PENETRATION TESTING REPORT FOR MUTHENGIA UNIVERSITY

Compiled by:

FEDNAND MWASHAMBA MTALAKI
AFRICAHACKON ACADEMY
Cohort 1: Squad 9

CONFIDENTIAL

Date: 3rd May 2025

TABLE OF CONTENT

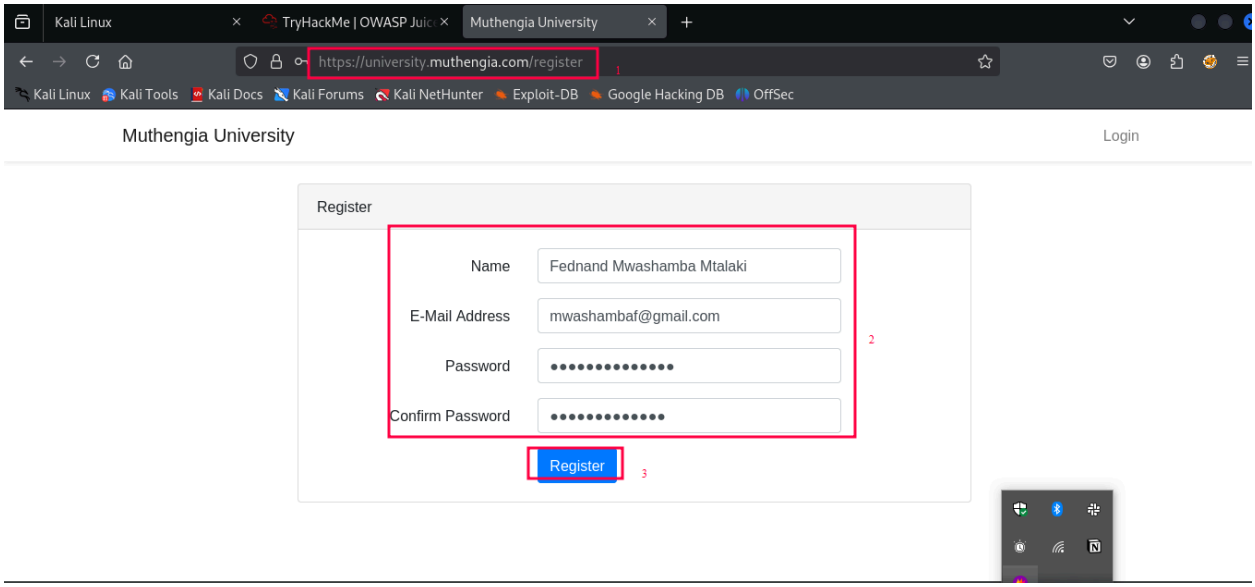
EXECUTIVE SUMMARY

The Penetration Testing report underpins the fundamental vulnerabilities that the website of Muthengia University is exposed to. We will provide the methodology, the findings, recommendations and final our conclusion.

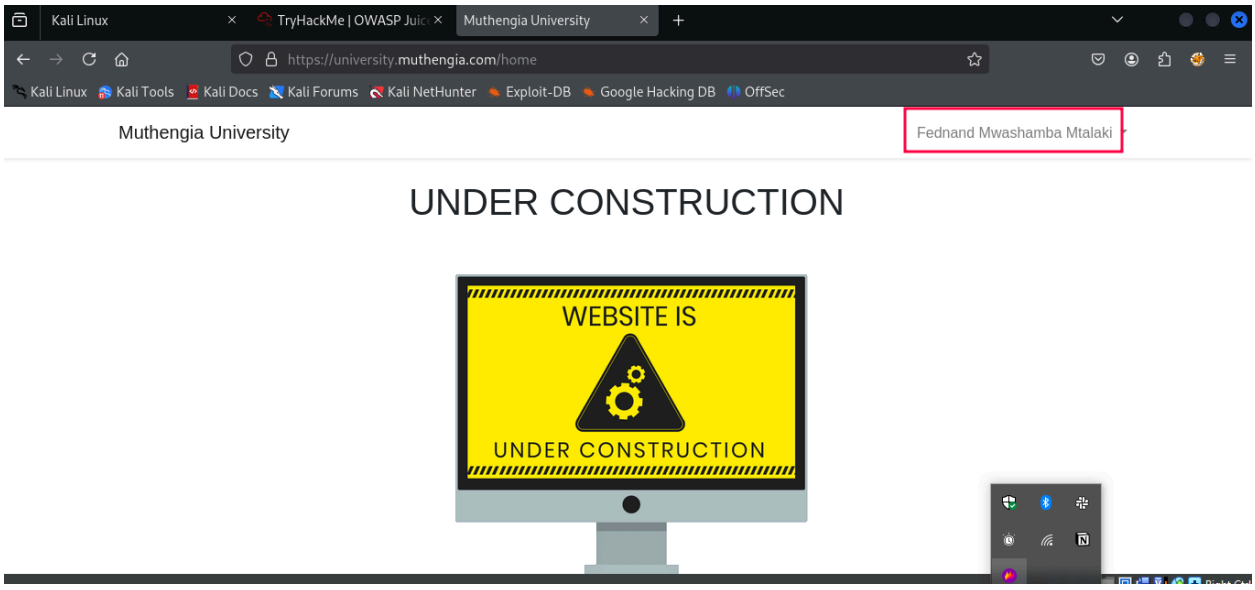
METHODOLOGY

Step 1: USER REGISTRATION

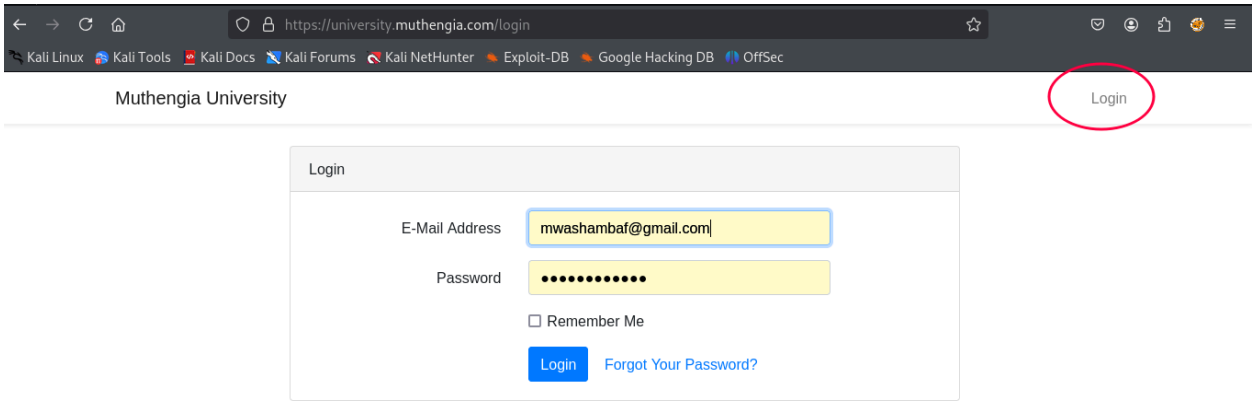
While undertaking this process I made sure my proxy is on, as well as Burp Suite to intercept the traffic



Successfully registered as a user



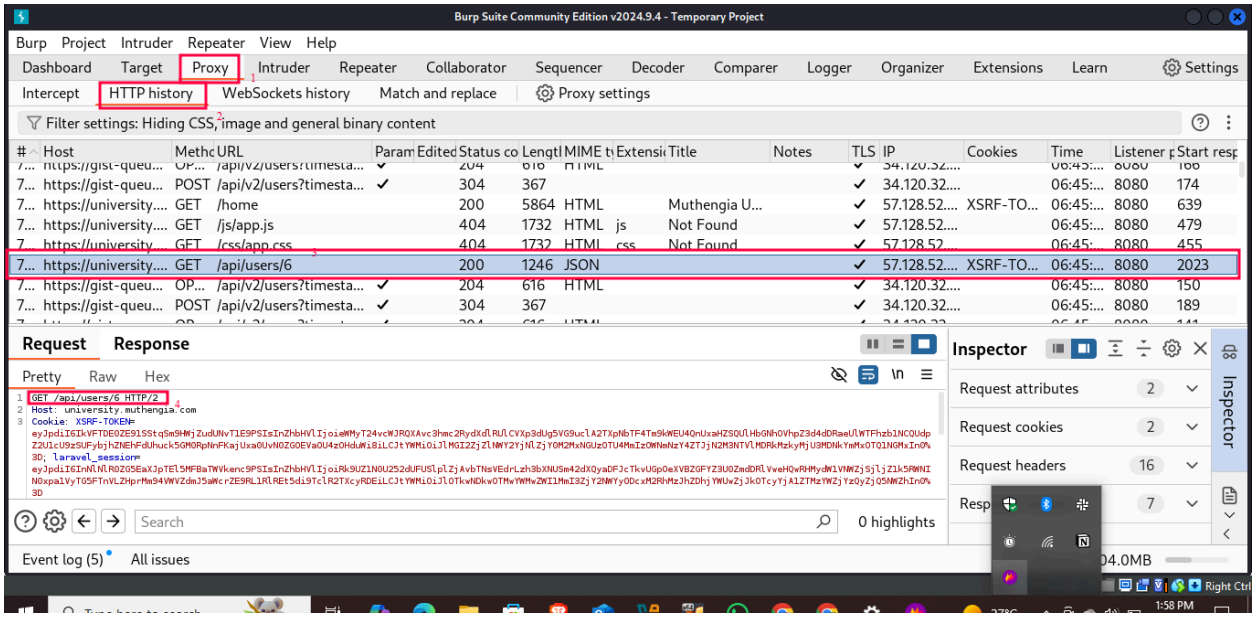
Login:



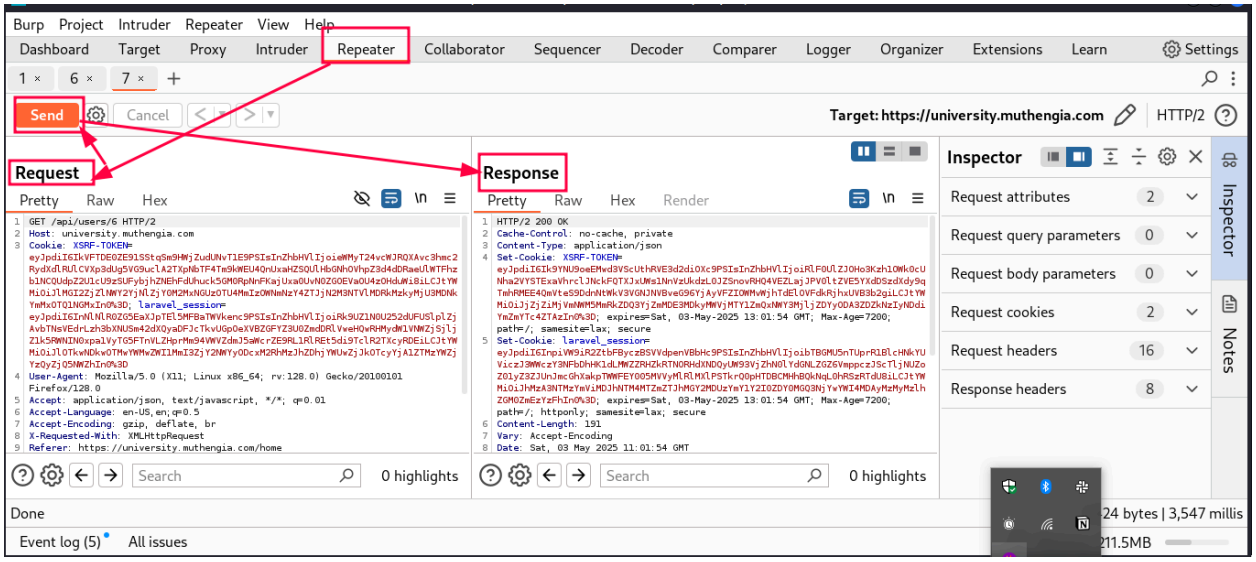
Step 2: TRAFFIC INTERCEPTION

With Burp Suite on, I was able to capture quite a number of traffic

However, my focus was on *Get request calling for API/Users/*

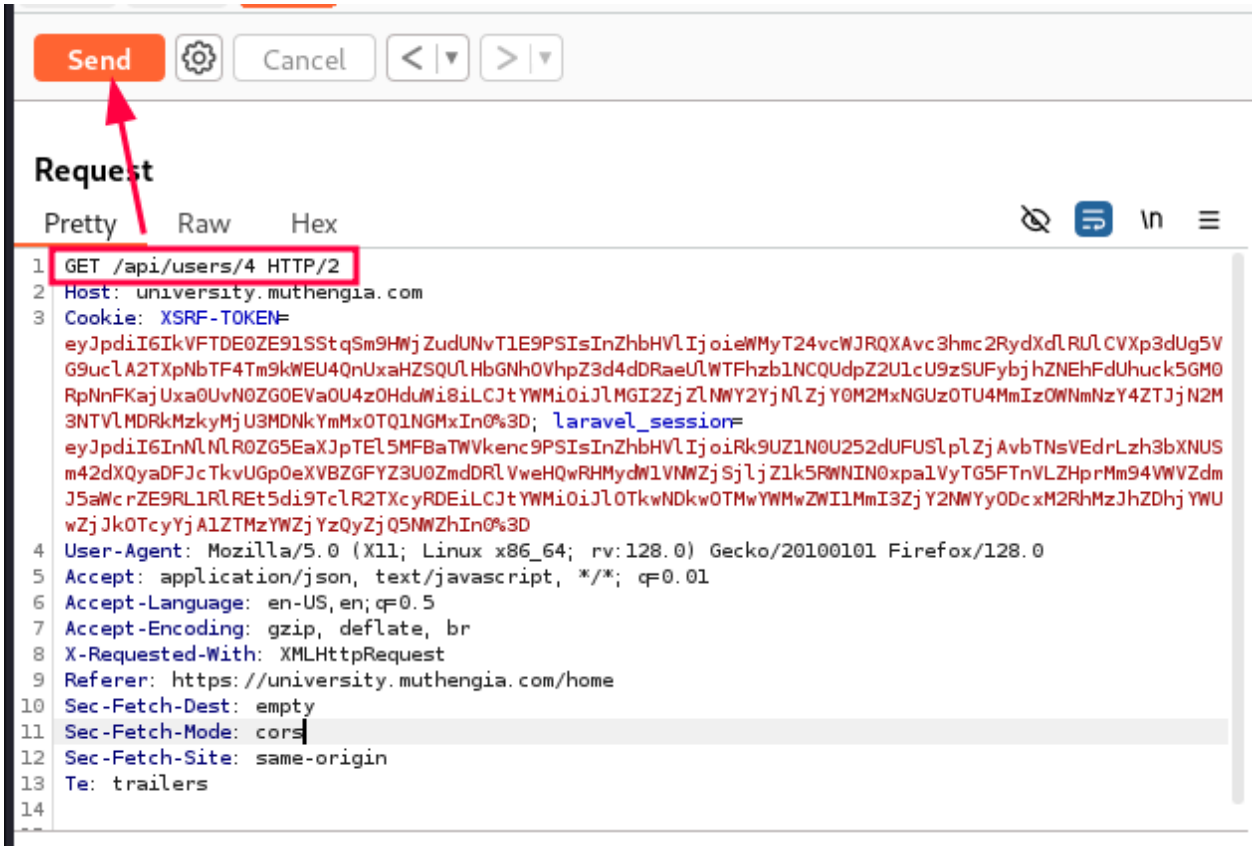


I then sent the Get request (*GET /api/users/6 HTTP/2*) to the **repeater** for manipulation. I got a response 200ok meaning successful request

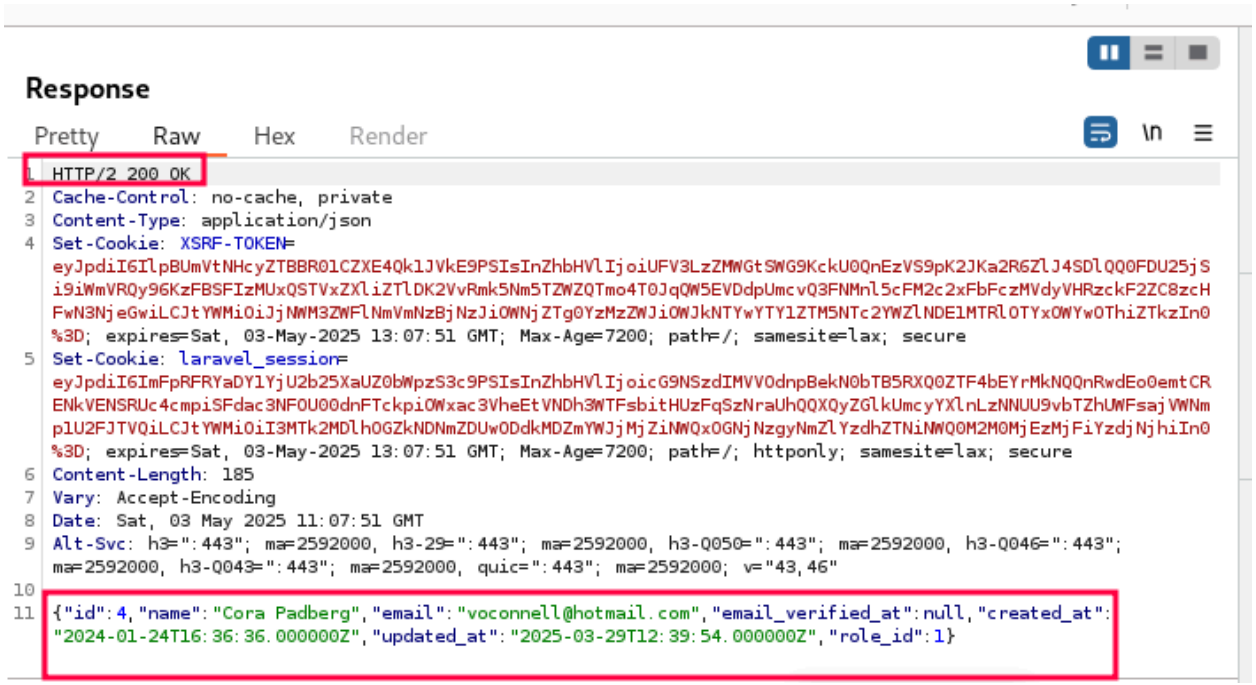


From the response, I was able to view user 6 information

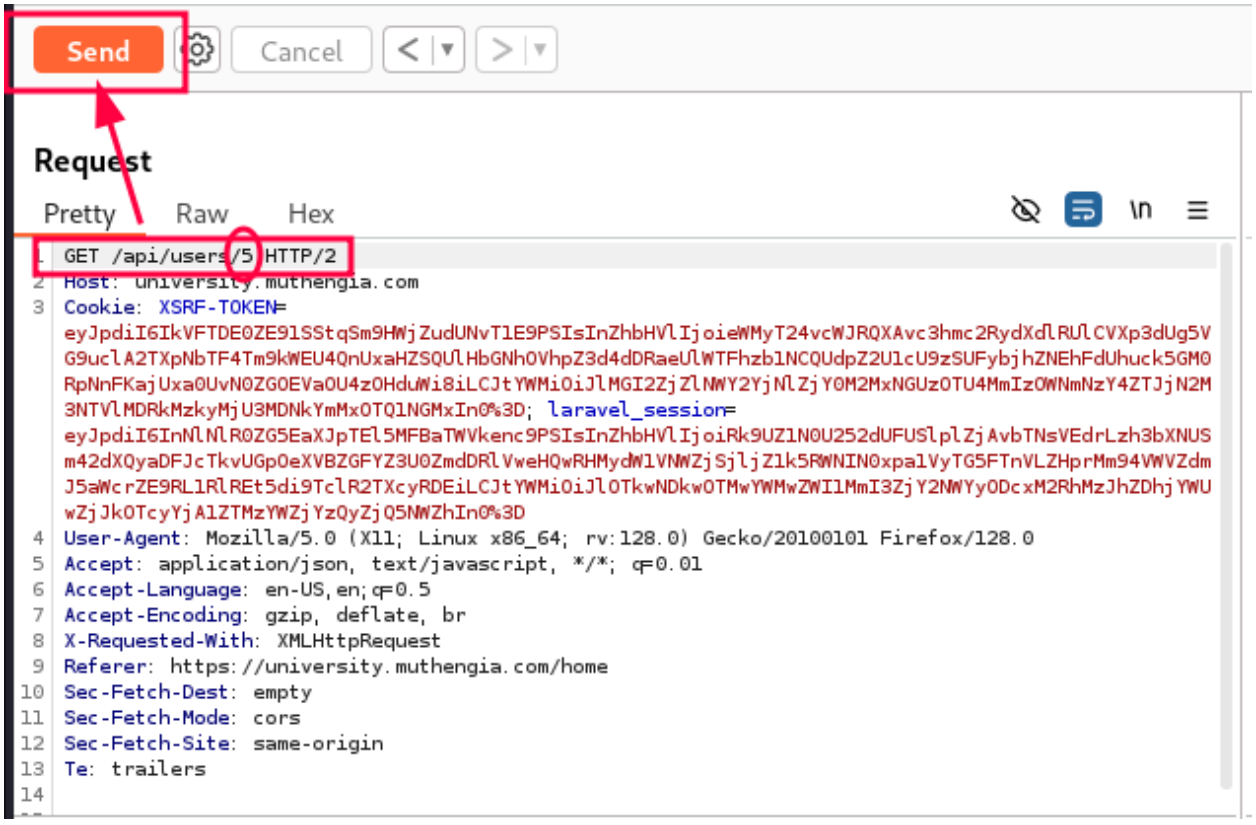
From the repeater, I edited the request; users to users/4



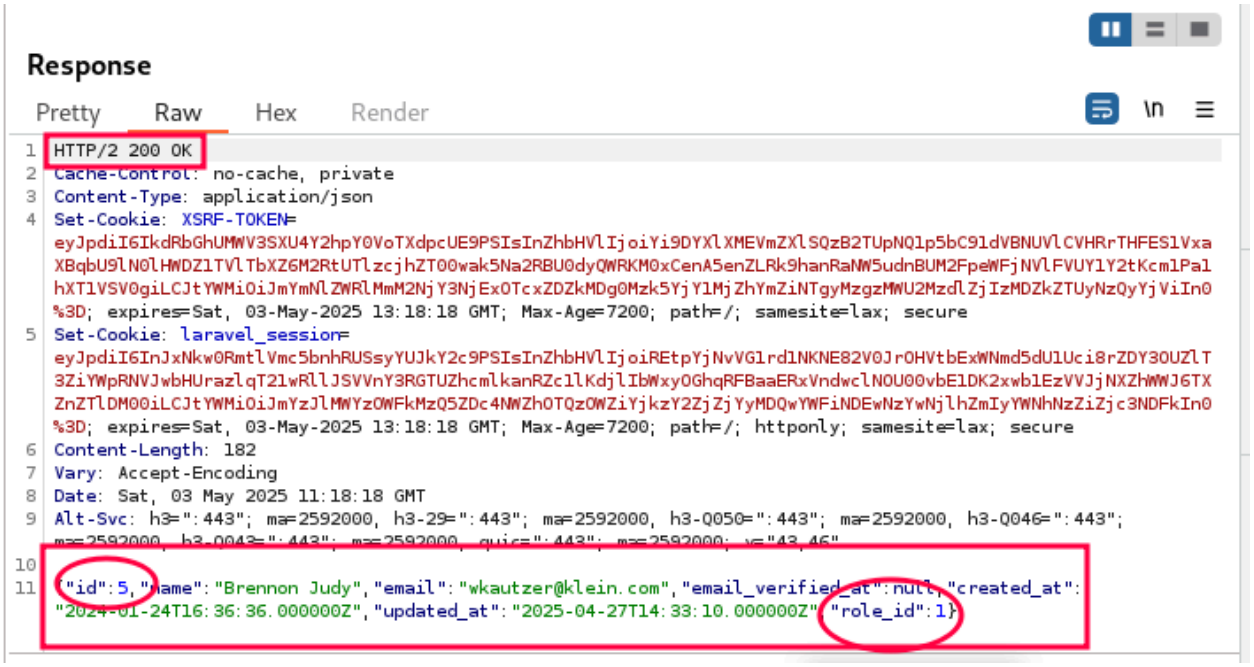
Then sent it, received the response below for user 4 endpoints



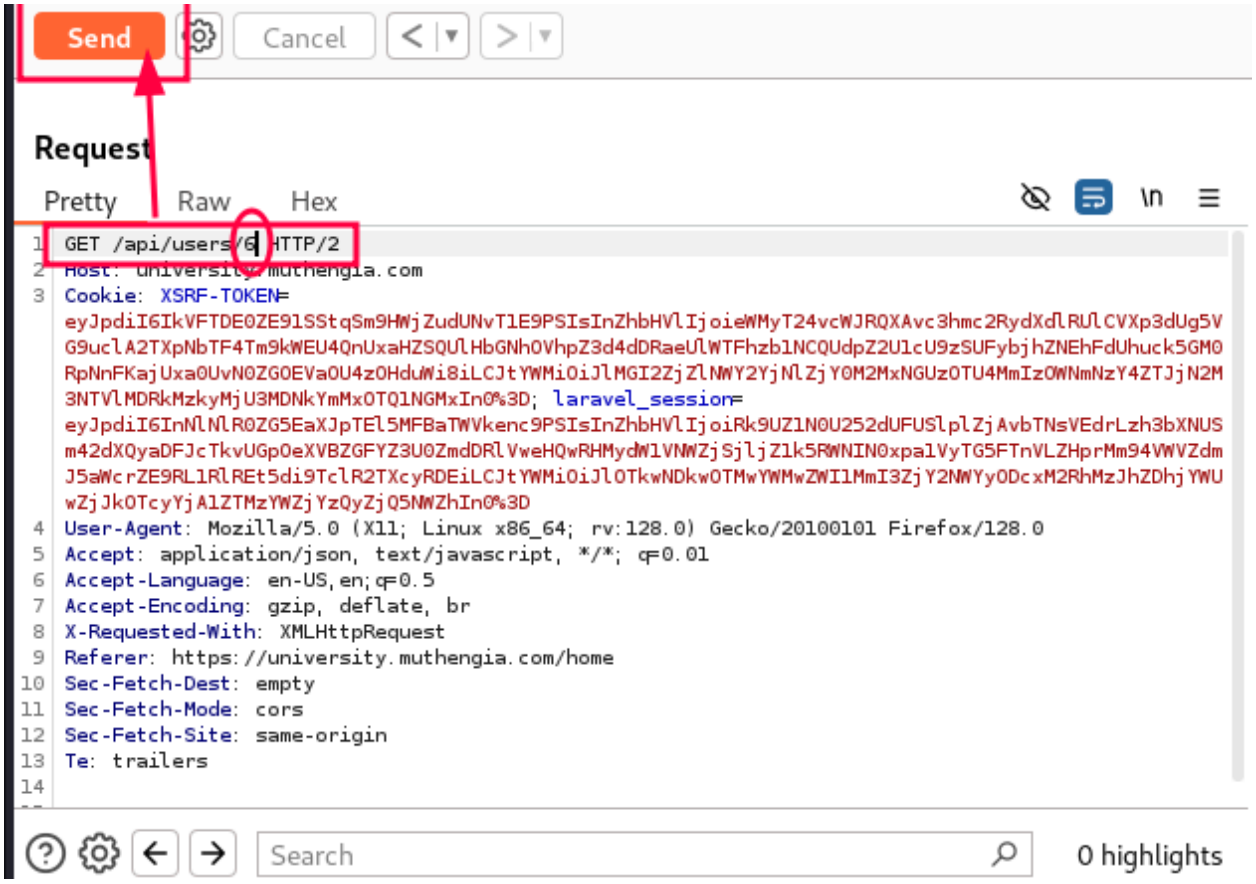
Followed the same steps to get user 5 output



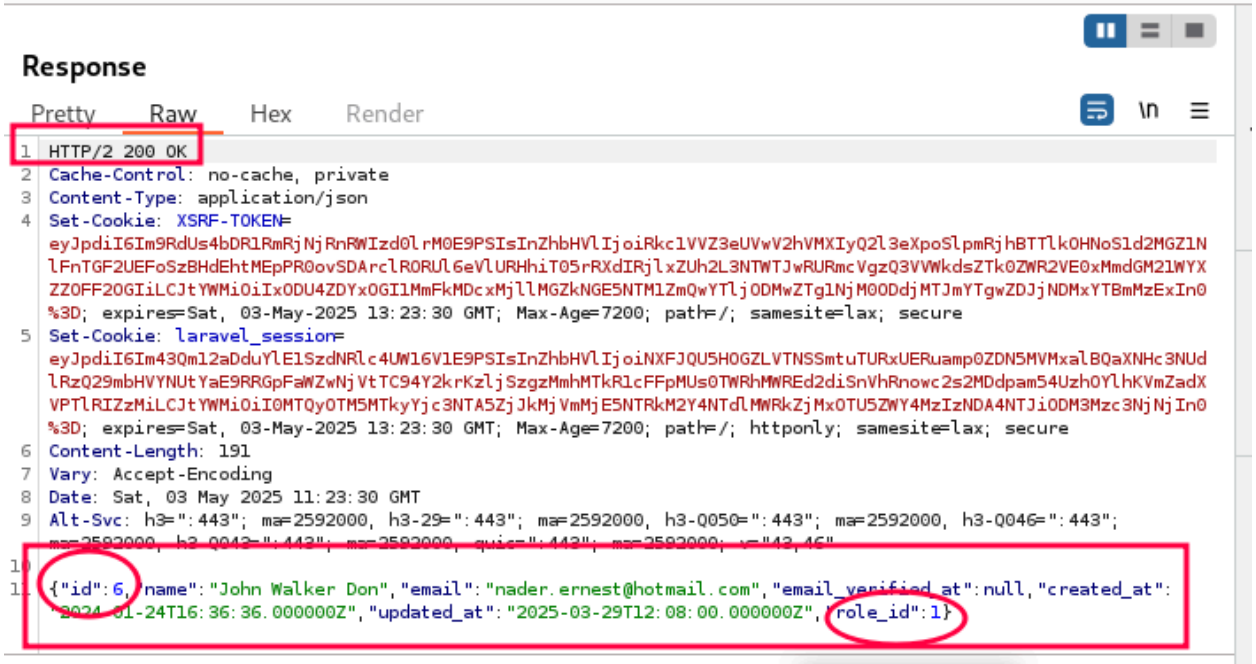
Got the response; 200ok a successful request



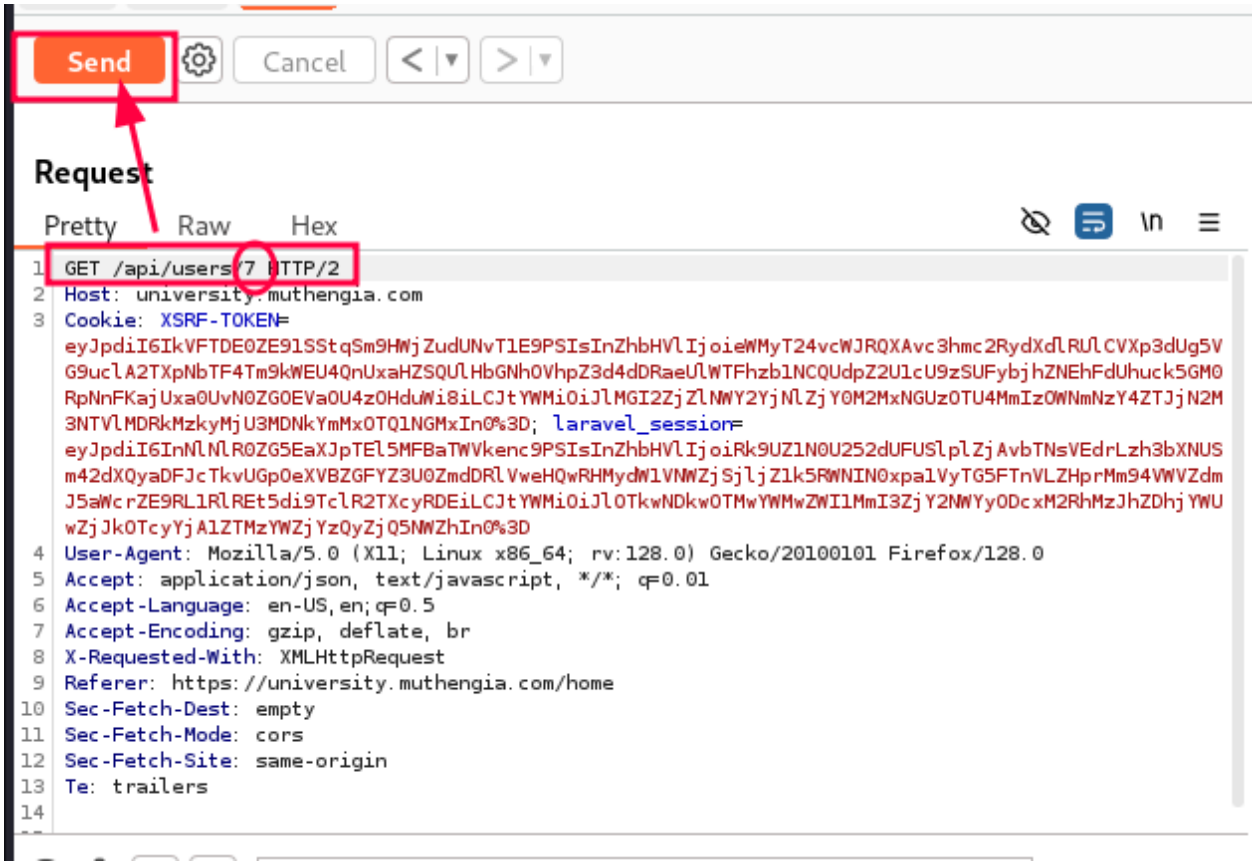
User 6; Request



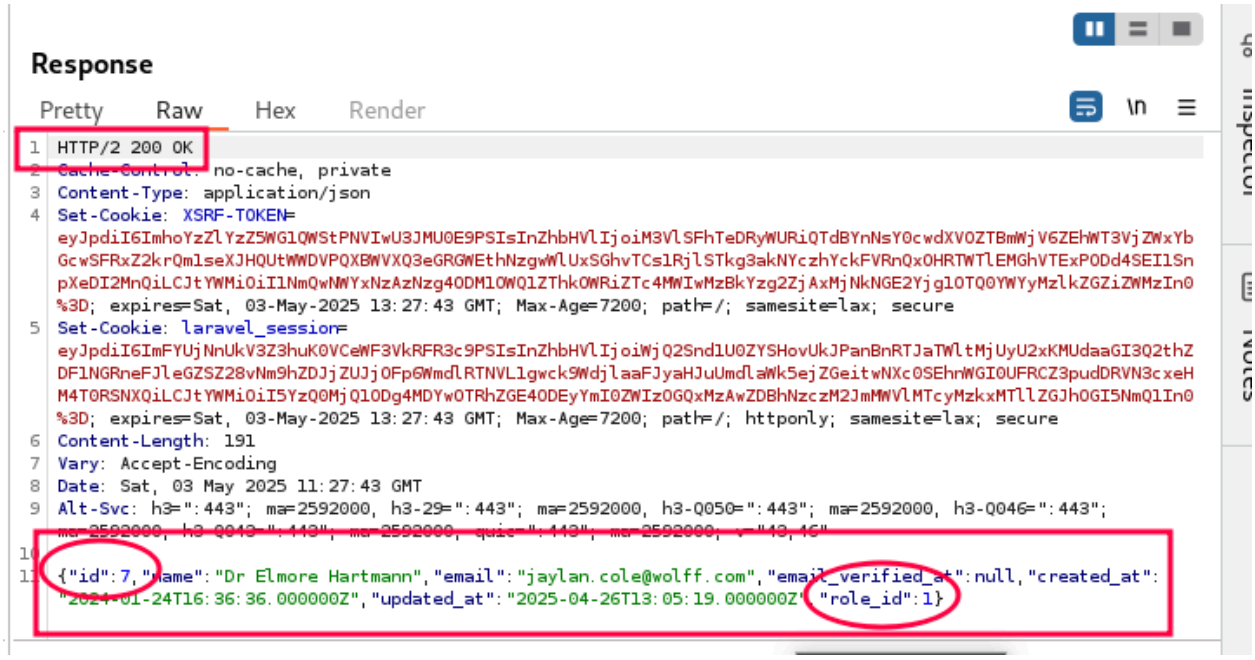
Response



User 7: Request



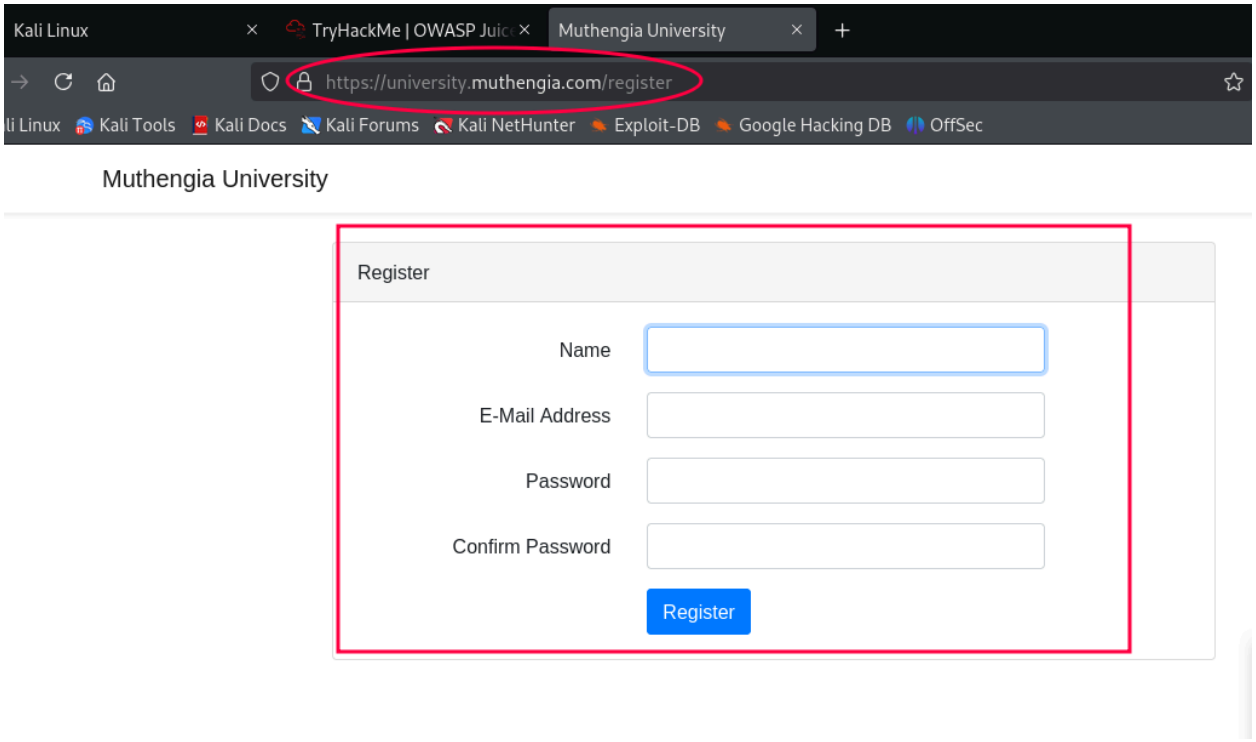
Response



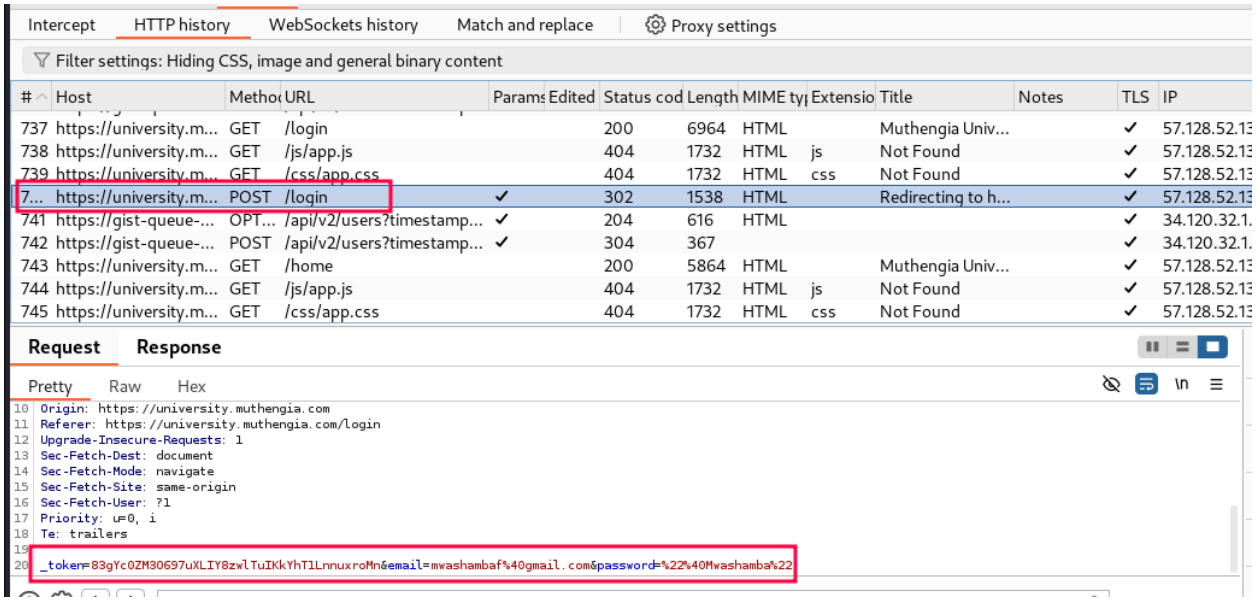
FINDINGS

The following are my findings;

1. The login page is vulnerable to exploitation from users through endpoint manipulation, since the site does not sanitize user input; allows new registrations



2. The site is parsing *sensitive data* in plaintext upon registration. This I find out from the Post request I intercepted and analysed



3. The site allows for *Broken Access Control*, where one can easily manipulate the requests and influence the output This I find out from my ability to manipulate the different API/users and obtain their information

RECOMMENDATIONS

Following the above findings, I recommend the following;

1. Proper input sanitization. The developers should put checks and balances, ensuring a user cannot directly call an api endpoint from the url

2. All the sensitive data should be encrypted both in transit and at rest.
    - ❖ Use of security headers; include security headers to prevent data leakages
    - ❖ Data encryption
    - ❖ Hashing of Passwords; not being stored in plaintext
3. Prevent the broken access control by putting the following into place:
    - Deny by default; allow not users access api endpoints unless explicitly granted the permission
    - Avoid Insecure Direct Object References; not exposing predictable identifiers of the endpoints, instead tie the data into the authenticated users
    - User stronger session management; use role based access to control this and enforce rate limits and time outs for the session tokens.

CONCLUSION

Using OWASP Risk Rating Method

Vulnerability	Likelihood	Impact	Risk
Sensitive Data Exposure	High	Critical	Critical
Broken Access Control	High	High	Critical
IDOR	High	High	High

The findings in this report are critical, which can lead to enormous loss of data, non compliances to the legal requirements, prone to serious attacks such as ransomwares and Cross site scripting. Therefore it is advised that Muthengia University takes action as soon as possible, since the impact in the event of an attack will be huge and costly.