

# Web Hacking Lab

Name: Eric Wellmaker

## 1 Overview

This lab utilizes the OWASP Juice Shop project to expose students to discovering web vulnerabilities through use of multiple tools and techniques.

### 1.1 Background

The OWASP Juice Shop is a commerce oriented web application which contains many vulnerabilities of varying difficulty to exploit which align with the OWASP Top 10 vulnerabilities. As is often the case there may be multiple ways to exploit a particular vulnerability. Use of a training ground such as Juice Shop allows an individual to practice with multiple tools or processes in identifying and exploiting vulnerabilities.

### 3.3 Complete the one and two star difficulty tutorials (25pts)

Your first task will be to uncover the Juice Shop scoreboard. This scoreboard will not only provide you feedback on your progress but allow you to launch the tutorials required to complete this section.

#### One star challenges

Score Board – Find the carefully hidden ‘Score Board’ page.

DOM XSS – Perform a DOM XSS attack with `<iframe src="javascript:alert('xss')">`

Bonus Payload - Use the bonus payload `<iframe width="100%" height="166" scrolling="no" frameborder="no" allow="autoplay" src="https://w.soundcloud.com/player/?url=https%3A//api.soundcloud.com/tracks/771984076&color=%23ff5500&auto_play=true&hide_related=false&show_comments=true&show_user=true&show_reposts=false&show_teaser=true"></iframe>` in the DOM XSS challenge.

Privacy Policy – Read our privacy policy.

#### Two star challenges

Login Admin – Log in with the administrator’s user account

Password Strength – Log in with the administrator's user credentials without previously changing them or applying SQL Injection.

View Basket – View another user’s shopping basket

Score Board 7%

Coding Score 0%

Progress indicators: 1 (4/12), 2 (3/12), 3 (0/23), 4 (0/25), 5 (0/18), 6 (0/11)

Buttons: Show all, Show solved, Show tutorials only, Show unavailable

Categories: Broken Access Control, Broken Anti Automation, Broken Authentication, Cryptographic Issues, Improper Input Validation, Injection, Insecure Deserialization, Miscellaneous, Security Misconfiguration, Security through Obscurity, Sensitive Data Exposure, Unvalidated Redirects, Vulnerable Components, XSS, XXE, Hide all

Name	Difficulty	Description	Category	Tags	Status
Admin Registration	★★★	Register as a user with administrator privileges.	Improper Input Validation		unsolved
Admin Section	★★	Access the administration section of the store.	Broken Access Control	Good for Demos	unsolved
Bjoern's Favorite Pet	★★★	Reset the password of Bjoern's OWASP account via the <code>Forgot Password</code> mechanism with the original answer to his security question.	Broken Authentication	OSINT	unsolved
Bonus Payload	★	Use the bonus payload <code>&lt;iframe width=100% height=166 scrolling=no frameborder=no allow=autoplay src=https://w.soundcloud.com/player?url=https%3A//api.soundcloud.com/tracks/771984076&amp;color=%23ff55006</code>	XSS	Shenanigans	solved
Bonus Payload	★	Use the bonus payload <code>&lt;iframe vidth=100% height=166 scrolling=no frameborder=no allow=autoplay src=https://w.soundcloud.com/player?url=https%3A//api.soundcloud.com/tracks/771984076&amp;color=%23ff55006 auto_play=true&amp;hide_related=false&amp;show_comments=true&amp;show_user=true&amp;show_reposts=false&amp;show_teaser=true&gt;&lt;/iframe&gt;</code> in the <code>DOM XSS</code> challenge.	XSS	Shenanigans, Tutorial	solved
Bully Chatbot	★	Receive a coupon code from the support chatbot.	Miscellaneous	Brute Force, Shenanigans	unsolved
CAPTCHA Bypass	★★★	Submit 10 or more customer feedbacks within 20 seconds.	Broken Anti Automation	Brute Force	unsolved
CSRF	★★★	Change the name of a user by performing Cross-Site Request Forgery from another origin.	Broken Access Control		unsolved
Confidential Document	★	Access a confidential document.	Sensitive Data Exposure	Good for Demos	unsolved
DOM XSS	★	Perform a <code>DOM XSS</code> attack with <code>&lt;iframe src='javascript:alert('xss')'&gt;</code> .	XSS	Good for Demos, Tutorial	solved

Login Admin	★★	Log in with the administrator's user account.	Injection	Good for Demos Tutorial	<input checked="" type="checkbox"/> solved
Login Amy	★★★★	Log in with Amy's original user credentials. (This could take 93.83 billion trillion billion centuries to brute force, but luckily she did not read the "One Important Sensitive Data Exposure Final Note".)		OSINT	<input type="checkbox"/> unsolved
Login Bender	★★★★	Log in with Benders user account.	Injection	Tutorial	<input type="checkbox"/> unsolved
Login Jim	★★★★	Log in with Jim's user account.	Injection	Tutorial	<input type="checkbox"/> unsolved
Login MC SafeSearch	★★	Log in with MC SafeSearch's original user credentials without applying SQL Injection or any other bypass.	Sensitive Data Exposure	OSINT Shenanigans	<input type="checkbox"/> unsolved
Manipulate Basket	★★★★	Put an additional product into another user's shopping basket.	Broken Access Control		<input type="checkbox"/> unsolved
Mass Dispel	★	Close multiple "Challenge solved"-notifications in one go.	Miscellaneous		<input type="checkbox"/> unsolved
Meta Geo Stalking	★★	Determine the answer to John's security question by looking at an upload of him to the Photo Wall and use it to reset his password via the Forgot Password mechanism.	Sensitive Data Exposure	OSINT	<input type="checkbox"/> unsolved
Missing Encoding	★	Retrieve the photo of Bjorn's cat in "melee combat-mode".	Improper Input Validation	Shenanigans	<input type="checkbox"/> unsolved
Outdated Allowlist	★	Let us redirect you to one of our crypto currency addresses which are not promoted any longer.	Unvalidated Redirects	Code Analysis	<input type="checkbox"/> unsolved
Password Strength	★★	Log in with the administrator's user credentials without previously changing them or applying SQL Injection.	Broken Authentication	Brute Force Tutorial	<input checked="" type="checkbox"/> solved
Payback Time	★★★★	Place an order that makes you rich.	Improper Input Validation		<input type="checkbox"/> unsolved
Privacy Policy	★	Read our privacy policy.	Miscellaneous	Good Practice Good for Demos	<input checked="" type="checkbox"/> solved
Privacy Policy	★	Read our privacy policy.	Miscellaneous	Good for Demos Tutorial	<input checked="" type="checkbox"/> solved
Privacy Policy Inspection	★★★★	Prove that you actually read our privacy policy.	Security through Obscurity	Good for Demos Shenanigans	<input type="checkbox"/> unsolved
Product Tampering	★★★★	Change the href of the link within the OWASP SSL Advanced Forensic Tool (O-SaRt) product description into <a href="https://owasp.stack.com">https://owasp.stack.com</a> .	Broken Access Control		<input type="checkbox"/> unsolved
Repetitive Registration	★	Follow the DRY principle while registering a user.	Improper Input Validation		<input type="checkbox"/> unsolved
Reset Jim's Password	★★★★	Reset Jim's password via the Forgot Password mechanism with the original answer to his security question.	Broken Authentication	OSINT	<input type="checkbox"/> unsolved
Score Board	★	Find the carefully hidden "Score Board" page.	Miscellaneous	Code Analysis Tutorial	<input checked="" type="checkbox"/> solved
Security Policy	★★	Behave like any "white-hat" should before getting into the action.	Miscellaneous	Good Practice	<input type="checkbox"/> unsolved
Upload Size	★★★★	Upload a file larger than 100 kB.	Improper Input Validation		<input type="checkbox"/> unsolved
Upload Type	★★★★	Upload a file that has no .pdf or .zip extension.	Improper Input Validation		<input type="checkbox"/> unsolved
View Basket	★★	View another user's shopping basket.	Broken Access Control	Good for Demos Tutorial	<input checked="" type="checkbox"/> solved
Visual Geo Stalking	★★	Determine the answer to Emma's security question by looking at an upload of her to the Photo Wall and use it to reset her password via the Forgot Password mechanism.	Sensitive Data Exposure	OSINT	<input type="checkbox"/> unsolved
Weird Crypto	★★	Inform the shop about an algorithm or library it should definitely not use the way it does.	Cryptographic Issues		<input type="checkbox"/> unsolved

### Task 3.4 Complete 'Zero Stars' challenge (one star difficulty) (20 pts)

The Zero Stars challenge is for you to find a way to give a devastating zero-star feedback to the store.

OWASP Juice Shop

You successfully solved a challenge: Zero Stars (Give a devastating zero-star feedback to the store.)

Score Board 7%      Coding Score 0%

1 4/12   2 3/12   3 0/22   4 0/25   5 0/18   6 0/11

Show all   Show solved

Show tutorials only   Show unavailable

Broken Access Control   Broken Anti Automation   Broken Authentication   Cryptographic Issues   Improper Input Validation   Injection

Insecure Deserialization   Miscellaneous   Security Misconfiguration   Security through Obscurity   Sensitive Data Exposure

Unvalidated Redirects   Vulnerable Components   XSS   XXE   Hide all

```

Burp Project In intruder Repeater Window Help
Dashboard Target Proxy In intruder Repeater Sequencer Decoder Comparer Logger Extender
Intercept HTTP history WebSockets history Options
Request to http://localhost:3000 [127.0.0.1]
Forward Drop Intercept Stop Action Open Browser
Pretty Raw Hex
1 POST /api/feedbacks/ HTTP/1.1
2 Host: localhost:3000
3 Content-Length: 78
4 sec-ch-ua: "Not:A-Brand";v="99", "Chromium";v="106"
5 Accept: application/json, text/plain, */*
6 Content-Type: application/json
7 sec-ch-ua-mobile: ?0
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36
9 sec-ch-ua-platform: "Linux"
10 Origin: http://localhost:3000
11 Sec-Fetch-Site: same-origin
12 Sec-Fetch-Mode: cors
13 Sec-Fetch-Dest: empty
14 Referrer: http://localhost:3000/
15 Accept-Encoding: gzip, deflate
16 Accept-Language: en-US,en;q=0.9
17 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss
18 Connection: close
19
20 {
  "captchaId":10,
  "captcha":"17",
  "comment":"zero rating (anonymous)",
  "rating":3
}

```

OWASP Juice Shop

Customer Feedback

Author: anonymous

Comment: zero rating

Rating: 3

CAPTCHA: What is 3\*6-1?

Result: 17

Submit

Waiting for localhost...

```

20 {
  "captchaId":10,
  "captcha":"17",
  "comment":"zero rating (anonymous)",
  "rating":0
}

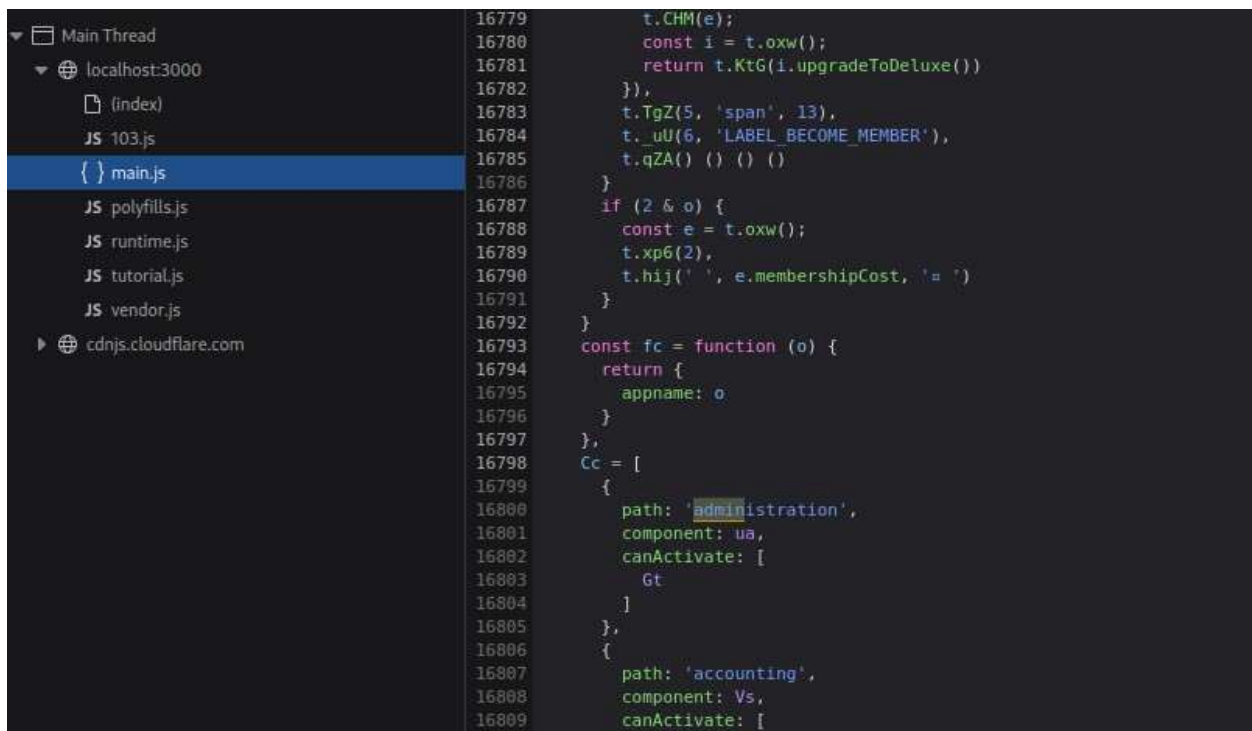
```

To accomplish the zero-feedback rating, I had to utilize another exploit program called Burp Suite. Burp is known for man-in-the middle attacks, so to accomplish giving a “zero rating” I had to open a window inside Burp Suite. From there, I had to intercept the traffic. In the adjacent window for giving customer feedback, I originally gave feedback of 3, then under the Burp intercept forwarding command, I could see the “rating”. I altered the rating to a zero, then hit submit, resulting in a zero rating for the exercise.

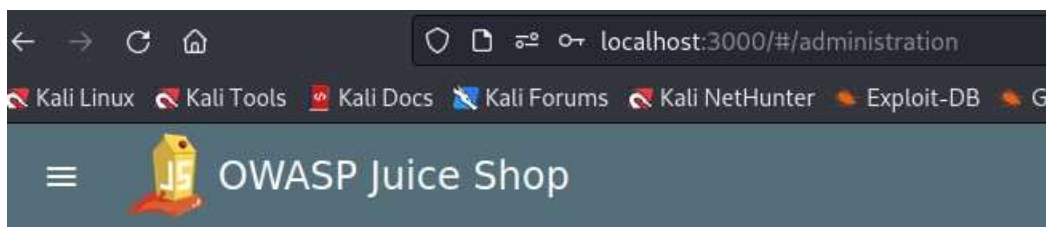
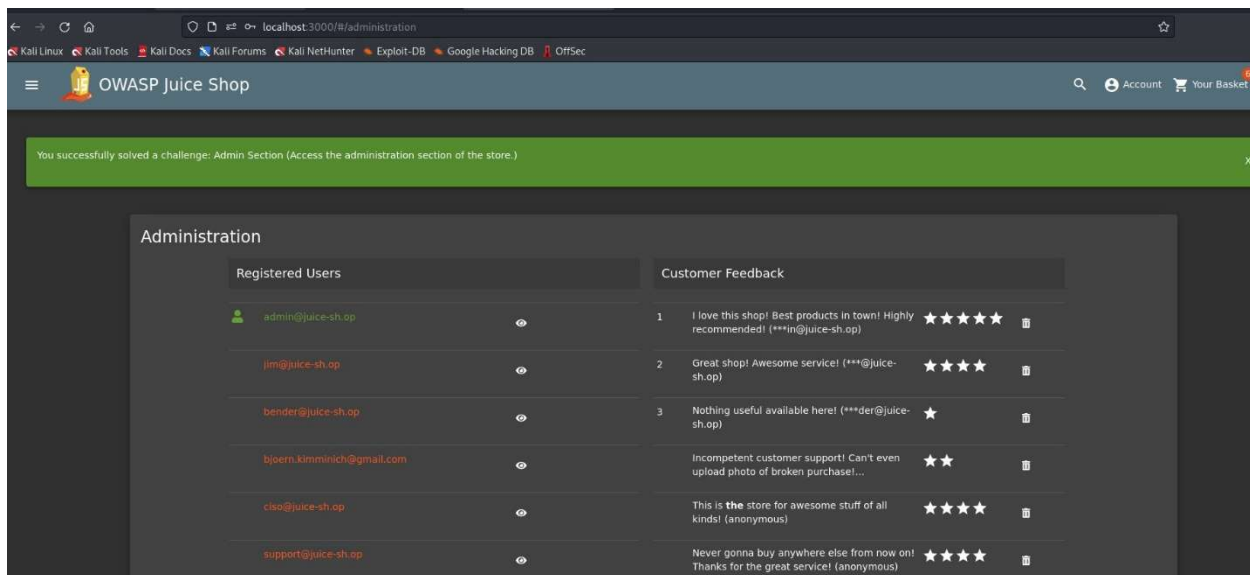
The OWASP Top Ten category for this type of vulnerability would be A1 (Injection Vulnerability) A5 (Security misconfiguration). Mitigation for the Injection Vulnerability would not be allowing a user to accept untrusted data from another application without first properly validating it. And implement access control mechanisms once and re-use them throughout the application, including minimizing Cross-Origin Resource Sharing (CORS) usage (OWASP TOP10., 2021). The other would be properly configured web browser, security software that would guard against vulnerability attacks and utilizing a segmented application architecture which would protect critical infrastructure components from being accessed in the event of a security breach (Simpson, M., & Antill, N., 2017).

### Task 3.5 Complete 'Admin Section' challenge (two star difficulty) (25 pts)

The Admin Section challenge is for you to access the administration section of the store.



```
16779     t.CHM(e);
16780     const i = t.oxw();
16781     return t.KtG(i.upgradeToDeluxe())
16782   }},
16783   t.TgZ(5, 'span', 13),
16784   t._uU(6, 'LABEL_BECOME_MEMBER'),
16785   t.qZA() () () ()
16786 }
16787 if (2 & o) {
16788   const e = t.oxw();
16789   t.xp6(2),
16790   t.hij(' ', e.membershipCost, 'a ')
16791 }
16792 }
16793 const fc = function (o) {
16794   return {
16795     appname: o
16796   }
16797 };
16798 Cc = [
16799   {
16800     path: 'administration',
16801     component: ua,
16802     canActivate: [
16803       Gt
16804     ]
16805   },
16806   {
16807     path: 'accounting',
16808     component: Vs,
16809     canActivate: [
```



For this exploit, I right clicked on the mouse and selected the Inspect (Q) option. From there, I navigated over to the Debugger tab and selected JS main.js. To make it easier to read, you click the {} and it will be in a readable script from 1-til the end of the script. From here I did a “control F” and searched for admin and located the path: ‘administration’. This is what I needed to complete the question. First, I had to login to the website as an admin, otherwise I would get a 403 error. I substituted the scoreboard for administration, and it redirected me to the admin section of the store where I could view all comments for clients store experience. The OWASP Top Ten category for this type of vulnerability would be A4 (Insecure direct object reference). This is the ability for a programmer to guess names when accessing database records (Simpson, M., & Antill, N., 2017). The recommended mitigation for this vulnerability would be to provide a two-way authentication key to the admin that requests to login with administrator privileges. This way, when a request is sent a secondary email or text message is sent to the registered device of the admin to further authenticate the admin. Another level of protection would be to separate tier layers on the system and network layers depending on the necessary level of usage and the type of data being stored (OWASP TOP10., 2021).

### Task 3.6 Complete ‘Deluxe Fraud’ challenge (three star difficulty) (30 pts)

The Deluxe Fraud challenge is for you to obtain a Deluxe Membership without paying for it. You should complete this challenge using your user account (i.e. not an admin or other user) and not user a credit card.

LABEL_NAME	LABEL_DIFFICULTY	LABEL_DESCRIPTION	LABEL_CATEGORY	LABEL_TAGS	LABEL_STATUS
Admin Registration	★★★	Register as a user with administrator privileges.	CATEGORY_IMPROPER_INPUT_VALIDATION		STATUS_UNSOLVED
Bjoern's Favorite Pet	★★★	Reset the password of Bjoern's OWASP account via the Forgot Password mechanism with the original answer to his security question.	CATEGORY_BROKEN_AUTHENTICATION	TAG_OSINT	STATUS_UNSOLVED
CAPTCHA Bypass	★★★	Submit 10 or more customer feedbacks within 20 seconds.	CATEGORY_BROKEN_ANTI_AUTOMATION	TAG_BRUTE_FORCE	STATUS_UNSOLVED
CSRF	★★★	Change the name of a user by performing Cross-Site Request Forgery from another origin.	CATEGORY_BROKEN_ACCESS_CONTROL		STATUS_UNSOLVED
Database Schema	★★★	Exfiltrate the entire DB schema definition via SQL Injection.	CATEGORY_INJECTION		STATUS_UNSOLVED
Deluxe Fraud	★★★	Obtain a Deluxe Membership without paying for it.	CATEGORY_IMPROPER_INPUT_VALIDATION		STATUS_SOLVED

For the final section, I had to log into my account and click the Label Deluxe Member. From there, I clicked again Label\_Become\_Member. From here I had to open Burp Suite in an adjacent window to complete the upgrade action. Going back to My\_Payment\_Options, I right clicked Inspect(Q). Clicking the pick an element option, help narrow down the search

```

orientation="horizontal"></mat-divider>
  <div class="custom-card ng-star-inserted" _ngcontent-wdl-c233="">
    <div _ngcontent-wdl-c233="" fxlayout="row" style="flex-direction: row; box-sizing: border-box; display: flex;">
      <div _ngcontent-wdl-c233="" fxflex="42%" style="flex: 1 1 100%; box-sizing: border-box; max-width: 42%;">
      <div _ngcontent-wdl-c233="" fxflex="38%" style="flex: 1 1 100%; box-sizing: border-box; max-width: 38%;">
      <div _ngcontent-wdl-c233="" fxflex="20%" style="flex: 1 1 100%; box-sizing: border-box; max-width: 20%;">
        <button class="mat-focus-indicator btn mat-raised-button mat-button-base mat-primary mat-button-disabled"
          _ngcontent-wdl-c233="" type="submit" color="primary" mat-raised-button="" style="float: right;" disabled="true">
          event
          <span class="mat-button-wrapper">
          <span class="mat-ripple mat-button-ripple" matripple="">
          <span class="mat-button-focus-overlay">
        </button>
      </div>
    </div>
  </div>
</div>

```

The section highlighted in blue is important because it determines whether or not the customer has paid. It is here that I will alter the code by erasing the “mat button-disabled” & “disabled=“true” in order



to make the balance icon clickable. From there, I went into Burp Suite to execute the alteration of code:

The screenshot shows a mobile application interface titled "My Payment Options". It features several sections: "Add new card" with a sub-option "Add a credit or debit card"; "Pay using wallet" with a "Wallet Balance 49.00" and a "Pay 49.00" button; "Add a coupon" with a sub-option "Add a coupon code to receive discounts"; and "Other payment options". At the bottom are "Back" and "Continue" buttons.

Below the interface is the Burp Suite developer tools window. The "Elements" tab is active, showing the DOM tree. The selected element is a button with the following HTML structure:

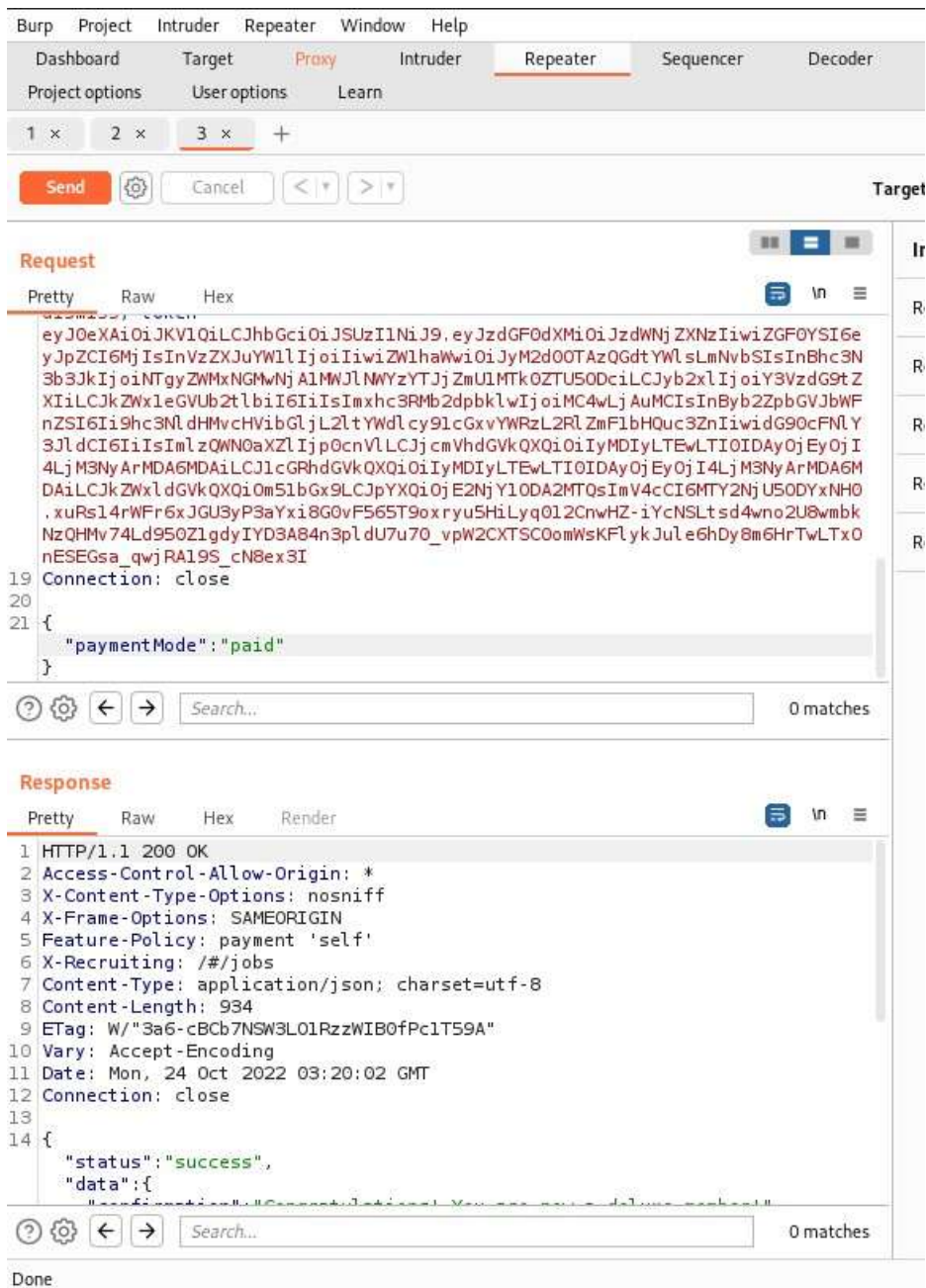
```
<div _ngcontent-ugb-c233 flex="20%" style="flex: 1 1 100%; box-sizing: border-box; max-width: 20%;">
  <button _ngcontent-ugb-c233 type="submit" color="primary" mat-raised-button class="mat-focus-indicator btn mat-raised-button" style="float: right;"> == $0
    <span class="mat-button-wrapper">...</span>
    <span matripple class="mat-ripple mat-button-ripple"></span>
    <span class="mat-button-focus-overlay"></span>
  </button>
</div>
```

The "Styles" panel on the right shows the default styling for the button, including a float: right; property.

Going back to Burp Suite, I utilized the intercept feature and on line 21 you can see {"paymentMode": "wallet"}







From the OWASP Top Ten this type of attack would be A9 (Using components with known vulnerabilities) & A10 (Unvalidated redirects and requests). I'll break down these three and how I was able to gain access according to OWASP. Using components with known vulnerabilities, I opted to utilize Burp Suite because of the Man-in-the-middle attack. It was this attack which allowed me to intercept traffic from the domain and alter the payment to "paid". Along with Burp Suite ties into A10 because the website allowed a unvalidated redirect request without further authentication (Simpson, M., & Antill, N., 2017). To mitigate both vulnerabilities, I would recommend a HTTPS webserver which has multiple layers of security to validate user requests. Ensure log data is encoded correctly to prevent injections or attacks on the logging or monitoring of systems & to ensure high-value transactions have an audit trail with integrity controls to prevent tampering or deletion, such as append-only database tables or similar.

Enforce “deny by default” firewall policies or network access control rules to block all but essential intranet traffic (OWASP TOP10., 2021).

## Reference

OWASP TOP10. (2021). OWASP Top 10:2021. <https://owasp.org/Top10/>

Simpson, M., & Antill, N. (2017). Hands-On Ethical Hacking and Network Defense (3rd ed.).