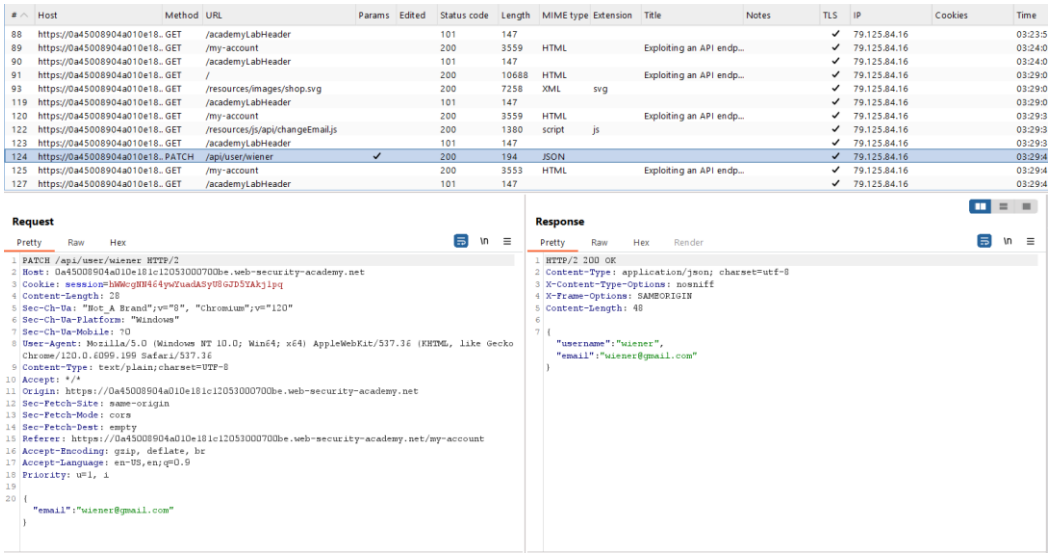
API Testing

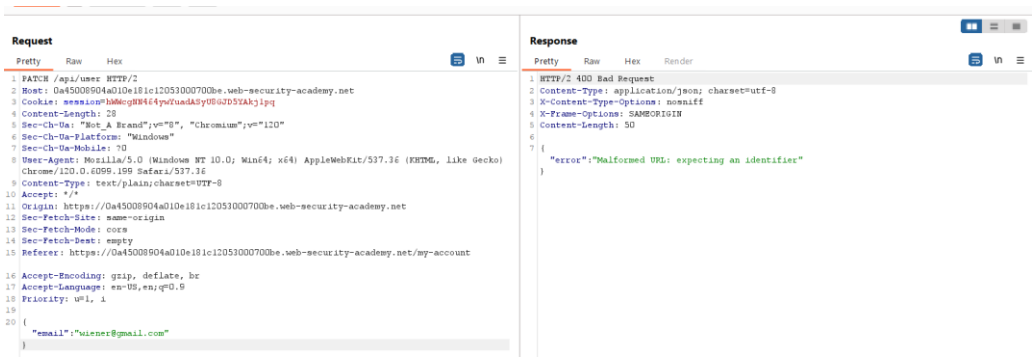
LAB 105 Exploiting an API endpoint using documentation

Valid credentials -- wiener:peter;

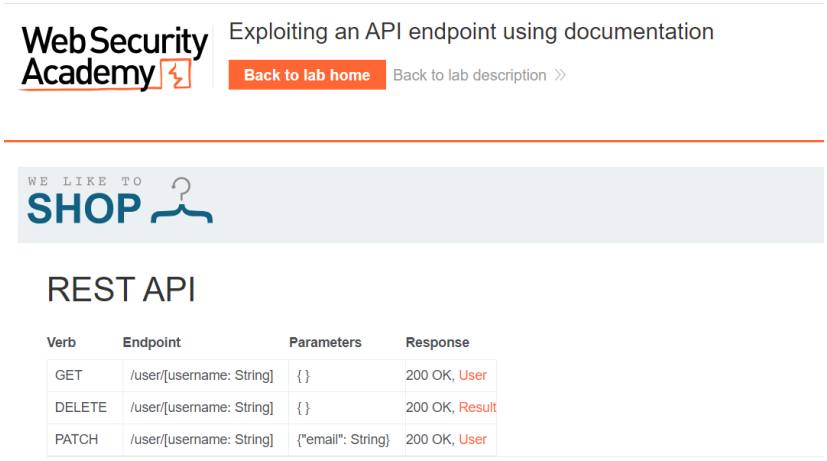
I have tested the application functions and discovered a PATCH /api/user/wiener method during email change procedure:



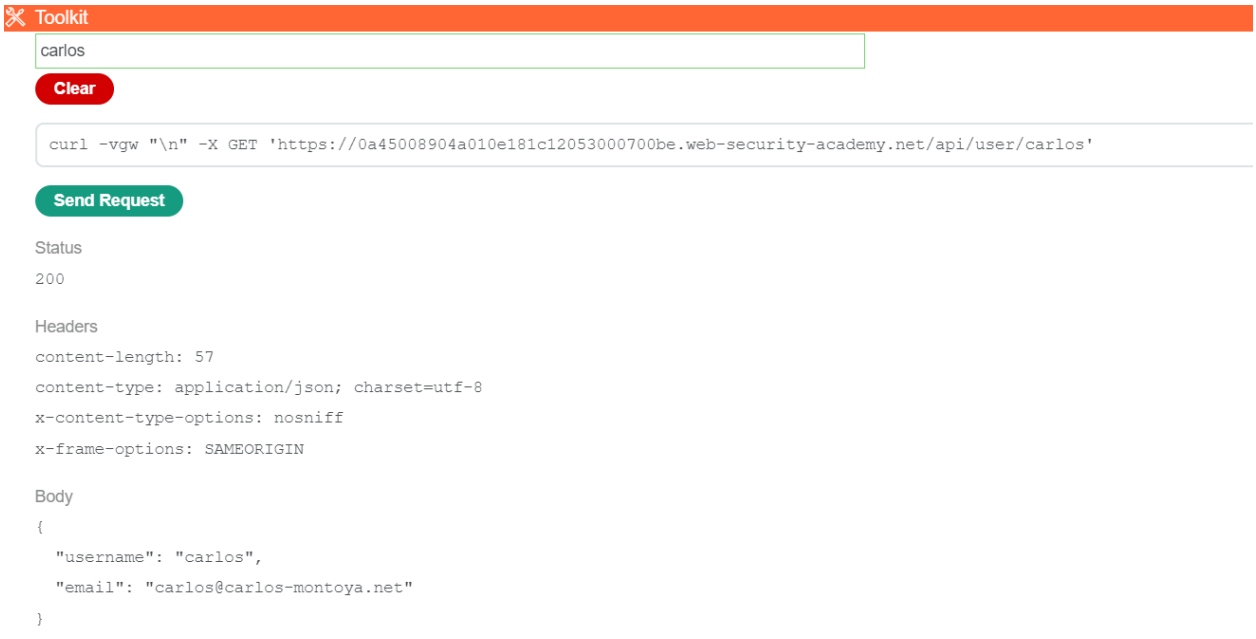
I have sent it to Burp Repeater and tried to traverse the API:



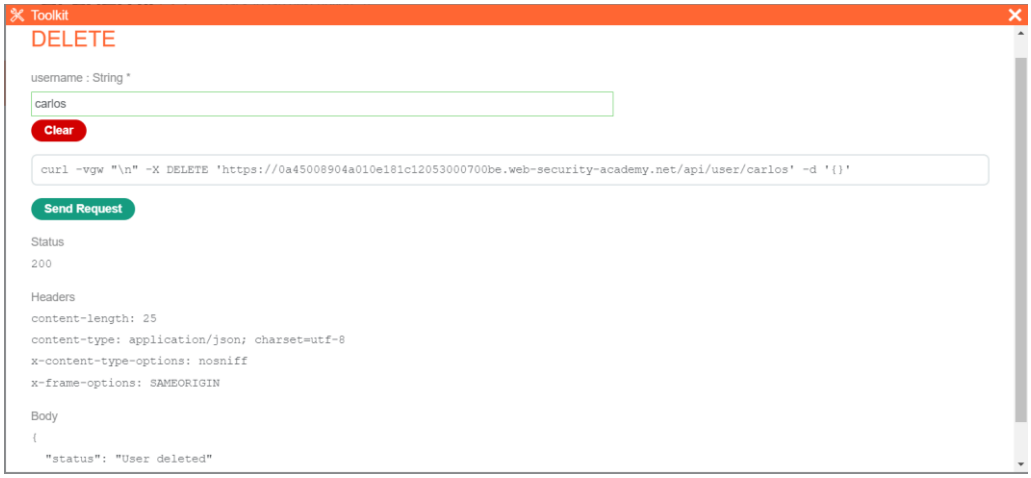
At /api/user, the error message “Malformed URL: expecting an identifier” is shown. I traversed further to /api and saw a positive response. It leads to API’s interactive GUI with options: GET, PATCH, DELETE



I have tested all three methods and discovered that user ‘carlos’ is indeed present in the system. Email: carlos@carlos-montoya.net.



Then, I deleted ‘carlos’:

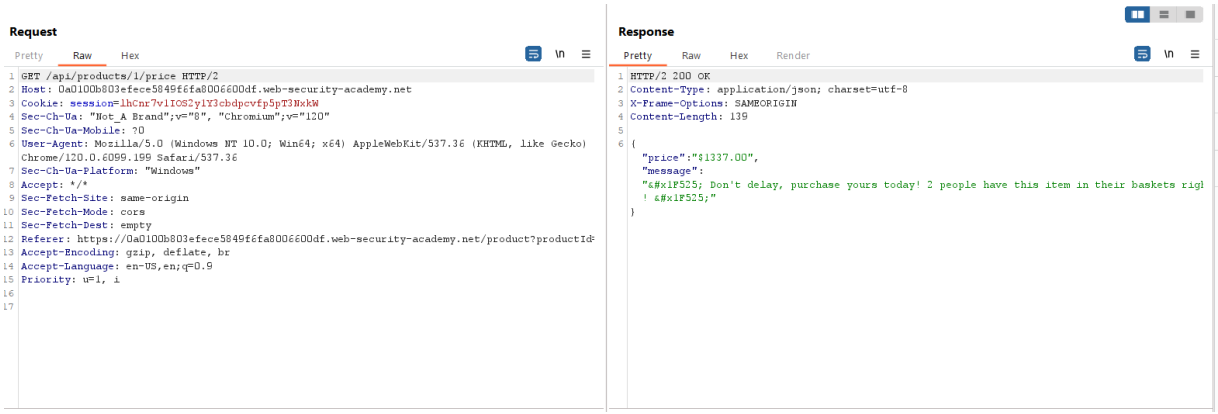


Congratulations, you solved the lab!

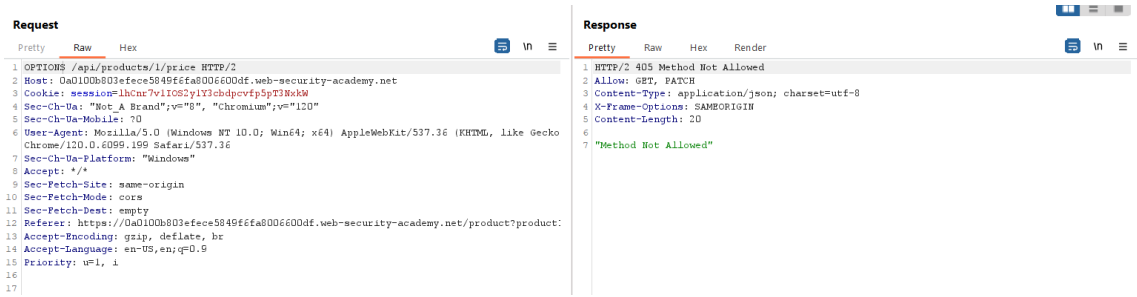
LAB 106 [Finding and exploiting an unused API endpoint]

Valid credentials – wiener:peter

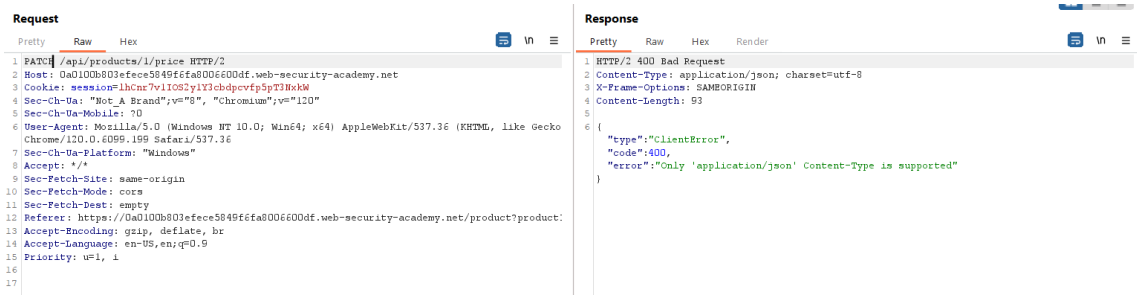
This time, the lab contains a functionality of adding a product to a cart. To display product details, the service uses API:



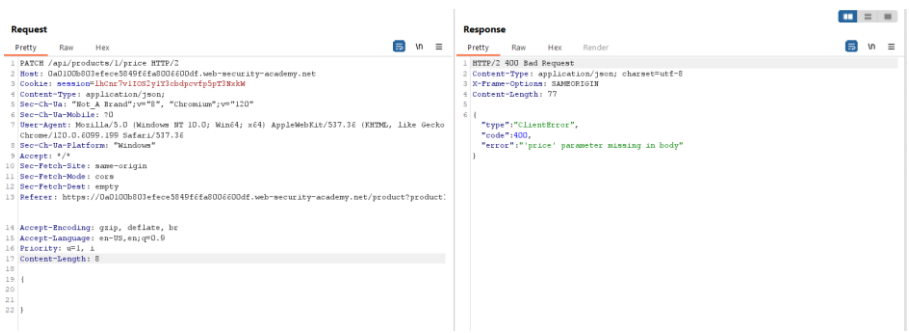
Attempt to change the method from GET to OPTIONS lead to an error and displays allowed methods: GET, PATCH.



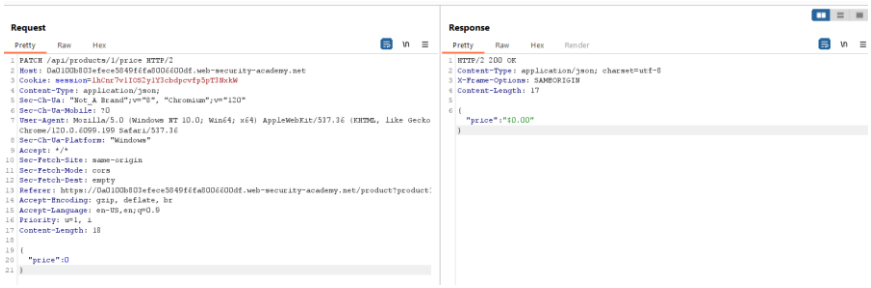
Let's change method to PATCH:



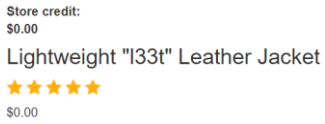
Error says, that Content-Type: application/json only allowed, so let's add content type to the request body and make an empty json:



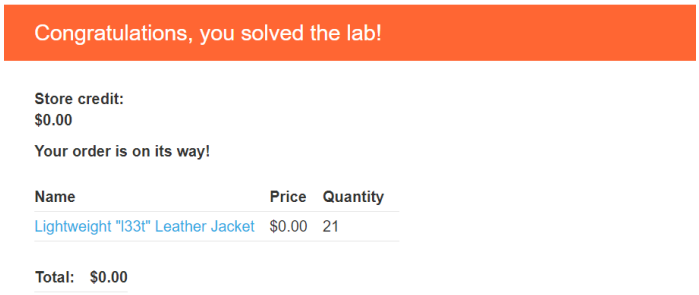
Now, it yells at price parameter missing in json body, so I can fix it by adding “price”: 0 and try to make the price $0 for the expensive jacket:



As one can see, the request was processed successfully and I did change the price for the jacket in a store:



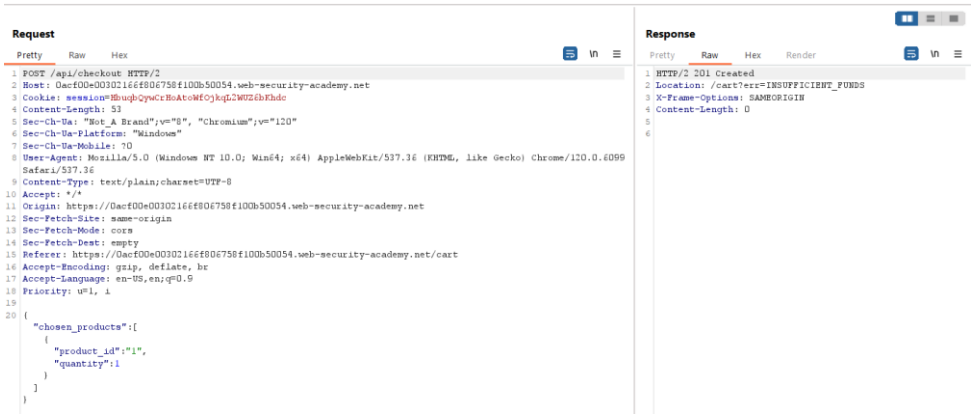
Now, add $0 jacket to the cart and check out:

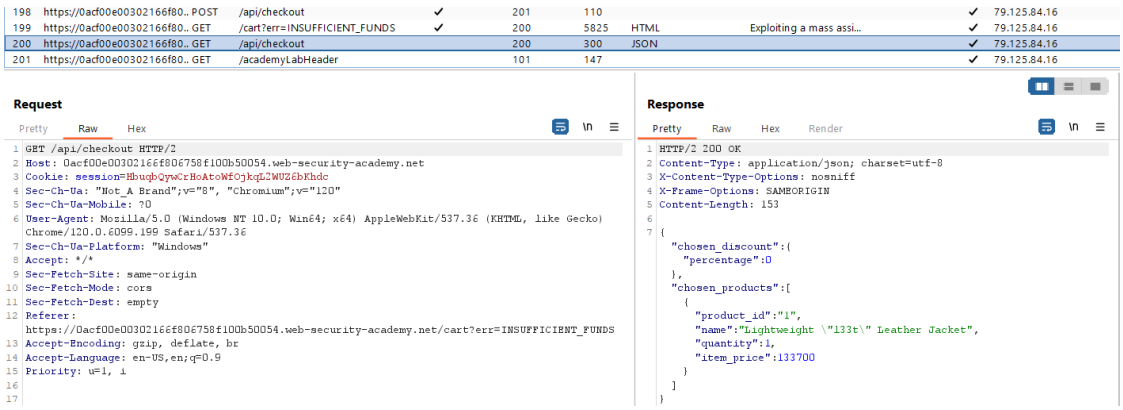


LAB 107 Exploiting a mass assignment vulnerability

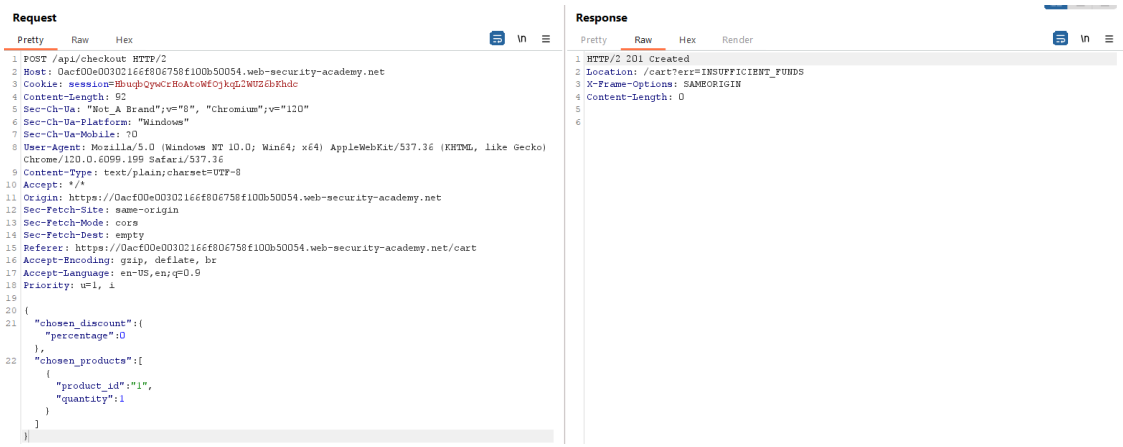
Valid credentials – wiener:peter.

This version contains a functionality for entering discount coupons. During checkout, it uses API, here’re the contents of POST /api/checkout and GET /api/checkout:

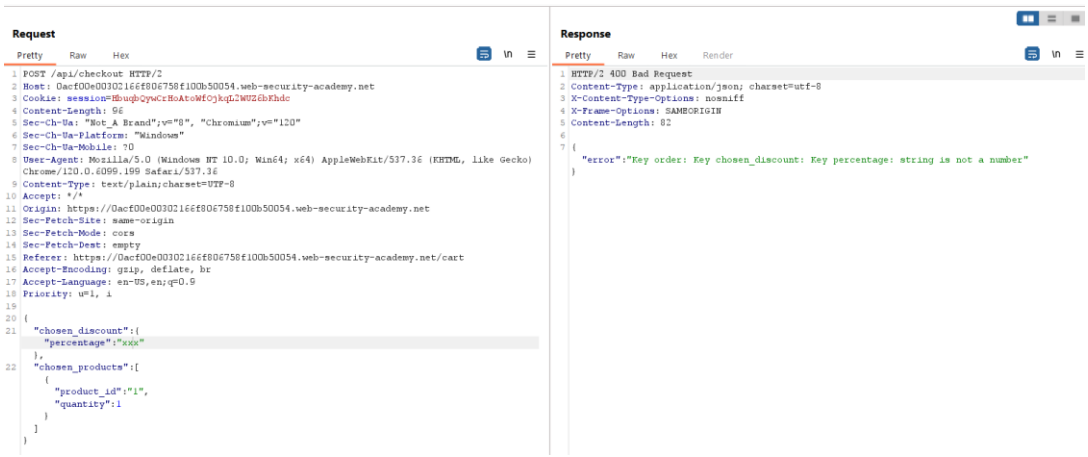




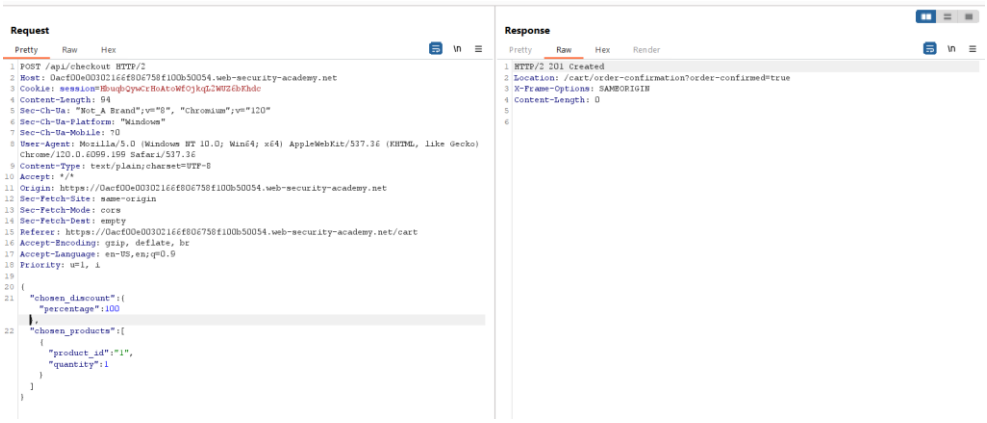
In GET method, I've noticed, that "chosen_discount": ["percentage":0] function is applied, So I added it to the POST request and send it again, using Burp Repeater:



There is no error present, which is a good sign and may look like the applications resolves such functions in json bodies of POST requests. Let's confirm it by using an invalid value "x":



The assumption was correct and the parameter is resolved, now, I will change the value to 100, meaning that I should get a 100% discount:



Successful, the product was successfully checked out!

Congratulations, you solved the lab!