

## CyberRange Vulnerabilities

By:

Michael Reda              ID: 900203291

Freddy Amgad              ID: 900203088

Shahd Elmahallawy      ID: 900194441

CSCE493005 –Web Security

Fall 2024

---

<b>Authentication and authorization</b>	<b>4</b>
1. Robots.txt	4
Description:	4
Mitigation:	4
2. Login with any Password	5
Description:	5
Mitigation:	5
3. Access By Admin Email	6
Description:	6
Mitigation:	6
4. BruteForce OTP	7
Description:	7
Mitigation:	7
5. Escalate a user to admin	7
Description:	9
Mitigation:	9
6. Delete Admin	10
Description:	10
Mitigation:	10
<b>Directory traversal:</b>	<b>11</b>
1. Challenge=../../CyberRange	11
Description:	12
Mitigation:	12
2. challenge=../ctf/	13
Description:	14
Mitigation:	14
3. /labs/(anything)	14
Description:	14
Mitigation:	16
<b>XSS:</b>	<b>16</b>
1. Stored XSS in the feedback message	16
Description:	16
Mitigation:	16
2. Stored XSS in user name.	17
Description:	17
Mitigation:	17

---

3. Stored XSS Lab Description	18
Description:	18
Mitigation:	18
<b>Business Logic:</b>	<b>19</b>
1. Increase Score	19
Description:	20
Mitigation:	20
2. The user can access hidden labs by just typing the name of the lab.	21
Description:	21
Mitigation:	22
3. Change the name	23
Description:	23
Mitigation:	23
4. Change the score to negative	24
Description:	24
Mitigation:	24
<b>Bugs:</b>	<b>25</b>
1. Lag in the flags	25
Description:	25
2. Error Fetching Space at the end	26
Description:	26
3. Submit the lab without a flag	27
Description:	27
4. Confirmed Password	28
Description:	28
<b>Bonus:</b>	<b>29</b>
1. No need to confirm the password.	29
Description:	29
Mitigation:	29

## Authentication and authorization

### 1. Robots.txt



```
User-agent: *
Disallow: /loginpagetocyberrange
```

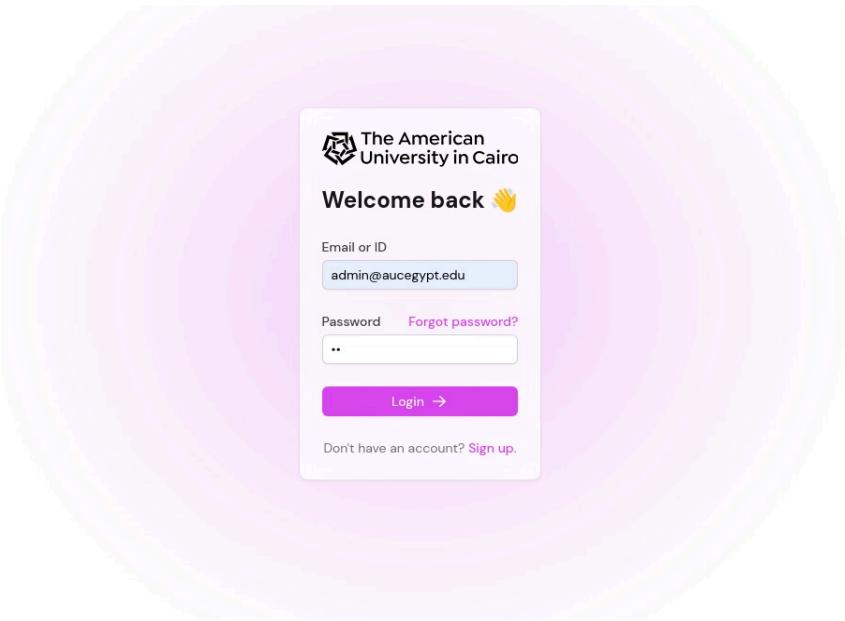
Description:

The first vulnerability is robots.txt which has the directory to login into the website

Mitigation:

Avoid placing sensitive directories or files in robots.txt that could disclose information to unauthorized users.

## 2. Login with any Password



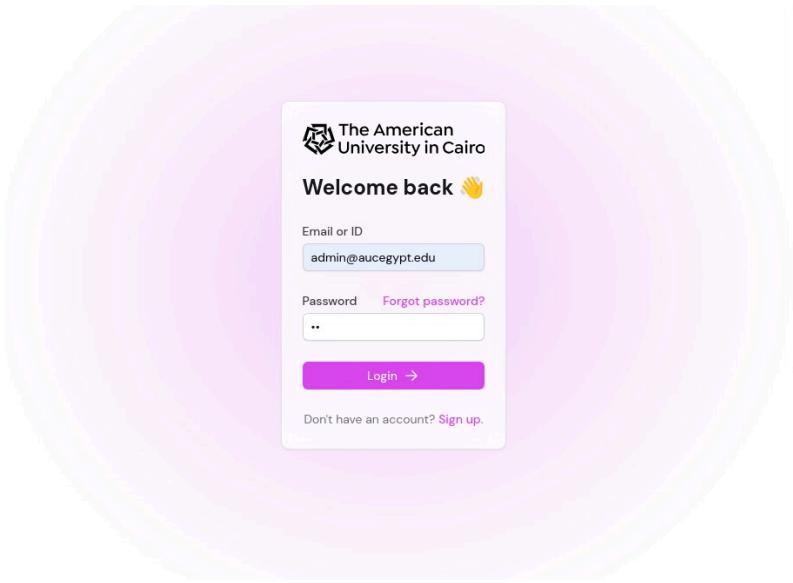
Description:

The user can enter any password no matter its length to log in. So, the website only checks the username.

Mitigation:

Add a condition to check the password in the backend by validating it in with the one stored in the database.

### 3. Access By Admin Email



Description:

Since the website only checks the username while logging in, you can easily guess the admin username, and get access to the admin portal.

Mitigation:

Change the admin username to something less guessable or less known.

## 4. BruteForce OTP

The screenshot shows the Burp Suite interface with the following details:

- Sniper attack** selected in the dropdown.
- Target**: 5008 (checkbox checked to update Host header to match target).
- Payloads** section:
  - Payload position: All payload positions
  - Payload type: Brute force
  - Payload count: 1,000
  - Request count: 1,000
- Payload configuration** section: This payload type generates payloads of specified lengths that contain all permutations of a specified character set.
  - Character set: 0123456789
  - Min length: 3
  - Max length: 3
- Payload processing** section: You can define rules to perform various processing tasks on each payload before it is used.
  - Add (checkbox checked)
  - Enabled (checkbox checked)
  - Rule (button)
  - Edit (button)
  - Remove (button)
  - Up (button)
  - Down (button)
- Payload encoding** section: This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.
  - URL-encode these characters: `\&lt;>?&lt;?&gt;`

Description:

OTP is only 3 digits (which is obvious from the frontend) we can easily brute force it.

Besides, OTP doesn't have an expiry date, so if the user didn't request a new OTP, it will be the same.

Mitigation:

1. Remove the suggest OTP form the frontend, so the user cannot know its length.
2. Add expiry time to the OTP

## 5. Escalate a user to admin

1. Normal user before making him admin and changing the token

The screenshot shows a browser window with the URL `54.93.205.45:10000/labs`. On the left, a sidebar menu includes Home, Labs (selected), CTF, Jeopardy, and Scoreboard. The main area displays categories: 0 labs, 0 labs, 2323 (11 labs), and Web Security (7 labs). Below this is a search bar with the placeholder "refe". The bottom half of the screen shows the Chrome DevTools Application tab. It lists cookies under the Storage section. The table has columns: Name, Value, Domain, Path, Expires / Max-Age, Size, HttpOnly, Secure, SameSite, Partition ..., Cross Site, and Priority. The data is as follows:

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Partition ...	Cross Site	Priority
name	refe	54.93.205.45	/	2024-12-25T20:08:56.000Z	8						Medium
role	user	54.93.205.45	/	2024-12-25T20:08:56.000Z	8						Medium
token	eyJhbGciOiJIUzI1NiIsInR5cCI6IkXV...	54.93.205.45	/	2024-12-25T20:08:56.000Z	209						Medium
v-0	250	54.93.205.45	/	Session	8						Medium

At the bottom of the DevTools, a message says: "A warning for 'Cookie' is found, but it is not used because the 'SameSite' attribute does not match. Consider setting a 'SameSite' attribute." The status bar at the bottom right shows "Default levels ▾ | No Issues | 1 hidden".

## 2. The Normal user after becoming an admin.

This screenshot is identical to the one above, showing the same browser interface and DevTools data. The only difference is the status bar at the bottom right, which now displays "DOM Invader is NOT enabled." and "augmented-dom-instrumentation.ts:1".

### 3. The normal user adding a category himself after changing the token.

The screenshot shows the Burp Suite interface. In the top navigation bar, 'Proxy' is selected. The main window displays a POST request to '/api/v1/admin/add-category' with the following payload:

```

POST /api/v1/admin/add-category HTTP/1.1
Host: 54.93.205.45:5008
Content-Length: 149
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCIkXVcJ9.eyJleHAiOjE3MzIxNTczNDEsImlhdGUiOiJlMjQwZmQ2MjQ0MjM1IiwiZW1haWQiOjYjZwL1iwcm9sZSIgZmFsbG93ZmF0ZT1jZW1haWQyMj9.aIaL2L54vtj-_SylQdM74vKLhvOKSUB3SwBymcC
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
Accept: */*
Origin: http://54.93.205.45:10000
Referer: http://54.93.205.45:10000/
Accept-Encoding: gzip, deflate, br
Content-Type: multipart/form-data; boundary=-----WebKitFormBoundaryH5dc1tgLoIE7nid
Accept-Language: en-US,en;q=0.9
Content-Disposition: form-data; name="name"
I am now admin
Content-Disposition: form-data; name="category"
I am now admin
Content-Disposition: form-data; name="refe"
I am now admin
Content-Disposition: form-data; name="role"
admin
Content-Disposition: form-data; name="tok.."
eyJhbGciOiJIUzI1NiIsInR5cCIkXVcJ9.eyJleHAiOjE3MzIxNTczNDEsImlhdGUiOiJlMjQwZmQ2MjQ0MjM1IiwiZW1haWQiOjYjZwL1iwcm9sZSIgZmFsbG93ZmF0ZT1jZW1haWQyMj9.aIaL2L54vtj-_SylQdM74vKLhvOKSUB3SwBymcC
Content-Type: application/json
Content-Length: 42
{
  "message": "Category added successfully"
}

```

The 'Application' tab in the bottom left shows a table of cookies. One cookie, 'tok..', has been modified to have a value of 'eyJhbGciOiJIUzI1NiIsInR5cCIkXVcJ9.eyJleHAiOjE3MzIxNTczNDEsImlhdGUiOiJlMjQwZmQ2MjQ0MjM1IiwiZW1haWQiOjYjZwL1iwcm9sZSIgZmFsbG93ZmF0ZT1jZW1haWQyMj9.aIaL2L54vtj-\_SylQdM74vKLhvOKSUB3SwBymcC'. The 'Response' tab shows a successful 200 OK response with the message 'Category added successfully'.

### Description:

The user can be escalated to admin by changing the role in the cookie to equal admin in the cookie and changing it in the token as well as adding the secret using jwt.io to reflect in the backend as well.

### Mitigation:

1. Ensure that JWTs are signed and validate the signature on the server side for every request to confirm that the token has not been tampered with.
2. Implement and enforce expiration dates for tokens to reduce the window of opportunity for an attacker to use a stolen token.
3. Instead of relying solely on client-side storage for critical data (like user role information), manage session states and roles on the server. When a session is established, the server should keep a record of the user's role and permissions and verify them for each request.

## 6. Delete Admin

The screenshot shows a browser interface with two main panels. On the left, there is a table titled 'Users' with columns for '#', 'Name', and 'Email'. Two rows are visible: one for 'refe' with email '12@aucegypt.edu' and another for 'hnaa' with email 'ia@aucegypt.edu'. On the right, a proxy tool window is open. The 'Intercept' tab is selected. The 'Request' section shows a raw HTTP request being intercepted. The URL is 'http://54.93.205.45:5008/api/v1/admin/delete-user?email=ia@aucegypt.edu'. The 'Inspector' panel on the right displays various request headers and attributes.

#	Name	Email	...
344423	refe	12@aucegypt.edu	...
121212q	hnaa	ia@aucegypt.edu	...

Request

```
Pretty Raw Hex
1. OPTIONS /api/v1/admin/delete-user?email=ia@aucegypt.edu HTTP/1.1
2. Host: 54.93.205.45:5008
3. Accept: */*
4. Access-Control-Request-Method: DELETE
5. Access-Control-Request-Headers: authorization,content-type
6. Origin: http://54.93.205.45:10000
7. User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
   AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70
   Safari/537.36
8. Sec-Fetch-Mode: cors
9. Referer: http://54.93.205.45:10000/
10. Accept-Encoding: gzip, deflate, br
11. Accept-Language: en-US,en;q=0.9
12. Connection: keep-alive
13.
14.
```

Inspector

- Request attributes: 2
- Request query parameters: 1
- Request body parameters: 0
- Request cookies: 0
- Request headers: 11

Description:

The Admin can delete other admins by changing the request when deleting a normal user, by intercepting the request, and changing the email in the request.

Mitigation:

Always validate inputs on the server-side to prevent unauthorized actions due to manipulated requests. Ensure the email address is verified and that it is the user's input before proceeding with any action.

## Directory traversal:

## 1. Challenge=../../CyberRange

The figure shows a screenshot of a browser-based proxy tool interface. The left panel, titled "Request", displays a list of 14 numbered log entries representing an incoming HTTP request. The right panel, titled "Response", displays a list of 10 numbered log entries representing the outgoing HTTP response. Both panels have tabs for "Pretty", "Raw", "Hex", and "Hackvertor". Below each panel is a search bar and a "0 highlights" indicator. The top right corner of the interface includes standard window control buttons.

**Request**

Pretty Raw Hex Hackvertor

1 OPTIONS /api/v1/shared/download-all?challenge=BabaShark HTTP/1.1  
2 Host: 54.93.205.45:5008  
3 Accept: \*/\*  
4 Access-Control-Request-Method: GET  
5 Access-Control-Request-Headers: authorization  
6 Origin: http://54.93.205.45:10000  
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)  
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70  
Safari/537.36  
8 Sec-Fetch-Mode: cors  
9 Referer: http://54.93.205.45:10000/  
10 Accept-Encoding: gzip, deflate, br  
11 Accept-Language: en-US,en;q=0.9  
12 Connection: keep-alive  
13  
14

**Response**

Pretty Raw Hex Render Hackvertor

1 HTTP/1.1 204 No Content  
2 Access-Control-Allow-Headers: authorization  
3 Access-Control-Allow-Methods: GET,DELETE,PUT,POST  
4 Access-Control-Allow-Origin: \*  
5 Vary: Origin  
6 Vary: Access-Control-Request-Method  
7 Vary: Access-Control-Request-Headers  
8 Date: Mon, 25 Nov 2024 00:04:47 GMT  
9  
10

## Description:

When the admin clicks on any of the challenges in the jeopardy. You can intercept the request:

Options /api/v1/shared/download-all?challenge=BabaShark HTTP/1.1

Change it to GET request and add the Authorization from any other GET request and then you can traverse to .. or ../../CyberRnage

- GET /api/v1/shared/download-all?challenge=.. HTTP/1.1
  - GET /api/v1/shared/download-all?challenge=.../CyberRange HTTP/1.1

## Mitigation:

Use secure coding practices like rejecting path traversal patterns (e.g., ".."). Implement strict access controls and audit logs. Canonicalize the URL.

2. challenge=../ctf/

The screenshot shows two panels of the NetworkMiner tool interface. The left panel is titled "Request" and the right panel is titled "Response".

**Request:**

- Pretty (selected), Raw, Hex, Hackvator
- 1 OPTIONS /api/v1/shared/download-attachment?filename=BabaShark.pcap
- 2 Host: 54.93.205.45:5008
- 3 Accept: \*/\*
- 4 Access-Control-Request-Method: GET
- 5 Access-Control-Request-Headers: authorization
- 6 Origin: http://54.93.205.45:10000
- 7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36
- 8 Sec-Fetch-Mode: cors
- 9 Referer: http://54.93.205.45:10000/
- 10 Accept-Encoding: gzip, deflate, br
- 11 Accept-Language: en-US,en;q=0.9
- 12 Connection: keep-alive
- 13
- 14

**Response:**

- Pretty (selected), Raw, Hex, Render, Hackvator
- 1 HTTP/1.1 204 No Content
- 2 Access-Control-Allow-Headers: authorization
- 3 Access-Control-Allow-Methods: GET,DELETE,PUT,POST
- 4 Access-Control-Allow-Origin: \*
- 5 Vary: Origin
- 6 Vary: Access-Control-Request-Method
- 7 Vary: Access-Control-Request-Headers
- 8 Date: Mon, 25 Nov 2024 00:08:01 GMT
- 9
- 10

At the bottom of both panels are standard NetworkMiner navigation and search controls.

Request		Response						
Pretty	Raw	Hex	Hackvtor	Pretty	Raw	Hex	Render	Hackvtor
1 GET /api/v1/shared/download-attachment?filename=BabaShark.pcap&challenge=..//tf/babaShark	HTTP/1.1			1 HTTP/1.1 200 OK				
2 Host: 54.93.205.45:5008				2 Accept-Ranges: bytes				
3 Accept: */*				3 Access-Control-Allow-Origin: *				
4 Access-Control-Request-Method: GET				4 Content-Length: 13144				
5 Access-Control-Request-Headers: authorization				5 Content-Type: application/vnd.tcpdump.pcap				
6 Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IlRpXCVJ9eyJleHai0jE3MzUwODQ4MzAsImIhdCIGMtczMjQSMjgzMCwiaWQiOjJtaWN0YWxMTIilCJuYWl1ljoiTW1jaGF1bCBSZWRhIiwiemc9sZS16mFkbWluIiwidXNlc19pZC16IjkwMDIwMzIiMSJS.eyJp-bVtmwl3phkMaCoqAc_e3BQirDnSHFT2NjkNT7wsIA				6 Last-Modified: Fri, 25 Oct 2024 20:36:47 GMT				
7 Origin: http://54.93.205.45:10000				7 Vary: Origin				
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36				8 Date: Mon, 25 Nov 2024 00:09:56 GMT				
9 Sec-Fetch-Mode: cors				9				
10 Referer: http://54.93.205.45:10000/				10 ȫȫ, ȫȫcȫȫ) SDOPVYçØE? ȫȫ-ȫȫ yȫȫsȫȫxȫȫ ȫȫconvdarkbyteruco@agt1d-serversnetnstldverisign-grscm]pn30				
11 Accept-Encoding: gzip, deflate, br				: ȫȫQ106cȫȫh0) SDOPVYçØE? ȫȫ-ȫȫ yȫȫsȫȫxȫȫ ȫȫconvdarkbyteruco@agt1d-serversnetnstldverisign-grscm]pn30				
12 Accept-Language: en-US,en;q=0.9				: ȫȫQ106cȫȫBixçPvYçQ) SDOPJp@0ç, ȫȫ-ȫȫ yȫȫsȫȫxȫȫ ȫȫconvdarkbyterulocaldomain0çc1f) SDOPVYçØE? ȫȫ-ȫȫ yȫȫsȫȫxȫȫ ȫȫconvdarkbyterulocaldomain@aroot-serversnetnstldverisign-grscm]pn30				
13 Connection: keep-alive				: ȫȫQ106cȫȫb**yÿÿÿÿ) SDOPVYçØE? ȫȫ-ȫȫ yȫȫsȫȫxȫȫ ȫȫconvdarkbyterulocaldomain@aroot-serversnetnstldverisign-grscm]pn30				
14				: ȫȫQ106cȫȫb**yÿÿÿÿ) SDOPVYçØE? ȫȫ-ȫȫ yȫȫsȫȫxȫȫ ȫȫconvdarkbyterulocaldomain@aroot-serversnetnstldverisign-grscm]pn30				
15				: ȫȫQ106cȫȫb**yÿÿÿÿ) SDOPVYçØE? ȫȫ-ȫȫ yȫȫsȫȫxȫȫ ȫȫconvdarkbyterulocaldomain@aroot-serversnetnstldverisign-grscm]pn30				

---

## Description:

When the admin clicks on any of the challenges in the jeopardy, and clicks on any of the files that are attached to the challenge. You can intercept the request:

```
Options /api/v1/shared/download-attachment?challenge=../ctf/ file name=BabaShark.pcap  
HTTP/1.1
```

Change it to GET request and add the Authorization from any other GET request and then you can traverse to ..//ctf/BabaShark (*or any challenge name*)

```
GET /api/v1/shared/download-attachment?challenge=../ctf/challenge HTTP/1.1
```

## Mitigation:

Use secure coding practices like rejecting path traversal patterns (e.g., ".."). Implement strict access controls and audit logs. Canonicalize the URL.

## 3. /labs/(anything)

### Description:

Changing the url from above after /labs to access any tabs there

The American University in Cairo

web

Labs / web

QWJQWJ    

hiiiiiii    

jojo    

Write a description for the labASW

<button onclick="alert('Hello! This is you')>

Write a description for the lab

+

Home     Labs     CTF     Users

Michael Reda

By traversing to the following: <http://54.93.205.45:10000/labs/web>, the user see the hidden lab.

Another shape for it even if there is no labs inside

The American University in Cairo

232322

Labs / 232322

232322

Home     Labs     CTF     Jeopardy     Scoreboard

refe

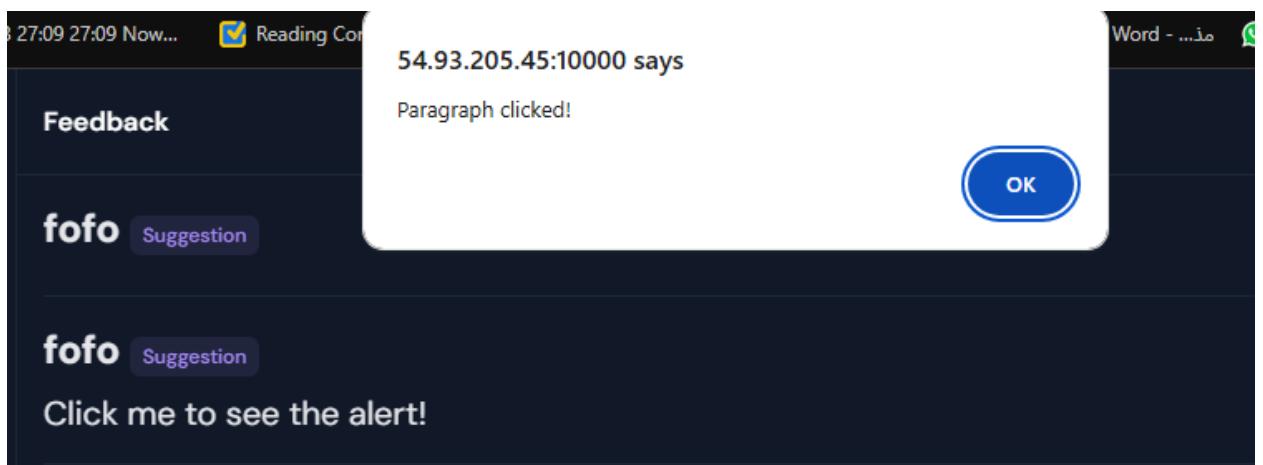
---

Mitigation:

Use secure coding practices like rejecting path traversal patterns (e.g., "../"). Implement strict access controls and audit logs. Canonicalize the URL.

## XSS:

### 1. Stored XSS in the feedback message



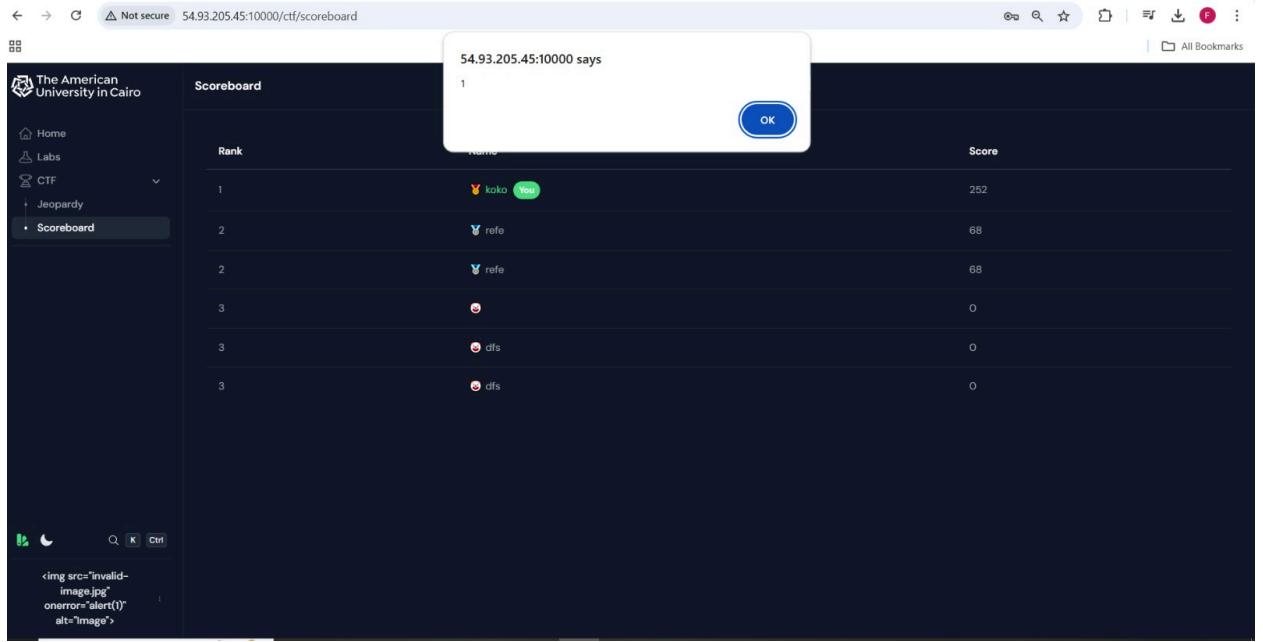
Description:

The user can inject an HTML code into the feedback message.

Mitigation:

Sanitize and validate user inputs before storing them and make sure that the outputs is not outputting unexpected outputs.

## 2. Stored XSS in user name.



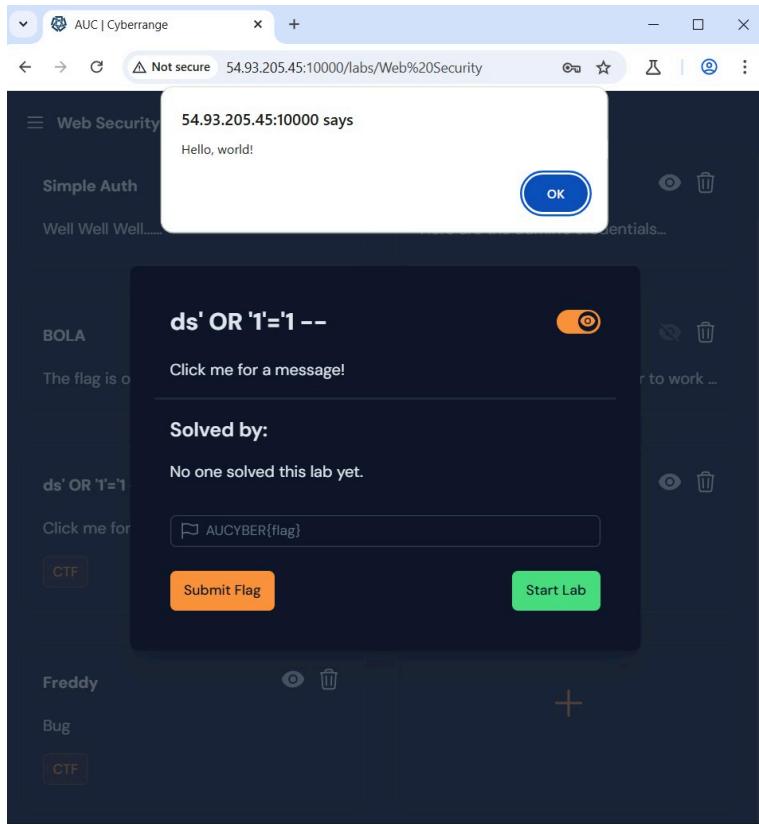
Description:

The user can inject an HTML code into the change name and save changes.

Mitigation:

Sanitize and validate user inputs before storing them and make sure that the outputs is not outputting unexpected outputs.

### 3. Stored XSS Lab Description



Description:

Can Store HTML in the lab description

Mitigation:

Sanitize and validate user inputs before storing them and make sure that the outputs is not outputting unexpected outputs.

# Business Logic:

## 1. Increase Score

### 1. Before submitting the flag:

The screenshot shows a web-based security challenge interface with a sidebar containing various challenges like DT4T3, Multi Layer Protection, and Babashark. The Babashark challenge is selected, showing a 'dss' sub-challenge. The challenge details page includes a hint: 'Click for hint, dw it won't lower your points.' A text input field contains the flag 'k'. A 'Submit Flag' button is visible. Below the challenge details, a 'Score: 0' badge is shown.

On the right, a NetworkMiner tool is open, showing a captured POST request to the URL `http://54.93.205.45:5008/api/v1/shared/submit-flag?flag=k&challenge=dss`. The request body is JSON with the flag value 'k'. The response shows a successful 200 OK status with a JSON message: "You have completed the challenge!!".

Request
Pretty Raw Hex 1 POST /api/v1/shared/submit-flag?flag=k&challenge=dss HTTP/1.1 2 Host: 54.93.205.45:5008 3 Content-Length: 0 4 Authorization: bearer eyJhbGciOiJIUzI1NiIsInR5cGwiOiJpXVCDQJ9_eyJleHAiOiE3MzUwOTExMDExImhhdC16MTczMyQSDTAmWSakawQz01xMTE1LChvW1ljoia29rbys1njbGU0j1jc2Vyi1widXNlc19pZCZGfNf3ccXhdby9s_eyllmjs5oGby167URhqAhsIDB0z7R2rUh0U6qkn5C4 5 Accept-Language: en-US,en;q=0.9 6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70 Safari/537.36 7 Content-Type: application/json 8 Accept: */* 9 Origin: http://54.93.205.45:10000 10 Referer: http://54.93.205.45:10000/ 11 Accept-Encoding: gzip, deflate, br

Response
Pretty Raw Hex Render 1 HTTP/1.1 200 OK 2 Access-Control-Allow-Origin: * 3 Content-Type: application/json 4 Vary: Origin 5 Date: Mon, 25 Nov 2024 01:44:37 GMT 6 Content-Length: 51 7 8 { 9     "message": "You have completed the challenge!!" }

### 2. After submitting the lab many times:

## Jeopardy

ss ✓

ss

Web Exploitation

12 Points

Easy

dss ✓

dss

Web Exploitation

12 Points

Easy

ss ✓

ss

Web Exploitation

12 Points

Easy

dss ✓

dss

Web Exploitation

12 Points

Easy

ss ✓

ss

Web Exploitation

12 Points

Easy

dss ✓

dss

Web Exploitation

12 Points

Easy

ss ✓

ss

Web Exploitation

12 Points

Easy

dss ✓

dss

Web Exploitation

12 Points

Easy

iky

iky

Miscellaneous

-4 Points

Insane

 Score: 252

## Description:

The user can take the request and submit the challenge multiple times to increase his score.

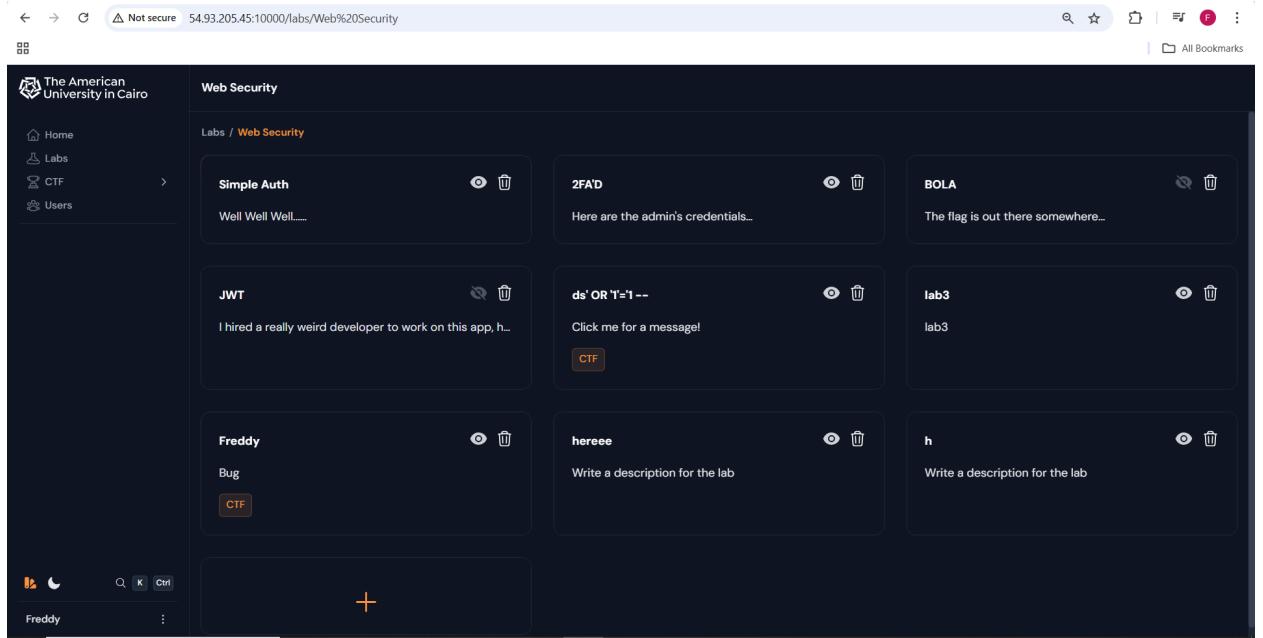
## Mitigation:

Implement a server-side check to ensure that a challenge can only be submitted once per user by maintaining a submission record. Reject duplicate submissions and notify the user.

## 2. The user can access hidden labs by just typing the name of the lab.

Description:

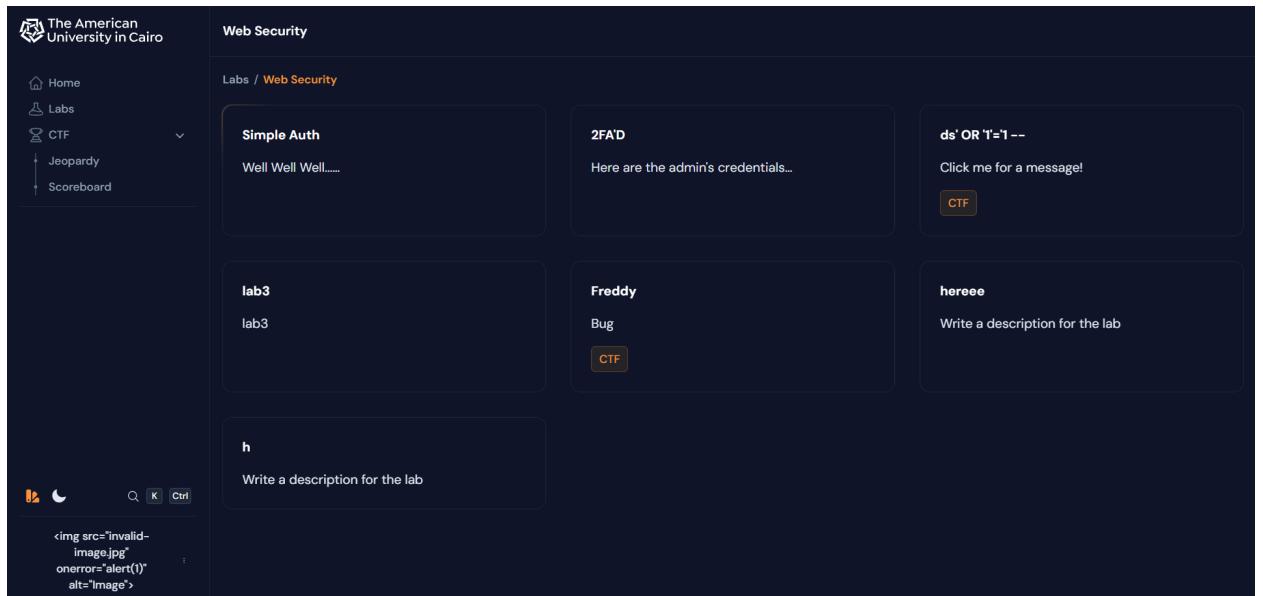
1. The hidden lab as shown is BOLA



The screenshot shows a web browser window with the URL 54.93.205.45:10000/labs/Web%20Security. The page title is 'Web Security'. On the left sidebar, there are links for Home, Labs, CTF, and Users. Under 'Labs / Web Security', there are several cards:

- Simple Auth**: Well Well Well.....
- 2FAD**: Here are the admin's credentials...
- BOLA**: The flag is out there somewhere...
- JWT**: I hired a really weird developer to work on this app, h...
- ds' OR '1='1 --**: Click me for a message!
- lab3**: lab3
- Freddy**: Bug
- hereee**: Write a description for the lab
- h**: Write a description for the lab

2. Then from the user's perspective



This screenshot shows the same web browser window after the user has accessed the 'BOLA' lab. The 'BOLA' card is missing from the list, while the other cards remain. The rest of the interface is identical to the first screenshot.

### 3. The original request of the user trying to access an existing lab

My

The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. The 'Request' pane displays a GET request to 'http://54.93.205.45:10000/terminal?container\_names=Lab&labname=Simple%20'. The 'Payload' field contains a large, complex string of characters, possibly a base64 encoded payload or exploit code. The 'Inspector' pane shows various request details like headers and cookies.

### 4. Then the result after changing the name in the request to the hidden lab

The screenshot shows the Burp Suite interface with the 'Intercept' tab selected. The 'Request' pane displays a POST request to 'http://54.93.205.45:10000/api/v1/shared/run-lab?name=BOLA'. The response body is a JSON object containing a welcome message and information about available endpoints.

```
[{"Welcome": "Welcome to DOCUMENTS API. This is a simple API for a document management system. It's secured but I'm not sure if it's secure enough.", "Guide": "Visit /api/v1 for more information on the available endpoints.", "BestAPIClientsForTesting": ["Bruno": "https://www.usebruno.com/", "Httpie": "https://httpie.io", "Insomnia": "https://insomnia.rest", "Postman": "https://www.postman.com", "ThunderClient": "https://www.thunderclient.com"}]
```

### Mitigation:

Validate the requested lab against a database of labs accessible to the user based on their permissions. Reject requests for labs not assigned to the user.

### 3. Change the name

Description:

1. Before changing the user name to another name that already exists

The screenshot shows a web application interface. On the left, there is a "Scoreboard" table with three rows:

Rank	Name	Score
1	koko	252
2	refo	68
3	gil	8

To the right of the scoreboard is a Network tab showing a POST request. The "request" pane displays the following JSON payload:

```
Accept-Language: en-US,en;q=0.9
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.6723.70
Safari/537.36
Content-Type: application/json
Accept: */*
Origin: http://54.93.205.45:10000
Referer: http://54.93.205.45:10000/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
{
  "name": "gil",
  "password_current": "12345678",
  "password_new": "12345678"
}
```

2. After changing the user name to a user name that already exists this copies the already existing user score to the user

The screenshot shows the same web application interface. The "Scoreboard" table now has two rows:

Rank	Name	Score
1	koko	252
2	refo	68

The "request" pane shows the same JSON payload as the previous screenshot, indicating that the user's score was copied from the user with name "refo".

3. By doing so the rank of the user will change to maybe a user with a higher rank

Mitigation:

Use email as the identifier and tie points to an immutable user ID, not the username, to prevent tampering.

## 4. Change the score to negative

The screenshot shows a Jeopardy-style challenge interface with the following details:

- Scoreboard:**

Rank	Name	Score
1	refe	68
2	gil	-4
- Success Message:** You have completed the challenge! (with a green checkmark icon)
- Points Message:** You earned -4 points, keep it up! (with a green checkmark icon)
- Challenge Categories:**
  - ROT4T3:** I sent the flag, but it rotated! (: SMULX{1\_4do4q5\_j0l4l3\_eq\_)'.  
Category: Cryptography, Points: 20, Difficulty: Easy.
  - Multi Layer Protection:** Security is all about obfuscation right?  
Category: Cryptography, Points: 50, Difficulty: Medium.
  - BabyShark:** HTTP is just plain text  
Category: Network Security, Points: 20, Difficulty: Easy.
  - BabaShark:** Who said DNS is always for the good.  
Category: Network Security, Points: 75, Difficulty: Hard.
  - re:** we  
Category: Miscellaneous, Points: 4, Difficulty: Medium.
  - new lab:** easy  
Category: Warmup, Points: 50, Difficulty: Easy.
  - f0fo is here:** f0fo  
Category: Forensics, Points: 10, Difficulty: Medium.
  - dss:** dss  
Category: Web Exploitation, Points: 12, Difficulty: Easy.
  - miky:** miky  
Category: Miscellaneous, Points: -4, Difficulty: Insane.

Description:

When the admin is creating the lab, he cannot change the score to negative in the frontend, but he can intercept the request and change it to a negative number.

Mitigation:

To mitigate this issue, implement server-side validation to enforce business rules, such as ensuring the score cannot be negative. Never rely solely on front-end controls for data validation. Reject invalid requests at the backend and log suspicious activities for monitoring and further investigation.

## Bugs:

### 1. Lag in the flags

The image consists of two vertically stacked screenshots of a web-based lab system interface. Both screenshots show a sidebar on the left with the following navigation menu:

- The American University in Cairo
- Home
- Labs
- CTF
- Jeopardy
- Scoreboard

The main content area displays two lab entries under the heading "Labs / web".

**Screenshot 1 (Top):** The first entry is labeled "QWJQWJ" and has the instruction "Write a description for the labASW". Below it is a text input field containing "t2" and a "Submitting..." button.

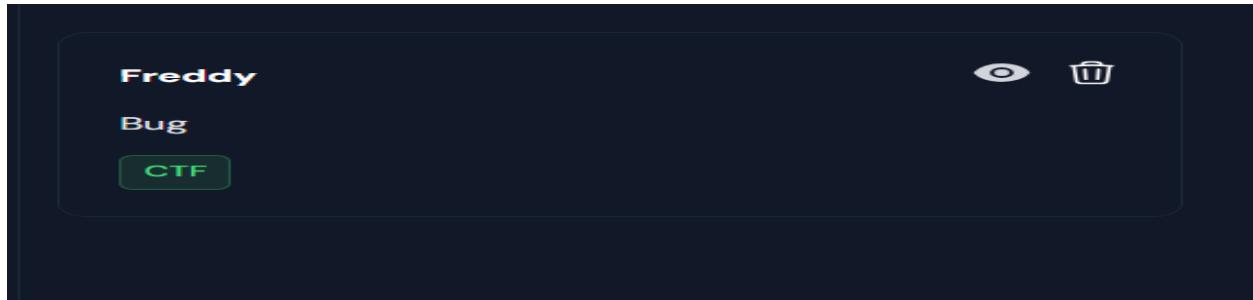
**Screenshot 2 (Bottom):** The second entry is labeled "jojo" and has the instruction "Write a description for the lab". Below it is a text input field containing "t2" and a "Submitting..." button.

In both screenshots, the "Submitting..." button is active, indicating that both submissions are in progress simultaneously. This visual representation highlights a bug where multiple flag values are being written at the same time across different labs.

## Description:

Bug of writing all the flag values at the same time in all labs.

## 2. Error Fetching Space at the end



Description:

The other Bug is that when the lab has a space at the end when it is retrieved to be deleted in the Frontend it will not be deleted; however, it will return deleted successfully.

### 3. Submit the lab without a flag

## Request

Pretty Raw Hex

```
1 POST /api/v1/shared/submit-flag?challenge=fofo%20ist%20here HTTP/1.1
2 Host: 54.93.205.45:5008
3 Content-Length: 0
4 Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ9.eyJleHAiOiE3MzUxMTk5NTcsImIhdCI6MTczMjUyNzk1NywiawQiOiIxM
iIisInhbWUiOiyZWLlIiwiim9sZSI6InVzZXIiLCJlc2VyX2lkIjoiMzQONDIZIn0.47IlfNTezKBBoCoyrsoIg7nrAHH
Usb3IybdHDptkALE
5 Accept-Language: en-US,en;q=0.9
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/130.0.6723.70 Safari/537.36
7 Content-Type: application/json
8 Accept: */*
9 Origin: http://54.93.205.45:10000
10 Referer: http://54.93.205.45:10000/
11 Accept-Encoding: gzip, deflate, br
12 Connection: keep-alive
13
```

?

Search 0 highlights

---

## Response

Pretty Raw Hex Render

```
5 Date: Mon, 25 Nov 2024 09:46:33 GMT
6 Content-Length: 51
7
8 {
    "message": "You have completed the challenge!!"
}
9
```

?

Search 0 highlights

## Description:

I created a challenge that when I submit without the flag it returns that “You have completed the challenge 🎉”; however, it is not reflected in the scoreboard.

## 4. Confirmed Password

The screenshot shows the Postman application interface with two sections: Request and Response.

**Request:**

- Method: POST
- URL: http://54.93.205.45:10000/api/users
- Headers:
  - Origin: http://54.93.205.45:10000
  - Referer: http://54.93.205.45:10000/
  - Accept-Encoding: gzip, deflate, br
  - Connection: keep-alive
- Body (Pretty):

```
8 Origin: http://54.93.205.45:10000
9 Referer: http://54.93.205.45:10000/
10 Accept-Encoding: gzip, deflate, br
11 Connection: keep-alive
12
13 {
  "email": "user90@saucegypt.edu",
  "id": "80",
  "name": "dfs",
  "password": "123456789",
  "confirmPassword": "1"
}
```

**Response:**

- Status: 200 OK
- Headers:
  - Access-Control-Allow-Origin: \*
  - Content-Type: application/json
  - Vary: Origin
  - Date: Mon, 25 Nov 2024 09:56:21 GMT
  - Content-Length: 44
- Body (Pretty):

```
1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 Content-Type: application/json
4 Vary: Origin
5 Date: Mon, 25 Nov 2024 09:56:21 GMT
6 Content-Length: 44
7
8 {
  "message": "User registered successfully!"
}
9
```

At the bottom, there are buttons for Event log (2) and All issues.

Description:

Users can make an account with a password that does not match the confirmed password.

## Bonus :

1. No need to confirm the password.

The screenshot shows a REST client interface with two sections: Request and Response.

**Request:**

```
Accept: */*
Origin: http://54.93.205.45:10000
Referer: http://54.93.205.45:10000/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
name:"dfs",
password_new:"123456789",
password_new:"1234567890"
```

**Response:**

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Type: application/json
Vary: Origin
Date: Mon, 25 Nov 2024 10:15:34 GMT
Content-Length: 47
{
    "message": "Account is updated successfully!"
}
```

Description:

The user can change his password without checking for the old password.

Mitigation:

Require the user to input their current password and validate it server-side before allowing a password change.