Sally Oketch

sally.oketch@womentechsters.org

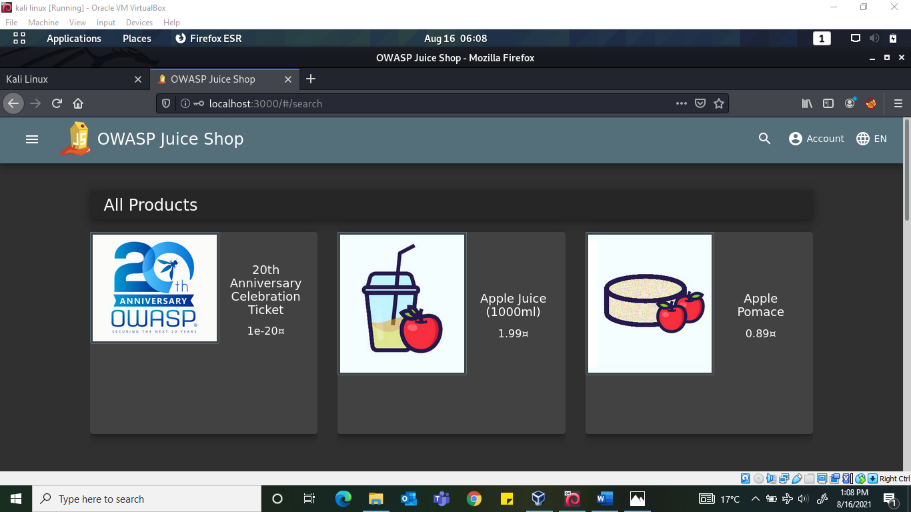
**Practical activity 10**

**Task 1 : Open for business!**

Deploy the machine( I’m using kali Linux virtual machine so I have started it).

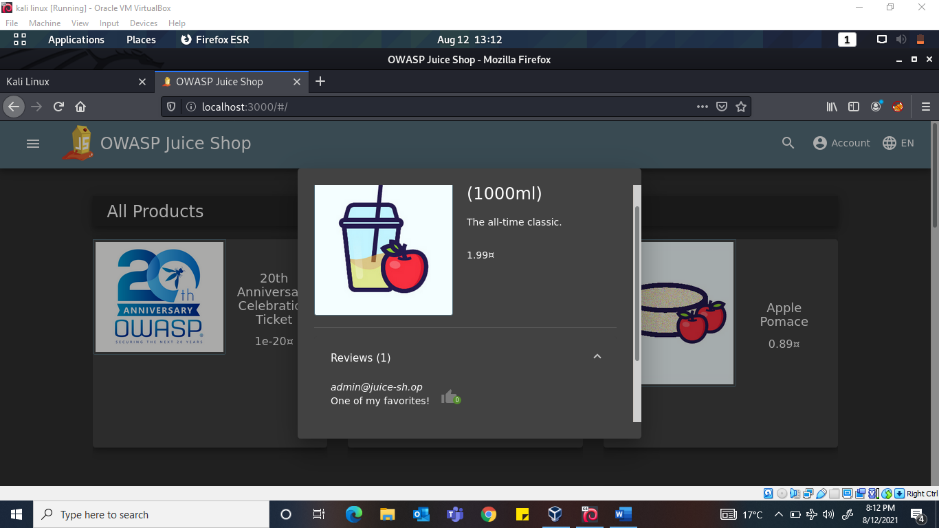
**Task 2: Let’s go on an adventure!**

Deploy the juice shop.



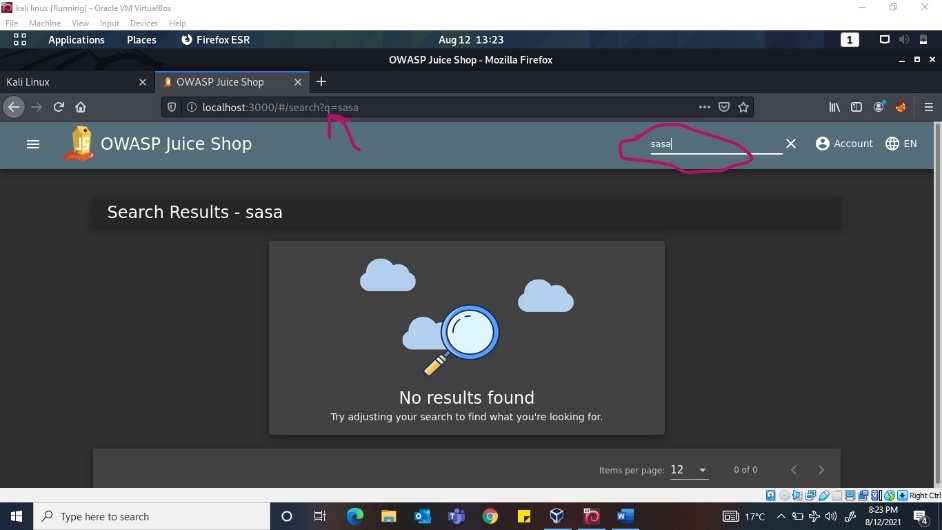
*Question 1: What’s the administrator’s email address?*

When we click apple juice and checking on the review. The email is [***admin@juice-sh.op***](mailto:admin@juice-sh.op)



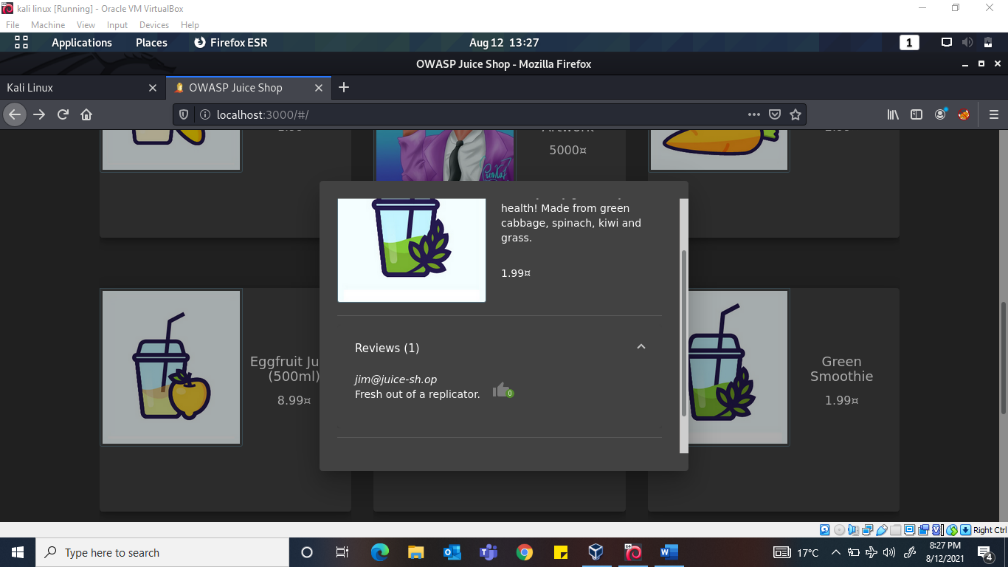
*Question 2: What’s the parameter is used for searching?*

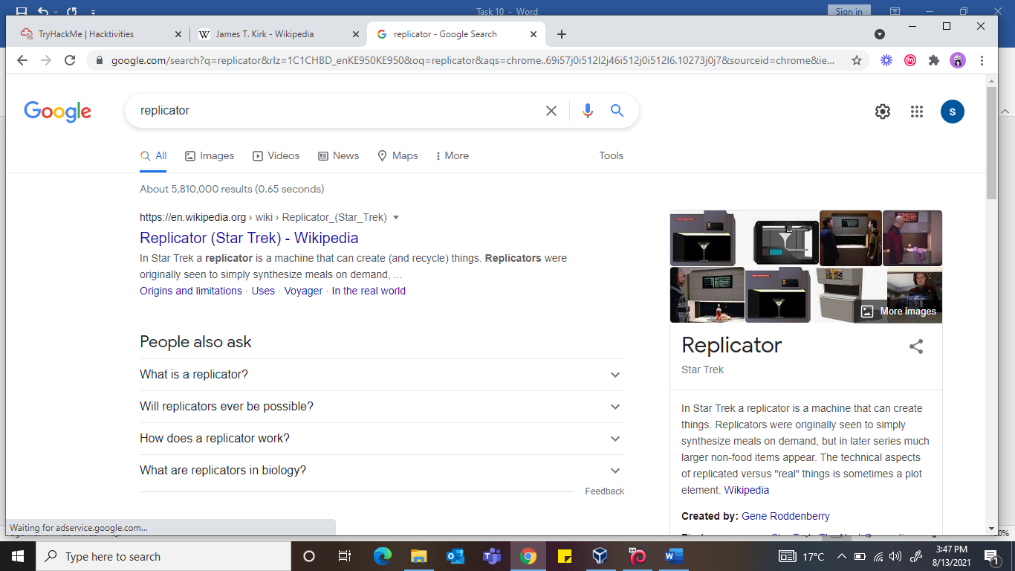
Click on the magnifying icon to open the search options and search anything and press enter. Look at the URL and note the ***q*** which represents the searching queries we input.



*Question 3: What show does Jim reference in his review?*

Jim wrote a review on green smoothie about a replicator which after googling it is a machine that creates things in a TV series called ***star trek***.

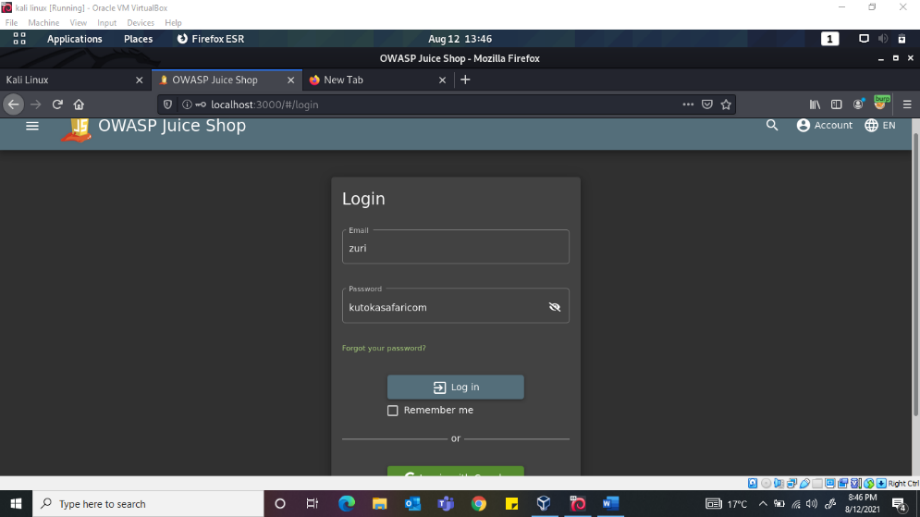


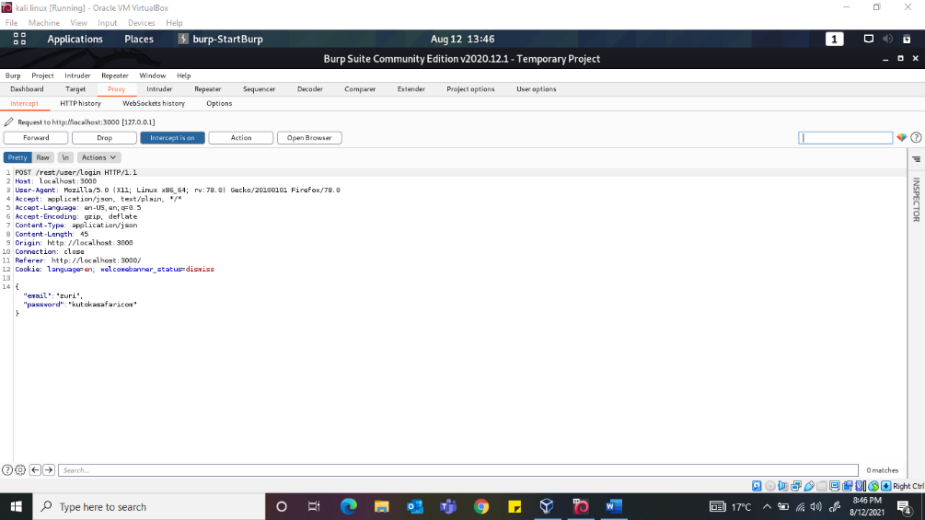


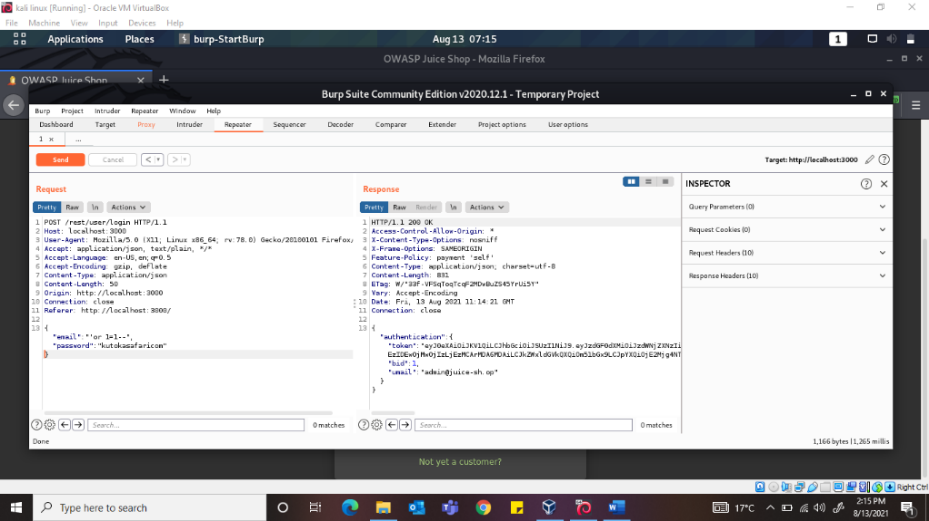
**Task 3: Inject the juice**

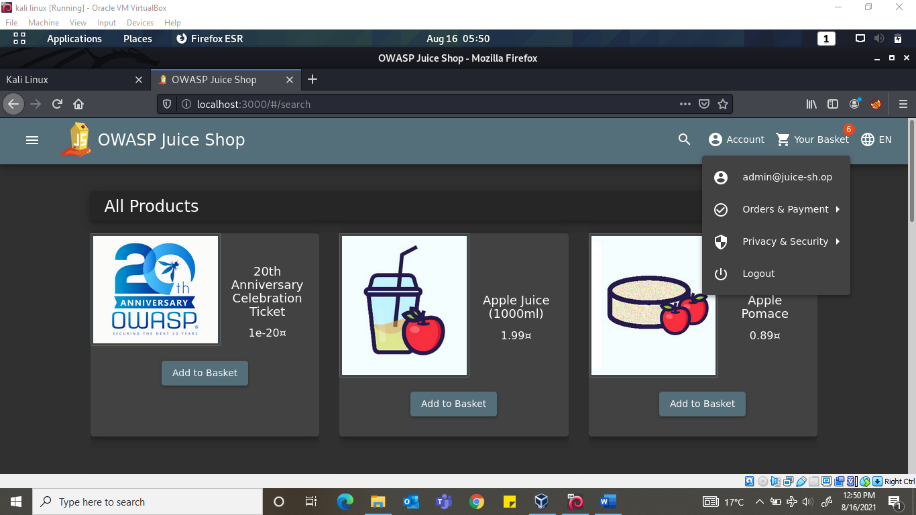
*Question 1: Log into the administrator account!*

First, we will try to login with some random email and password since we don’t have valid credentials. We will also start burp suite to intercept the request. After successfully intercepting the request, now change the input variables and write an SQL query to launch SQL injection. Change the email to ” ‘ or 1=1—” and forward it to the server to complete the request.





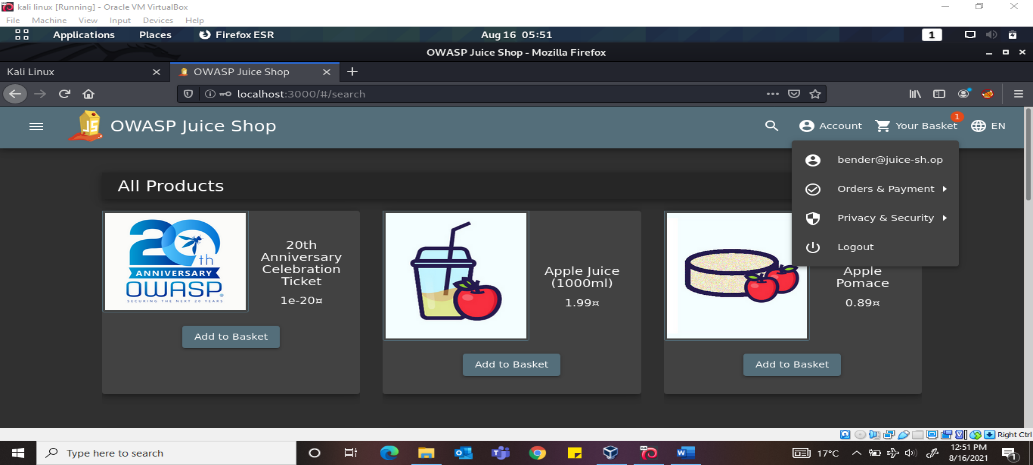
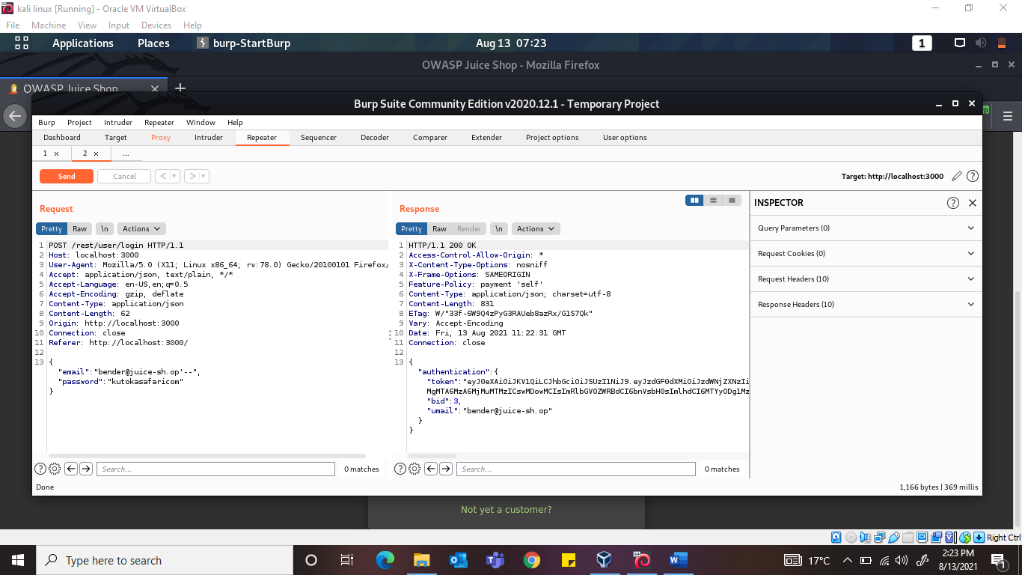


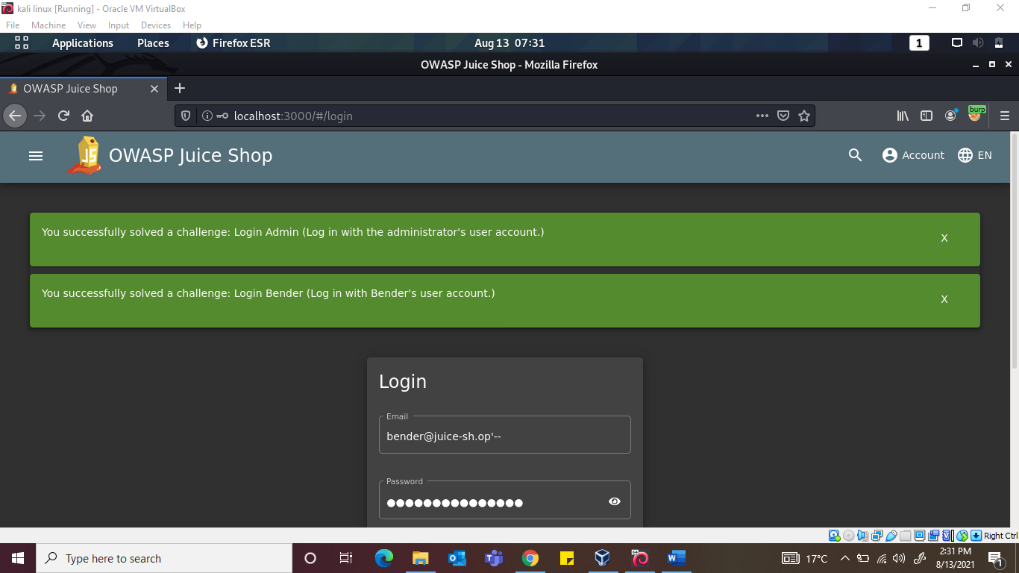


**Explainer:** *the character* ***‘*** *will close the brackets in the SQL query,* ***or*** *in a SQL statement will return true if either side of it is true.* ***1=1*** *is always true hence it will tell the server that the email is valid and log us into user id 0 which is of the administrator, -- character is used to comment out data in SQL, any restrictions on the login will no longer work as they are interpreted as a comment.*

*Question 2: Log into the bender account!*

Same process as the question above, but in the email we input [bender@juice-sh.op](mailto:bender@juice-sh.op) since it is a valid email address. Then add ‘ – to bypass the login system.

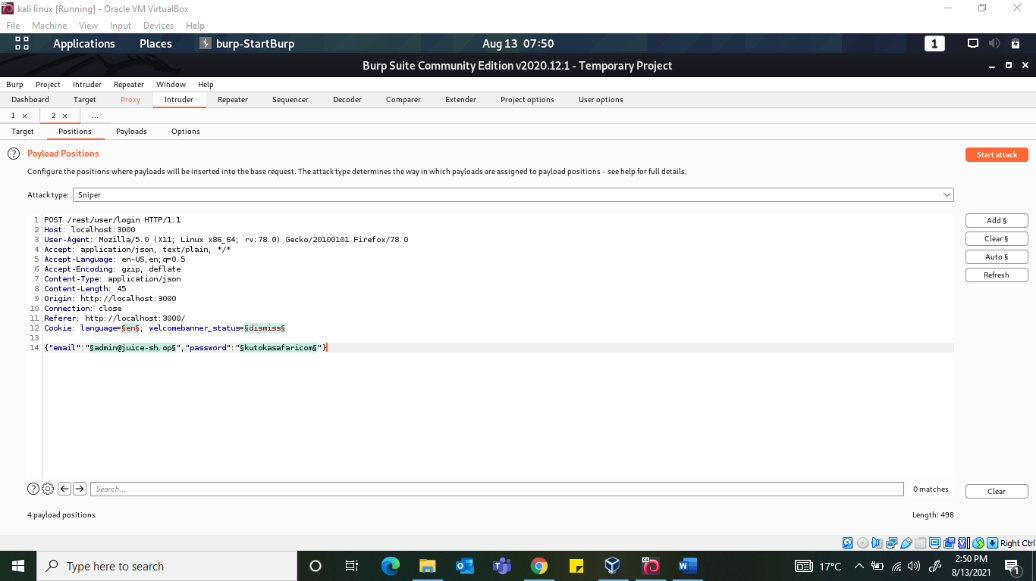




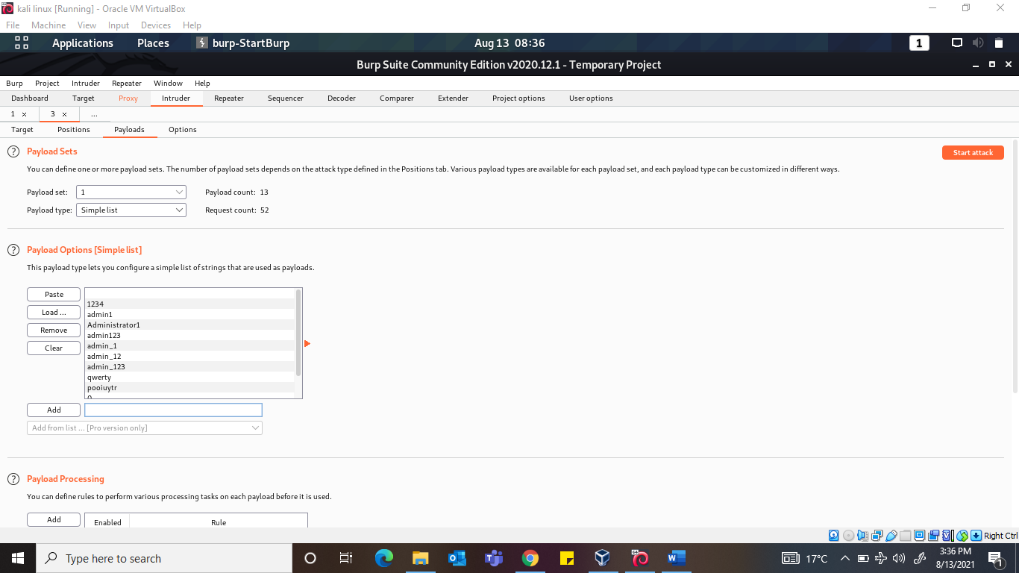
**Task 4: Who broke my lock?**

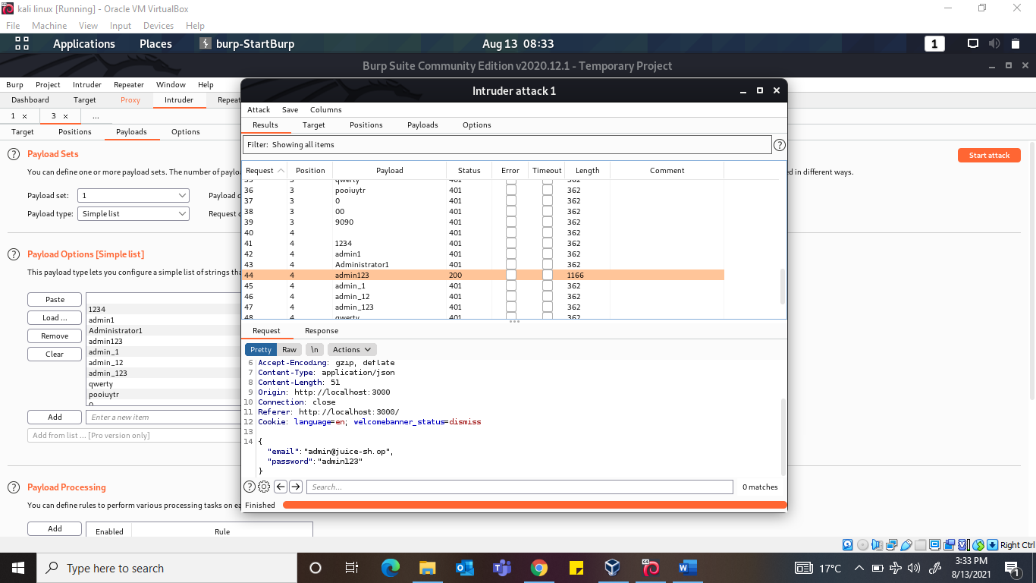
*Question 1: Bruteforce the administrator account’s password?*

Lets capture a login request and send it to intruder,



and select the wrong password we entered and then payloads to customize payload settings. I added some common passwords in the payload options and left the ret as default then pressed start attack.

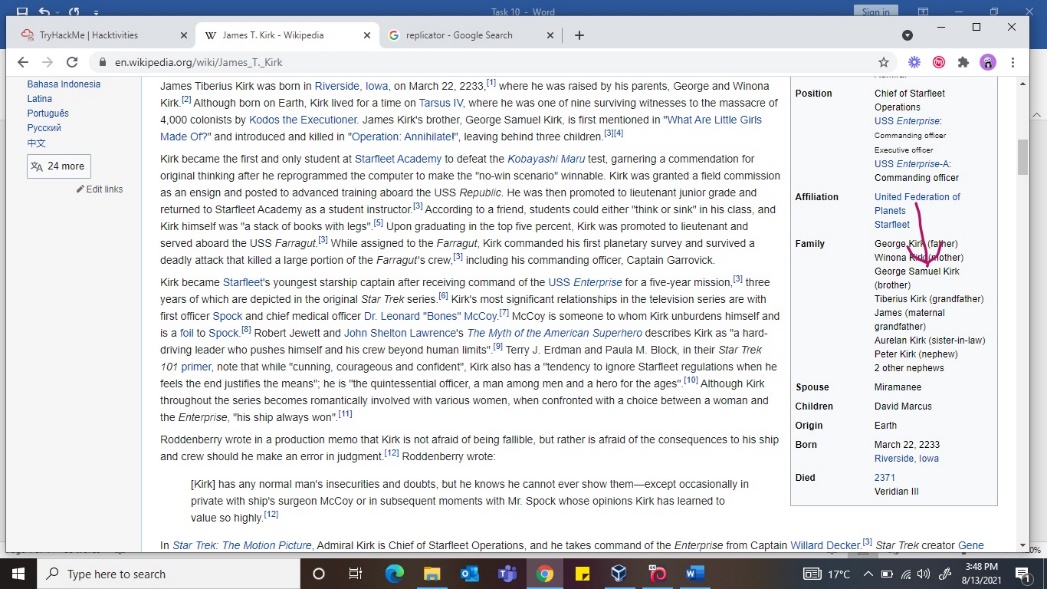


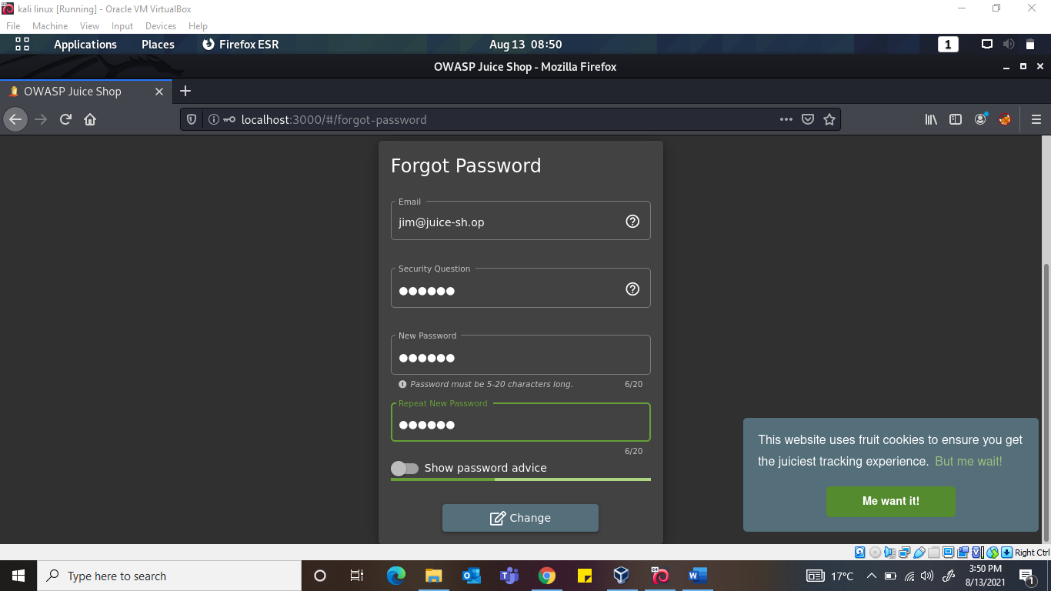


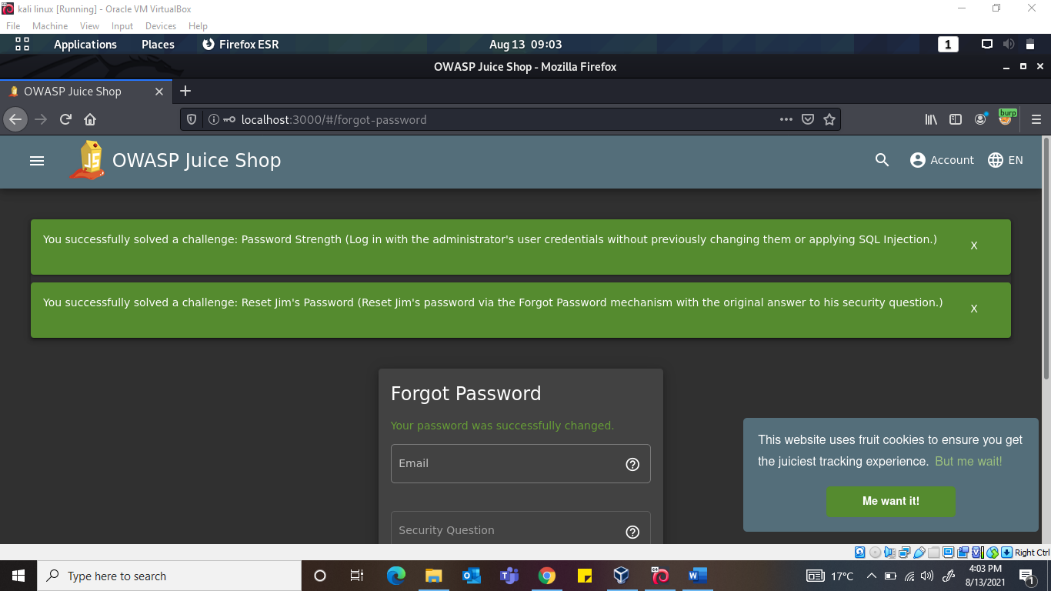
Burp will try all the passwords to get one that matches and luckily for me I have one that is successful the one with a status 200 the ones with 401 are not successful. The password ***admin123***

*Question 2: Reset Jim’s password!*

Here we will have to be more creative and look for more information about Jim. From the email we got from his review we have [jim@juice-sh.op](mailto:jim@juice-sh.op), hopefully he has a dull mind and has information that might compromise his password online. We go to the login page and click forgot password takes us to the reset password page, the security question is “your eldest sibling’s middle name”. Since the user has some connection with star trek we will look for any person associated with it with an almost similar name. Jim is sometimes used to refer to James so James T. Kirk is our guy(maybe). In his Wikipedia we see he has an elder brother called ***Samuel***. Let us try using this info to reset the password. Press change and voila! We have successfully reset Jim’s password.



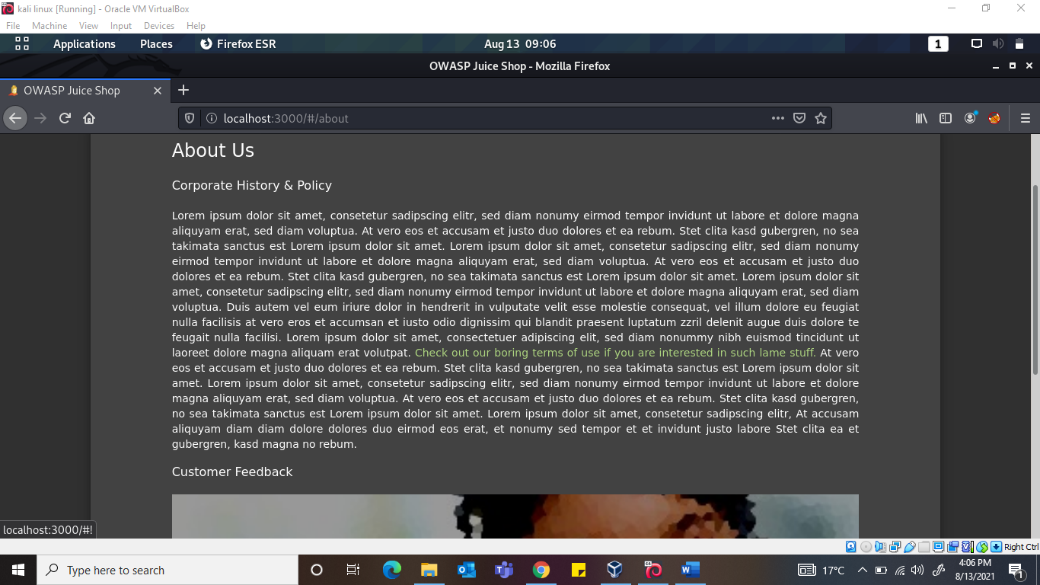


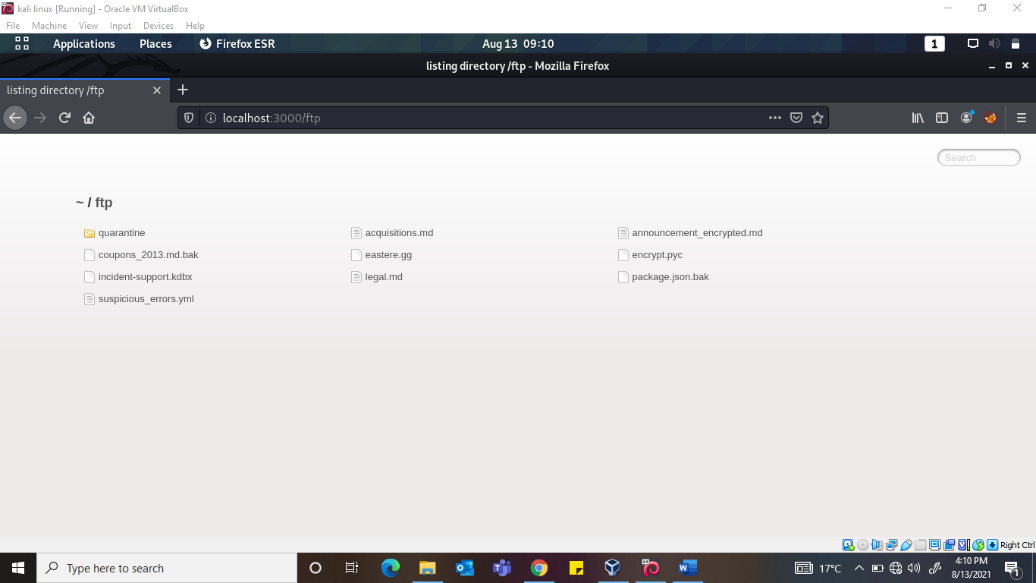


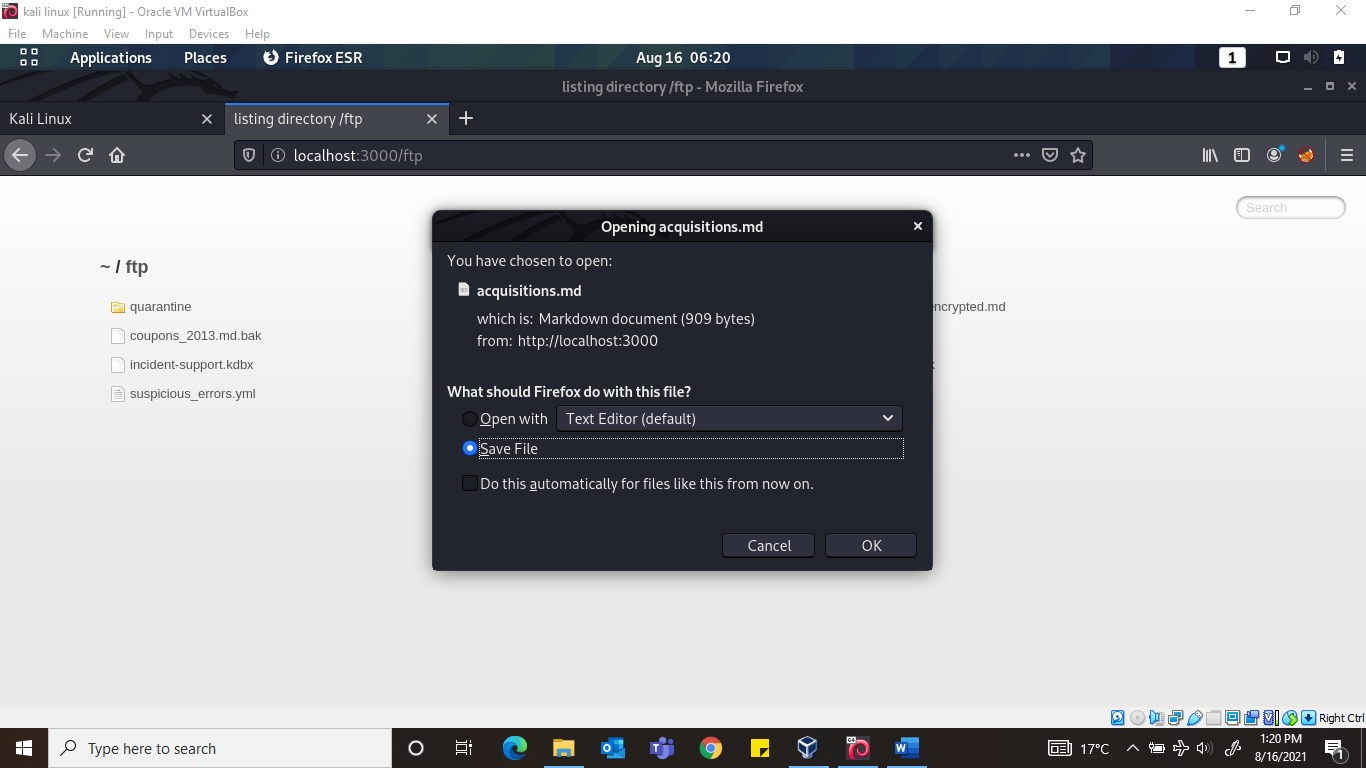
**Task 5: AH! Don’t look!**

*Question 1: Access the Confidential Document!*

*We navigate to the about us page and hover over the “check out our boring terms of use….” And we see the link* [*http://localhost:3000/ftp/legal.md*](http://localhost:3000/ftp/legal.md)*. We will go to the* [*http://localhost:3000/ftp/*](http://localhost:3000/ftp/) *location and download the acquisitions.md and save it. We can also see everything on that page.*

**

**

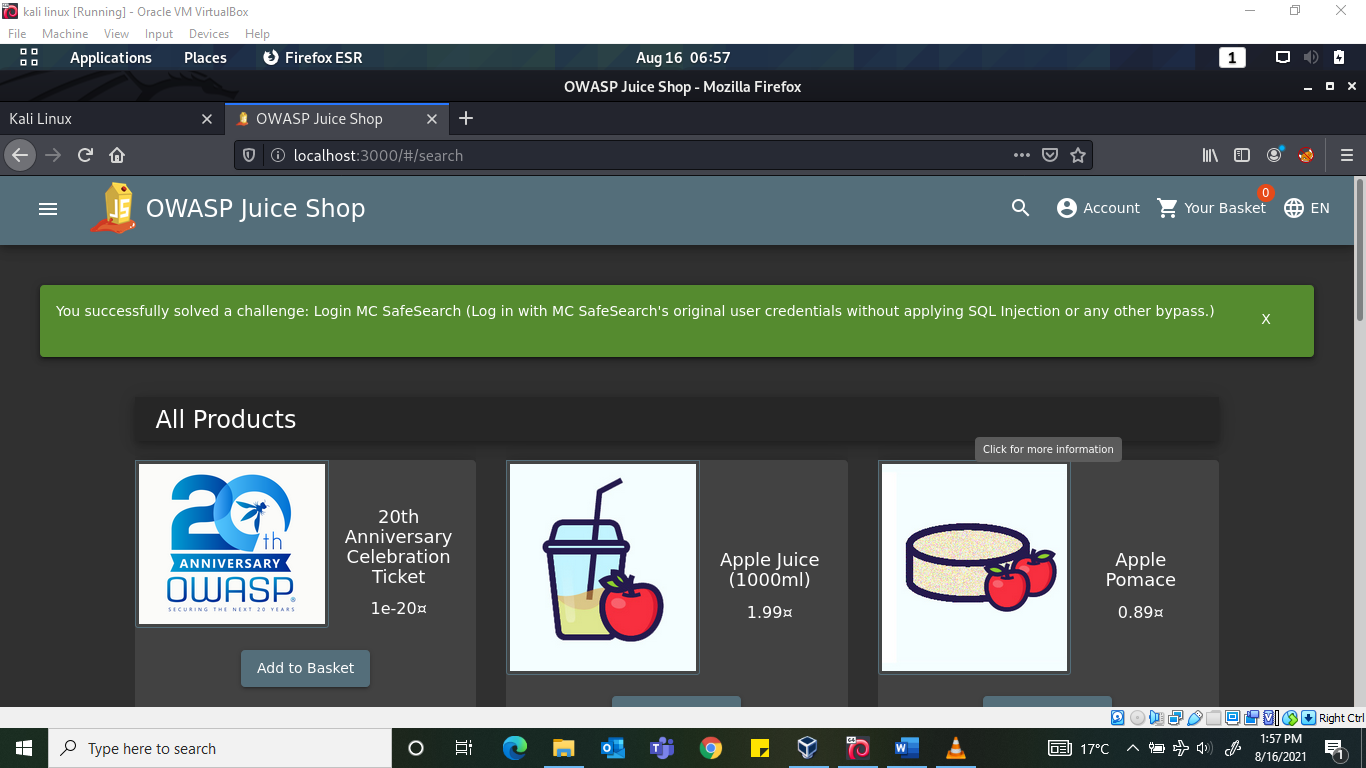


*Question 2: Log into MC Safesearch’s account!*

After watching the video there are certain parts of the song that stand out.

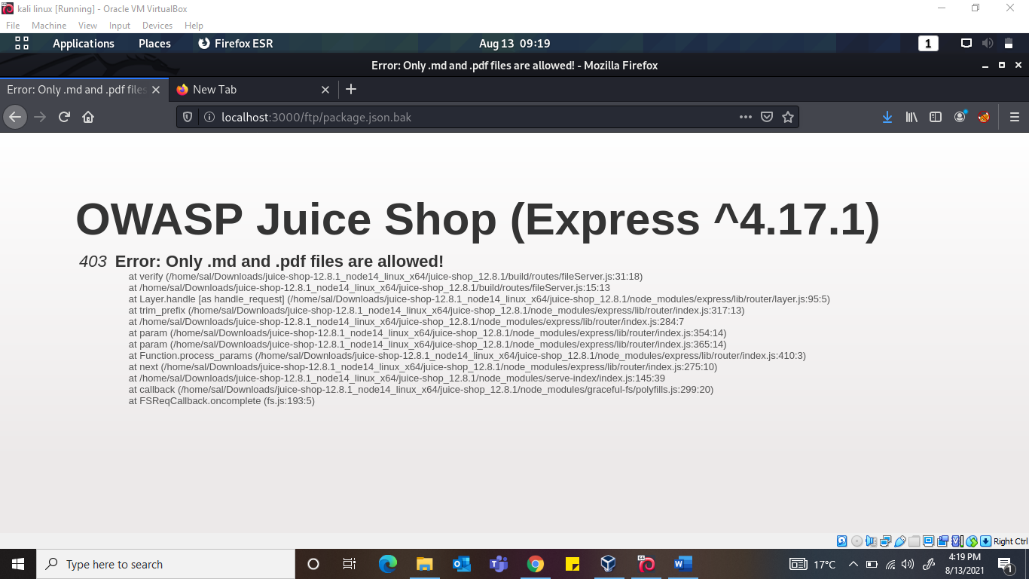
He notes that his password is “**Mr. Noodles**” but he has replaced some “**vowels into zeros**“, meaning that he just replaced the **o**‘s into **0**‘s.

We now know the password to the **mc.safesearch@juice-sh.op**account is “**Mr. N00dles**“



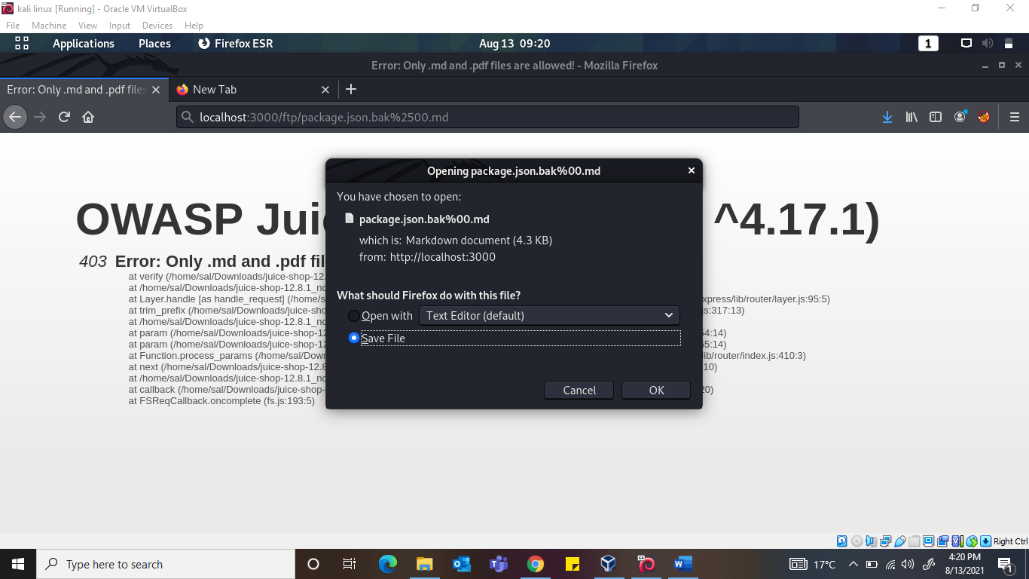
*Question 3: Download the backup file!*

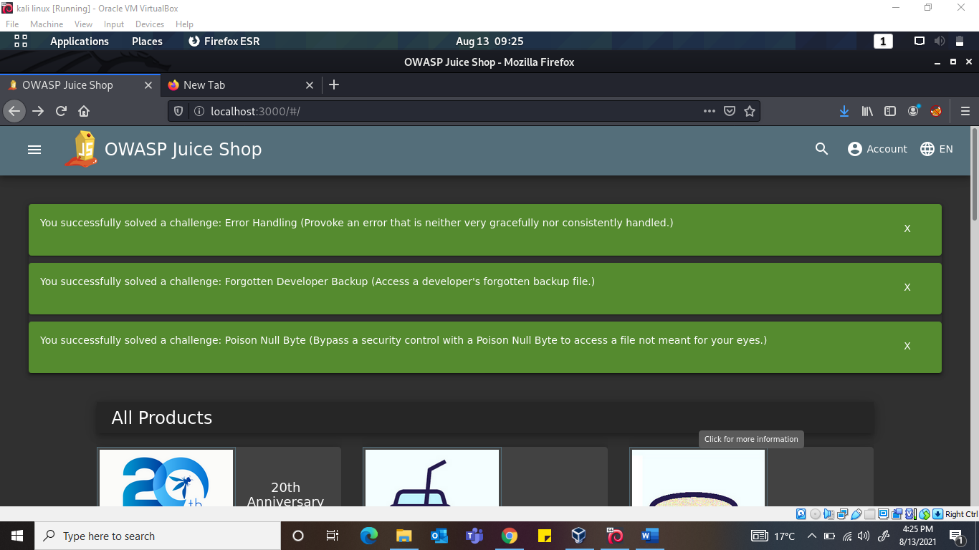
*Go to the* [*http://localhost:3000/ftp/*](http://localhost:3000/ftp/) *folder and try downloading package.json.bak.* But it seems we are met with a 403 which says that only .md and .pdf files can be downloaded.



To get around this, we will use a character bypass called “**Poison Null Byte**“. A Poison Null Byte looks like this: **%00**.

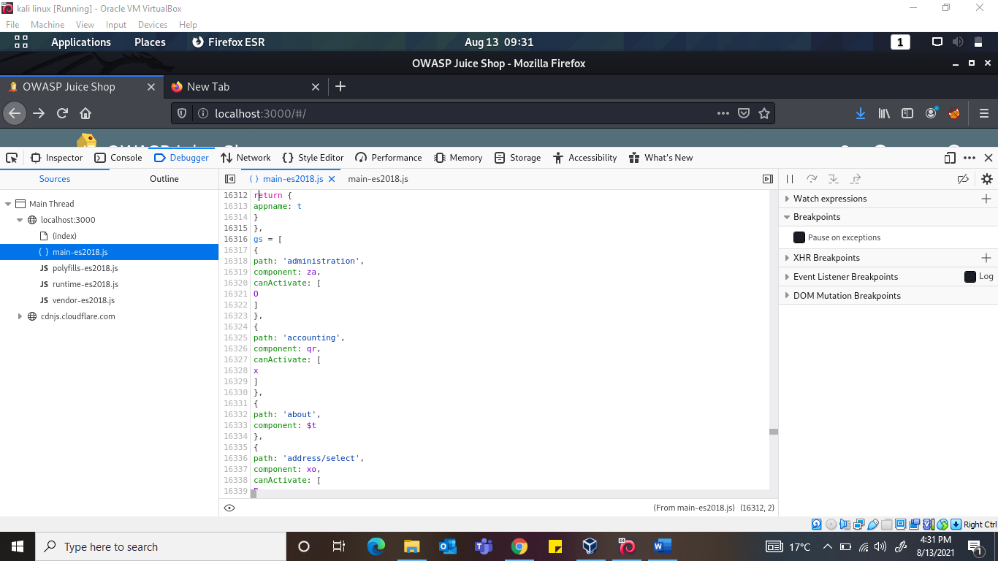
Note that we can download it using the url, so we will encode this into a url encoded format. The Poison Null Byte will now look like this: **%2500**. Adding this and then a **.md**to the end will bypass the **403** error!





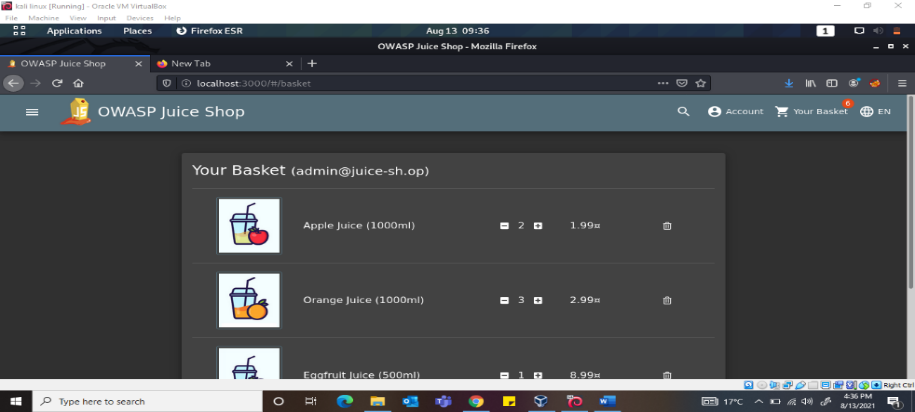
**Task 6: Who is flying this thing?**

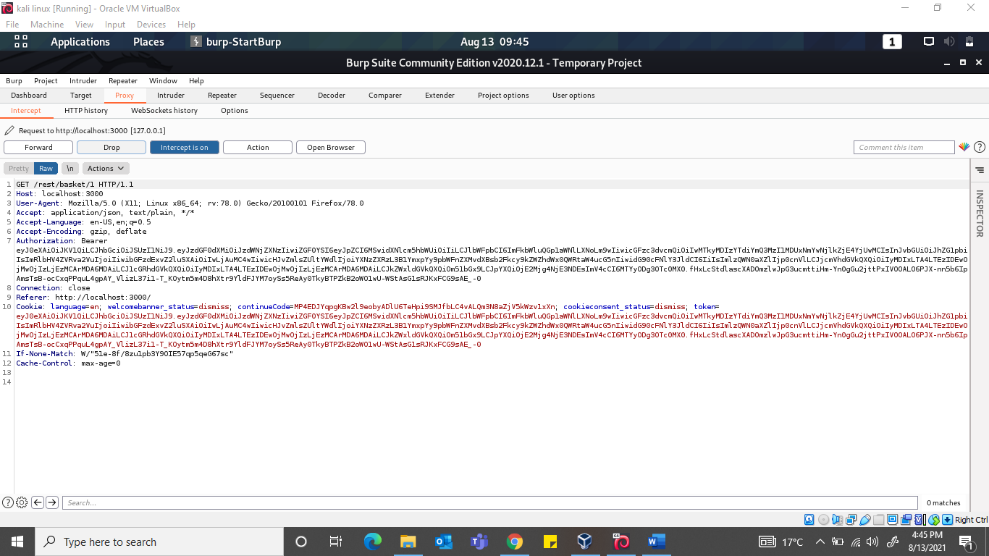
*Question 1: Access the administration page!*

**

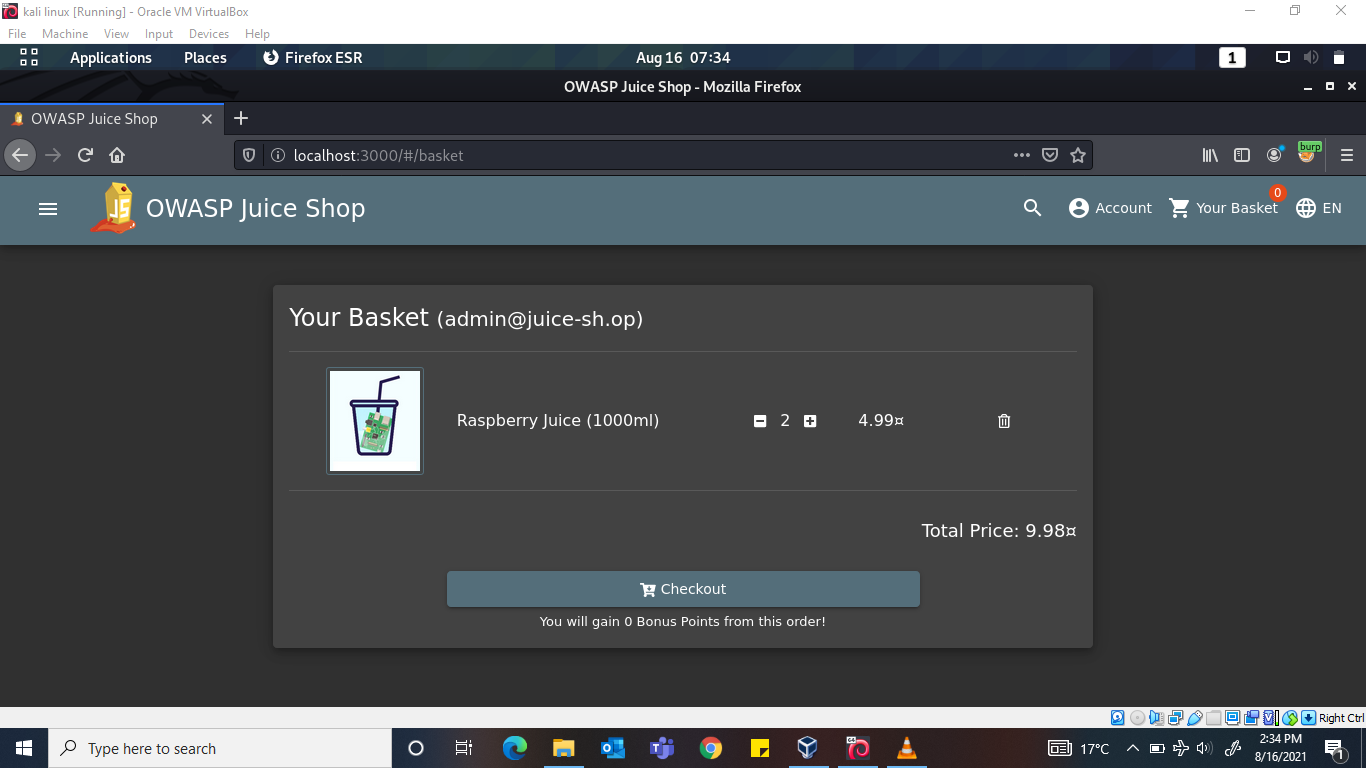
*Question 2: View another user’s shopping basket!*

Login to the Admin account and click on ‘Your Basket’. Make sure Burp is running so you can capture the request! Forward each request until you see: GET /rest/basket/1 HTTP/1.1.



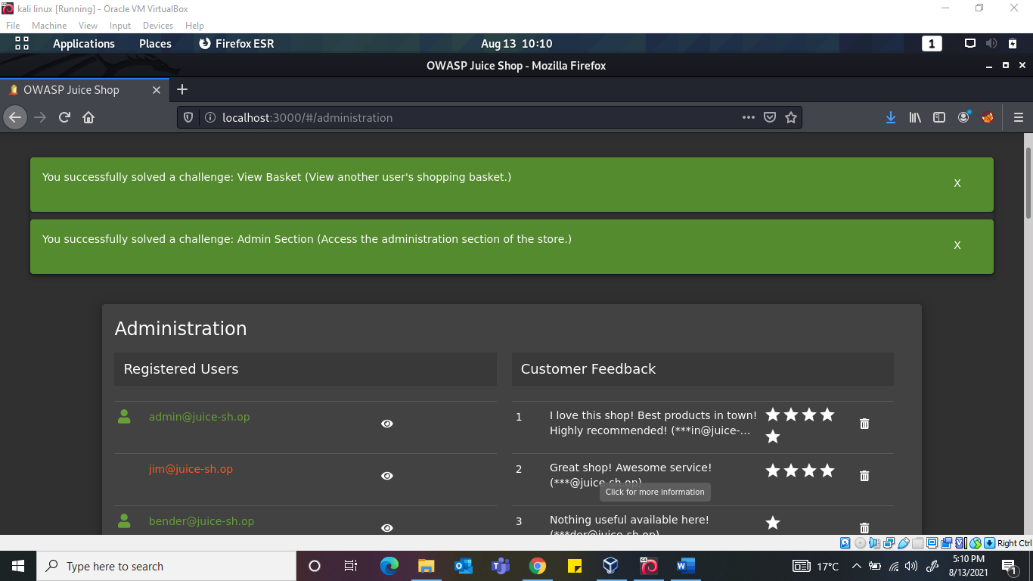


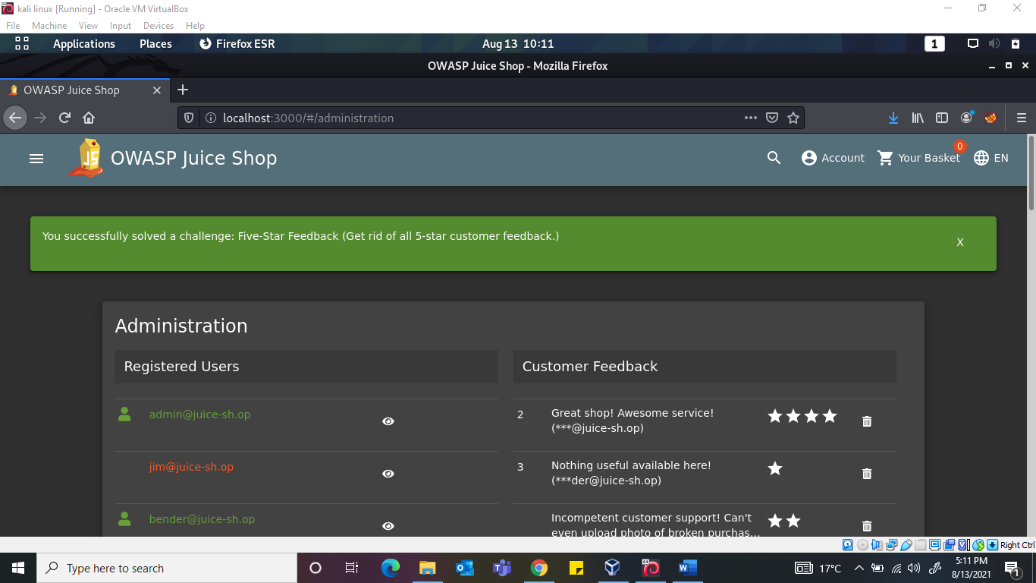
Change 1 after /basket/ to 2 which will now show the basket UserID 2.



*Question 3: Remove all 5 star reviews!*

*Navigate to the /administration page again and click the bin icon next to the review with 5 stars.*

**

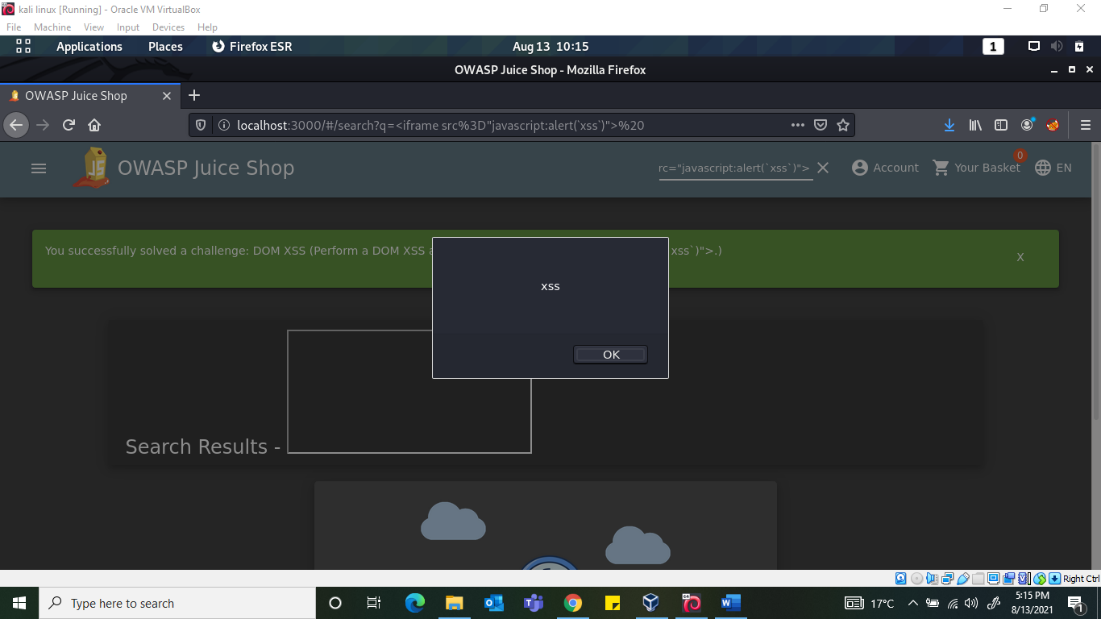
**

***Task 7: Where did that come from?***

*Question 1: Perform a DOM XSS!*

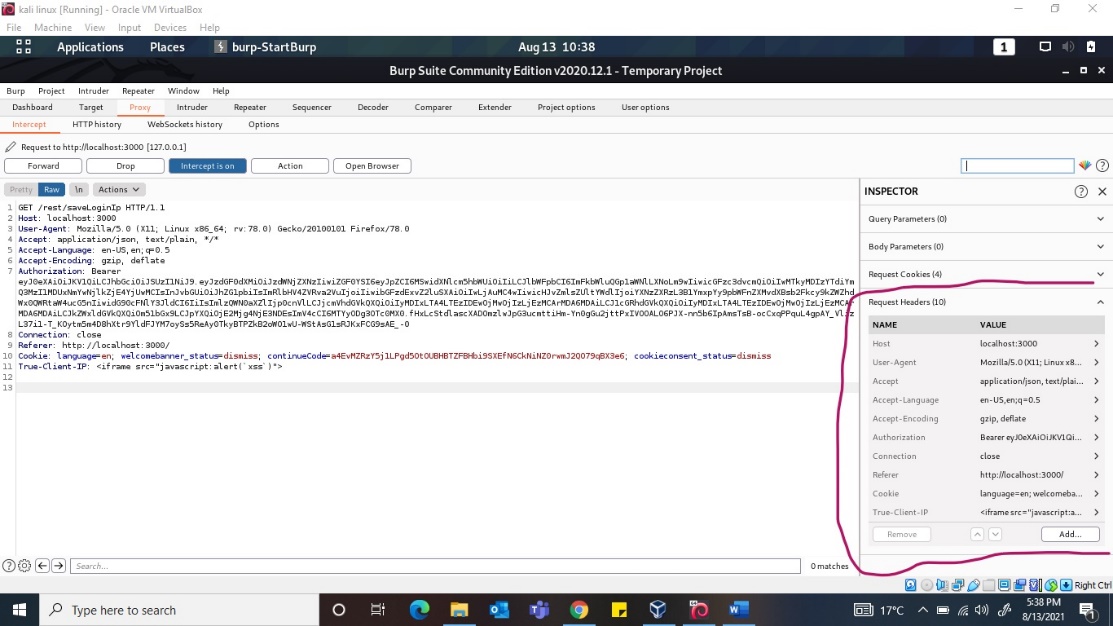
[XSS](https://portswigger.net/web-security/cross-site-scripting) or [Cross-site scripting](https://owasp.org/www-community/attacks/xss/) is a vulnerability that allows attackers to run JavaScript in web applications. These are one of the most found bugs in web applications. Their complexity ranges from easy to extremely hard, as each web application parses the queries in a different way.

After we input <iframe src=”javascript:alert(‘xss’)”> in search bar and press enter.

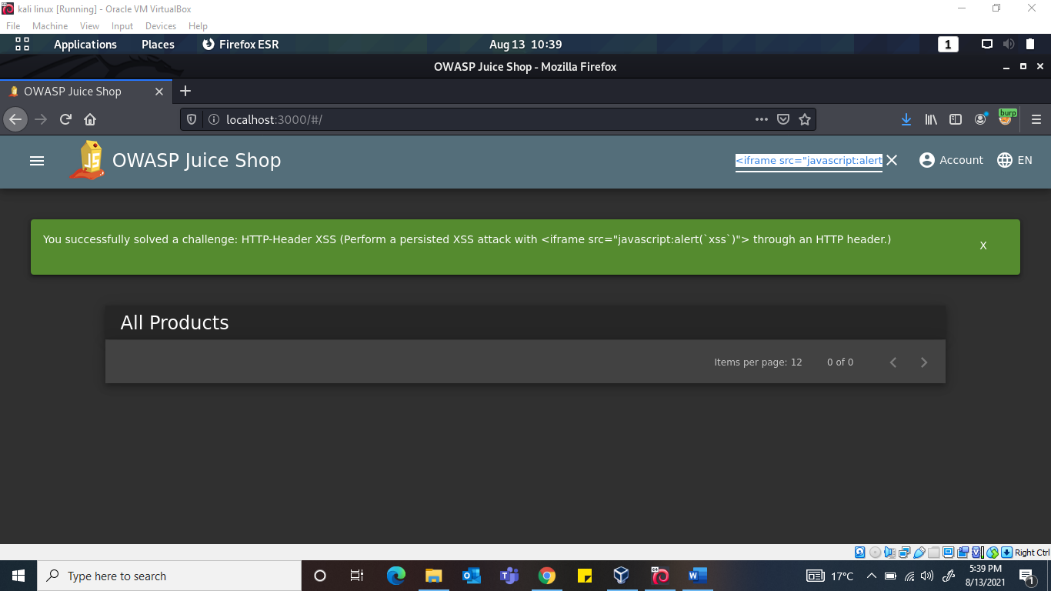


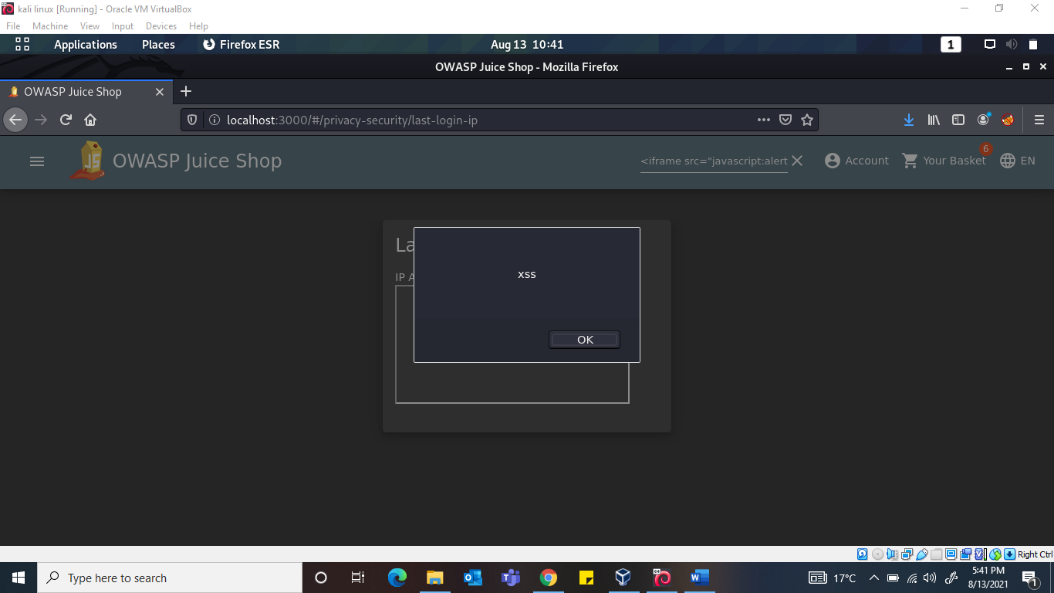
*Question 2: Perform a persistent XSS!*

Hop on to Burp and move to Headers section and create a new custom header. Fill out what is given in the docs.



Then forward the request to the server!  
When signing back into the admin account and navigating to the Last Login IP page again, we will see the XSS alert.



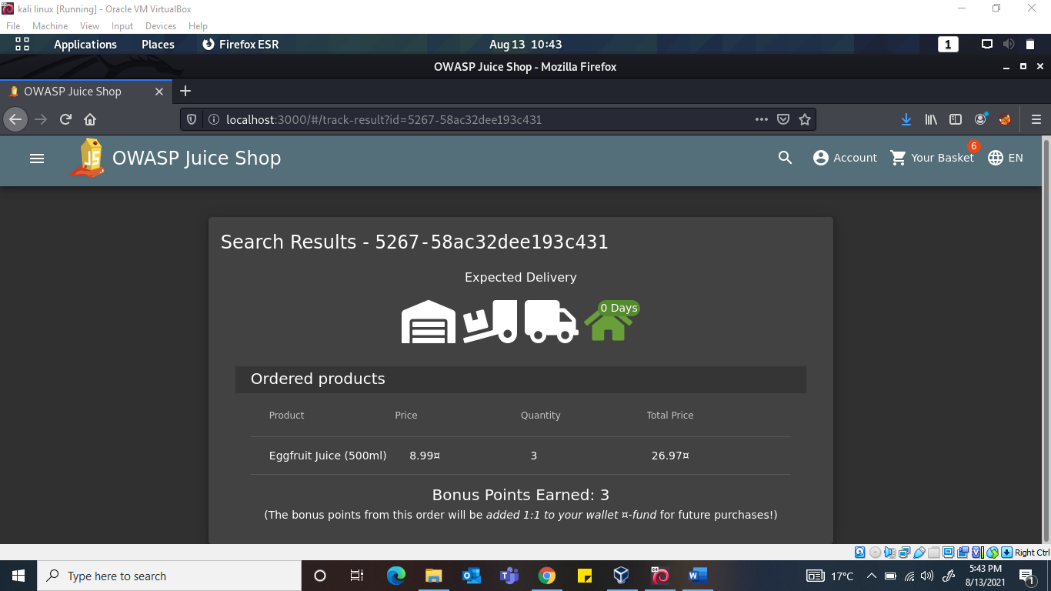


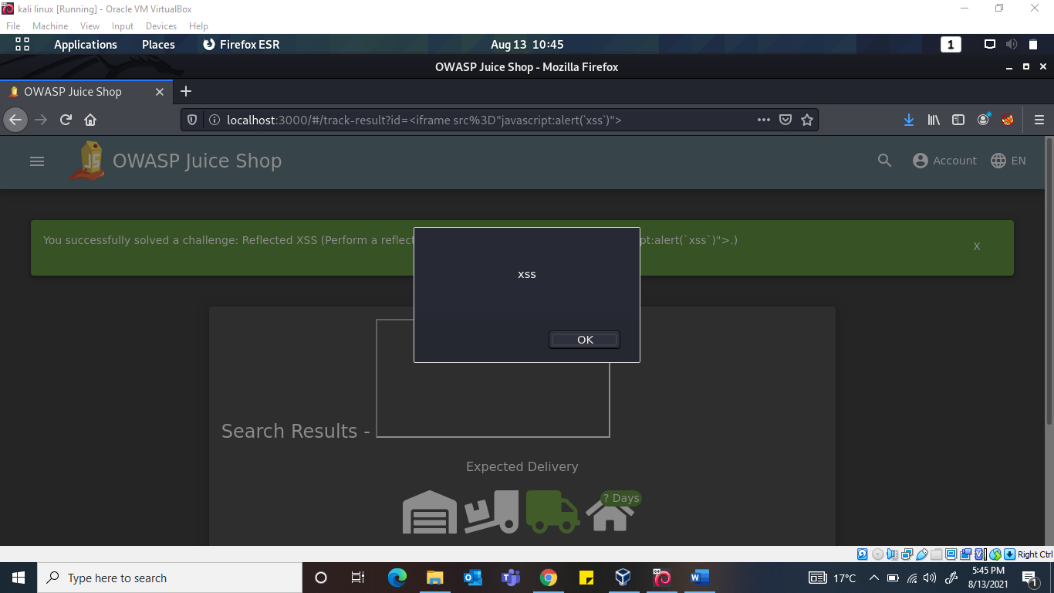
*Question 3: Perform a reflected XSS!*

First, we are going to need to be on the right page to perform the reflected XSS! Login into the **admin**account and navigate to the ‘**Order History**‘ page. From there you will see a “**Truck**” icon, clicking on that will bring you to the track result page. You will also see that there is an **id** paired with the order.

We will use the iframe XSS, **<iframe>src=”javascript:alert(xss)”>** , in the place of the **5267-f73dcd000abcc353**

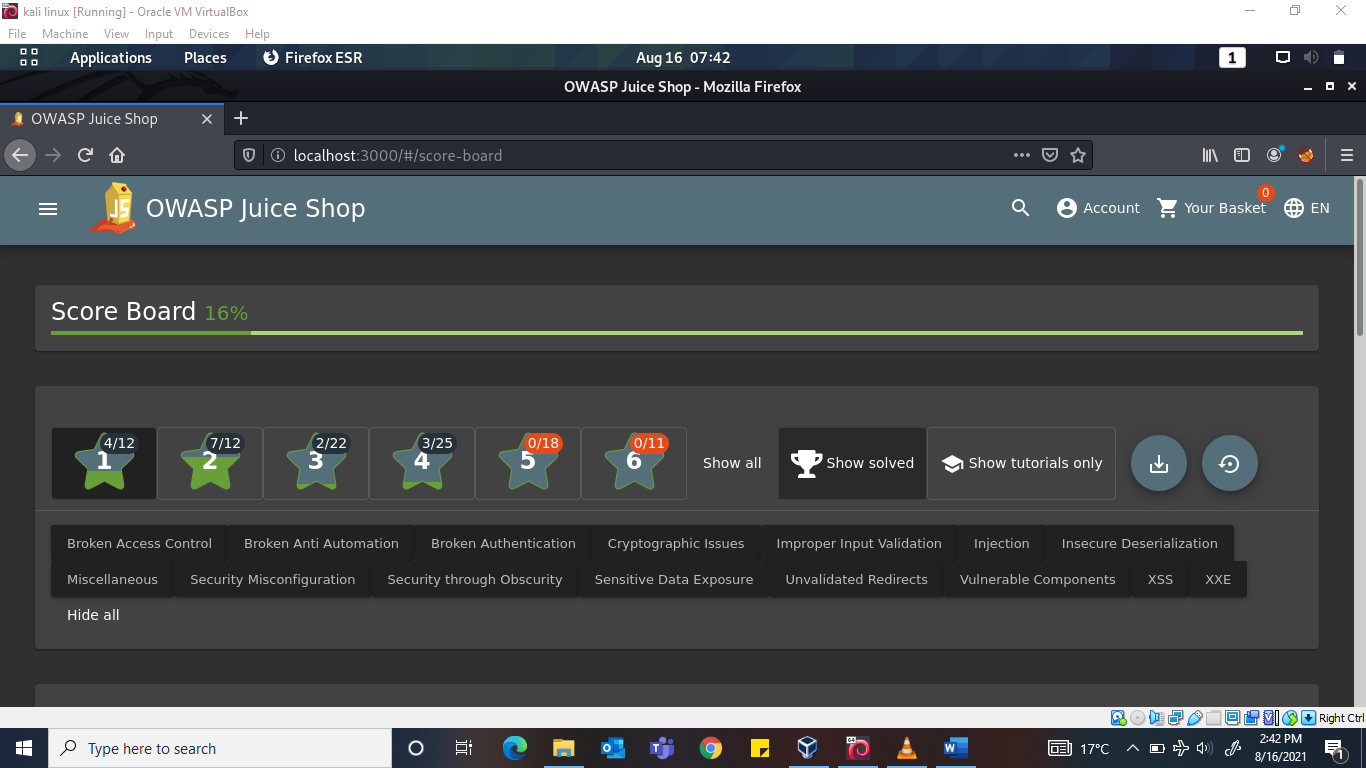
After submitting the URL, refresh the page and you will then get an alert saying XSS!





**Task 8: Access the scoreboard page**

/#/score-board/



My final results from the terminal since I was on my machine…