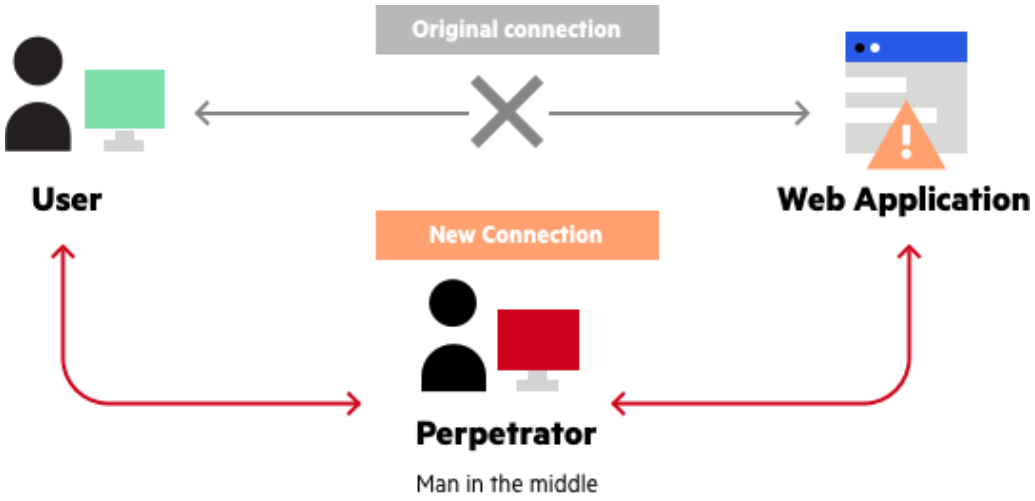
MAN-IN-THE-MIDDLE ATTACK

26th Sep 2022

OVERVIEW

A man-in-the-middle (MITM) attack is a general term for when a perpetrator positions himself in a conversation between a user and an application—either to eavesdrop or to impersonate one of the parties, making it appear as if a normal exchange of information is underway.

Information obtained during an attack could be used for many purposes, including identity theft, unapproved fund transfers or an illicit password change.



INSTALLATION OF BURP SUITE (COMMUNITY EDITION)

Step 1: Download

Download the latest version of Burp Suite Community Edition.

Step 2: Install

Run the installer and launch Burp Suite. When asked to select a project file and configuration, just click **Next** and then **Start Burp** to skip this for now.

USAGES OF BURP SUITE

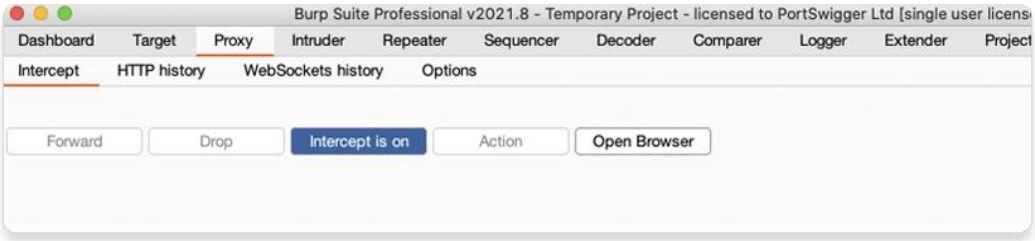
- **Intercepting & Modifying HTTP traffic with Burp Proxy.**
- **Burp Intruder & many more.**

INTERCEPTING & MODIFYING A REQUEST

Burp Proxy lets you intercept HTTP requests and responses sent between Burp's browser and the target server. This enables you to study how the website behaves when you perform different actions.

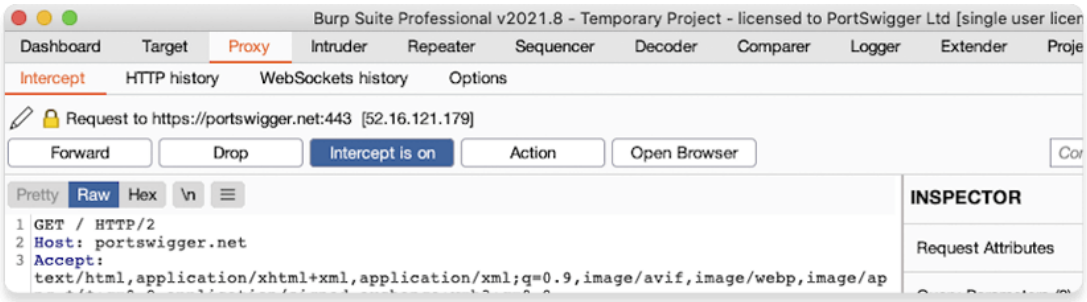
Step 1: Launch Burp's browser

- **Go to the Proxy > Intercept tab.**
- **Click the Intercept is off button, so it toggles to Intercept is on.**



Step 2: Intercept a request

Using Burp's browser, try to visit https://portswigger.net and observe that the site doesn't load. Burp Proxy has intercepted the browser's HTTP request before it could reach the server. You can see this intercepted request on the Proxy > Intercept tab.

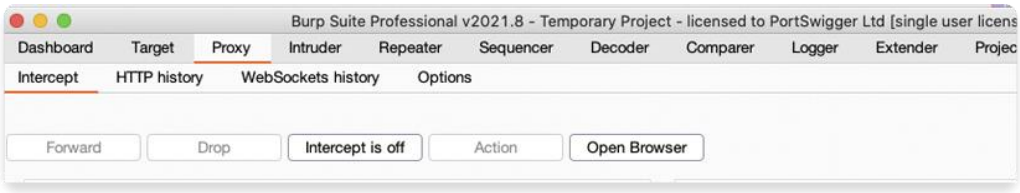


Step 3: Forward the request

Click the **Forward** button several times to send the intercepted request, and any subsequent ones, until the page loads in Burp's browser.

Step 4: Switch off the interception

Due to the number of requests browsers typically send, you often won't want to intercept every single one of them. Click the **Intercept is on** button so that it now says **Intercept is off**.



BURP INTRUDER

Step 1: Access the lab

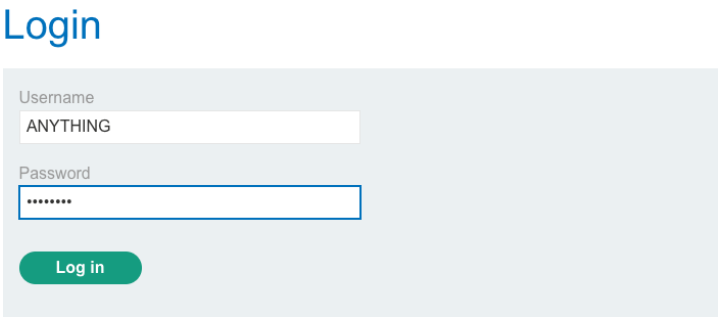
Open Burp's browser, and use it to access the following URL:

https://portswigger.net/web-security/authentication/password-based/lab-username-enumeration-via-different-responses

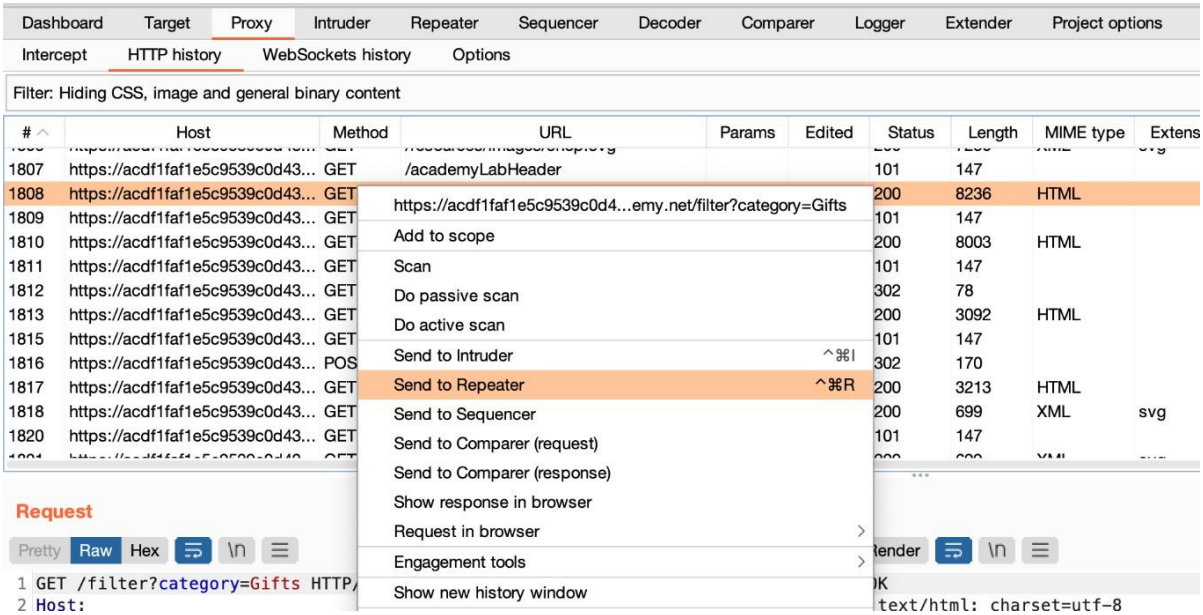
Click **Access the lab** and log in to your PortSwigger account if prompted. This opens your instance of a deliberately vulnerable blog website.

Step 2: Try to log in

Click **My account**, then try to log in using an invalid username and password. In Burp Suite, go to the Proxy > HTTP history tab. This shows all of the requests you have made in Burp's browser since opening it.



Find the POST /login request and send it to Burp Intruder.



Step 3: Set the payload positions

Notice that Burp Intruder has automatically inserted § characters in various positions throughout the request. These mark the beginning and end of a payload position, where Burp Intruder will attempt to insert payloads during the attack.

Payload Positions

Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: https://ac3d1ff51eb082f4c0b66c5e00ca00ac.web-security-academy.net

```
1  POST /login HTTP/1.1
2  Host: ac3d1ff51eb082f4c0b66c5e00ca00ac.web-security-academy.net
3  Cookie: session=§k1w8xY6wUGZi6HhOLRjxDdo2fT8slujQ§
4  Content-Length: 35
5  Cache-Control: max-age=0
6  Sec-Ch-Ua: "(Not(A:Brand";v="8", "Chromium";v="98"
7  Sec-Ch-Ua-Mobile: ?0
8  Sec-Ch-Ua-Platform: "macOS"
9  Upgrade-Insecure-Requests: 1
10 Origin: https://ac3d1ff51eb082f4c0b66c5e00ca00ac.web-security-academy.net
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.80
13 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicat
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://ac3d1ff51eb082f4c0b66c5e00ca00ac.web-security-academy.net/login
19 Accept-Encoding: gzip, deflate
20 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
21 Connection: close
22
23 username=§ANYTHING§&password=§ANYTHING§
```

For this attack, we only need a single payload position in the username parameter. Click **Clear §** to clear the default positions. Highlight the value of the username parameter, then click **Add §**.

Payload Positions

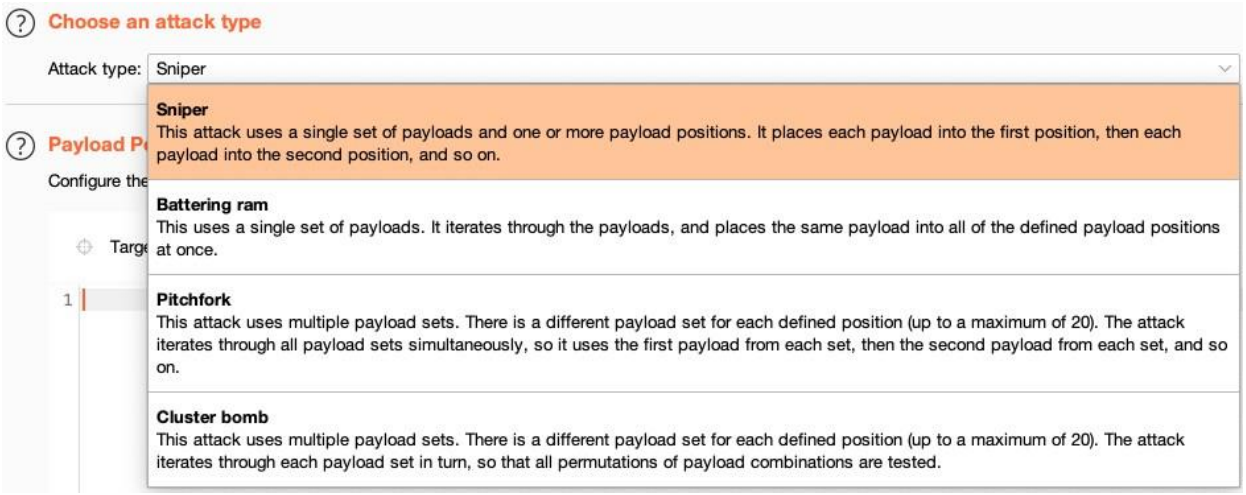
Configure the positions where payloads will be inserted, they can be added into the target as well as the base request.

Target: https://ac3d1ff51eb082f4c0b66c5e00ca00ac.web-security-academy.net

```
1  POST /login HTTP/1.1
2  Host: ac3d1ff51eb082f4c0b66c5e00ca00ac.web-security-academy.net
3  Cookie: session=k1w8xY6wUGZi6HhOLRjxDdo2fT8slujQ
4  Content-Length: 35
5  Cache-Control: max-age=0
6  Sec-Ch-Ua: "(Not(A:Brand";v="8", "Chromium";v="98"
7  Sec-Ch-Ua-Mobile: ?0
8  Sec-Ch-Ua-Platform: "macOS"
9  Upgrade-Insecure-Requests: 1
10 Origin: https://ac3d1ff51eb082f4c0b66c5e00ca00ac.web-security-academy.net
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/98.0.4758.80
13 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,applicat
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: ?1
17 Sec-Fetch-Dest: document
18 Referer: https://ac3d1ff51eb082f4c0b66c5e00ca00ac.web-security-academy.net/login
19 Accept-Encoding: gzip, deflate
20 Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
21 Connection: close
22
23 username=§ANYTHING§&password=ANYTHING
```

Step 4: Select an attack type

At the top of the screen, you can select different attack types. For now, just make sure this is set to **Sniper**.



Step 5: Add the payloads

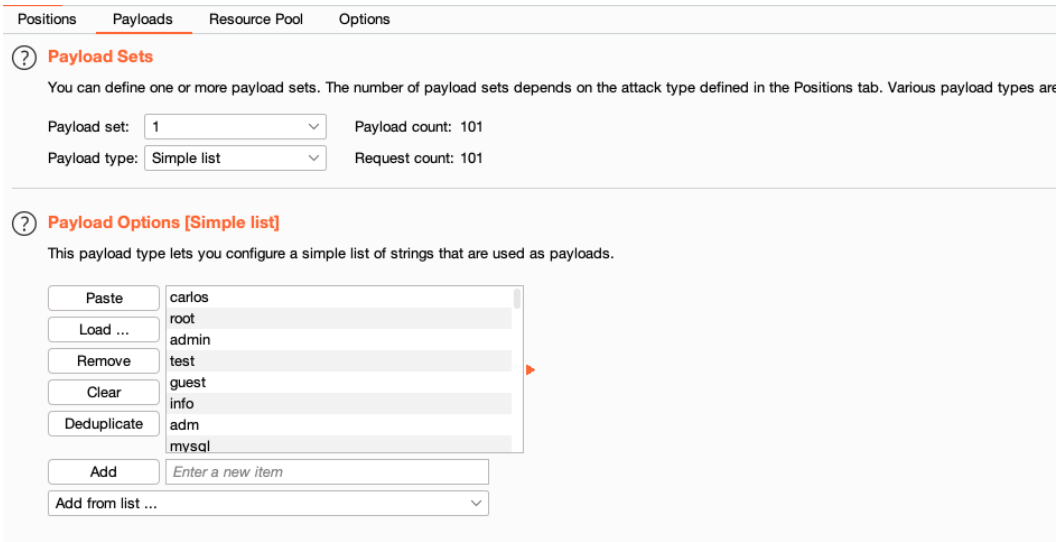
You now just need to configure the list of payloads that you want to use. For this demonstration, we'll try sending the request with different usernames to test how the login mechanism behaves.

Go to the **Payloads** sub-tab.

Leave the **Payload type** set to **Simple list**.

In the **Payload options** section, click **Paste** to add the copied usernames to the list.

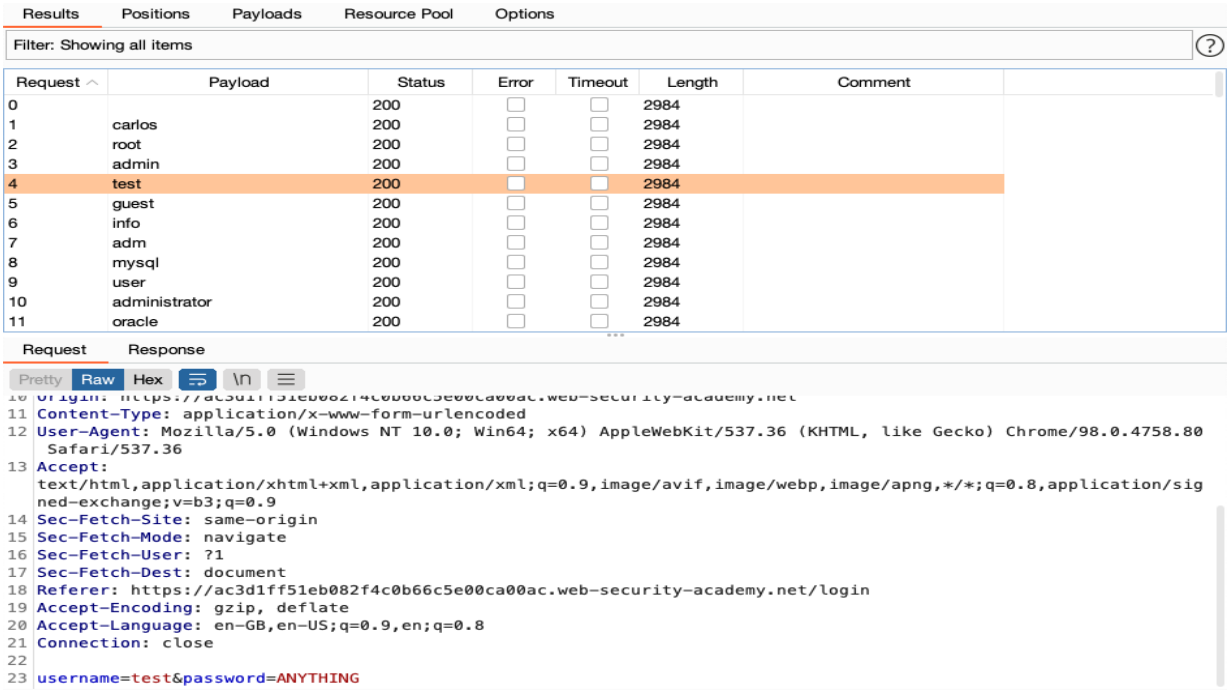
In the **Payload sets** section, you can see how many payloads you have added, and how many requests this attack will send. For this attack, you should see **Payload count: 101 / Request count: 101**.



Step 6: Start the attack

In the upper-right corner, click **Start attack**. This opens a new attack window in which you can see each of the requests that Burp Intruder is making.

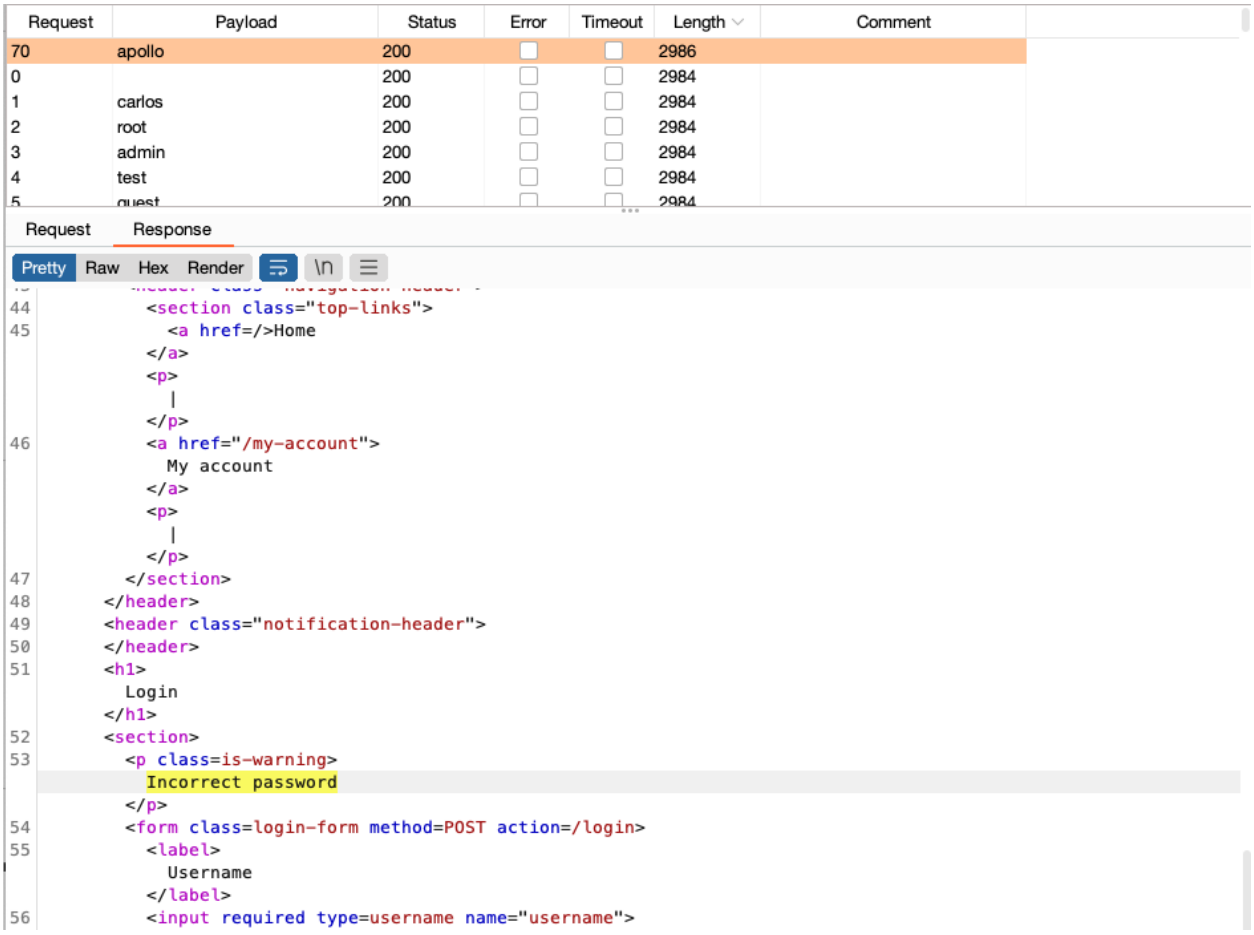
If you select one of the entries in the table, you can view the request and response in the message editor. Notice that the username parameter contains a different value from our payload list in each request.



Step 8: Study the response

Select any request from the list to display it in the message editor.

Studying the responses, notice that most contain an Invalid username error message, but the one with the different length response has an Incorrect password error message. This different response strongly suggests that this username might be valid in this case.



Request	Payload	Status	Error	Timeout	Length	Comment
70	apollo	200			2986	
0		200			2984	
1	carlos	200			2984	
2	root	200			2984	
3	admin	200			2984	
4	test	200			2984	
5	guest	200			2984	

Request Response

Pretty Raw Hex Render \n

```
44          <section class="top-links">
45            <a href=/>Home
              </a>
              <p>
                |
              </p>
46            <a href="/my-account">
                My account
              </a>
              <p>
                |
              </p>
47          </section>
48        </header>
49        <header class="notification-header">
50        </header>
51        <h1>
            Login
          </h1>
52        <section>
53          <p class=is-warning>
              Incorrect password
            </p>
54          <form class=login-form method=POST action=/login>
55            <label>
                Username
              </label>
56            <input required type=username name="username">
```