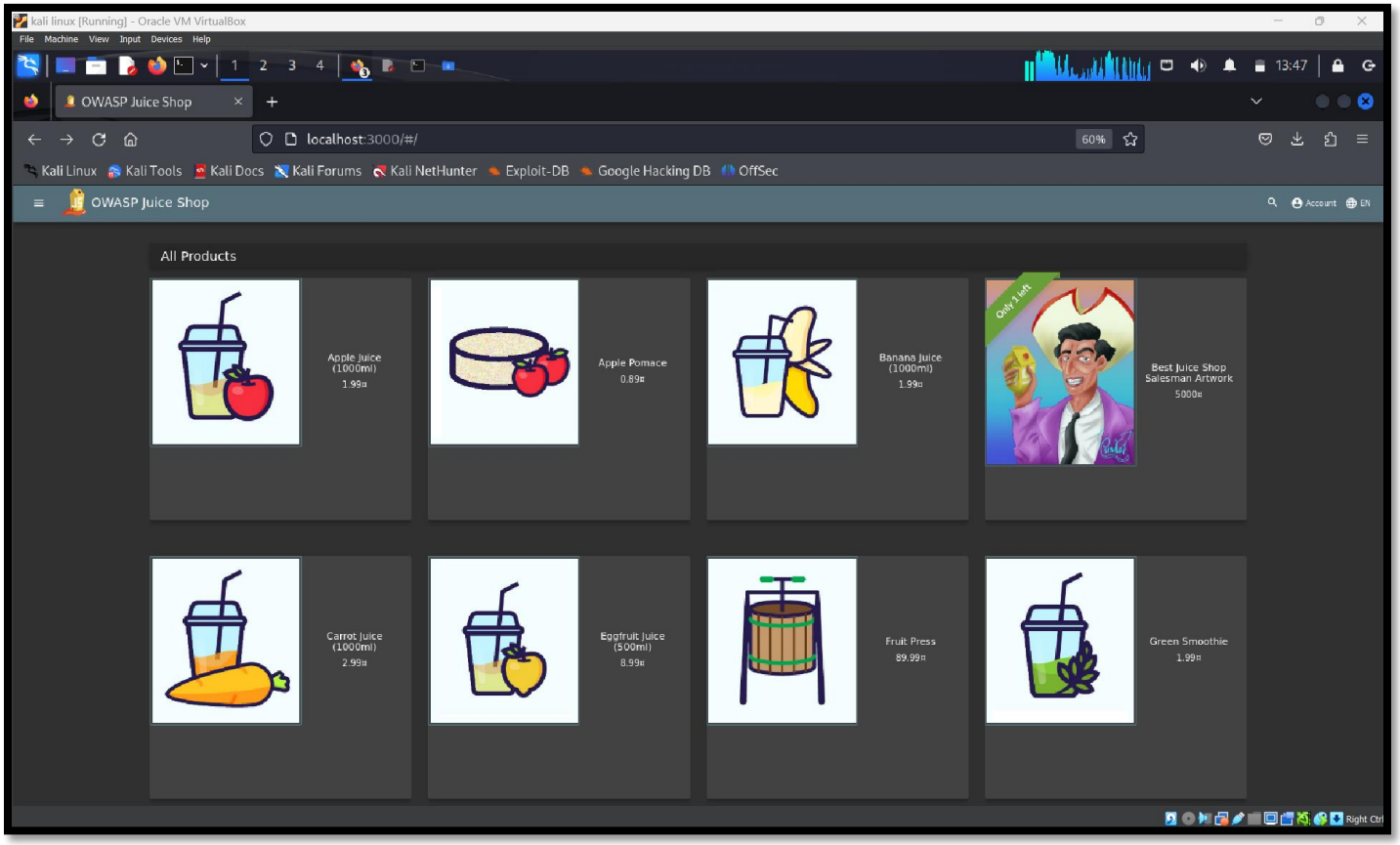
Subject: Security Testing

SENG 8061

Project Part - 02

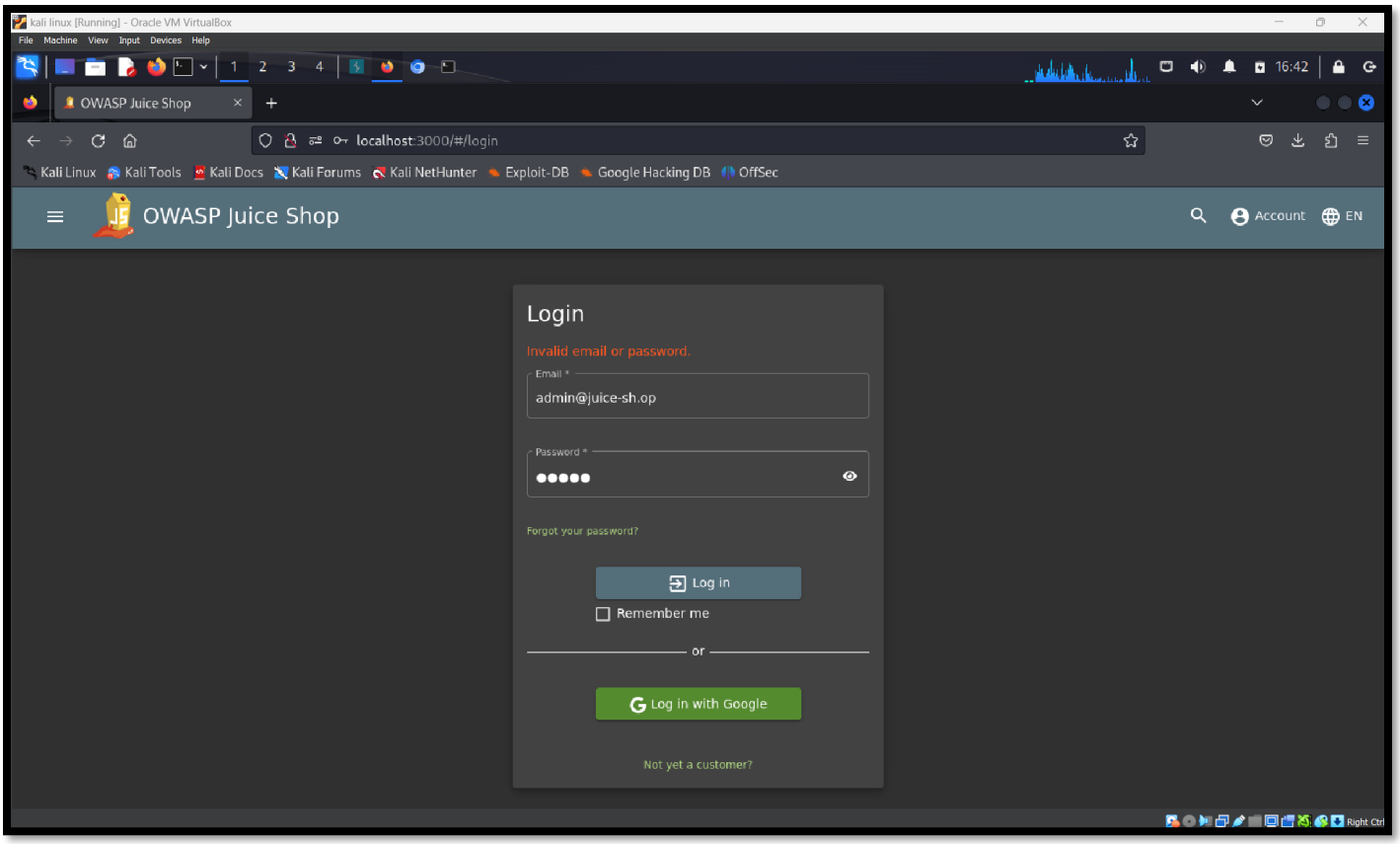
Name	Student ID
Shivani Varu	8941914
Mohammed Rafique	8954785

Screenshot 1: Go to http://localhost:3000/

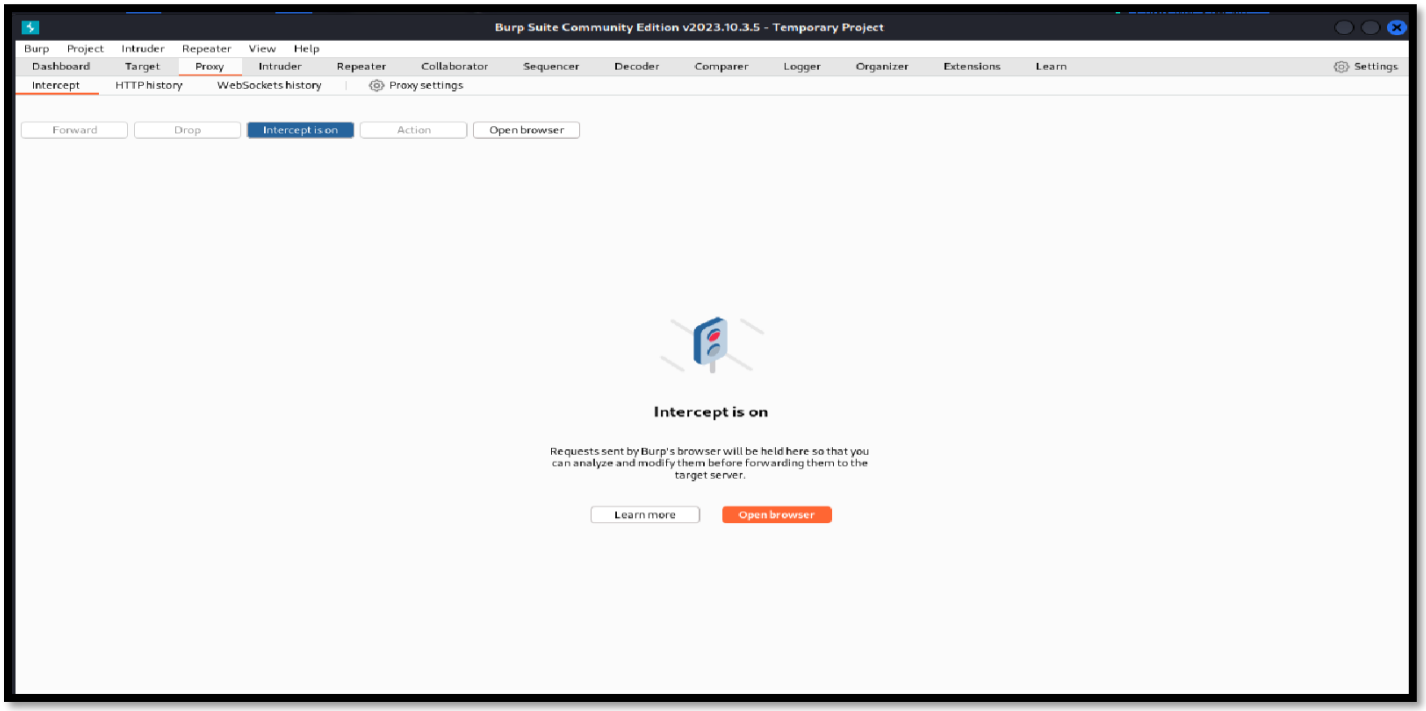


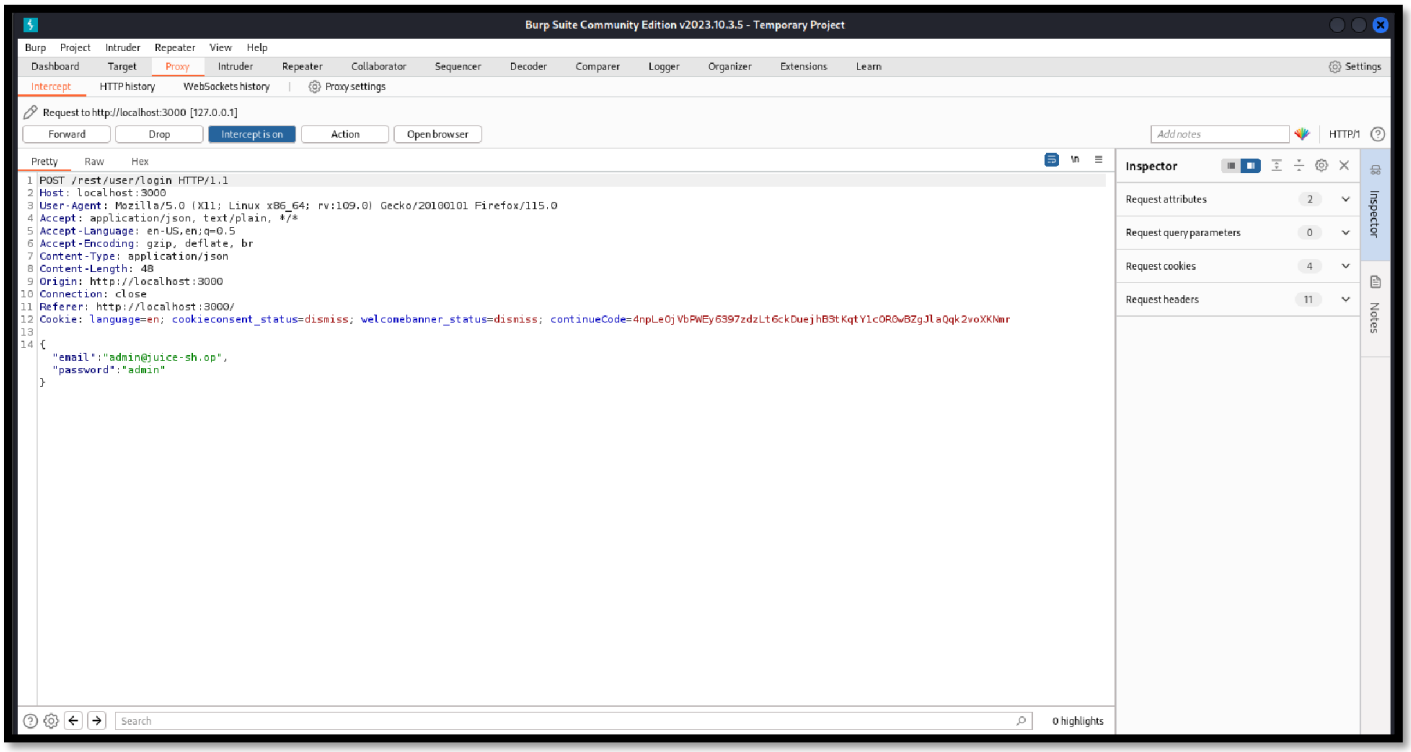
TASK 1
Log in with the administrator's user credentials without previously changing them or applying SQL Injection.

In our demonstration, we logged in to the provided URL using the administrator's email (admin@juice.sh.op) and a default password (admin123). However, we received an invalid email or password error. We then utilized Burp Suite to intercept the request and analyzed it further.

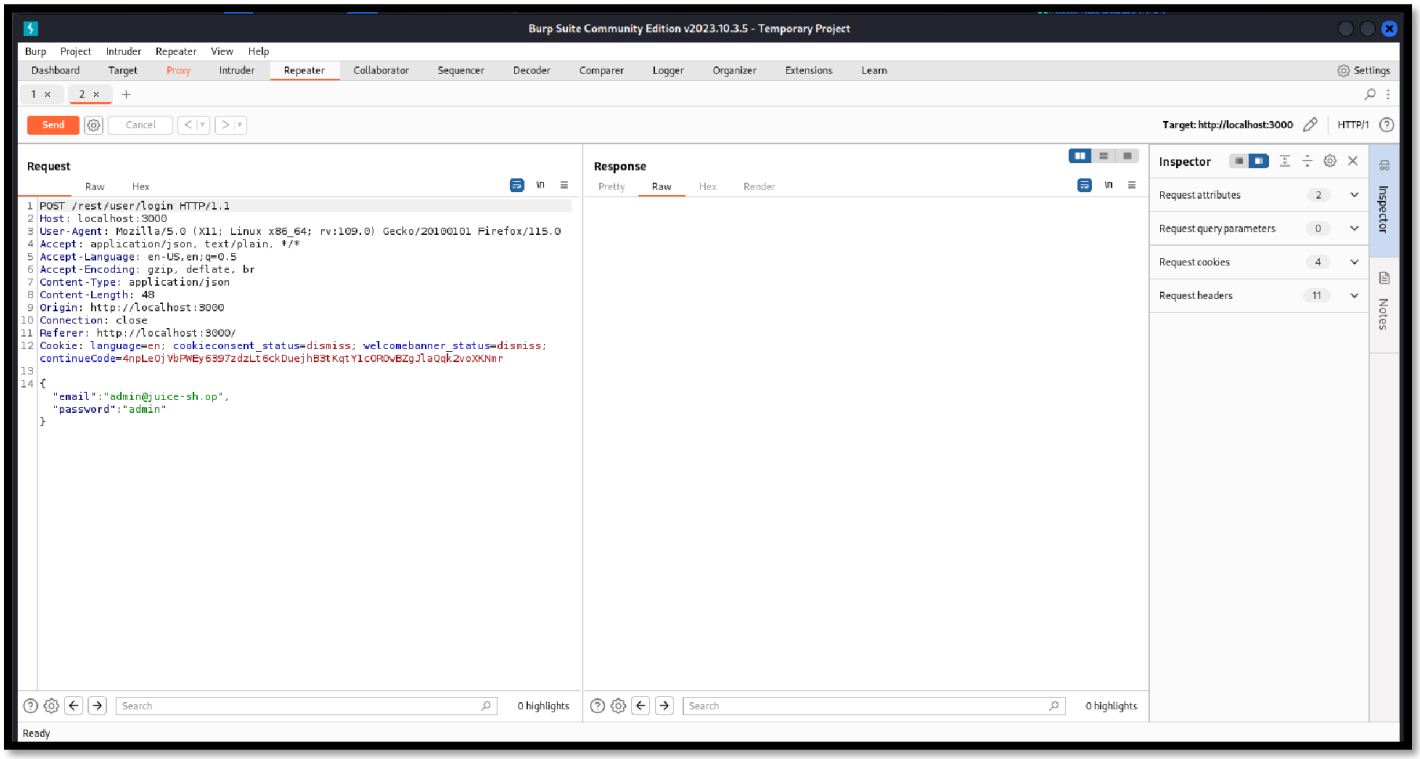


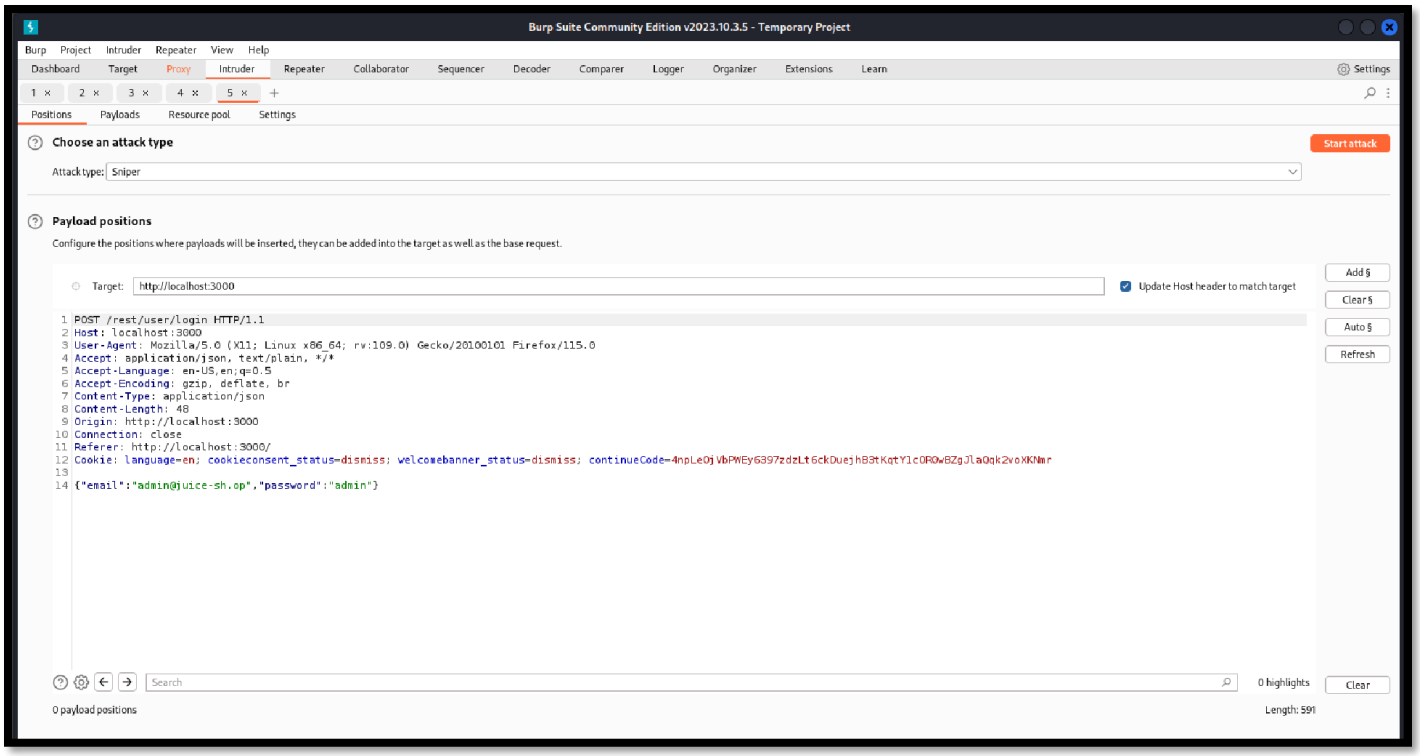
When we refresh browser, we were able to see intercept page, we click on forward button to check what we entered in username password, then we right click on moue to send to repeater.



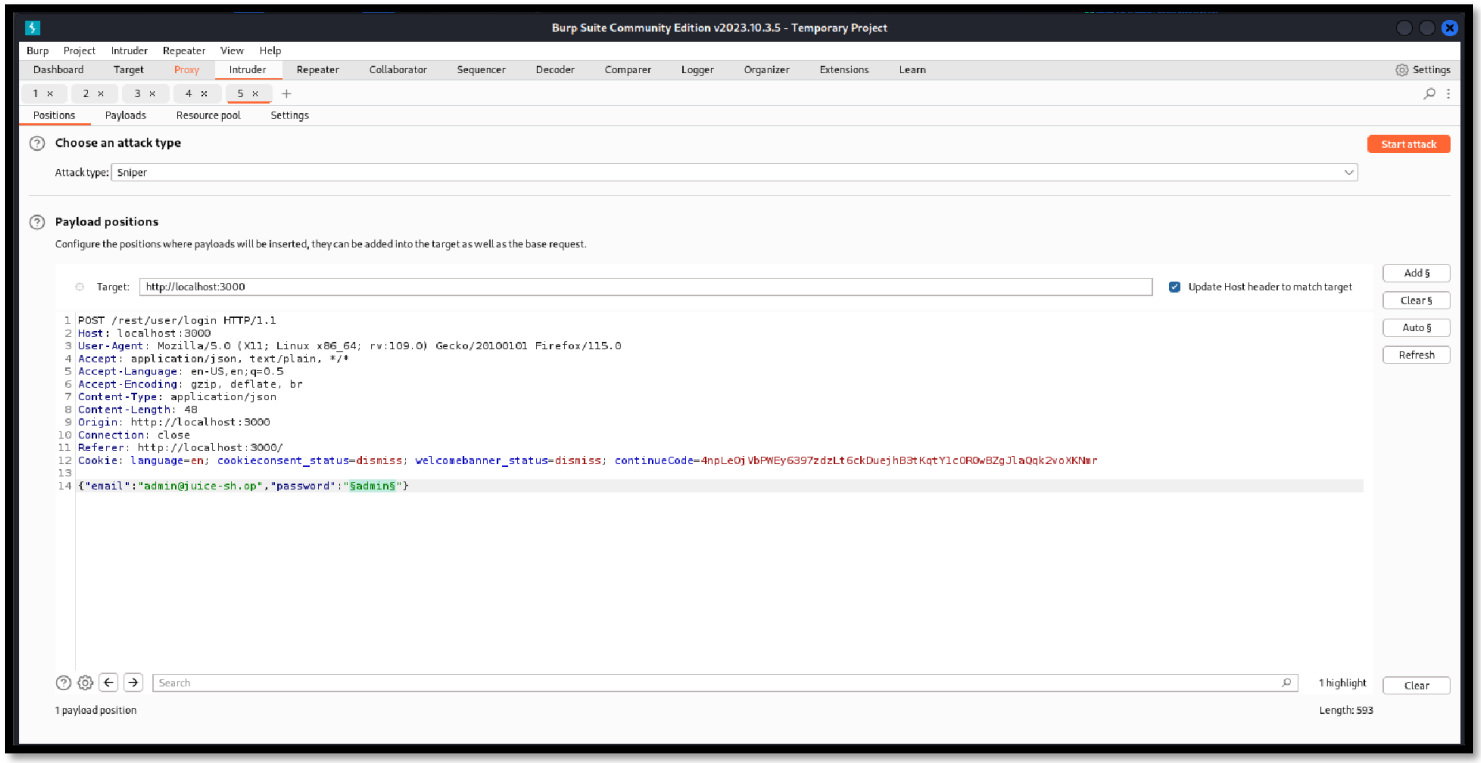


Then we sent this to repeater



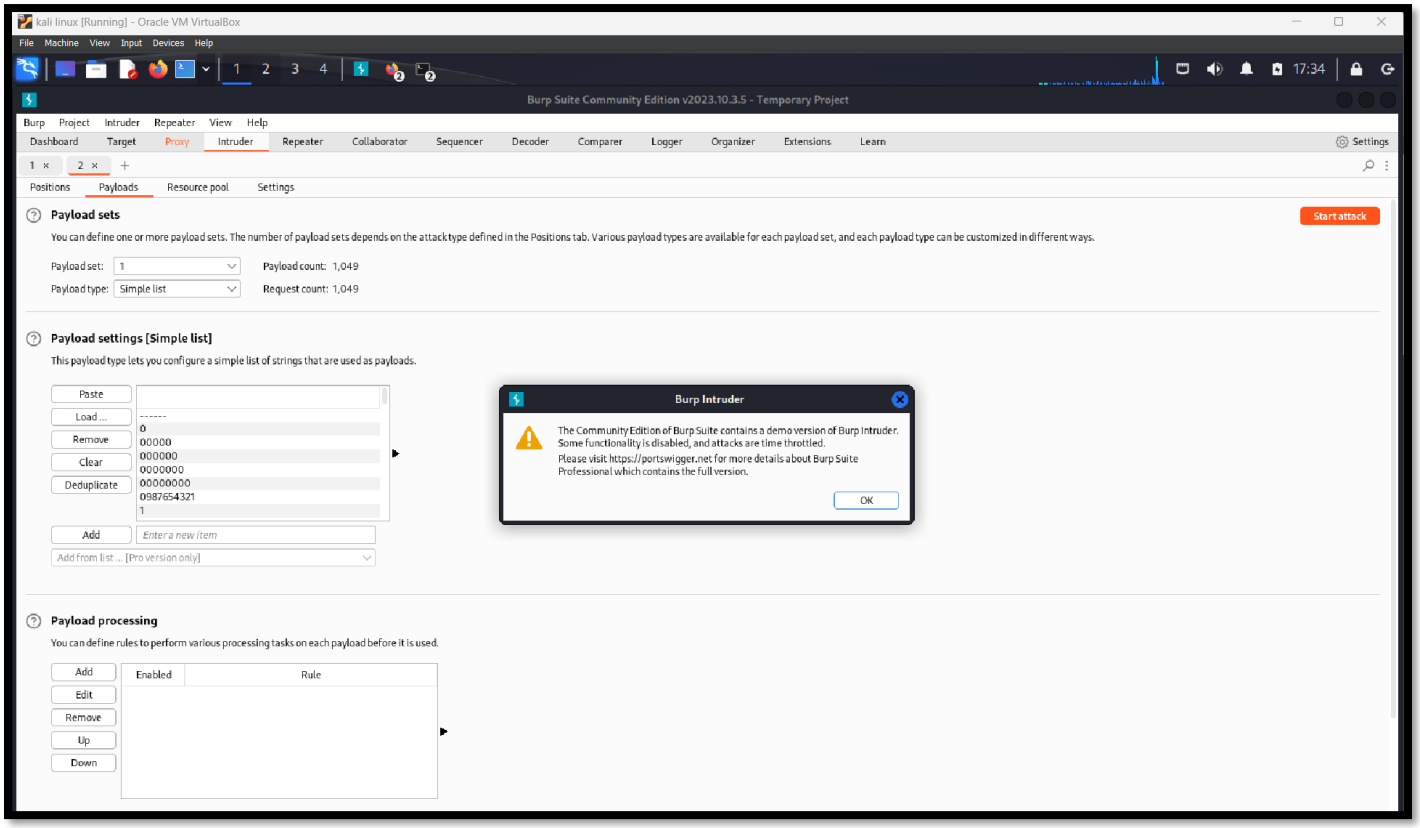


Then we'll use the intruder to automate the passwords required to access the administrator's account. The invader is a powerful tool for automating strings. In order to list the admin password, we will use the intruder payload from the burp suite.

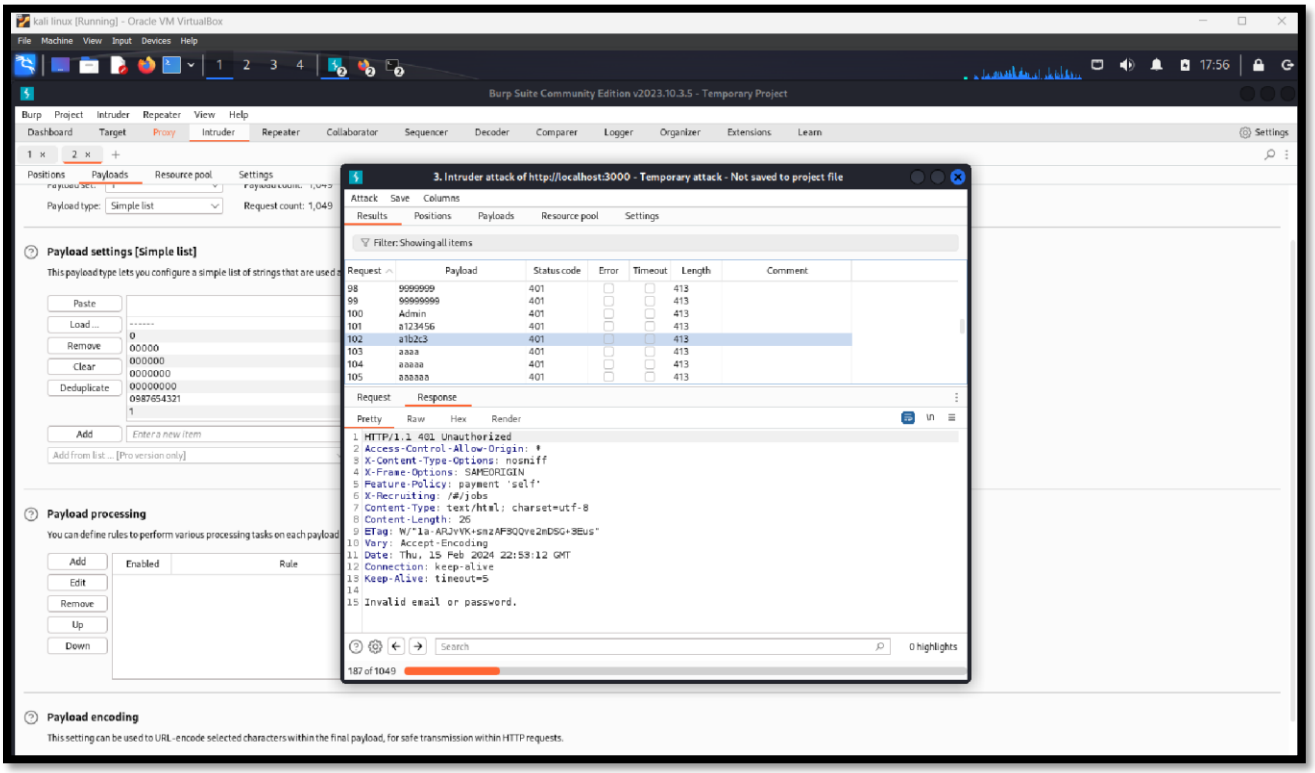


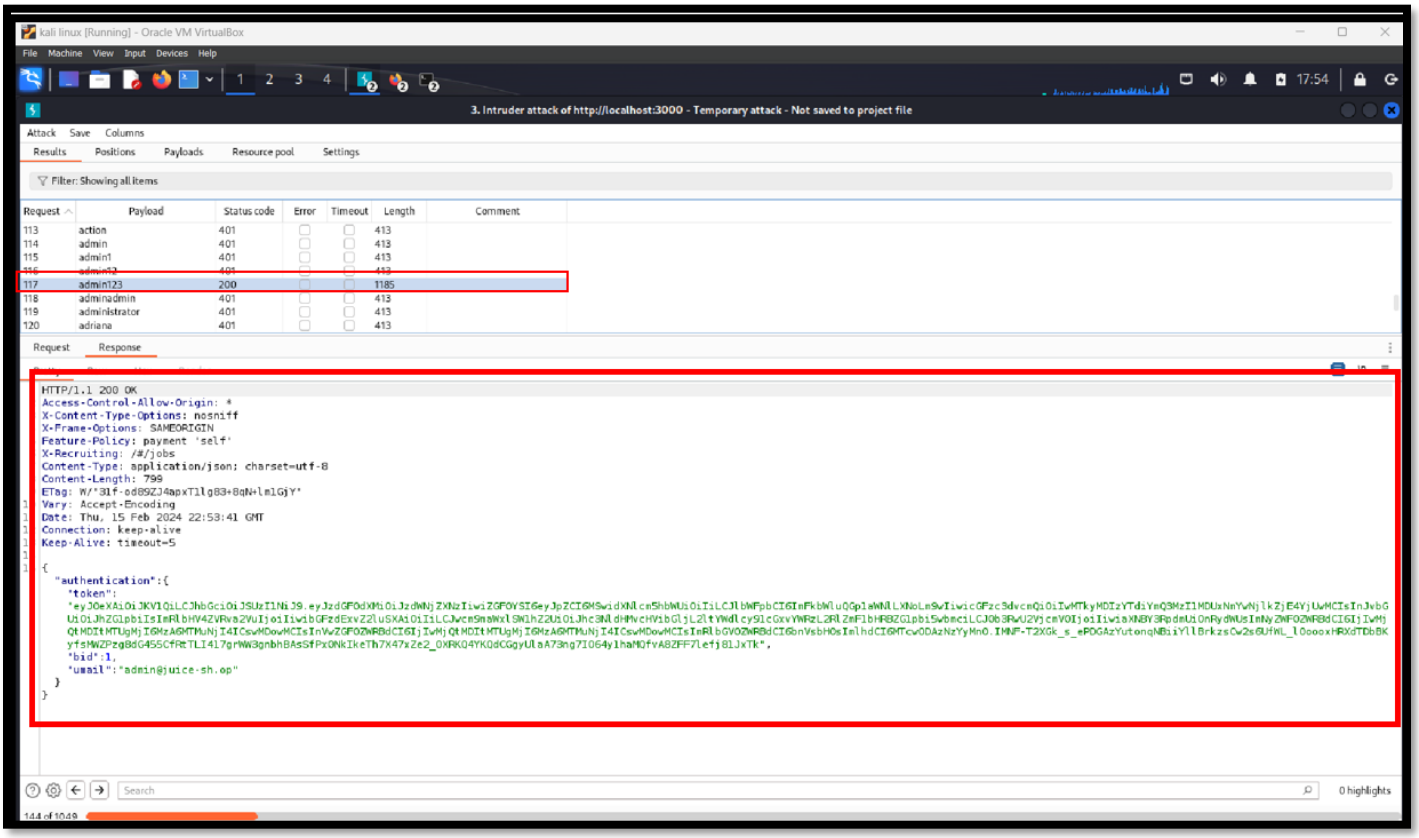
Next, we choose admin as the password and click the Add button. Next, we try to set up the password parameter in the burp and add password payloads to this password parameter by using the intruder.

Next, we selected the payload tab and loaded files from the supplied best105.txt file from eConestoga, which included many frequently used passwords. Next, we click the "Start Attack" button to launch an attack. A dialog box appears, and we click the "OK" button to proceed.

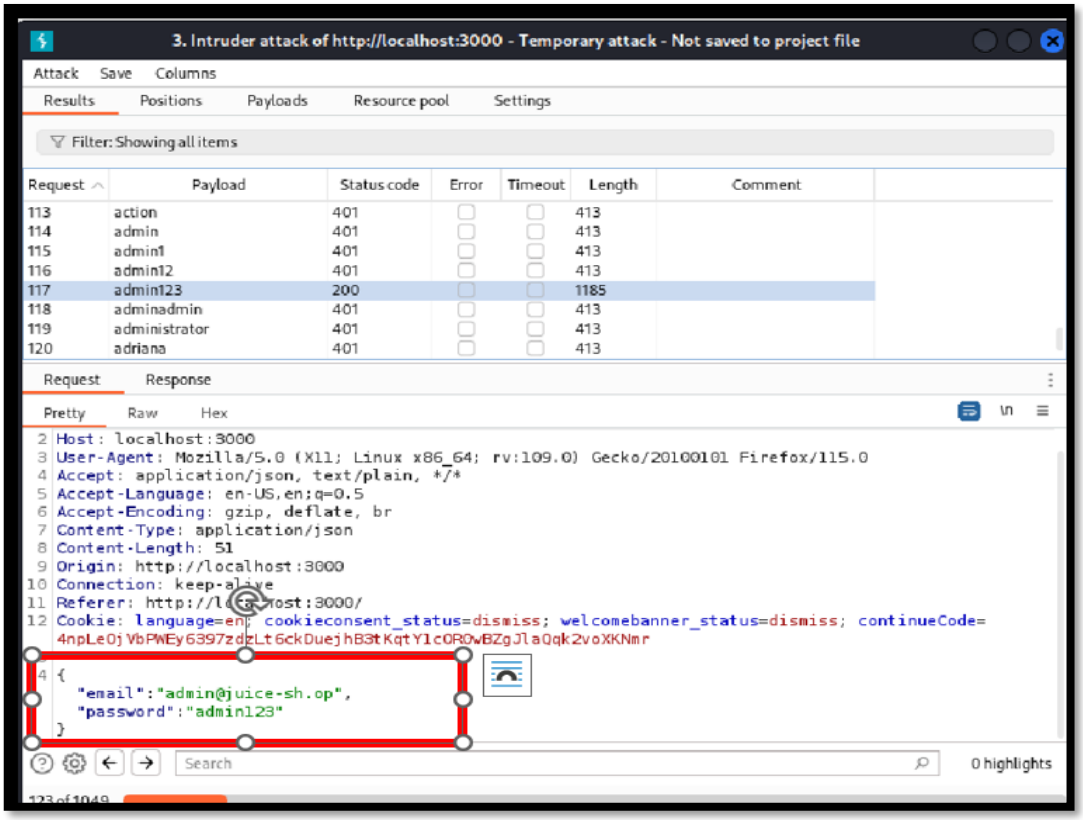


We attempted to filter strings with invalid email addresses or passwords using the intruder, and then we initiated an attack.

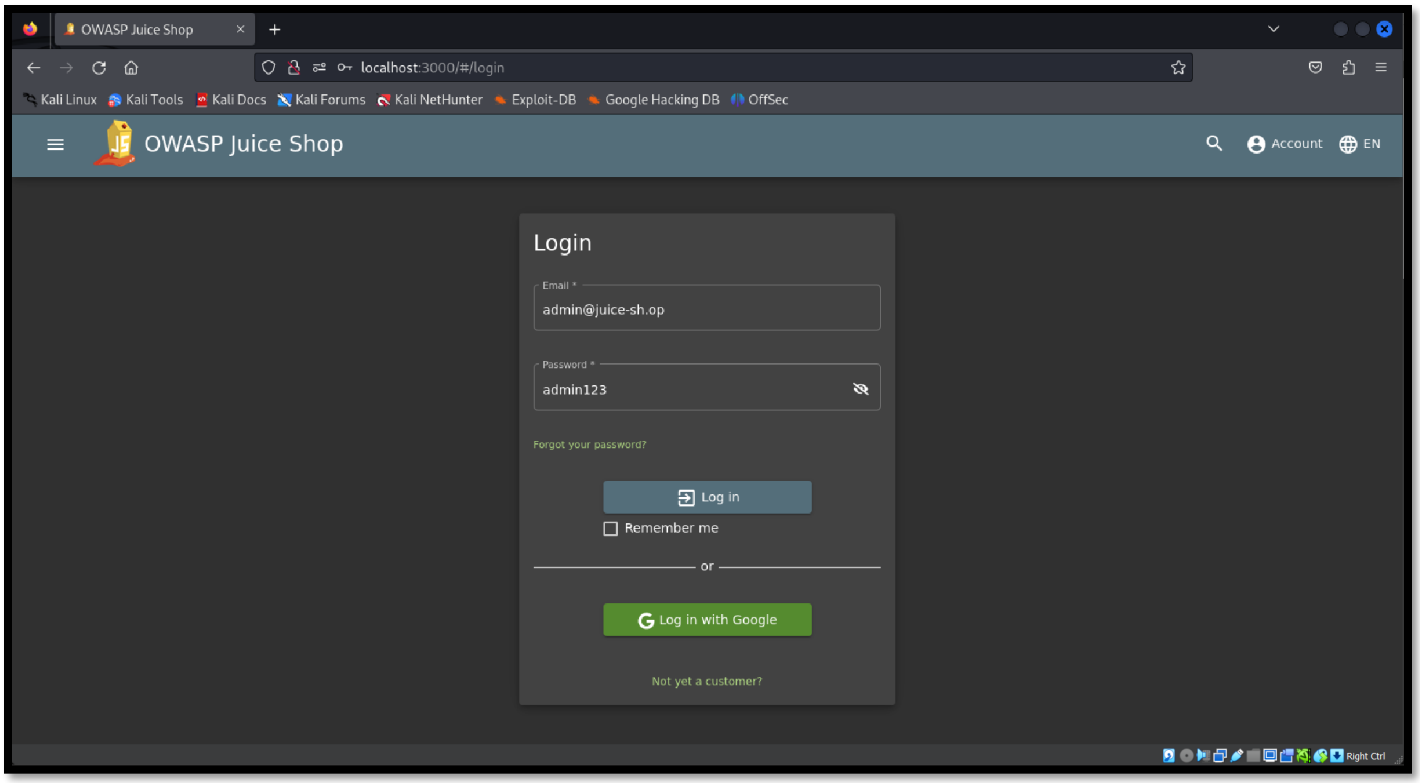




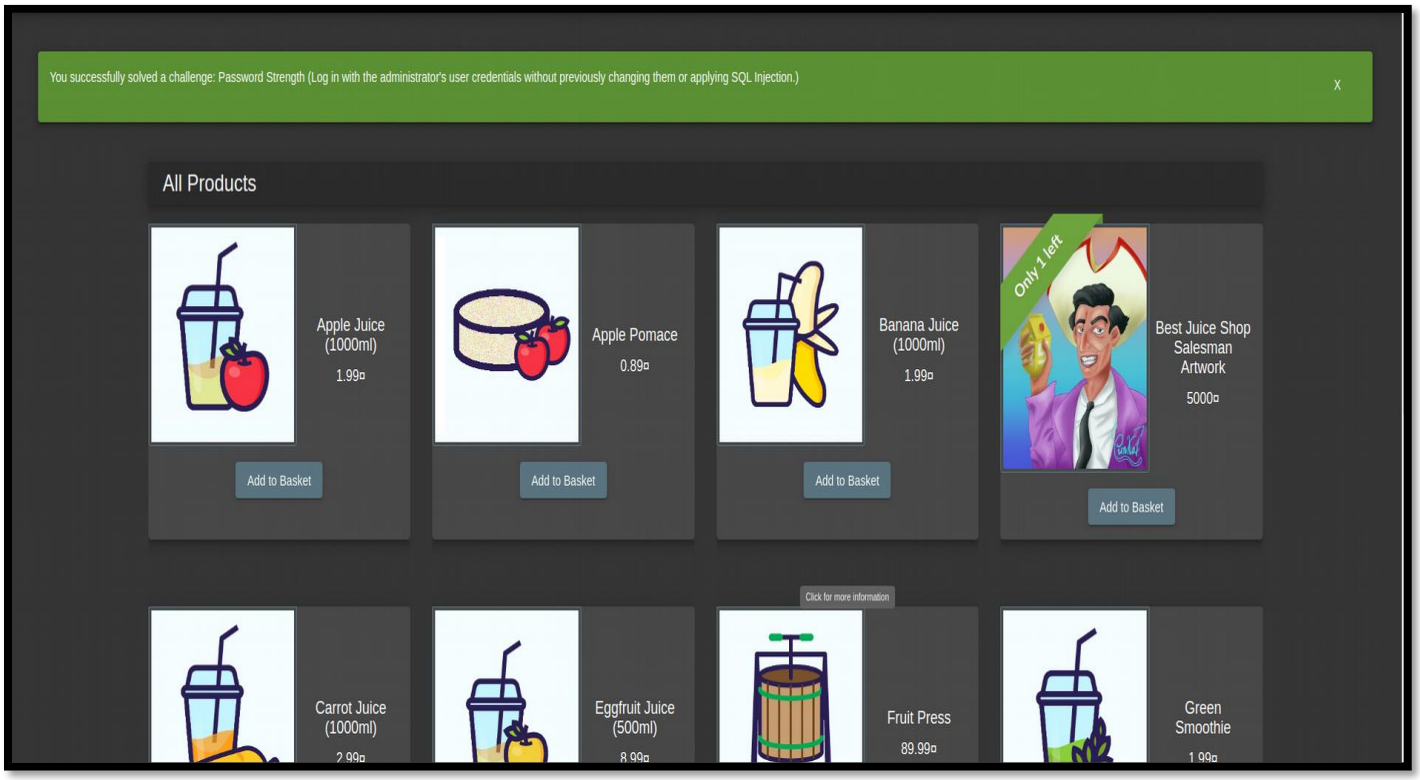
Here, we observed that the payload admin123 has a length of 1185, which appears to be different from other payloads. As a result, we were able to identify it and observe that the status code is 200, indicating that the admin password is admin123 and that an authentication token is included in the response.



We discovered the admin password when inspecting the requests.



We completed this task by listing the passwords with the Burp Suite program.



Vulnerabilities Exploited by Task 1:

- Weak Password Mechanism
- Lack of Account Lockout Mechanism
- Insufficient Input Validation
- Modifications of Requests Generated by the Client-Server

How Vulnerabilities Can Be Fixed or Addressed:

- Multifactor authentication for email addresses and passwords
- Strong password enforcement (>8 characters with special characters)
- Unique passwords for each account
- Session timer for automatic logouts
- Limited login attempts
- Application firewalls for website protection
- CAPTCHA or Two-Step Verification
- Input Validation and Sanitization

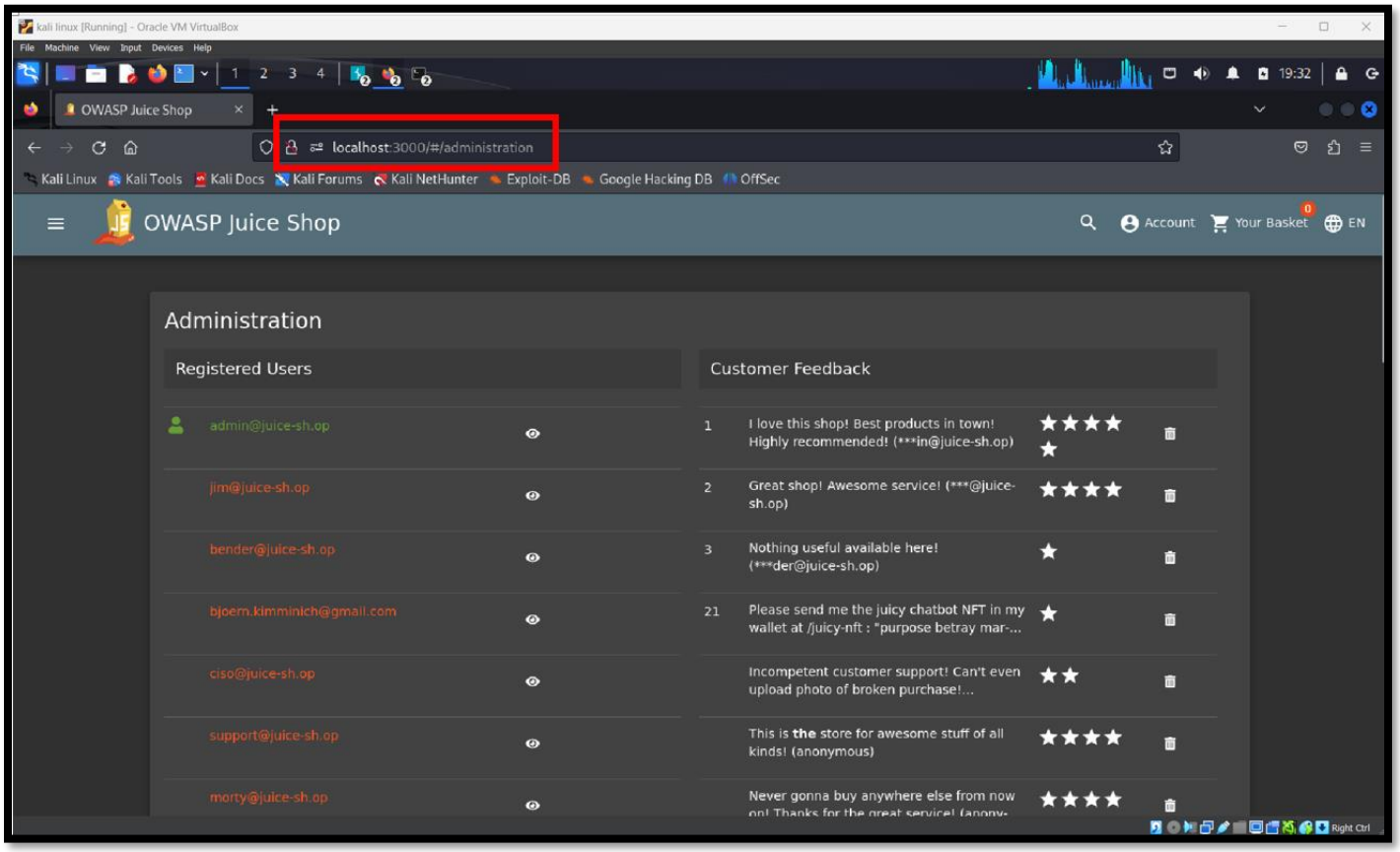
Real-World Scenarios:

- Exploitation of Weak Passwords
- Full Application Compromise
- Access to Sensitive Data
- Unlimited Login Attempts

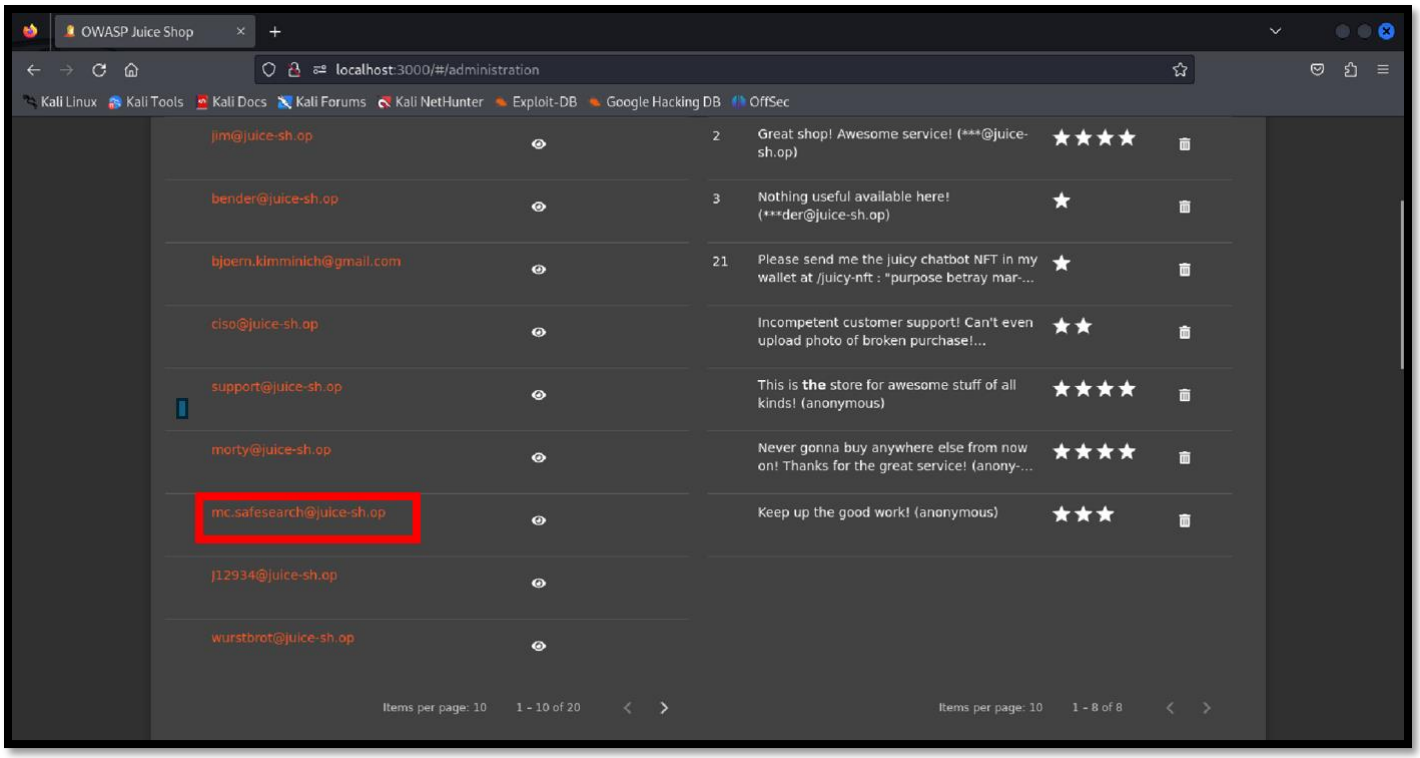
TASK 2

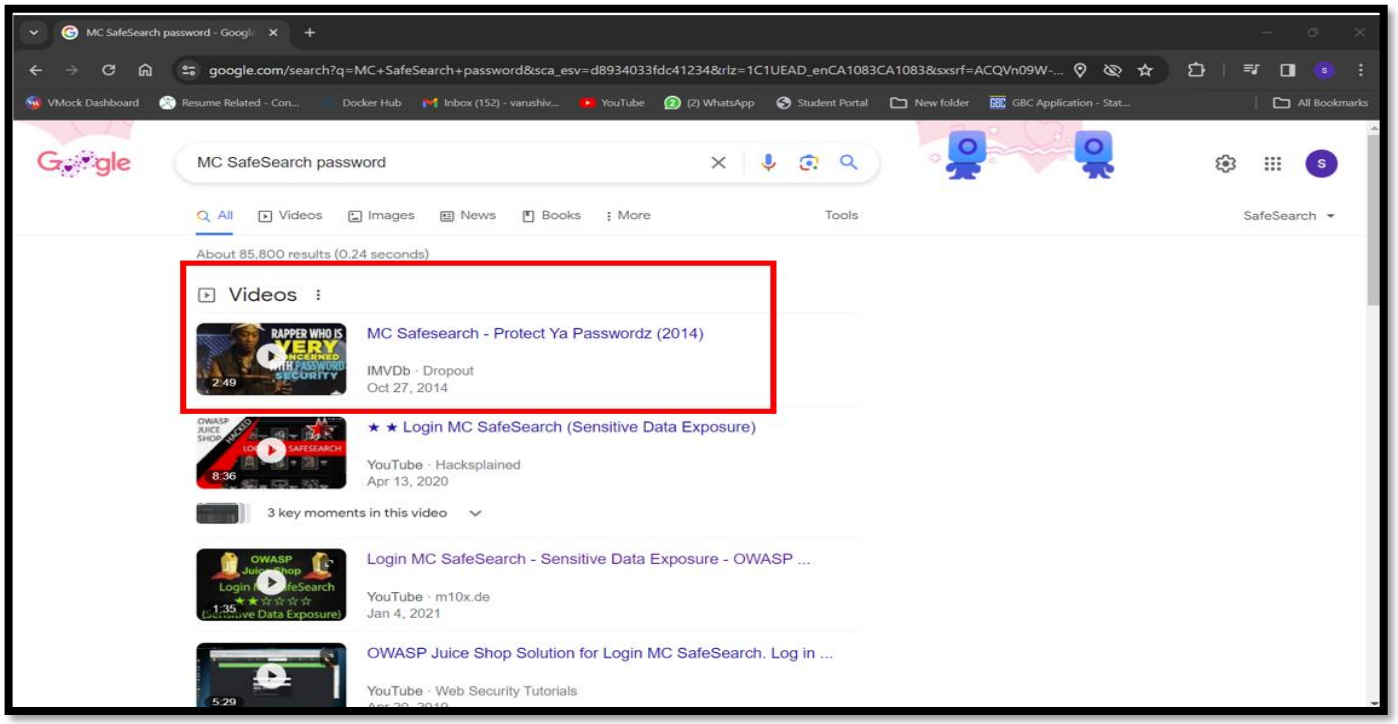
Log in with MC SafeSearch's original user credentials without applying SQL Injection or any other bypass.

In this scenario, we had already obtained the admin's username and password from the previous task. With this information, we accessed the administration page directly from localhost/#/administration, as we had done in our last project's part 1, task 2's admin section.

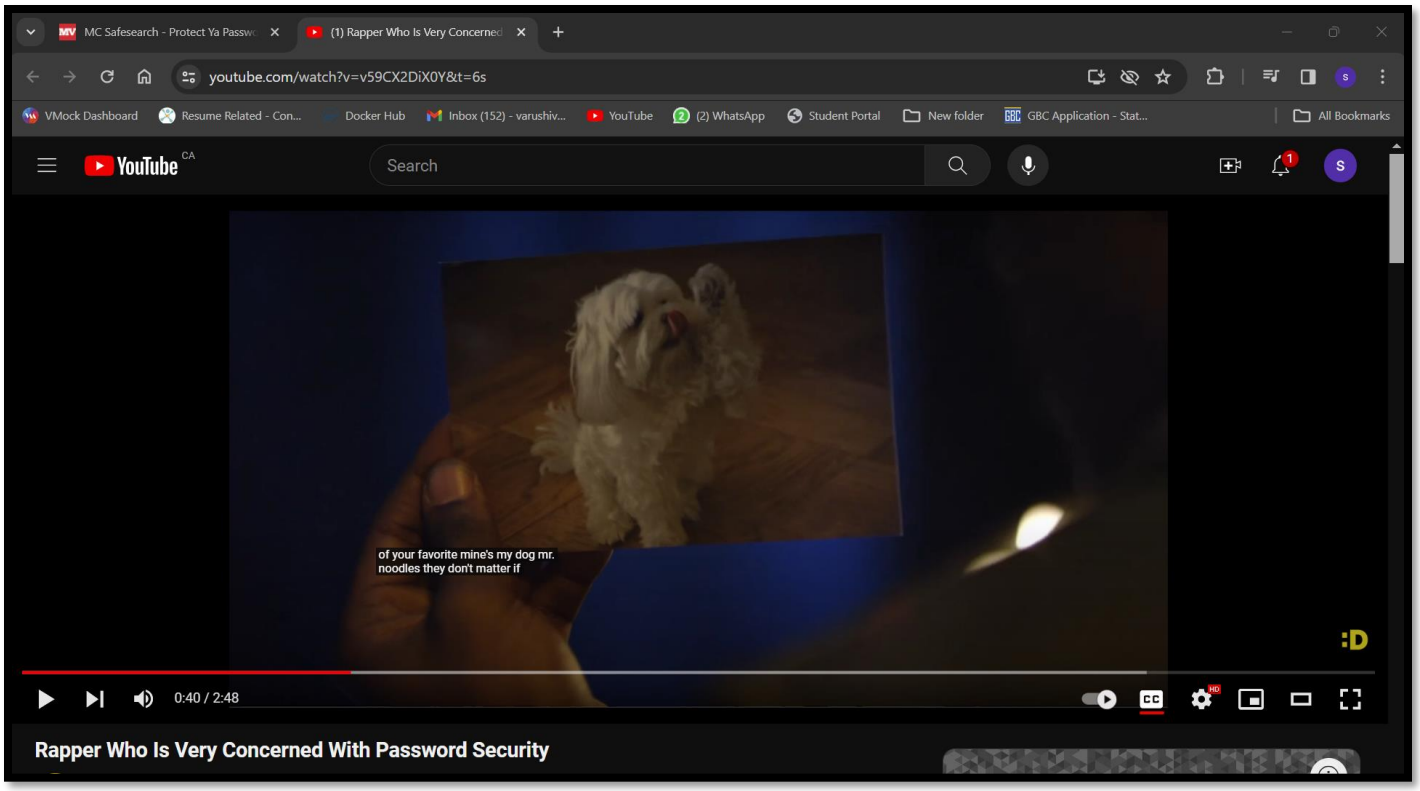


Upon reaching this page, we discovered that the username was mc.safesearch, which we promptly copied. To uncover the password, we searched for his name in the Google search bar and stumbled upon one of his songs, which provided us with the necessary clue.

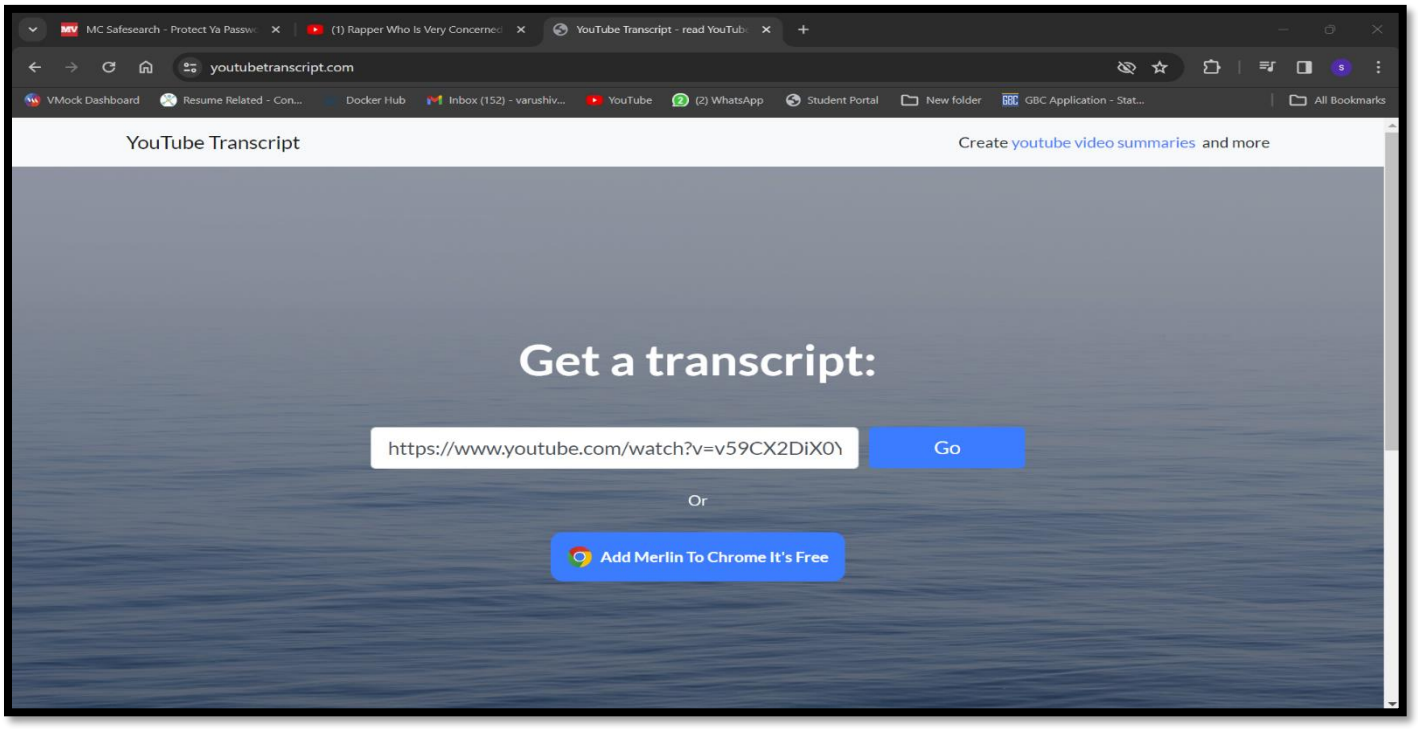




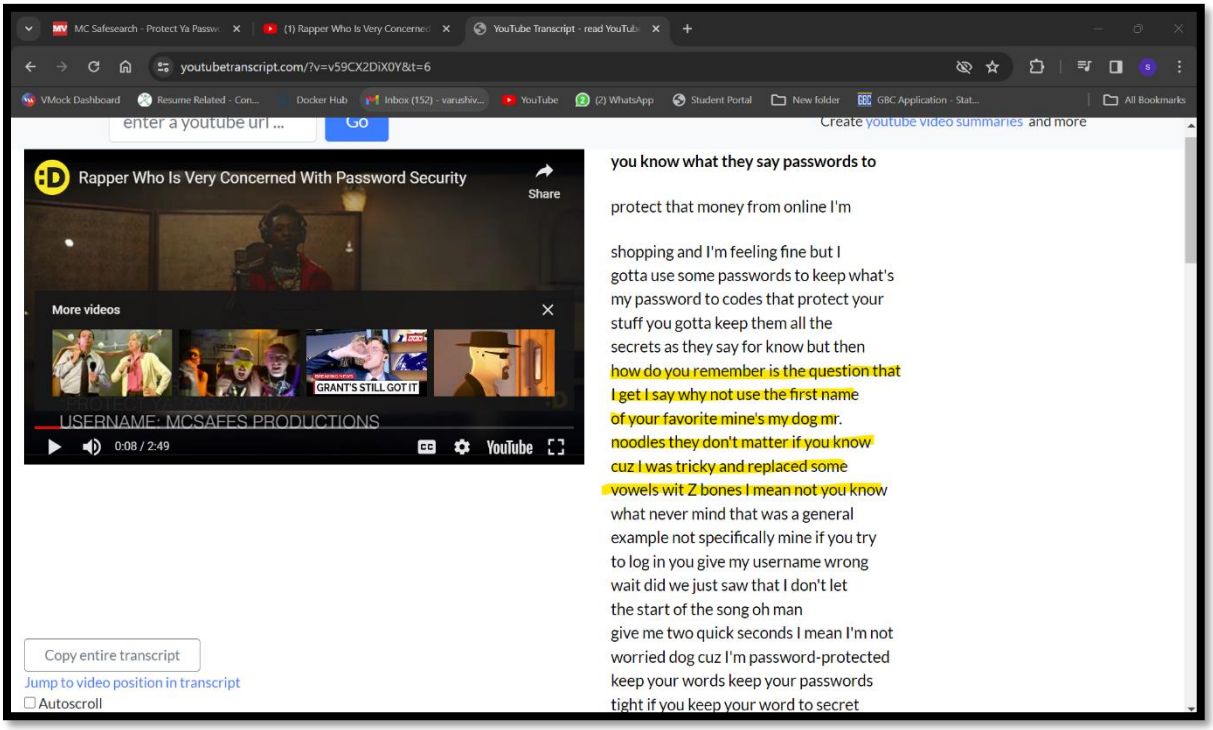
therefore, we found his youtube video song, we watched it, and able to find his password, by watching his lyrics as captions.



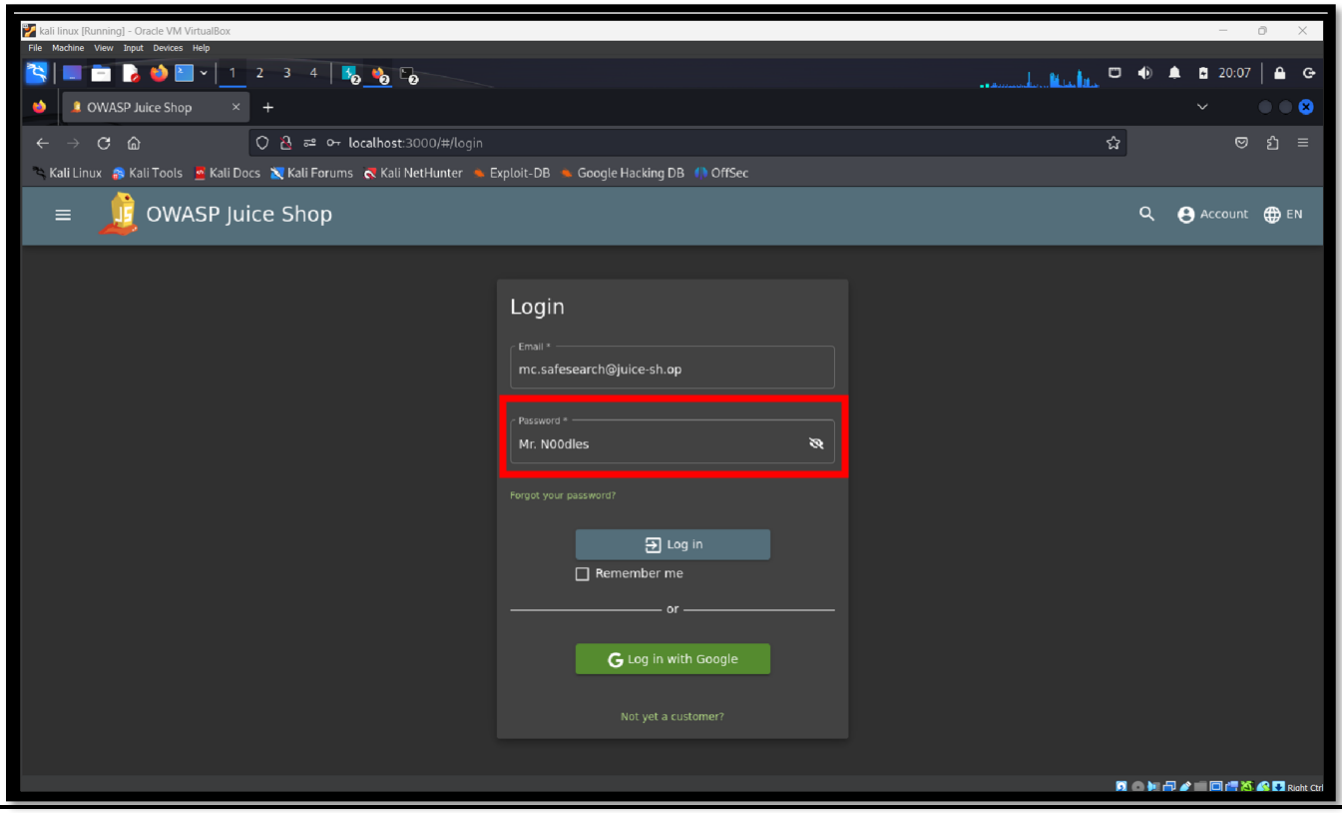
to see properly his lyrics, we search on google for YouTube video lyrics, we open the website and paste the YouTube link on it, view whole the lyrics.



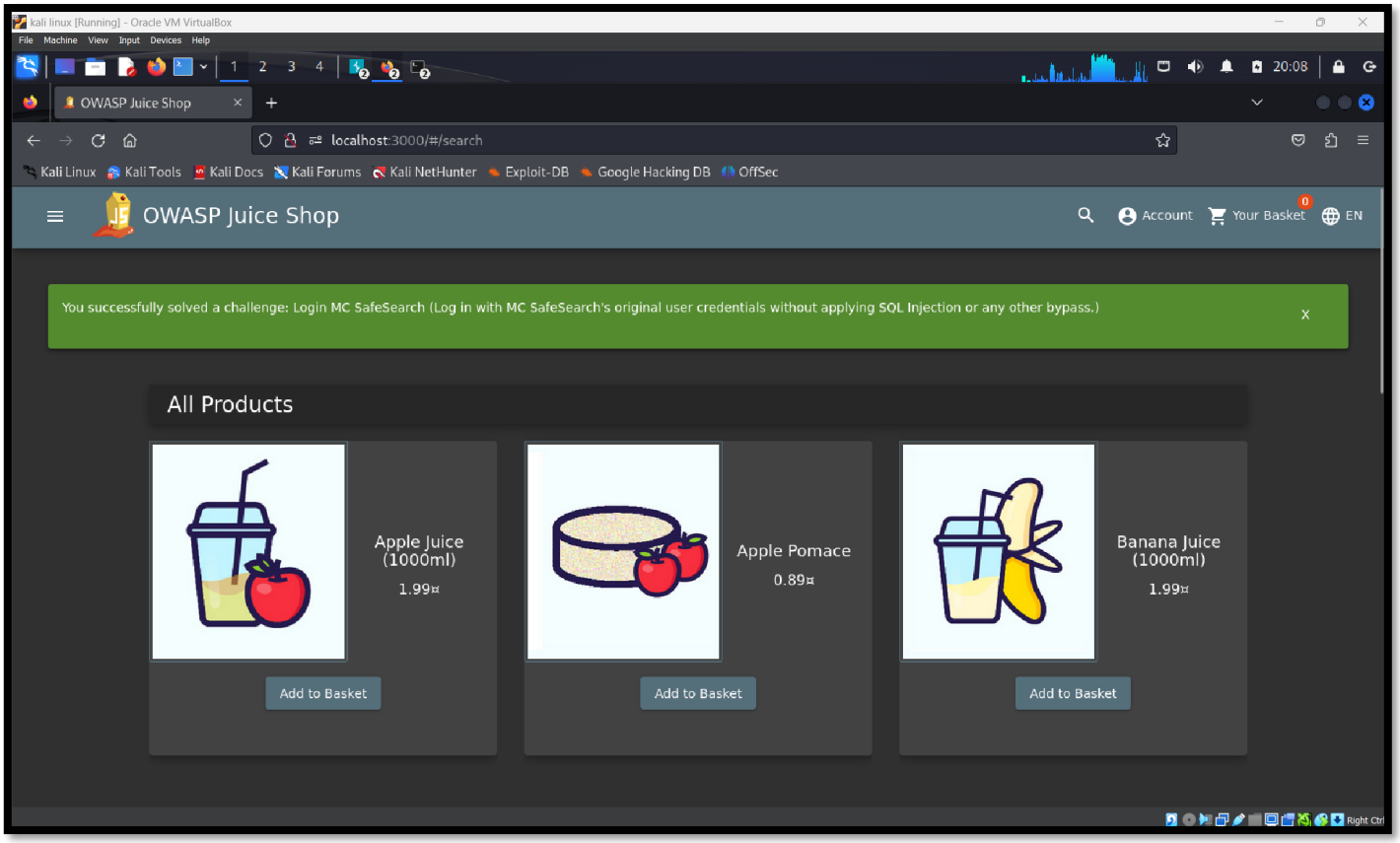
In this song, he explains how to remember your password with the spoken word. He states that " you can use the name of your favorite pet. For example, Mr. Noodles is my favorite pet." In the rap, he also states, "If you know my password, I'll replace the vowels with zeros, so it doesn't matter if you say no."



By watching his YouTube video song and paying close attention to the lyrics as captions, we were able to decipher his password. In the song, he suggested using the name of one's favorite pet as a password, giving "Mr. Noodles" as an example. Additionally, he mentioned replacing vowels with zeros in his password for security. Based on these hints, we came to the conclusion that the password might be "Mr.N00dle," with the vowels "O" changed to zeros per his directive.



Following this finding, we logged out of the administrative page and attempted to get in using the original MC SafeSearch user credentials. In the end, we were able to access the account and therefore do this work successfully.



Vulnerabilities Exploited by Task 2:

- Weak Password Creation
- Absence of Account Lockout Mechanism

How Vulnerabilities Can Be Fixed or Addressed:

- Recommend the use of strong, unique passwords free of identifiable personal information.
- Implement two-factor authentication.
- Implement strict guidelines for password length.
- Implement account lockouts.

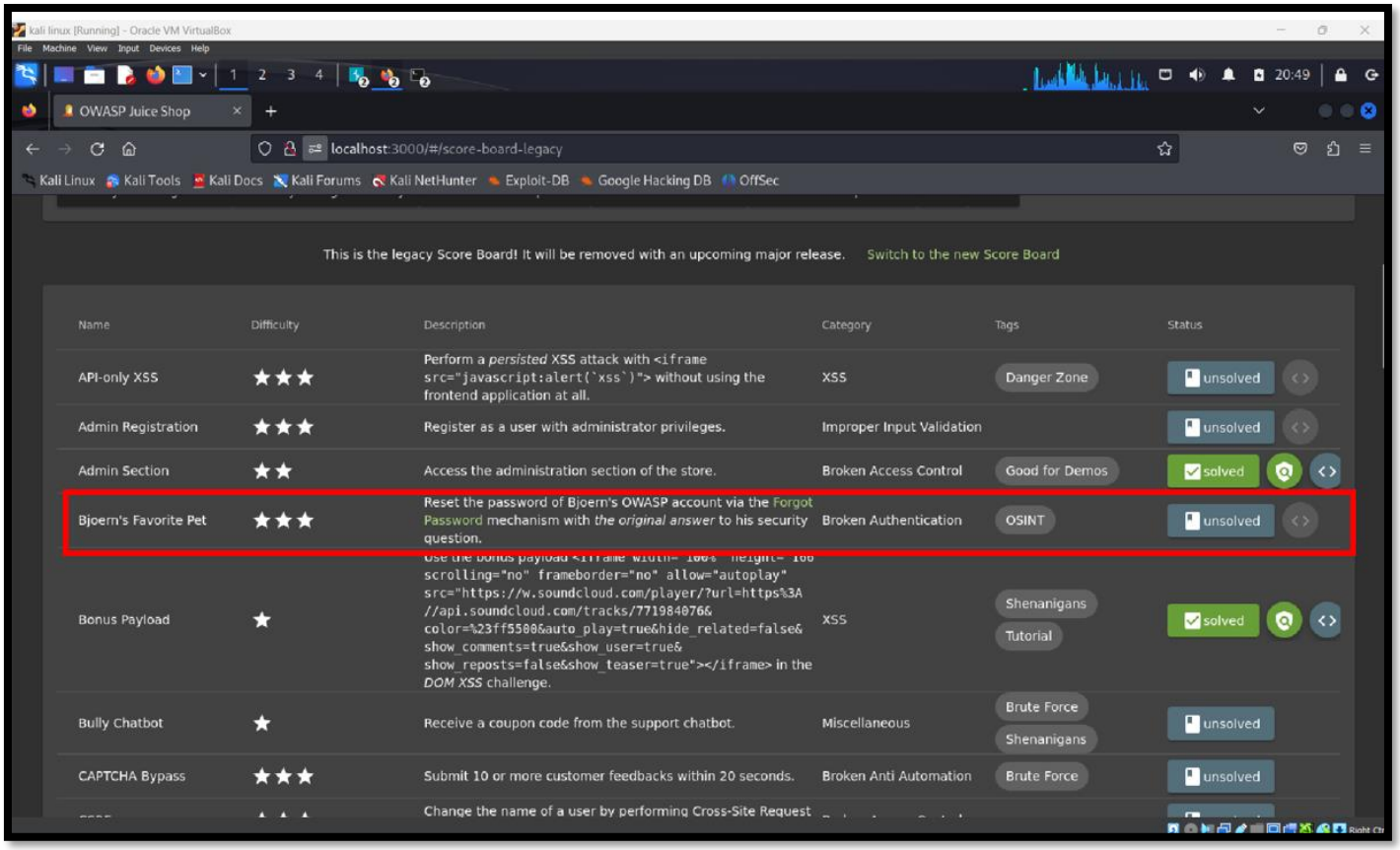
Real-World Scenarios:

- Weak passwords (such as those derived from pet names or other sensitive information) may allow unwanted access.
- For example, the LinkedIn breach in 2012 (MC SafeSearch - Protect Ya Passwordz (2014), 2014)
- The Adobe breach in the same year resulted in the theft of almost three million encrypted customer credit card records and login data for 38 million "active users," highlighting the impact of weak password security. (MC SafeSearch - Protect Ya Passwordz (2014), 2014)

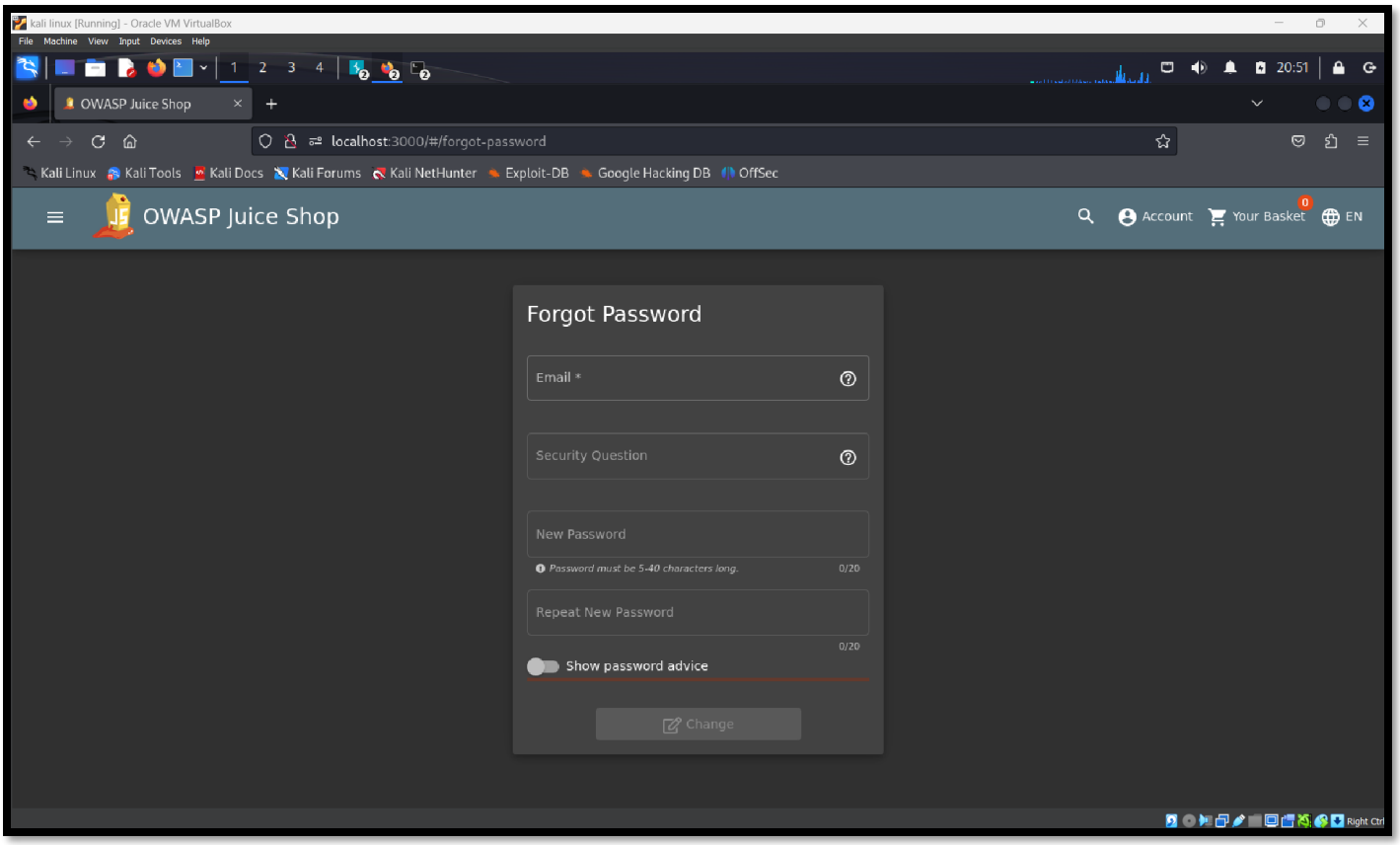
TASK 3

Reset the password of Bjoern's OWASP account via the Forgot Password mechanism with the truthful answer to his security question.

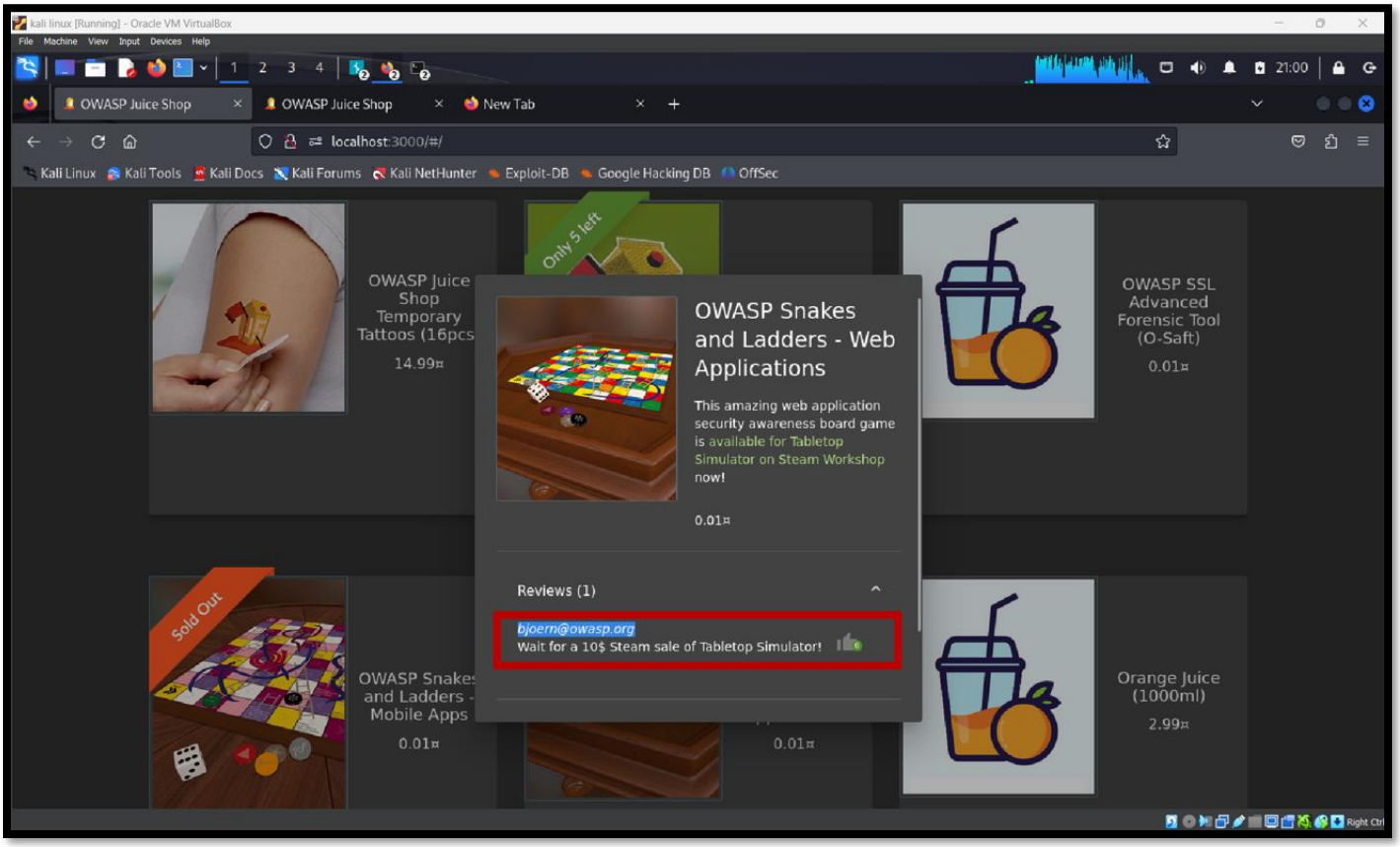
We analyze the OWASP Juice Shop security vulnerabilities caused by broken authentication. The purpose of the "Forgot Password Screen" was to retrieve the email address and security question in order to reset the login password. The answer to the security question is to track data uploaded by users. To solve this task, we apply the Björn's Favourite Pet approach.



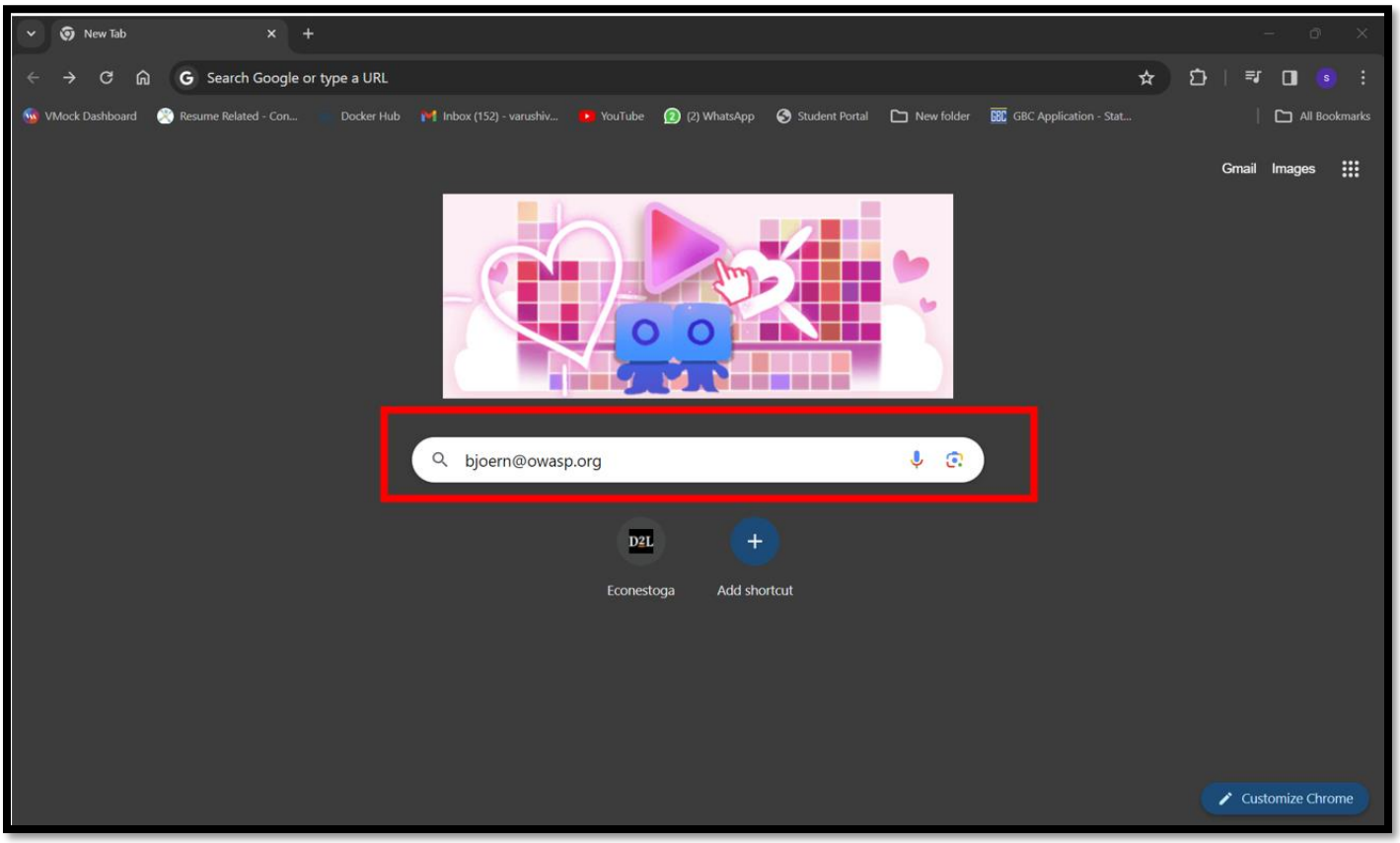
To reset the password, we must first click on the forgot password link. This task has three stars.



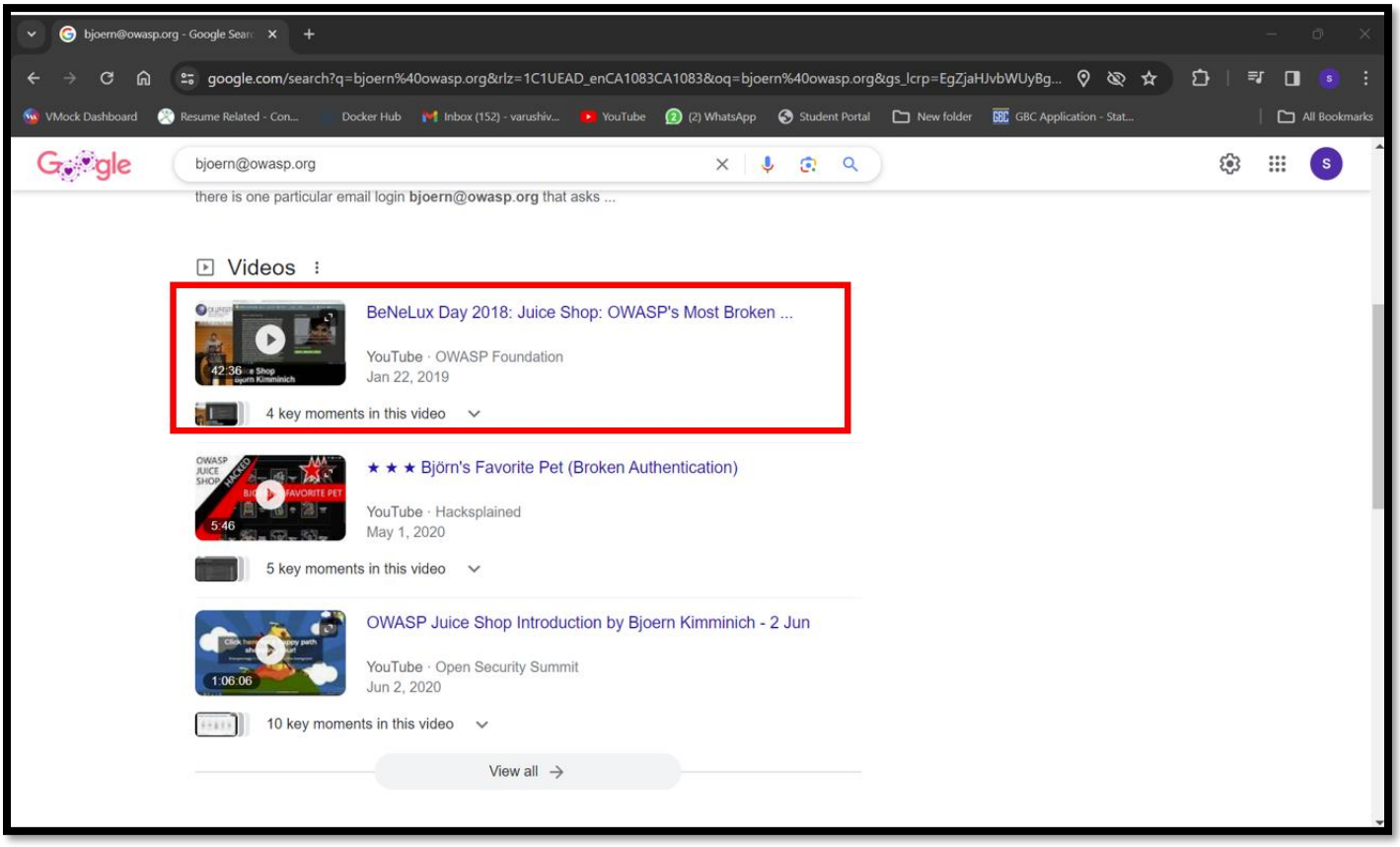
To find bjorn's user name we went through all the OWASP Juice shop website, we tried to find out details which are relevant to bjorn, and finally we were able to found his username.



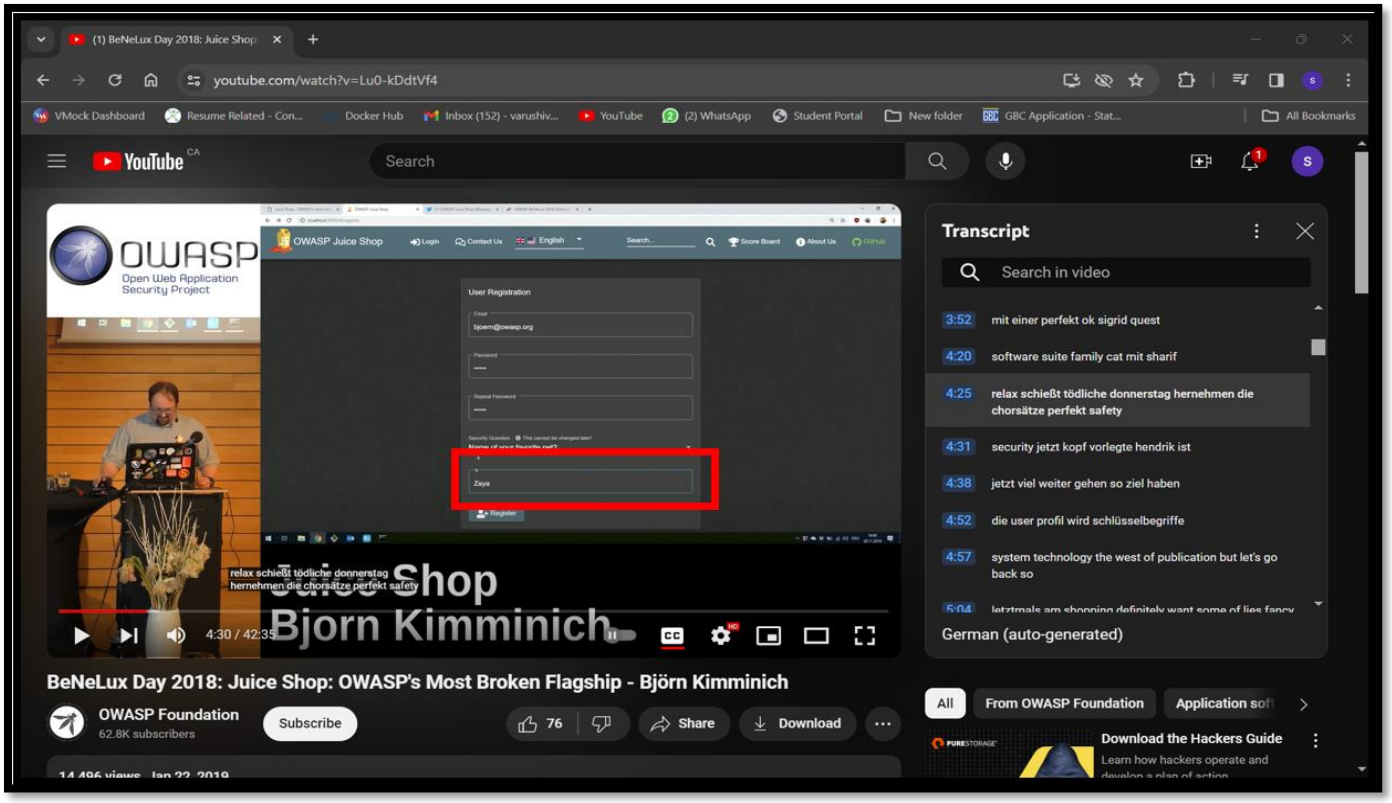
Now to find out password, we just typed Bjorn's favorite pet's name into Google to find out more information on setting up the email address in Björn's Favorite Pet.



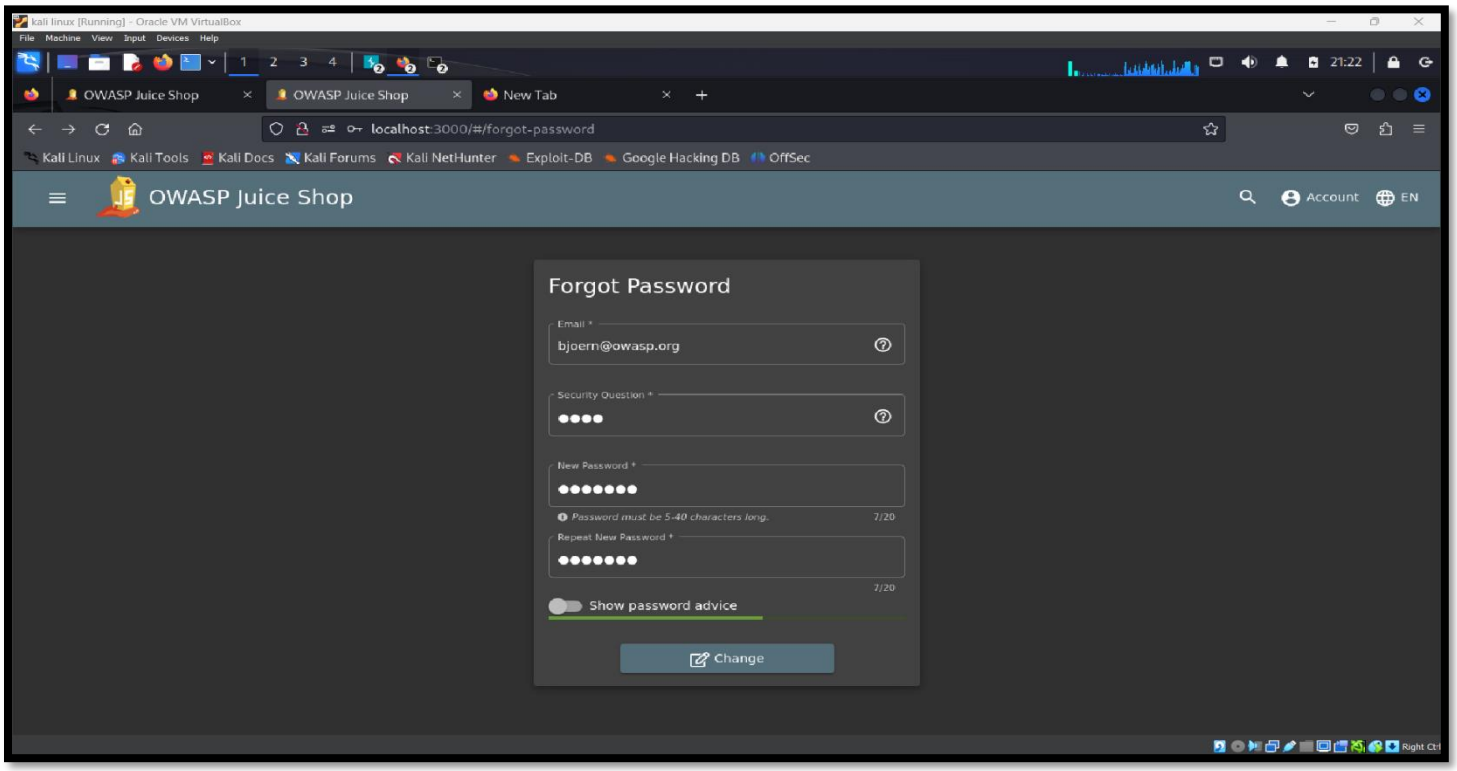
After that, we saw a list of links with relevant information, and we also found a video via a link on YouTube.



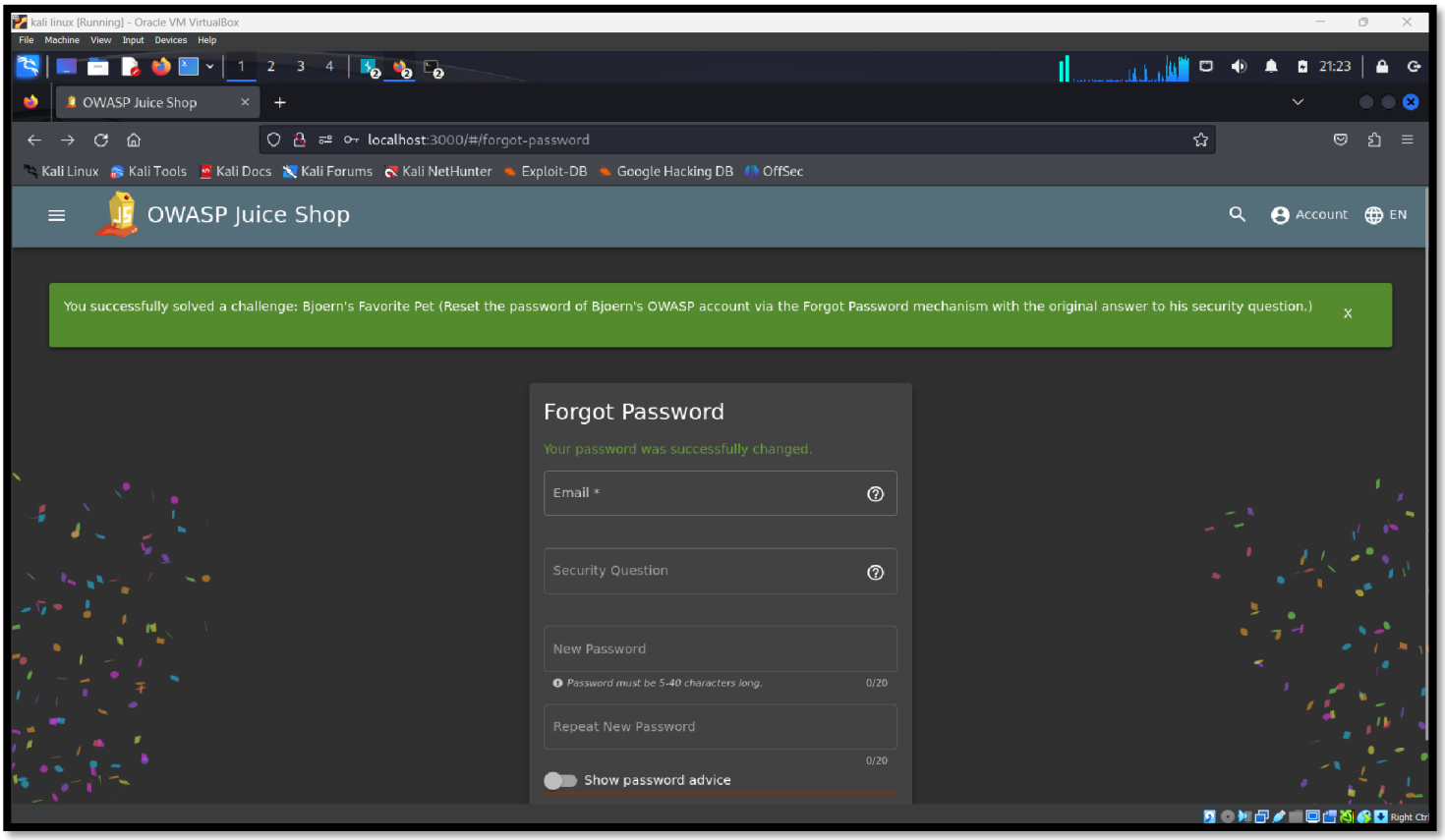
By watching his video with transcript on YouTube, we were saw that one guy was talking about his pet name. ZAYA. This person provides an open description of every detail related to the OWASP Juice Shop in this video. We can learn the email address and the response to the security question by doing this. He entered the email address bjoern@owasp.org and described his favorite pet, Zaya, along with the security question answer.



Thus, we have been able to locate the email address and security answer.



We enter the information in the "Forgot Password" box, enter the password in the "New Password" box, and then confirm the new passord by repeating it. Press the button for change.



Finally, we were able to successfully reset the password and set the new password.

Vulnerabilities' Exploited by Task 3:

- Weak Password Reset Mechanism:
- Inadequate Security Question
- Insufficient Verification of User Input
- Lack of Multi-factor Authentication (MFA)

How Vulnerabilities Can Be Fixed or Addressed:

- Secure Password Reset Mechanism
- Secure Authentication Processes
- Secure Configuration
- User Education
- Regular Security Audits and Testing

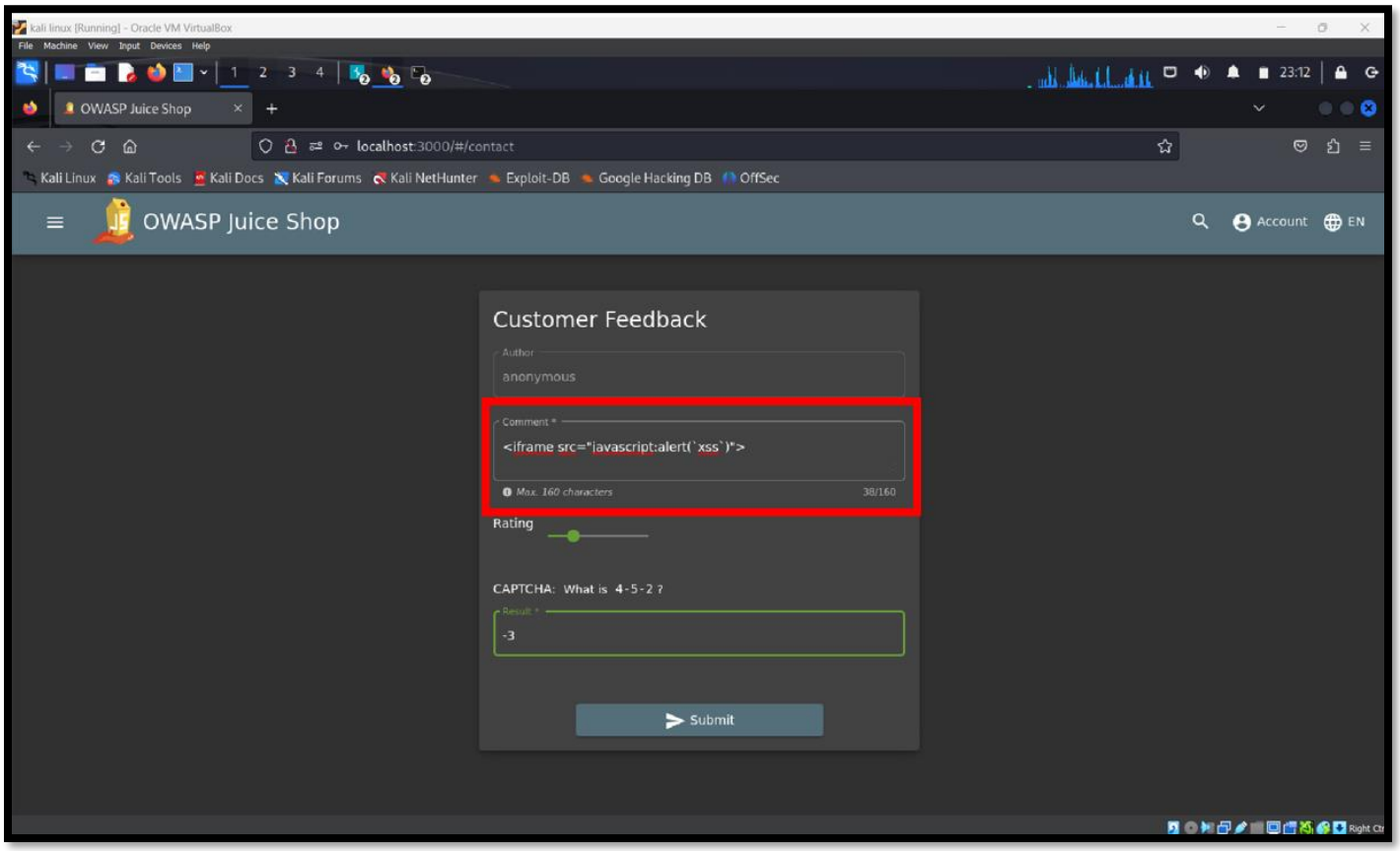
Real-World Scenarios:

- Exploiting weak authentication vulnerabilities, such as those seen in OWASP Juice Shop, can lead to serious security breaches.
- Personal information, such as pet names and birthdates, can be used by attackers to exploit password reset vulnerabilities. Attackers can use unauthorized access to steal sensitive data, perform activities, or increase privileges.
- Sensitive information, such as financial or healthcare data, can lead to financial loss, identity theft, and reputational damage.
- To reduce risks, organizations ought to focus on strong authentication and periodically analyze and improve their level of security.

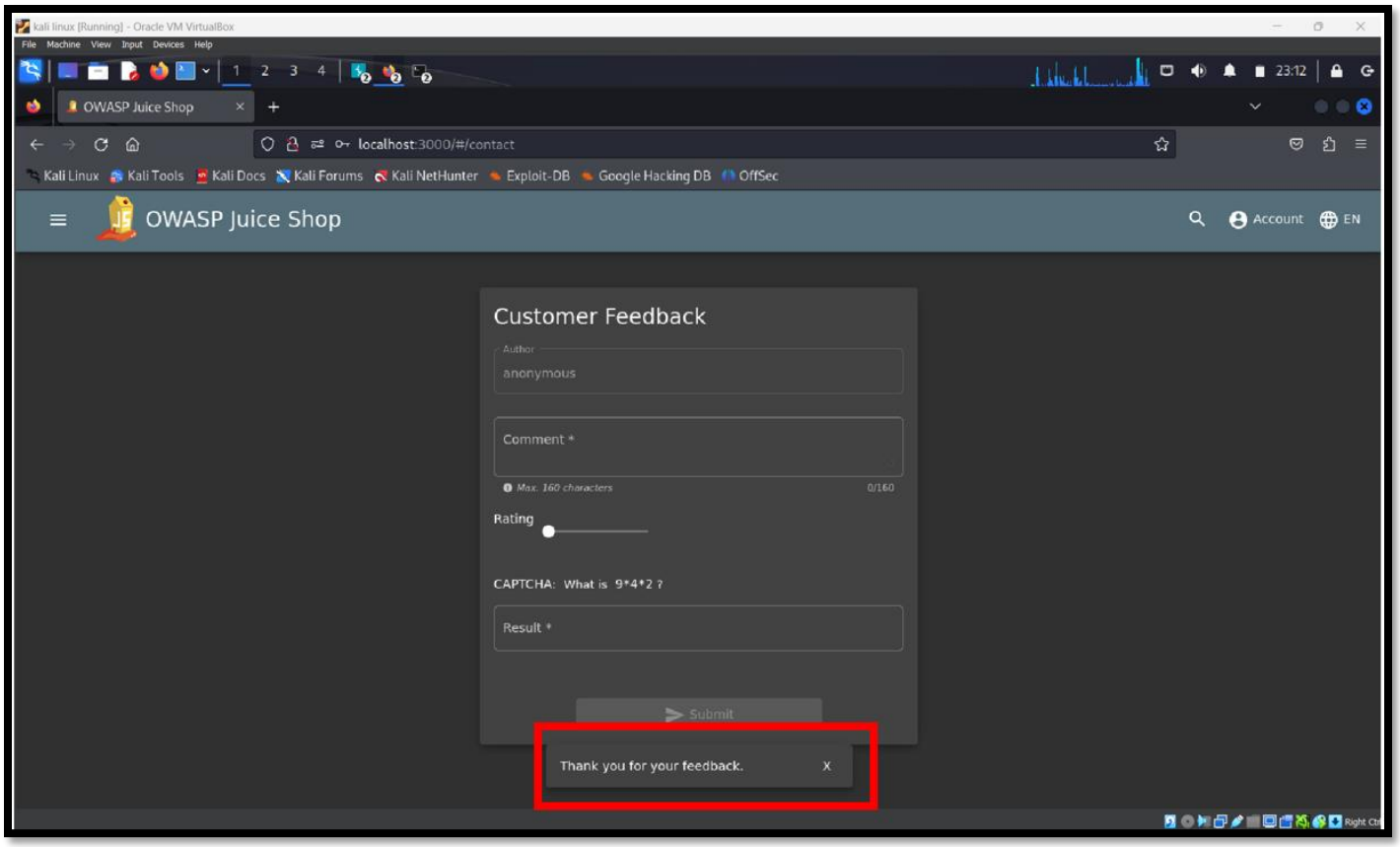
TASK 4

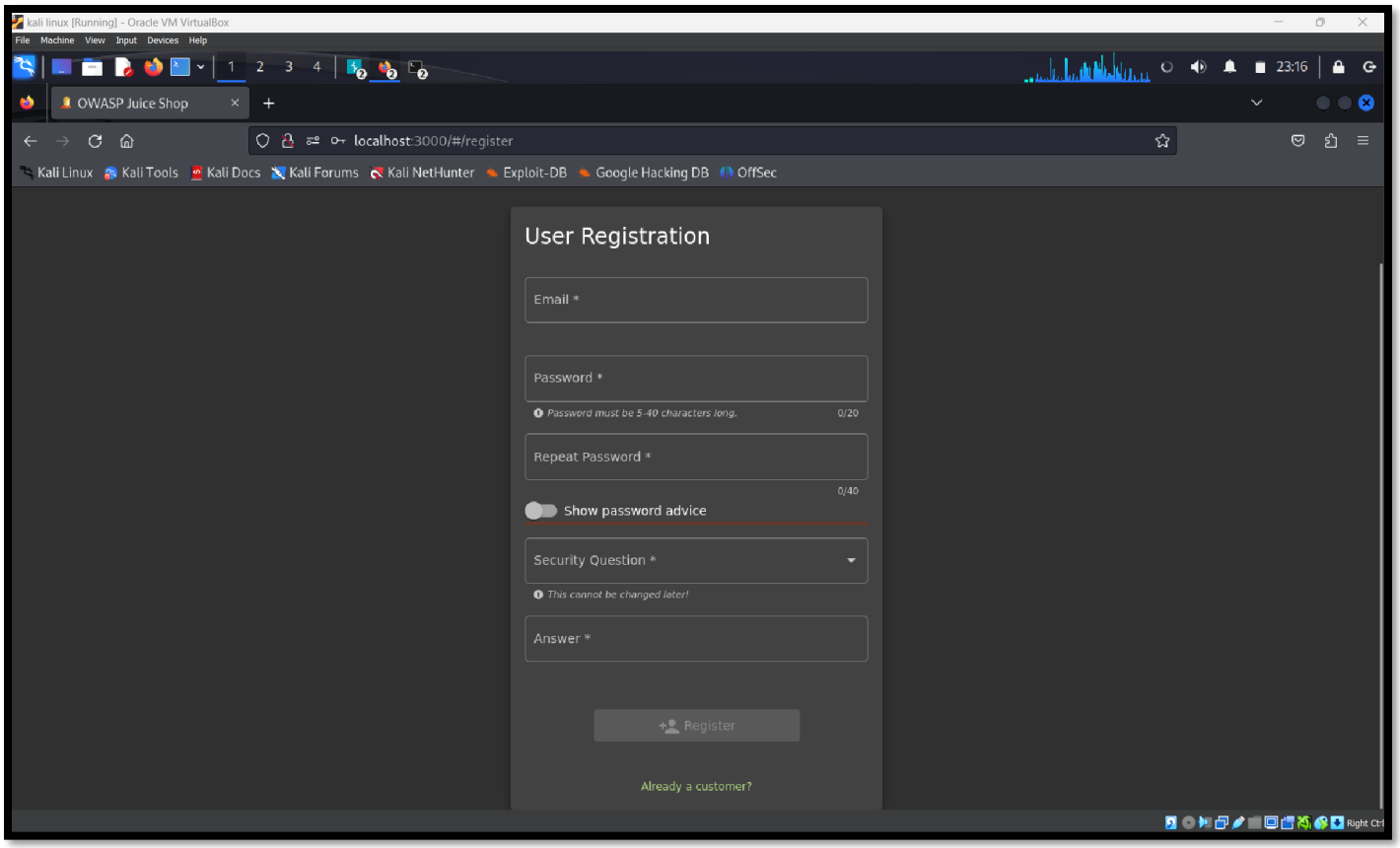
Perform a persisted XSS attack with <iframe src="javascript:alert(`xss)">` bypassing a client-side security mechanism.

In this challenge, we went around the client-side security mechanism. There were two places in the Juice Shop application where we could enter information to check for client-side security controls. Thus, we attempted to enter the XSS JavaScript payload in the customer's feedback.

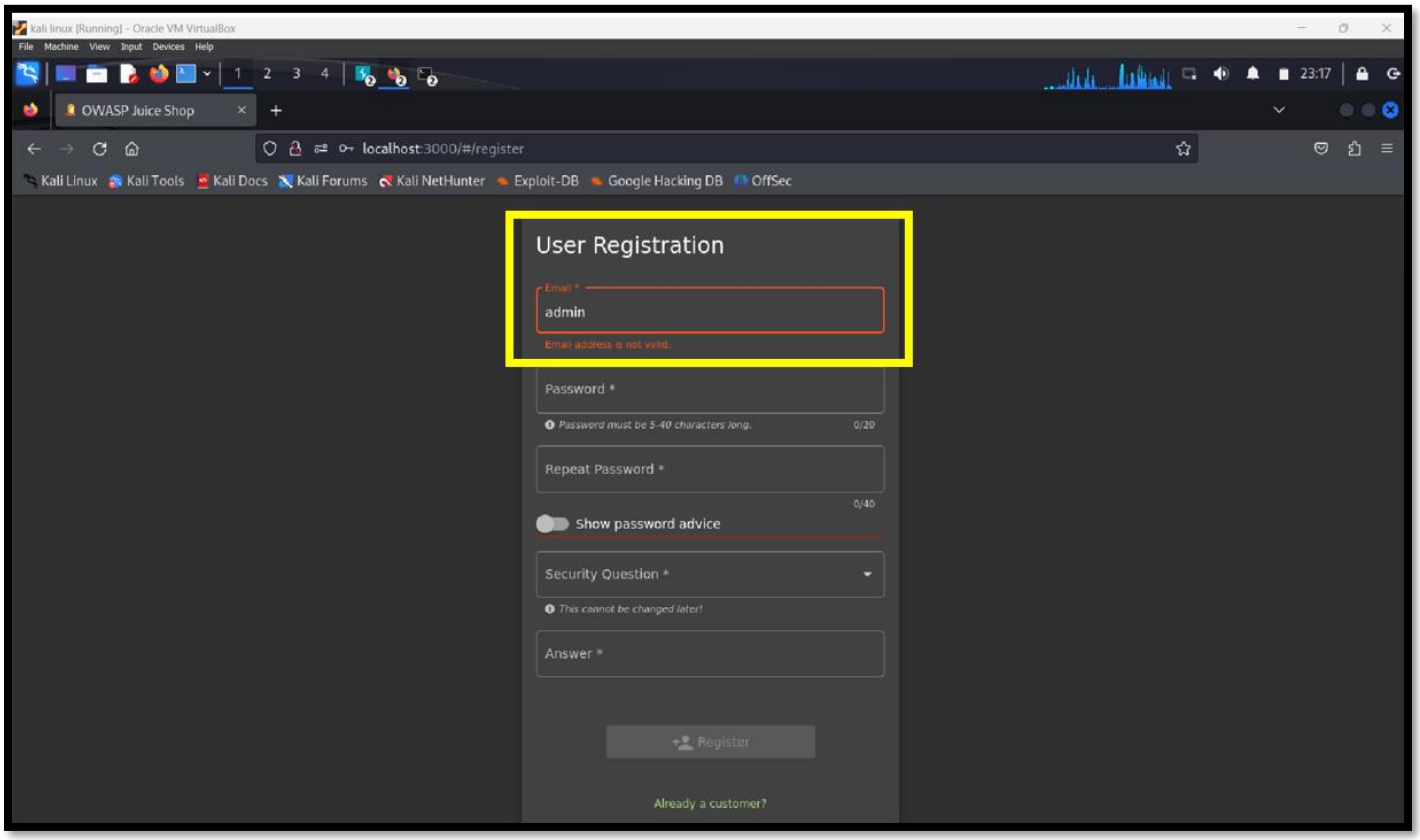


It accepted the receiving of the payload and reported response success. Thus, it was found that there was no client-side security control on the customer feedback form.

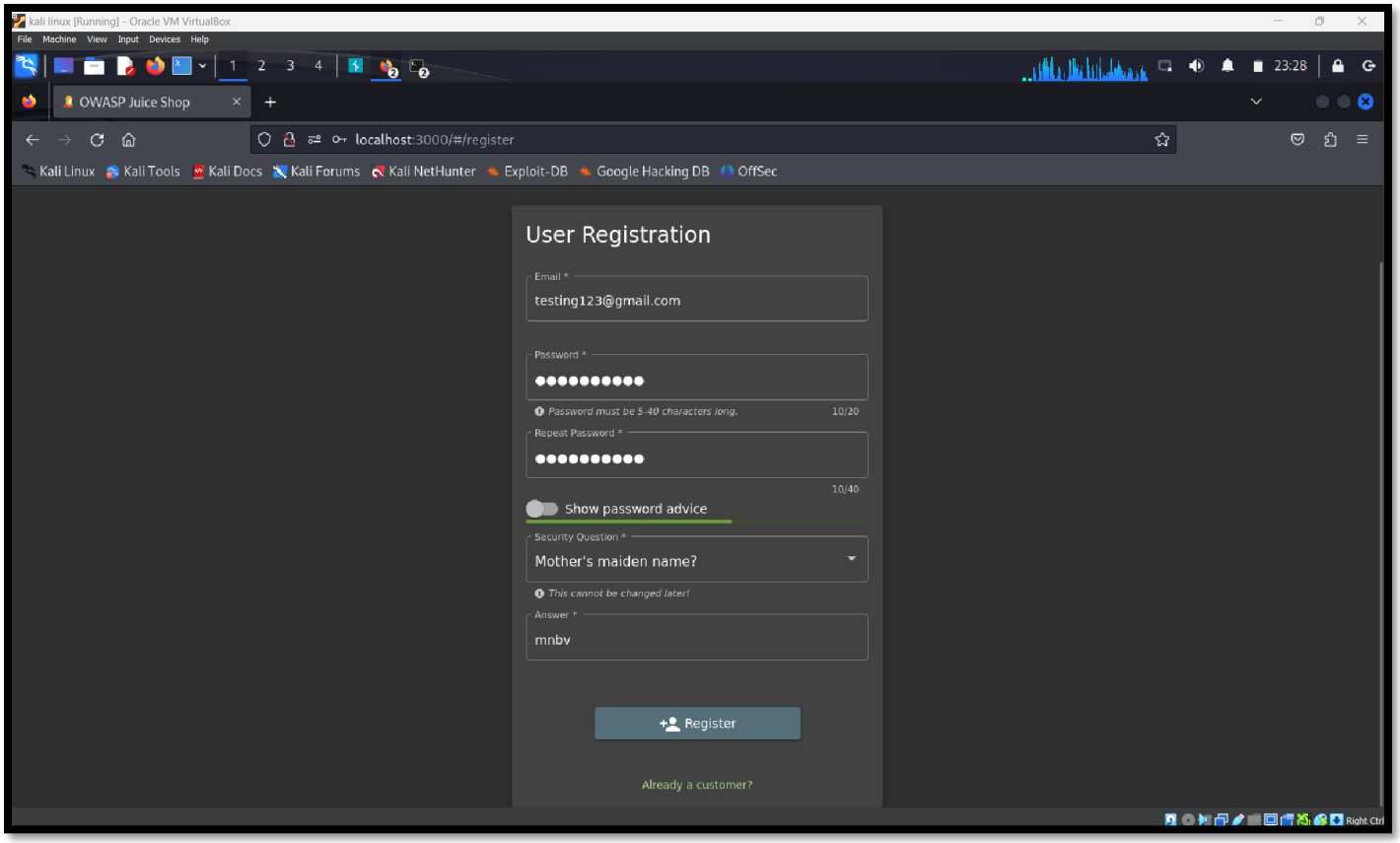




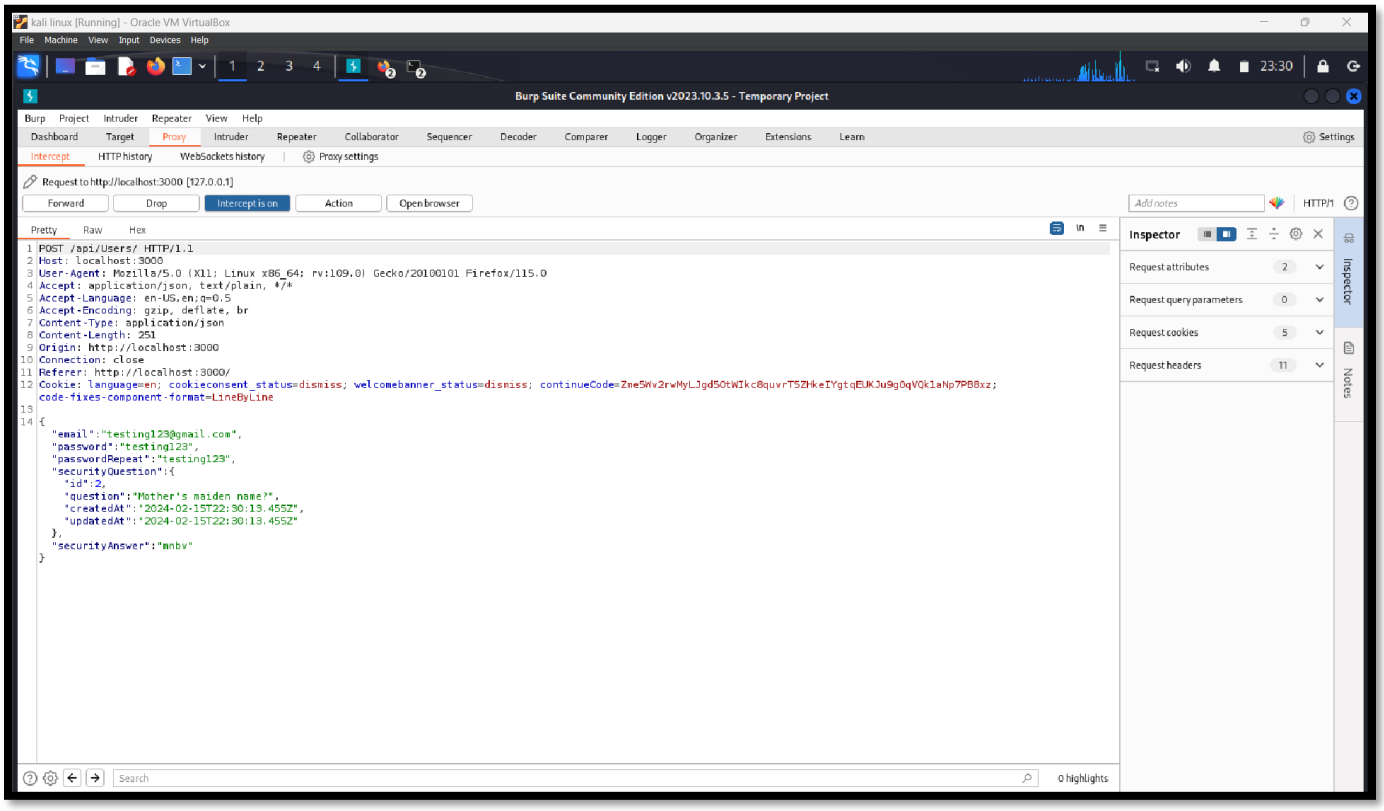
We looked through the registration page to see what we could find. When we wrote "admin" as the email field, it showed that it accepted only email addresses. Here was where we discovered the client-side security control, which specified that only valid email addresses were accepted for registration.



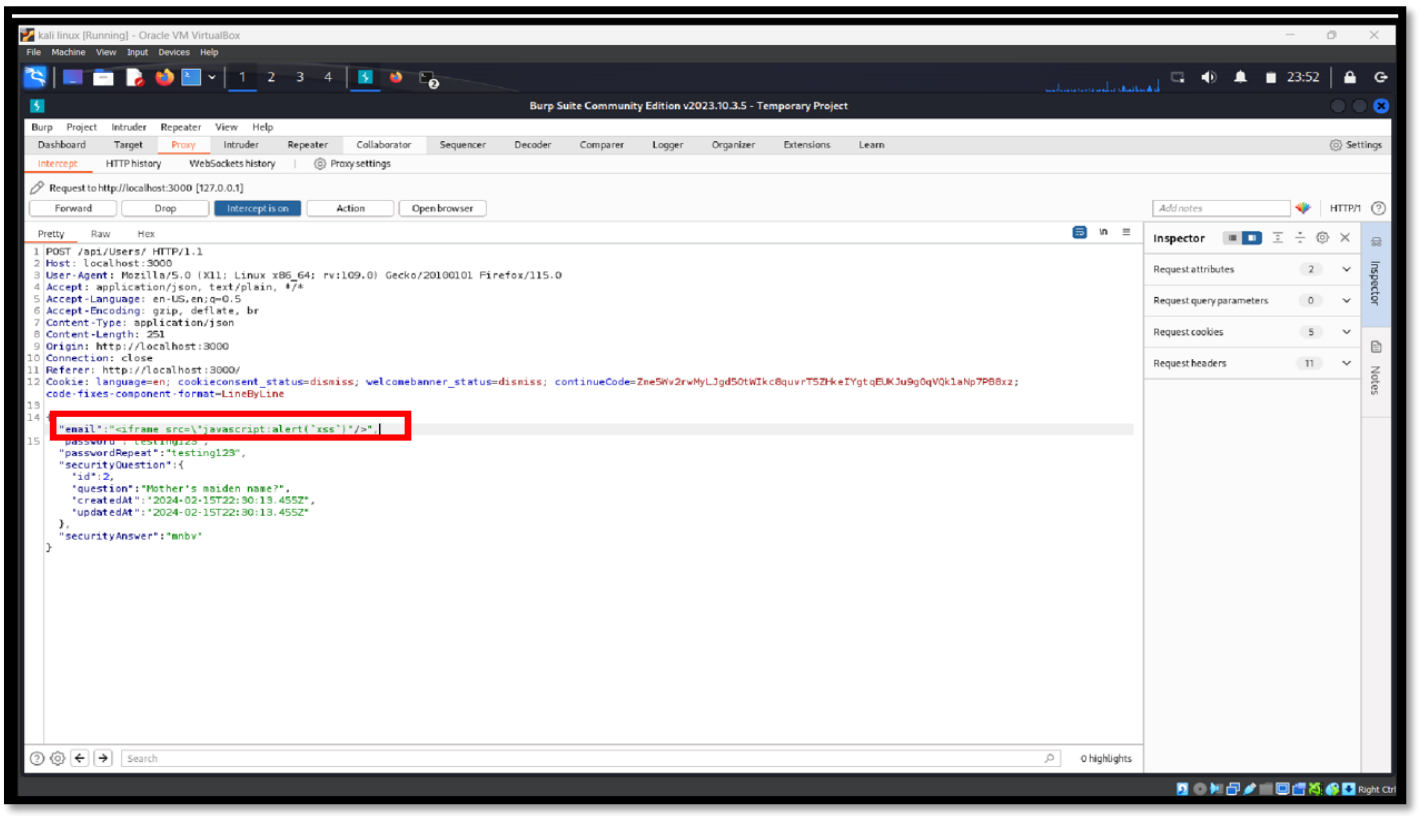
To complete this task, we needed to register with the Juice Shop and submit a request to the server.



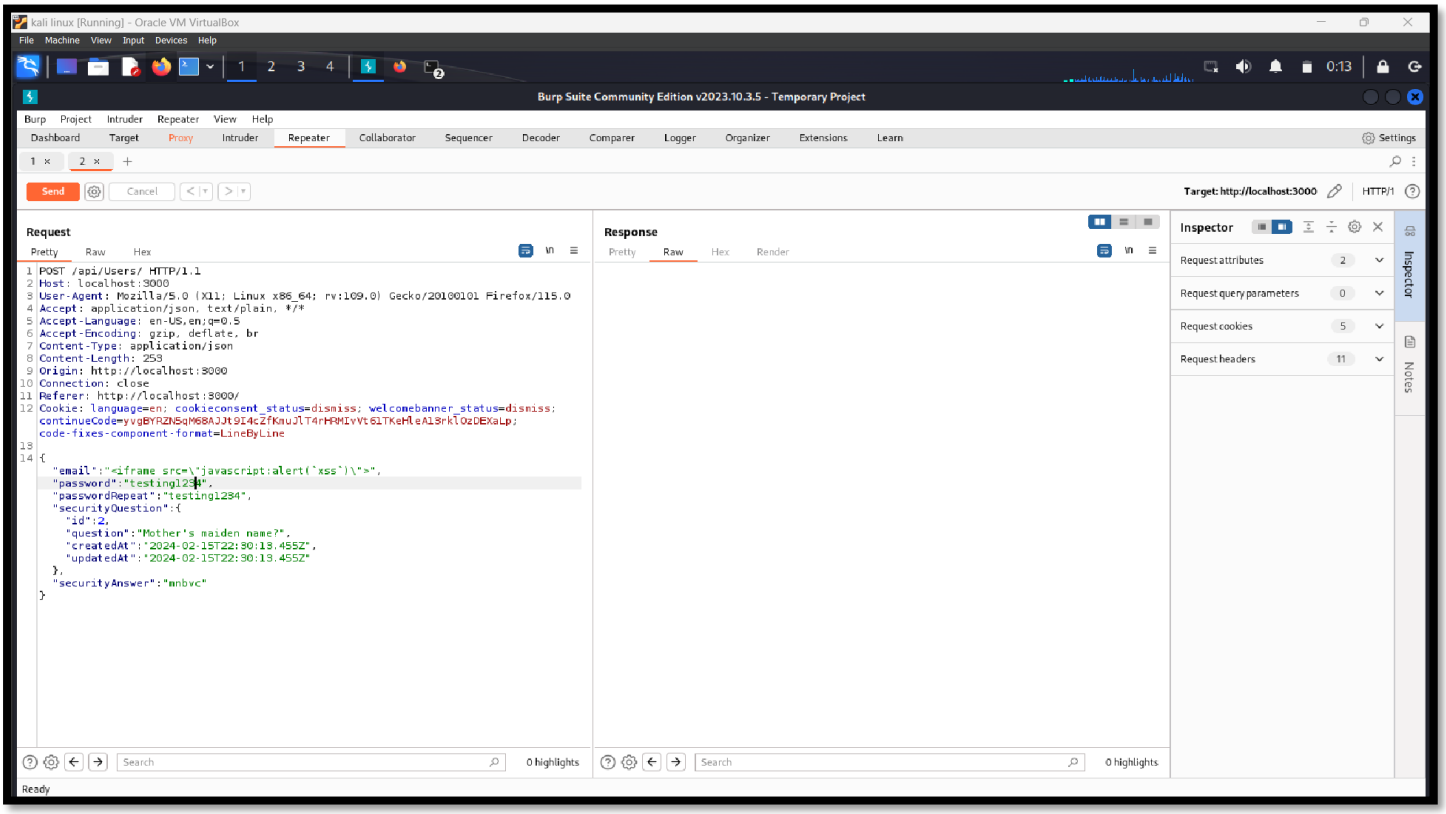
We used an application called Burp Suite to monitor the request; this tool truly assisted in modifying the request to bypass the client-side security control.



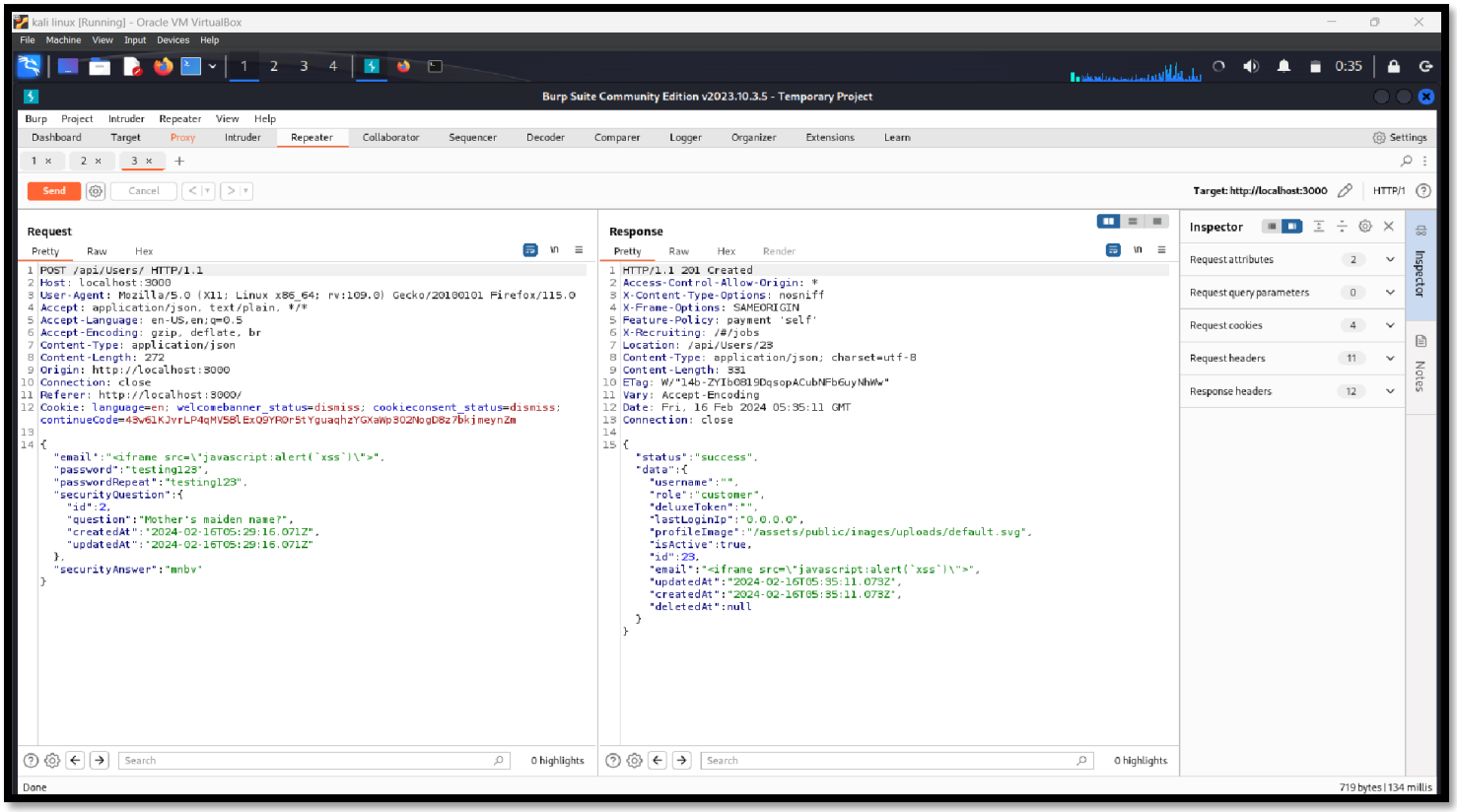
This was the request that the website created, and as we could see from the Burp Suite, it only allowed us to enter an email address rather than a random string of characters. We modified the request with the Burp Suite to try to bypass the client-side security.

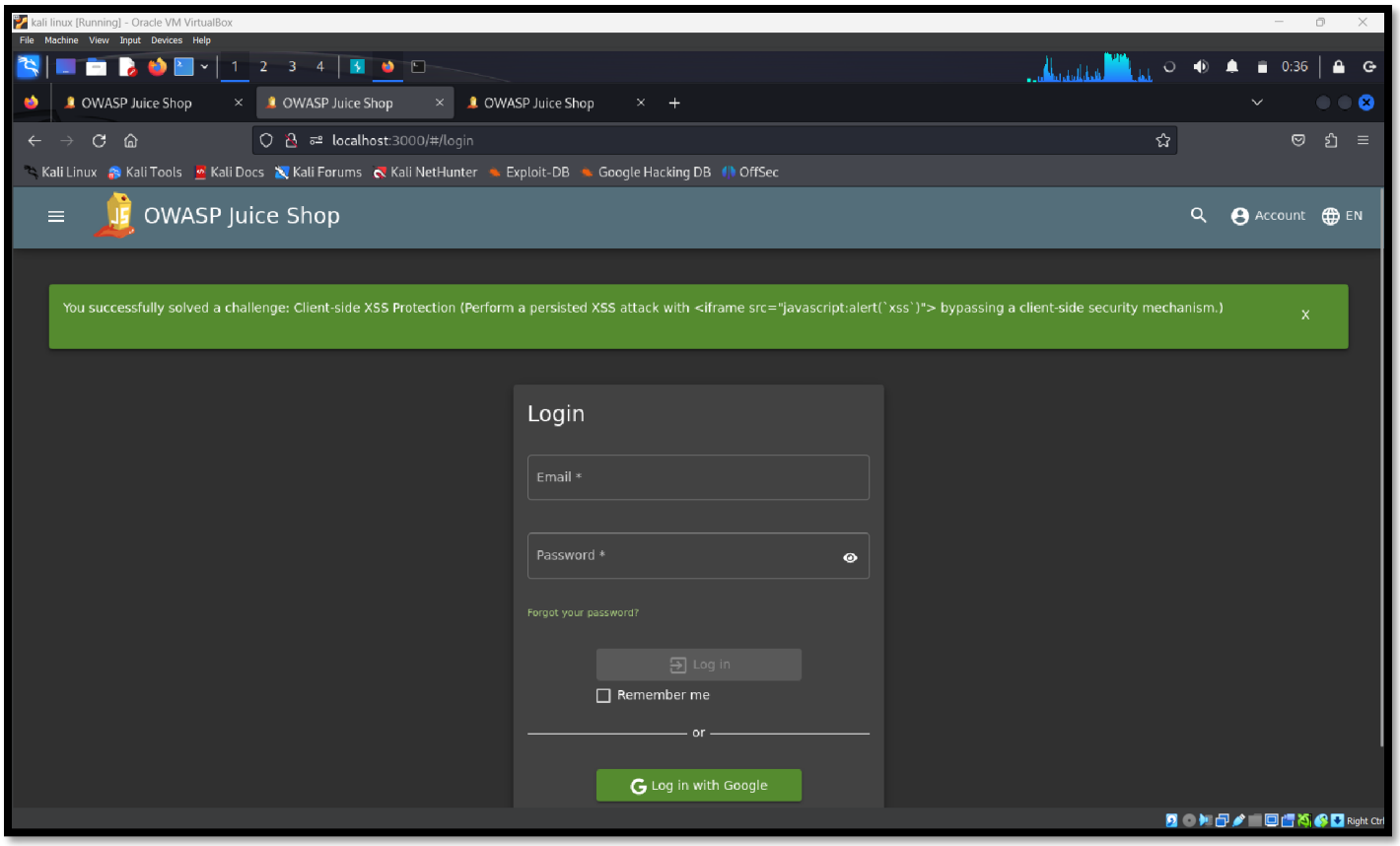


Instead of using our registration email address, we injected the XSS script. We also noticed that it was not recognizing our XSS payload, so we had to use a backslash to escape the apostrophe. Then we started Burp Suite to send the injected XSS script to repeater.

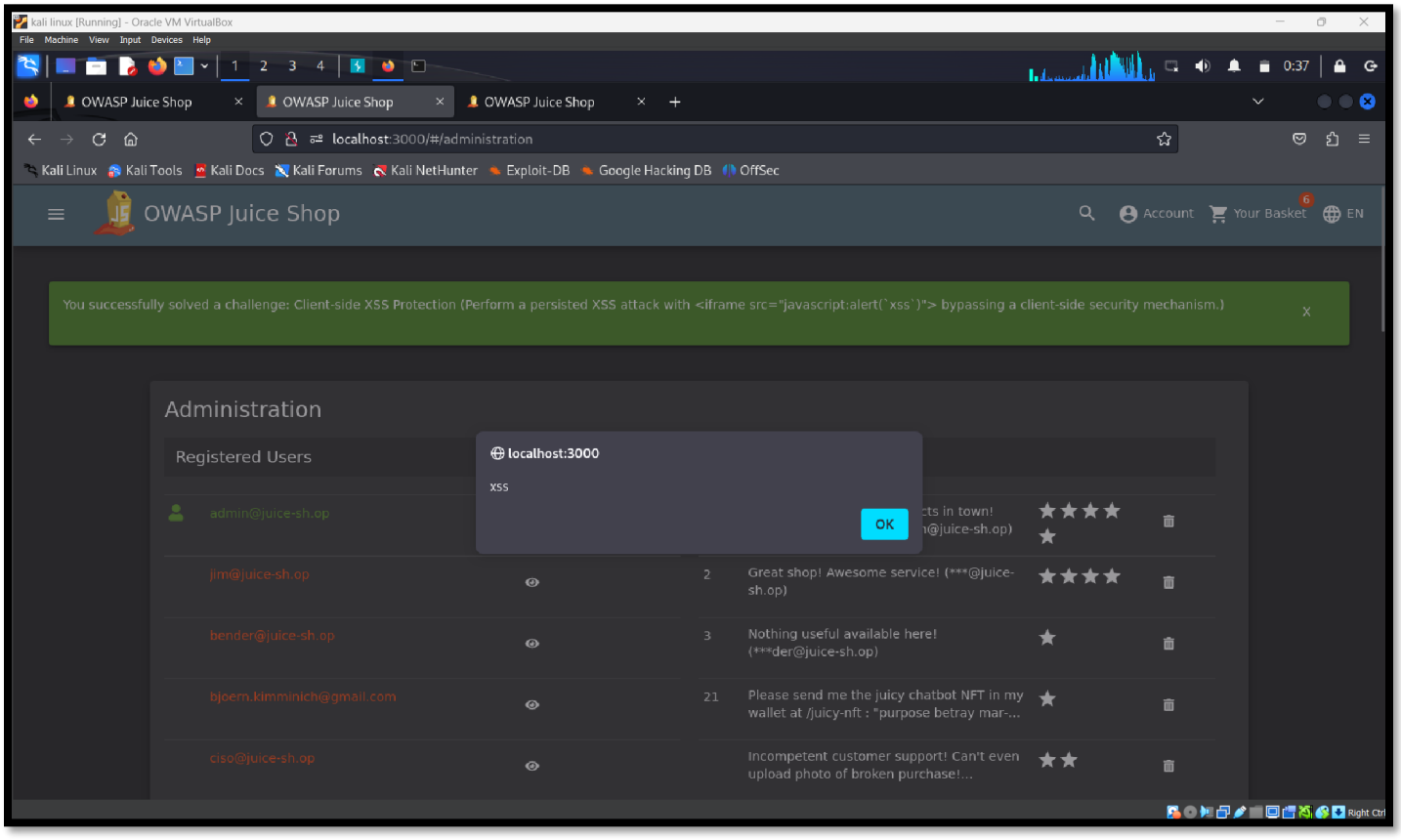


Here, we got 200 as a success response, which meant our injection was successfully implemented.





We entered the admin page using the admin credentials and, using SQL injection in task 2, we were able to see that the user had registered, and that localhost was reporting "XSS." This indicated that the client-side security measure had been successfully bypassed.



Vulnerabilities' Exploited by Task 4:

- Manipulation of Requests Sent to the Client-Server
- Malicious Code Inserted in Client-Server Data
- Bypassed Client-Side Security Mechanism
- Unauthorized Access using Credentials.
- Lack of Validation for User Data
- Delivery of Malware to Users
- Potential for Further Attacks

How Vulnerabilities Can Be Fixed or Addressed:

- Input Validation
- Avoidance of XSS in Design
- Automated Vulnerability Scanning
- Implementation of Content Security Policy (CSP)
- Sanitized Inputs and Validation
- Deployment of Web Application Firewall (WAF)

Real-World Scenarios:

- Numerous web applications, including social media sites, e-commerce platforms, and banking portals, are vulnerable to XSS vulnerabilities.
- Attackers can deploy malicious scripts through vulnerabilities such as persistent XSS, which are present in product review sections of retail websites.
- These scripts could either obtain session cookies or redirect users to phishing websites when they access affected pages.
- Insufficient execution of client-side security protocols, like email verification and input validation, may expose the application and its users to cross-site scripting (XSS) attacks.

References

1. Swinhoe, M. H. a. D. (2023, November 8). The 15 biggest data breaches of the 21st century. CSO Online.https://www.csoonline.com/article/534628/the-biggest-data-breaches-of-the-21st-century.html
2. YouTube Transcript - read YouTube videos. (n.d.). https://youtubetranscript.com/
3. Dropout. (2014, October 27). Rapper who is very concerned with password security [Video]. YouTube. https://www.youtube.com/watch?v=v59CX2DiX0Y
4. MC SafeSearch - Protect ya Passwordz (2014). (2014, October 27). IMVDb. https://imvdb.com/video/mc-safesearch/protect-ya-passwordz

Video Recording Link:

https://youtu.be/LjBvedyGKo0