comment. Request 10 is highlighted in green, showing a status code of 302 and a response length of 191. The payload for request 10 is "dragon".

Request	Payload 1	Payload 2	Status code	Response received	Error	Timeout	Length	Comment
0			200	253			3249	
1	123456		200	242			3249	
2	password		200	304			3249	
3	12345678		200	252			3249	
4	qwerty		200	289			3249	
5	123456789		200	302			3249	
6	12345		200	233			3249	
7	1234		200	276			3249	
8	1111111		200	248			3249	
9	1234567		200	248			3249	
10	10	dragon	302	215			191	
11	11	123123	200	282			3336	
12	12	baseball	200	295			3336	
13	13	abc123	200	237			3336	
14	14	football	200	287			3336	
15	15	monkey	200	281			3336	
16	16	letmein	200	287			3336	
17	17	shadow	200	241			3336	
18	18	master	200	238			3336	
19	19	666666	200	280			3336	
20	20	qwertyuiop	200	267			3336	

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"**.

The screenshot shows a web browser window with the URL `https://0a3e004f03a6fb4c80cbc77c00f9009b.web-security-academy.net/...`. The page header includes the Web Security Academy logo and the lab title "Username enumeration via response timing". The lab status is "Solved", indicated by a green "LAB Solved" badge. Below the header, there is an orange banner with the text "Congratulations, you solved the lab!" and links to "Share your skills!" (with Twitter and LinkedIn icons) and "Continue learning >>". The main content area is titled "My Account" and displays the user's information: "Your username is: azureuser" and "Your email is: azureuser@normal-user.net". Below this information is a form with an "Email" label, a text input field, and a green "Update email" button.

Web Security Academy

Username enumeration via response timing

LAB Solved

Back to lab description >>

Congratulations, you solved the lab!

Share your skills! Continue learning >>

Home | My account | Log out

My Account

Your username is: azureuser

Your email is: azureuser@normal-user.net

Email

Update email