# CYBR 525 – Web Hacking Lab

Name: \_\_\_\_\_\_\_Alice Mace\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Date: 27 October 2023\_\_\_\_\_\_\_

## 1 Overview

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

## 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.

## 2 Lab Environment

You will complete this exercise in the BU virtual environment, refer to the announcement from your instructor informing you of which environment you should use and for your login credentials. Instructions for accessing and logging into the virtual environment are included in your lesson under the Course Materials link.

This lab runs using the virtualized Kali system you have been given access to.

Information on Juice Shop tasks, tutorials, hints and answers to some challenges are available at <https://owasp.org/www-project-juice-shop/> . Note, this site is not accessible from the BU virtual environment and must be accessed from an internet connected device.

## 3 Tasks

Complete the below tasks. At certain points red text that begins with ‘Response required’ will indicate information which must be provided and submitted with the completed assignment. Your responses can be pasted directly into this document and uploaded to Blackboard or you may create a separate document with your answers to the lab questions.

For tasks 3.4, 3.5, and 3.6, a portion of your answer will be a write-up of your process. This write-up should be similar to what you would provide in a penetration test report if you were describing a vulnerability you discovered and were able to exploit. Your write-up should include a discussion of how you accomplished the exploit, screen shots of key activities, provide the OWASP Top Ten category or categories of what vulnerability or vulnerabilities allowed the exploit, and provide recommended mitigations to the Juice Shop owners.

### 3.1 System Familiarization

Using an internet connected computer (not your virtual machine in the BU virtual environment) go to <https://pwning.owasp-juice.shop/> and read the below sections. These sections will give you background on how Juice Shop works, how to track your progress, and information on some common web hacking tools and processes.

* Architecture overview - <https://pwning.owasp-juice.shop/introduction/architecture.html>
* Challenge tracking - <https://pwning.owasp-juice.shop/part1/challenges.html>
* Hacking exercise rules - <https://pwning.owasp-juice.shop/part1/rules.html>

### 3.2 Walking the “happy path”

The first step to testing any web site/application is getting familiar with the way it is supposed to work. Your reading will tell you this is often called “happy path” testing.

Log into the virtual environment and your kali workstation.

* Open a terminal and enter the command:

docker run --rm –p 3000:3000 bkimminich/juice-shop

* When you see the message that the server is listening on port 3000, launch the Firefox browser and navigate to <http://localhost:3000> . Complete the activities in <https://pwning.owasp-juice.shop/part1/happy-path.html>
* You should leave the terminal window with the docker command open. When you are done with your juice-shop work you can hit ctrl-c in that window to stop the juice-shop processes. Reissue the docker command to continue working, your progress will be saved

### 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.

Follow the instructions and hints here <https://pwning.owasp-juice.shop/part2/score-board.html> to find the scoreboard.

Once the scoreboard is visible complete the below one and two star challenges. You can (and should) use the tutorial mode to walk through the exploit. If you get stuck additional hints and solutions are in the Challenge solutions section https://pwning.owasp-juice.shop/appendix/solutions.html.

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.

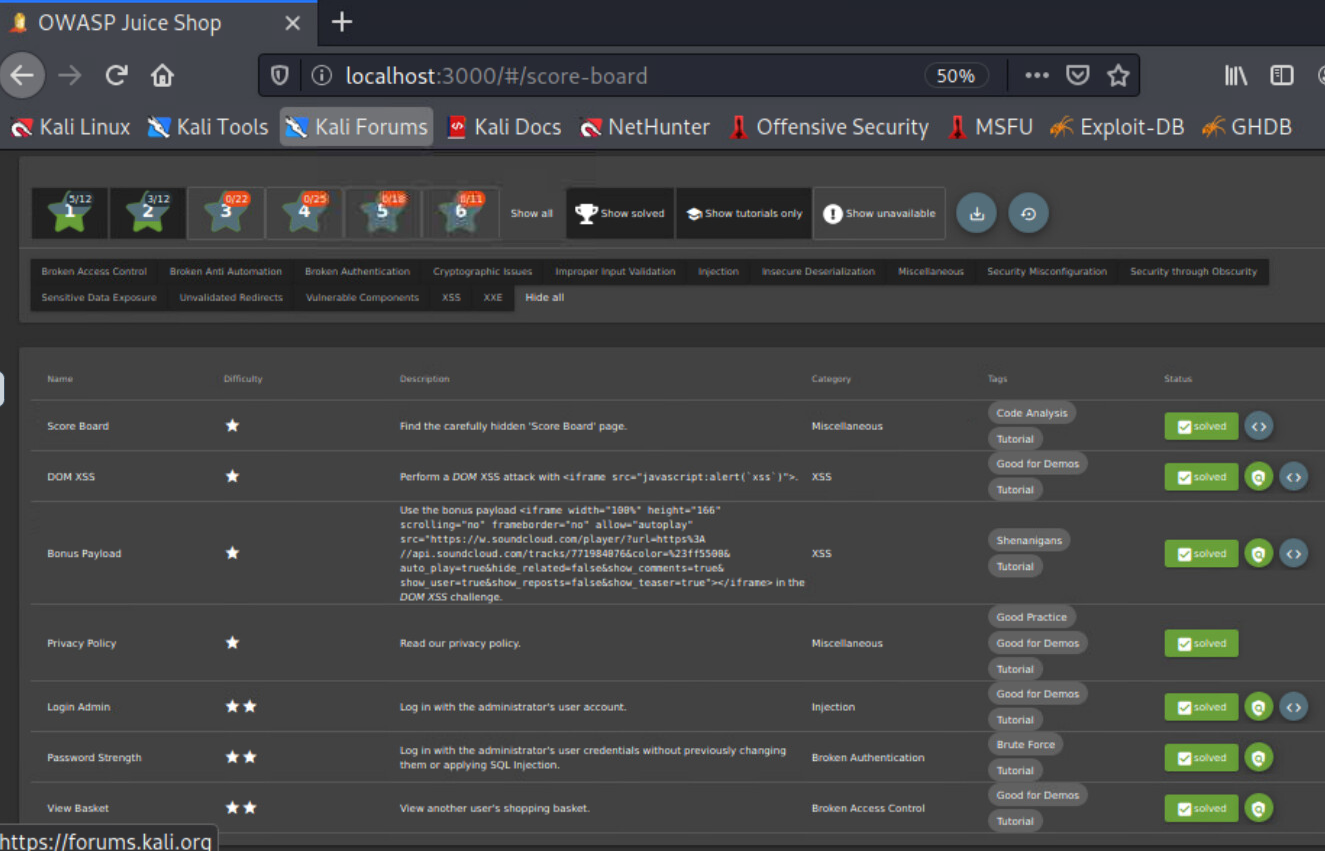
Privacy Policy – Read our privacy policy.

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

Response required: Once you complete the one and two star challenges make a screen shot of your score board and post it below.

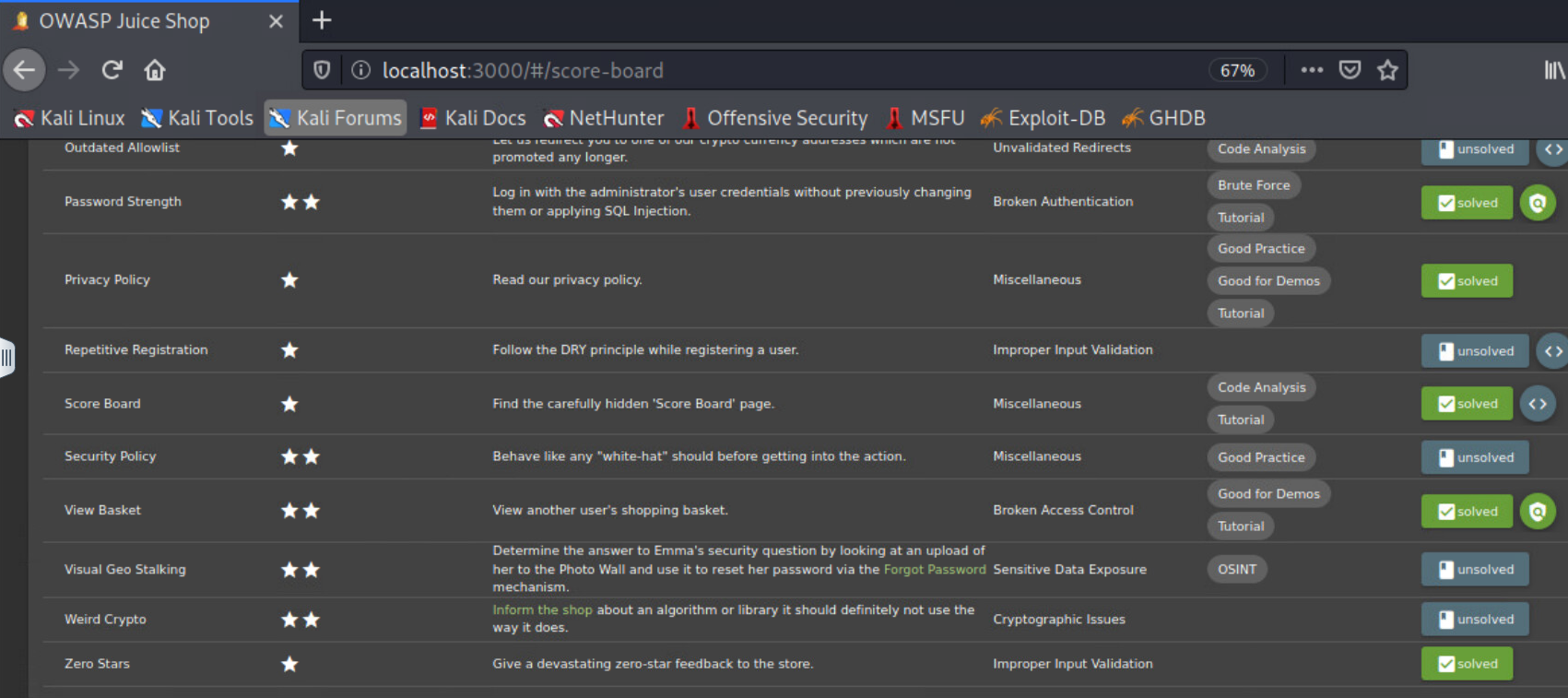
**Reminder – for tasks 3.4, 3.5 and 3.6 your write-up should be structured as you would for a pentest report. You should include a summary of the impact of your exploit, a discussion of how you accomplished the exploit (including screen shots of key activities), provide the OWASP Top Ten category or categories of what vulnerability or vulnerabilities allowed the exploit, and provide recommended mitigations to the Juice Shop owners.**

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

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

Response Required: There are two parts to your response. First, provide a screen shot of this item completed on your scoreboard. Second, provide a write-up of the process you used to complete this challenge. Your write-up should include a discussion of how you accomplished the exploit, screen shots of key activities, use the OWASP Top Ten to identify the category of what vulnerability or vulnerabilities allowed the exploit, and provide recommended mitigations to the Juice Shop owners.

**Part 1:**



**Part 2:**

**Attack Narrative**

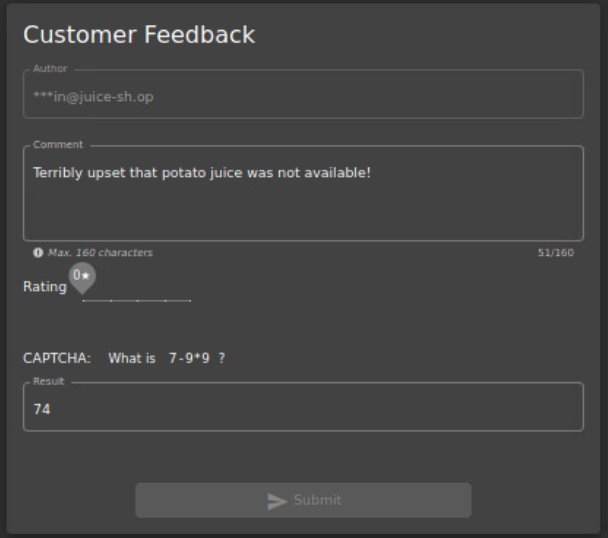
**Objective**

It was requested to complete the *Zero Stars* exploitation in OWASP Juice Shop. In this process, the attacker had to find a way to work with vulnerabilities in the Juice Shop code that allowed the attacker to leave a zero-star feedback review to the store.

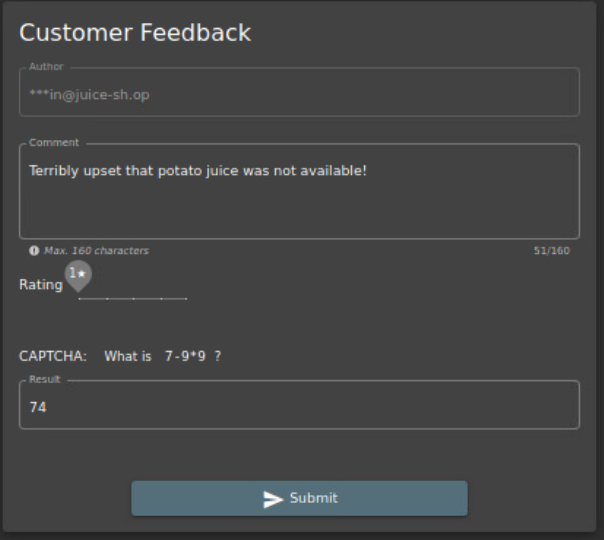
**Attack Path Identification**

After an account was created, the attacker began to become familiar with the customer feedback options. Initial investigation of this process led to the discovery that posts could only be submitted if they met three criteria: contained comment text, had a star rating of at least 1, and completed the CAPTCHA equation. Once all three criteria were met, the feedback Submit button would become enabled and submittal was allowed. Several other unique characteristics were discovered as noted below.

* When filling out the review form, the rating could display as 0 if right-clicked, but it would not allow for submission.



* When left-clicked or dragged, the rating value would no longer go back to 0, but did allow for submission.

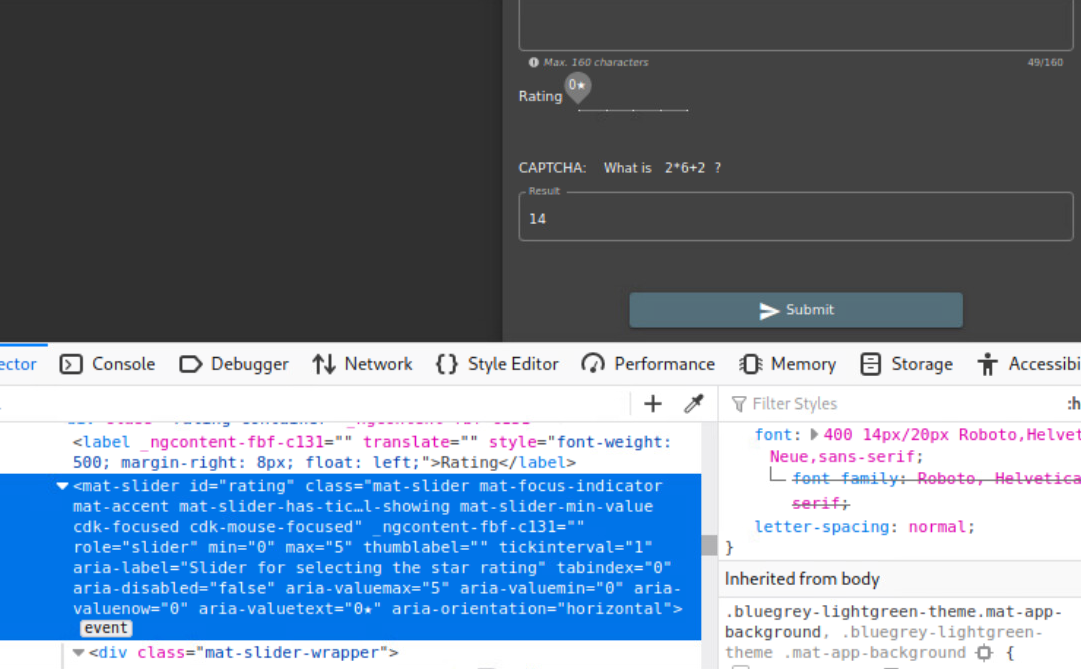


The Inspect Element option was then opened and reviewed code to find the portions responsible for what parameters determined an allowable submission. It was discovered that the code entire code was editable. The attacker began by attempting to only edit the allowable rating value from 1 to 0, but that did not permit submission of the feedback. Both the *mat-slider*, controlling what rating is presented, and the *button id*, controlling the accessibility of the Submit button, portions of the code were also able to be manipulated. These two were key in the ability to select the desired rating and to submit the feedback as detailed in the following section.

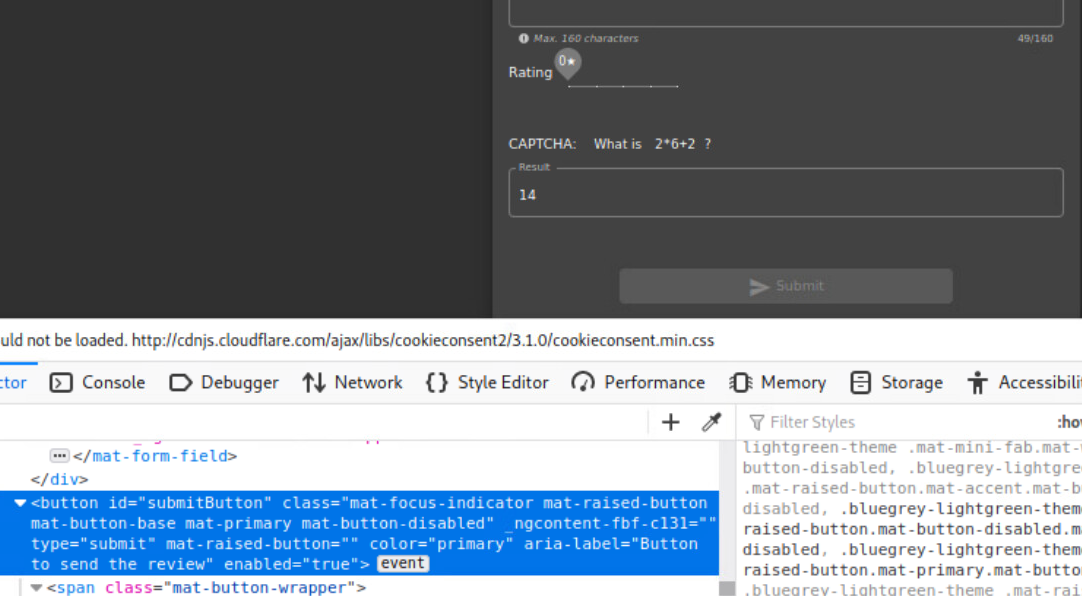
**Exploited Vulnerabilities**

The *mat-slider* and *button id* editable code allowed the attacker to manipulate the values of the targeted feedback process to be able to submit a 0-star rating. The two portions noted below were what was exploited to achieve this objective.

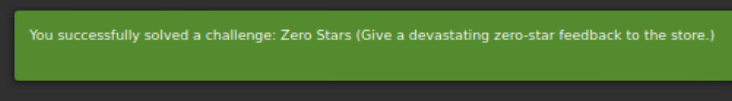
* Within the *mat-slider* code, the *“slider” min=”1”* was edited to allow for minimum values acceptable of the rating to become 0 from 1



* Then the attacker changed the *“Button to send the review”* option from *disabled=”true”*  to *enabled=”true”* so that the option to submit would be allowable



* Back on the feedback form, the attacker then right clicked on the Submit button to activate the code change and it allowed submittal of the zero-star rated feedback



**Conclusion**

Juice Shop has a number of concerns with its cybersecurity configuration with four key risks driving the exploitation allowance from this objective. The specific goal was to be able to submit a zero-star rated feedback to the store, this was achievable due to its vulnerabilities. The following provides additional information on those risks and recommendations to make the Juice Shop more secure.

**Security Risks**

The four following risks from OWASP’s Top 10 Web Application Security Risks from 2021 are applicable to the vulnerabilities that allowed for exploitation of the feedback form: Broken Access Control, Insecure Design, Security Misconfiguration, Vulnerable and Outdated Components. The Broken Access Control risk is number one out of the Top 10, and is primarily met due to the “violation of the principle of least privilege or deny by default, where access should only be granted for particular capabilities, roles, or users, but is available to anyone” (OWASP, n.d.). The Insecure Design risk is rated as number four and, while having a broad concept, is primarily focused on missing or ineffective control designs (OWASP, n.d.-b). The Security Misconfiguration risk is rated number five and is flagged due to similar concerns focused on missing appropriate security hardening and improperly configured permissions, unnecessary features enabled, and the form code was not set to secured values that prevented tampering (OWASP, n.d.-c). The Vulnerable and Outdated Components risk is rated number six from OWASP and is focused on known vulnerabilities with software or other components, the failure to update to fix vulnerabilities, and unsecured component configurations (OWASP, n.d.-d).

**Risk Rating**

The overall risk identified to the Juice Shop as a result of this penetration test is High. The ability for any user to access and edit code would result in a potentially devastating attack to the Juice Shop if conducted by malicious entities or accidentally altered by unknowing users.

**Recommendations**

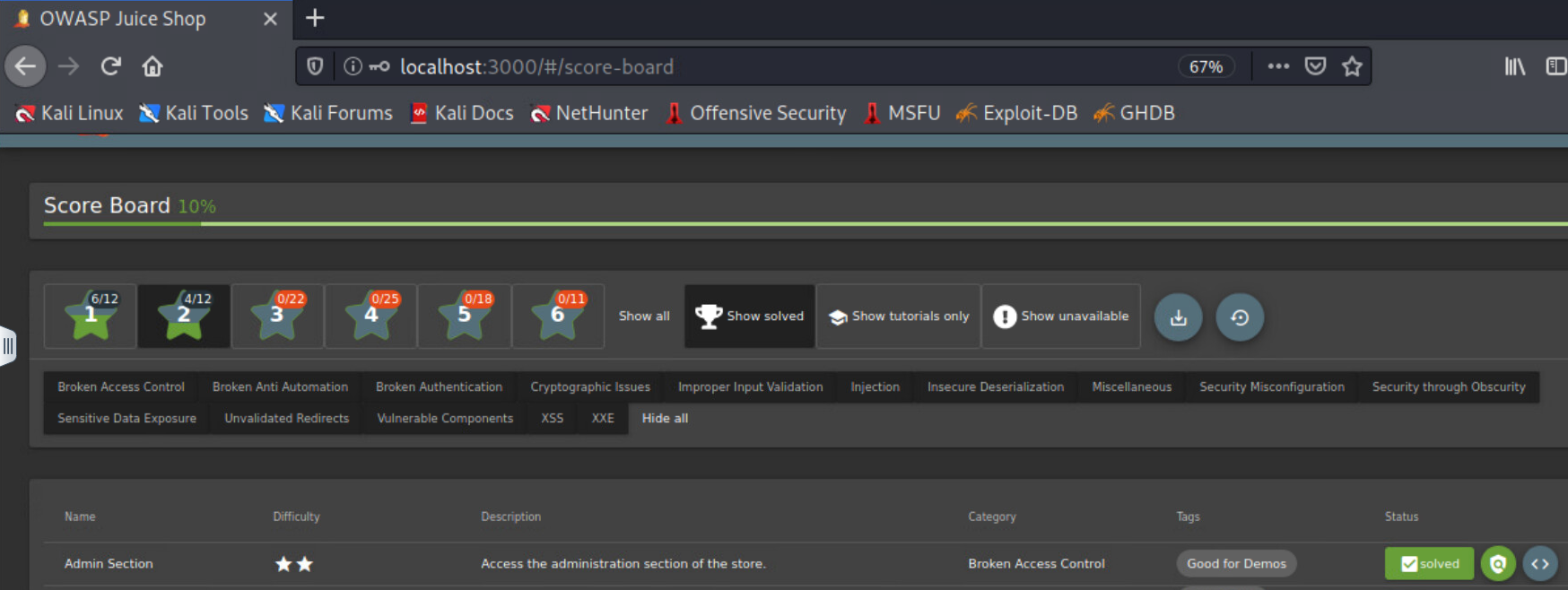
The importance of security controls to an already vulnerable network cannot be understated. Due to the four primary risks discovered that contributed to the ability to use the feedback form in an undesired manner, its recommended that the following security measures be implemented:

1. Instill the control design principle of least privilege and/or deny by default for the ability to access or edit code and other applicable features. This would prevent users without specific credentials from making the changes that were done to exploit this vulnerability. It would also ensure that if a proper user is allowed to view and edit code, they are not able to change all portions without additional permissions.
2. Use secured values to prevent tampering of the portions of code that should never, or rarely, have to be changed for additional access control.
3. Policies and processes should also be implemented for two-person integrity to be required for any major changes or updates to reduce security risks and allow for validation of the change.

### 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.

Response Required: There are two parts to your response. First, provide a screen shot of this item completed on your scoreboard. Second, provide a write-up of the process you used to complete this challenge. Your write-up should include a discussion of how you accomplished the exploit, screen shots of key activities, use the OWASP Top Ten to identify the category of what vulnerability or vulnerabilities allowed the exploit, and provide recommended mitigations to the Juice Shop owners.

**Part 1:**

**Part 2:**

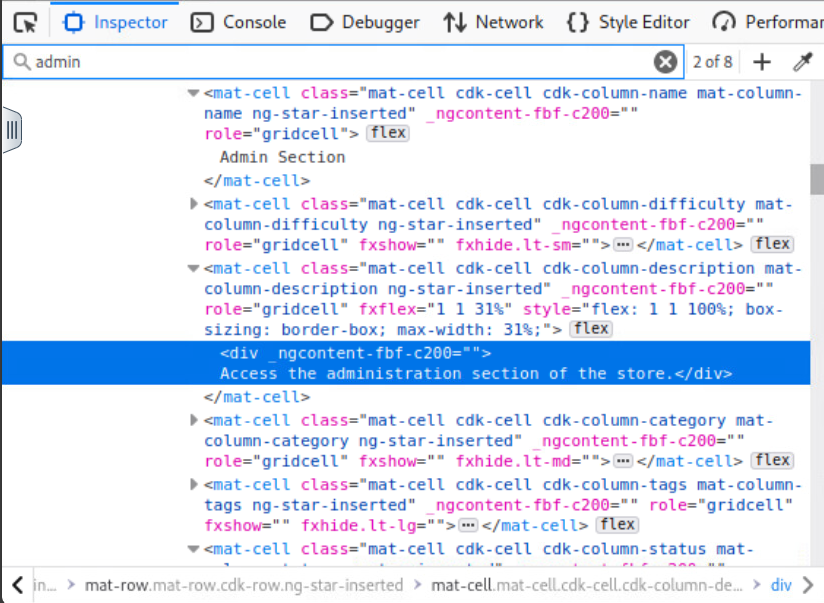
**Attack Narrative**

**Objectives**

It was requested to complete the Admin Section exploitation in OWASP Juice Shop. In this process, the attacker had to find a way through broken access control vulnerabilities to gain access to the administration section of the Juice Shop store.

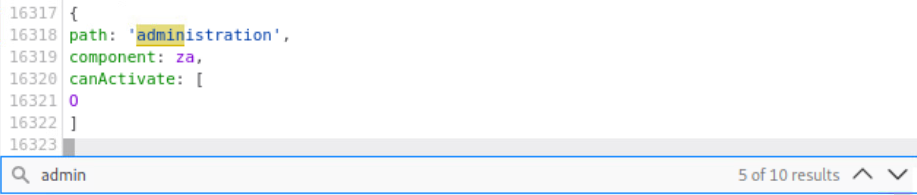
**Attack Path Identification**

The attacker began by inspecting elements of the Juice Shop’s code to try to find where the administration section could be accessed from. When conducting a search for “*admin*”, the *Admin Section* was immediately discovered and a few lines below contained the phrase “*Access the administration section of the store.</div>*” regarding the description of this task, as shown below.

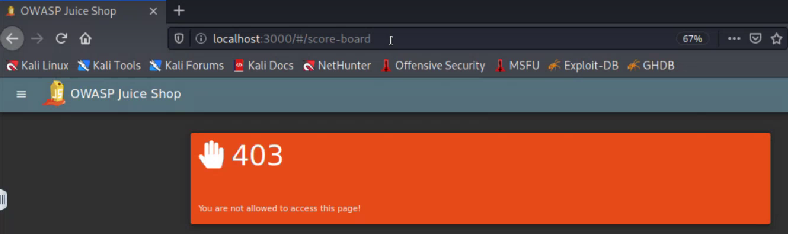


**Exploited Vulnerabilities**

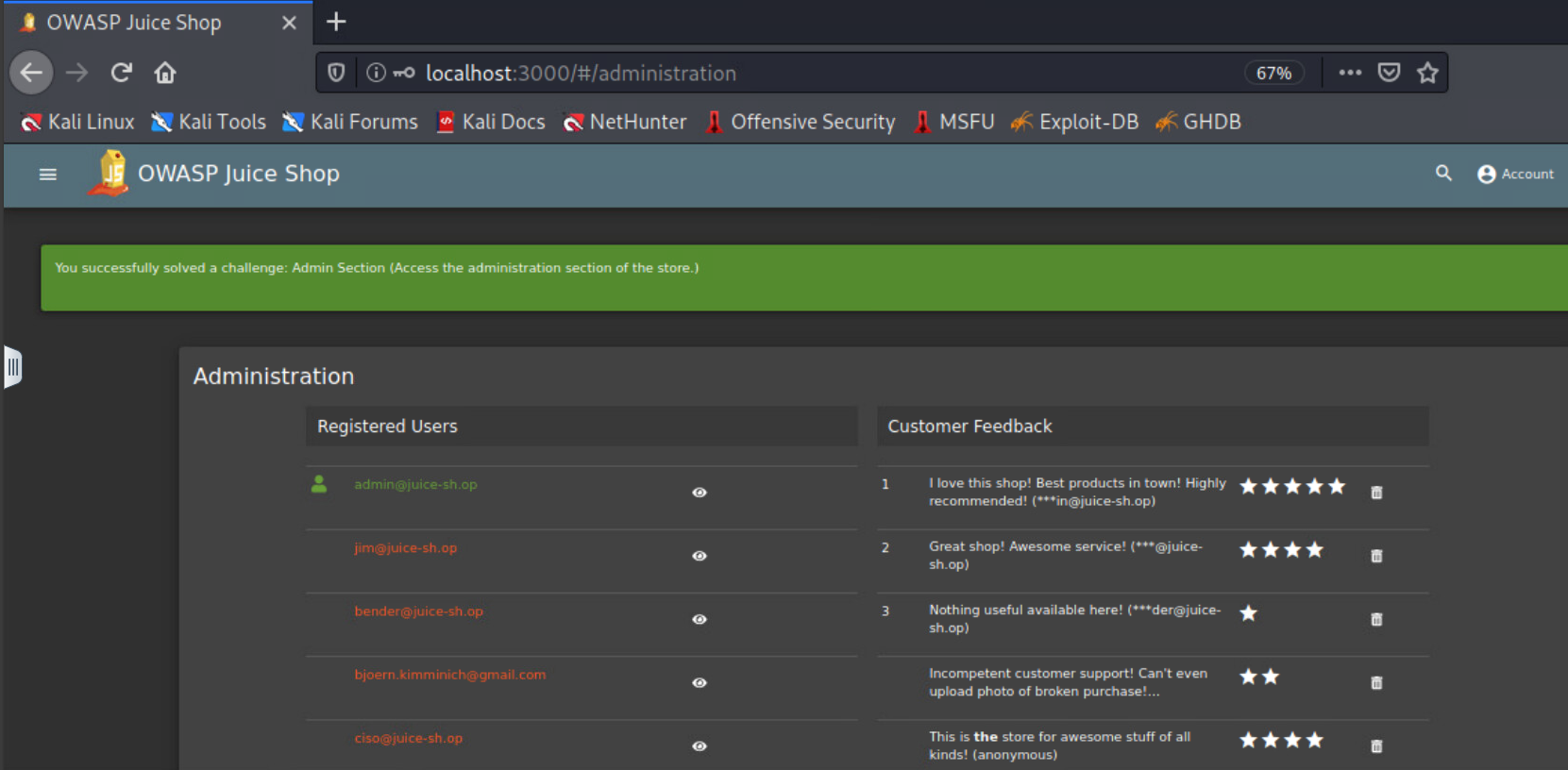
Without finding anything additional of direct value, the attacker went to the Debugger option and conducted another search for “admin”. In a prior exercise, this process was used to find what URL extension path was needed to reach the Score Board. After reviewing the results, the attacker found line 16318 that listed *path: ‘administration’*, shown below.



The attacker then tried entering the *administration* URL extension path and received a 403 error, likely due to being logged in under the attacker’s own credentials.



In a prior exercise, the administrator’s email and password were discovered to be *admin@juice-sh.op* and *admin123*. Since this does not meet the criteria for a secure password, it allowed the attacker to guess the password once the email address had been found in comment and feedback options. Using these credentials from the prior exercise, the attacker logged in as the administrator and then attempted to reach the *administration* path again. This time the process was successful and the attacker was able to access the administration section of the Juice Shop, as depicted on the next page.



**Conclusion**

The Juice Shop’s cybersecurity configuration risks and vulnerabilities allowed for the successful completion of this objective. The desired outcome was to access the administrative section of the website which was achievable primarily due to authentication and password vulnerabilities. The following provides additional information on these risks and recommendations to make the Juice Shop more secure.

**Security Risks**

The four following risks from OWASP’s Top 10 Web Application Security Risks from 2021 are applicable to the vulnerabilities that allowed for exploitation of the administration section of the store: Broken Access Control, Insecure Design, Security Misconfigurations, and Identification and Authentication Failures. While several of these have already been discussed in the prior exploitation, different approaches and considerations apply to this scenario. Broken Access Control also encompasses bypassing access control checks through URL modification (OWASP, n.d.). Insecure Design prior description stands with the risk of missing or ineffective control designs (OWASP, n.d.-b). The Security Misconfiguration focus on missing appropriate security hardening and improperly configured permissions is valid in this exploit, likely as well as default accounts and passwords not being changed (OWASP, n.d.-c). Identification and Authentication Failures is rated seventh in OWASP’s Top 10 and encompasses the brute force password attack; “use of default, weak, or well-known passwords”; and “missing or ineffective multi-factor authentication” (OWASP, n.d.-e).

**Risk Rating**

The overall risk identified to the Juice Shop as a result of this penetration test is High. The ability to easily obtain the administrators login credentials and use them to access content and options only meant for administrations could have a distressing impact to the Juice Shop’s website.

**Recommendations**

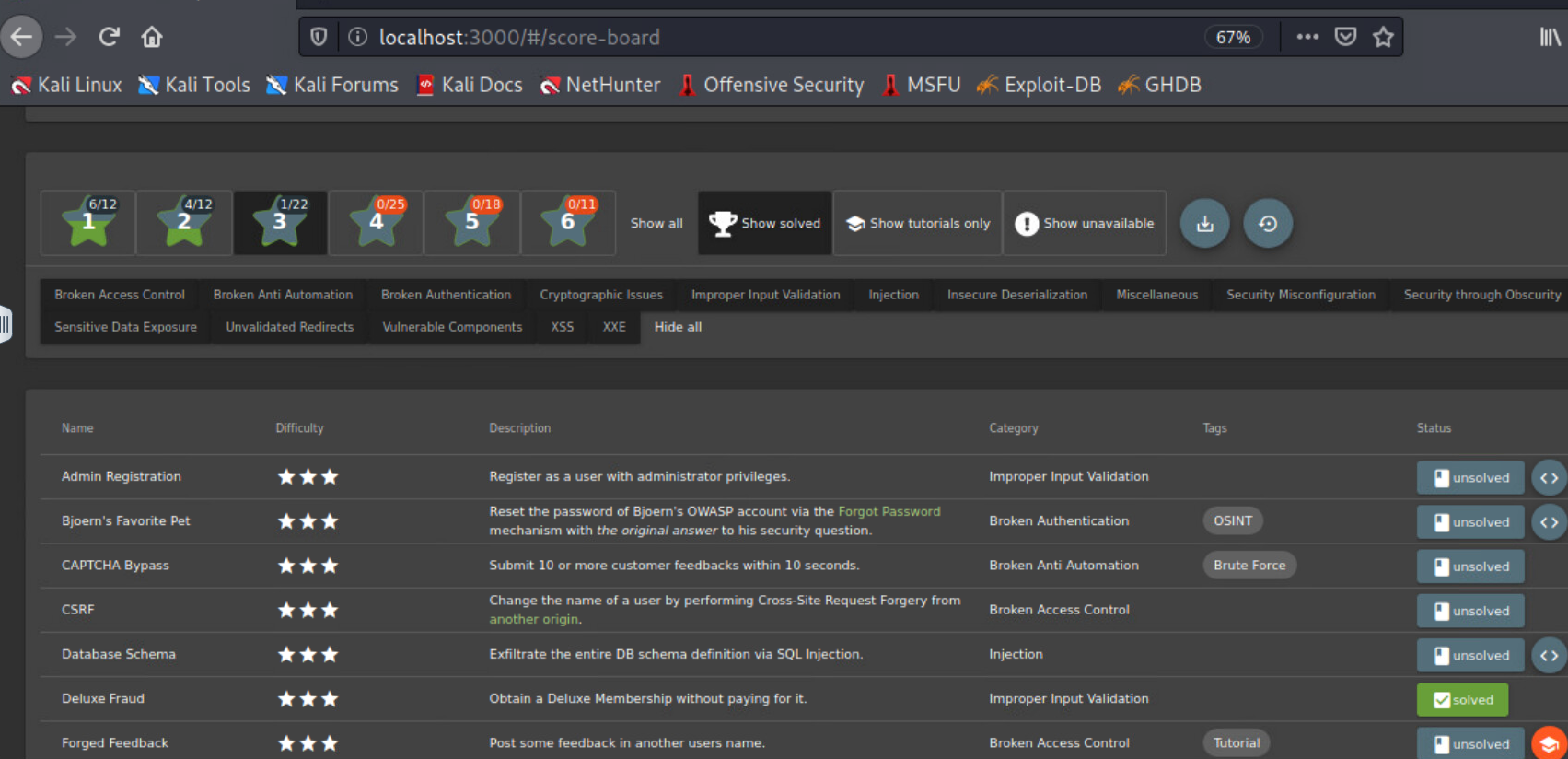
The importance of security controls to an already vulnerable network cannot be understated. Due to the four primary risks discovered that contributed to the ability to obtain unauthorized access to administrative portions of the Juice Shop, its recommended that the following security measures be implemented:

1. Require complex passwords for all users that must be changed on a recurring basis. Incorporating capital and lower-case letters as well as a mix of numbers and special characters can decrease the chance of brute force password attacks.
2. Implement two factor authentication. Utilizing a token or secondary authentication method would likely have prevented the attacker from being able to login with the administrator credentials.
3. Incorporate additional login requirements when accessing administrative portions of the network so that credentials must be reentered and/or a separate set of credentials used before access to these portions is granted.

### 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 using a credit card.

Response Required: There are two parts to your response. First, provide a screen shot of this item completed on your scoreboard. Second, provide a write-up of the process you used to complete this challenge. Your write-up should include a discussion of how you accomplished the exploit, screen shots of key activities, use the OWASP Top Ten to identify the category of what vulnerability or vulnerabilities allowed the exploit, and provide recommended mitigations to the Juice Shop owners.

**Part 1:**

**Part 2:**

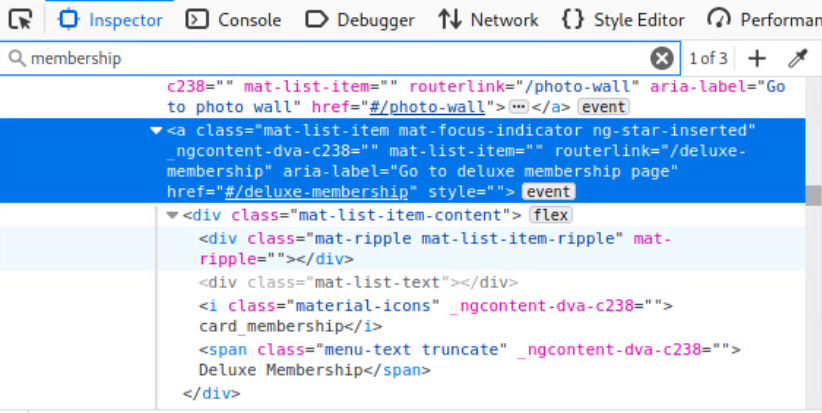
**Attack Narrative**

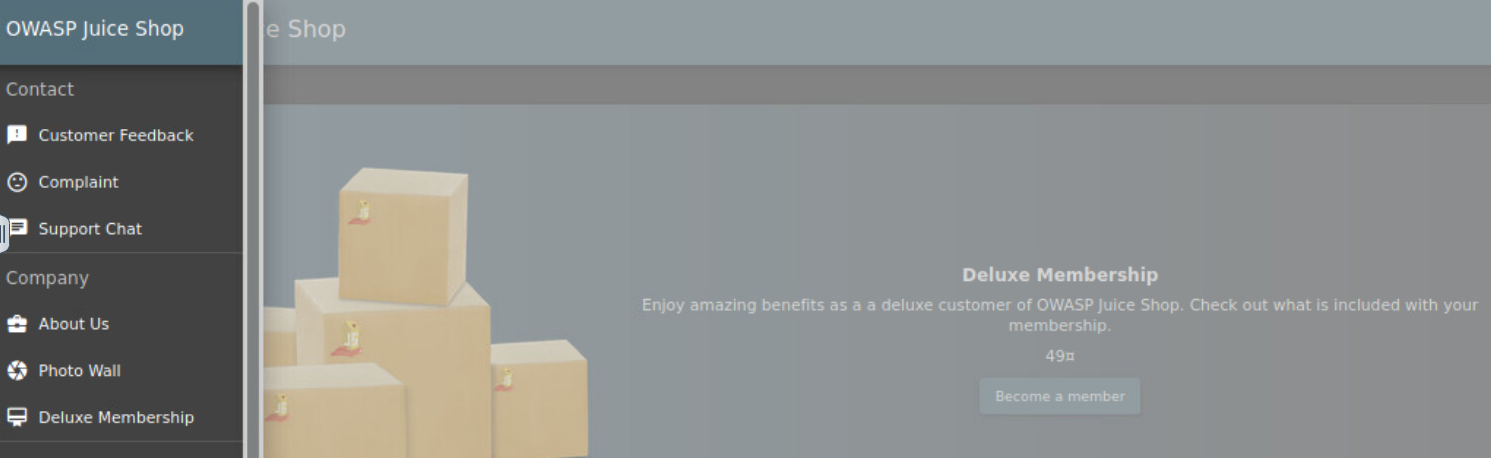
**Objectives**

It was requested to complete the Deluxe Fraud exploitation in OWASP Juice Shop. In this process, the attacker had to find a way to obtain a Deluxe Membership with Juice Shop without paying for it and without using administrative credentials.

**Attack Path Identification**

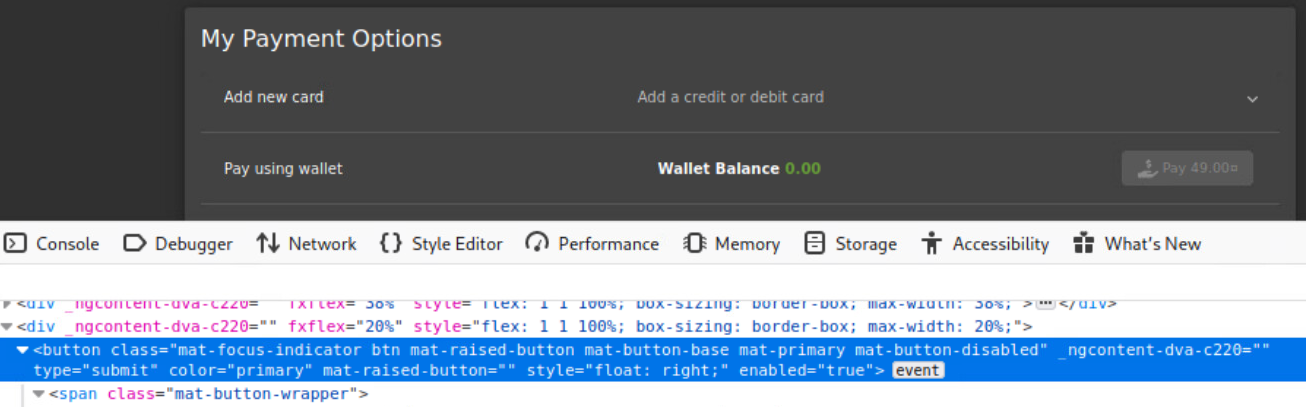
The attacker began by looking through account options and inspecting elements of the Juice Shop’s code to try to find where the Deluxe Membership details may reside. Only two mentions of this membership appeared with the initial inspection and the link to the Deluxe Membership page was discovered under the menu options on the left.



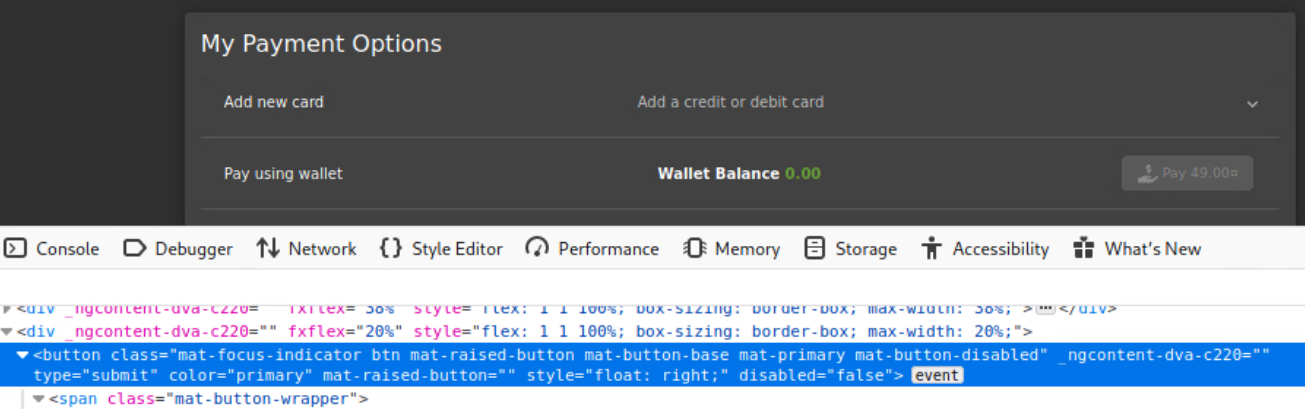


**Exploited Vulnerabilities**

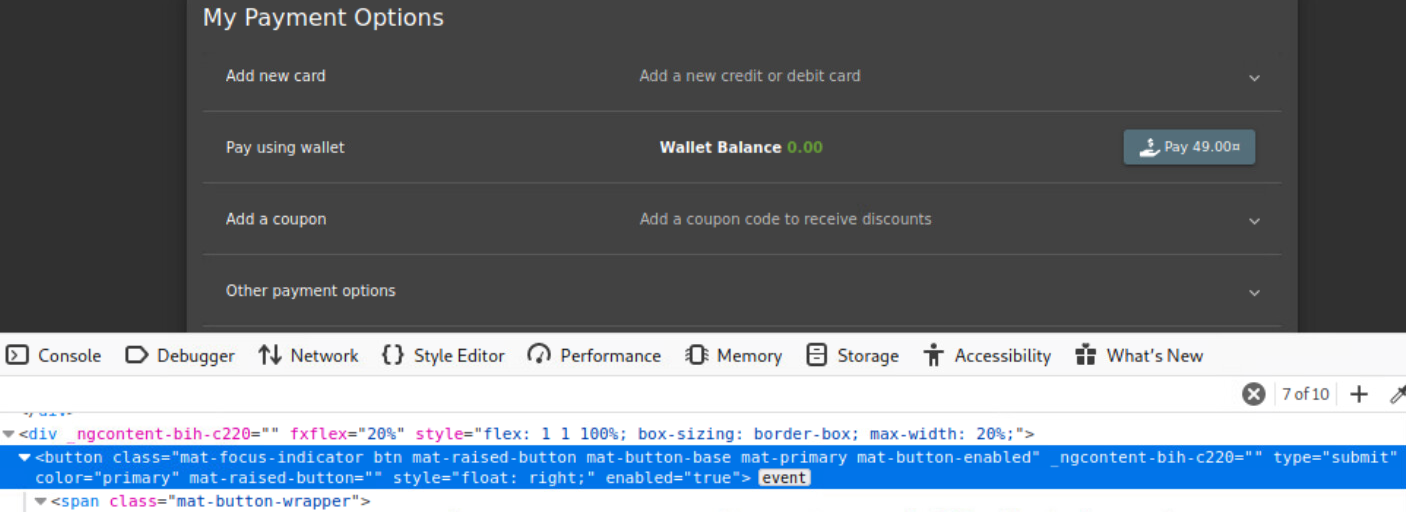
After selecting the *“Become a member”* button, the attacker began to inspect the My Payment Options form and source code. It was discovered that the option to edit the code for the *Pay 49.00* button was available. The attacker then attempted to enable it and changed it from *disabled=”true”* to *enabled=”true”* without any results to the *Pay* button.



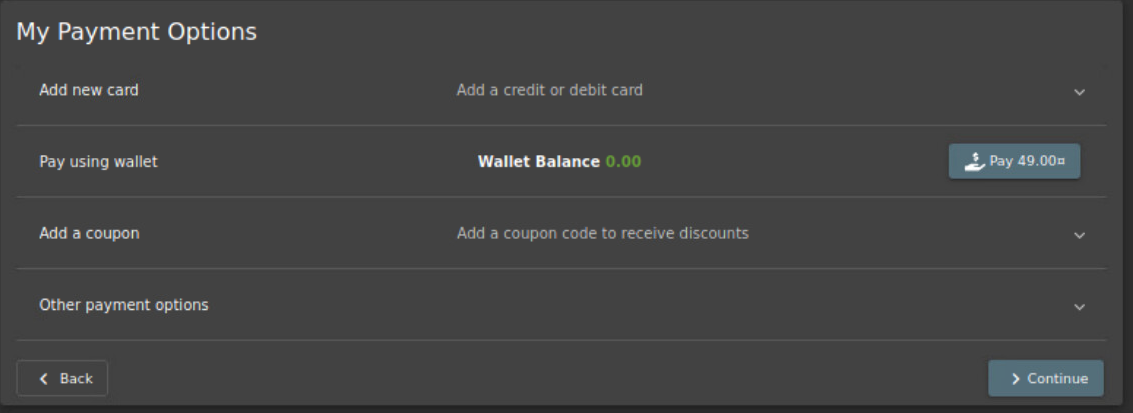
Then the attacker tried changing the same line to *disabled=”false”* without results again.



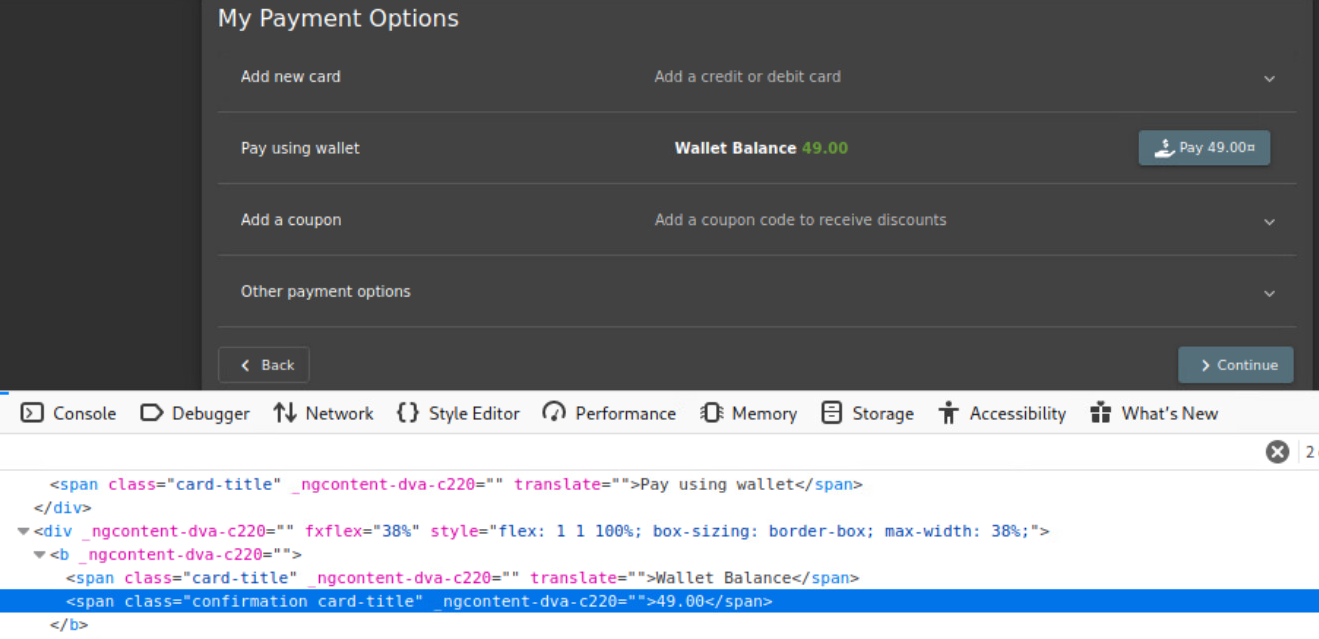
Finally, the attacker evaluated earlier in the same line of code and edited primary *mat-button-disabled* to *primary mat-button enabled* and *enabled=”true”* which achieved the desired result of enabling the *Pay* button.



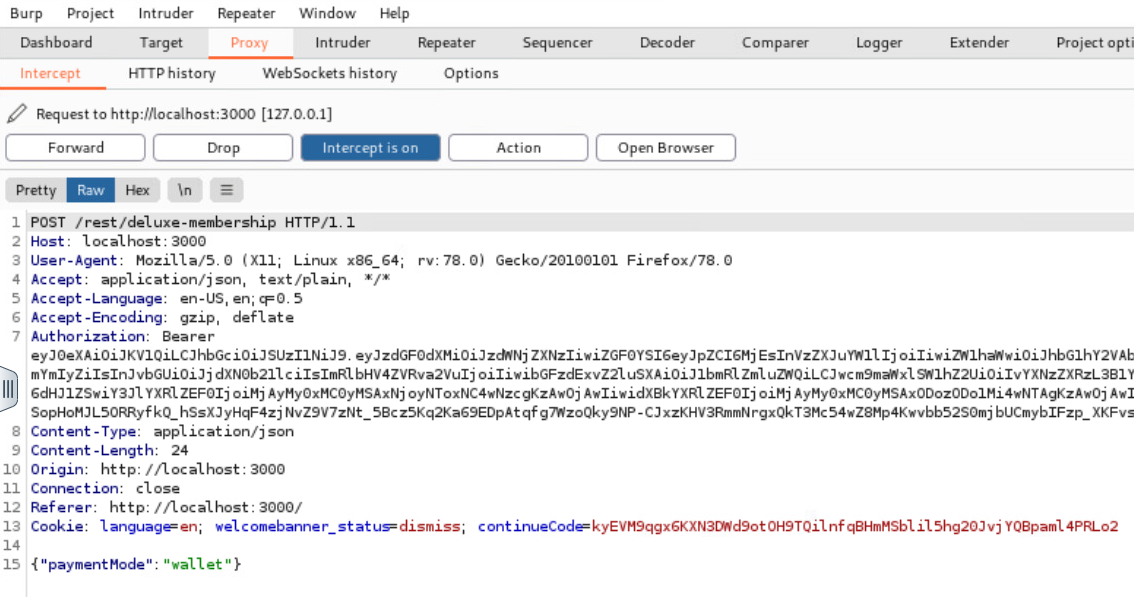
The attacker then selected the *Pay* button which also enabled the *Continue* button.



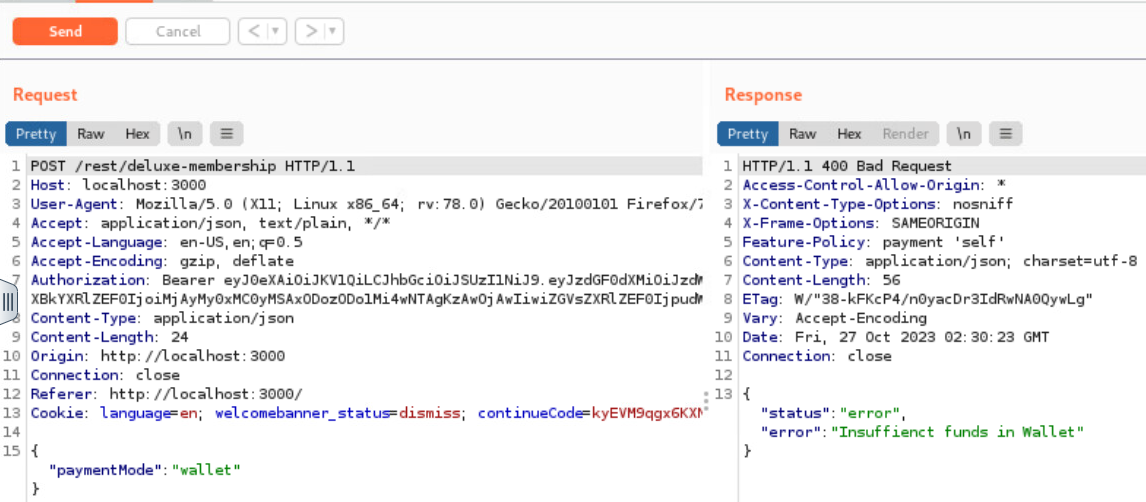
No actions executed when the *Continue* button was selected, so the attacker began to inspect the source code for that button and payment as well. The attacker attempted to alter the balance of the wallet to accept as payment, but even after altering there was no acceptance of payment.



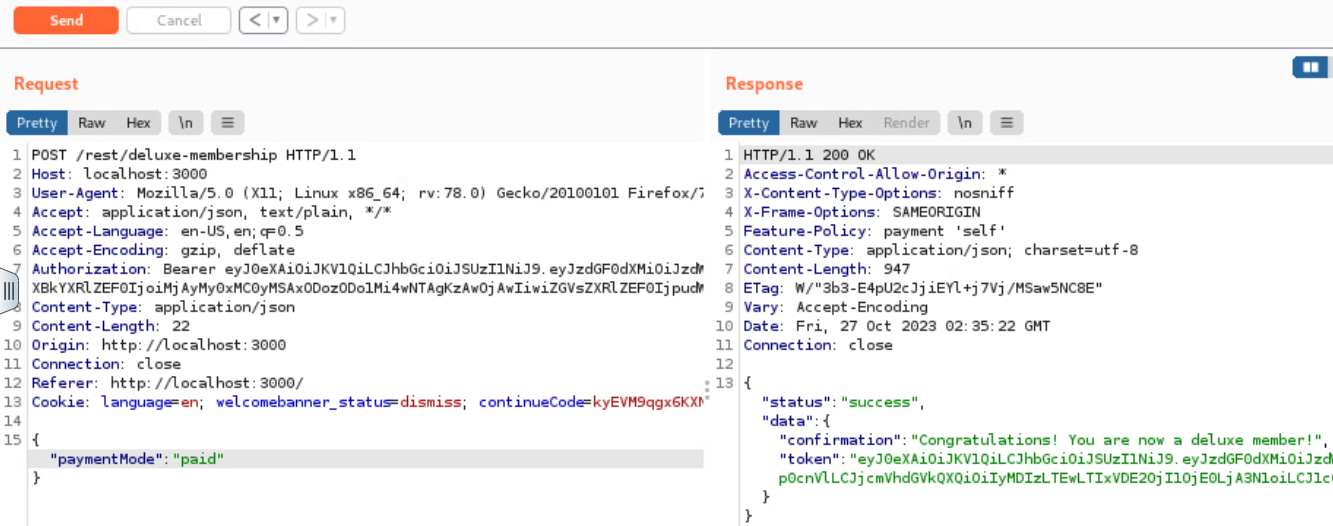
The attacker then launched Burp Suite as a proxy tool to better inspect the Juice Shops applications. This led to the discovery of the *paymentMode* which is *“wallet”*.

****

This was then sent to the Repeater in Burp Suite and request sent as is which resulted in the error message *“Insufficient funds in Wallet”.*

****

The *paymentMode* was then edited from *“wallet”* to *“paid”* which when ran resulted in a success status and confirmation message of *“Congratulations! You are now a deluxe member!”*. The Juice Shop scoreboard also reflects a completed challenge message confirming the success of the exploitation.



**Conclusion**

The Juice Shop’s cybersecurity configuration risks and vulnerabilities allowed for the successful completion of this objective. The desired outcome was to find a way to obtain a Deluxe Membership without paying for it which was achievable primarily due to security and improper input validation vulnerabilities. The following provides additional information on these risks and recommendations to make the Juice Shop more secure.

**Security Risks**

The four following risks from OWASP’s Top 10 Web Application Security Risks from 2021 are applicable to the vulnerabilities that allowed for exploitation of the feedback form: Broken Access Control, Injection, Insecure Design, Security Misconfiguration, Vulnerable and Outdated Components. Just as with the prior section, several of these have already been discussed in other exploitations, however different approaches and considerations apply to this scenario. The Broken Access Control risk is included due to the “violation of the principle of least privilege or deny by default, where access should only be granted for particular capabilities, roles, or users, but is available to anyone” (OWASP, n.d). This is similar to what was seen in task 3.4 in the ability to access and manipulate code without being a systems administrator or the Juice Shop requiring any additional credentials. Injection is rated number three in OWASP’s Top 10 from 2021 and was seen exploited when the attacker was able to send the payment method change from Burp Suite and inject it into the Juice Shop. The Insecure Design is still applicable due to the missing or ineffective control designs, as have previously been described (OWASP, n.d.-b). The Security Misconfiguration risk is due to similar concerns surrounding missing appropriate security hardening and improperly configured permissions, unnecessary features enabled, and the form code was not set to secured values that prevented tampering (OWASP, n.d.-c).

**Risk Rating**

The overall risk identified to the Juice Shop as a result of this penetration test is High. The ability to manipulate the Deluxe Membership form and to inject a change into the payment status is a major vulnerability within Juice Shop’s configuration.

**Recommendations**

The importance of security controls to an already vulnerable network cannot be understated. Due to the four primary risks discovered that contributed to the ability to obtain a Deluxe Member account without paying for it within Juice Shop, its recommended that the following security measures be implemented:

1. Implement the principle of least privilege to prevent non-system administrators from being able to access and edit source code.
2. Lock down browser and site security measures to prevent capture, alteration, and injection of potentially malicious code.
3. Incorporate hardening processes and regularly update configurations for security notes, updates, and patches to prevent unauthorized access and tampering (OWASP, n.d.-c).

**References:**

OWASP. (n.d.). A01:2021 – Broken Access Control. A01 Broken Access Control - OWASP Top 10:2021. https://owasp.org/Top10/A01\_2021-Broken\_Access\_Control/

OWASP. (n.d.-b). A04:2021 – insecure design. A04 Insecure Design - OWASP Top 10:2021. https://owasp.org/Top10/A04\_2021-Insecure\_Design/

OWASP. (n.d.-c). A05:2021 – security misconfiguration. A05 Security Misconfiguration - OWASP Top 10:2021. https://owasp.org/Top10/A05\_2021-Security\_Misconfiguration/

OWASP. (n.d.-d). A06:2021 – vulnerable and outdated components. A06 Vulnerable and Outdated Components - OWASP Top 10:2021. https://owasp.org/Top10/A06\_2021-Vulnerable\_and\_Outdated\_Components/

OWASP. (n.d.-e). A07:2021 – identification and authentication failures. A07 Identification and Authentication Failures - OWASP Top 10:2021. https://owasp.org/Top10/A07\_2021-Identification\_and\_Authentication\_Failures/

OWASP. (n.d.-?). *OWASP Top 10 2021*. OWASP Top 10:2021. https://owasp.org/Top10/