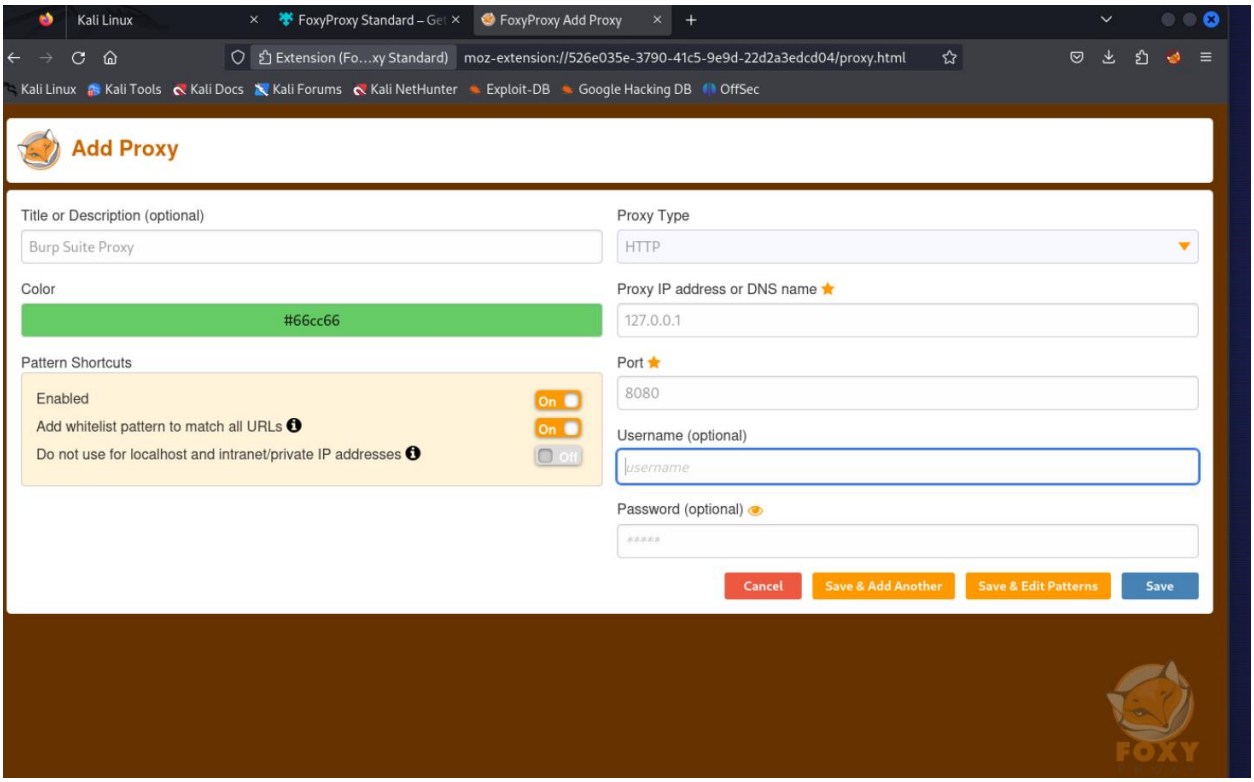
Chapter 15

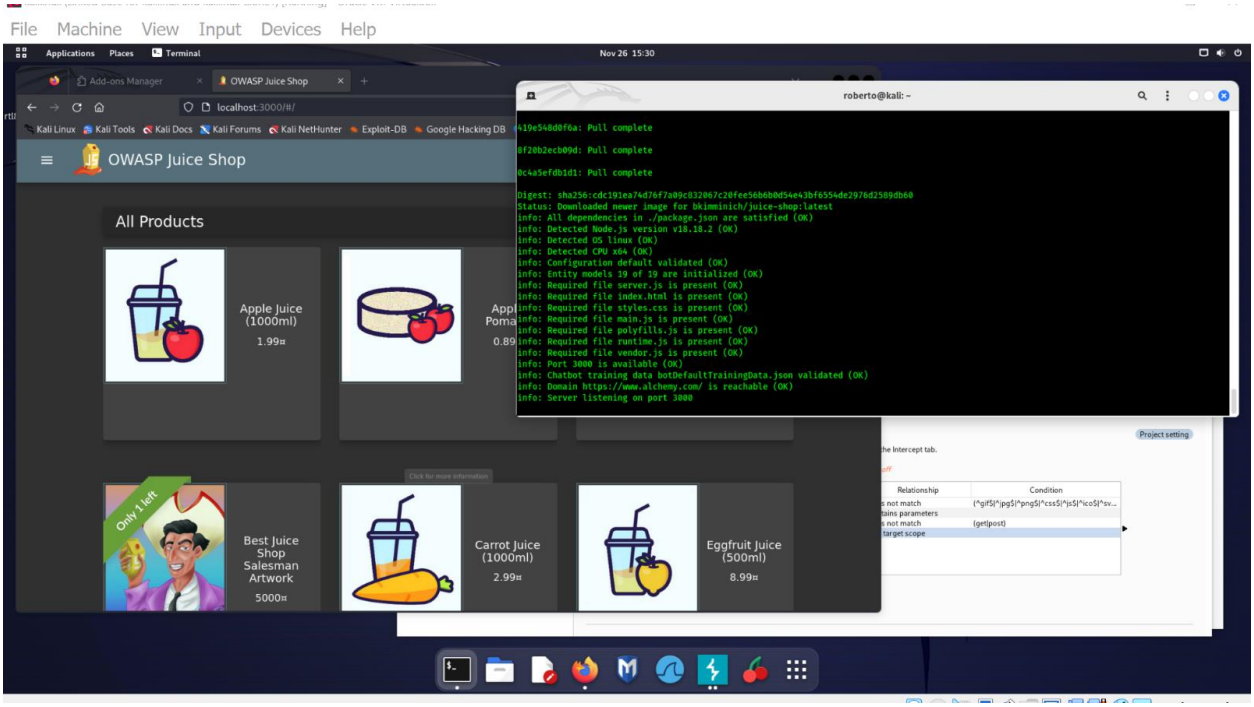
1. Document vulnerabilities that you are able to successfully exploit on the server. Describe in detail what you did and what level of access you were able to obtain. If you obtain a user account with limited privileges, document whether you were able to escalate the privileges to root. Document each exploit that you are able to successfully launch.

- The purpose of this lab is to gain knowledge about the significance and necessity of carrying out web application penetration testing. I also learned how penetration testers and other cybersecurity experts find security flaws in online applications with the aid of the OWASP Top 10. Additionally, using the OWASP, I learned how to execute vulnerability detection and exploitation on a web application. Doing this, I will incorporate tools such as Burp Suite and OWASP Juice Shop.
- **Tools used**:
- Foxy Proxy Is an add-on for your web browser that lets you set up several profiles for different web proxy configurations, making it easy to switch between proxies without having to manually adjust them.
- Burp Suite Is a proxy-based tool that enables penetration testing professionals to change request messages from the client side by intercepting communication between the attacker's web browser and the online application.
- Kali Linux
- **Critical security risks and vulnerabilities:**
- Broken access control
- Cryptographic failures
- Injection
- Insecure design
- Security misconfiguration
- Vulnerable and outdated components
- Identification and authentication failures
- Software and data integrity failures
- Security logging and monitoring failures
- Server-side request forgery
- As a result, I learned the principles of web applications and the way that HTTP functions between a web browser and a web application during this chapter. Additionally, the following exercises possessed me with the ability to model different kinds of online application cyberattacks on susceptible apps in order to identify and take advantage of security holes in a target.

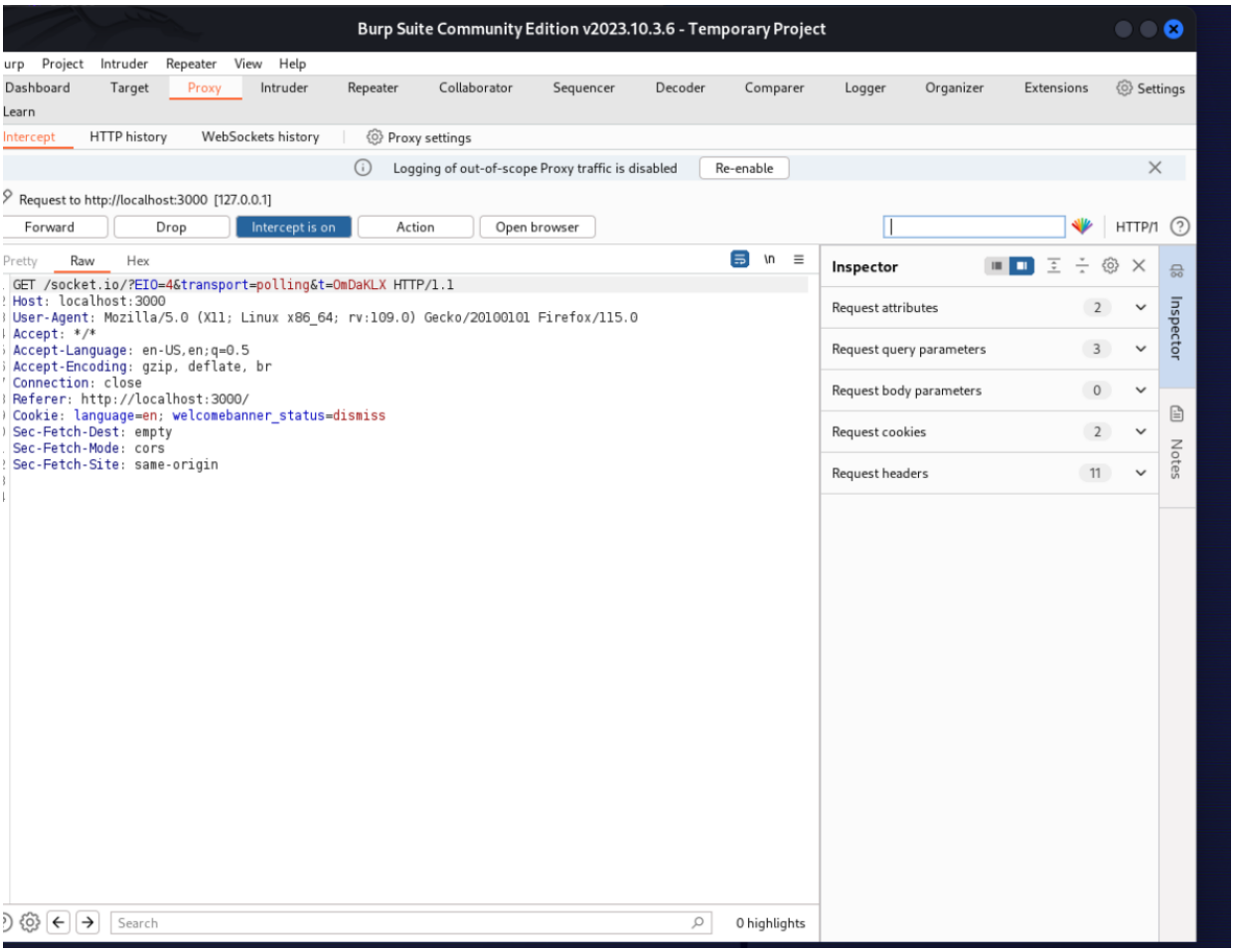
- Getting started with FoxyProxy and Burp Suite
- To initiate this lab, I went through the entire process to set up Foxy Proxy. Setting up and verifying all configurations are checked off is crucial in this lab, as a mistake will result in issues.



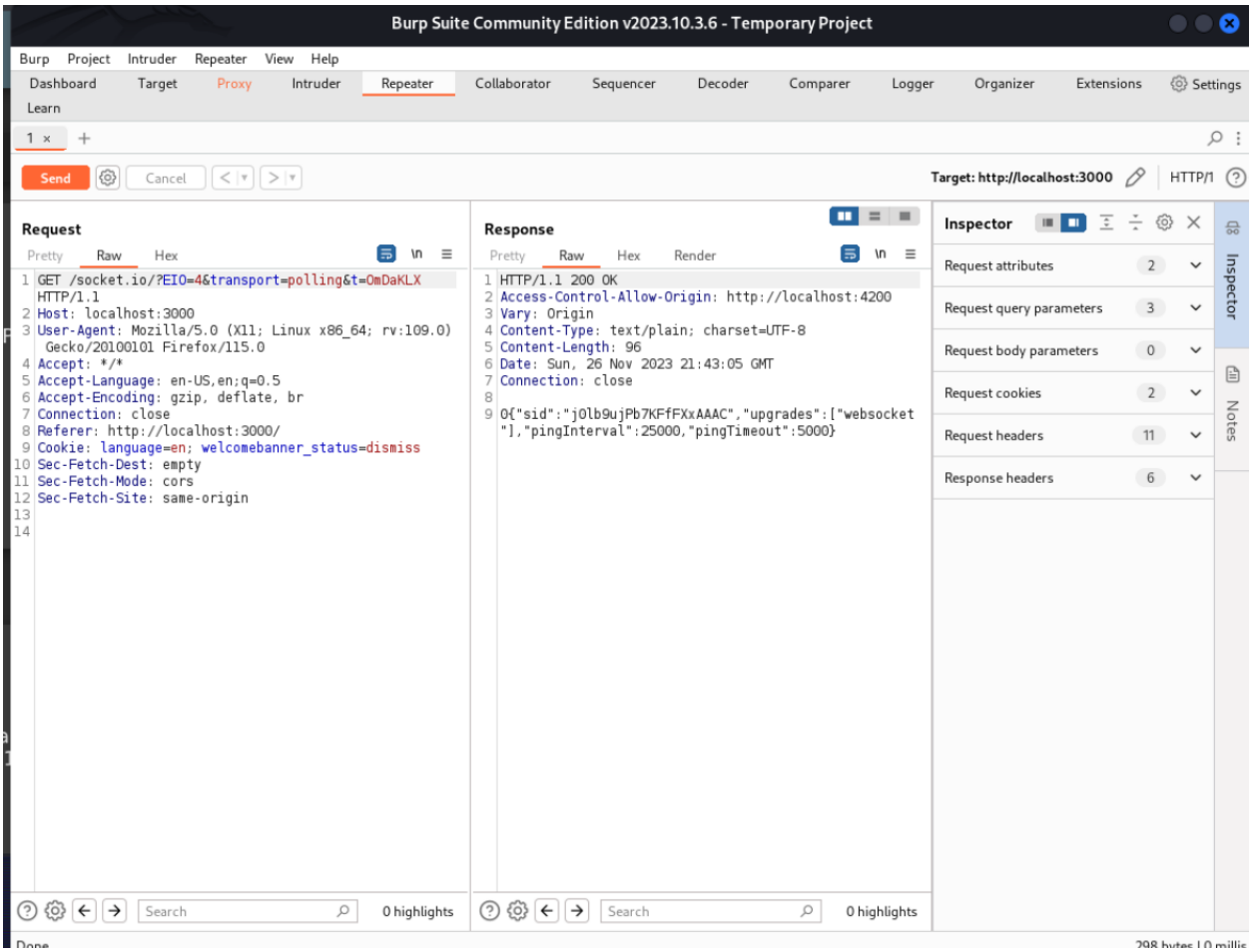
- Next, I had to initiate the Burp Suite application. The Kali Linux one was outdated so the features were limited. To fix the issue, I went through and updated the application successfully.
- After going through the steps from the book, I successfully set up the Burp Suite application and booted up the OWASP Juice Box application with no problem.



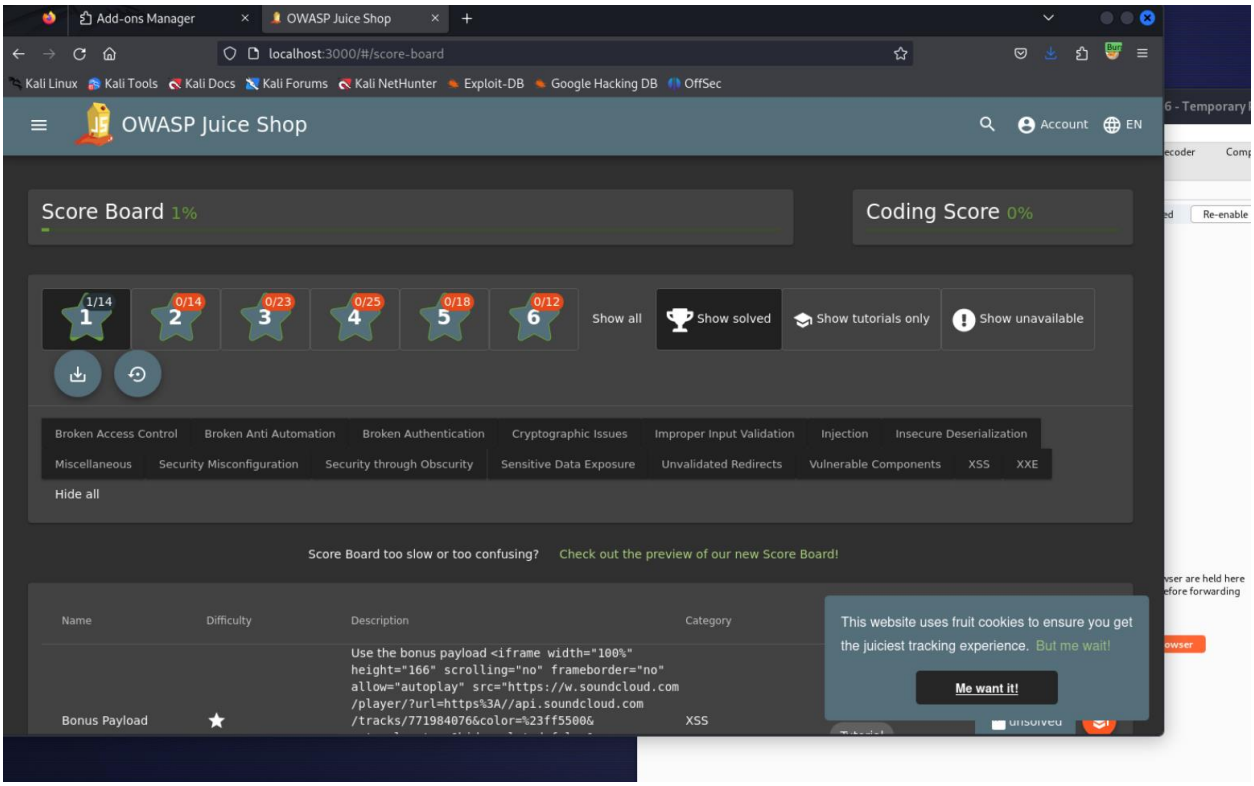
- This section helped me get familiar with the features within the Burp Suite application.
- By turning on the intercept, Burp Suite will be able to intercept the web request as it leaves the browser and heads straight to the web application.
- By completing this, Burp Suite recorded a message containing an HTTP GET request.



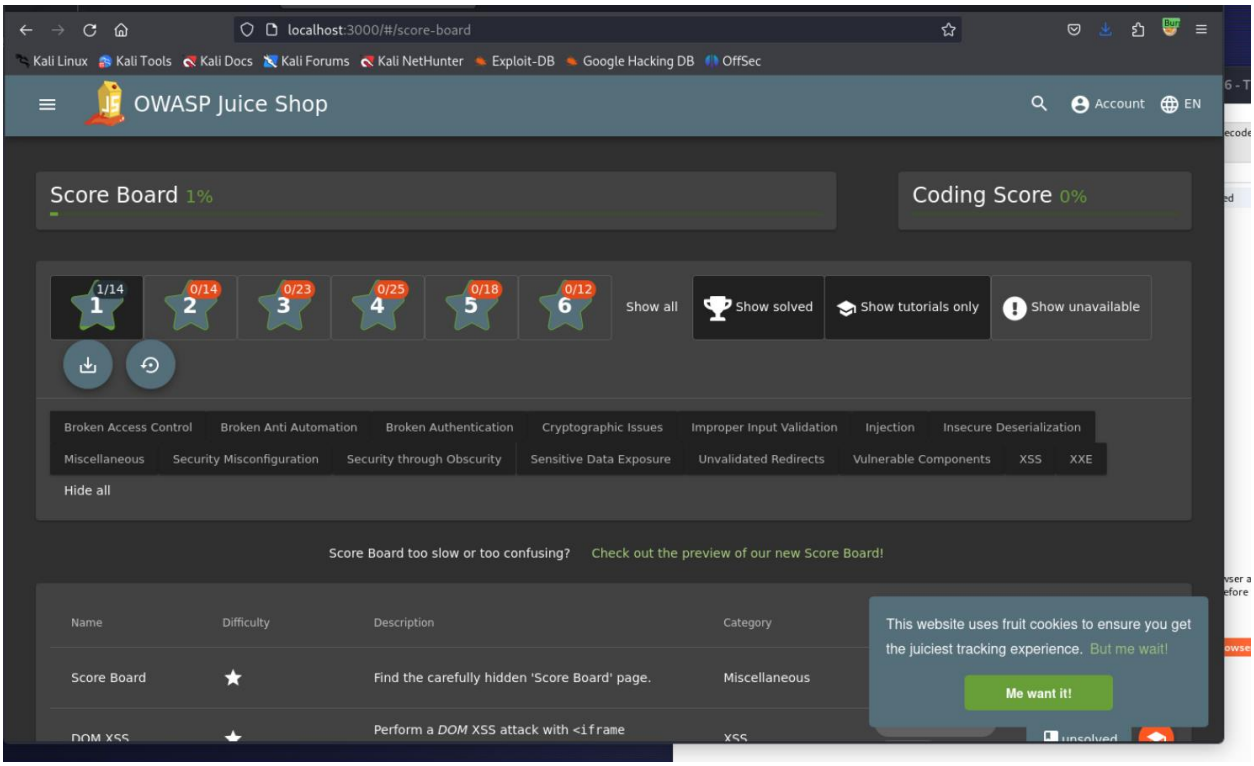
- Next, I forwarded the HTTP GET message to the repeater, which is another feature in the Burp Suite application.
- Burp Suite's Repeater functionality lets penetration testers edit a web request message before delivering it to the target web application.
- After sending it to the repeater, I click send, which will forward it to the response section next to the request. By doing this, I am able to modify the parameters in the request message and even hijack the web application.



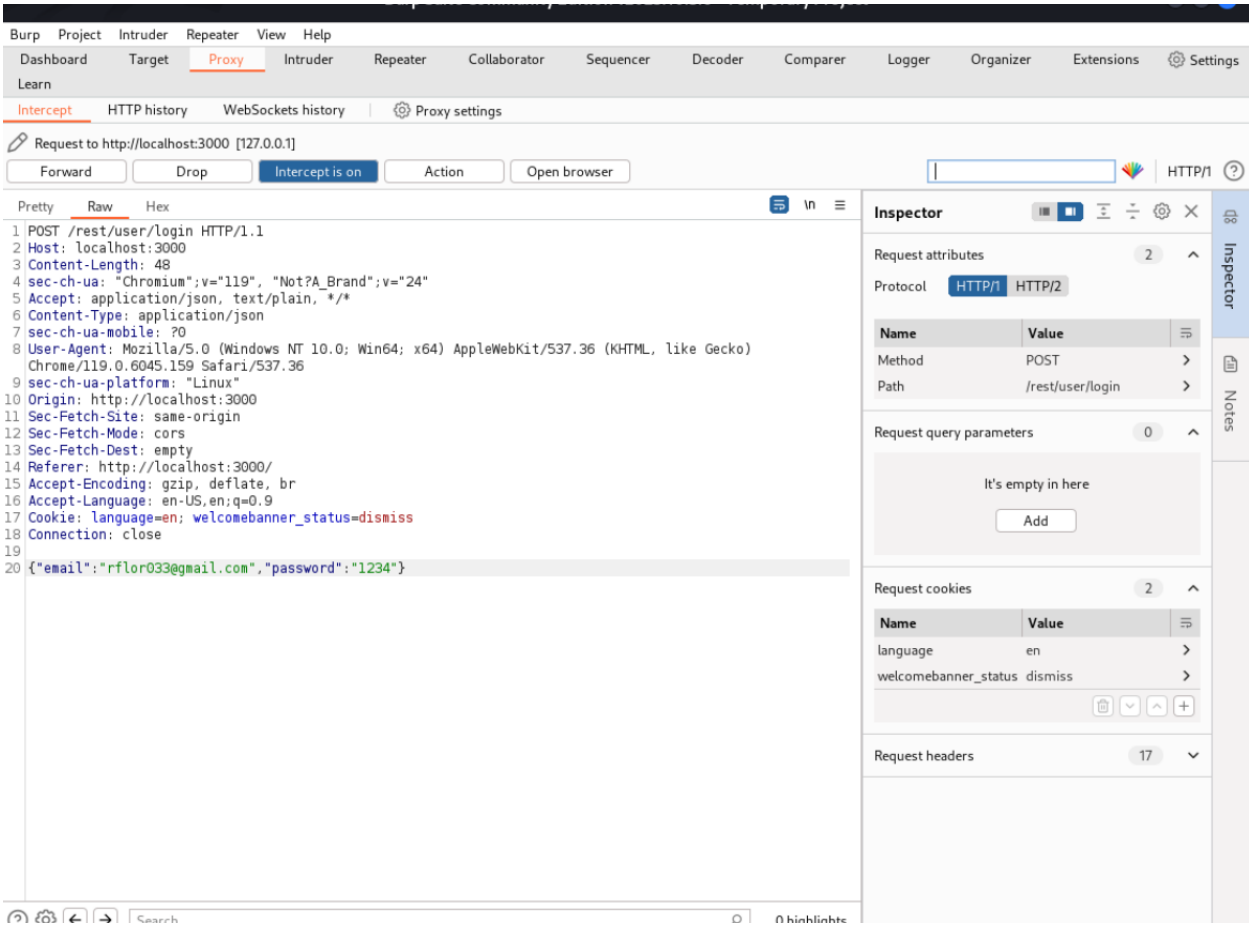
- After completing those tasks, I referred to the OWASP score board and was able to see that I had successfully completed the challenge.
- After completing this section, I now understand the fundamentals of using Burp Suite and how to set up Foxy Proxy as a web browser's proxy switcher.



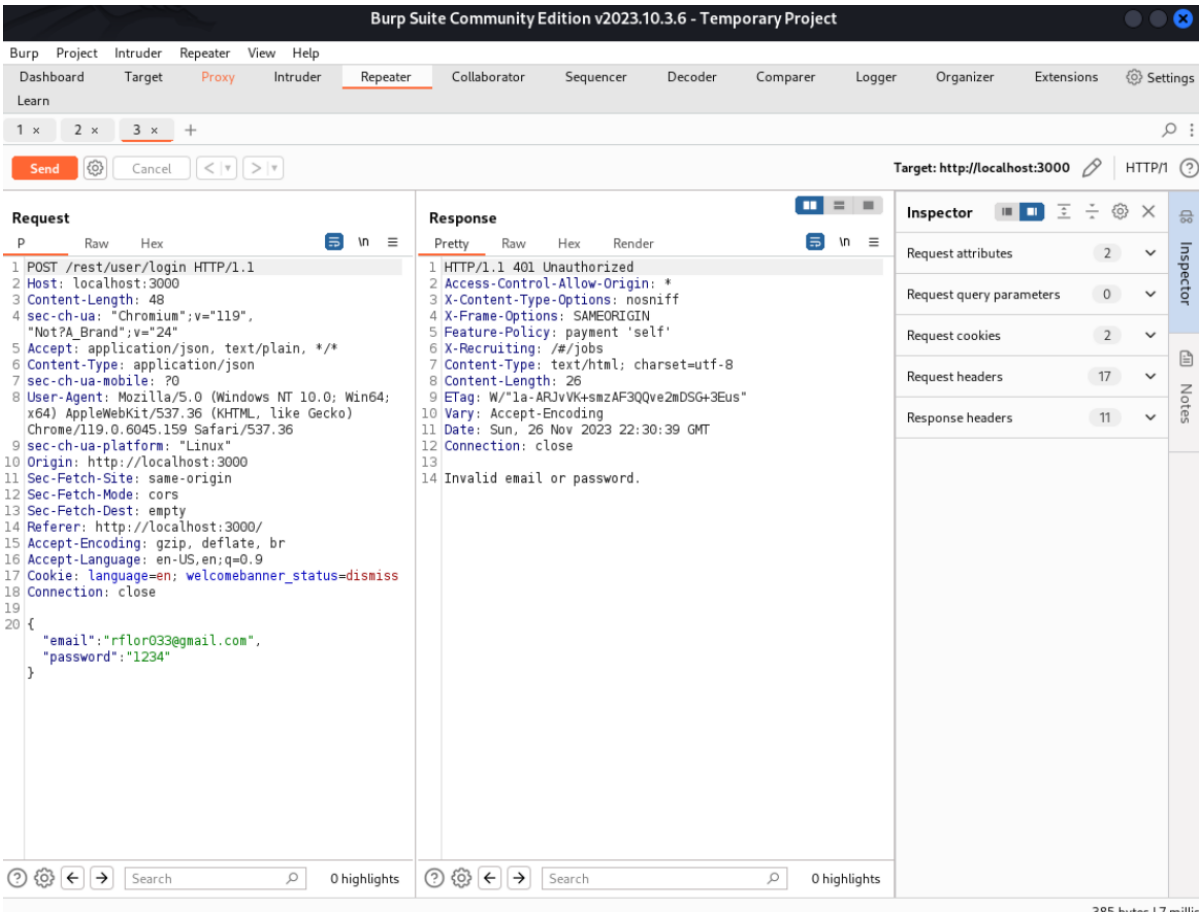
- Performing a SQL injection attack
- In this section, using Burp Suite on Kali Linux, I discovered how to leverage SQL injection to obtain administrator access to a web application, as in OWASP Juice Shop.
- I ensured that my Kali Linux is powered on and verified that the OWASP Juice Shop Docker instance is running.



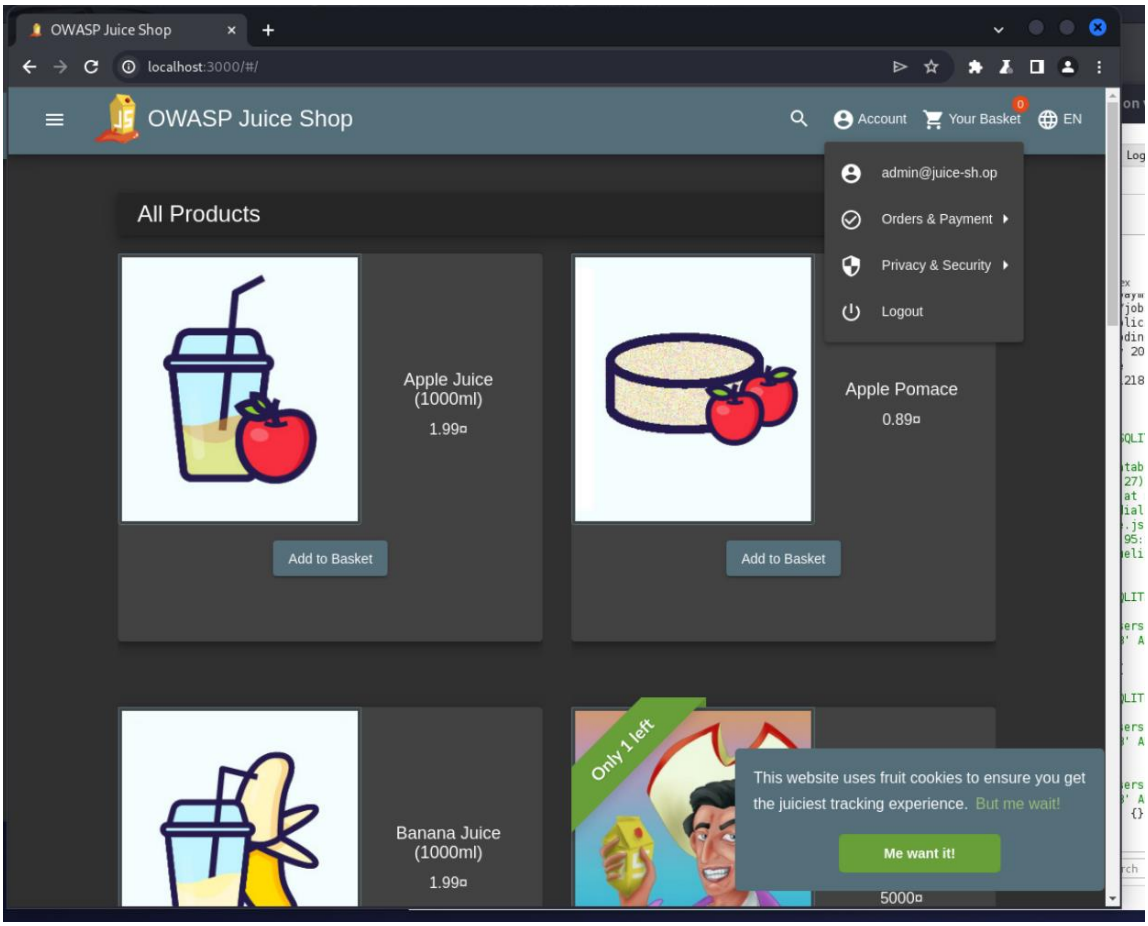
- Prior to commencing with this section, I made sure FoxyProxy was configured to use the Burp Proxy configurations, launch Burp Suite, and confirm that intercept was enabled.
- I went to the log-in section of the OWASP Juice Shop web application and entered a random email and password.
- In the screenshot below, I refer back to the HTTP POST message and see the random email address and password I entered.

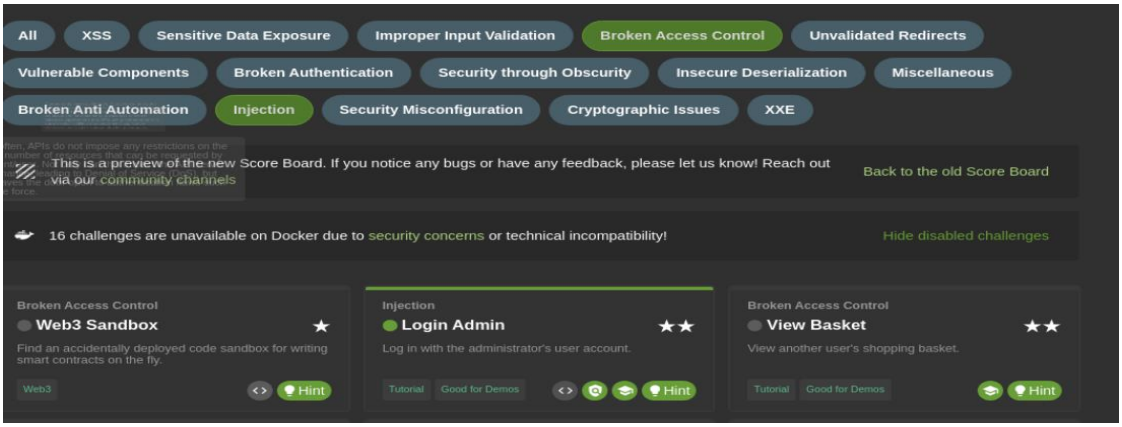


- Additionally, in the response section, I can see the unauthorized status and a message indicating the email or password is invalid.
- The next part will involve me committing a SQL injection attack. I injected the email as admin@email.com into the request tab, which will prompt this login information to the web application.

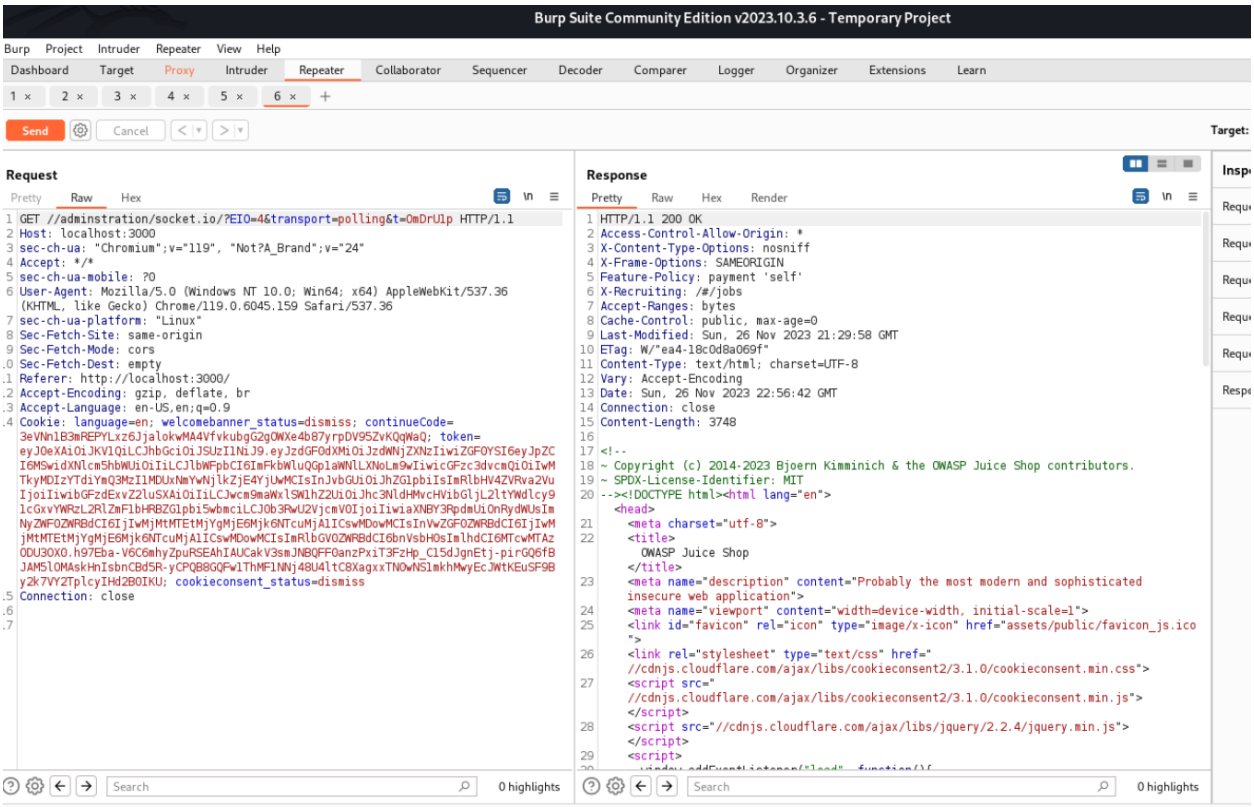


- The web application is vulnerable to a SQL injection attack
- Upon successfully completing the tasks, I know that the web application is vulnerable to a SQL injection attack.
- Additionally, I am able to successfully gain access to the administrator's user account on the vulnerable web application.

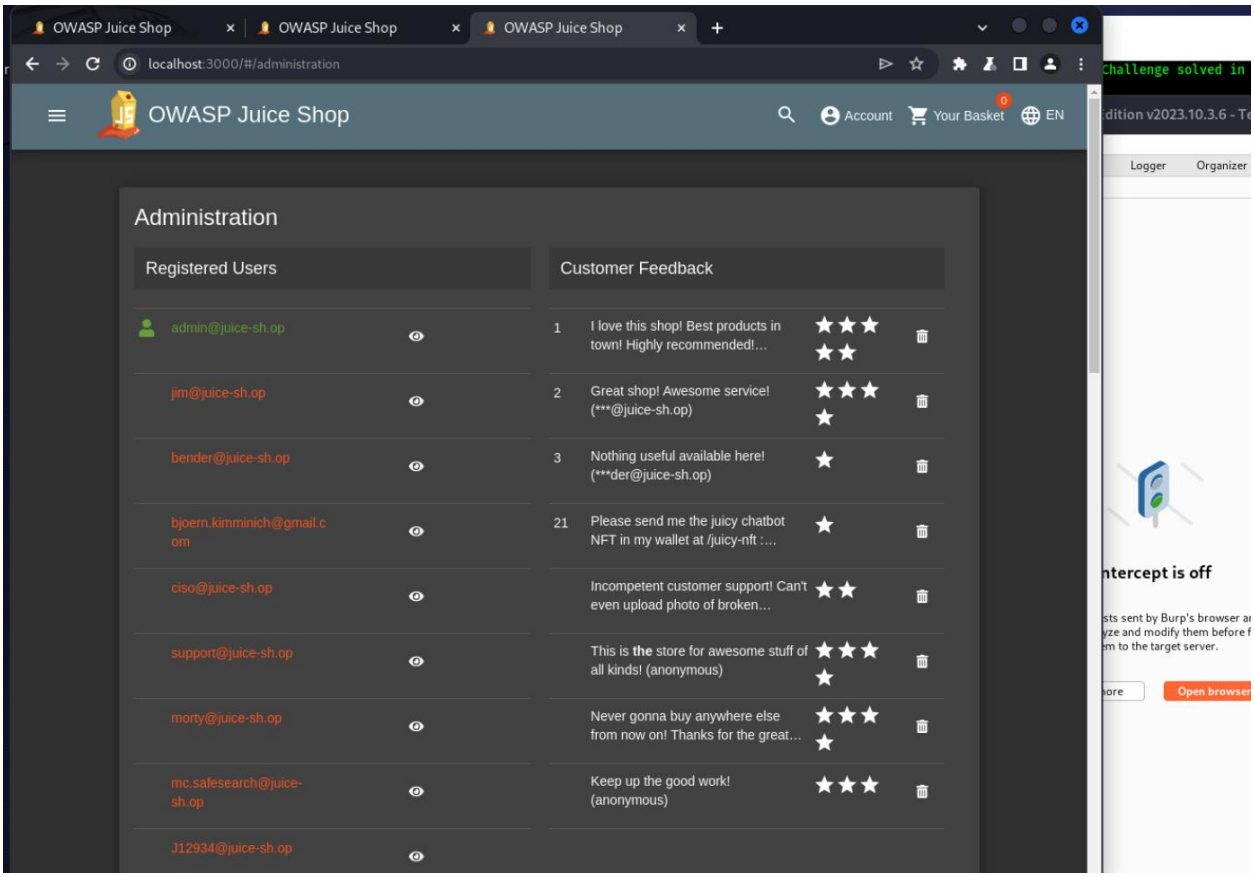




- Exploring broken access control
- In this section, I utilized Burp Suite to find and take advantage of faulty access control on a weak web site, as in OWASP Juice Shop.
- In the Repeater, modify the first line to retrieve the /administration/ directory or page from the web application.
- Next, after Repeater sends the updated request message to the web application, the HTTP status code indicates Repeater is able to successfully retrieve it.

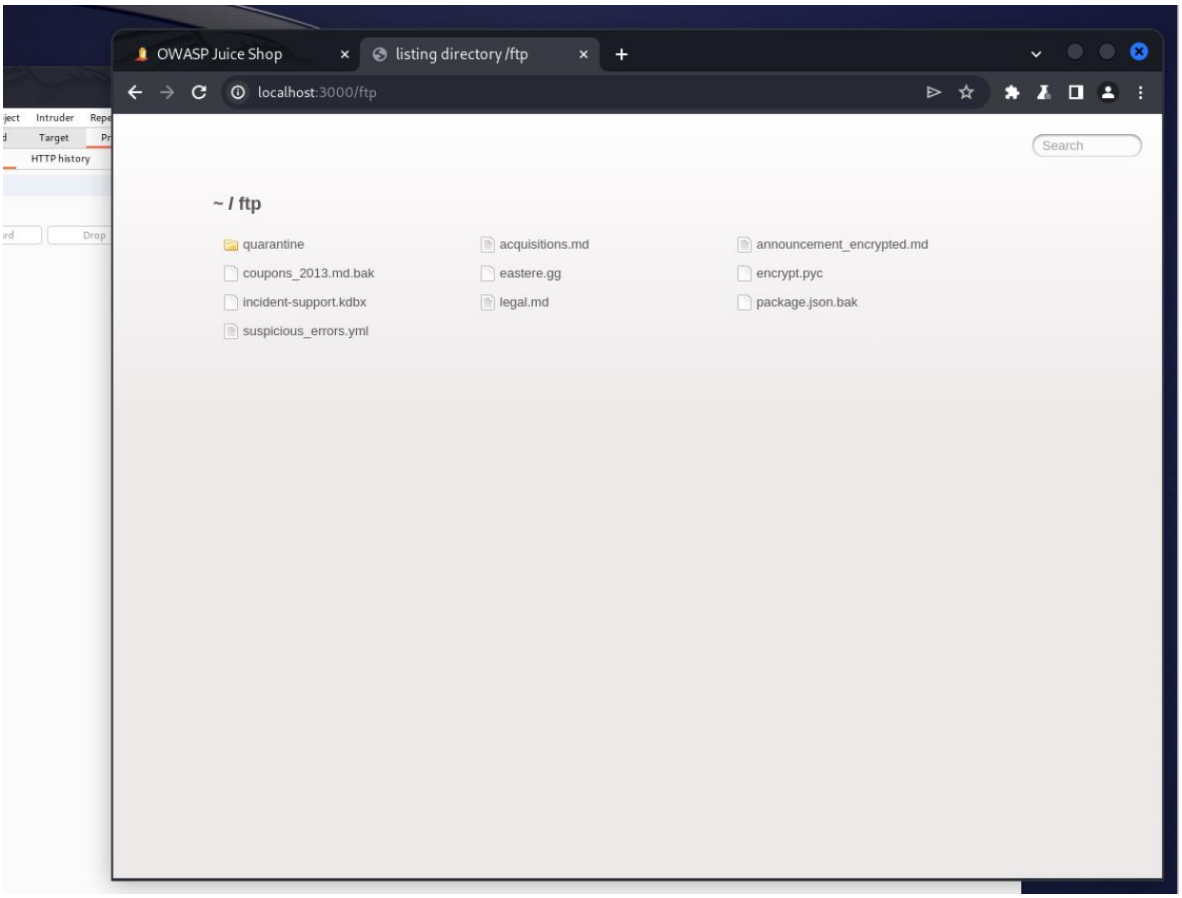


- As a result, when I opened a browser and typed the URL to http://localhost:3000/#/administration/, I was prompted with the administrative section within the vulnerable web application.



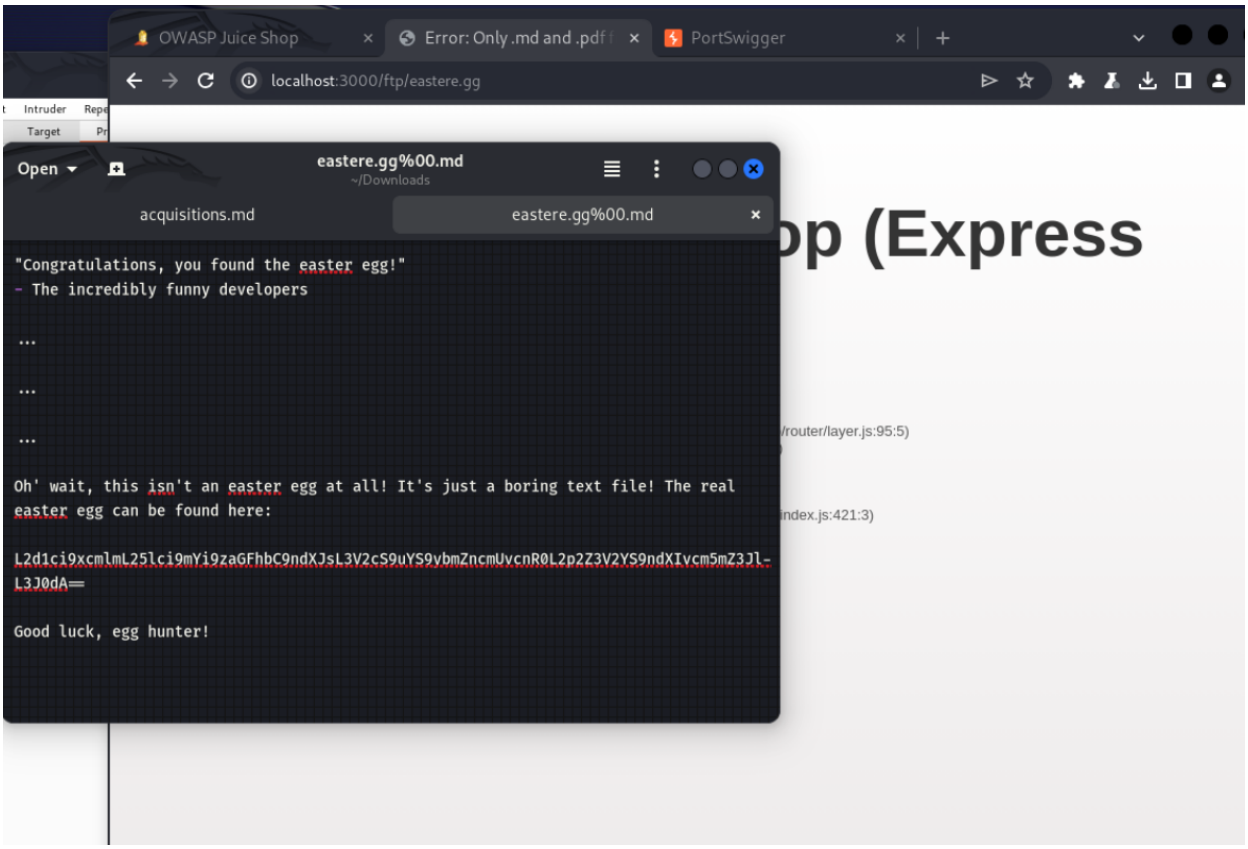
- Exploiting cryptographic failures
- This section of the lab involved me completing the Nested Easter Egg Challenge in order to illustrate the security vulnerabilities and potential exploits.

- The set-up process is the same as the rest of the previous challenges. I am using Kali Linux and started at the OWASP Juice Shop Score Board section.
- The first step is to input the following link and browse the directory to view the contents, I will be using eatere.gg file.



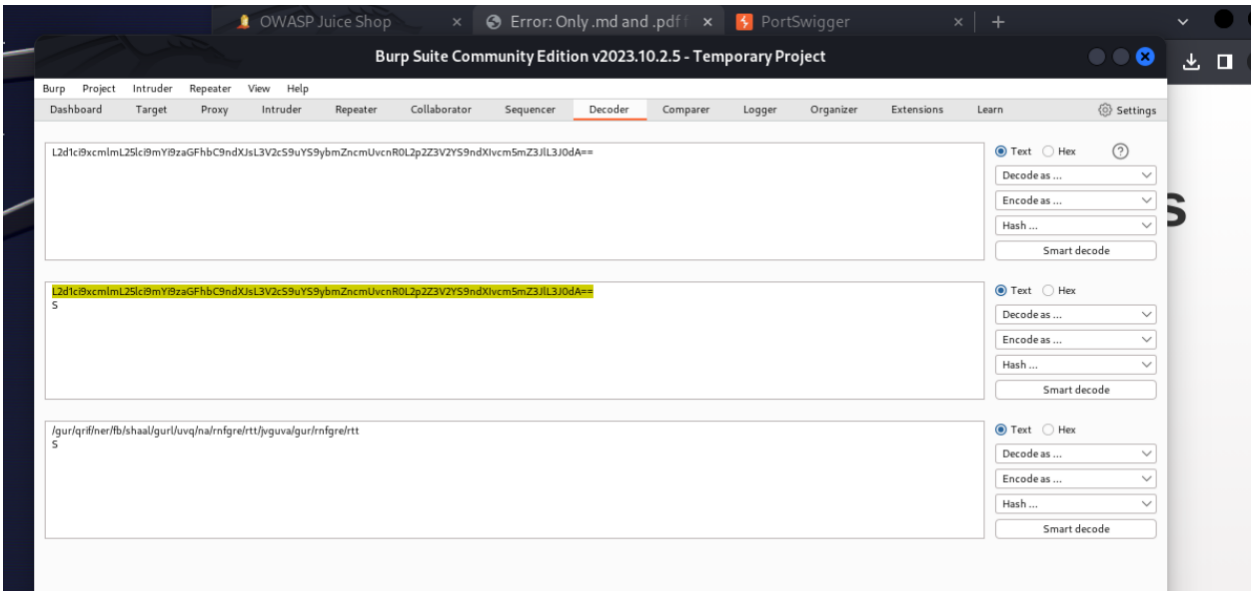
- Upon opening the file after converting, the contents are shown within the mousepad editor.

- The contents of the file display a hash value. I will use an identifier to decrypt the contents of the value.



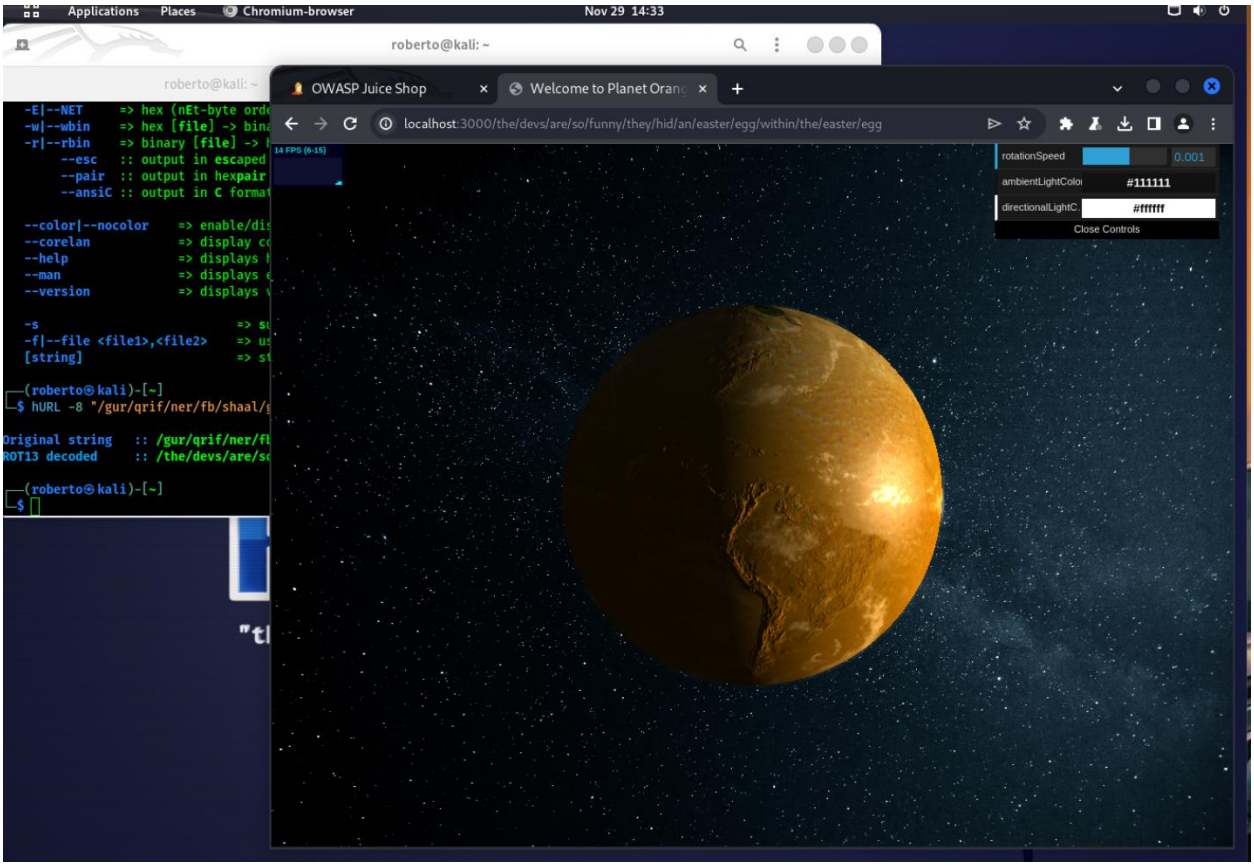
- The decoder tool I used is from the Burp Suite application.

- The result demonstrated a cryptographic hash for something that looks like the path of a web address.



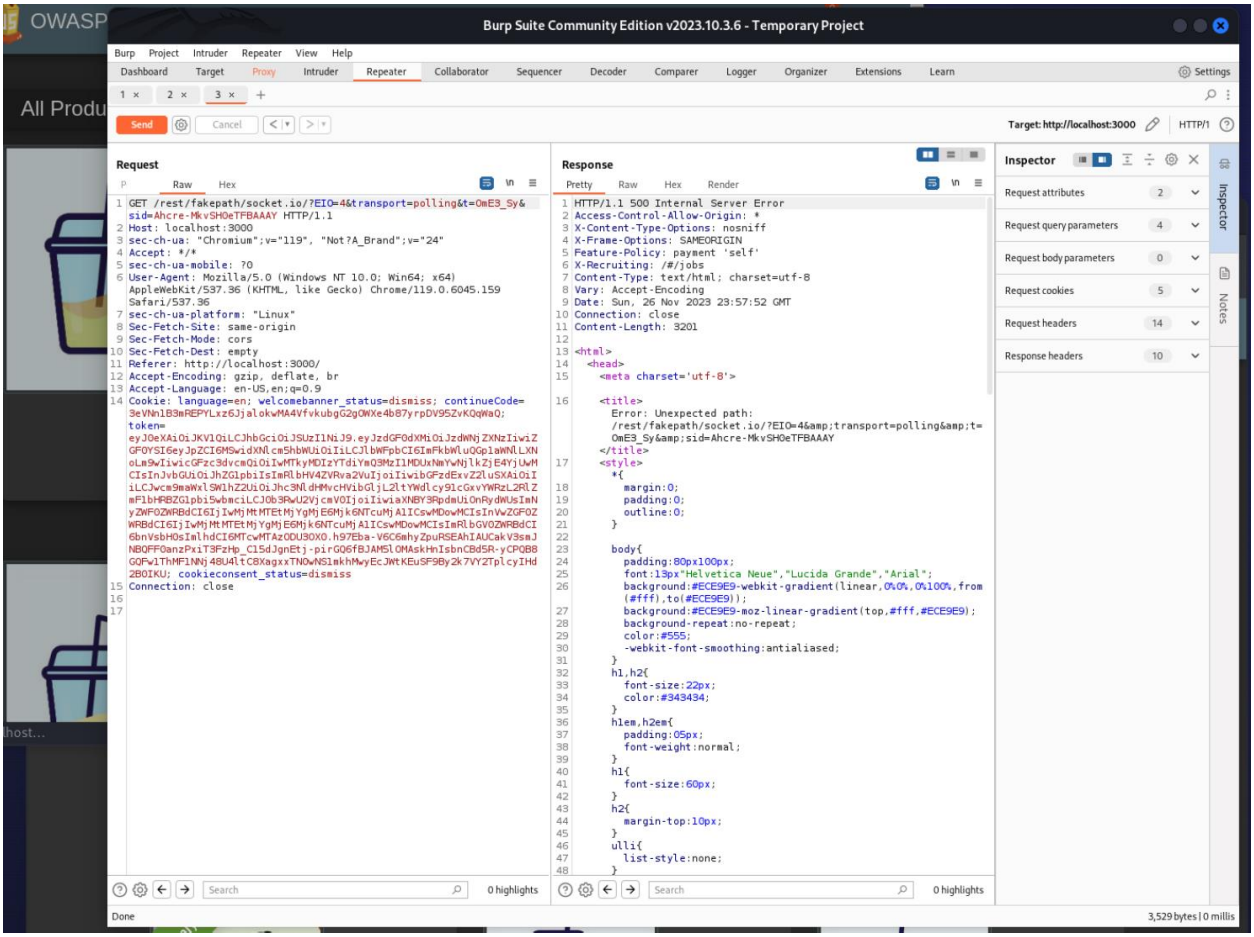
- The next step was to install hURL, which is a tool used to encode and decode various types of character-offset encryption ciphers.

- As a result, I was promoted with the plaintext message that it had been successfully decrypted. Using the message, I pasted it into my web browser and got the following result.



- Exploiting security misconfigurations

- In this section, I went through the steps in order to take advantage of security lapses that can be discovered in a weak online application like OWASP Juice Shop.
- To start with this section, I ensured that my Kali Linux was powered on and had an OWASP Juice Shop Docker instance up and running.
- By refreshing the OWASP Juice Shop page, I am prompted to create a new instance and send that information to the repeater section.
- After this, I appended a fake path to the HTTP header (/rest/fakepath), but I will be prompted with a server error since it was made up.



- To complete this section, I uploaded an unsupported file to the OWASP Juice web application, such as the one below (an xml file). When hitting the submit file button, I understood now that the web application accepts the unsupported file type, which is another indication of a security misconfiguration that can be exploited.