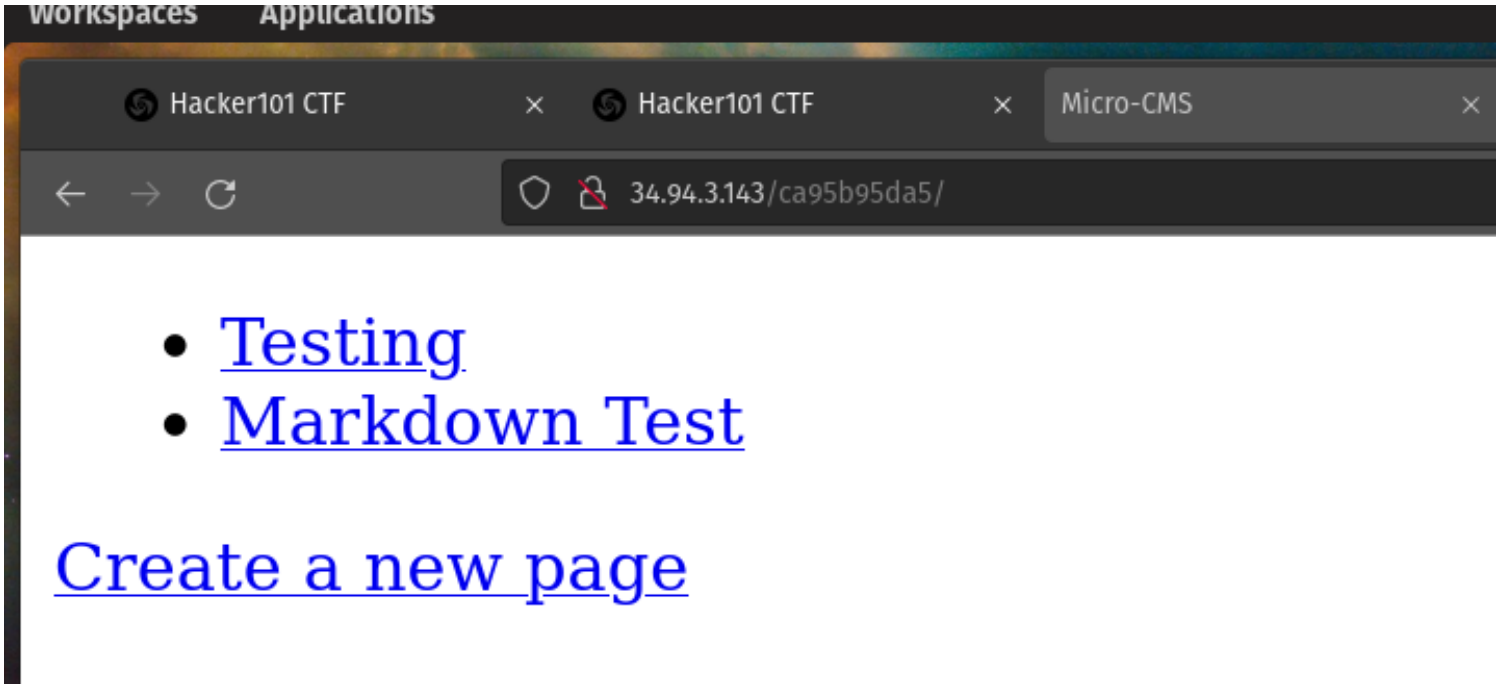
Hacker101-Level1

It is a page where I will improve my skills on web application security as it allows me to practice in a very easy way vulnerabilities such as XSS, SQLi, XXE, etc.

In this write up I will show the technical process I executed to complete level 1 (Micro-CMS).

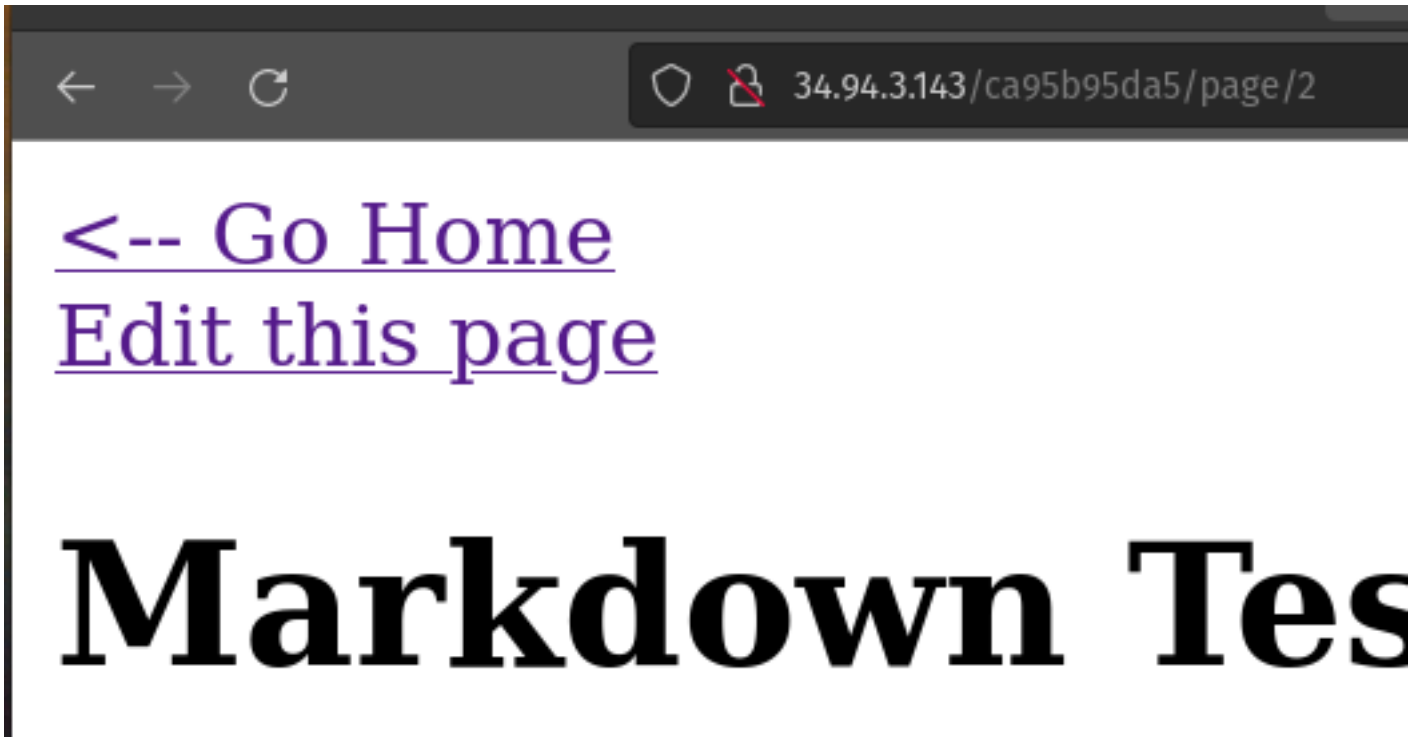


When entering level 1 I can see that this is a page a little more elaborate and we can perform some functions, something basic in web pages is to understand how it works so we should feel free to experiment.

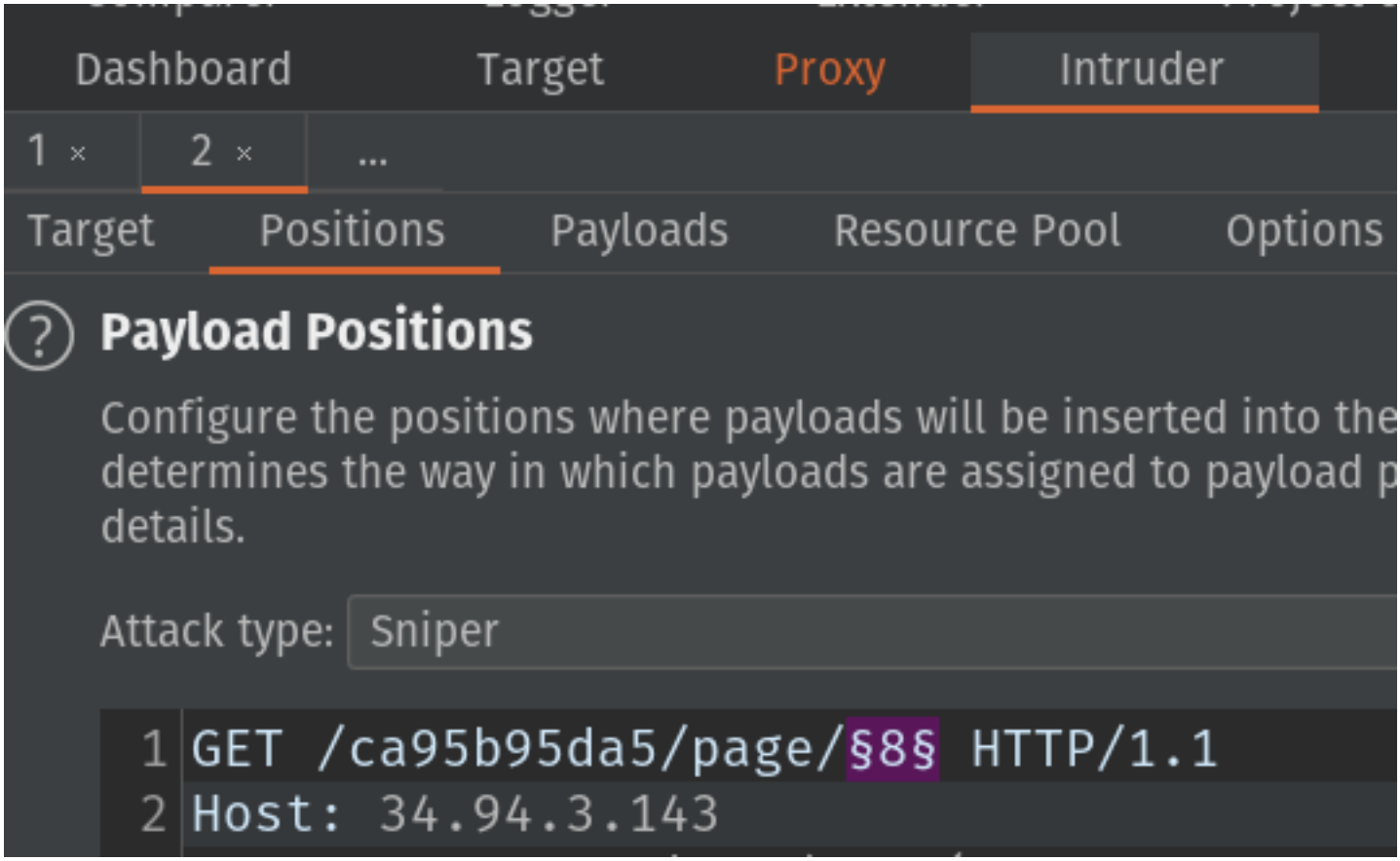
While testing the website I noticed the following:

→ We can edit the pages already created
→ We can use markdown
→ We can create a new page
→ When adding a page we can see the index in the URL

In the two pages created above we can inject XSS, which would allow an attacker to do many things, plus the index of the pages changes drastically when creating a new one.

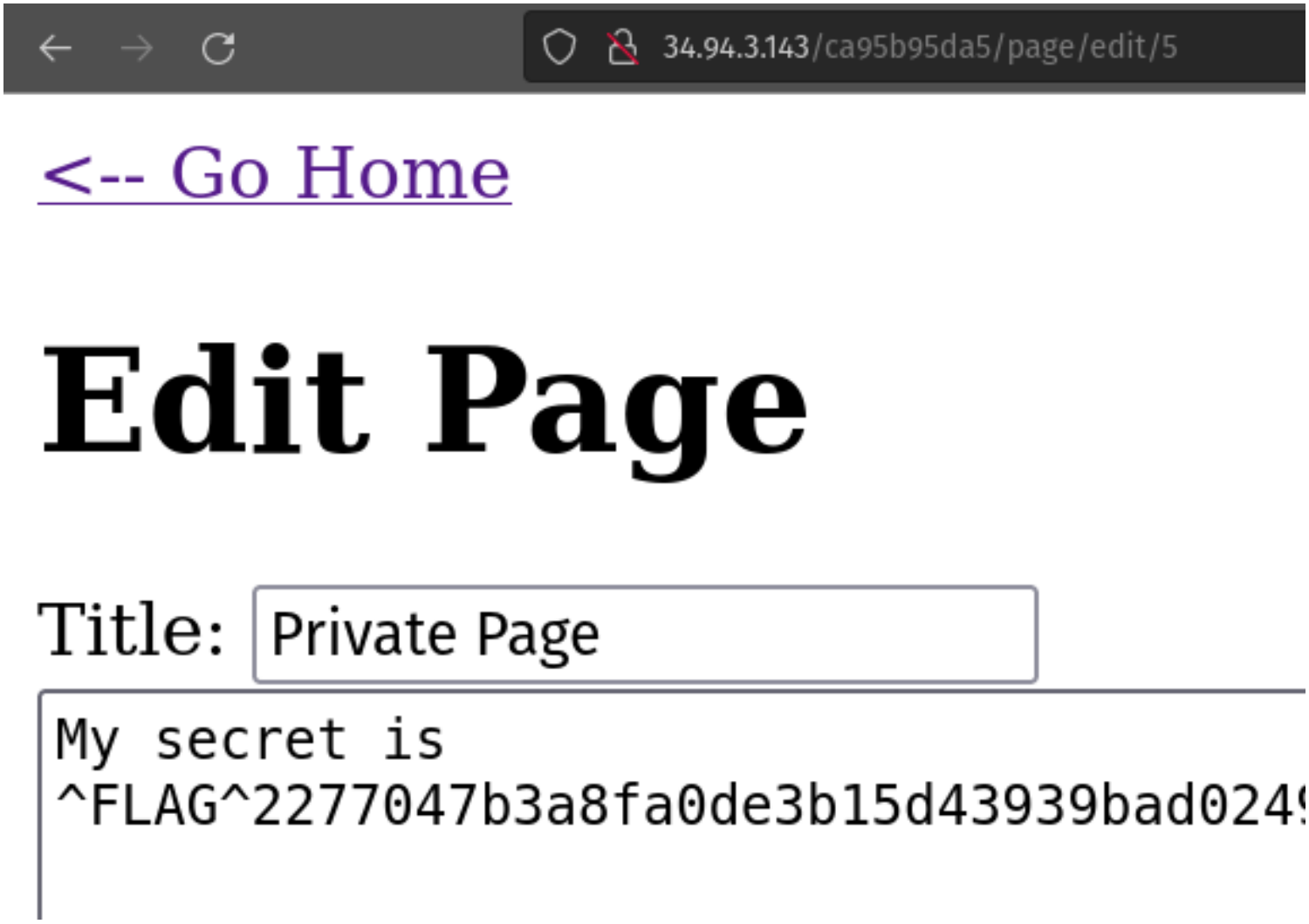


To find out more details we can intercept the request with BurpSuite



When reviewing the status codes everything seems normal but there is one 403 that catches my attention as it is the only one.

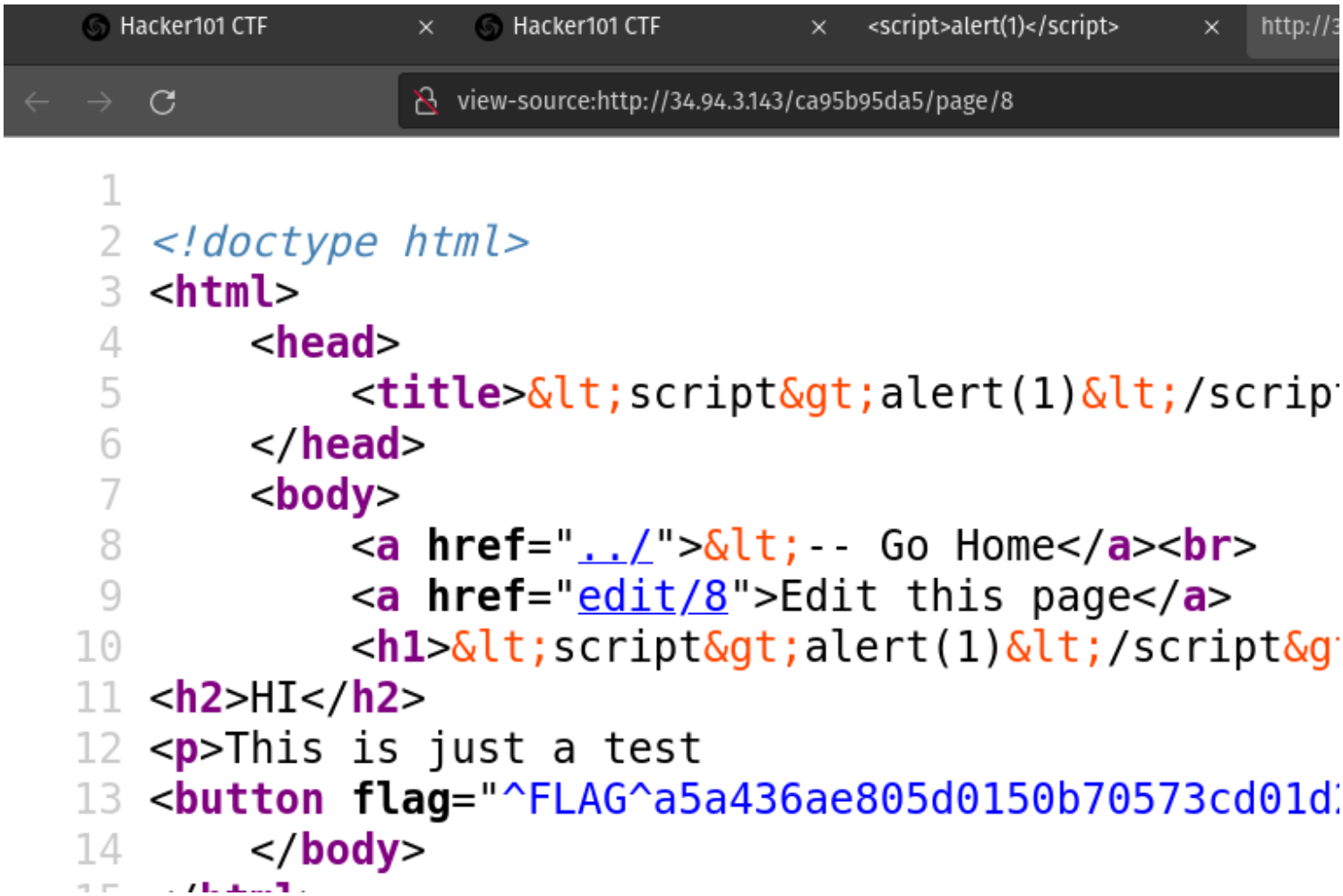
Since we can edit the pages I tried the one with the ID 5 and got the first flag



XSS is one of the most common vulnerabilities we can find in web applications, so to test I used the following payload in the page I created:

<button onclick="alert(1)"></button>

While reviewing the source code I found the flag!

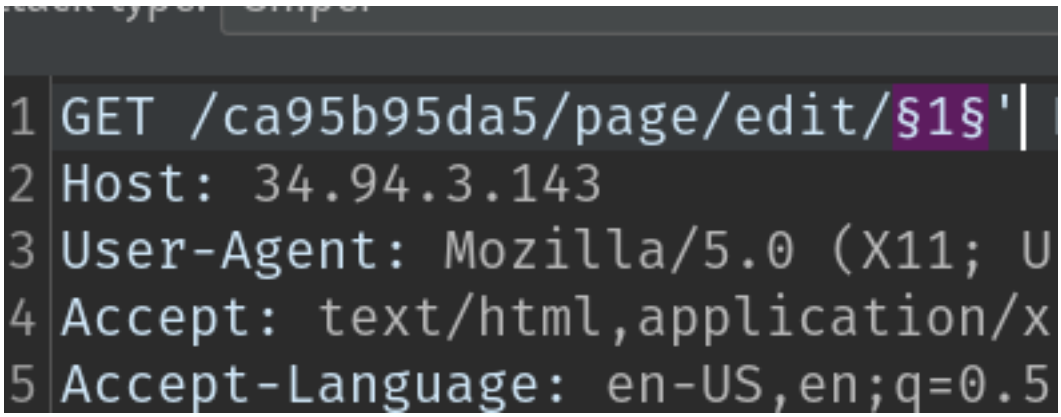


With web vulnerabilities we must test ALL possible entry points, so to get the next flag I used a different payload, I modified the title of the page I created:

<script>alert(1)</script>

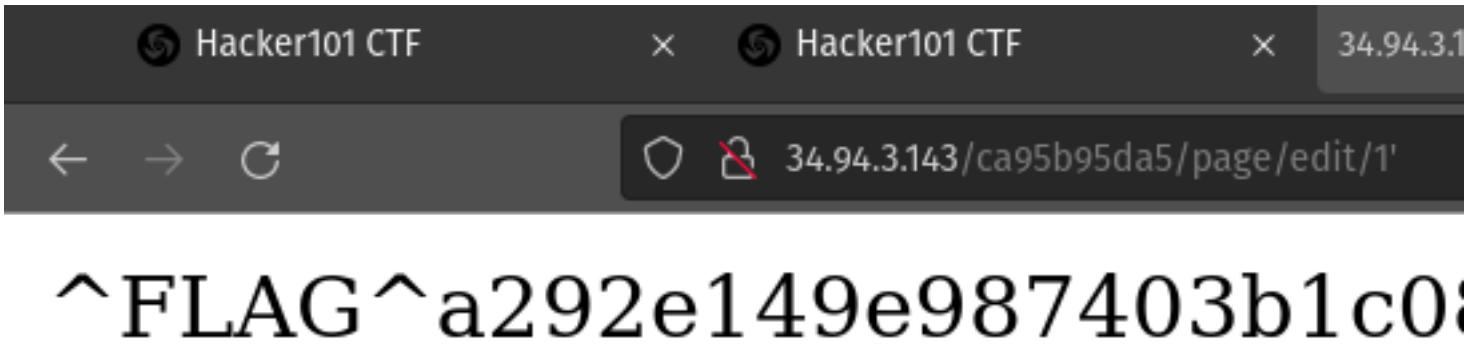
This type of XSS was saved and then executed by the server on the main page!

To find the last flag I spent quite some time trying XSS payloads but found nothing, so I remembered the URL, you should always keep an eye on the URL, so I decided to re-intercept it with BurpSuite!



This got me thinking... where does he get the pages from?

So I thought it might be a database and used the typical SQLi payload and got the flag!



Conclusion:

In order to successfully complete this level you must understand the following:

→ Cross Site Scripting
→ SQLi

To learn more about this check the following link: https://portswigger.net/web-security/