# Vulnerabilities in multi-factor authentication

## LAB 24 2FA simple bypass

Valid user credentials: `wiener:peter`

Victim's credentials `carlos:montoya`

Checked the authorization logic by trying it wit ha valid user, noticed that during the second step of authentication, and noticed that URL is now without ?login2.

Thus, I logged in as 'carlos', and got on the entering verification code page, removed login2 from the URL and got on to user's page. Lab's done!

## LAB 25 2FA broken logic

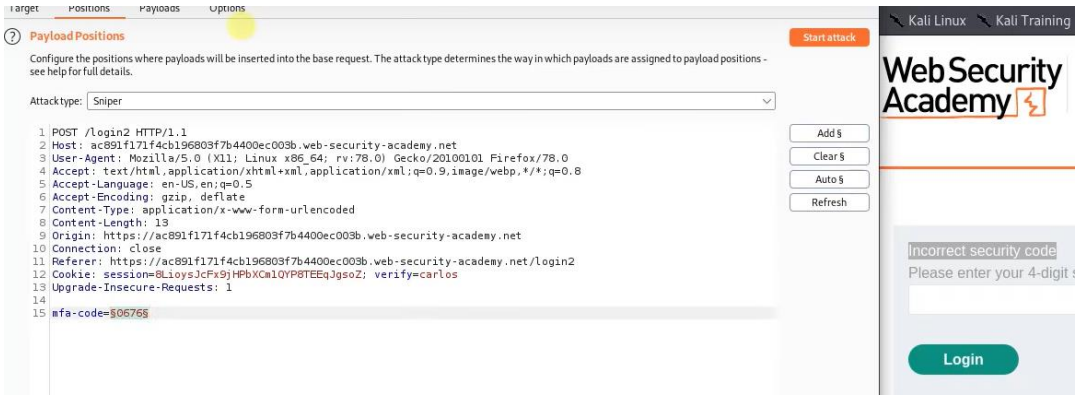
Valid user credentials: `wiener:peter`

Victim's credentials `carlos`

Firsltly, I inspected how 2FA process works. Noted, that it uses verify parameter in POST/login2, therefore it is possible to indirectly log in to victim and generate a his security code by changing valid parameter in Burp Repeater.

Once the replace is done, 2FA code should be generated for 'carlos'.

Next step is to bruteforce the code, using Burp Intruder brute force attack:




After quick brute force, only 1 request returned 302 response, meaning that the verification code (0581) was successfully found.

Redirecting to that URL in browser, it brings me to the 'carlos' user page. Lab's done!

**LAB 26 [2FA bypass using a brute-force attack]**

Victim's credentials: `carlos:montoya`

Inspecting this lab's 2FA logic, I noticed that user is being logged out after 2 unsuccessful login attempts. This can be easily bypassed by any script kiddie by making a short Burp Macro that will automatically log in after each logout. To do so, I followed to Project options > Sessions. In the Session Handling Rules panel, clicked Add. The Session handling rule editor dialog opens.

In the dialog, in the Scope tab, selected the option Include all URLs. (under URL Scope)

In Details tab and under Rule Actions, clicked Add > Run a macro.

Under Select macro click Add to open the Macro Recorder. Select the following 3 requests:

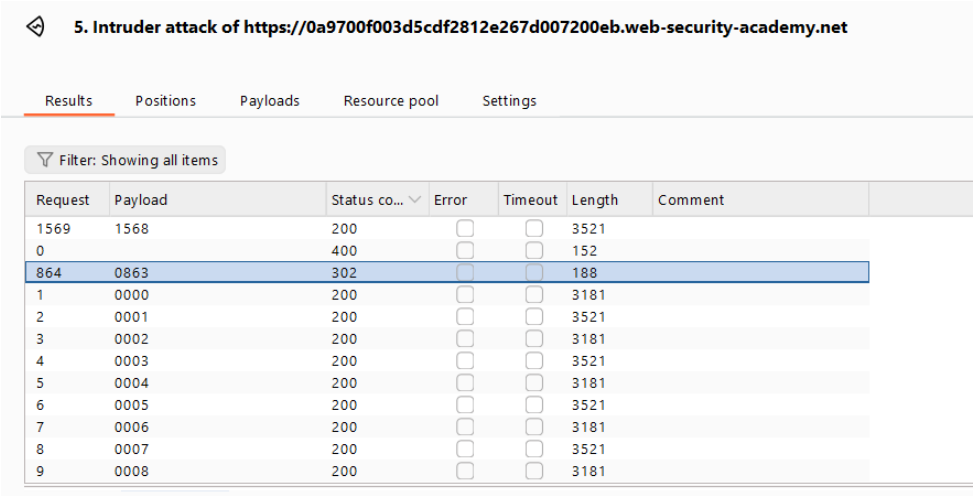
GET /login

POST /login

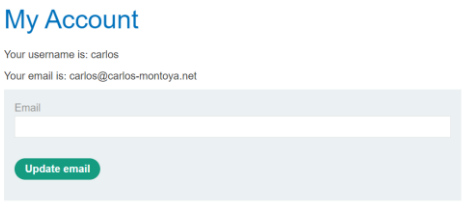
GET /login2

Clicked Test macro and check that the final response contains the page asking me to provide the 4-digit security code. This confirms that the macro is working correctly.

Finally, I sent the POST /login2 request to Burp Intruder and put a bruteforce for mfa-code parameter. I had to adjust the speed of brute force to 1 request at a time, so I would not have constantly being logged out:




Once I received 302 response code, I opened the response link in browser and was redirected to 'carlos' user page.


Lab's done!