OWASP Juice Shop

OWASP Juice Shop

This room uses the Juice Shop vulnerable web application to learn how to identify and exploit common web application

I did no writeup since the challenge/room is a writeup in itself

[Task 1] Open for business!

Within this room, we will look at OWASP's TOP 10 vulnerabilities in web applications. You will find these in all types in all types of web applications. But for today we will be looking at OWASP's own creation, Juice Shop!

We will be using Burp Suite, so if you haven't already got it set up, here is a link to the 'Burp Suite' room.
In addition, its highly recommend to check out the 'Web Fundamentals' room.
Juice Shop is a large application so we will not be covering every topic from the top 10.

We will, however, cover the following topics which we recommend you take a look at as you progress through this room.
<-->
Injection
Broken Authentication
Sensitive Data Exposure
Broken Access Control
Cross-Site Scripting XSS
<-->
Credits to OWASP and Bjorn Kimminich

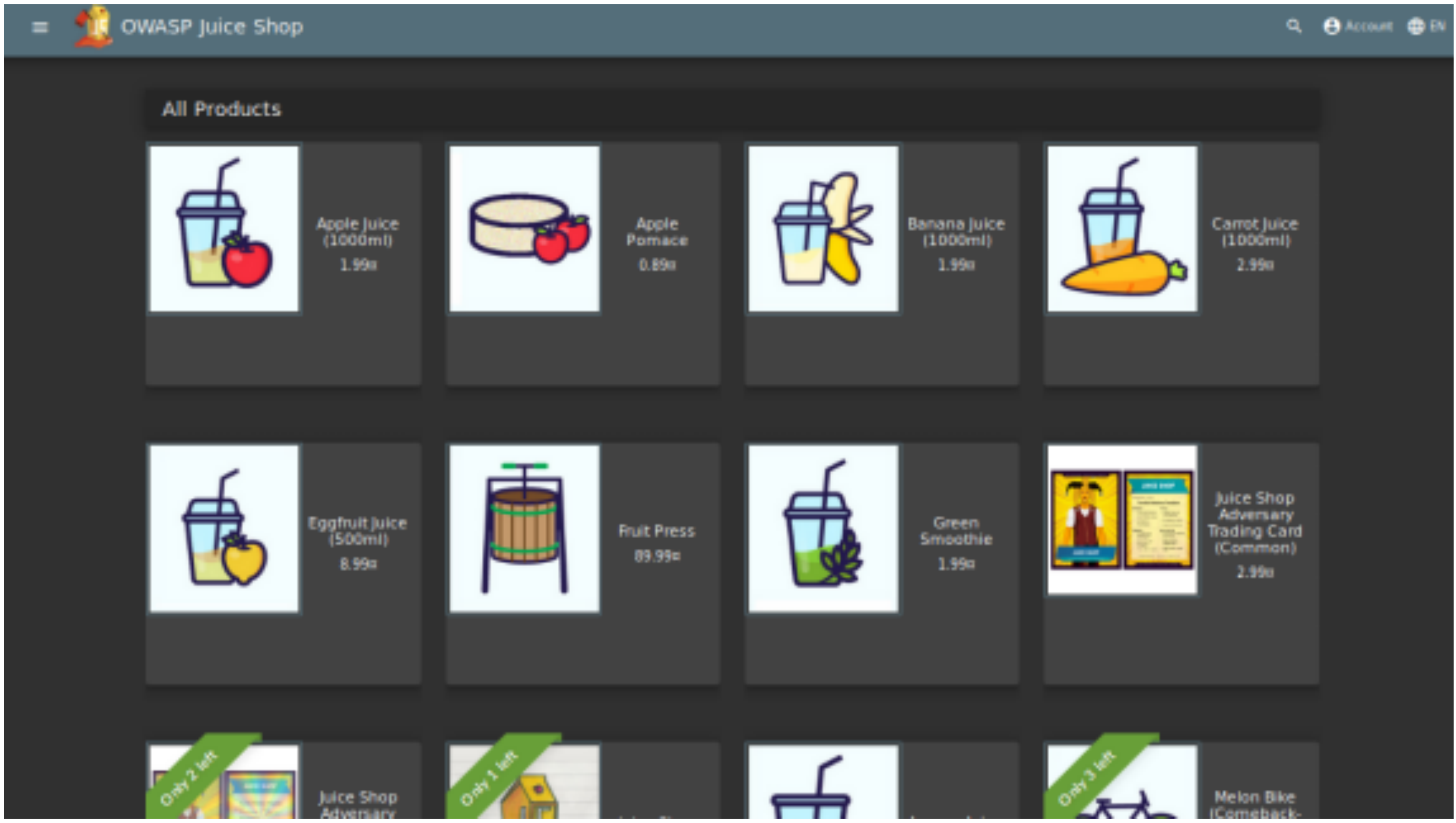
#1
Deploy the VM attached to this task to get started! You can access this machine by using your browser-based machine, or if you're connected through OpenVPN

No answer needed

#2
Once the machine has loaded, access it by copying and pasting its IP into your browser; if you're using the browser-based machine, paste the machines IP into a browser on that machine

No answer needed

[Task 2] Let's go on an adventure!



Before we get into the actual hacking part, it's good to have a look around. In Burp, set the Intercept mode to off and then browse around the site. This allows Burp to log different requests from the server that may be helpful later.
This is called walking through the application, which is also a form of reconnaissance!

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

admin@juice-sh.op

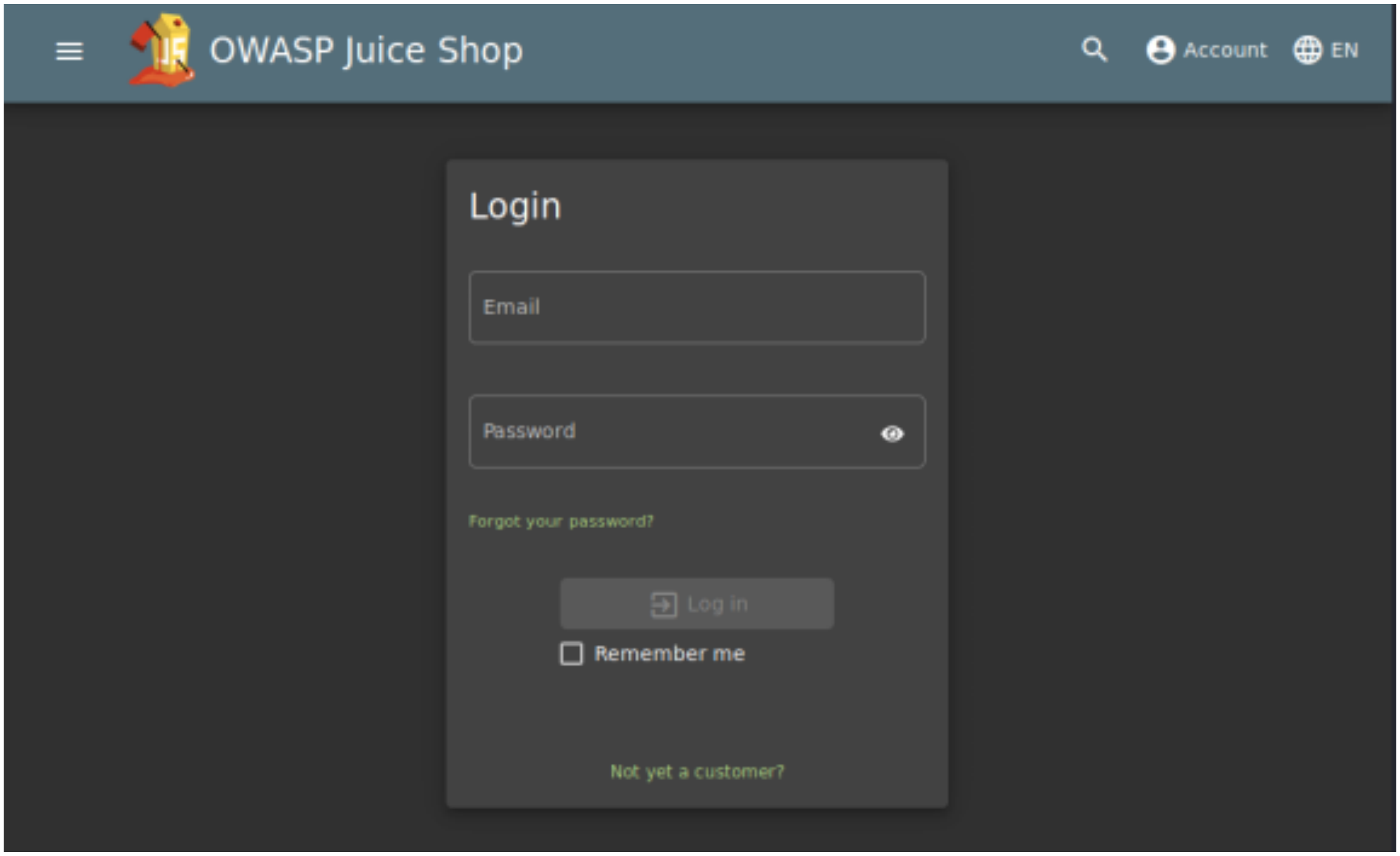
#2
What paramater is used for searching?

q

#3
What show does Jim reference in his review?

star trek

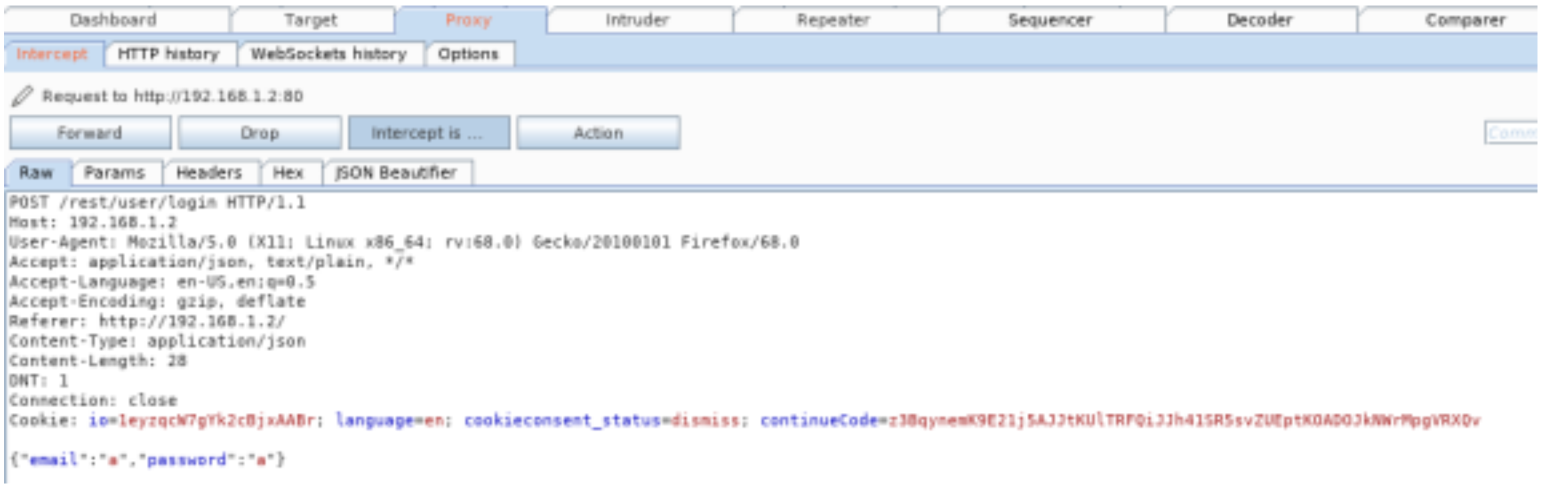
[Task 3] Inject the juice



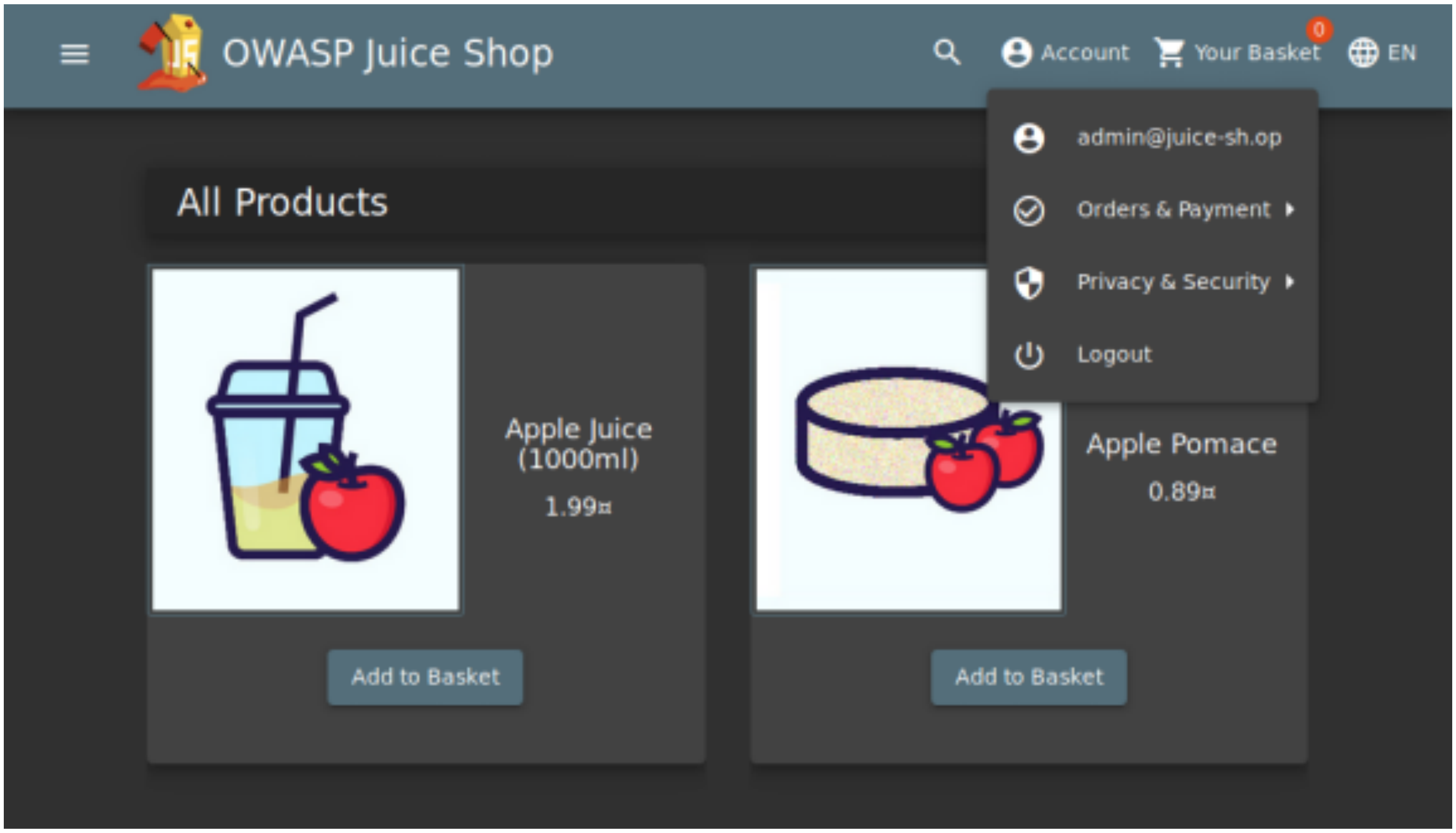
This task will be focusing on injection vulnerabilities. Injection vulnerabilities are quite dangerous to a company as they can potentially cause downtime and/or loss of data. Identifying injection points within a web application is usually quite simple, as most of them will return an error. There are many types of injection attacks, some of them are:

- **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.**
- **Command Injection - Command Injection is when web applications take input or user-controlled data and run them as system commands. An attacker may tamper with this data to execute their own system commands. This can be seen in applications that perform misconfigured ping tests.**
- **Email Injection - Email injection is a security vulnerability that allows malicious users to send email messages without prior authorization by the email server. These occur when the attacker adds extra data to fields, which are not interpreted by the server correctly.**

But in our case, we will be using SQL Injection.
For more information: Injection



```
POST /rest/user/login HTTP/1.1
Host: 192.168.1.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.2/
Content-Type: application/json
Content-Length: 28
DNT: 1
Connection: close
Cookie: io=1eyzqcW7gYk2cBjxAABr; language=en; cookieconsent_status=dismiss; continueCode=z3BqynemK9E21j5AJJtKUlTRFQiJJh41SR5svZUEptKOADOJkNWrMpgVRXQv

{"email":"a","password":"a"}
```

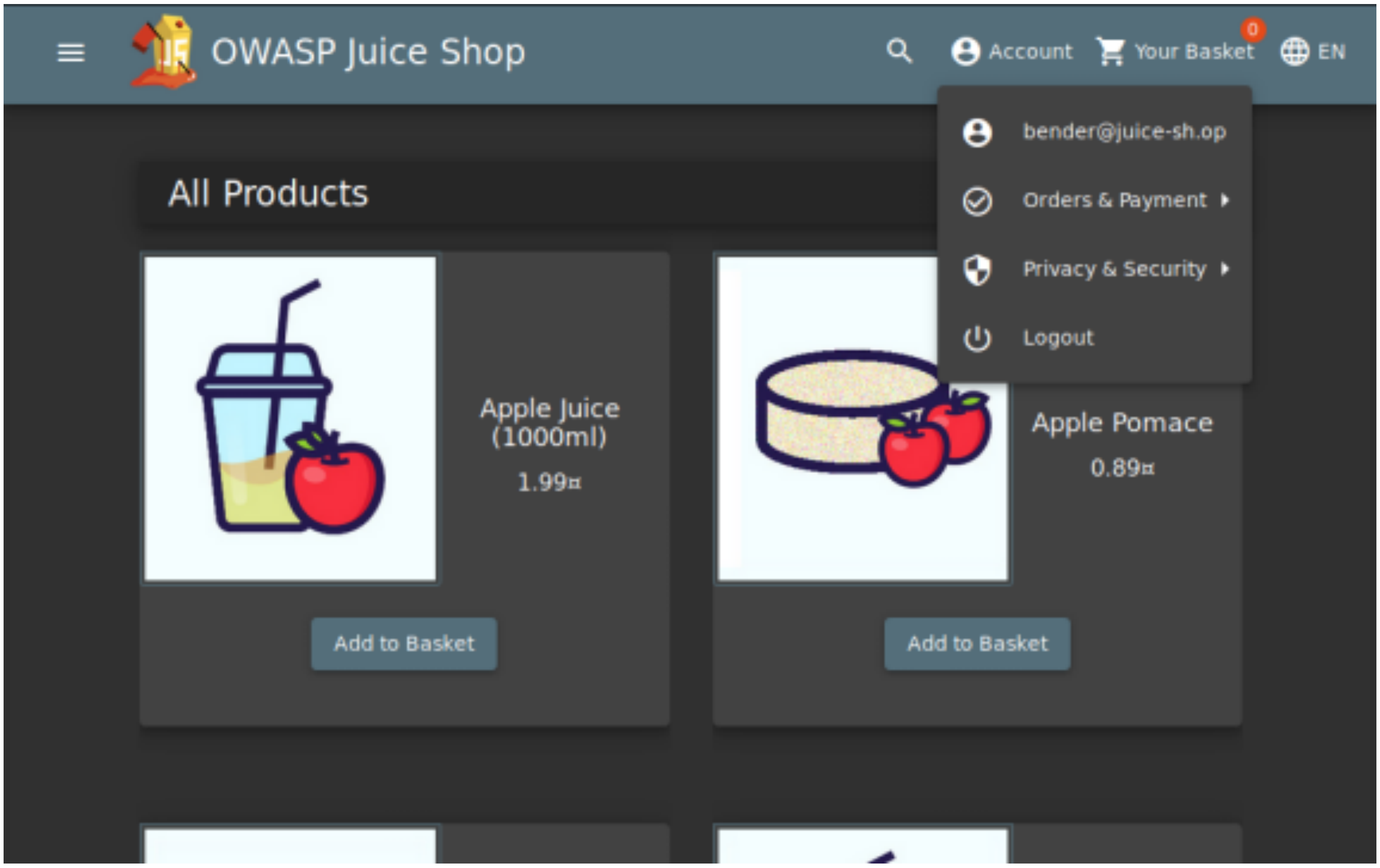


#1

Question #1: Log into the administrator account!
After we navigate to the login page, enter some data into the email and password fields. Before clicking submit make sure Intercept mode is on. This will allow us to see the data been sent to the server!

We will now change the email to: ' or 1=1-- and forward it to the server.
Why does this work?
1. The character ' will close the brackets in the SQL query
2. '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. Thus it will tell the server that the email is valid, and log us into user id 0, which happens to be the administrator account.
3. The -- character is used in SQL to comment out data, any restrictions on the login will no longer work as they are interpreted as a comment. This is like the # and // comment in python and javascript respectively.

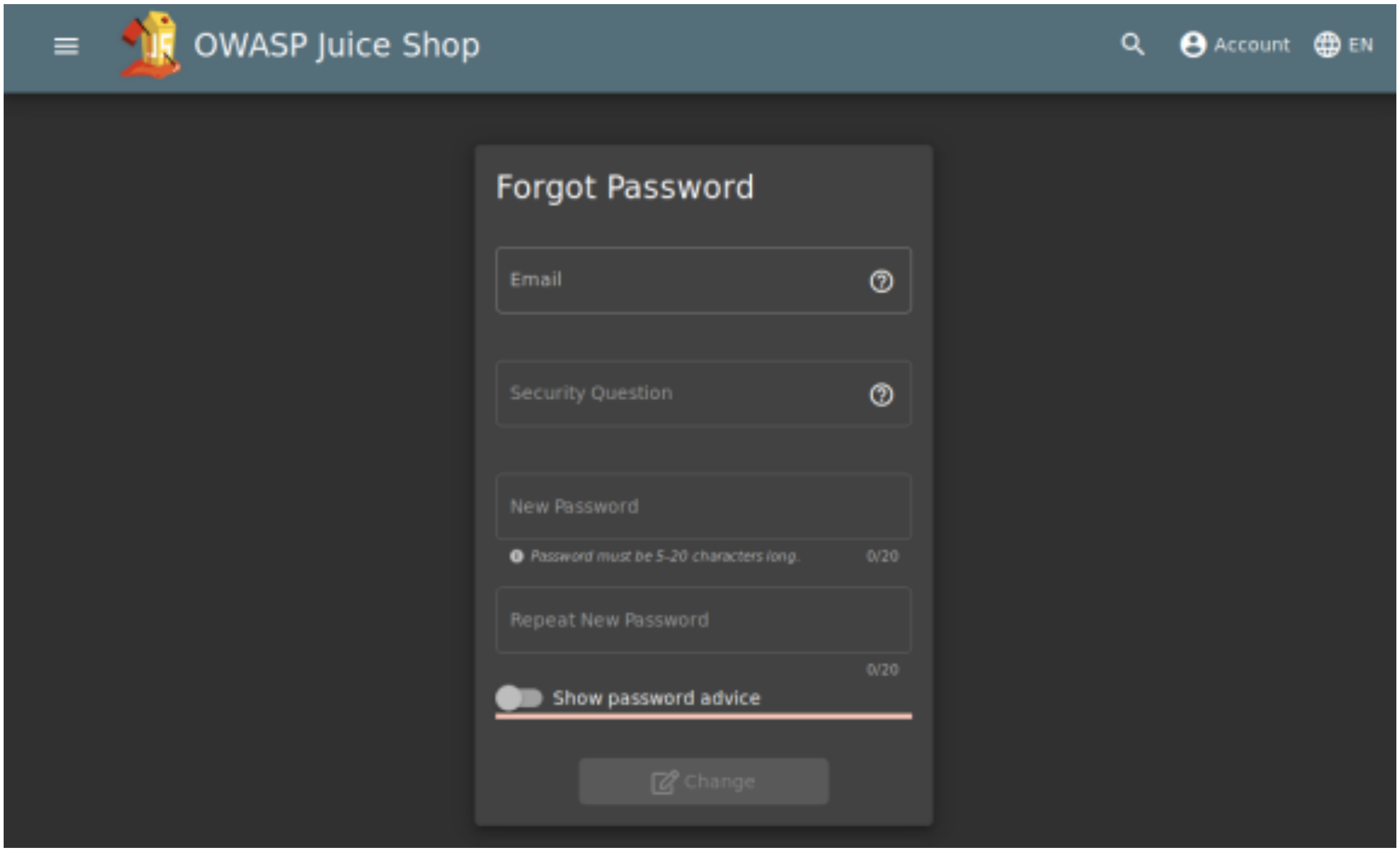
32a5e0f21372bcc1000a6088b93b458e41f0e02a



#2
Question #2: Log into the Bender account! Similar to what we did in Question #1, we will now log into Bender's account! Capture the login request again, but this time we will put: bender@juice-sh.op'-- as the email. Now, forward that to the server! But why don't we put the 1=1? Well, as the email address is valid (which will return true), we do not need to force it to be true.

fb364762a3c102b2db932069c0e6b78e738d4066

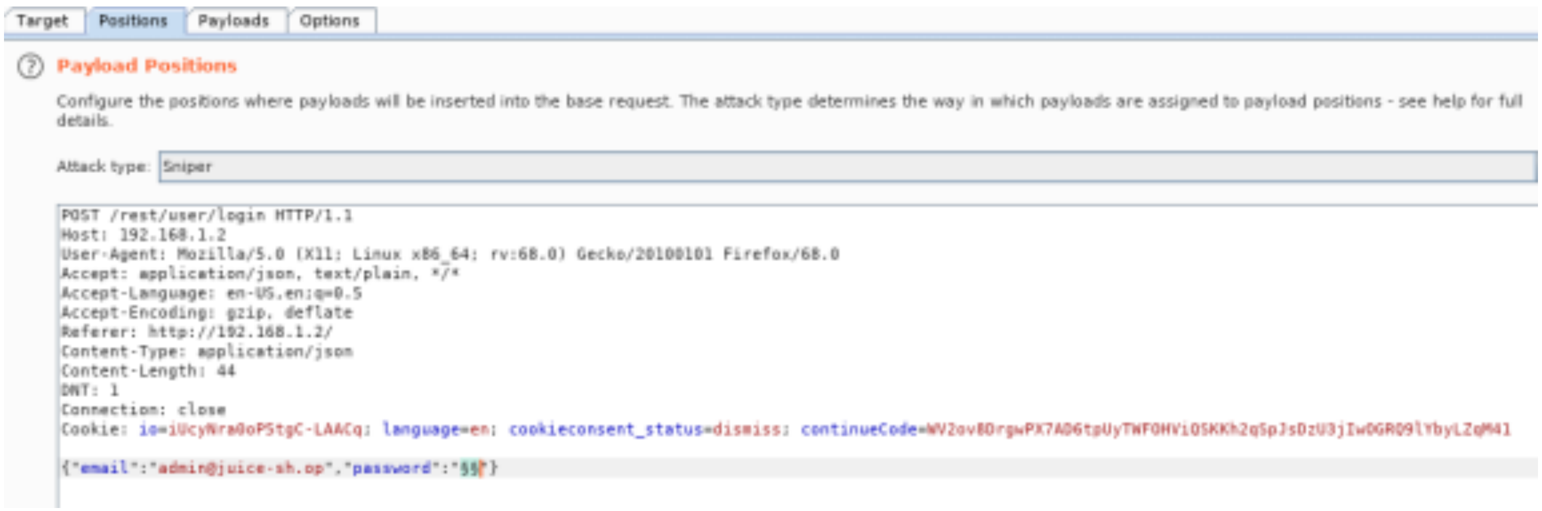
[Task 4] Who broke my lock?!



In this task, we will look at exploiting authentication through different flaws. When talking about flaws within authentication, we include mechanisms that are vulnerable to manipulation. These mechanisms, listed below, are what we will be exploiting.

- **Weak passwords in high privileged accounts**
- **Forgotten password pages**

More information: Broken Authentication

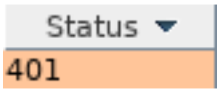


Configure the positions where payloads will be inserted into the base request. The attack type determines the way in which payloads are assigned to payload positions - see help for full details.

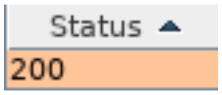
Attack type: Sniper

```
POST /rest/user/login HTTP/1.1
Host: 192.168.1.2
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://192.168.1.2/
Content-Type: application/json
Content-Length: 44
DNT: 1
Connection: close
Cookie: io=iUcyNra0oPStgC-LAACq; language=en; cookieconsent_status=dismiss; continueCode=WV2ov8DrgwPX7AD6tpUyTWFOHViOSKKh2q5pJsDzU3jIw0GRQ9lYbyLZqM41

{"email":"admin@juice-sh.op","password":"§§"}
```

A **failed** request will receive a **401 Unauthorized**


Status
401

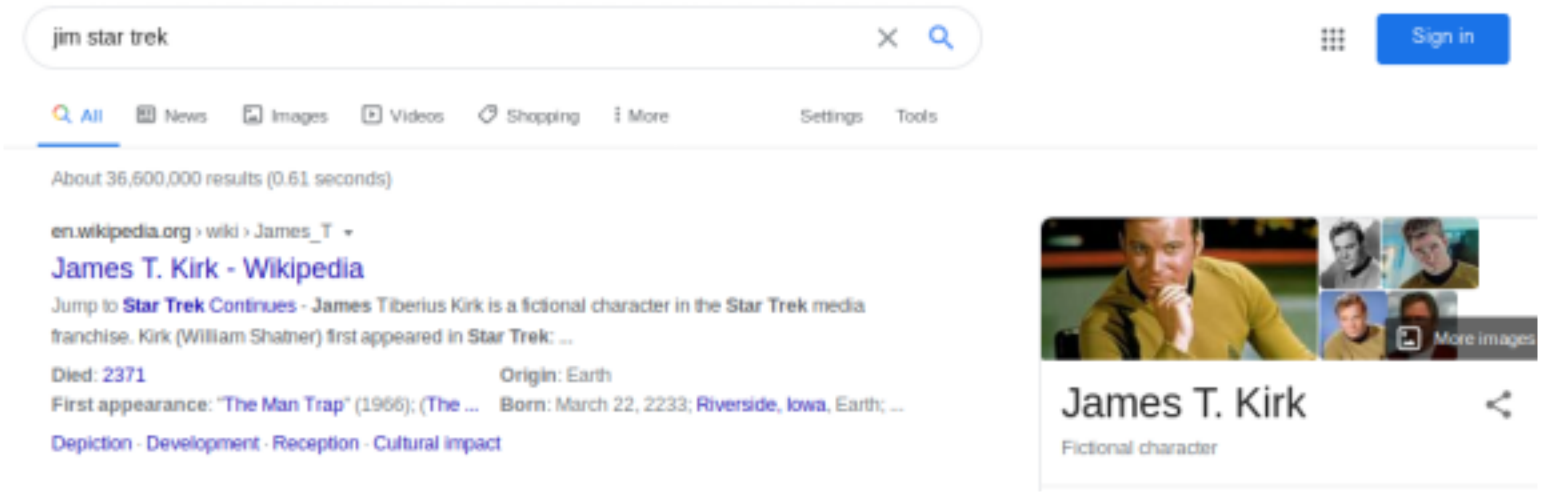
Whereas a **successful** request will return a **200 OK**


Status
200

Once completed, login to the account with the password.

#1
Question #1: Bruteforce the Administrator account's password! We have used SQL Injection to log into the Administrator account but we still don't know the password. Let's try a brute-force attack! We will once again capture a login request, but instead of sending it through the proxy, we will send it to Intruder. Goto Positions and then select the Clear § button. In the password field place two § inside the quotes. It should look like the image below. For the payload, we will be using the best1050.txt from Seclists. (Which can be installed: apt-get install seclists) You can load the list from /usr/share/seclists/Passwords/Common-Credentials/-best1050.txt Once the file is loaded into Burp, start the attack. You will want to filter for the request by status. A failed request will receive a 401 Unauthorized Whereas a successful request will return a 200 OK. Once completed, login to the account with the password.

c2110d06dc6f81c67cd8099ff0ba601241f1ac0e


jim star trek

About 36,600,000 results (0.61 seconds)

en.wikipedia.org › wiki › James_T

James T. Kirk - Wikipedia

Jump to Star Trek Continues - James Tiberius Kirk is a fictional character in the Star Trek media franchise. Kirk (William Shatner) first appeared in Star Trek: ...

Died: 2371
Origin: Earth
First appearance: "The Man Trap" (1966); (The ...
Born: March 22, 2233; Riverside, Iowa, Earth; ...

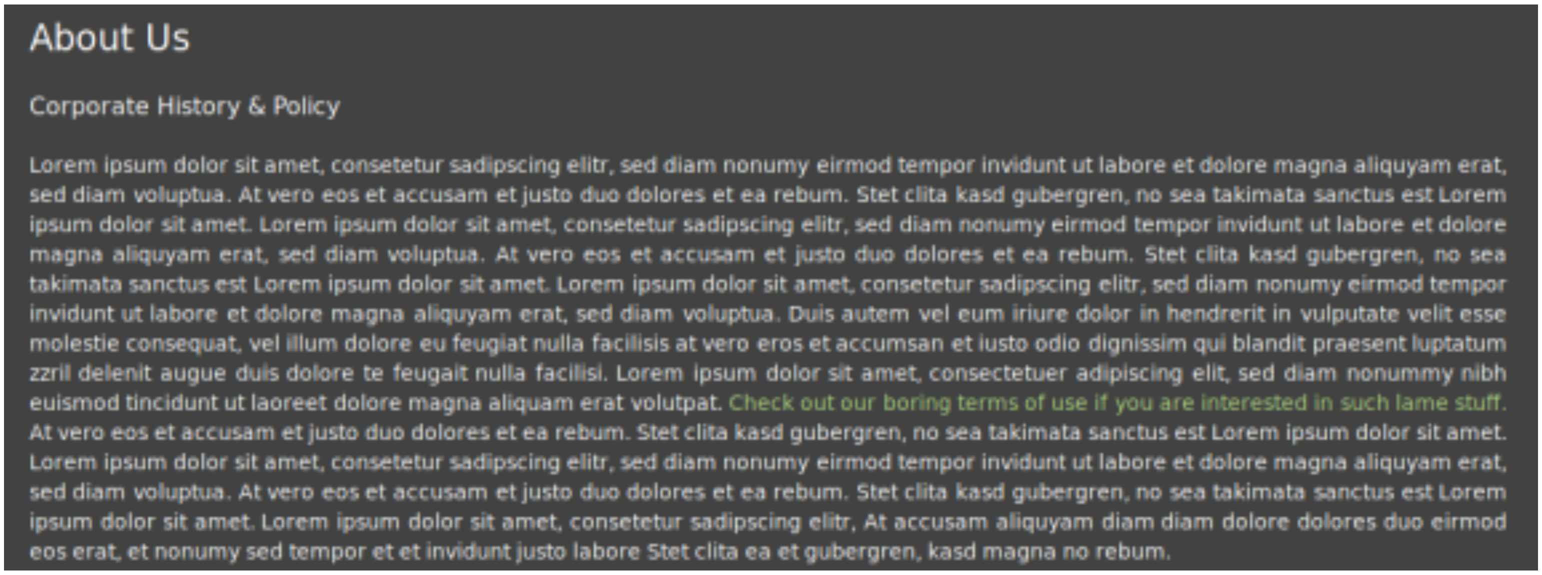
Depiction · Development · Reception · Cultural impact

James T. Kirk
Fictional character

#2
Question #2: 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 Jame T. Kirk from Star Trek. Looking through the wiki page we find that he has a brother. Looks like we have found our security question answer!Inputting that into the Forgot Password page allows you to successfully change his password.You can change it to anything you want!

094fbc9b48e525150ba97d05b942bbf114987257

[Task 5] AH! Don't look!

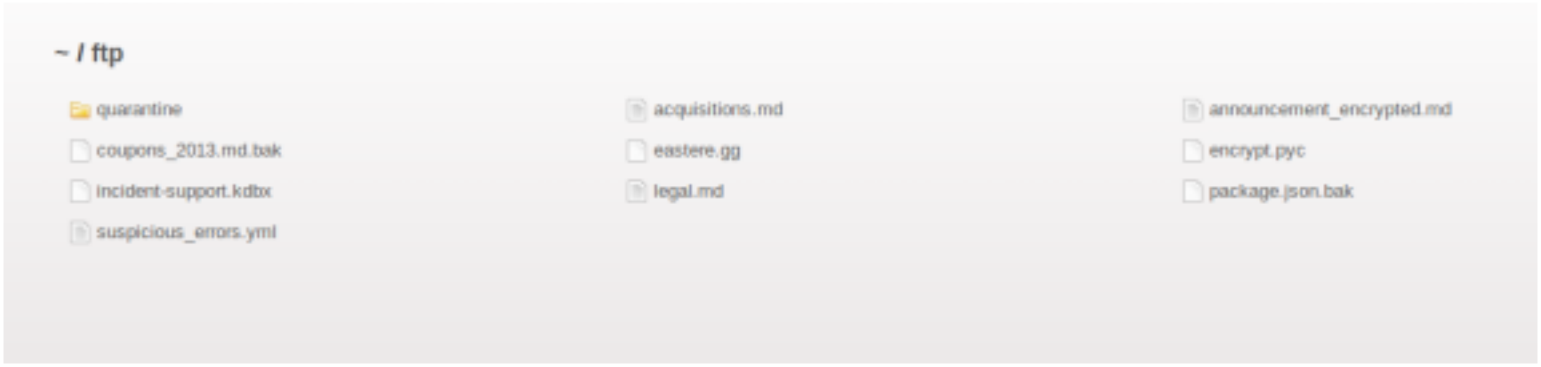


About Us

Corporate History & Policy

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi. Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Check out our boring terms of use if you are interested in such lame stuff. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, At accusam aliquyam diam diam dolore dolores duo eirmod eos erat, et nonumy sed tempor et et invidunt justo labore Stet clita ea et gubergren, kasd magna no rebum.

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.
More information: Sensitive Data Exposure



~ / ftp

```
quarantine              acquisitions.md    announcement_encrypted.md
coupons_2013.md.bak     eastere.gg         encrypt.pyc
incident-support.kdbx   legal.md           package.json.bak
suspicious_errors.yml
```

#1

Question #1: Access the Confidential Document!
Navigate to the About Us page, and hover over the "Check out our terms of use". You will see that it links to /ftp/legal.md. Navigating to that /ftp/ directory reveals that it is exposed to the public!

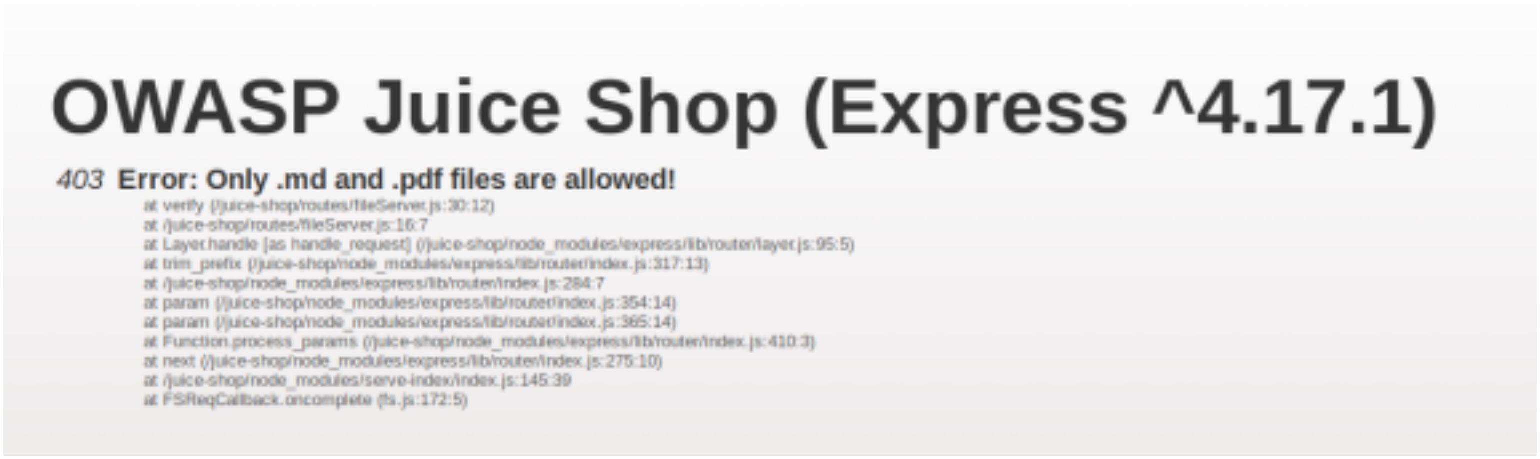
We will download the acquisitions.md and save it. It looks like there are other files of interest here as well.

edf9281222395a1c5fee9b89e32175f1ccf50c5b

#2

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!



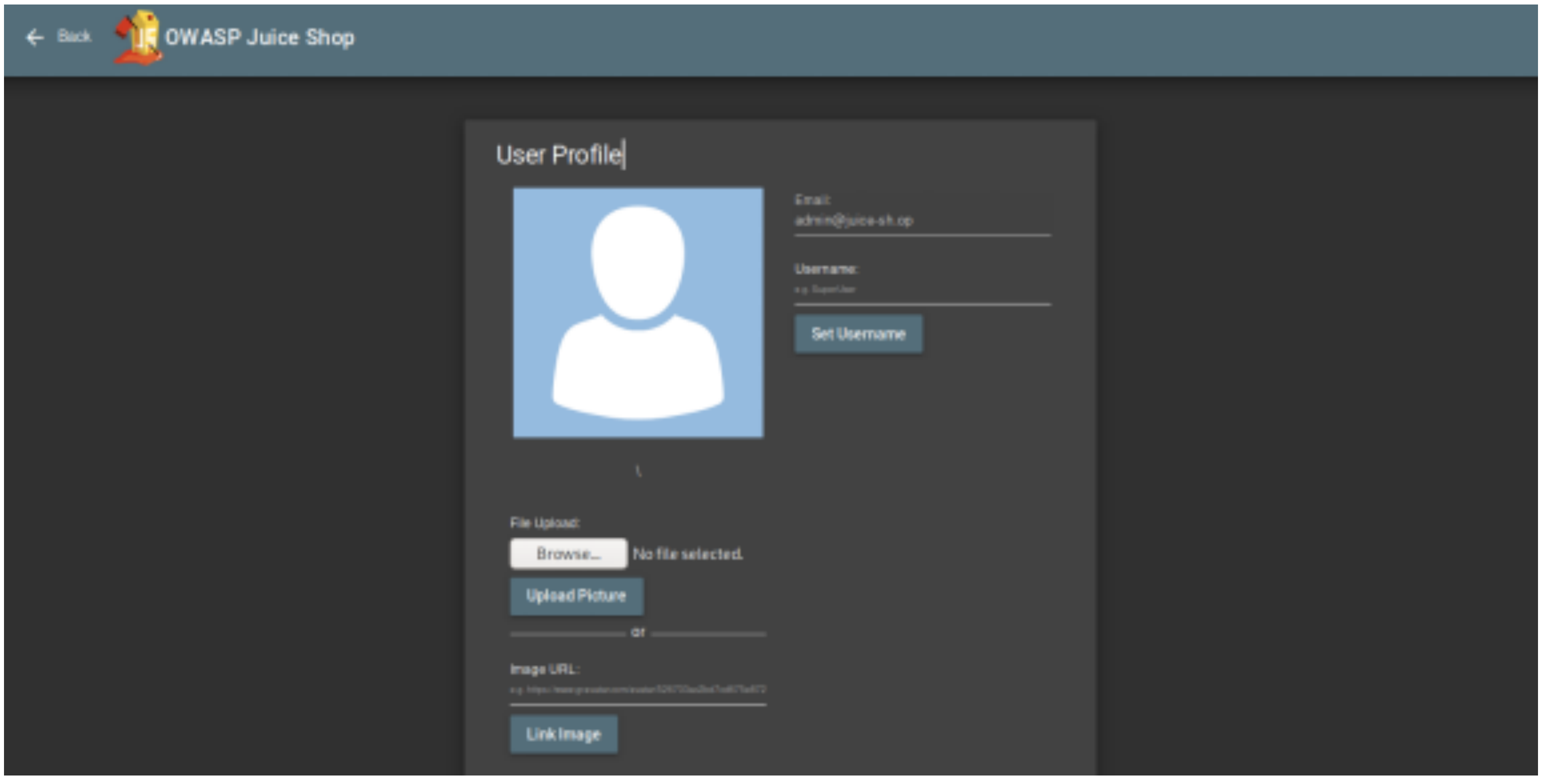
```
OWASP Juice Shop (Express ^4.17.1)

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)
```

#3

Question #3: Download the Backup file!
We will now go back to the /ftp/ folder and try to download the backup file. It 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, 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 will bypass the 403 error!
How 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.

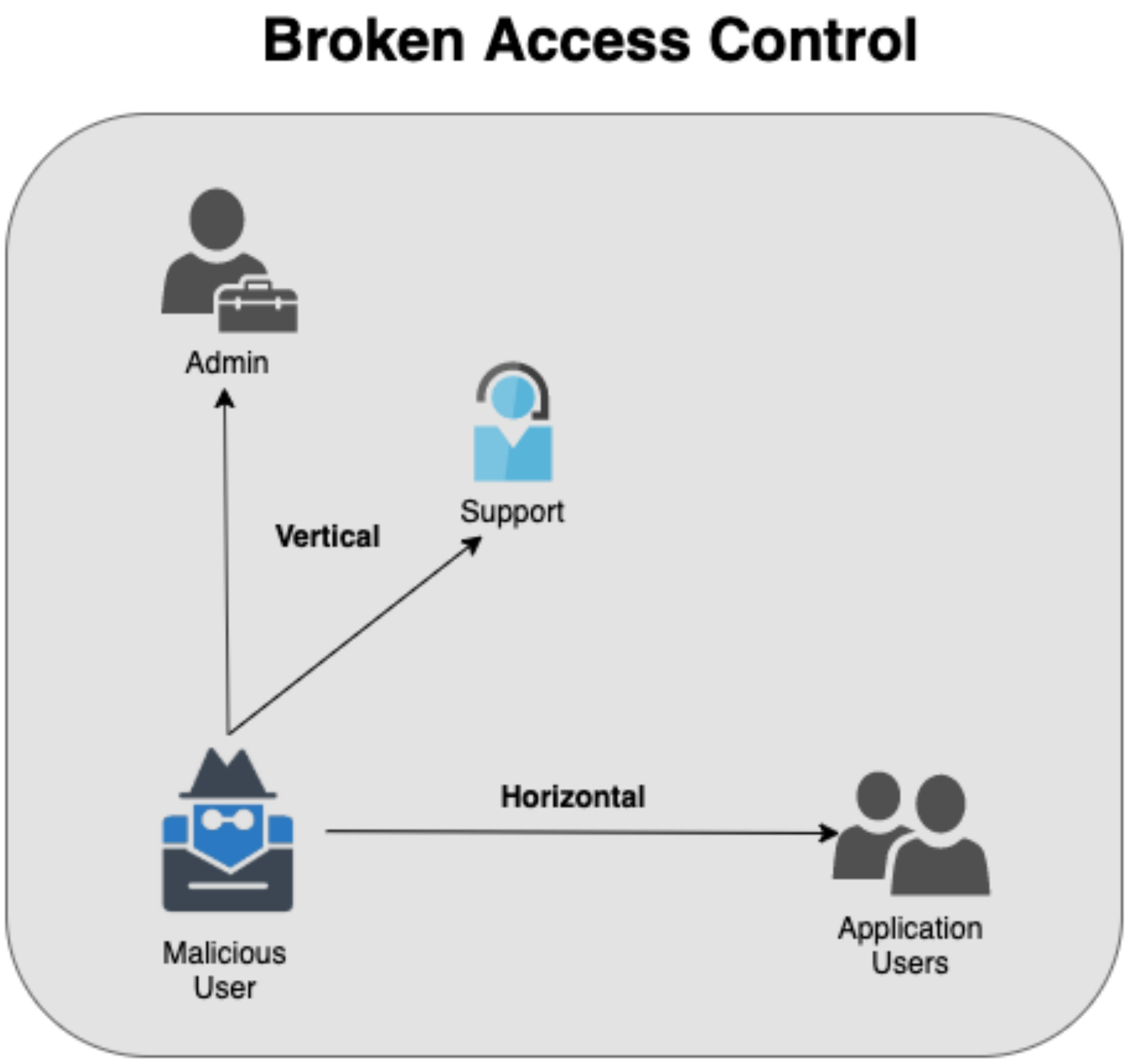
[Task 6] Who's flying this thing?



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.

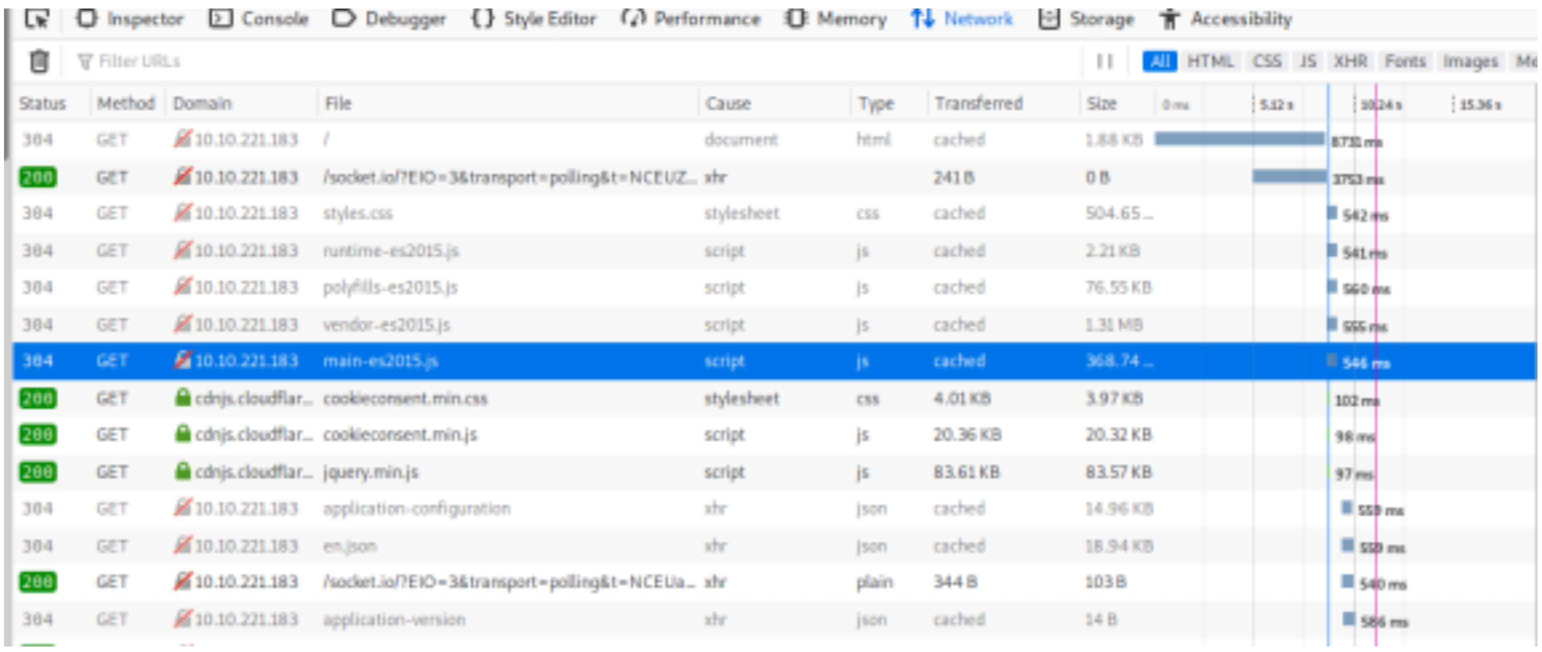
When Broken Access Control exploits or bugs are found, it will be categorised into one of two types:

Type	**Description**
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.

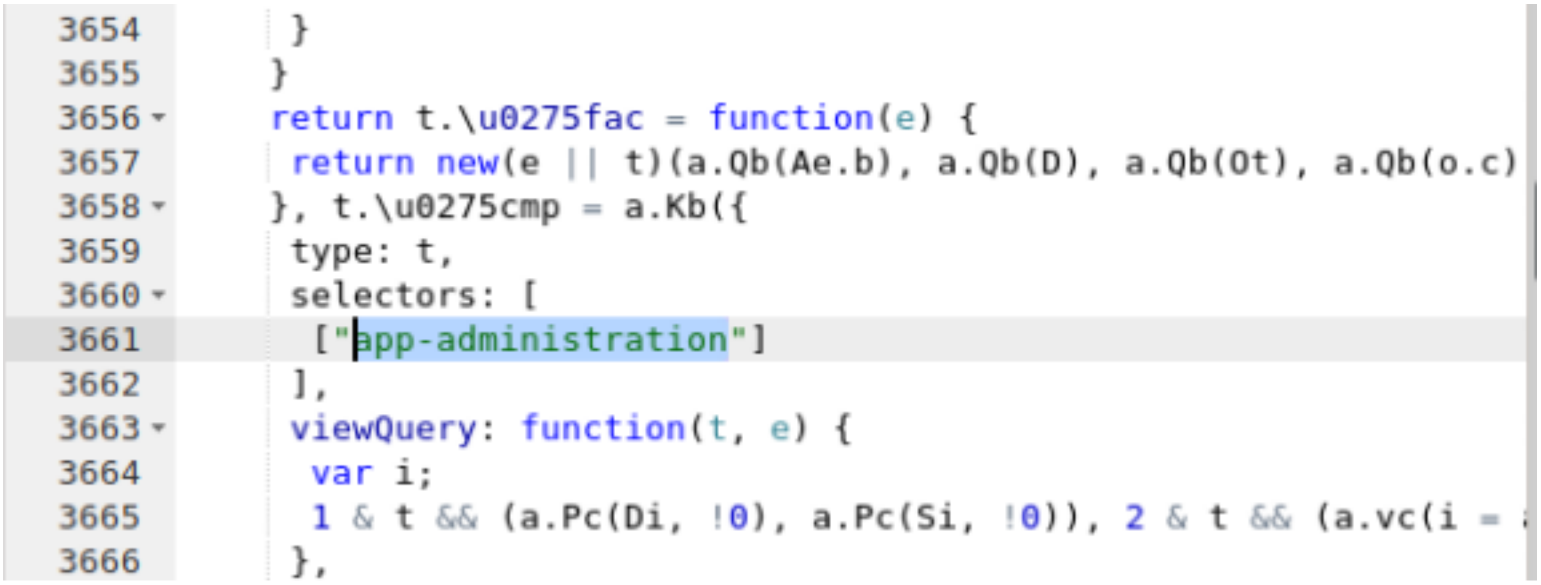


Credits: Packetlabs.net

More information: Broken Access Control



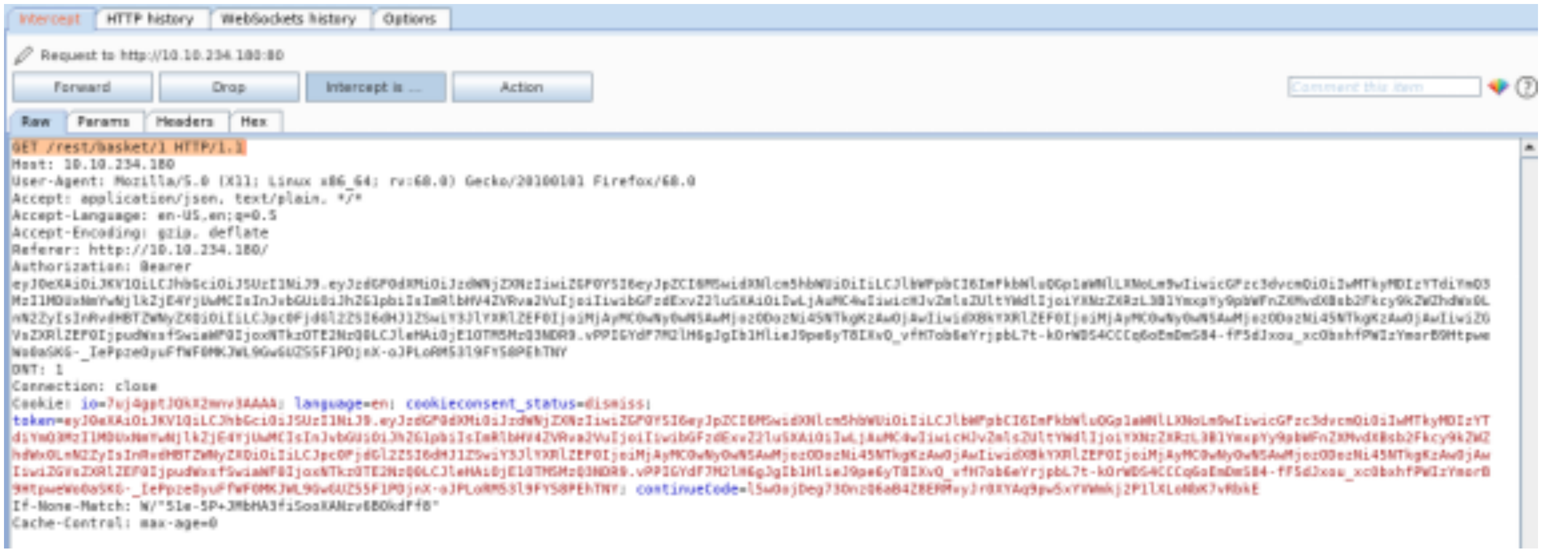
Status	Method	Domain	File	Cause	Type	Transferred	Size	0 ms
304	GET	10.10.221.183	/	document	html	cached	1.88 KB	8731 ms
200	GET	10.10.221.183	/socket.io/?EIO=3&transport=polling&t=NCEUZ...	xhr		241 B	0 B	3753 ms
304	GET	10.10.221.183	styles.css	stylesheet	css	cached	504.65 ...	542 ms
304	GET	10.10.221.183	runtime-es2015.js	script	js	cached	2.21 KB	541 ms
304	GET	10.10.221.183	polyfills-es2015.js	script	js	cached	76.55 KB	560 ms
304	GET	10.10.221.183	vendor-es2015.js	script	js	cached	1.31 MB	555 ms
304	GET	10.10.221.183	main-es2015.js	script	js	cached	368.74 ...	546 ms
200	GET	cdnjs.cloudflar...	cookieconsent.min.css	stylesheet	css	4.01 KB	3.97 KB	102 ms
200	GET	cdnjs.cloudflar...	cookieconsent.min.js	script	js	20.36 KB	20.32 KB	98 ms
200	GET	cdnjs.cloudflar...	jquery.min.js	script	js	83.61 KB	83.57 KB	97 ms
304	GET	10.10.221.183	application-configuration	xhr	json	cached	14.96 KB	559 ms
304	GET	10.10.221.183	en.json	xhr	json	cached	18.94 KB	559 ms
200	GET	10.10.221.183	/socket.io/?EIO=3&transport=polling&t=NCEUa...	xhr	plain	344 B	103 B	540 ms
304	GET	10.10.221.183	application-version	xhr	json	cached	14 B	586 ms



```
        }
      }
      return t.\u0275fac = function(e) {
        return new(e || t)(a.Qb(Ae.b), a.Qb(D), a.Qb(Ot), a.Qb(o.c)
      }, t.\u0275cmp = a.Kb({
        type: t,
        selectors: [
          ["app-administration"]
        ],
        viewQuery: function(t, e) {
          var i;
          1 & t && (a.Pc(Di, !0), a.Pc(Si, !0)), 2 & t && (a.vc(i =
        },
```

#1
Question #1: Access the administration page! First, we are going to open the Network Viewer on Firefox. This can be done with the keyboard shortcut: Ctrl+Shift+E Or by navigating to it in the Web Developers menu. We are then going to refresh the page and look for a GET request for main-es2015.js   We will then go to that page at: http://MACHINE_IP/main-es2015.js This step is optional but if you prefer to see it formatted better try a js Beautifier at https://codebeautify.org/jsviewer. 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 "app-administration" This hints towards a page called "Administration" but going there while not logged in doesn't work. But because this is an Administrator page, it makes sense that we need to be in the Admin account.

946a799363226a24822008503f5d1324536629a0

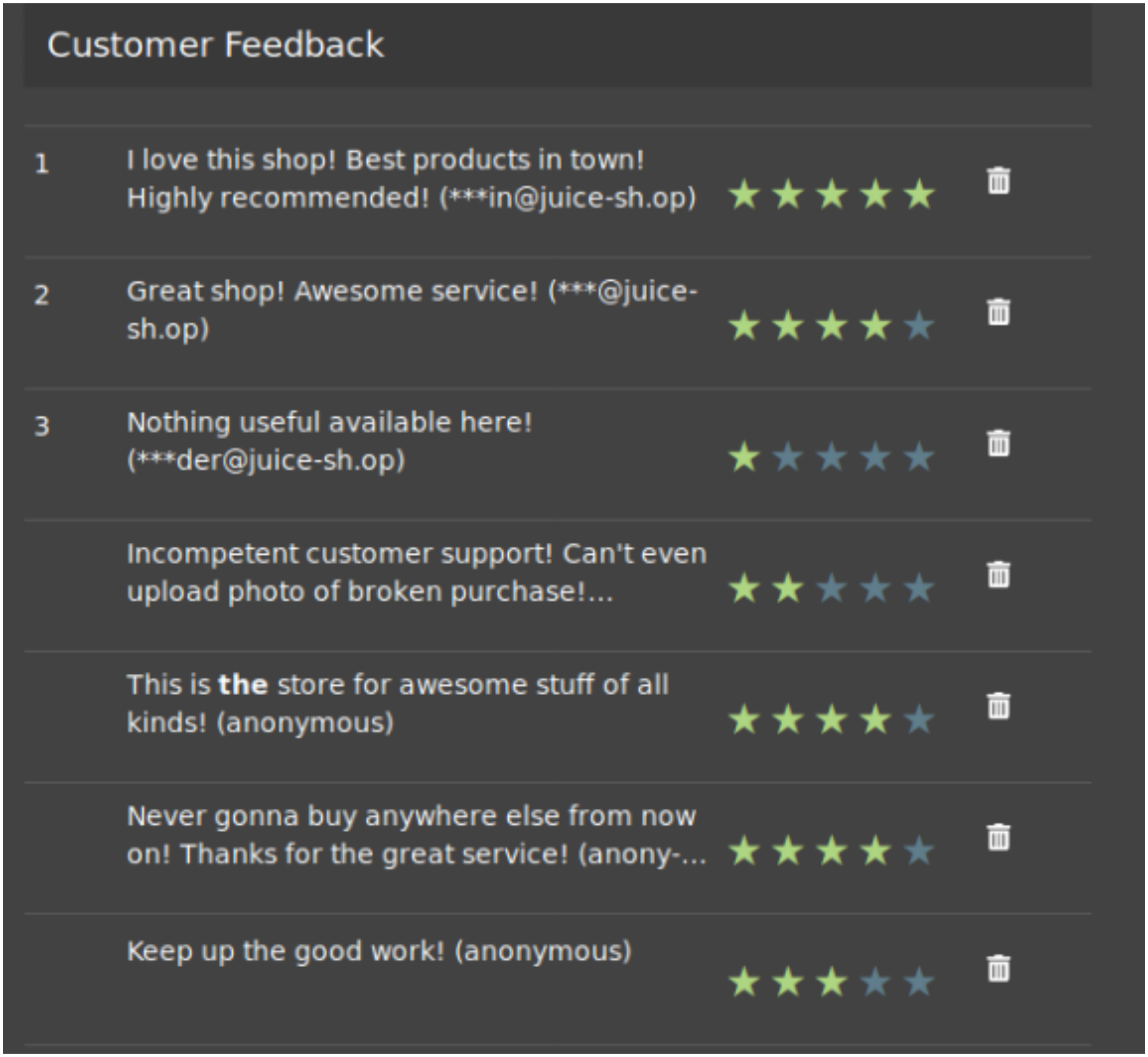


Request to http://10.10.234.180:80

```
GET /rest/basket/1 HTTP/1.1
Host: 10.10.234.180
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://10.10.234.180/
Authorization: Bearer
[illegible]
DNT: 1
Connection: close
Cookie: io=7uj4gptJOkX2mnv3AAAA; language=en; cookieconsent_status=dismiss;
token=[illegible]; continueCode=[illegible]
If-None-Match: W/"51e-5P+JMbHA3fiSooXANrv6BOkdFf8"
Cache-Control: max-age=0
```

#2
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 Now, we are going to change the number after /basket/ to 2 It will now show you the basket of UserID 2. You can do this for other UserIDs as well, provided that they have one!

41b997a36cc33fbe4f0ba018474e19ae5ce52121

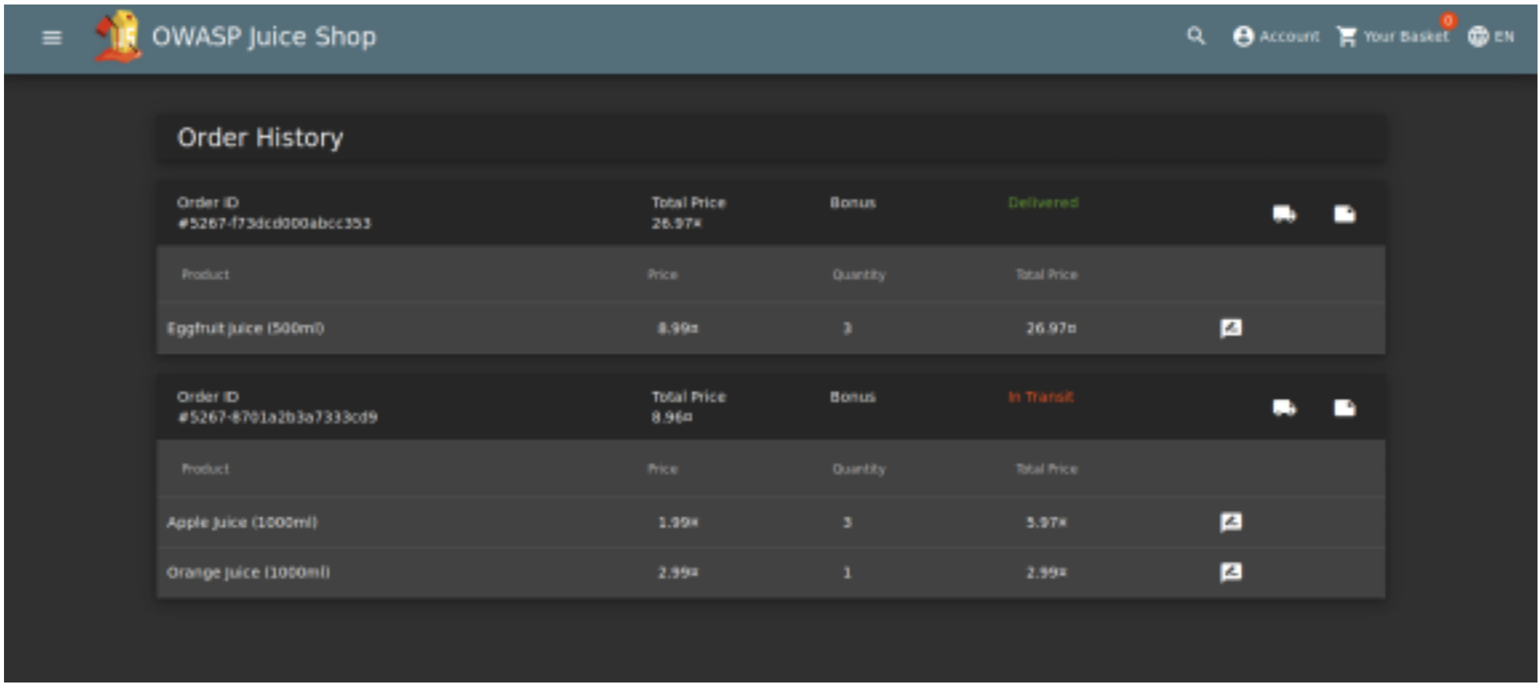

Customer Feedback

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

#3
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!

50c97bcce0b895e446d61c83a21df371ac2266ef

[Task 7] Where did that come from?



Order History

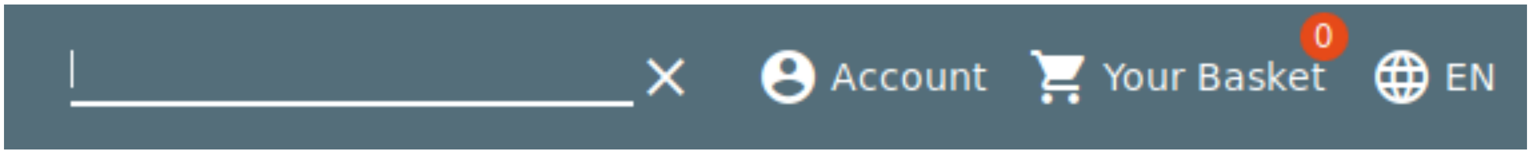
Order ID #5267-f73dcd000abcc353 — Total Price 26.97¤ — Bonus — Delivered

Product	Price	Quantity	Total Price
Eggfruit Juice (500ml)	8.99¤	3	26.97¤

Order ID #5267-8701a2b3a7333cd9 — Total Price 8.96¤ — Bonus — In Transit

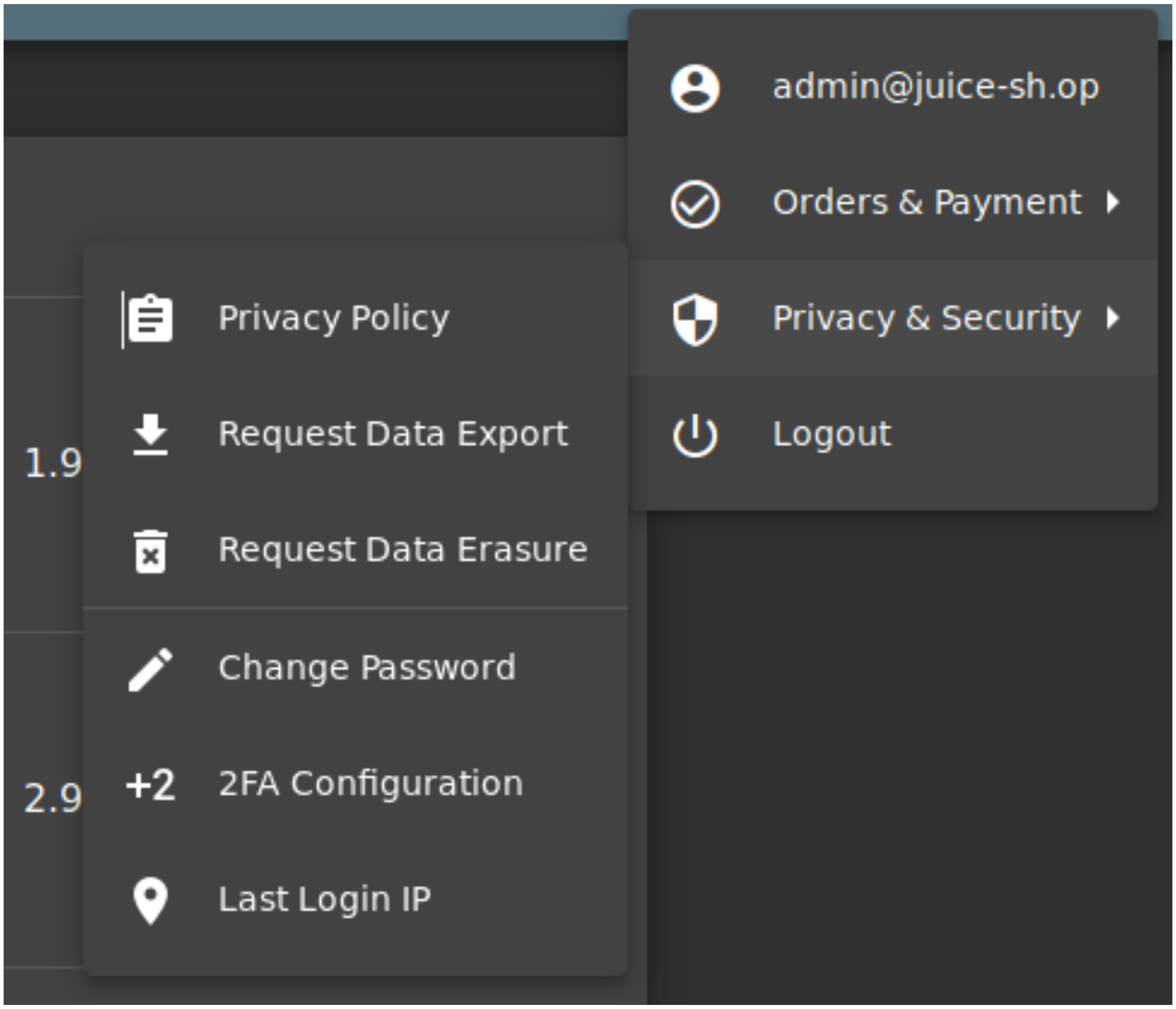
Product	Price	Quantity	Total Price
Apple Juice (1000ml)	1.99¤	3	5.97¤
Orange Juice (1000ml)	2.99¤	1	2.99¤

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:
1. DOM (Special)
2. Persistent (Server-side)
3. Reflected (Client-side)
DOM XSS *(Document Object Model-based Cross-site Scripting)* uses the HTML environment to execute malicious javascript. This type of attack commonly uses the *<script></script>* HTML tag.
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 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.
More information: Cross-Site Scripting XSS



#1
Question #1: Perform a DOM XSS! We will use the iframe tag with an alert: <iframe src="javascript:alert(`xss`)"> This type of XSS is also called XFS (Cross-Frame Scripting), is one of the most common forms of detecting XSS within web applications. Websites that allow the user to modify the iframe will most likely be vulnerable to XSS.

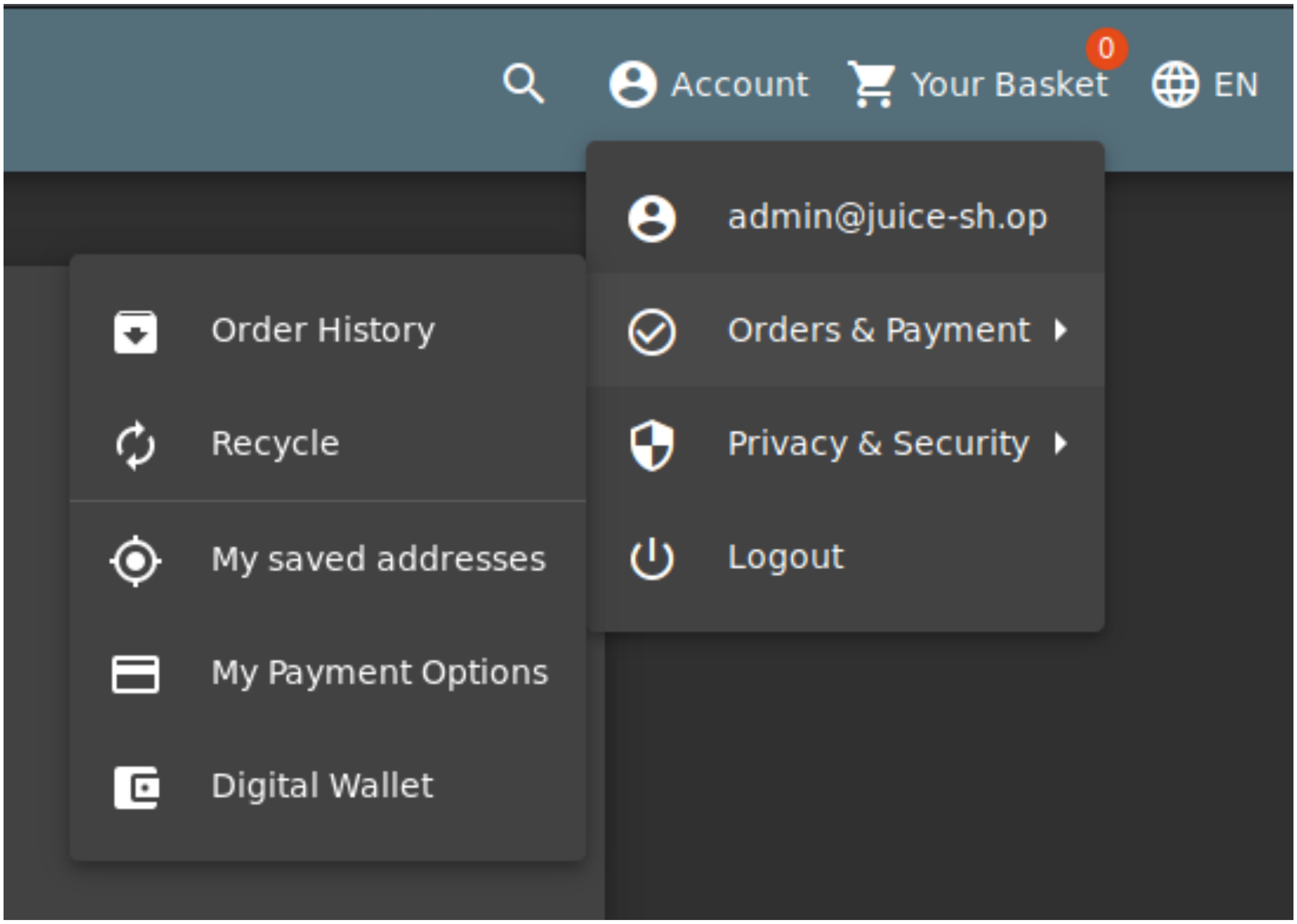
9aaf4bbea5c30d00a1f5bbcfce4db6d4b0efe0bf

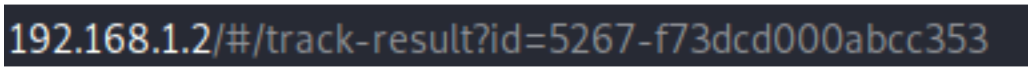


click me	click me
True-Client-IP	<iframe src="javascript:alert(`xss`)">

#2
Question #2: Perform a persistent XSS! We are going to navigate to the "Last Login IP" page for this attack. As it logs the 'last' login IP we will now logout so that it logs the 'new' IP. 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: 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!

149aa8ce13d7a4a8a931472308e269c94dc5f156





```
192.168.1.2/#/track-result?id=5267-f73dcd000abcc353
```

#3
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!

23cefee1527bde039295b2616eeb29e1edc660a0

[Task 8] Exploration!

If you wish to tackle some of the harder challenges that were not covered within this room, check out the /score-board/ section on Juice-shop.
Here you can see your completed tasks as well as other tasks in varying difficulty.

#1
Have fun!

7efd3174f9dd5baa03a7882027f2824d2f72d86e