

THM-The-Owasp-JuiceShop-WriteUp

Step1:

Connect openvpn:

The screenshot shows two side-by-side interfaces. On the left, the 'OpenVPN Access Details' screen displays the VPN Server Name as 'EU-Regular-2' and the Internal Virtual IP Address as '10.9.3.126'. The 'Server status' is shown as 'Online' with a green dot. On the right, the 'Machines' screen shows 'EU-Regular-2' selected in the 'VPN Server' dropdown. A note says: 'If you're switching for the first time, you will need to redownload your configuration file. For best performance, please use the server that's geographically closest to you.' Below the note are 'Download configuration file' and 'Regenerate' buttons.

Step2:

Run the ip we get when we start the THM machine.

The screenshot shows the 'Target Machine Information' screen with a title of 'OWASP-Juice-Shop', a target IP address of '10.10.166.27', and an expiration of '0m 0s'. It includes buttons for '?', 'Add 1 hour', and 'Terminate'. Below this is a task list with one item: 'Task 1' (radio button) and 'Open for business!'. A note states: 'Within this room, we will look at OWASP's TOP 10 vulnerabilities in web applications. You will find these in all types of web applications. But for today we will be looking at OWASP's own creation, Juice Shop!' with a 'Start Machine' button and a yellow trophy icon.

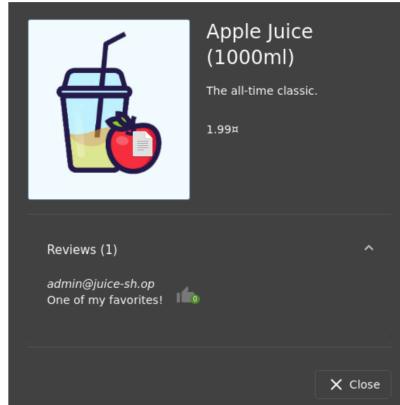
The screenshot shows a browser window displaying the 'OWASP Juice Shop' website at the URL '10.10.166.27'. The page header includes the OWASP logo and the text 'OWASP Juice Shop'. Below the header, there is a navigation bar with 'All Products' and several other menu items. The main content area is currently empty, showing a dark background.

Step3:

Then wait for the website to load for 2-5 minutes and then see the the Administrator's email

address in the 'Apple Juice' products review section.

The reviews show each user's email address. Which, by clicking on the Apple Juice product, shows us the Admin email!



admin@juice-sh.op

✓ Correct Answer



Apple Juice (1000ml)

The all-time classic.

1.99¤

Reviews (1)

admin@juice-sh.op
One of my favorites! 

 Close

Step4:

I have searched it using query parameter.

10.10.166.27/#/search?q=a

Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

OWASP Juice Shop

Search Results - a

Question #2: What parameter is used for searching?



Click on the magnifying glass in the top right of the application will pop out a search bar.



We can then input some text and by pressing **Enter** will search for the text which was just inputted.

Now pay attention to the URL which will now update with the text we just entered.



We can now see the search parameter after the **/#/search?** the letter **q**

q ✓ Correct Answer

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

Step5:

If we google “replicator” we will get the results indicating that it is from a TV show called Star

Replicator

Star Trek franchise element



First appearance *Star Trek: The Next Generation*

Created by Gene Roddenberry

Genre Science fiction

In-universe information

Type Matter converter

Function Synthesis of organic and inorganic materials via rearrangement of subatomic particles

Affiliation Starfleet

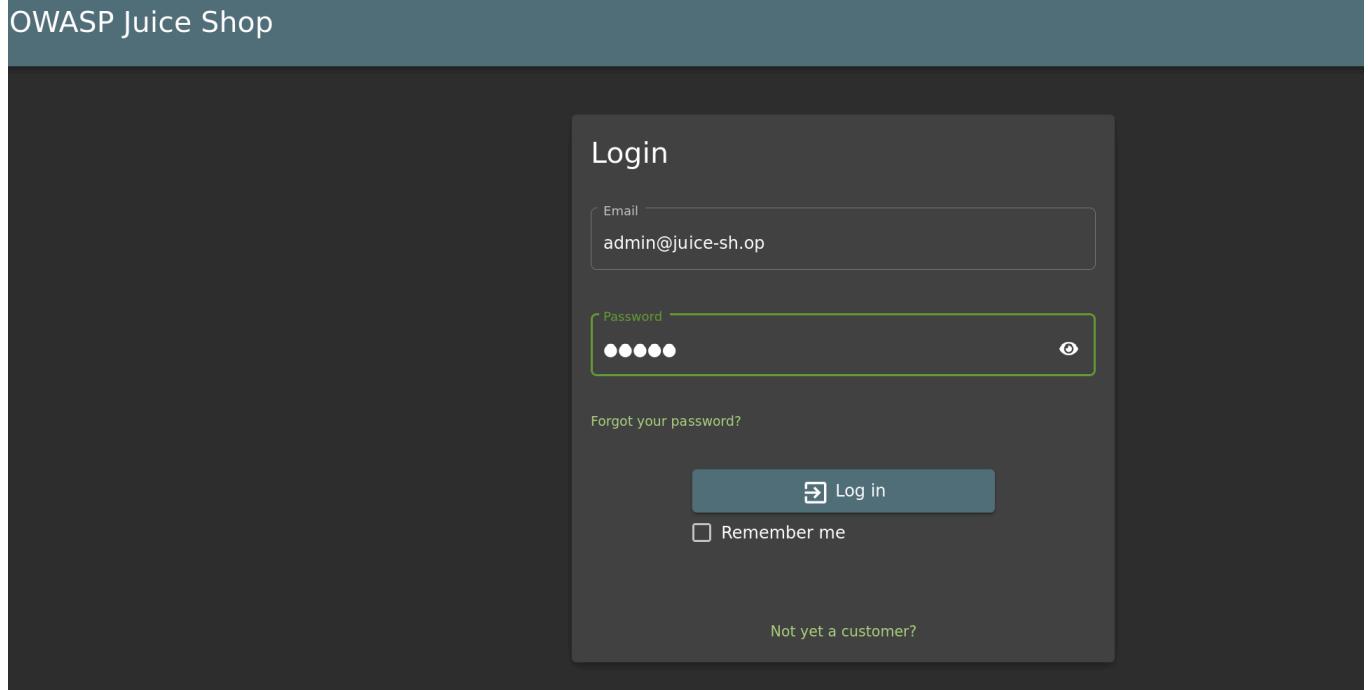
whether food, medicaments, or spare

Step6:

While using sql injection lets login to the admin account. But don't forget to turn off the Intercept

mode.

Try to login the website



Then in burpsuite see what is happening

```
04:04:53 24 J... HTTP → Request GET http://10.10.214.15/socket.io/?EIO=3&transport=polling&t=PUWz_FS
04:05:18 24 J... HTTP → Request GET http://10.10.214.15/socket.io/?EIO=3&transport=polling&t=PUW-5M6
04:05:19 24 J... HTTP → Request GET http://10.10.214.15/rest/user/whoami
04:05:19 24 J... HTTP → Request GET http://10.10.214.15/rest/user/whoami
04:05:19 24 J... HTTP → Request POST http://10.10.214.15/rest/user/login
```

Request

Pretty	Raw	Hex
5 Accept-Language: en-US,en;q=0.5		
6 Accept-Encoding: gzip, deflate, br		
7 Content-Type: application/json		
8 Content-Length: 46		
9 Origin: http://10.10.214.15		
.0 Connection: keep-alive		
.1 Referer: http://10.10.214.15/		
.2 Cookie: io=7rW-037sT50M3wCfAAAD; language=en; cookieconsent_status=dismiss		
.3 Priority: u=0		
.4		
.5 {		
"email": "admin@juice-sh.op",		
"password": "any"		
}		

It shows us what we have used to login, and then we change the email.

Step7:

when send the request to repeater and then on repeater we click the send button which will give me the token of the admin then the website says you have logged in successfully.

The screenshot shows the Burp Suite interface with a network request sent to `http://10.10.148.15`. The request is a POST to `/rest/user/login` with the following JSON payload:

```
POST /rest/user/login HTTP/1.1
Host: 10.10.148.15
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 41
Origin: http://10.10.148.15
Connection: keep-alive
Referer: http://10.10.148.15/
Cookie: io=kOpd3gKTHdSp3wAAAA; language=en
Priority: u=0
{
  "email": " or l=1-",
  "password": "test"
}
```

The response received is a 200 OK with the following headers:

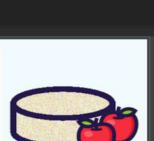
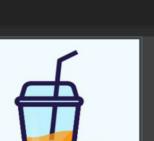
```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
Content-Type: application/json; charset=utf-8
Content-Length: 824
ETag: W/"338-7daEzPM28lxRjgyn0XcuBZrSo"
Vary: Accept-Encoding
Date: Wed, 25 Jun 2025 09:53:57
Connection: keep-alive
{
  "authentication": {
    "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJzdGFxdWMiOiJzdWnjZXNzIiwiZGFnSTGeypJcZGMswidXlcmhbWhLoIiLCIwbWpbC1EtMfbwLqqg1awNLXyNLoLmduWiwiicGFzc3dvcnQiOiwMTkyMDIzYtYdYmQ3Mz1MDUxmwNwNjUKjE4yJUMwCI sInJvbGwiOjhZGlibhsimRlbWq42Vra2wUjoiIwbfDzdxvZ2lSxaO1Iw-iJ AUwC4wLhvHnzlszulywdLijoiYXnzXZRxZl3B1YmxpyYspbwFnxZMvdxb82fc cy9kZWzhdwMslnN2z1sInRvHBTzXQj0i1LCjpcOfjd1zS16dhl1zSw1i3o1 YXRLEFOijo1MjAyNS0wN0yNSwA0dxNyNj0iNC43MjcgkzAw0jAw1iwdBKYxRlZEF OjioiMjAyNS0wN0yNSwA0dxNyNj0iNC43MjcgkzAw0jAw1iZ0vsZXRlZEF0ijpudw xsfSwaiWF0joxNzLwODQyMzUoLCLj1ehAlOjE3TA4NjAxNT9_GLCJUnCOTjDxAU7 cKD8w0OPrBr04dbwUjPwbd3dKeu08EWL2-tbTAkycgdAK967te7vwfZflbcj3GC oSSHvZJRdsdodTfLla192xAHK4AT_VjEvh-t19QFB-Ue1XoA13qFxP5b_cfVIZn2wqu CqlMmnM6s1ZBwB7mHcpY",
    "bid": 1,
    "umail": "admin@juice-shop.com"
  }
}
```

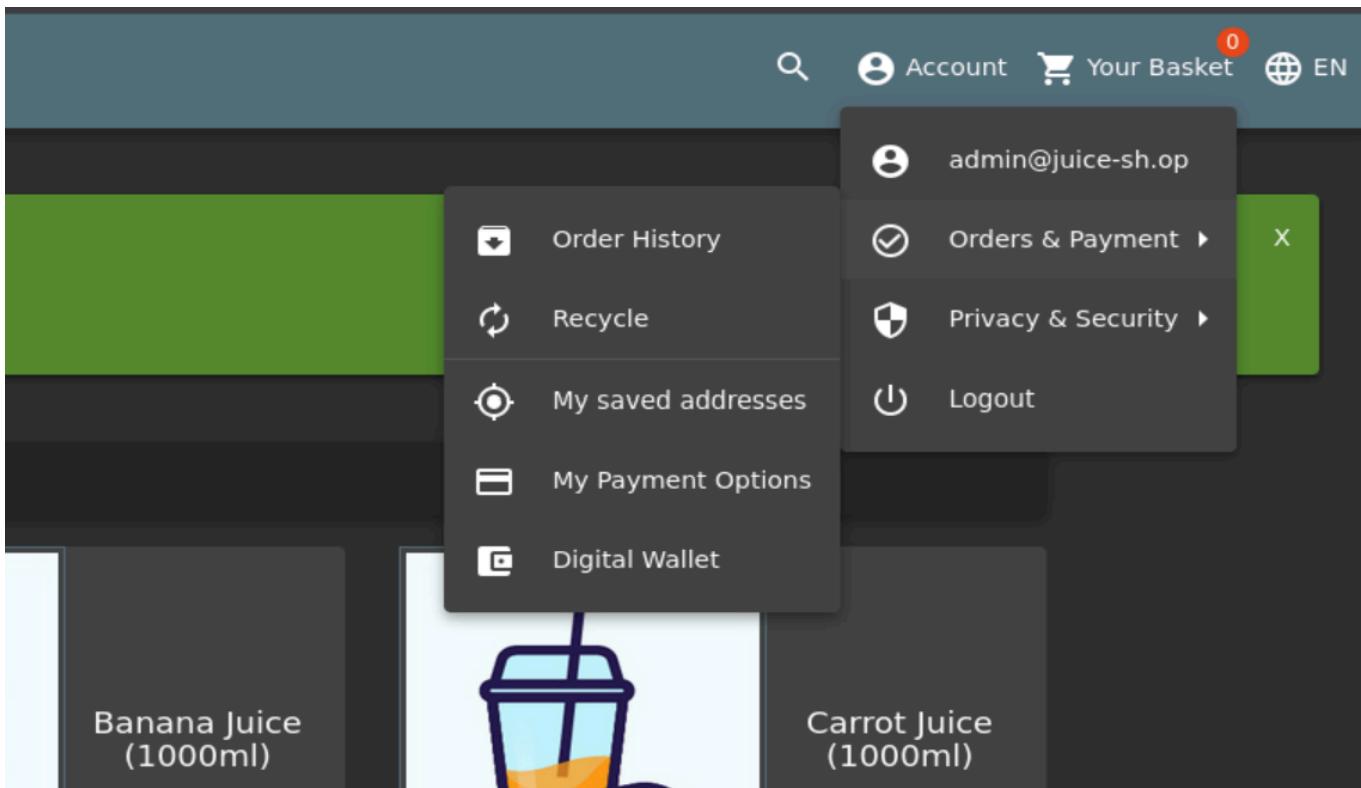
OWASP Juice Shop OWASP Juice Shop

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

32a5e0f21372bcc1000a6088b93b458e41f0e02a [Copy to clipboard](#)

All Products

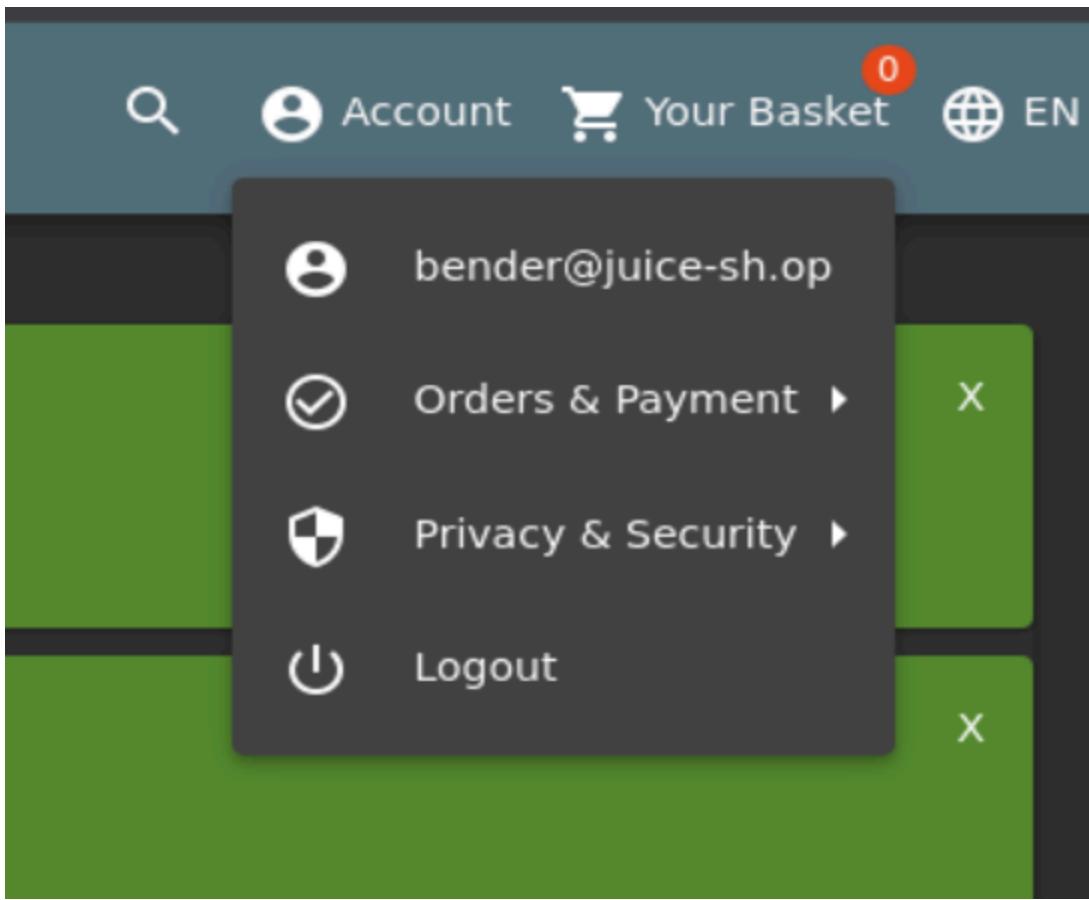
 Apple Juice (1000ml) 1.99¤ Add to Basket	 Apple Pomace 0.89¤ Add to Basket	 Banana Juice (1000ml) 1.99¤ Add to Basket	 Carrot Juice (1000ml) 2.99¤ Add to Basket
 Kiwi Juice (1000ml) 1.99¤ Add to Basket	 Lime Juice (1000ml) 1.99¤ Add to Basket	 Grapefruit Juice (1000ml) 1.99¤ Add to Basket	 Lemon Juice (1000ml) 1.99¤ Add to Basket



Step8:

Next we will try to login to another user.

A screenshot of the application showing a success message and product categories. At the top, there are two green notification bars. The first bar says "You successfully solved a challenge: Login Admin (Log in with the administrator's user account.)" with a copy link "32a5e0f21372bcc1000a6088b93b458e41f0e02a" and a "Copy to clipboard" button. The second bar says "You successfully solved a challenge: Login Bender (Log in with Bender's user account.)" with a copy link "fb364762a3c102b2db932069c0e6b78e738d4066" and a "Copy to clipboard" button. Below these is a dark grey header with the text "All Products". Underneath is a row of five small images representing different products: a blue drink, a yellow drink, a sandwich, a blue drink, and a blue drink.



So this shows us it have worked properly.

Step9:

lets find out who is intruding also find the amdin passwors.

POST /rest/user/login HTTP/1.1
Host: 10.10.221.230
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 47
Origin: http://10.10.221.230
Connection: keep-alive
Referer: http://10.10.221.230/
Cookie: iot=tct0xamatY97eSAAA; language=en; cookieconsent_status=dismiss; continueCode=Pwma6xDQa3kY5bEJRzoqLnyBwpdq1QdKeMNmr8P1lv4v9VjZ2g7QGg4Rzx
Priority: u=0
{"email": "admin@juice-sh.op", "password": "!\$*"}

Then click the clear button and clear the §

The screenshot shows the Burp Suite interface. The top menu bar includes Burp, Project, Intruder, Repeater, View, Help, and a temporary project name. The tabs at the top are Dashboard, Target, **Proxy**, **Intruder**, Repeater, Collaborator, Sequencer, Decoder, Comparer, Logger, Organizer, Extensions, and Learn. Below the tabs, there are 8, 10, 11, and a plus sign buttons. A search bar contains "Sniper attack". The Target field shows "http://10.10.221.230" with a checkbox for "Update Host header to match target". Buttons for "Add §", "Clear §", and "Auto §" are present. The main pane displays a POST request to /rest/user/Login with various headers and a JSON payload containing email and password fields. The right side features the "Payloads" panel with a "Payloads" icon, a note about highlighting request parts, and buttons for "Close", "Learn more", and "Don't show this again". The bottom status bar shows 0 highlights, 0 payload positions, and a length of 602.

Bruteforce the Administrator account's password!

For the payload, we will be using the best1050.txt from Seclists. (Which can be installed via:
apt-get install seclists)

```
(fidez㉿kali)-[~/Desktop/class]
$ sudo apt-get install seclists
[sudo] password for fidez:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  seclists
0 upgraded, 1 newly installed, 0 to remove and 1859 not upgraded.
Need to get 557 MB of archives.
After this operation, 1,970 MB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling/main amd64 seclists all 2025.2-0kali [557 MB]
```

- How to work with this

Send the Request to Intruder

In Burp, with the login request still visible in Proxy > Intercept, right-click it

Select "Send to Intruder"

Go to the Intruder tab

Click Positions

Set attack type to Sniper or Cluster Bomb depending on whether you're bruteforcing only password or both username and password

Highlight the value of the password parameter and click "Add §"

Load the Wordlist File

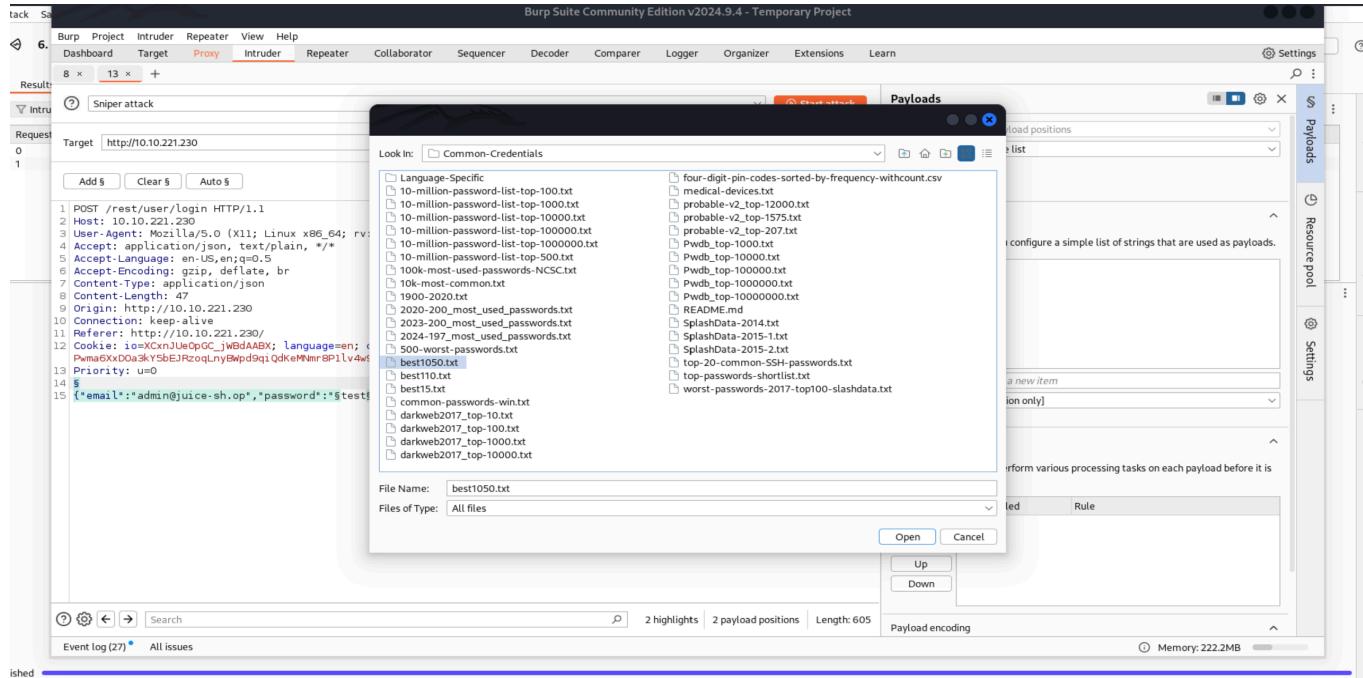
Go to the *Payloads* tab

In Payload Set 1, under *Payload Options*, click *Load*

Navigate to the wordlist:

/usr/share/seclists/Passwords/Common-Credentials/best1050.txt

Select and open it — the payloads should load into the list



Then send an attack and wait for the request to be 200 which is OK. Then you have found the password.

7. Intruder attack of http://10.10.235.104

Attack Save

7. Intruder attack of http://10.10.235.104

Attack Save ⚡ ⓘ ⓘ

Results Positions

Intruder attack results filter: Showing all items

Request	Payload	Status code	Response rec...	Error	Timeout	Length	Comment
111	access14	401	207			367	
112	account	401	210			367	
113	action	401	215			367	
114	admin	401	208			367	
115	admin1	401	203			367	
116	admin12	401	204			367	
117	admin123	200	264			1164	
118	adminadmin	401	212			367	
119	administrator	401	216			367	
120	adriana	401	213			367	

Request Response

Pretty Raw Hex

```

1 POST /rest/user/login HTTP/1.1
2 Host: 10.10.235.104
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Content-Type: application/json
8 Content-Length: 51
9 Origin: http://10.10.235.104
10 Connection: keep-alive
11 Referer: http://10.10.235.104/
12 Cookie: io=$LJMhqxZ2H4efcMUAAAO; language=en; cookieconsent_status=dismiss
13 Priority: u=0
14
15 {
    "email": "admin@juice-sh.op",
    "password": "admin123"
}

```

Search 0 highlights

122 of 1049

So when you find the password the flag will be given Which is the answer.

You successfully solved a challenge: Password Strength (Log in with the administrator's user credentials without previously changing them or applying SQL Injection.) X

c2110d06dc6f81c67cd8099ff0ba601241f1ac0e [Copy to clipboard](#)

You successfully solved a challenge: Login Admin (Log in with the administrator's user account.) X

32a5e0f21372bcc1000a6088b93b458e41f0e02a [Copy to clipboard](#)

The screenshot shows a dark-themed login page with a central light gray box. The title 'Login' is at the top. Below it is an 'Email' input field containing 'admin@juice-sh.op'. Below that is a 'Password' input field containing 'admin123'. To the right of the password field is a small icon. Below the fields is a link 'Forgot your password?'. At the bottom are two buttons: a blue 'Log in' button with a key icon and a white 'Remember me' checkbox. At the very bottom of the light gray box is a link 'Not yet a customer?'.

Reset Jim's password!

Believe it or not, the reset password mechanism can also be exploited! When inputted into the email field in the Forgot Password page, Jim's security question is set to "Your eldest siblings middle name?". In Task 2, we found that Jim might have something to do with Star Trek. Googling "Jim Star Trek" gives us a wiki page for James T. Kirk from Star Trek.

s

Starfleet

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

Children

David Marcus

Origin

Iowa, United States, Earth

h

Therefore his brothers middle name is *Samuel*. Then now lets try it on forget to get the Flag.

The screenshot shows a browser window for the OWASP Juice Shop application at the URL 10.10.230.242/#/forgot-password. The page title is "Forgot Password". A green success message box at the top states: "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.)". Below this, a button labeled "Copy to clipboard" is visible. The main form fields are: "Email" (placeholder "jim@kali.local"), "Security Question" (placeholder "What is your mother's maiden name?"), "New Password" (placeholder "password123", note: "Password must be 5-20 characters long. 0/20"), "Repeat New Password" (placeholder "password123", note: "0/20"), and a "Show password advice" toggle switch. At the bottom is a "Change" button. In the bottom right corner of the page, there is a cookie consent banner: "This website uses fruit cookies to ensure you get the juiciest tracking experience. [But me wait!](#)" with a "Me want it!" button.

Step10:

Now lets do **Task 5 : AH! Don't look!**.

A web application should store and transmit sensitive data safely and securely. But in some cases, the developer may not correctly protect their sensitive data, making it vulnerable.

Most of the time, data protection is not applied consistently across the web application making certain pages accessible to the public. Other times information is leaked to the public without the knowledge of the developer, making the web application vulnerable to an attack.

Access the Confidential Document!

The screenshot shows the OWASP Juice Shop website with the URL 0.10.230.242/ftp/legal.md in the address bar. The page title is "About Us". Under "Corporate History & Policy", there is a large block of Lorem ipsum placeholder text. Below it, under "Customer Feedback", are two pixelated profile pictures with arrows for navigation. The left profile has the text "ecommended! (**@juice-sh.op)" and the right profile has "Great shop! Awesome service! (**@juice-sh.op) (★★★★★)". At the bottom, there's a "Follow us on Social Media" section with links for Twitter, Facebook, Slack, Reddit, and a Press Kit. The top right corner shows account and search icons.

OWASP Juice Shop

About Us

Corporate History & Policy

... (large block of placeholder text)

Customer Feedback

< >

ecommended! (**@juice-sh.op)

Great shop! Awesome service! (**@juice-sh.op)
(★★★★★)

Follow us on Social Media

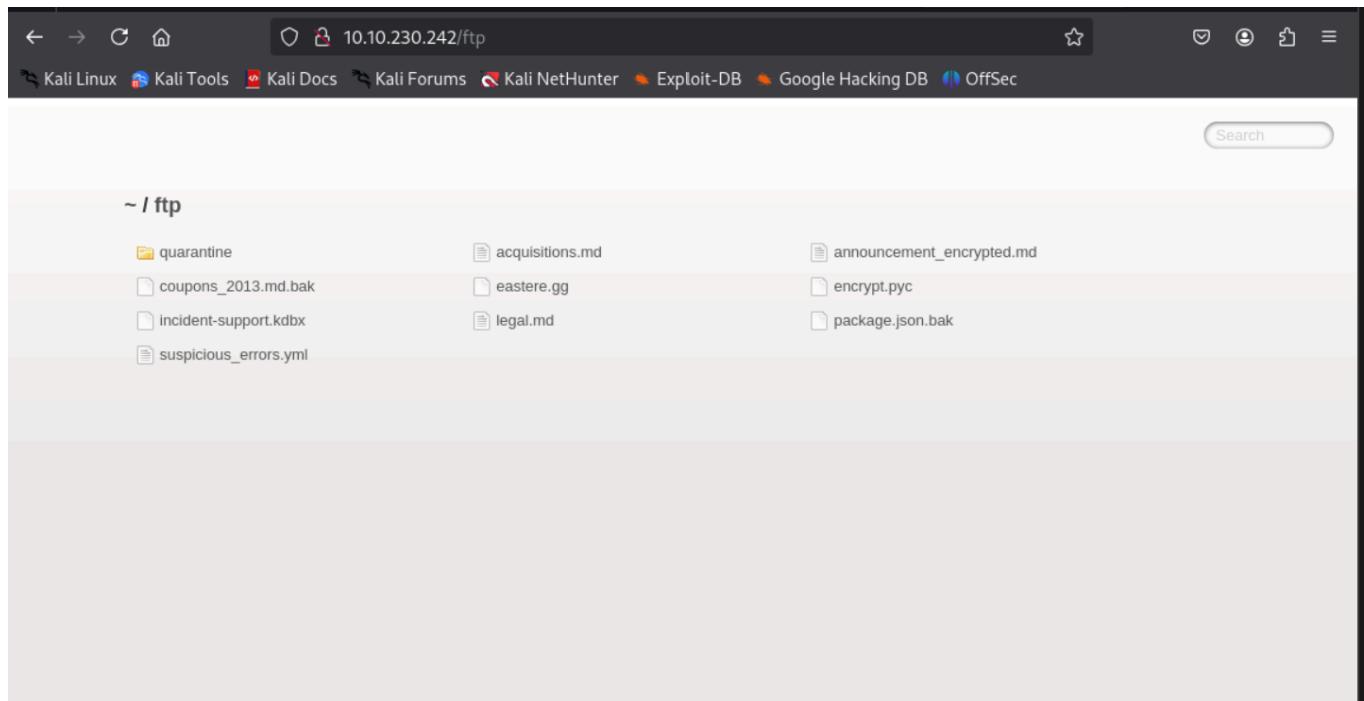
Twitter Facebook Slack Reddit Press Kit

0.10.230.242/ftp/legal.md

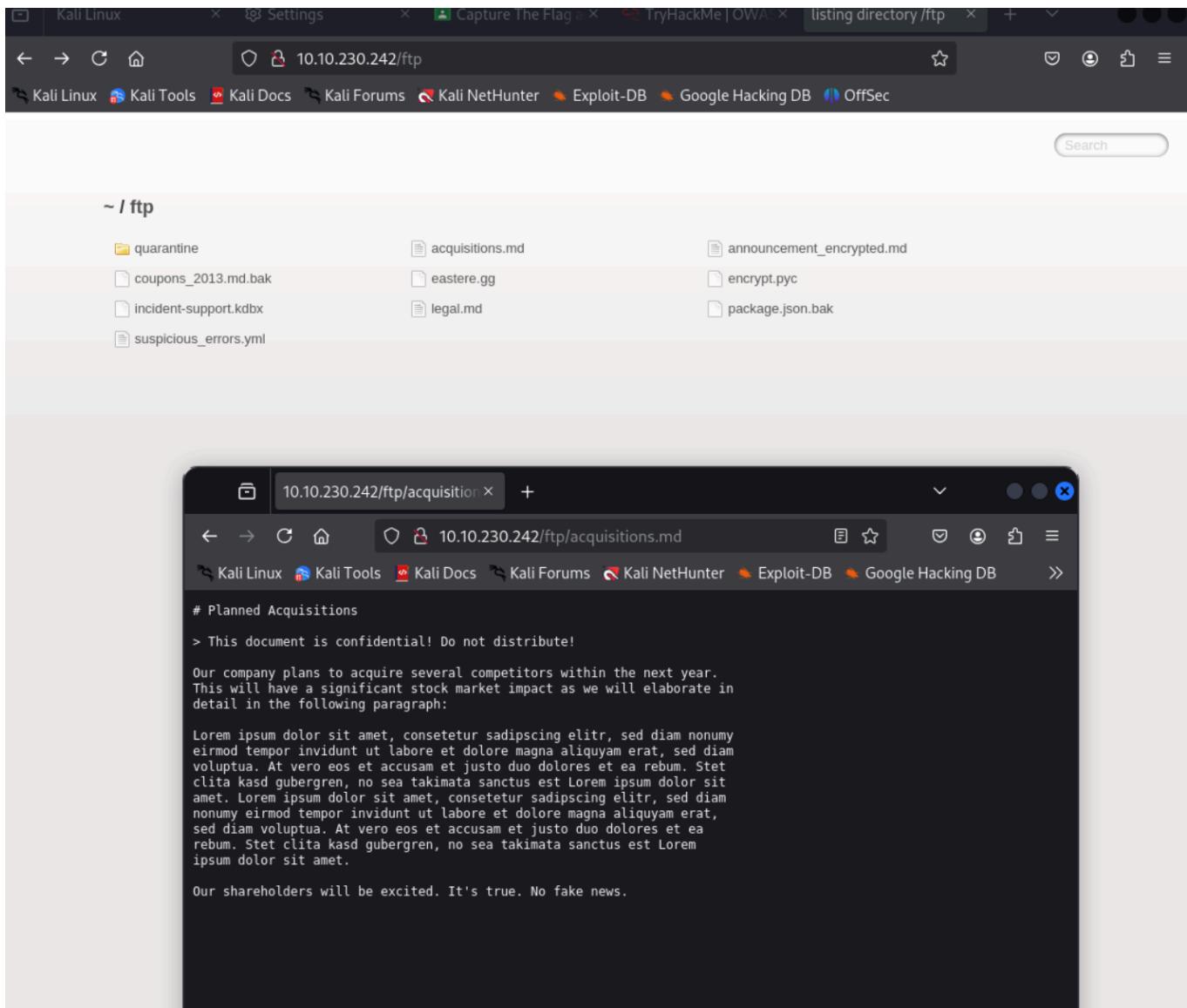
We can see /ftp/legal.md

We will download the acquisitions.md and save it. It looks like there are other files of interest here as well.

After downloading it, navigate to the home page to receive the flag!



Then download the *acquisitions.md*.



Then go the home page of the website to get the flag.

You successfully solved a challenge: Confidential Document (Access a confidential document.)

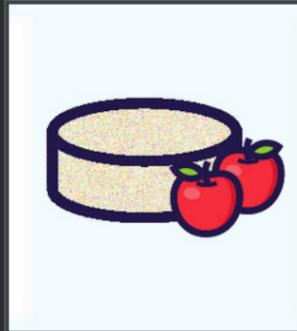
X

edf9281222395a1c5fee9b89e32175f1ccf50c5b

All Products



Apple Juice
(1000ml)
1.99¤



Apple Pomace
0.89¤

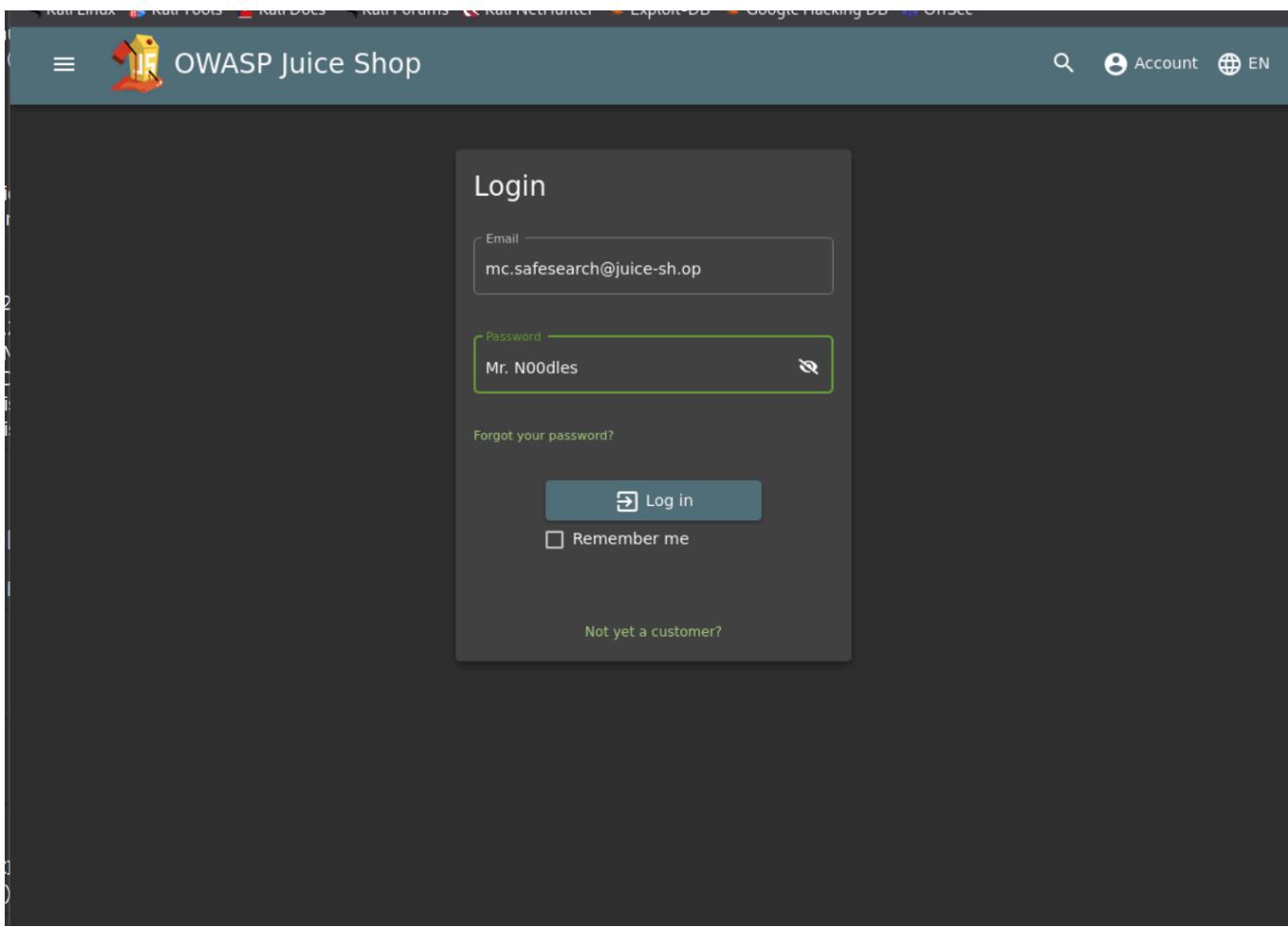
[Click for more information](#)

Log into MC SafeSearch's account!

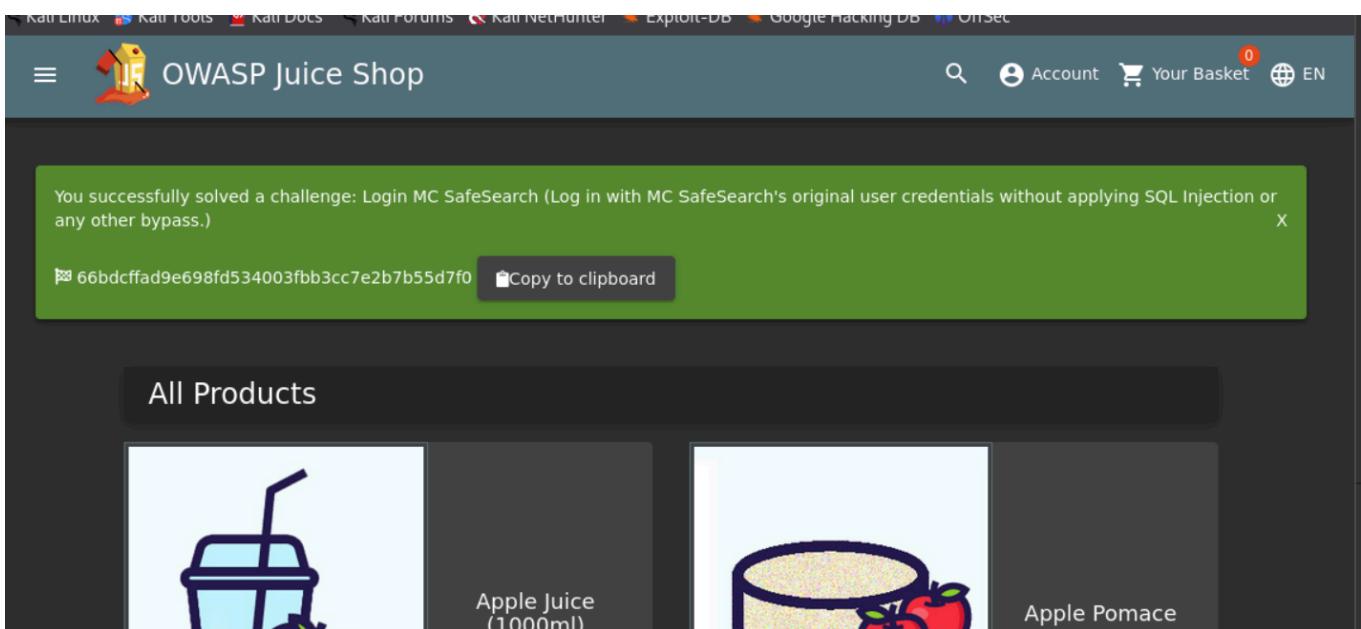
After watching the video there are certain parts of the song that stand out.

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**"



Then after logging in we will see the flag in the home page.



Download the Backup file!

We will now go back to the <http://10.10.87.89/ftp/> folder and try to download package.json.bak. But it seems we are met with a 403 which says that only .md and .pdf files can be downloaded.

The screenshot shows a web browser window with the URL `10.10.87.89/ftp/package.json.bak`. The page title is "OWASP Juice Shop (Express ^4.17.1)". Below the title, a red box highlights the error message: "403 Error: Only .md and .pdf files are allowed!". A stack trace is visible below the error message.

```
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)
```

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

Note: as we can download it using the url, we will need to 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!

```
10.10.206.192/ftp/package.json.bak%2500.md
```

Why does this work?

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.

The screenshot shows a browser window with the URL `10.10.87.89/ftp/package.json.bak%2500.md`. The page title is "OWASP Juice Shop (Express ^4.17.1)". A modal dialog is displayed with the message "403 Error: Only .md and .pdf files are allowed!" and a stack trace:

```
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)
```

Then go to the websites home page to see the flag.

The screenshot shows the OWASP Juice Shop Express homepage. At the top, two green notifications are displayed:

- You successfully solved a challenge: Error Handling (Provoked an error that is neither very gracefully nor consistently handled.)
- You successfully solved a challenge: Forgotten Developer Backup (Accessed a developer's forgotten backup file.)

Below the notifications, the "All Products" section is shown with two items:

Product	Description	Price
Apple Juice (1000ml)	A glass of juice next to a red apple.	1.99¤
Apple Pomace	A container of pomace next to two red apples.	0.89¤

Step11:

Now lets do **Task 6 : Who's flying this thing?**

Modern-day systems will allow for multiple users to have access to different pages. Administrators most commonly use an administration page to edit, add and remove different elements of a website. You might use these when you are building a website with programs such as Weebly or Wix.

When Broken Access Control exploits or bugs are found, it will be categorised into one of **two types**:

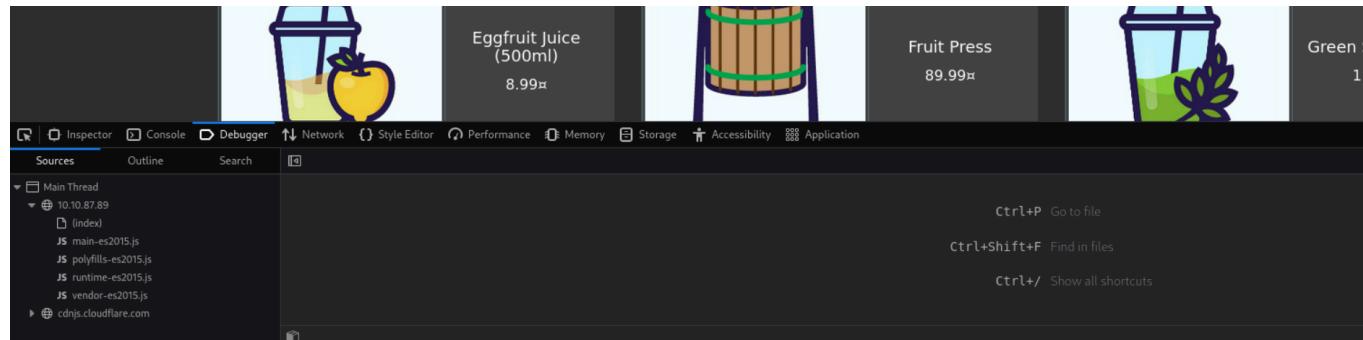
Horizontal Privilege Escalation	Occurs when a user can perform an action or access data of another user with the same level of permissions.
Vertical Privilege Escalation	Occurs when a user can perform an action or access data of another user with a higher level of permissions.

Access the administration page!

We go to inspect .

We are then going to refresh the page and look for a javascript file for main-es2015.js

We will then go to that page at: <http://10.10.87.89/main-es2015.js>



To get this into a format we can read, click the { } button at the bottom

Now search for the term "admin"

You will come across a couple of different words containing "admin" but the one we are looking for is "path: administration"

```

22255     appname: t
22256   }
22257 },
22258   Xs = [
22259     {
22260       path: 'administration',
22261       component: Xi,
22262       canActivate: [
22263         ...
22264       ],
22265     }
22266   ];

```

admin

No source map found

This hints towards a page called "`/#/administration`" as can be seen by the **about** path a couple lines below, but going there while not logged in doesn't work.

As this is an Administrator page, it makes sense that we need to be in the **Admin account** in order to view it.

A good way to stop users from accessing this is to only load parts of the application that need to be used by them. This stops sensitive information such as an admin page from been leaked or viewed.

So we first have to login to access the admin(login as admin user). Then navigate to Administration path.

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`

Time	Type	Direction	Method	URL	Status code	Length
07:29:31...	HTTP	→ Request	GET	http://10.10.87.89/rest/user/whoami		
07:29:31...	HTTP	→ Request	GET	http://10.10.87.89/rest/basket/1		

Request

Pretty Raw Hex

```

1 GET /rest/basket/1 HTTP/1.1
2 Host: 10.10.87.89
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: application/json, text/plain, */*

```

Inspector

Request attributes 2

Request query parameters 0

Now, we are going to change the number **1** after /basket/ to **2**

```
GET /rest/basket/2 HTTP/1.1
```

It will now show you the basket of UserID 2. You can do this for other UserIDs as well, provided that they have one!

Then in burpsuite after changing to 2 click *forward* so we can get the flag in our web page.

You successfully solved a challenge: View Basket (View another user's shopping basket.)
41b997a36cc33fbe4f0ba018474e19ae5ce52121 Copy to clipboard

Your Basket (admin@juice-sh.op)

Item	Quantity	Price
Apple Juice (1000ml)	2	1.99¤
Orange Juice (1000ml)	3	2.99¤
Eggfruit Juice (500ml)	1	8.99¤

Remove all 5-star reviews!

Navigate to the <http://10.10.87.89/#/administration> page again and click the bin icon next to the review with 5 stars!

Customer Feedback

1	I love this shop! Best products in town! Highly recommended! (**@juice-sh.op)	★★★★★	trash
2	Great shop! Awesome service! (**@juice-sh.op)	★★★★★	trash
3	Nothing useful available here! (**der@juice-sh.op)	★★★★★	trash
	Incompetent customer support! Can't even upload photo of broken purchase!...	★★★★★	trash
	This is the store for awesome stuff of all kinds! (anonymous)	★★★★★	trash
	Never gonna buy anywhere else from now on! Thanks for the great service! (anonymous)	★★★★★	trash
	Keep up the good work! (anonymous)	★★★★★	trash

Then when we remove it we get the Flag.

You successfully solved a challenge: Five-Star Feedback (Get rid of all 5-star customer feedback.)

50c97bcc0b895e446d61c83a21df371ac226ef | Copy to clipboard

Administration		Customer Feedback
Registered Users		
1	admin@juice-sh.op	2 Great shop! Awesome service! (**@juice-sh.op)
2	jim@juice-sh.op	3 Nothing useful available here! (**der@juice-sh.op)
3	bender@juice-sh.op	Incompetent customer support! Can't even upload photo of broken purchase!...
4	bjorn.kimminich@gmail.com	This is the store for awesome stuff of all kinds! (anonymous)

Step12:

Now lets do **Task 7 : Where did that come from?**

XSS or Cross-site scripting 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.

There are three major types of XSS attacks:

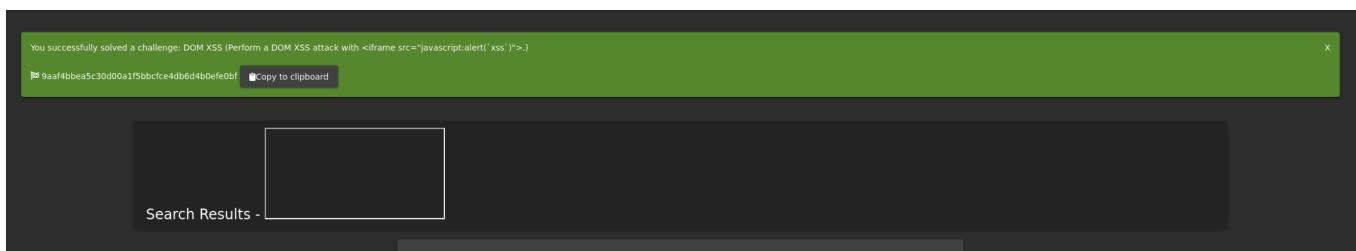
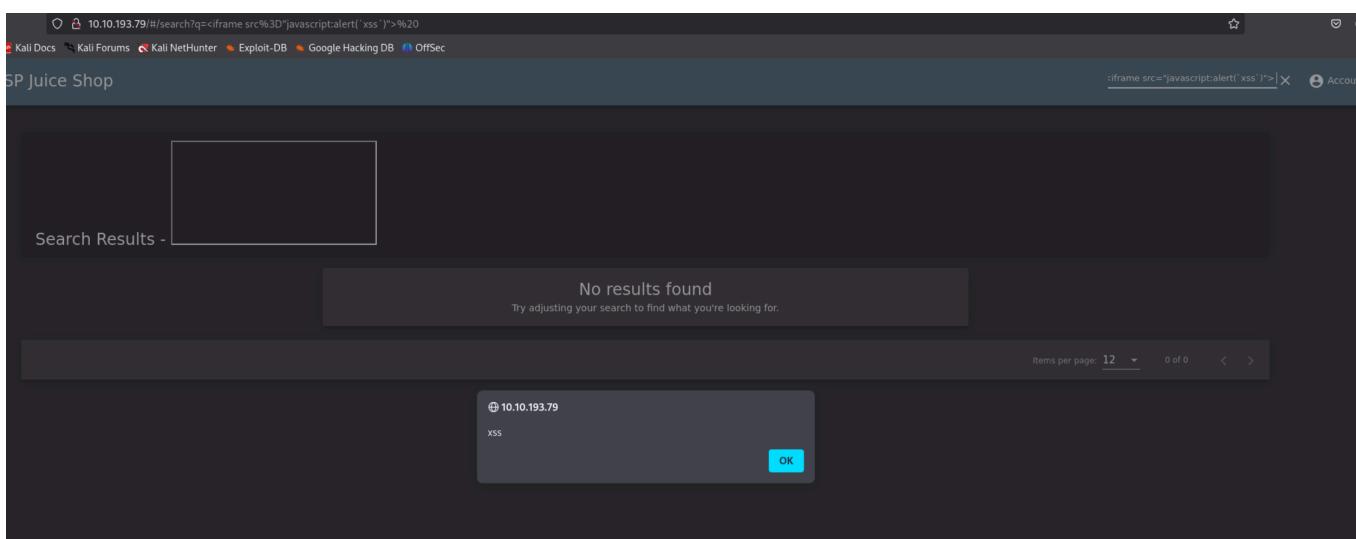
DOM (Special)	DOM XSS (<i>Document Object Model-based Cross-site Scripting</i>) uses the HTML environment to execute malicious javascript. This type of attack commonly uses the <code>HTML</code> tag.
Persistent (Server-side)	Persistent XSS is javascript that is run when the server loads the page containing it. These can occur when the server does not sanitise 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 sanitise search data.

Perform a DOM XSS!



We will be using the `iframe` element with a javascript alert tag:

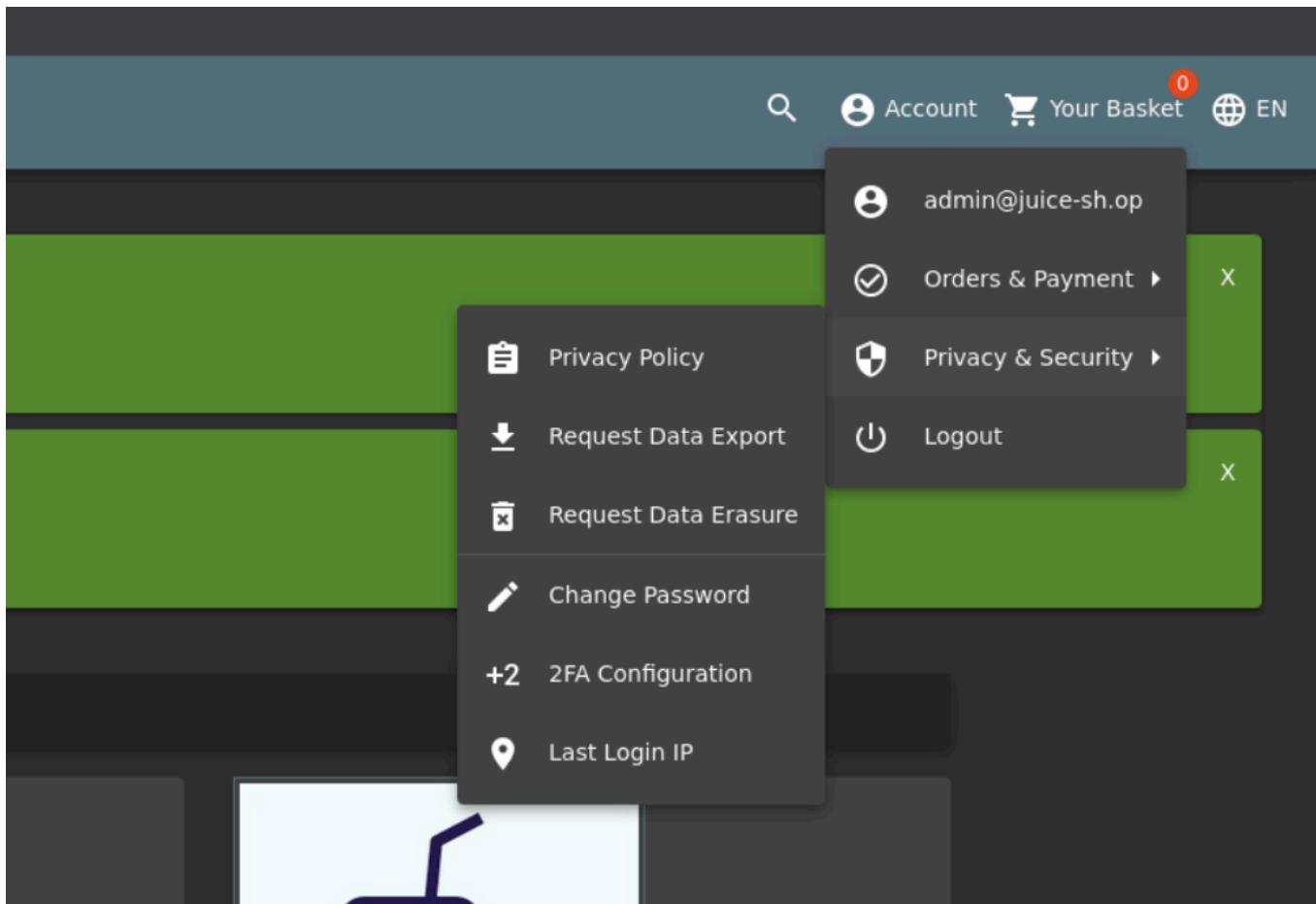
Inputting this into the **search bar** will trigger the alert.



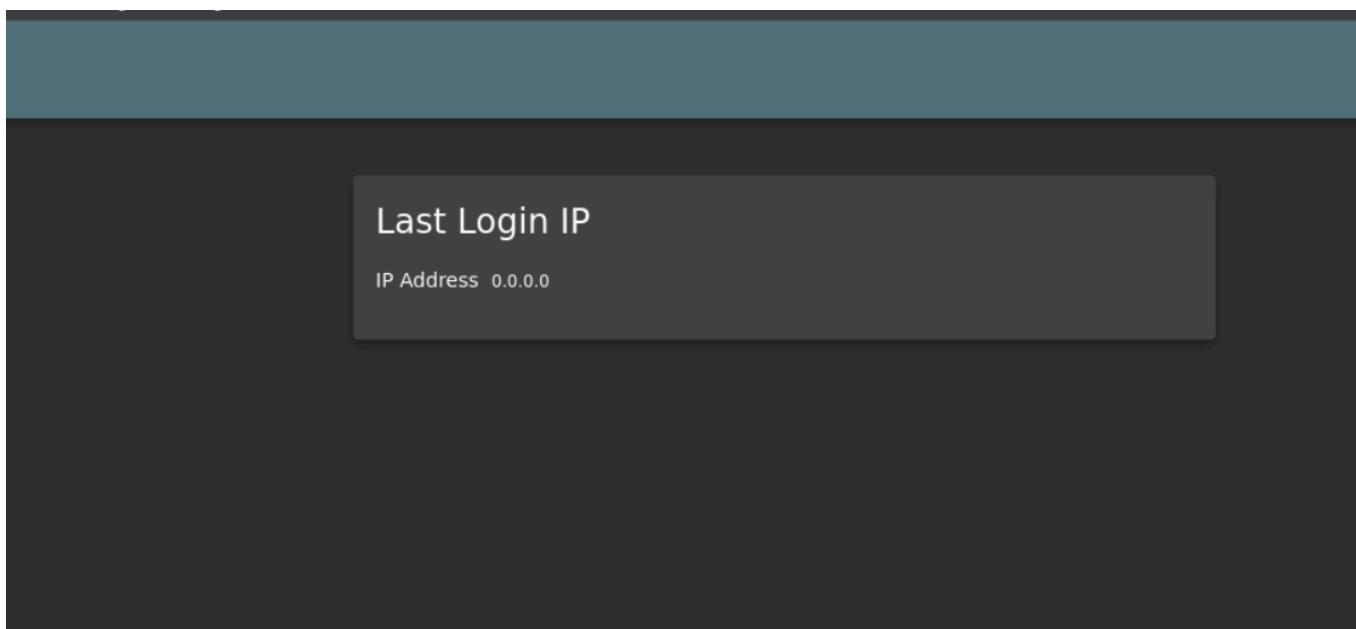
Perform a persistent XSS!

First, login to the **admin** account.

We are going to navigate to the "**Last Login IP**" page for this attack.



It should say the last IP Address is 0.0.0.0 or 10.x.x.x



As it logs the 'last' login IP we will now logout so that it logs the 'new' IP.

Intercept is on

Make sure that Burp **intercept is on**, so it will catch the logout request.

We will then head over to the Headers tab where we will add a new header:

<i>True-Client-IP</i>	