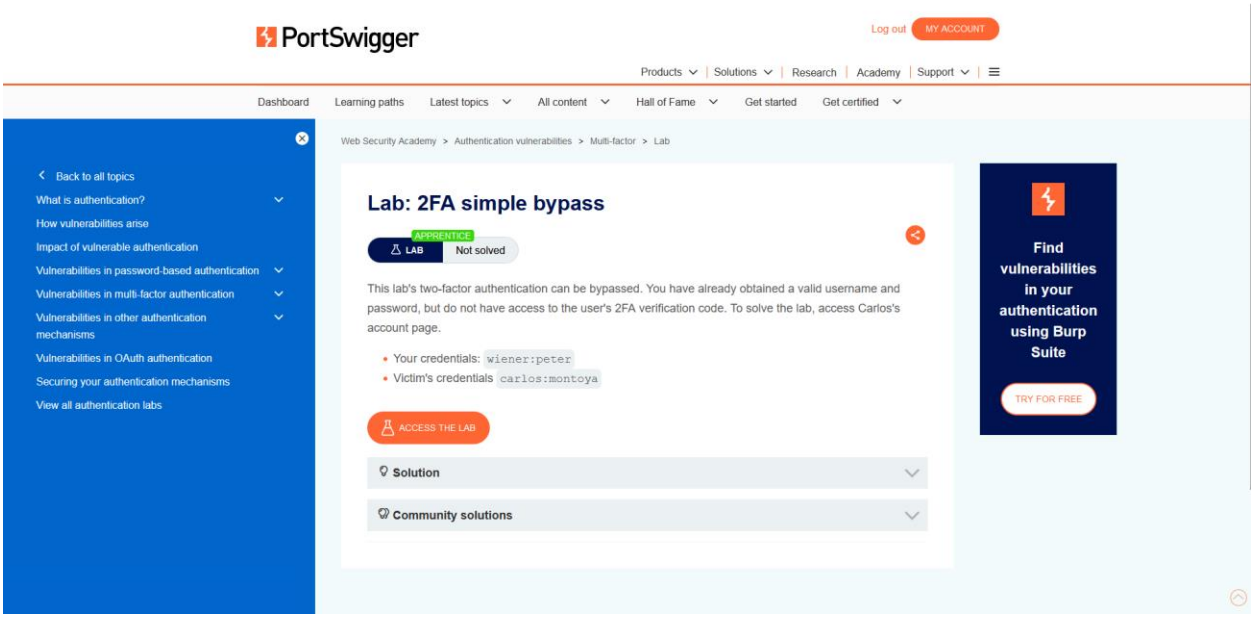
Lab 4 - Authentication-Vulnerabilities (multi-factor authentication)	
Name	Dang Hoang Nguyen
Student ID	SE171946

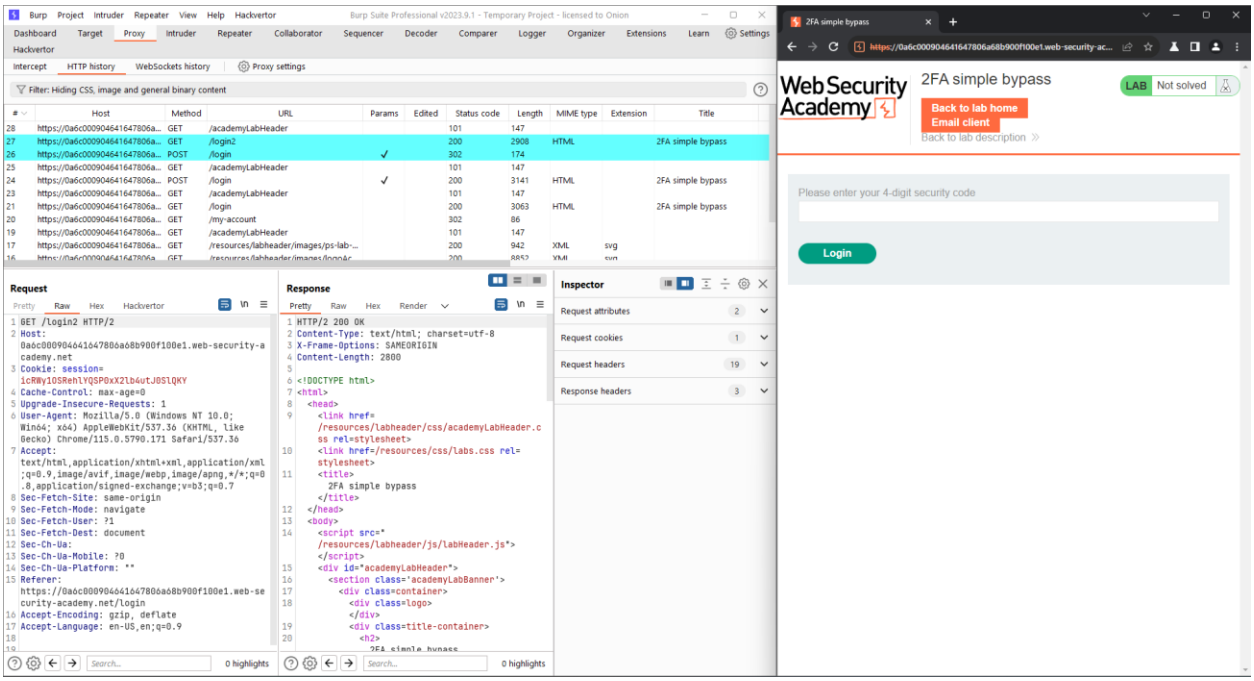
Giới Thiệu: Trong bài lab này ta sẽ tìm hiểu về lỗ hổng multi-factor authentication

I. 2FA simple bypass <Here>



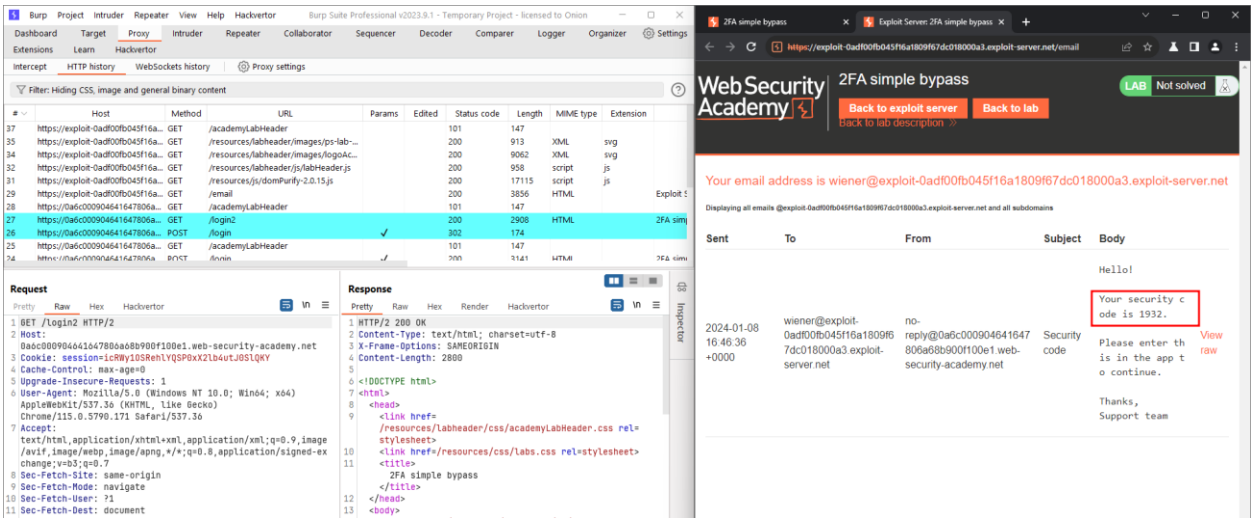
- ⇨ Mục tiêu của bài lab này là truy cập được vào trang tài khoản của **Carlos**
- ⇨ Thông tin mà bài lab cung cấp là
  - Account dùng để test: **wiener:peter**
  - Account nạn nhân: **carlos:montoya**

1. Đầu tiên ta sẽ thử login vào bằng tài khoản dùng để test



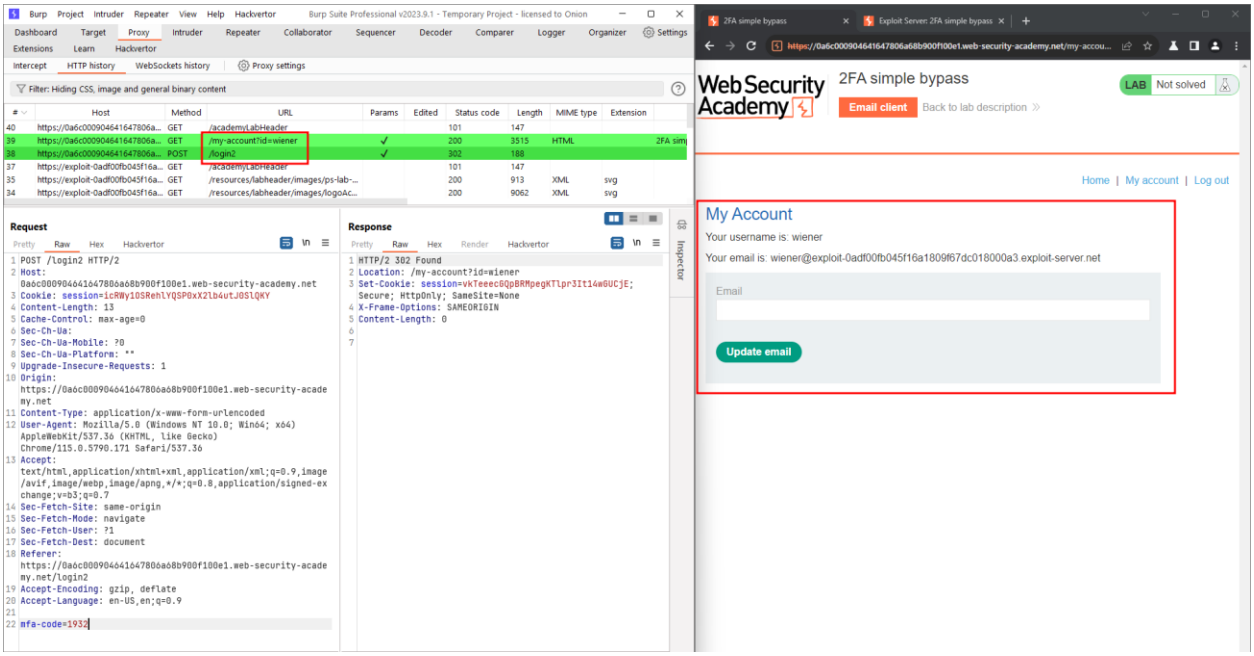
⇨ Có thể thấy được sau khi gửi request với method POST lên thì nó sẽ tiếp tục redirect sang trang khác yêu cầu ta nhập **security code.**

2. Tiếp theo bấm vào nút **Email client**



⇨ Có thể thấy được trong mail gửi đến user là security code mà trước đó đã yêu cầu.

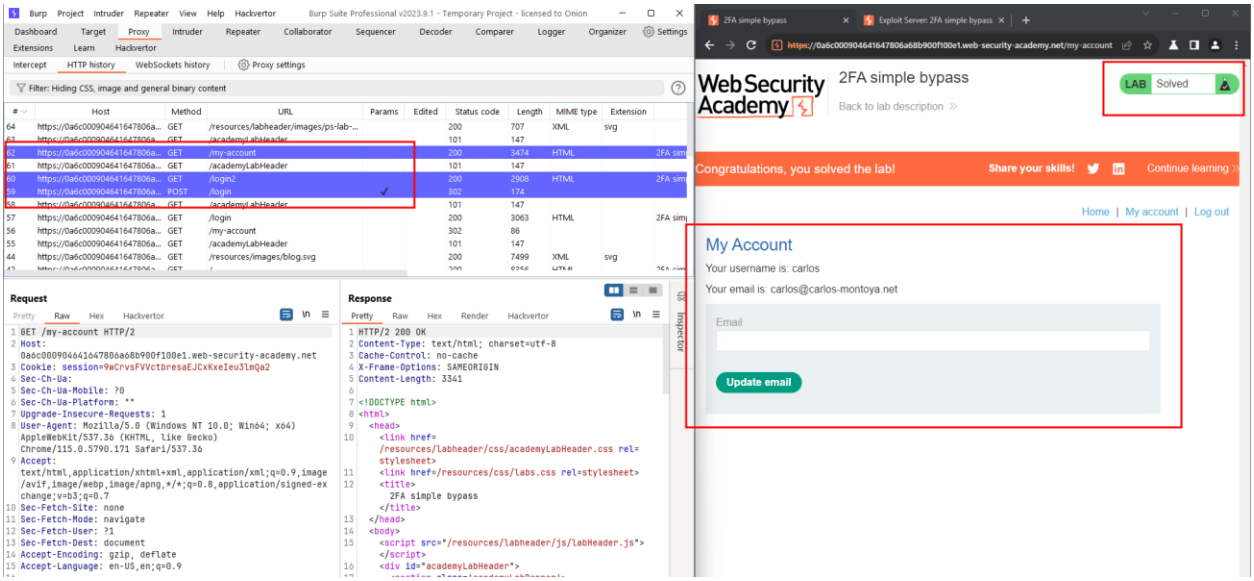
3. Giờ ta quay lại trang nhập security code và nhập "**1932**"



⇨ Có thể thấy sau khi nhập xong code thì nó sẽ được redirect đến trang quản lý account bằng đường dẫn "**/my-account**"

4. Từ vấn đề trên ta đặt ra giải thiết rằng: "Ta có thể login vào user Carlos, khi có yêu cầu nhập security code thì mình bypass trực tiếp được không?"

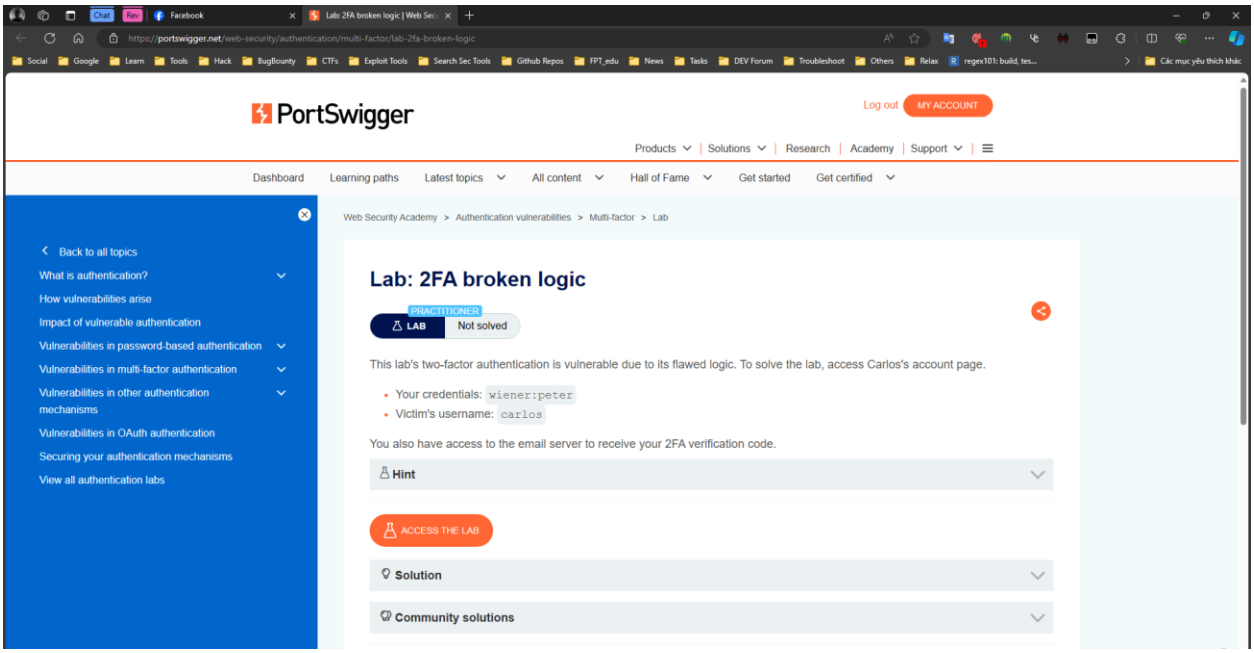
Để thử nghiệm thì giờ ta sẽ thử login bằng account của Carlos đã cho sẵn



⇨ Có thể thấy rằng khi bị yêu cầu nhập security code "**/login2**" để bỏ qua ta sẽ điều hướng trực tiếp đến "**/my-account**" mà không cần nhập mã code.

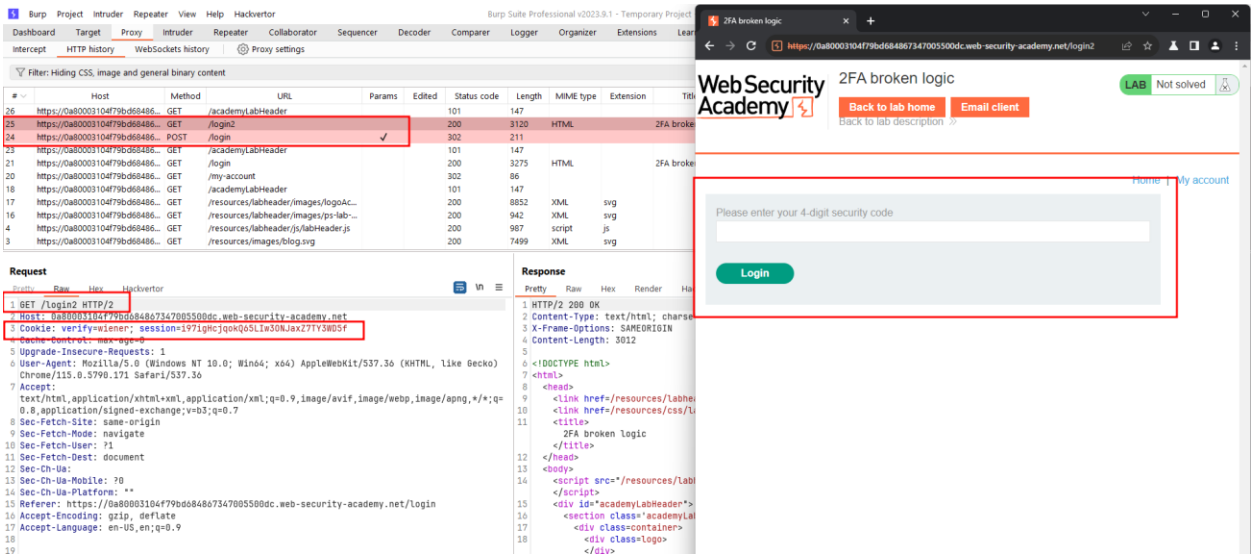
⇨ Vậy bài lab này đã được solve

II. 2FA broken logic <Here>



- ⇨ Mục tiêu của bài lab này tương tự như trên. Tuy nhiên thông tin mà bài lab cung cấp như sau:
  - Account dùng để test: **wiener:peter**
  - Account nạn nhân: **carlos**
  - Khác với bài lab trên là bài lab này không có password của nạn nhân

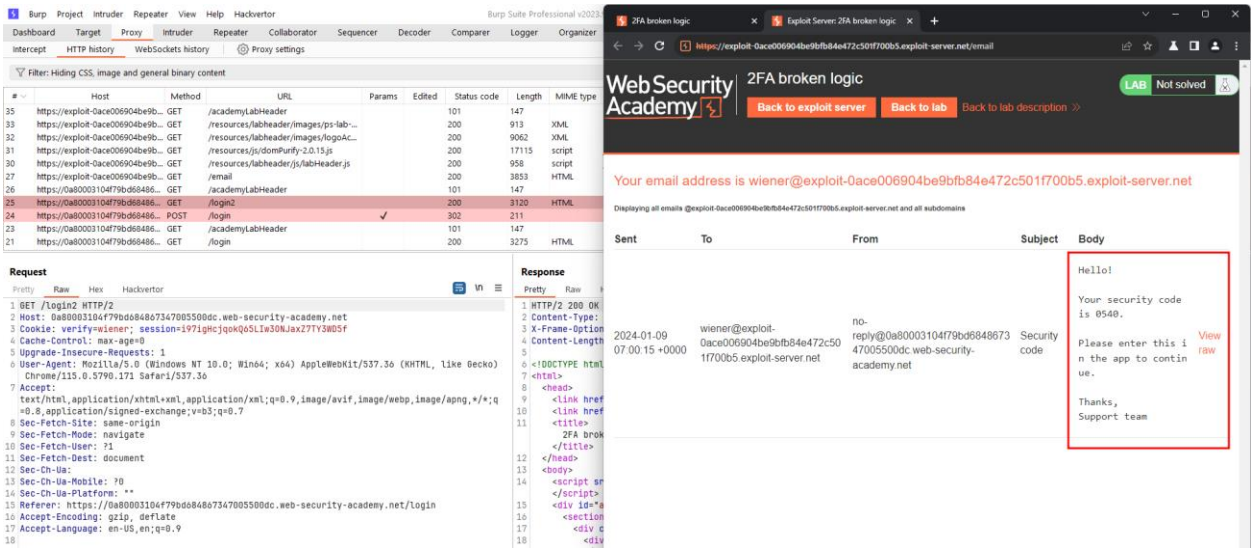
1. Đầu tiên ta truy cập vào bài lab, sau đó login bằng tài khoản test



- ⇨ Có thể thấy được thì quá trình login vào cũng tương tự như trên.

⇨ Tuy nhiên có thể thấy rằng ở đường dẫn "**/login2**", ở header **Cookie** thì có thêm một biến **verify** nữa.

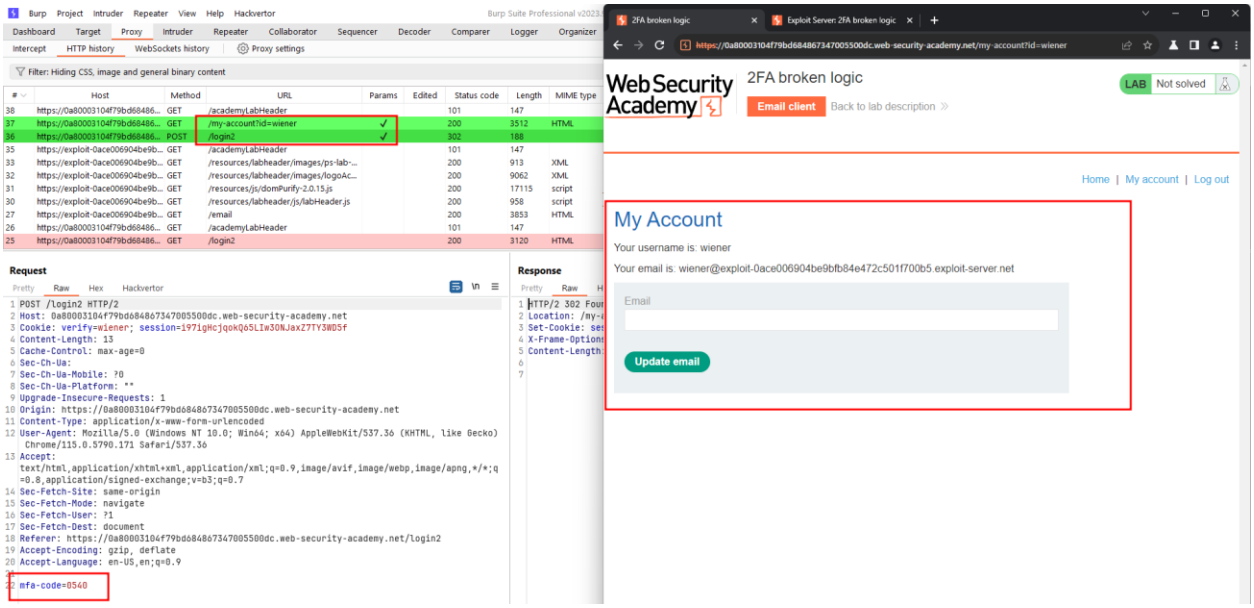
2. Tiếp theo bấm vào nút **Email client**



⇨ Có thể thấy thì cơ chế và quá trình cũng giống như bài lab trên thì trong đó cũng có security code

⇨ Giờ ta sẽ thử nhập security code đó để xem quá trình diễn ra như thế nào

3. Sau khi nhập security code



⇨ Sau khi nhập security code thì quá trình diễn ra vẫn giống như vậy

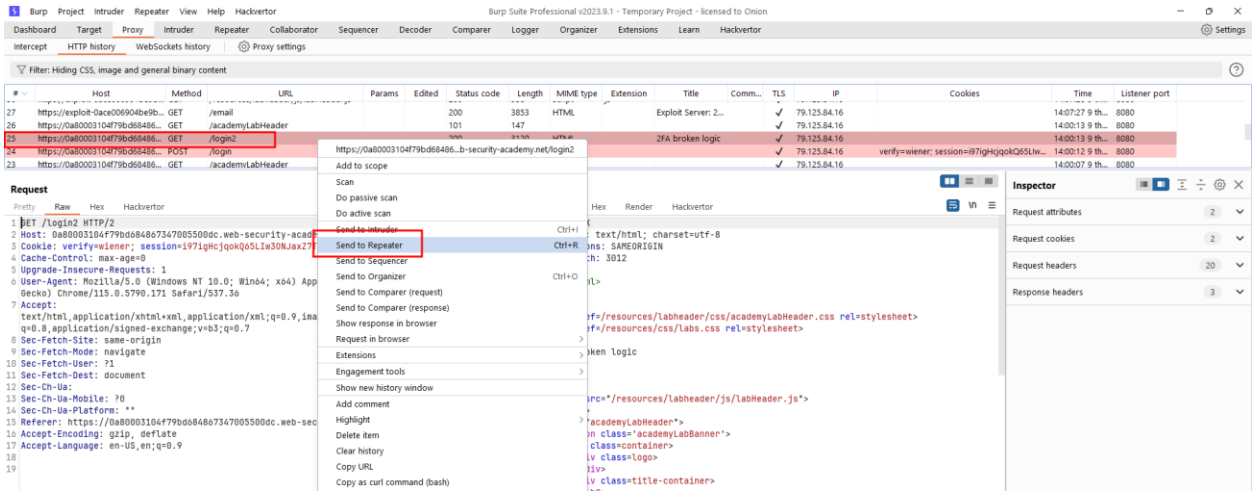
⇨ Cái khác ở đây đó là biến verify ở header Cookie.

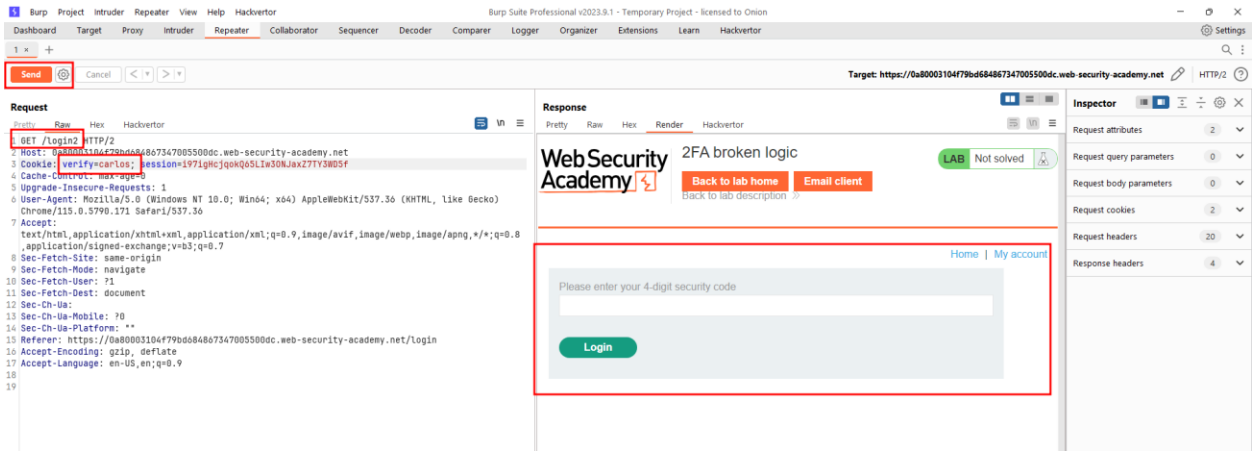
4. Không giống như bài lab trước đó là có password của nạn nhân còn bài này thì không. Ta tự đặt ra giải thiết rằng: "*Liệu ta có thể bỏ qua quá trình nhập password mà đi thẳng đến quá trình nhập security code không?*"

   Từ request có method POST trước đó, ta đưa ra một phép thử rằng: "*Nếu ta thay giá trị của biến verify thành carlos và từ đây brute-force mfa-code thì có khả thi không?*"

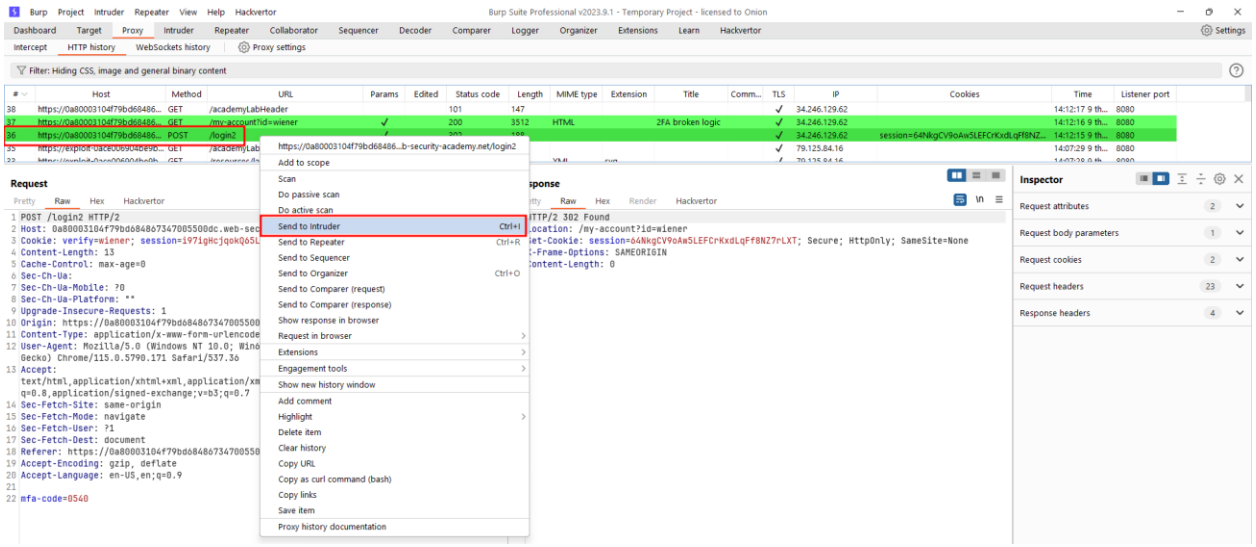
   Note: Để đảm bảo thì trước tiên phải logout tài khoản trước đó ra để đảm bảo session không còn được sử dụng

   Tiếp theo ta sẽ gửi request "**GET /login2**" đến **Repeater**. Sau đó sửa giá trị của **verify** thành **carlos** rồi bấm **send**. Điều này đảm bảo rằng sẽ có **mfa-code** được tạo riêng cho user **carlos**

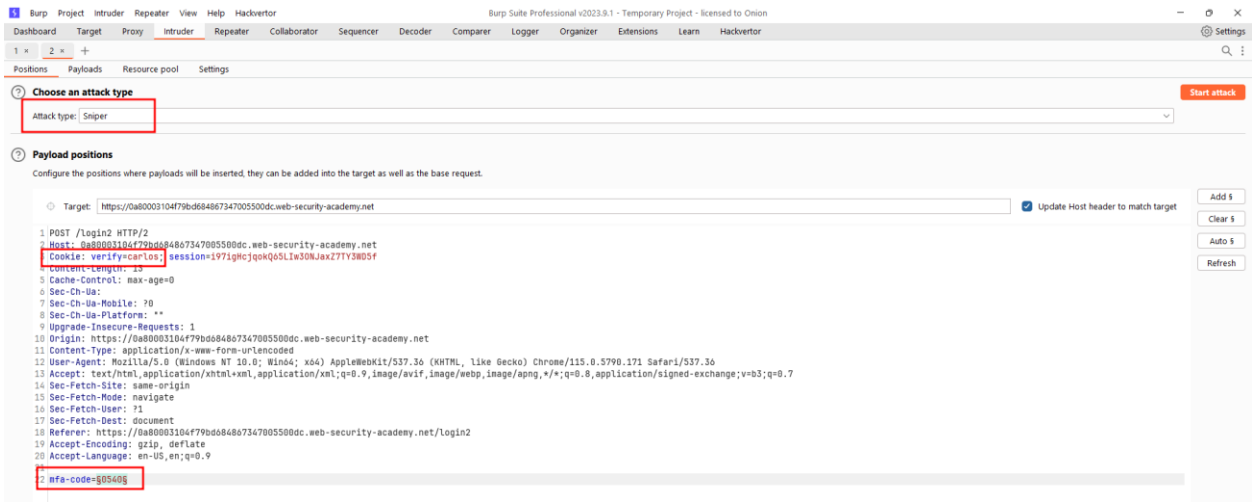




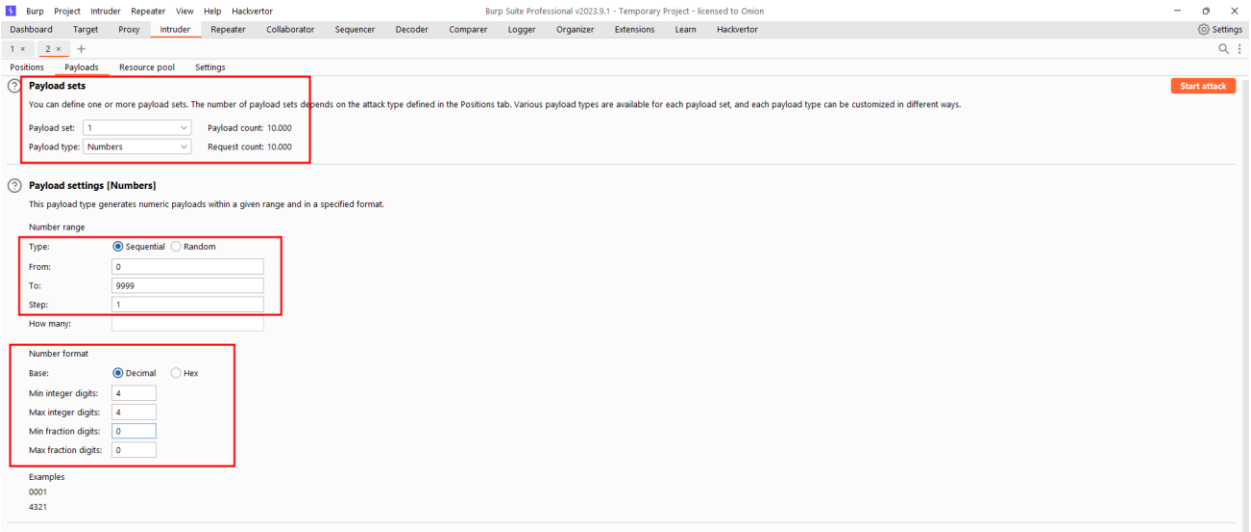
Sau đó quay lại **Proxy** gửi request "**POST /login2**" đến **Intruder**



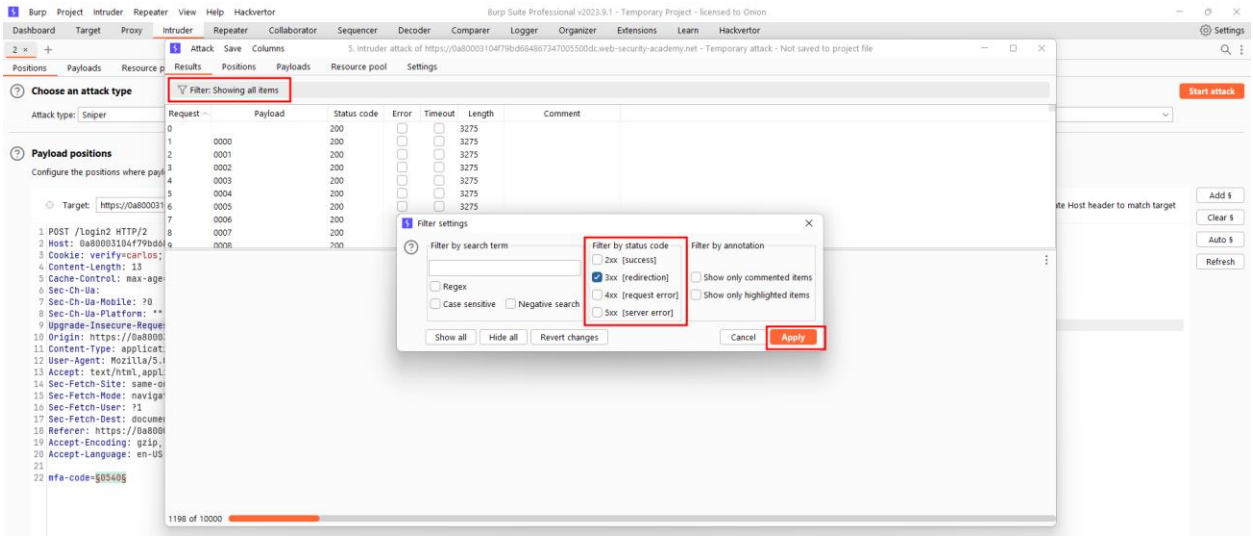
Ta sửa thông tin như sau:



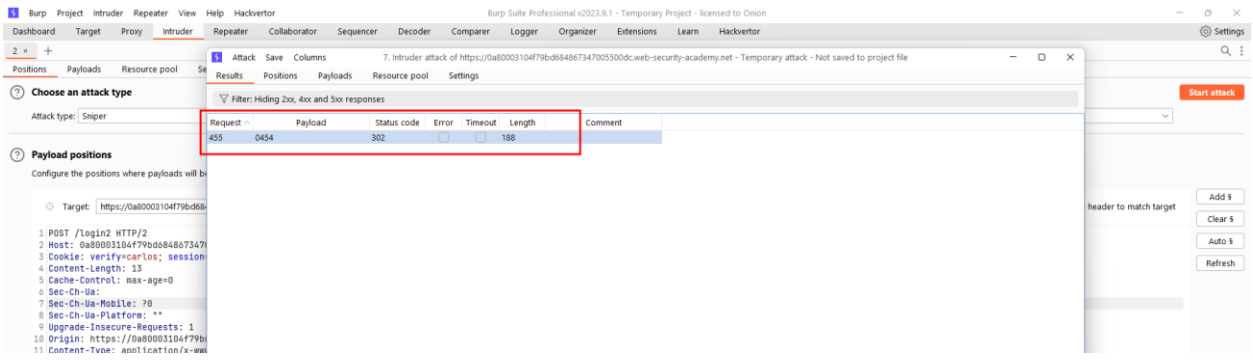
Về Payloads, ta cấu hình như sau



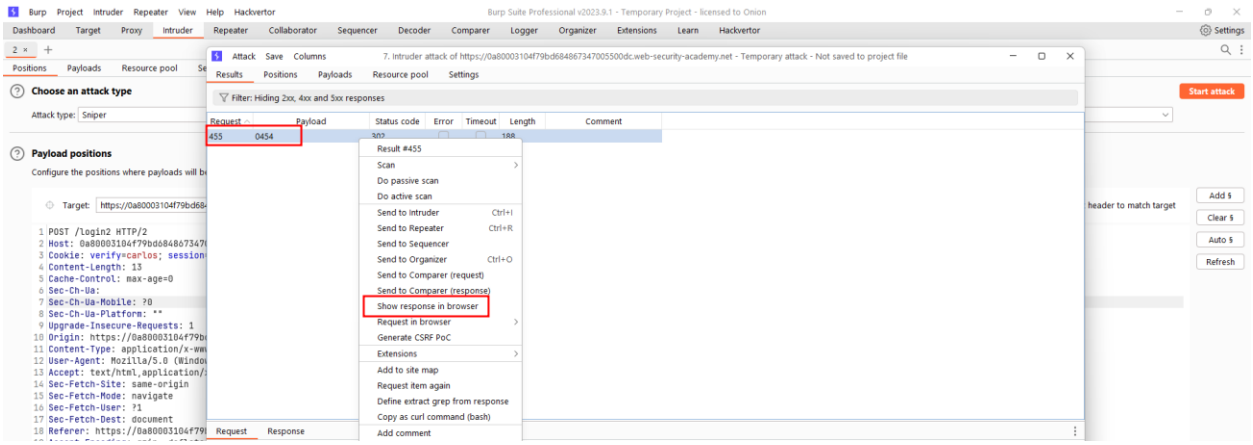
Sau đó **Start attack**, khi xong thì tìm request có HTTP **Status code** là **302**. Để dễ dàng hơn ta sẽ áp dụng filter



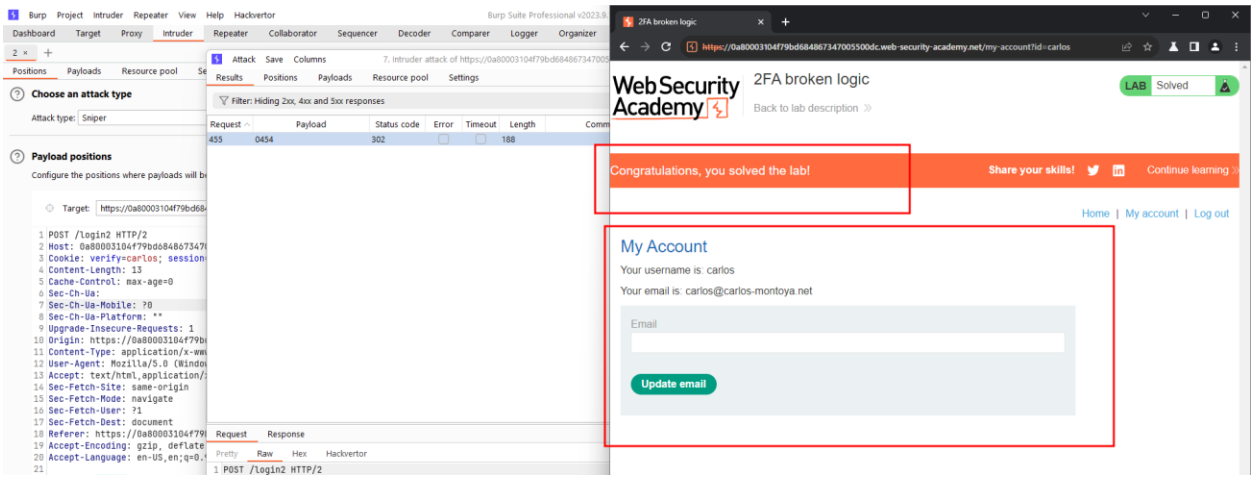
Sau khi chờ được một lúc thì ta đã tìm được code



Giờ ta sẽ truy cập vào trang web bằng session bằng cách



Sau đó dán vào trình duyệt



⇨ Sau khi dán vào thì trình duyệt đã chuyển hướng đến đường dẫn **/my-account**, đã solve được lab.