TOPIC 2

==========

Micro-CMS v1

Objective

=================

Flag 0

======

1. Stored XSS

===========

First visit the ‘create a new page’ link. When I create a new page, the details of the new page are reflected in the response. This is a good indication that the website might be vulnerable to XSS (Cross-site scripting). I test for XSS by editing the page title with this payload:

hello<script>alert(1);</script>

Going back home, the payload executes and I get the first flag.

=======================================

Flag 1

========

2. SQLi

===========

When editing a page, I notice that the page id is passed in the URL. I test this parameter for SQL injection by placing a ‘ (single quote) at the end of the id parameter and I get the second flag:

=====================

Flag 2

==========

3. Unauthorized access

When I created my first page, I observed that it was assigned an id of 12.

When I visit the two pages provided before, I observe that the pages have an id of 1 and 2. So I try to retrieve pages between 2 and 12. Page 7 responds with a 403 forbidden error while others respond with 404.

I poke around the system to look for other areas the page id is present and observe that the page id is also used when retrieving a page for editing.

I switch the page id to 7, refresh the page and get the third flag:

Flag 3

==========

Stored Xss

===============

The last place to test is the page body. When creating or editing a page, I observe that the page body allows markdown but not scripts.

Since the input is reflected in the page, I have to find a way to bypass the markdown filter to execute XSS. After searching and trying different payloads, I come across this payload:

<button onclick=alert(‘xss’)>click</button>

The payload executes successfully but there is no flag displayed.  
Viewing the source code, I find the flag: