OWASP Juice Shop

Penetration Test Report

Version	**1.0**
Authors	**Mohamed Ehab,** **Bahy Mohamed**
Issue Date	**7/12/2023**

Table of Contents

1.0 OWASP Juice Shop Penetration Test Report.....2

1.1 Introduction.....2

1.2 Scope.....2

2.0 Executive Summary.....3

2.1 Recommendations.....3

3.0 Risk Assessment.....4

3.1 Likelihood.....4

3.2 Impact.....4

4.0 Findings Summary.....5

5.0 Vulnerability and Remediation Report.....7

6.0 Information Gathering.....8

7.0 High Findings.....9

8.0 Medium Findings.....23

9.0 Low Findings.....31

1.0 OWASP Juice Shop Penetration Test Report

1.1 Introduction

Subject of this document is a summary of penetration tests performed against web applications owned by OWASP Juice Shop. Test was conducted according to rules of engagement defined and approved at the beginning by both parties – customer and contractor. Black-box pentesting assignment was requested.

Black-box penetration test classification means that penetration tester has no internal knowledge about target system architecture. He/she can use information commonly available on the Internet. More data can be collected during the reconnaissance phase based on observation of target system behavior. Black-box penetration test results give overview of vulnerabilities exploitable from outside the company network. It shows how unauthenticated actors can take advantage of weaknesses existing in tested web applications.

Time frame defined:

Penetration test start: 1/12/2023,

Penetration test end: 7/12/2023

.

1.2 Scope

To perform a Black Box Web Application Penetration Test against the web applications of the organization named OWASP Juice Shop.

This is what the client organization defined as scope of the tests:

- Dedicated Web Server: LocalHost
- Domain: http://localhost:3000/
- Subdomains: all subdomains

2.0 – EXECUTIVE SUMMARY

Conducted penetration test uncovered several security weaknesses present in web applications owned by OWASP Juice Shop company

When performing the penetration test, there were several alarming vulnerabilities that were identified on OWASP Juice Shop networks. When performing the attacks, I was able to gain access to every user account and admin account, primarily due to SQL injection. During the testing, I had access to all user data. Brief description:

1. Subdomain Found: 0

2. 8 High, 5 Med, 7 Low issues have been identified.

2.1 – Recommendations

I recommend patching the vulnerabilities identified during the testing to ensure that an attacker cannot exploit these systems in the future. One thing to remember is that these systems require frequent patching and once patched, should remain on a regular patch program to protect additional vulnerabilities that are discovered at a later date.

Additionally, enhance security by escaping control characters in user inputs to prevent SQL injection and other injection-based exploits. Validate server-side inputs to process only expected, sanitized data, reducing the risk of manipulation. Implement strong authentication and access controls for the FTP server, including multi-factor authentication .

Audit system logs regularly, implement intrusion detection and prevention systems, and conduct periodic penetration testing for proactive vulnerability management. Promote security awareness through regular training for administrators and users. A proactive approach to security is essential in the ever-evolving landscape of cybersecurity.

To prevent Insecure Direct Object References (IDOR), ensure that proper access controls are implemented throughout the application. Adopt a principle of least privilege, granting users access only to the resources necessary for their roles.

3.0 – RISK ASSESSMENT

3.1 Likelihood

The likelihood is a measurement of the capacity to carry out an attack. The factor will be the difficulty or skill required to carry out the attack.

Risk	Description
Critical	An attacker is near-certain to carry out the threat event
High	An untrained user could exploit the vulnerability. The vulnerability is obvious or easily accessed
Medium	The vulnerability required some hacking knowledge to carry out the attack
Low	The vulnerability required significant time, skill , access and other resource to carry out the attack

3.2 Impact

The impact is a measurement of the adverse effect carrying out an attack would have on the organization.

Risk	Description
Critical	An attack would cause catastrophic or severe effect on operation, asset or other organization
High	An attack would severely degrade mission capability. The attack may result in damage to asset(data exposure)
Medium	An attack would degrade the mission capability. An attack would allow for primary function to application to resume, but at reduced effectiveness
Low	An attack would degrade mission capability in a limited capacity. The attack may result in marginal damage to assets

4.0 -- FINDINGS SUMMARY

Findings Name In Score Board	Likelihood	Impact	Rating	Category	Stars
Change Bender's Password	Medium	Critical	High	Broken Authentication	★★★★★
Email Leak	Low	High	High	Sensitive Data Exposure	★★★★★
Christmas Special	Medium	Low	Low	SQL Injection	★★★★
User Credentials	Medium	Critical	High	SQL Injection	★★★★
Deluxe Fraud	Medium	Medium	Medium	Improper Input Validation	★★★
CAPTCHA Bypass	Medium	High	High	Broken Anti Automation	★★★
Forged Feedback	Medium	High	Medium	Broken Access Control	★★★
Forged Review	Medium	High	Medium	Broken Access Control	★★★
Login Bender	Medium	High	High	SQL Injection	★★★
Empty User Registration	Medium	Low	Low	Improper Input Validation	★★

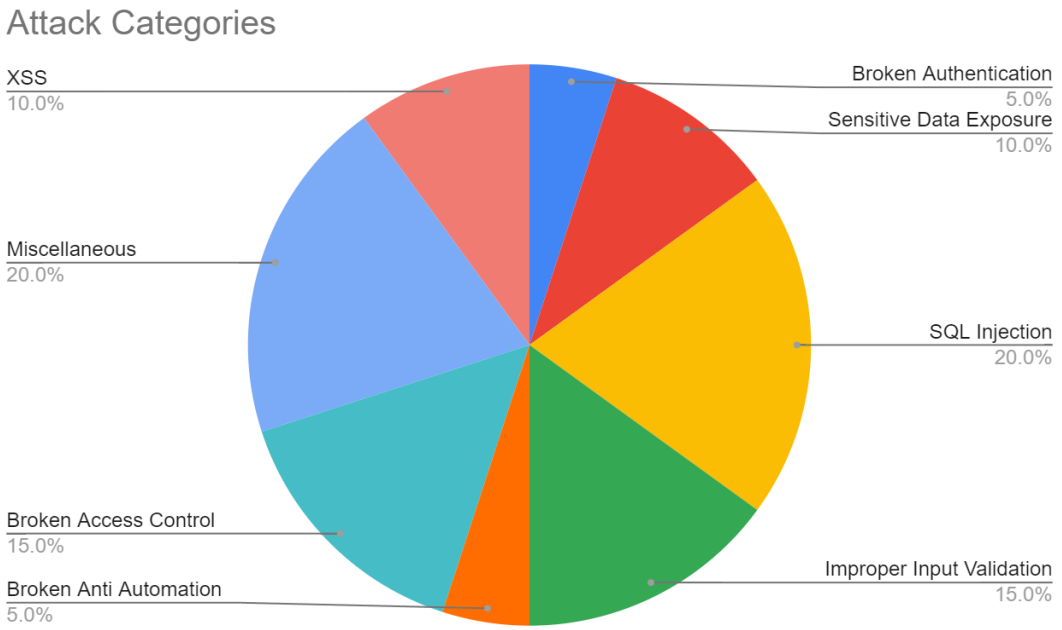
Login Admin	Medium	Critical	High	SQL Injection	★★
Security Policy	Medium	Low	Low	Miscellaneous	★★
View Basket	Medium	High	High	Broken Access Control	★★
Bonus Payload	Medium	Low	Low	XSS	★
Bully Chatbot	Critical	Medium	Medium	Miscellaneous	★
Confidential Document	High	High	High	Sensitive Data Exposure	★
DOM XSS	Medium	Medium	Medium	XSS	★
Privacy Policy	Critical	Low	Low	Miscellaneous	★
Score Board	Medium	Low	Low	Miscellaneous	★
Zero Stars	Medium	Low	Low	Improper Input Validation	★

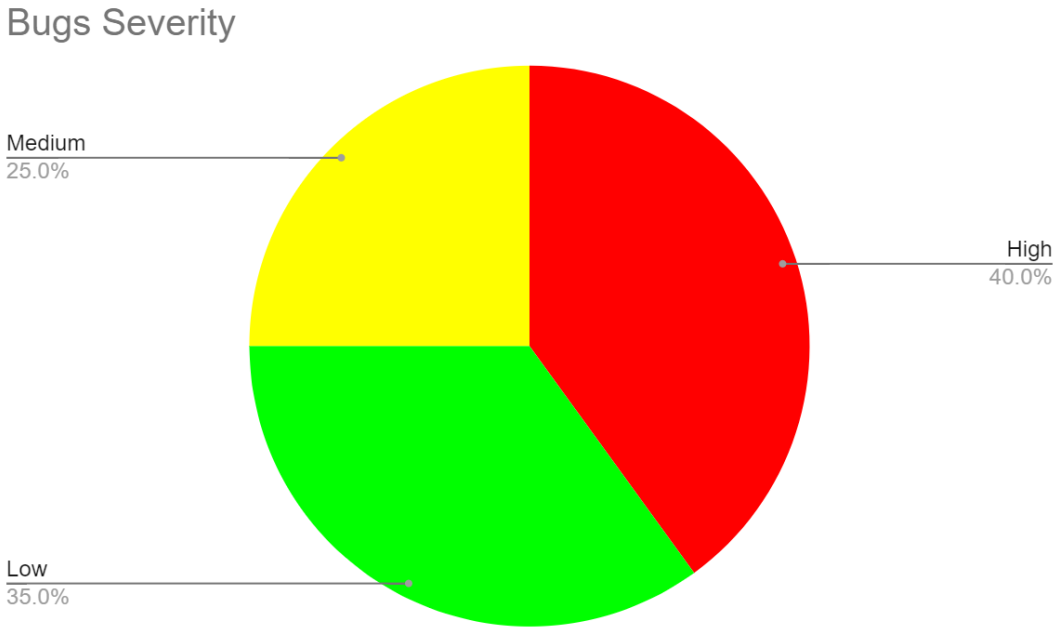
5.0 – VULNERABILITY & REMEDIATION REPORT

Penetration test finding classification, description and recommendations mentioned in the report are taken mostly from OWASP TOP 10 project documentation available on site: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project.

The OWASP TOP 10 is a list of definitions of web application vulnerabilities that pose the most significant security risks to organizations when exploited.

Vulnerability Summary





6.0 – INFORMATION GATHERING

The information gathering portion of a penetration test focuses on identifying the scope of the penetration test. I started the pentest by finding all subdomains.

I have used multiple tools to make sure that I haven't missed any domain.

Tools Used:

1. wfuzz
2. Sublist3r
3. nikto

Above tools discovered multiple subdomains, the working one's are below:

No Subdomains were found

7.0 High Findings

7.1. Title: Change Bender's Password

Rating: **High**

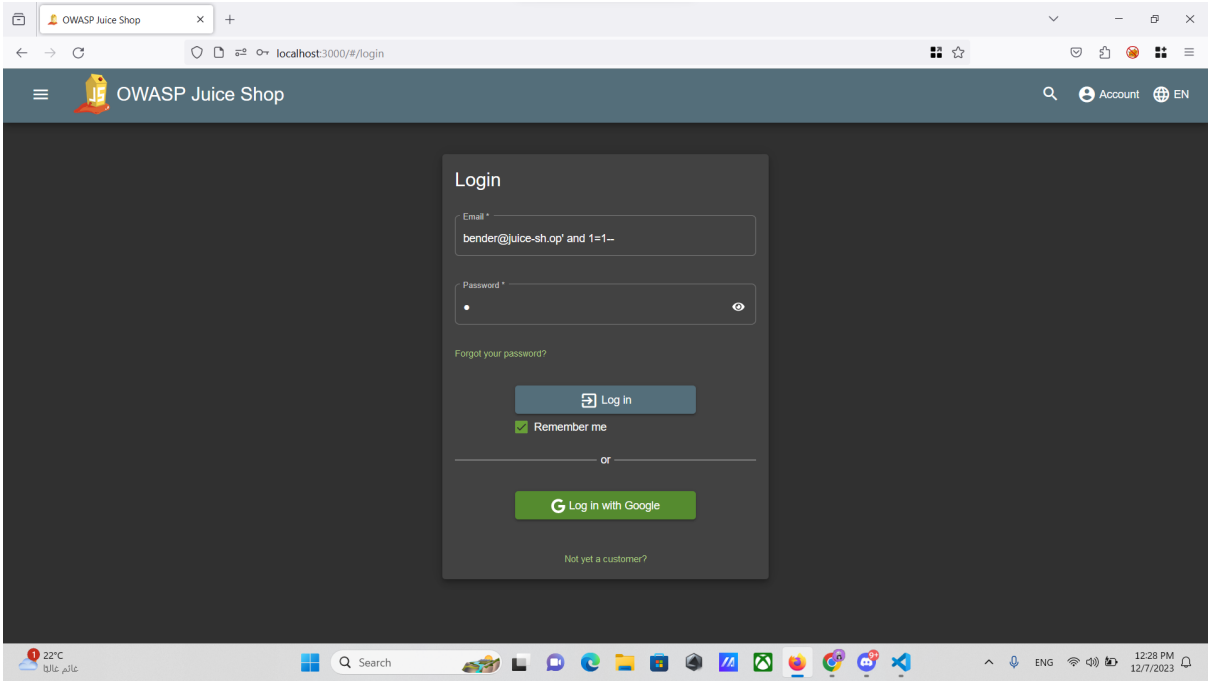
URL: http://localhost:3000/#/privacy-security/change-password

Description: Change Bender's password into slurmCl4ssic without using SQL Injection or Forgot Password.

Proof of Concept:

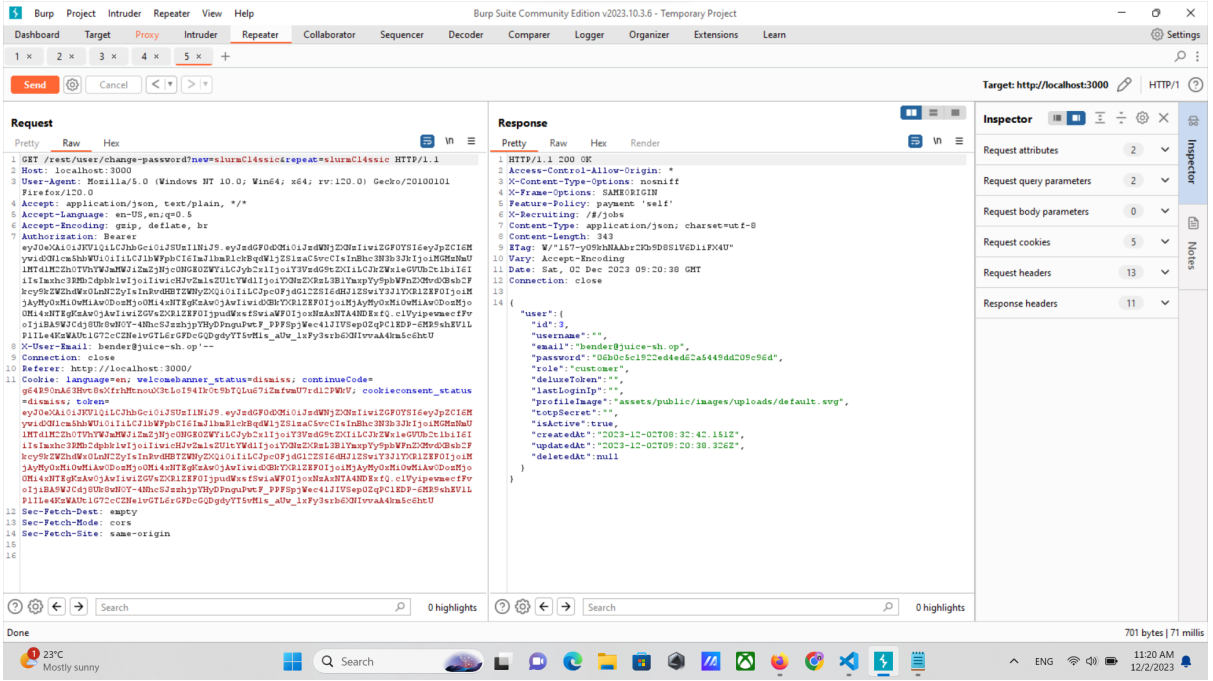
Step 1 : login with Bender's credentials by this payload in email field : bender@juice-sh.op' and 1=1–

and password = anything

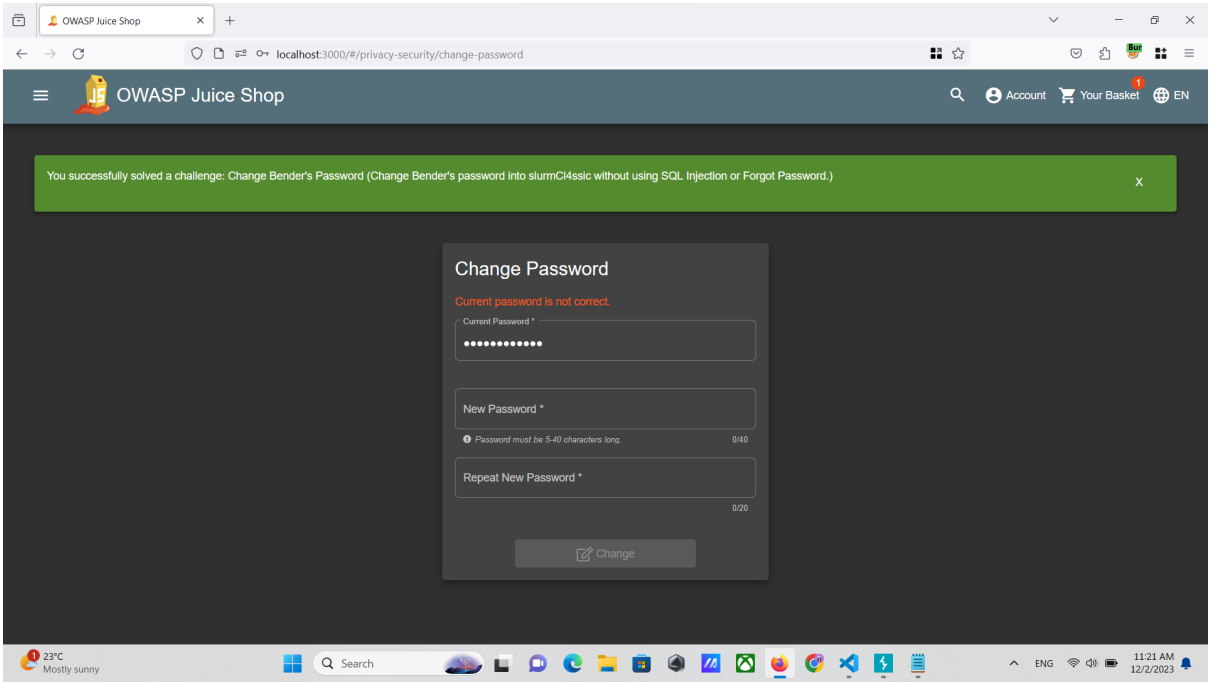


Step 2: browse to reset password page

Step 3: enter any values for all current password , new password and repeat new password
Step 4: intercept with burp that request and delete current attribute in the request



Step 5: as you see in picture he accepted the request because he perform a only client side check on current password value



Remediation Steps:

only perform server side extra checks on the current password value.

7.2. Title: Email Leak

Rating: **High**

URL: http://localhost:3000/login

Description: Perform an unwanted information disclosure by accessing data cross-domain.

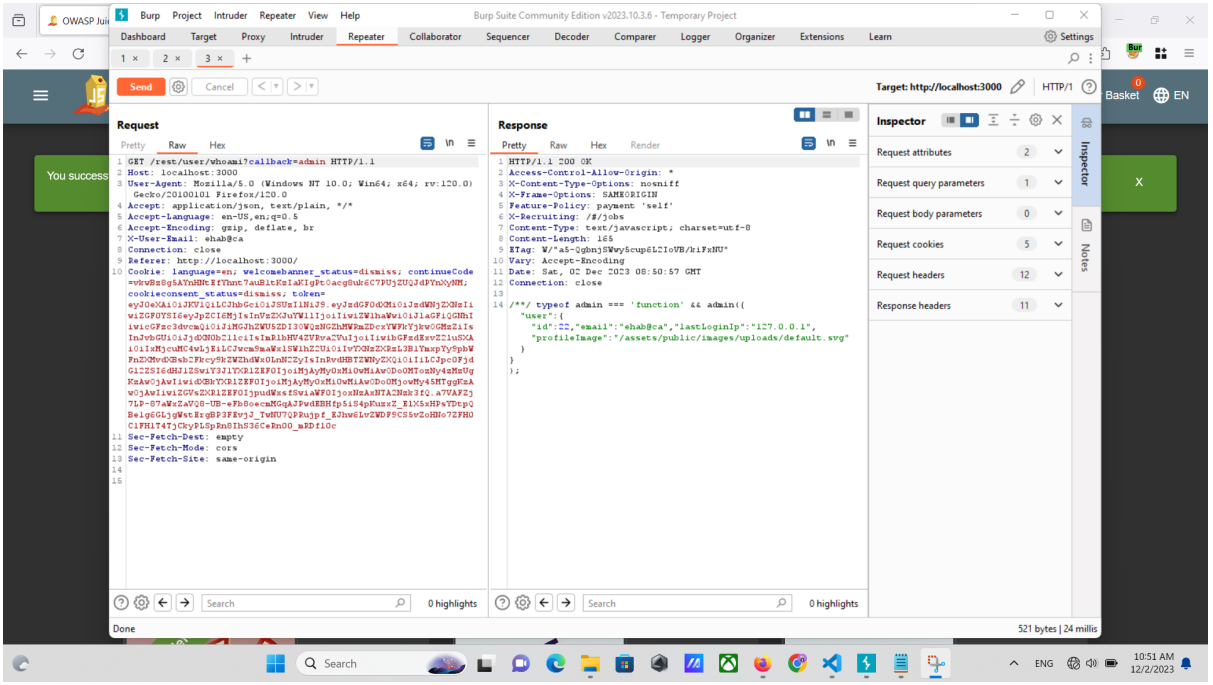
Proof of Concept:

Step 1 : browse login page

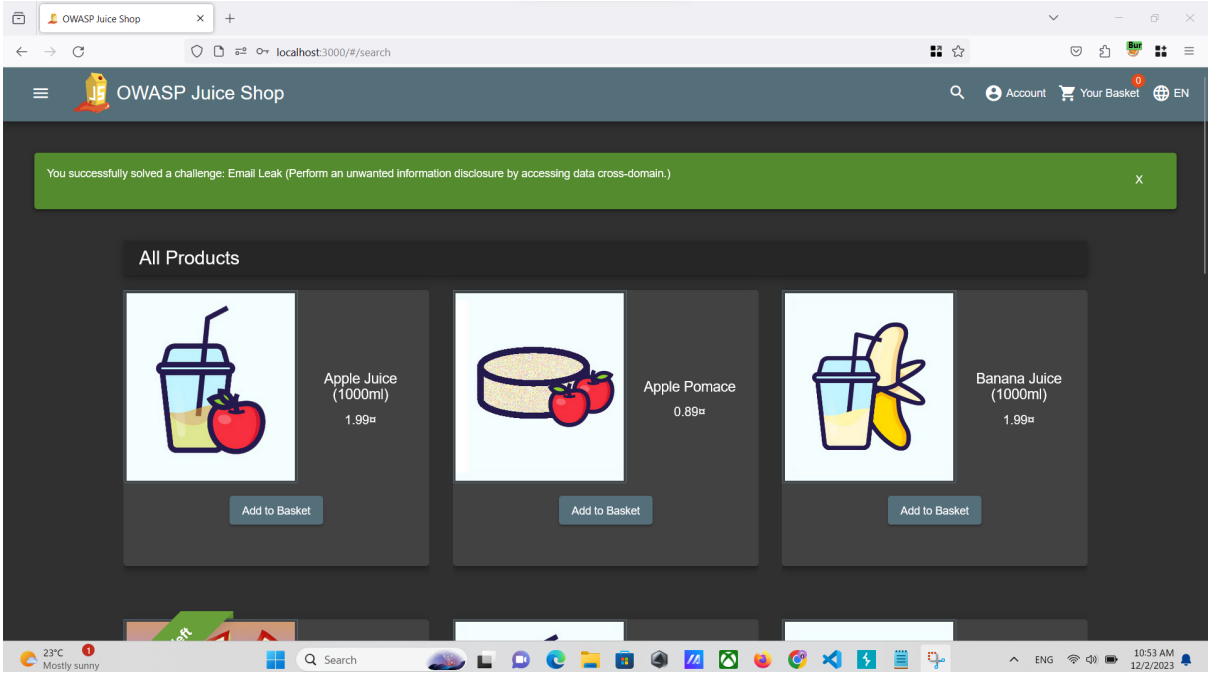
Step 2: enter your credentials

Step 3: intercept the login request with burp
Step 4: keep forwarding one by one until you find get request including your token and request line includes whoami

Step 5: delete the authorization Bearer and modify request line to perform callback by adding to resource ?callback=admin to get admin information



Step 6: as you see in picture he accepted the request because he accept request missing authorization Bearer



Remediation Steps:
make server reject login request missing authorization Bearer.

7.3. Title: User Credentials

Rating: High

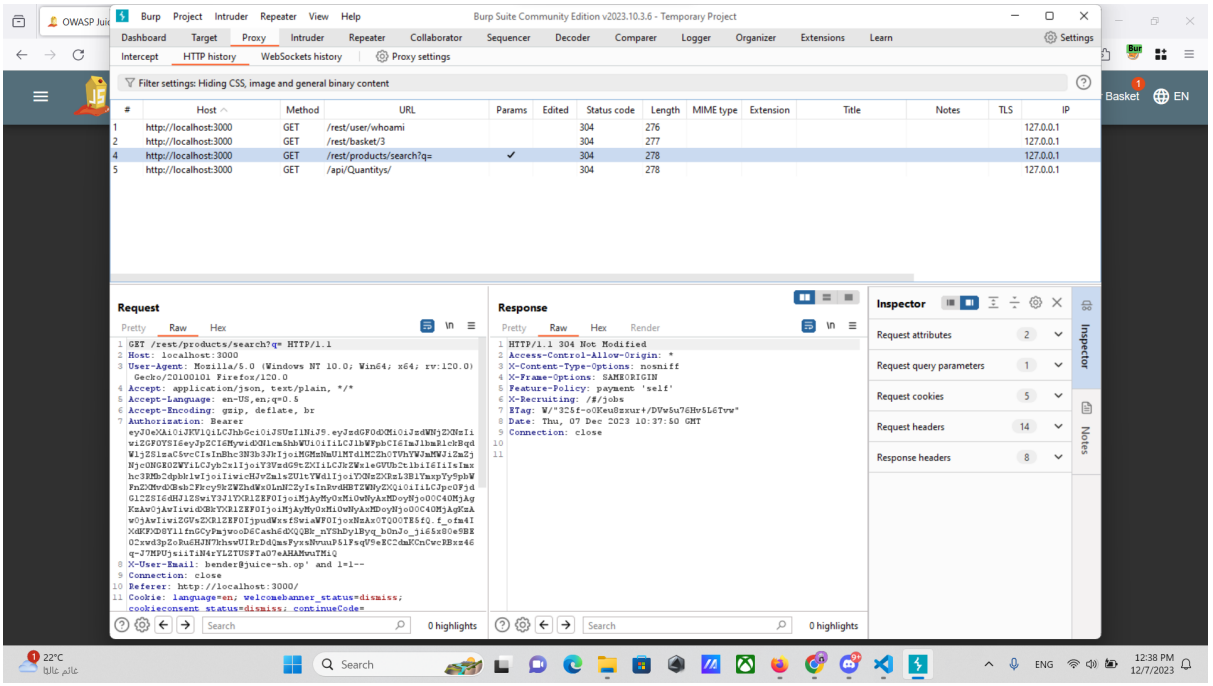
URL:
http://localhost:3000/rest/products/search?q=%’))union+select+username,email,password,role,deluxeToken,lastLoginIp,profileImage,topSecret,isActive+from+users--

Description: Retrieve a list of all user credentials via SQL Injection.

Proof of Concept:

Step 1 : search for any thing like orange

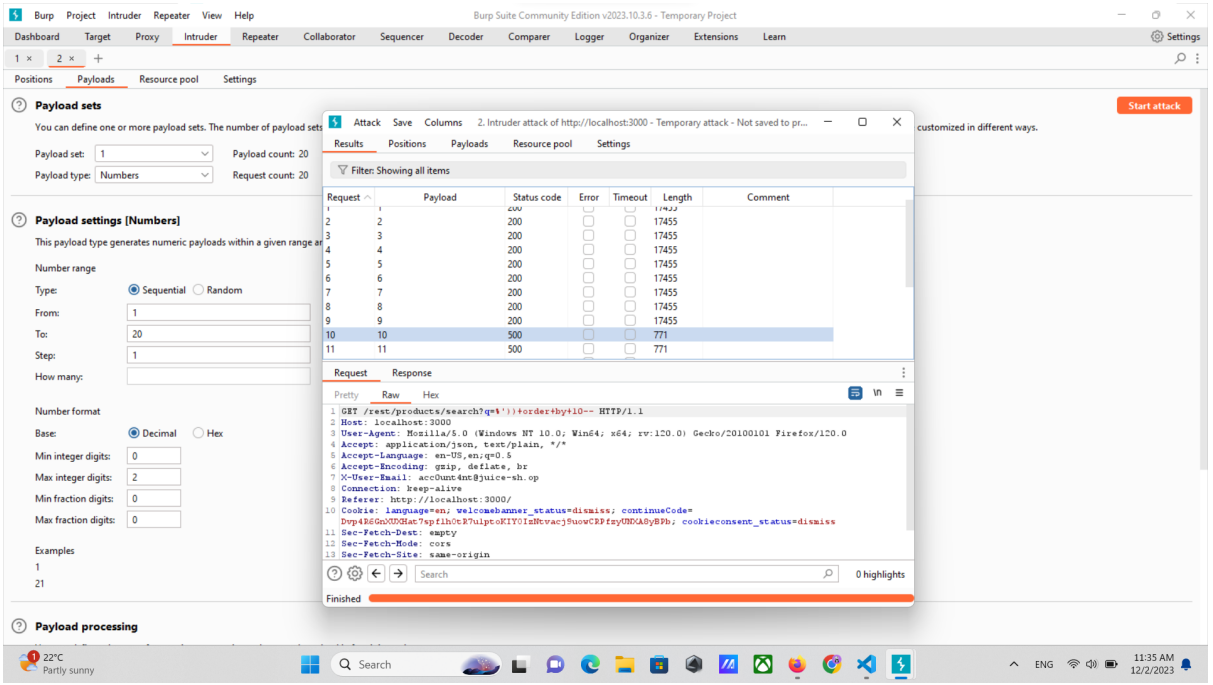
Step 2: look in http history of burb for request of the search query



Step 3: send request to repeater and test for sql injection using this payload %’))--

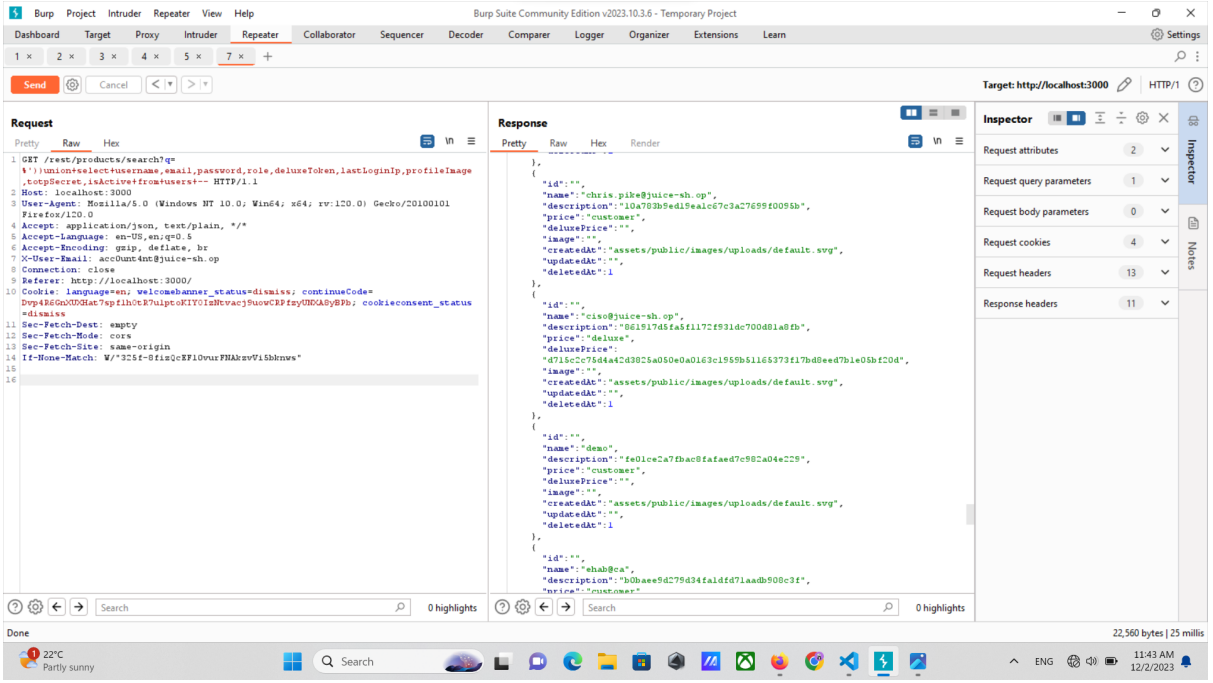
Step 4 :Find out how many parameters are returned by the sql query by using order by technique

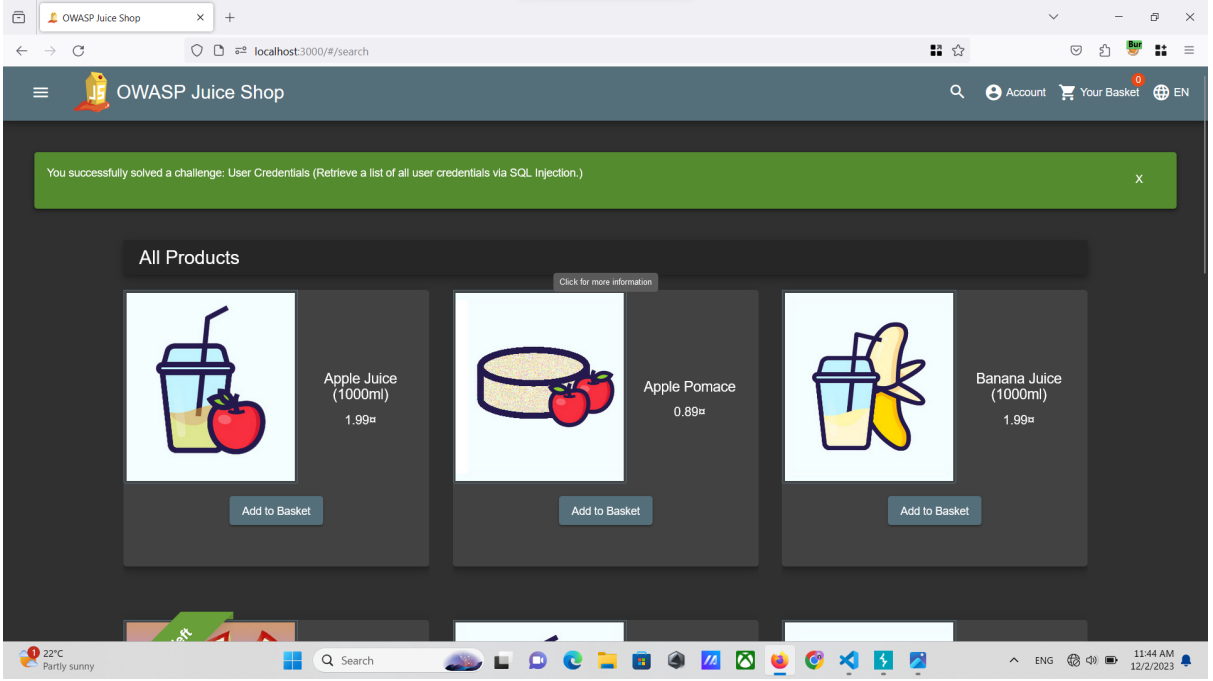
using intruder



Step 5: after finding how many attributes are returned we can guess attributes names and find some attributes name from other api requests

Step 6:add this payload to resource in the request line
%’))union+select+username,email,password,role,deluxeToken,lastLoginIp,profileImage,topSecret,isActive+from+users--





Remediation Steps:

you can solve this vulnerability by escaping the special characters that lead to sql injection

7.4. Title: CAPTCHA Bypass

Rating: **High**

URL: http://localhost:3000/api/feedbacks/

Description: Attacker can Submit 10 or more customer feedbacks within 20 seconds.

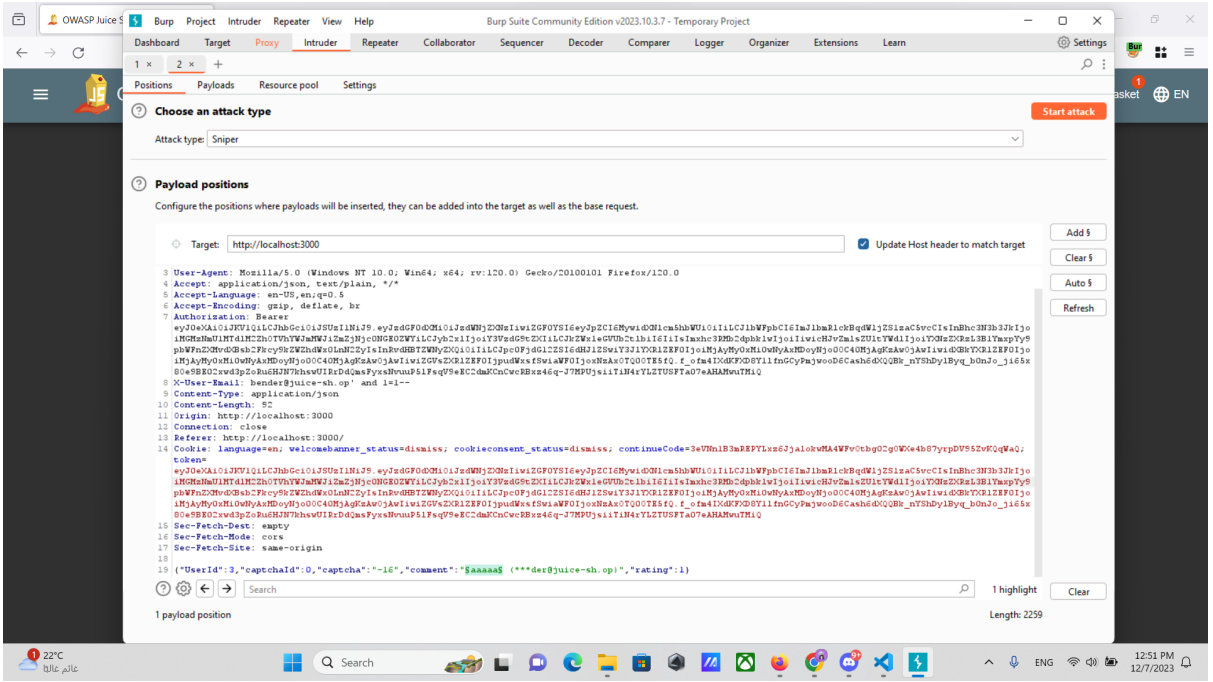
Proof of Concept:

Step 1 : login with your credentials

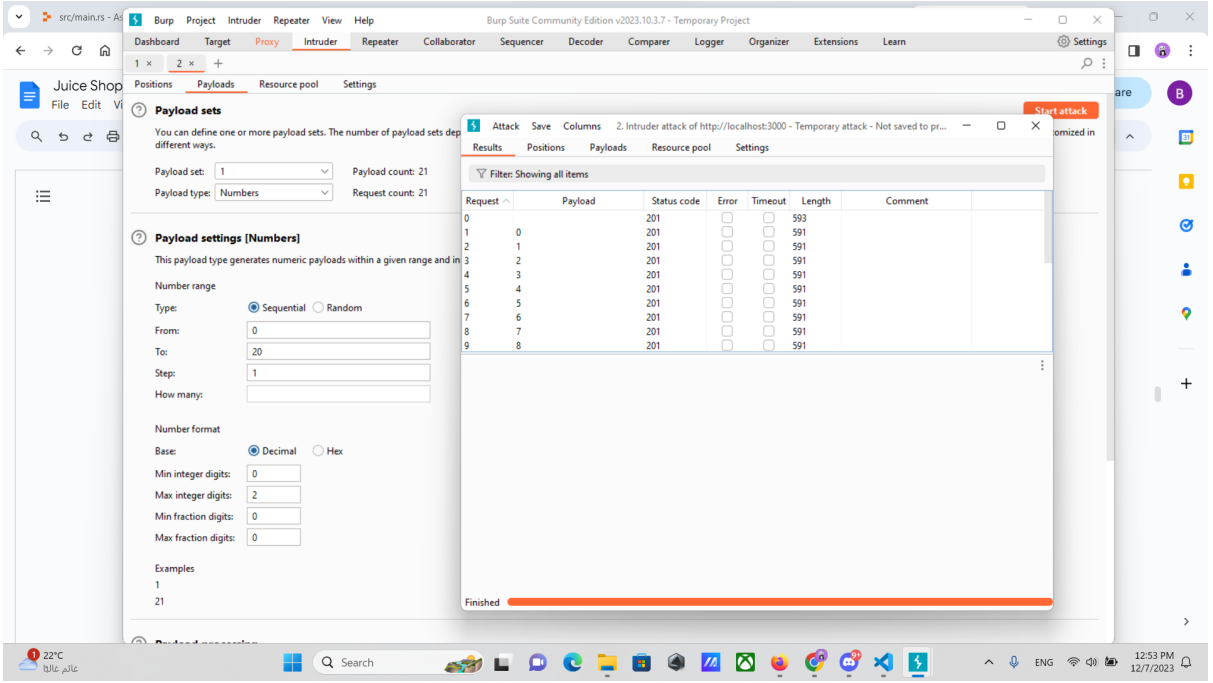
Step 2: browse to customer feedback

Step 3: enter feedback but intercept it in the burp

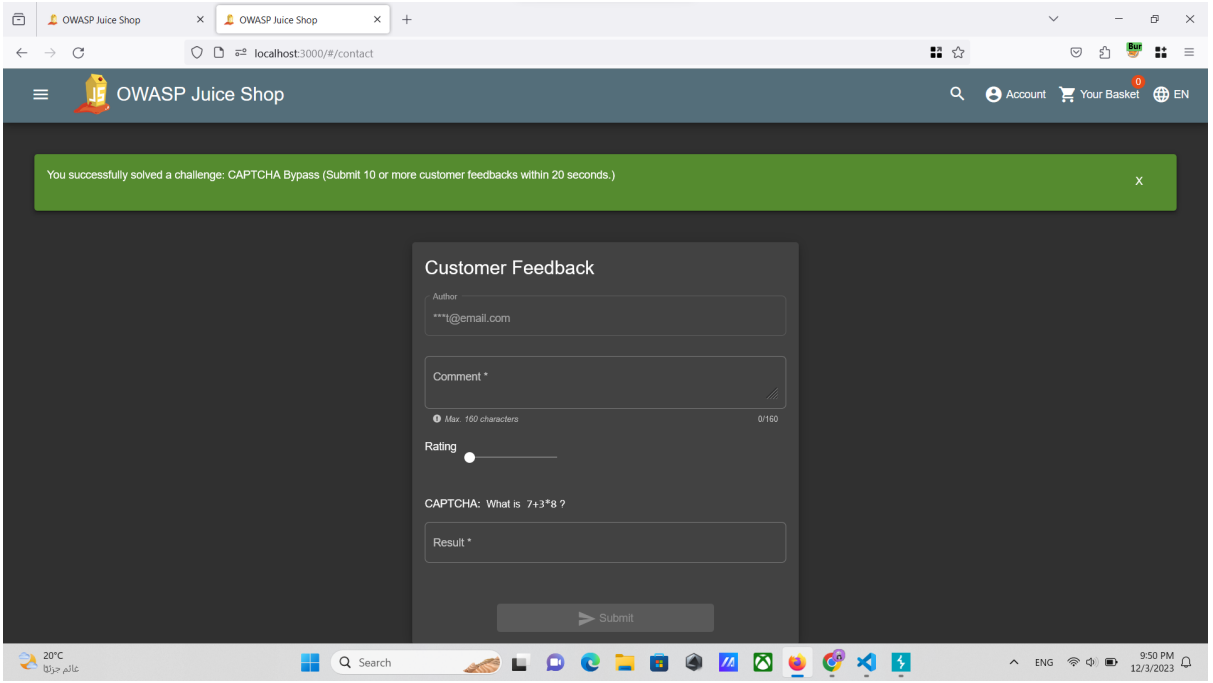
Step 4: send the feedback request to intruder and set the position is comment section value



step 5: go to payloads give the payloads option 11 different values



step 6: start the attack and now you will find the lab solved as in the below picture



Remediation Steps:

Consider using more advanced captcha technologies such as Google's reCAPTCHA or hCaptcha. These services employ advanced algorithms to distinguish between human and automated interactions.

7.5. Title: Login Bender

Rating: **High**

URL: http://localhost:3000/login

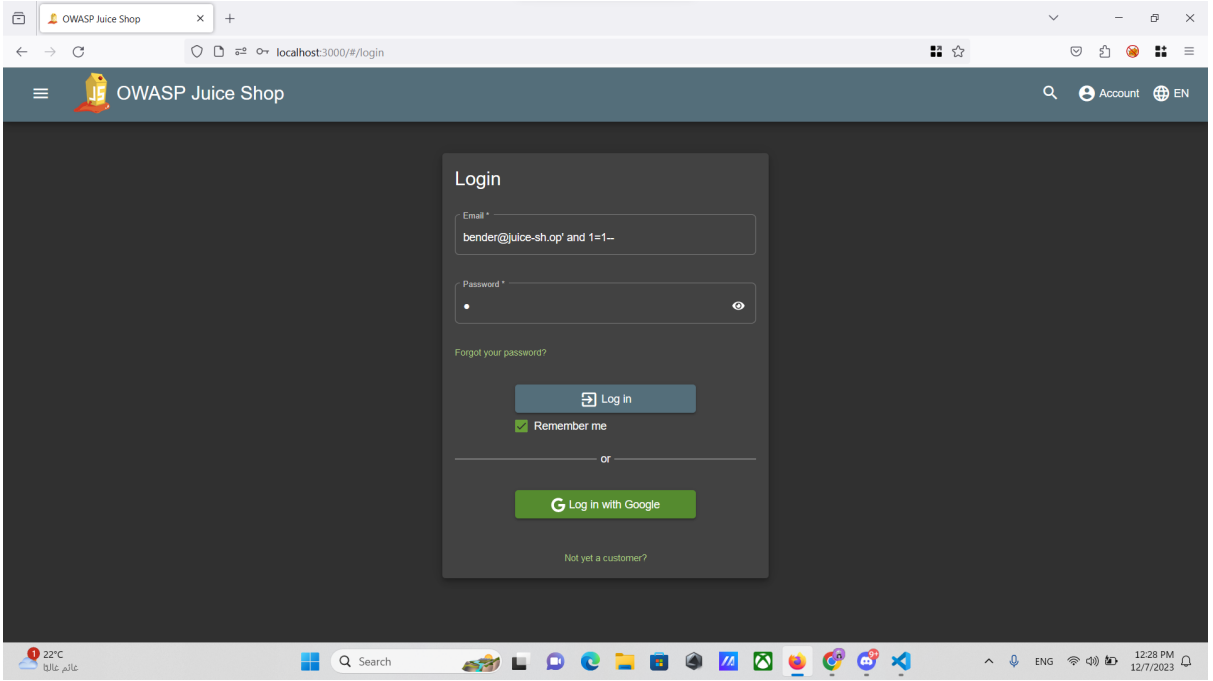
Description: Log in with Bender's user account

Proof of Concept:

Step 1 : browse to login page

Step 2: enter benders email which is bender@juice-sh.op and add this payload to the mail ' and 1=1– which makes the value entered as mail is bender@juice-sh.op' and 1=1–

Step 3: enter any password value



Step 4: click on login button and you would be logged in bender account

Remediation Steps:
you can solve this vulnerability by escaping the special characters that lead to sql injection

7.6. Title: Login Admin

Rating: **High**

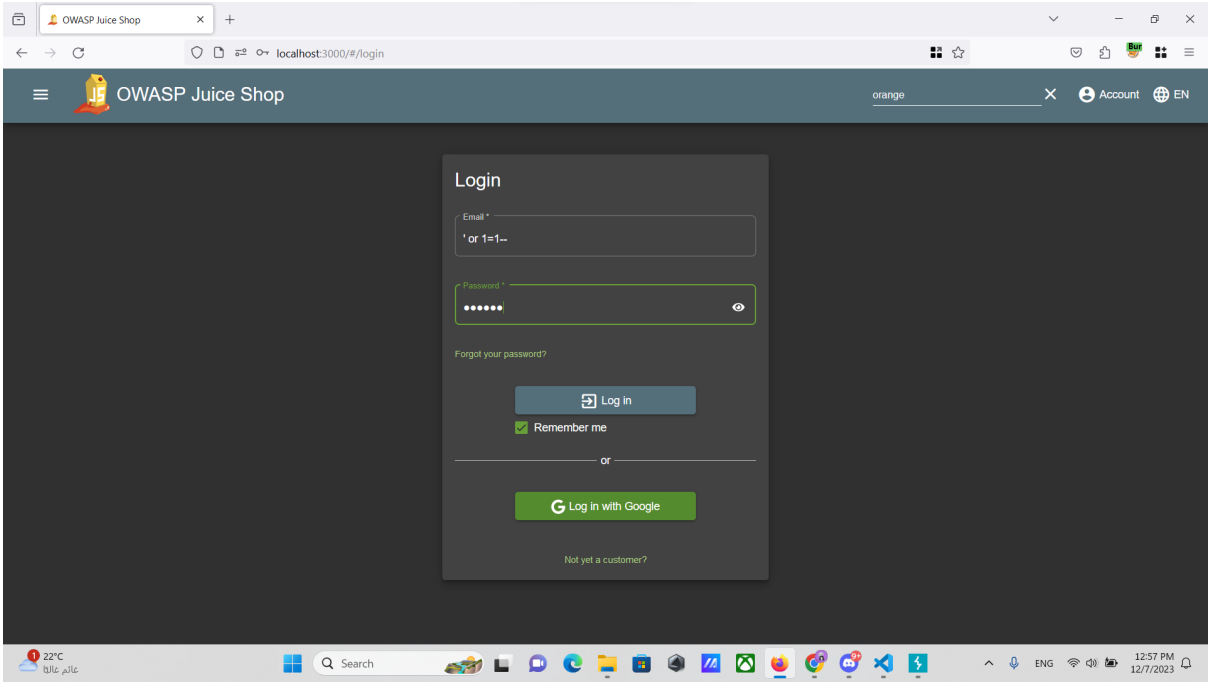
URL: http://localhost:3000/login

Description: Log in with the administrator's user account.

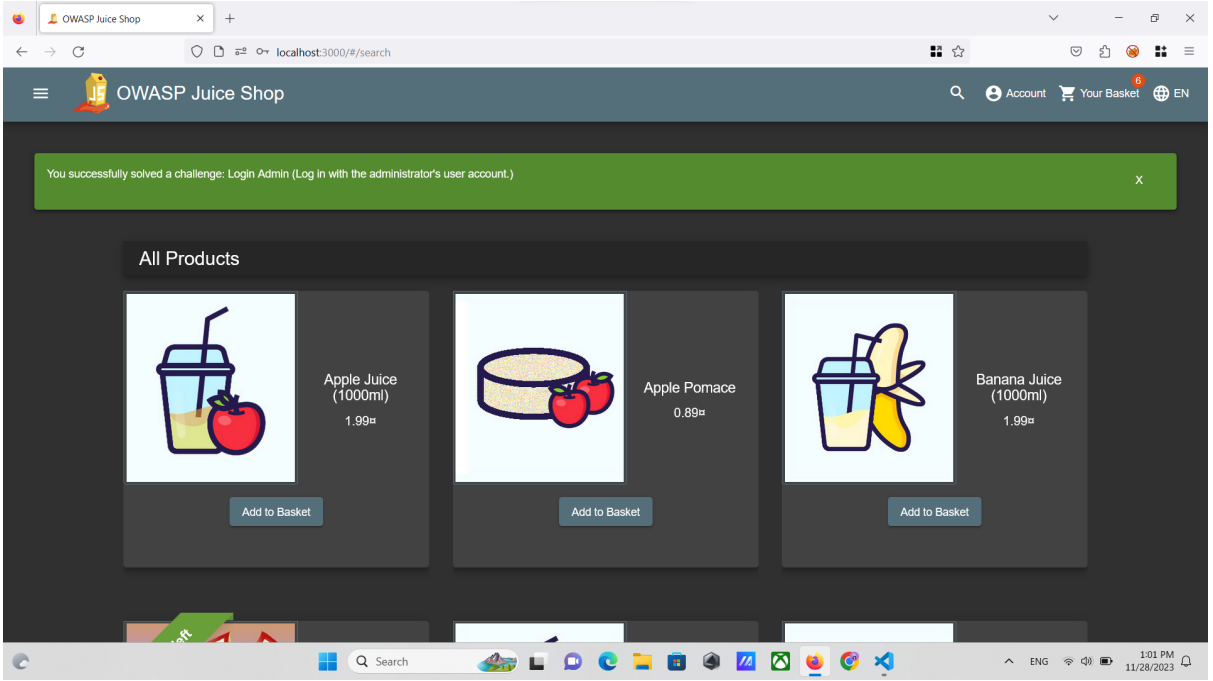
Proof of Concept:

Step 1 : browse to login page

Step 2: enter this payload in email field and password field ‘OR 1=1 -- -



Step 3: click on login button and you would be logged in admin account as in below picture



Remediation Steps:
you can solve this vulnerability by escaping the special characters that lead to sql injection

7.7. Title: View Basket

Rating: **High**

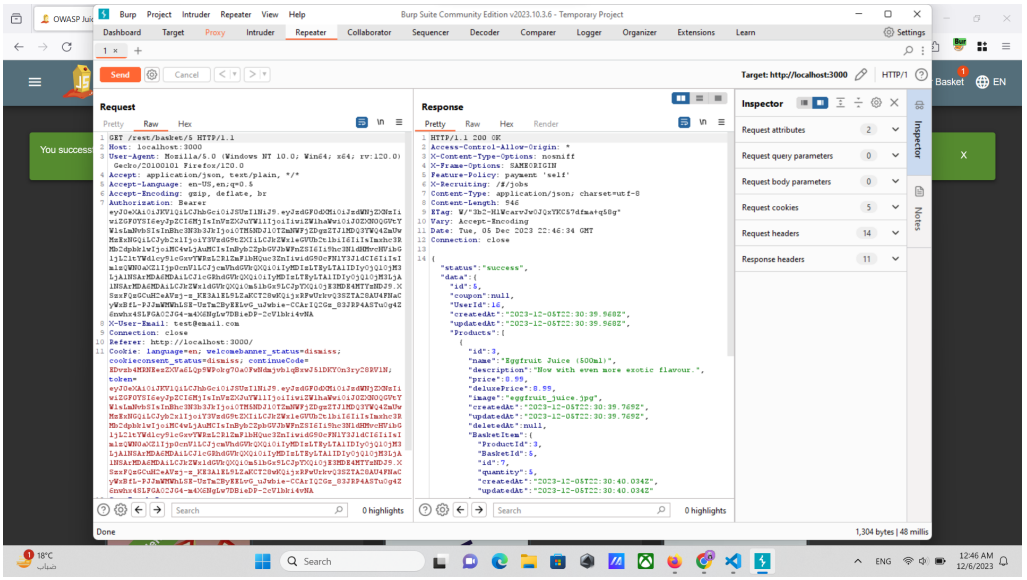
URL: http://localhost:3000/#/

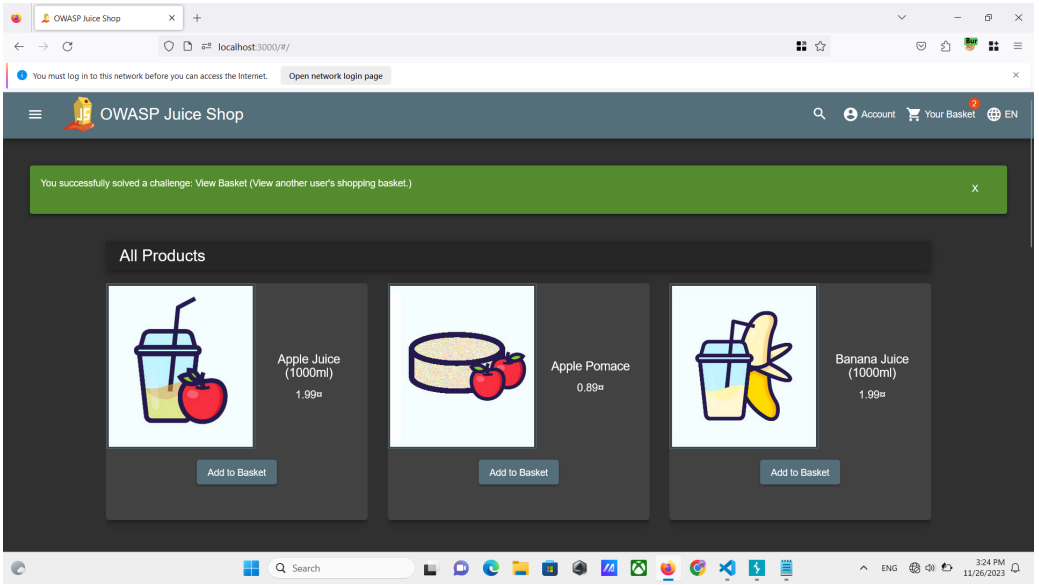
Description: Attacker can get details of any user Basket contents.

Proof of Concept:

Step 1 : Add any item to your basket then intercept the request that gets the basket detail.

Step 2 : Change the basket id and forward the request.





Remediation Steps:
Validate in the server that the Basket is owned by the user who made the request

7.8. Title: Confidential Document

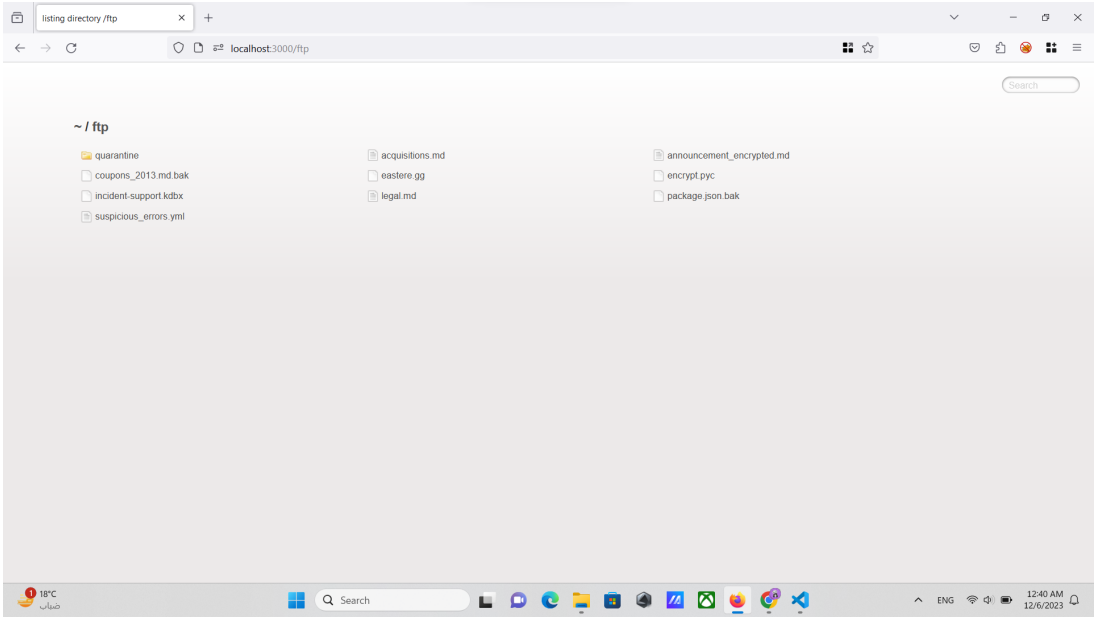
Rating: **High**

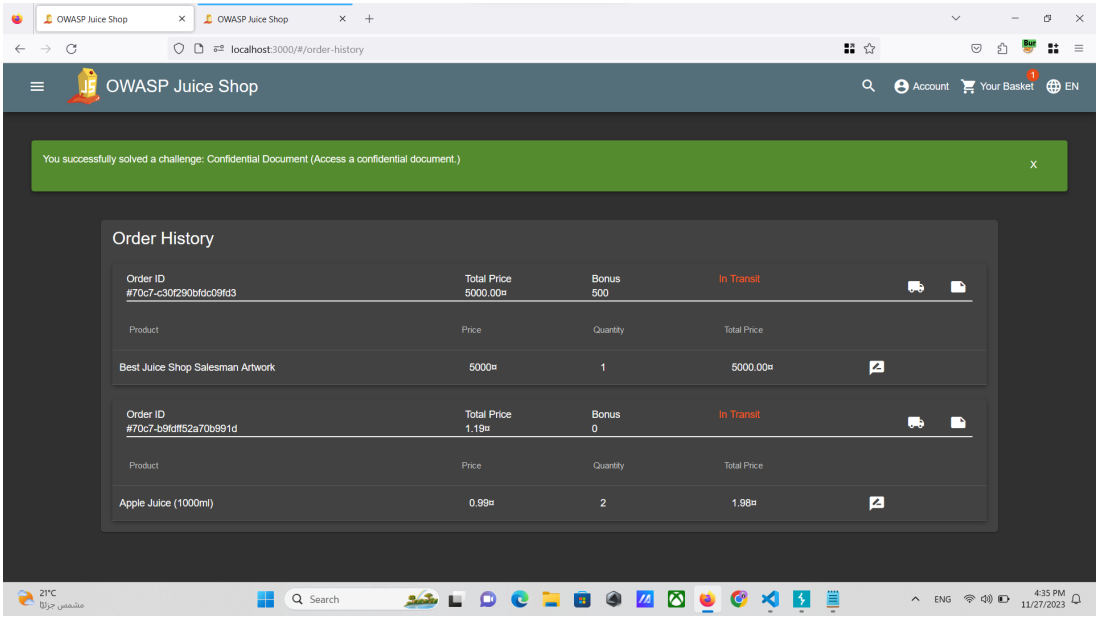
URL: http://localhost:3000/ftp/acquisitions.md

Description: Attacker can access ftp server and read confidential documents

Proof of Concept:

Step 1 : Brute force directories till you find ftp





Remediation Steps:
Make the ftp require authentication

8.0 Medium Findings

8.1. Title: Deluxe Fraud

Rating: Medium

URL: http://localhost:3000/#/payment/deluxe

Description: Obtain a Deluxe Membership without paying for it.

Proof of Concept:

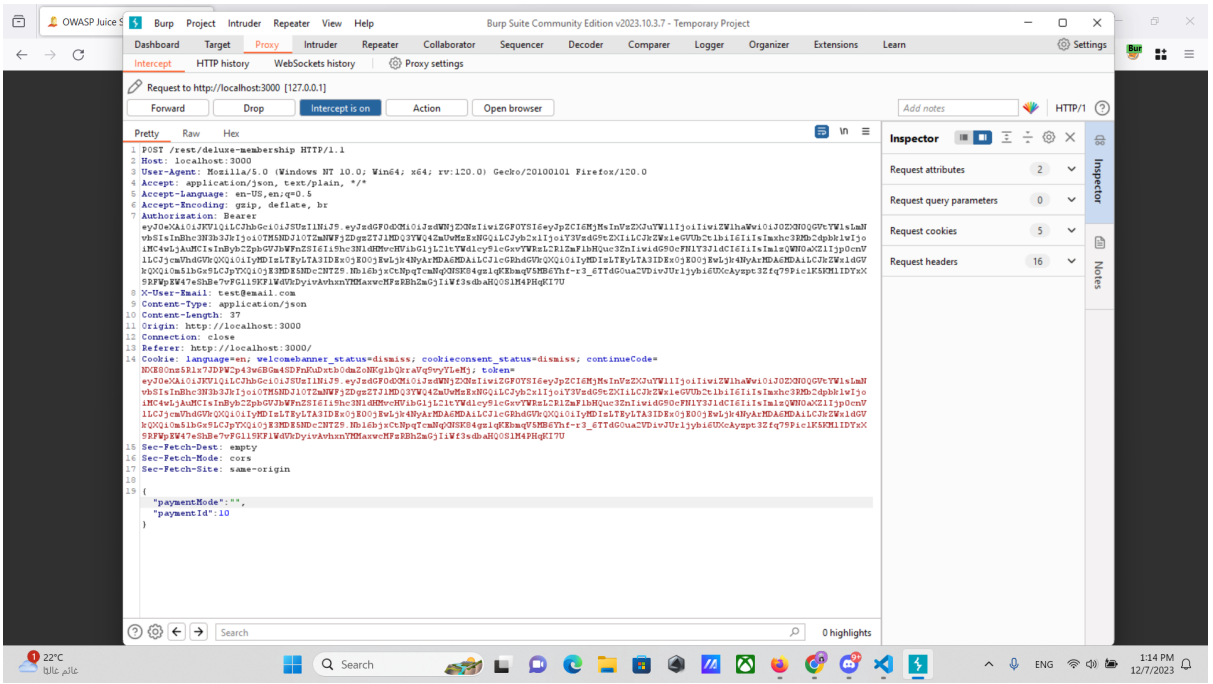
Step 1: login with your Credentials

Step 2: browse to Deluxe membership

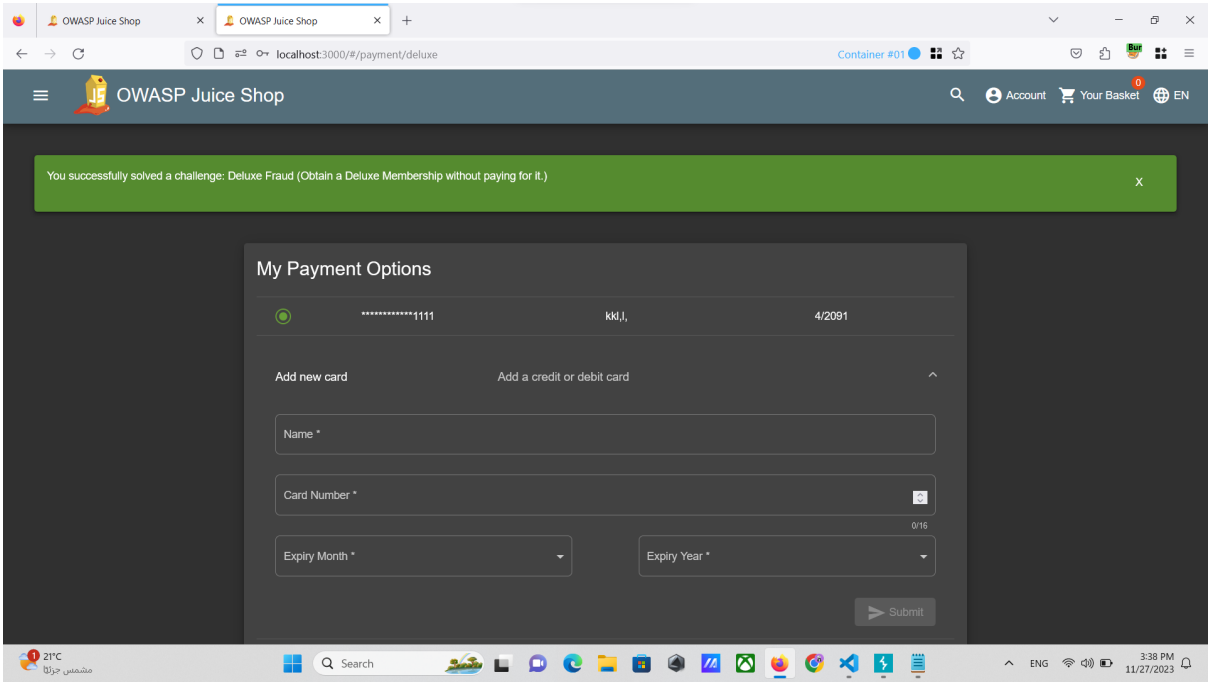
Step 3: press become member button

Step 4: right click on pay 49.00 button

Step 5: Intercept the request and set payment mode to ""



Step 6: Forward the request



Remediation Steps:

block client side from sending request paid handle paid case in the server side

8.2. Title: Forged Feedback

Rating: Medium

URL: http://localhost:3000/api/feedbacks/

Description: Post some feedback in another user's name.

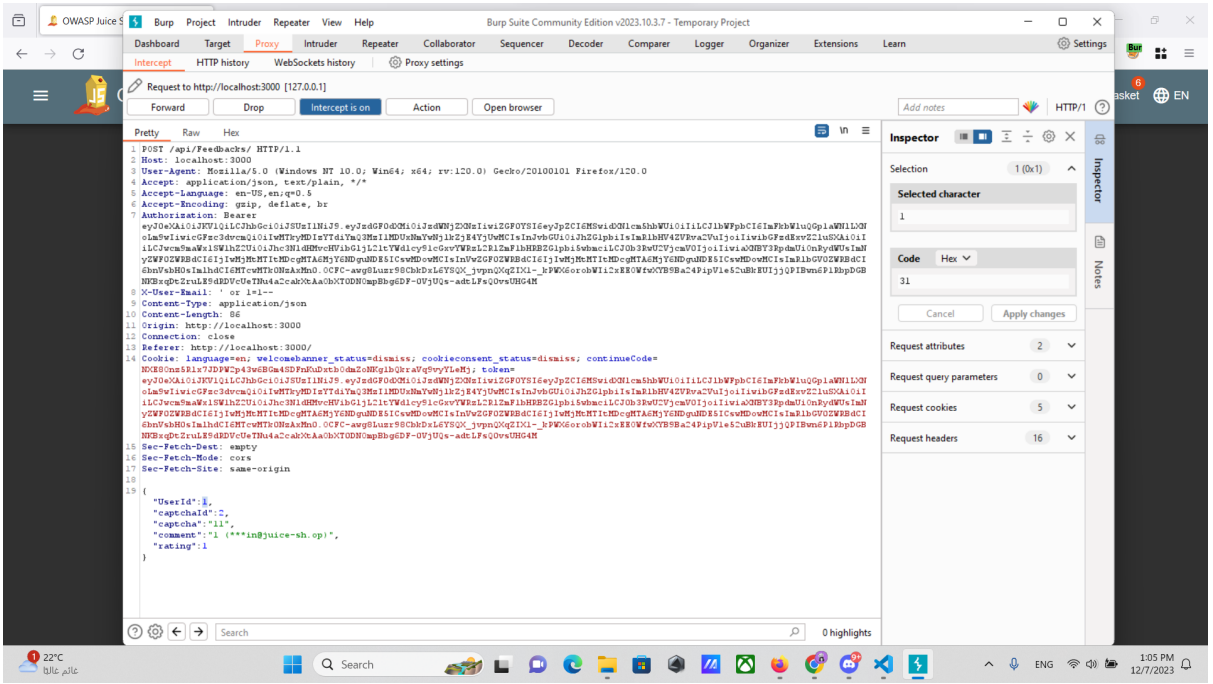
Proof of Concept:

Step 1: login with your Credentials

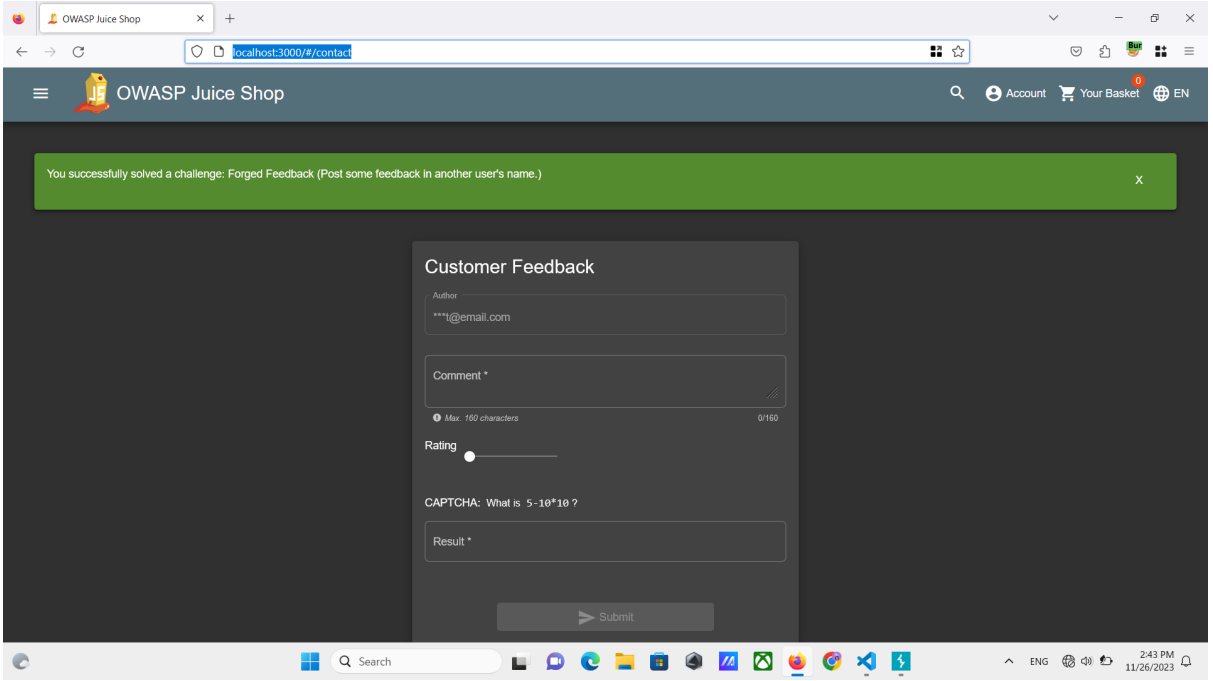
Step 2: browse to feedback section

Step 3: enter feedback values and send it but intercept it in burp

Step 4: change UserId value in json object to the id of the user who wants him to send the feedback.



Step 5: and it will be accepted as in the below picture



Remediation Steps:

server side check that id of the current logged in user is the same id sending feedback

8.3. Title: Forged Review

Rating: Medium

URL: http://localhost:3000/rest/products/1/reviews

Description: Post a product review as another user or edit any user's existing review.

Proof of Concept:

Step 1: login with your Credentials

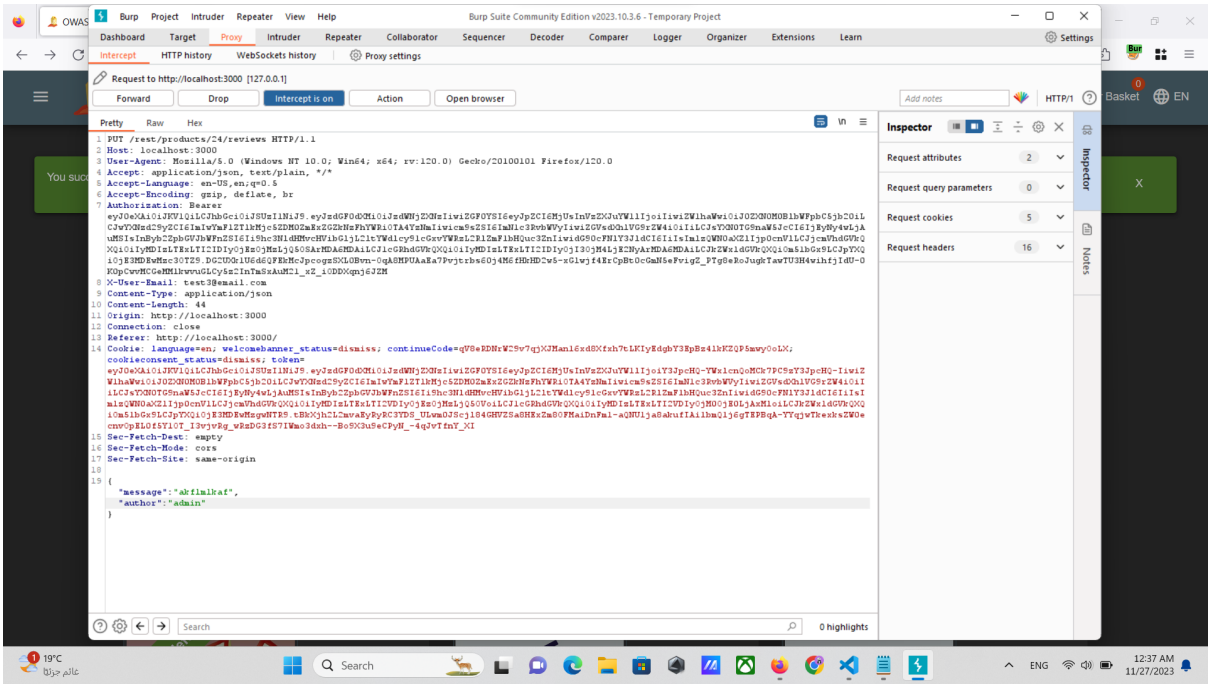
Step 2: click on any product in dashboard

Step 3: activate intercept in burp

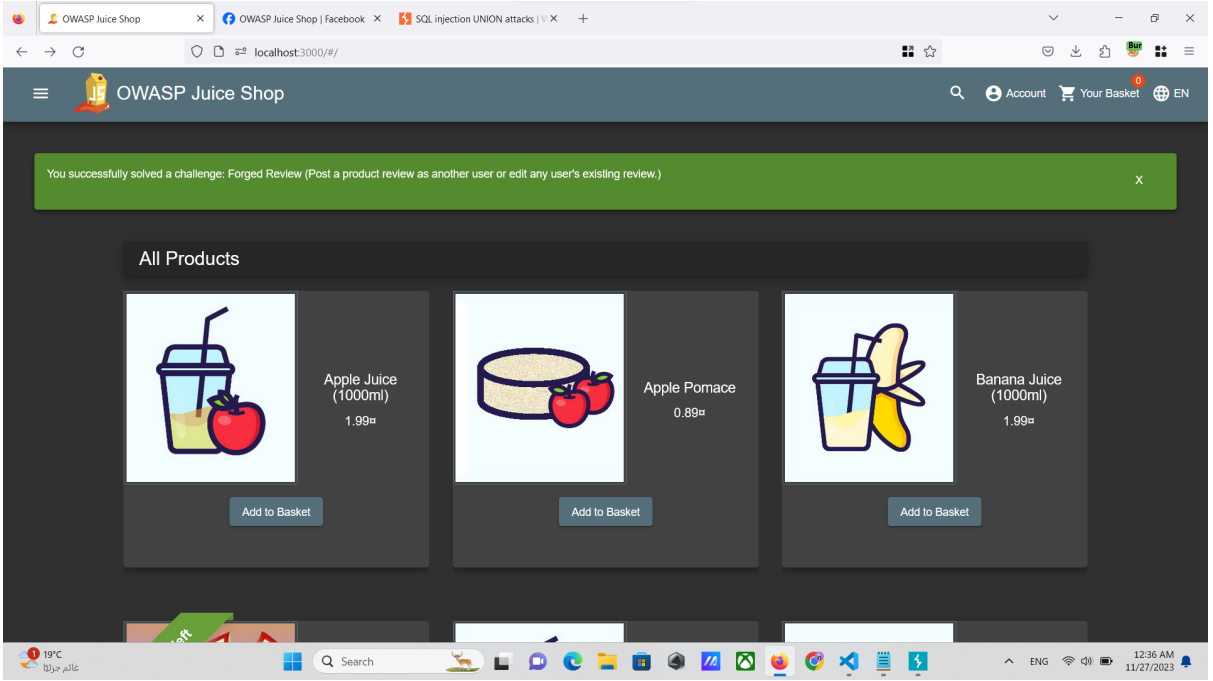
Step 4: go back to site enter review value such as good product and click send button

Step 5: send intercepted request to repeater

Step 6: change value of the author in JSON object to the author that want him to send the review



Step 7: and send modified request and it the response will be accepted



Remediation Steps:

server side check that author of the current logged in user is the same author sending review

8.4. Title: Bully Chatbot

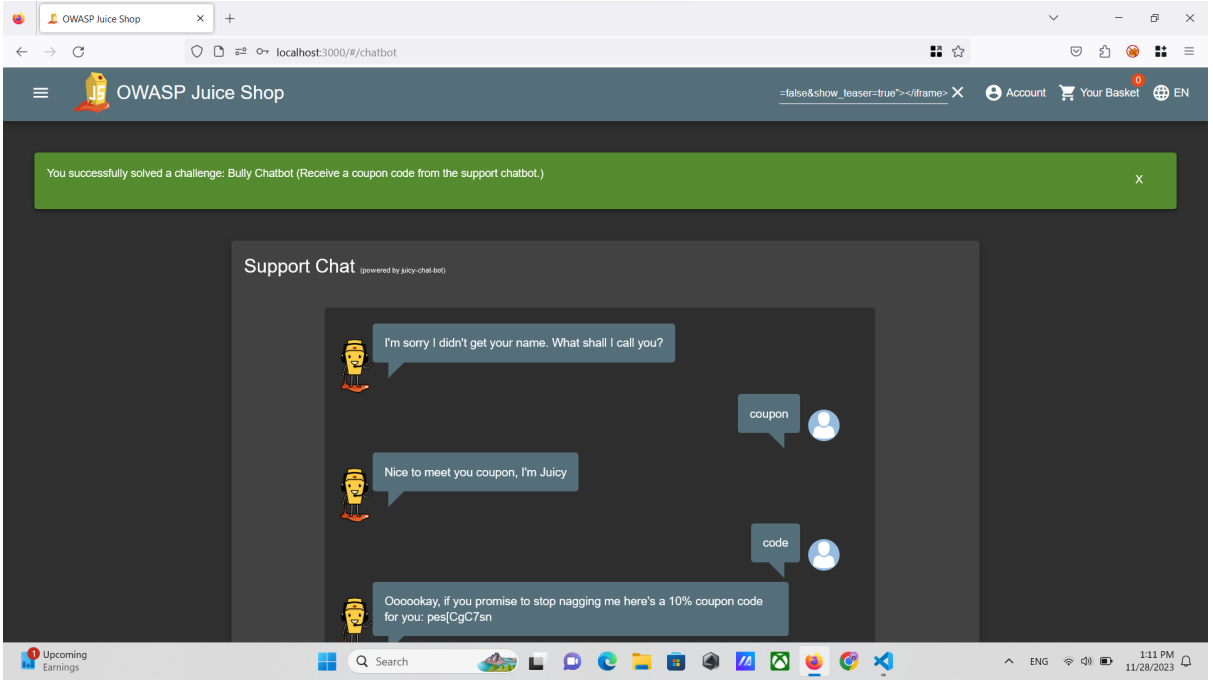
Rating: Medium

URL: http://localhost:3000/#/chatbot

Description: Attacker can keep spamming the bot the word "coupon" and eventually the bot gives a promotion code to the attacker

Proof of Concept:

Step 1: Go to the chatbot page and keep spamming the word "coupon".



Remediation Steps:

prevent the chatbot from accessing sensitive data such as promotion codes

8.5. Title: DOM XSS

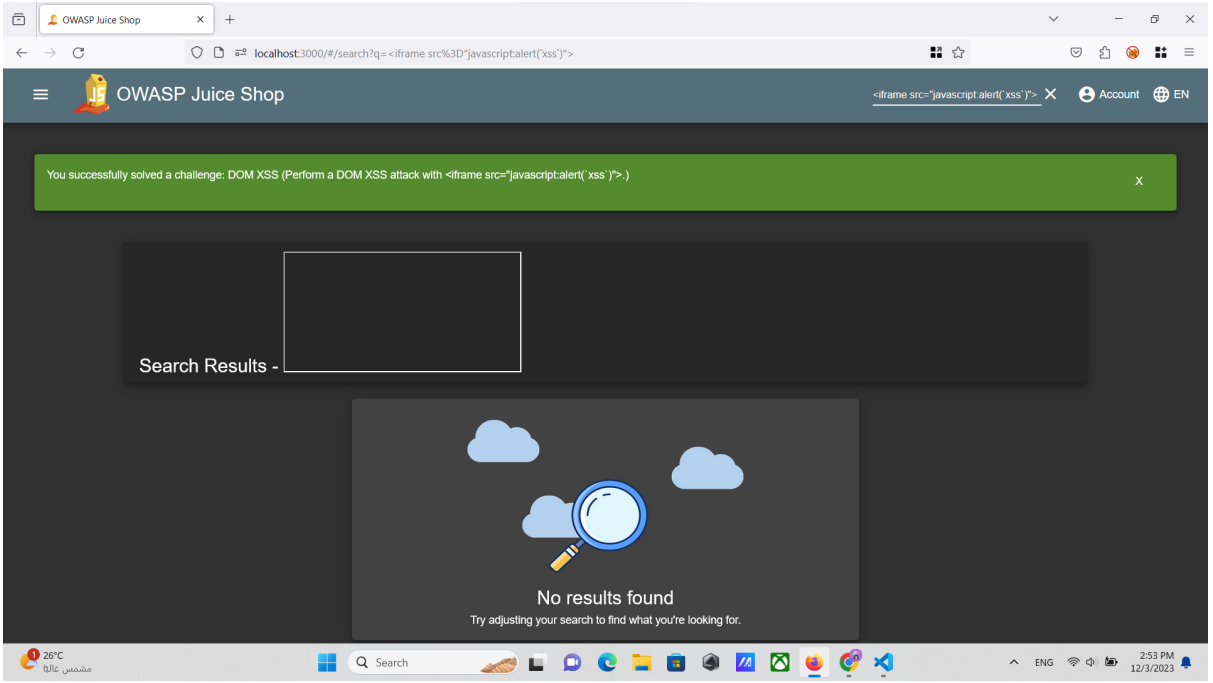
Rating: Medium

URL: http://localhost:3000/#/

Description: Attacker can perform DOM XSS attack in search field.

Proof of Concept:

Step 1: Use this script in the search field: <iframe src="javascript:alert('xss')">



Remediation Steps:

Sanitize inputs in the server side form control characters of javascript.

9.0 Low Findings

9.1. Title: Christmas Special

Rating: Low

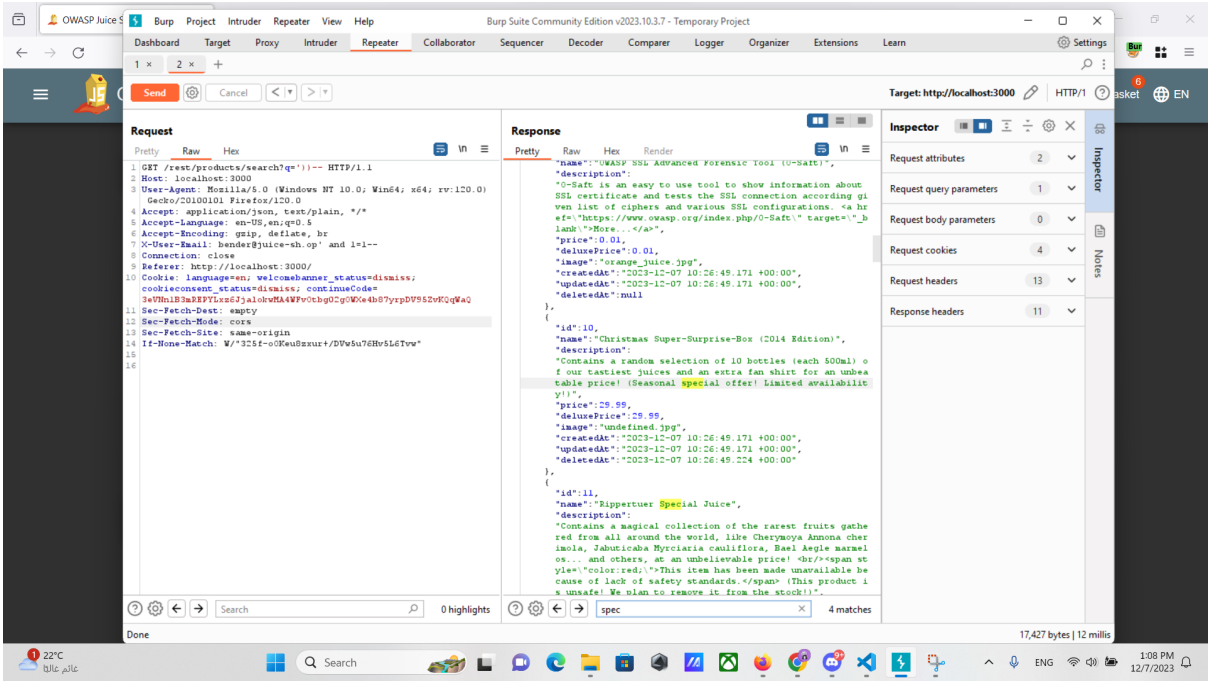
URL: http://localhost:3000/#/search?q=’))--

Description : Order the Christmas special offer of 2014.

Proof of Concept:

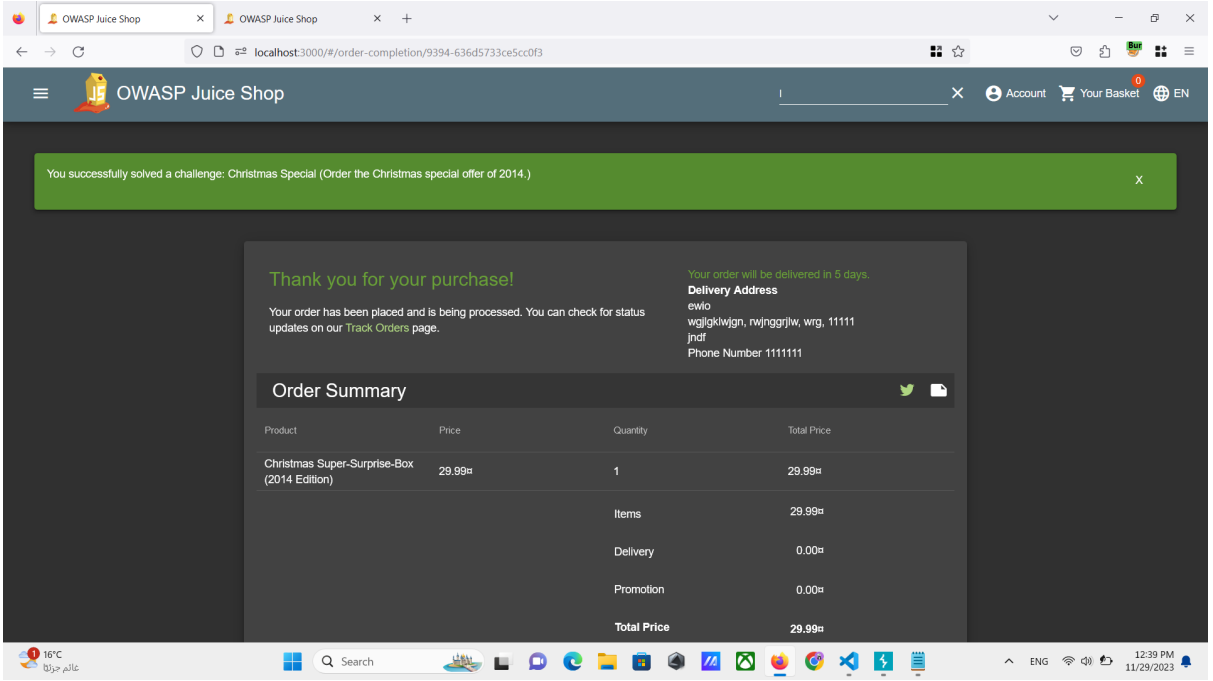
Step 1 : open dashboard

Step 2 : enter this payload in search field ’))--



Step 3 : it will reveal Christmas special offer of 2014.

Step 4 : Now you can order it as you see in the below picture.



Remediation Steps:

you can solve this vulnerability by escaping the special characters that lead to sql injection

9.2. Title: Empty User Registration

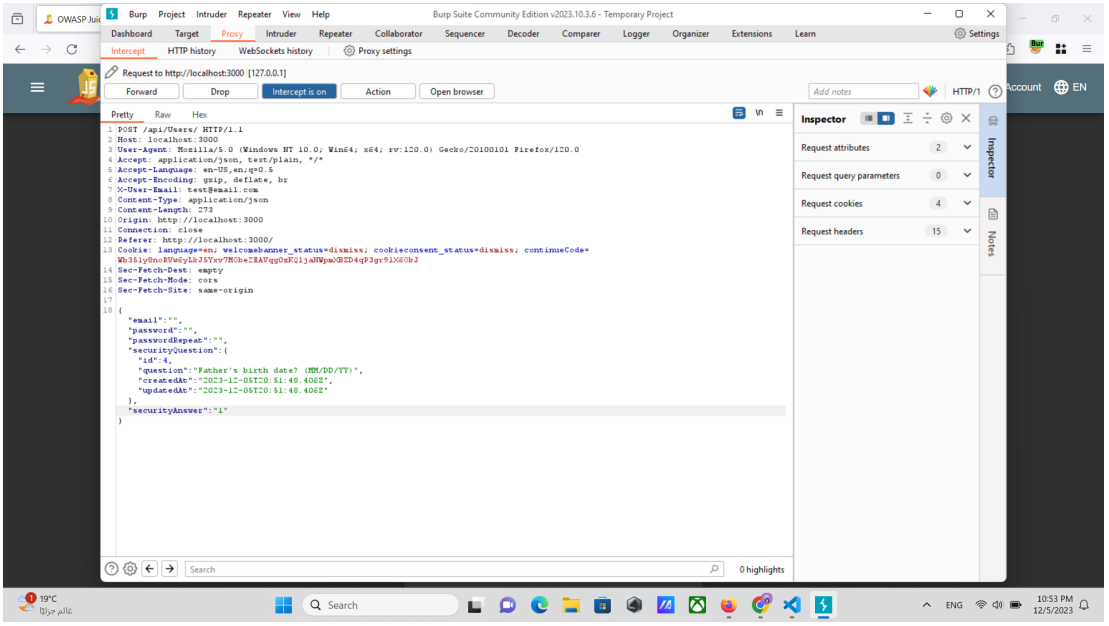
Rating: Low

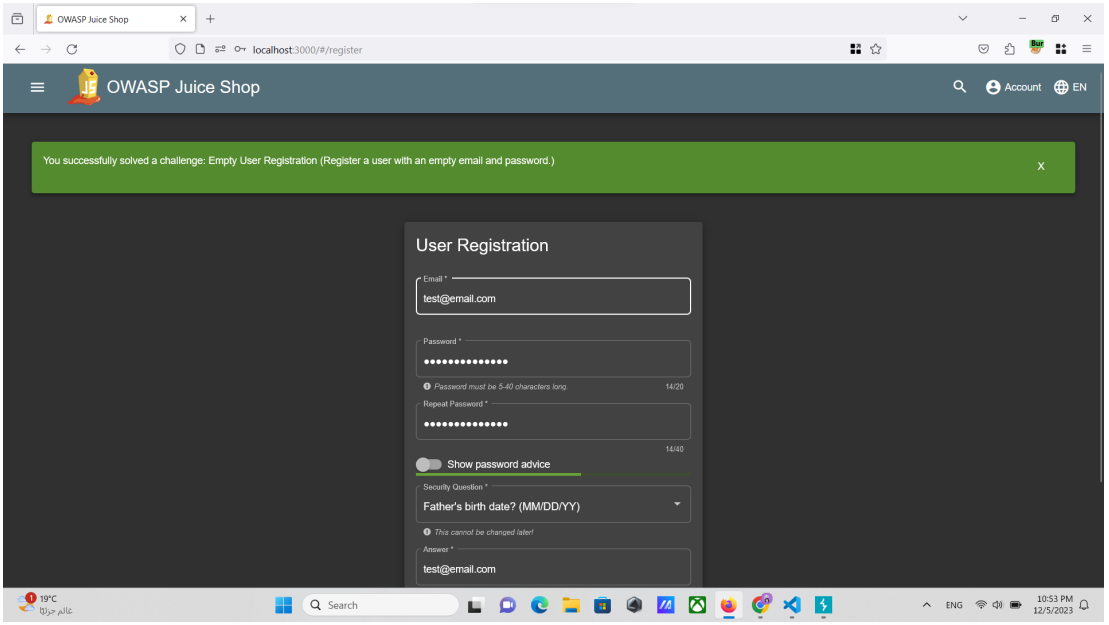
URL: http://localhost:3000/#/register

Description : Register a user with an empty email and password.

Proof of Concept:

Step 1 : Input all data in registration form,then intercept the request and remove email and password.





Remediation Steps:

Validate on the server side that inputs exist.

9.3. Title: Security Policy

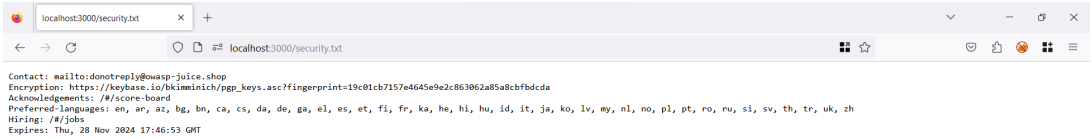
Rating: Low

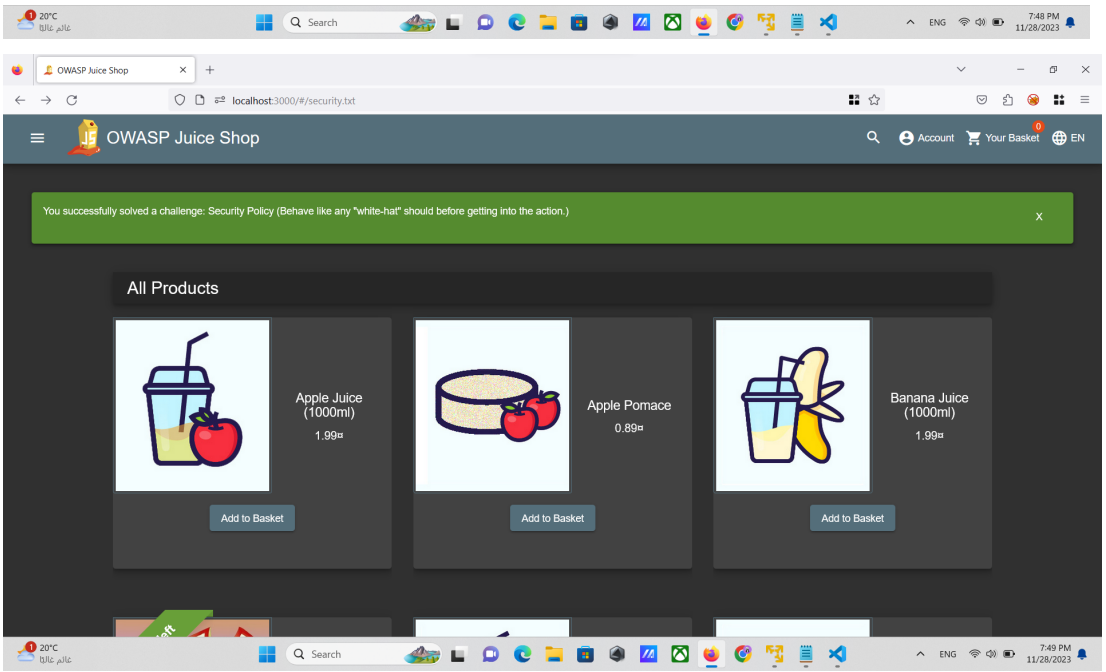
URL: http://localhost:3000/security.txt

Description : Found security.txt file containing some data

Proof of Concept:

Step 1 : Brute Forced directories to find the file.





Remediation Steps:

Do not put important files on the website.

9.4. Title: Bonus Payload

Rating: Low

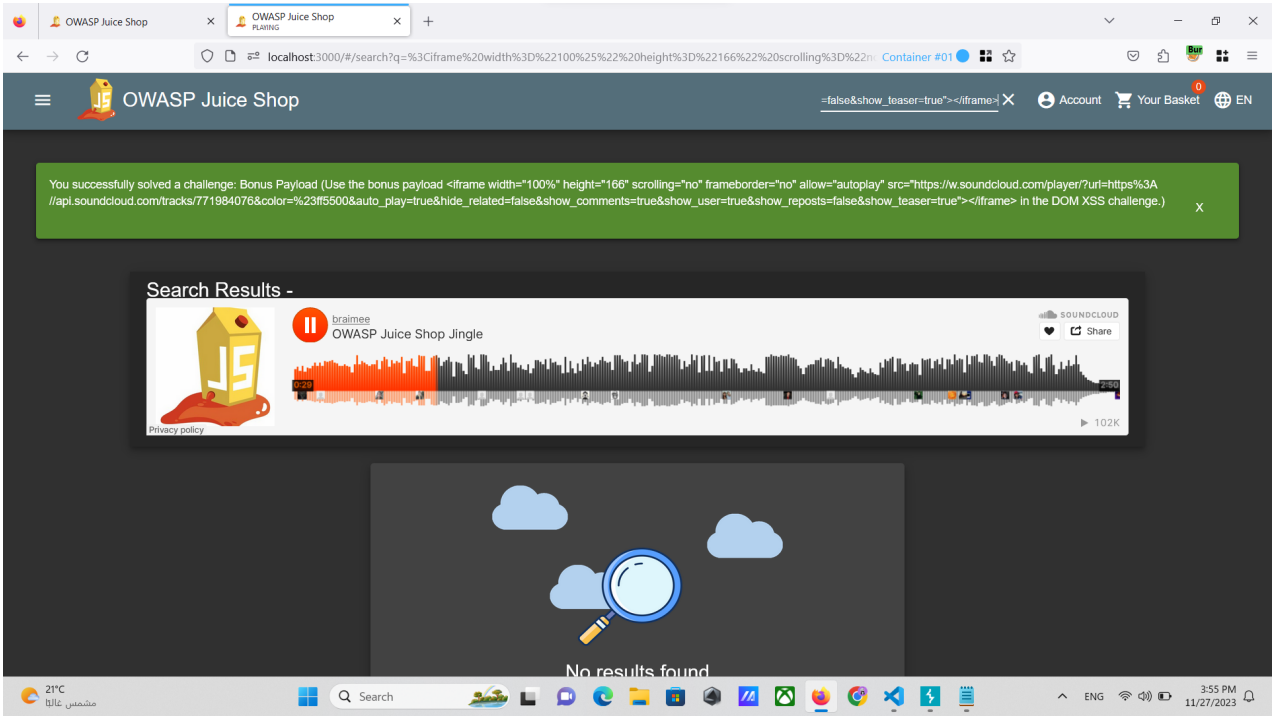
URL: localhost:3000/#/

Description : Bonus payload provided by the website.

Proof of Concept:

Step 1 : After finding the score-board you can find a provided payload.

Step 2 :Payload<iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay" src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/771984076&color=%23ff5500&auto_play=true&hide_related=false&show_comments=true&show_user=true&show_reposts=false&show_teaser=true"></iframe>



Remediation Steps:

Sanitize inputs in the server side form control characters of javascript.

9.5. Title: Privacy Policy

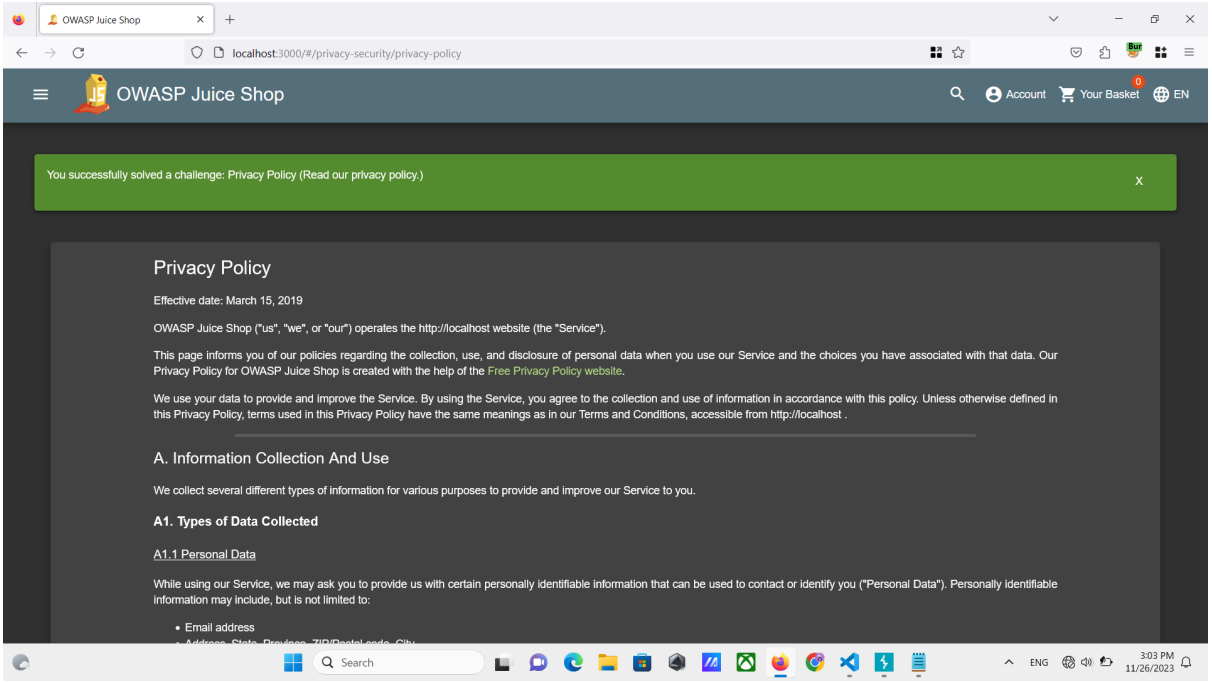
Rating: Low

URL: http://localhost:3000/#/privacy-security/privacy-policy

Description : Found privacy policy page providing valuable info about the website.

Proof of Concept:

Step 1 : Navigate the website.



Remediation Steps:

Remove important data that is not important to the user to read or It is not his right to read.

9.6. Title: Score Board

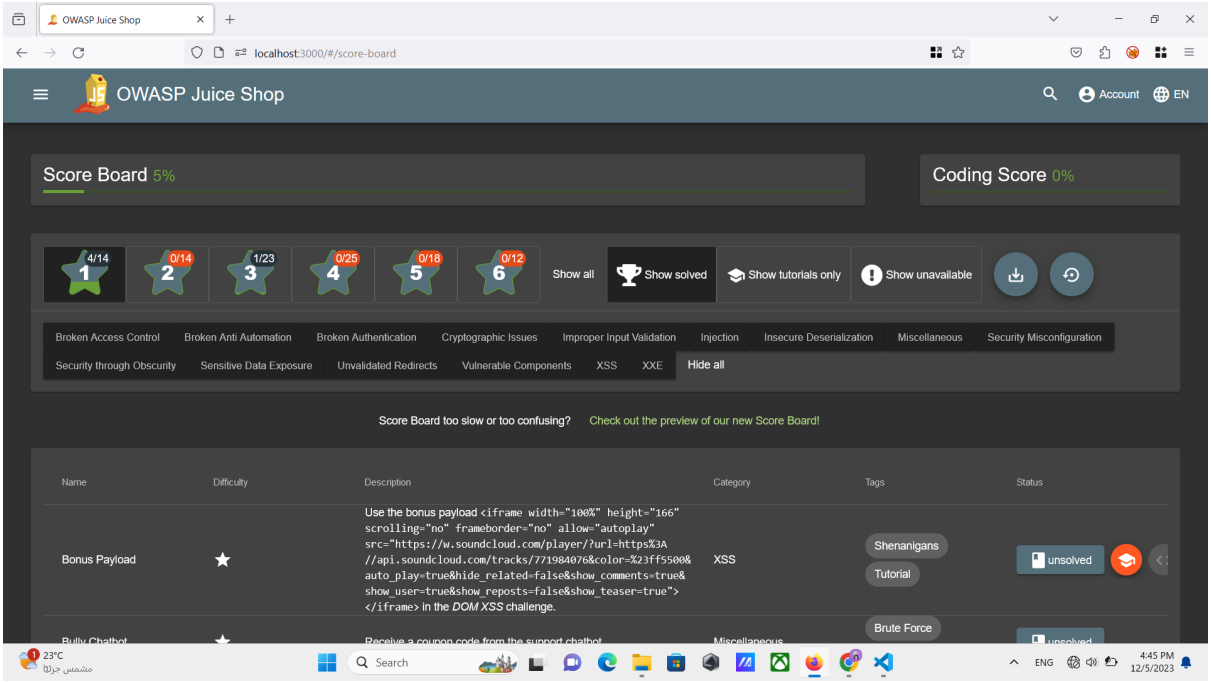
Rating: Low

URL: http://localhost:3000/#/score-board

Description : Found score board page providing valuable info about other bugs.

Proof of Concept:

Step 1 : By searching through the source code the link was found in main.js



Remediation Steps:

It is not a bug .

9.7. Title: Zero Stars

Rating: Low

URL: http://localhost:3000/#/contact

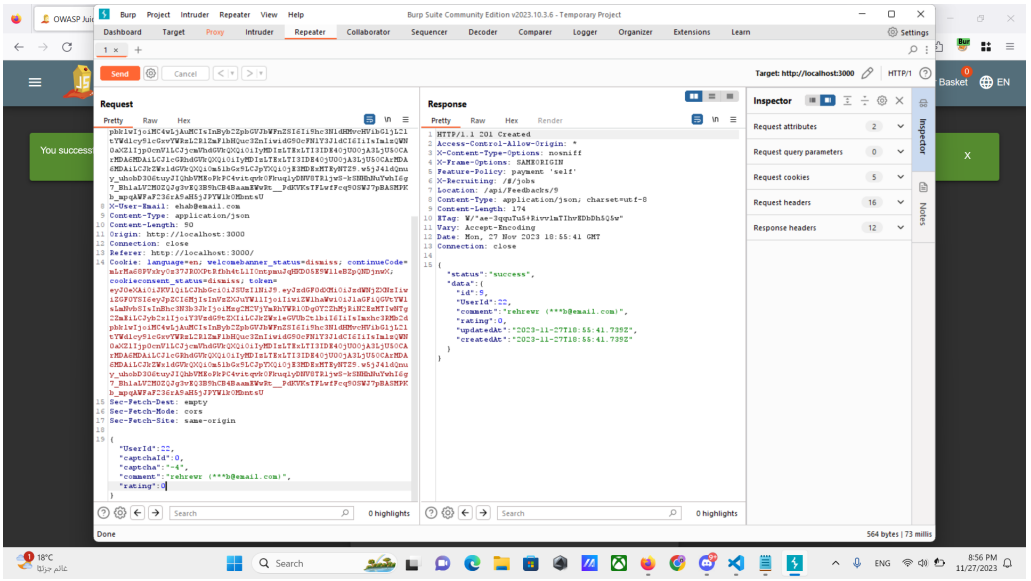
Description : An attacker can manipulate the feedback score to set it to 0 or any other number.

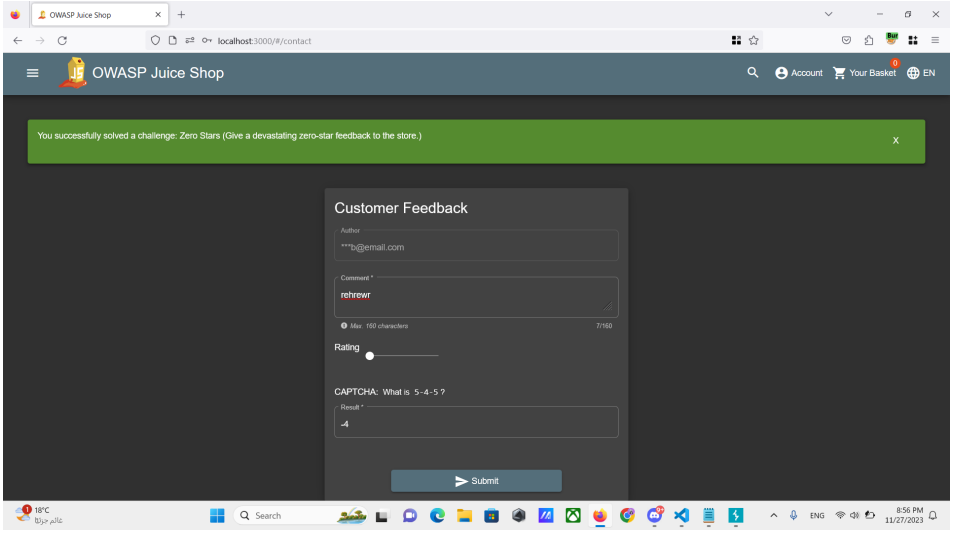
Proof of Concept:

Step 1 : Fill the form and click submit.

Step 2 :Intercept the request.

Step 3 : Change the rating value to 0 and forward the request.





Remediation Steps:

Check the inputs in the server side to make sure that the rating is an integer between 0 and 5.