



CYBER SECURITY PROJECT

TEAM MEMBERS:

Reem Ahmed Moustafa 20305591
Mohanad Gaber Shehata 2305391
Youssef Ahmed Wahid 2305105





01

introduction

Attacking the OWASP Juice Shop involves exploiting common vulnerabilities to understand web application weaknesses and how attackers exploit them. The Juice Shop is intentionally designed with security flaws, making it an excellent platform for learning penetration testing.

02

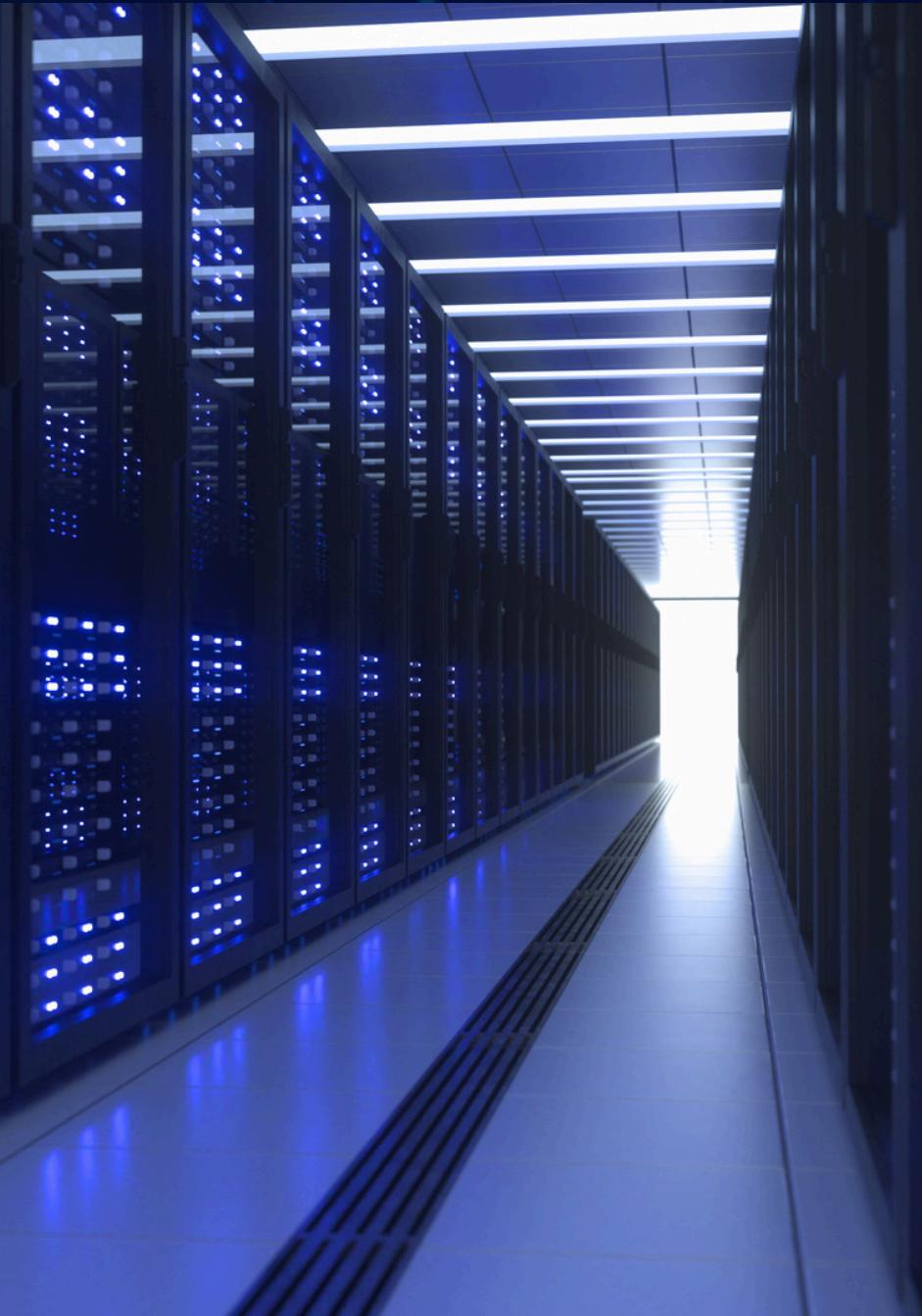
Objectives

- Learn and apply penetration testing techniques.
- Understand the root causes of common web vulnerabilities.
- Document findings and create a video report demonstrating the attacks and their impacts.

03

Deliverables

- Video Report: A 10-15 minute video detailing the attack scenarios, methodology, and outcomes, narrated by the team members.
- GitHub Repository: Includes a detailed report and scripts.



• EXECUTIVE SUMMARY

1- Purpose of the Test:

The goal of this project is to assess the security posture of the OWASP Juice Shop application by identifying vulnerabilities through simulated attacks. This test provides insights into the application's weaknesses and highlights areas for improvement.



• Key Findings:

1. Critical Vulnerabilities:
 - SQL Injection: Allows unauthorized access to sensitive data.
 - Cross-Site Scripting (XSS): Enables execution of arbitrary scripts in user browsers.
 - Brute Force on Admin Credentials: Lack of rate-limiting enables unauthorized access.
2. Impact: These vulnerabilities can lead to data breaches, unauthorized access, and compromised user trust.

● Summary of Recommendations:

- Implement input sanitization and validation to prevent SQL Injection and XSS.
- Enforce strong authentication policies, including rate-limiting and account lockouts.
- Conduct regular penetration testing to identify and mitigate vulnerabilities proactively.

● Scope and Methodology :

1. Scope:-

Target Application: OWASP Juice Shop

Assets Tested: Web application, APIs, and backend systems

2. Methodology:

- Approach: Black-box testing to simulate an attacker's perspective.
- Tools Used:- Burp Suite: For intercepting and analyzing requests.
 - OWASP ZAP: For automated scanning of vulnerabilities.
 - BrupSuite: For brute-forcing login credentials.
- Gobuster and Dirb: For discovering hidden paths and resources

• Vulnerability Findings :

1. Cross-Site Scripting (XSS)

- Description: The application reflects unsanitized user input in the product search bar.
- Risk: High. Allows execution of malicious scripts, which could steal user session cookies.

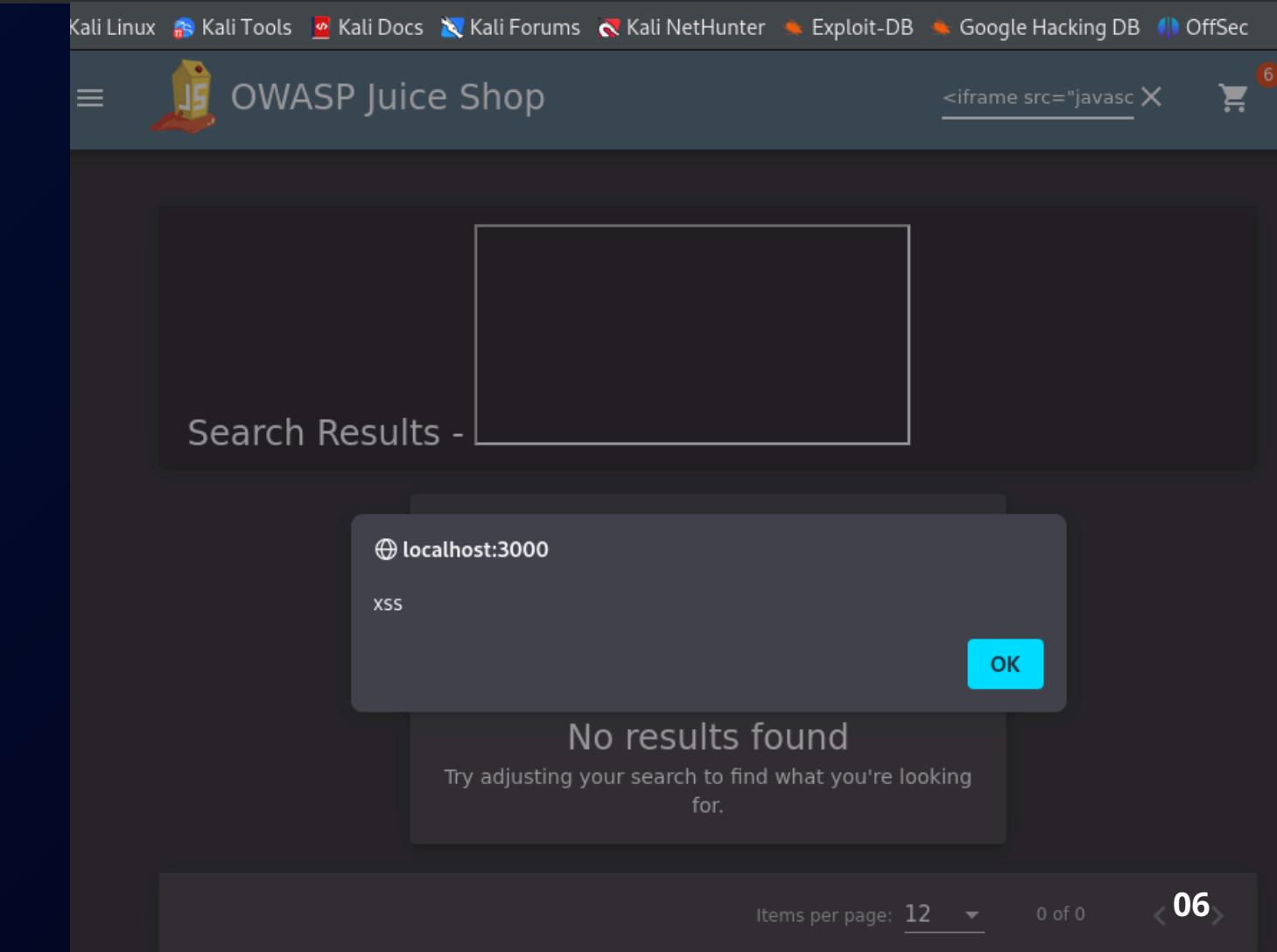
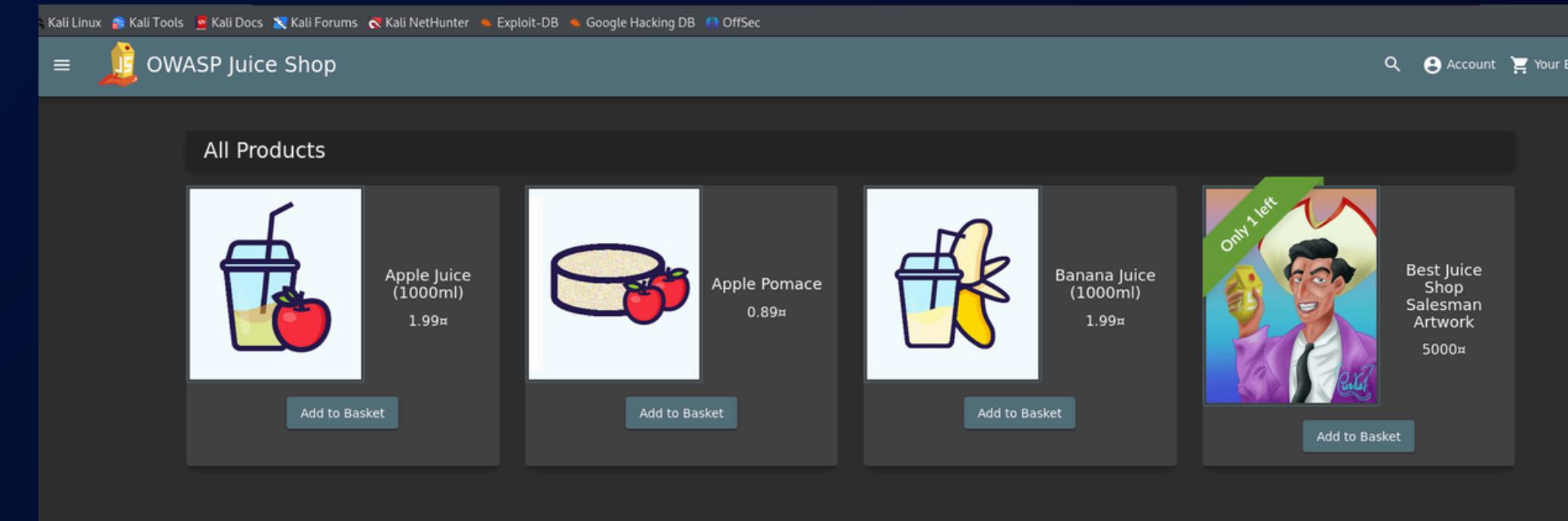
- Evidence:

- Screenshot of JavaScript alert box execution.-

Remediation Steps:

- Escape and sanitize all user inputs.
- Use Content Security Policy (CSP) to mitigate script execution.

- move to search bar and write this command
- <iframe src="javascript:alert('xss')">



2. Enumeration to Find Admin Path

Objective:

Discover hidden paths in the application, such as the admin page, for further exploitation.

Steps:

1. Use a tool like Gobuster or Dirb to brute-force the URL structure.
2. bash
3. Copy code
4. dirb https://juice-shop.herokuapp.com /usr/share/wordlists/dirb/common.txt -v | grep admin
5. Analyze the results to identify URLs such as /admin or /management.
6. Verify the discovered admin path by visiting the URL in a browser.

Outcome:

- Identification of the admin functionality.
- Access to sensitive parts of the application for further exploitation.

```
(kali㉿kali)-[~]
$ dirb https://juice-shop.herokuapp.com /usr/share/wordlists/dirb/common.txt -v | grep admin
+ https://juice-shop.herokuapp.com/_admin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/_vti_bin/_vti_adm/admin.dll (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/~admin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/~administrator (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/~sysadmin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin.cgi (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin.php (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin.pl (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin_ (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin_area (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin_banner (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin_c (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin_index (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin_interface (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin_login (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin_logon (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin1 (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin2 (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin3 (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin4_account (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin4_colon (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin-admin (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin-console (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admincontrol (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admincp (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/adminhelp (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/admin-interface (CODE:200|SIZE:3748)
```

+ https://juice-shop.herokuapp.com/direct**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/e107_**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/ezsqlite**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/file**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/globes_**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/hpwebjet**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/iis**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/index_**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/indy_**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/Indy_**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/INSTALL_**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/irc-mac**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/list**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/login**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/logo_sys**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/mac**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/my**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/navsite**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/new**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/news**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/openvpn**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/pg**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/php**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/phpldap**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/phpmy**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/phpmy**admin2** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/phppg**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/project-**admins** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/pure**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/r**admin**d (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/r**admin**d-1 (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/resin-**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/server_**admin**_small (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/shop**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/site**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/soho**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/sql**admin** (CODE:200|SIZE:3748)
+ https://juice-shop.herokuapp.com/sql-**admin** (CODE:200|SIZE:3748)

<https://juice-shop.herokuapp.com/pgadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/phpadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/phpldapadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/phpmyadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/phpmyadmin2> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/phppgadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/project-admins> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/pureadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/radmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/radmin-1> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/resin-admin> (CODE:200|SIZE:3748)
https://juice-shop.herokuapp.com/server_admin_small (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/shopadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/siteadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/sohoadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/sqladmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/sql-admin> (CODE:200|SIZE:3748)
https://juice-shop.herokuapp.com/ss_vms_admin_sm (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/sshadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/staradmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/sysadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/sys-admin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/sysadmin2> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/sysadmins> (CODE:200|SIZE:3748)
https://juice-shop.herokuapp.com/system_admin (CODE:200|SIZE:3748)
https://juice-shop.herokuapp.com/system_administration (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/system-admin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/system-administration> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/topicadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/ur-admin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/useradmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/vadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/vmailadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/vsadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/wbsadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/webadmin> (CODE:200|SIZE:3748)
<https://juice-shop.herokuapp.com/wizmysqladmin> (CODE:200|SIZE:3748) 07
<https://juice-shop.herokuapp.com/wp-admin> (CODE:200|SIZE:3748)

3. Brute Force on Admin Credentials

- Description: Lack of rate-limiting allows attackers to repeatedly guess admin credentials.

- Risk: Critical. Provides unauthorized access to administrative functionalities.

- Evidence:- Logs showing brute-force attempts.- Screenshot of successful login.

- Remediation Steps:

- Enforce account lockouts after a defined number of failed attempts.

- Implement CAPTCHA on login pages

The screenshot shows a Firefox browser window with the URL `192.168.47.134:3000/#/basket`. The page displays a shopping cart with items: Eggfruit Juice (500ml) and Total Price: 21.94. Below the cart, there is a "Checkout" button. At the bottom of the browser, the Network tab of the developer tools is selected, showing a list of network requests. One request, highlighted in blue, is a GET request to `192.168.47.1.../socket.io/?EIO=4&transport=polling&t=OVM4fr` with a status code of 200 and a response body containing JSON data: `status: "success"` and `data: Object { id: 2, UserId: 2, createdAt: "2023-05-01T06:11:04.305Z", ... }`.

The image contains three side-by-side screenshots of the Burp Suite interface. The left screenshot shows the Proxy tab with a list of captured requests, including a POST to the login endpoint. The middle screenshot shows the Intruder tool's payload editor with a simple list of password guesses like "admin", "fegvrg", and "barney". The right screenshot shows the results of the attack, displaying the login page with the "fegvrg" password successfully logged in. The Burp Suite interface includes tabs for Project, Target, Proxy, Intruder, Repeater, View, Help, and various tool-specific panels like Inspector and Resource pool.

This image shows a screenshot of a penetration testing environment with multiple windows open, focused on the OWASP Juice Shop application.

The top-left window is Burp Suite Community Edition v2023.10.1.1 - Temporary Project. It displays the "Tasks" tab with a live passive crawl from Proxy (all traffic) showing 2 items added to site map and 2 responses processed. The "Issue activity [Pro version only]" tab lists various security issues found, such as Suspicious input transformation (reflected), SMTP header injection, Serialized object in HTTP message, Cross-site scripting (DOM-based), XML external entity injection, External service interaction (HTTP), Web cache poisoning, Server-side template injection, SQL injection, and OS command injection. The "Event log" tab shows a single entry: "Proxy service started on 127.0.0.1:8080" at 22:13:50 22 Sep 2023.

The top-right window is the OWASP Juice Shop login page. It features a "Login" form with fields for "Email" (admin@juice-shop.op) and "Password" (felfvrg). Below the form are links for "Forgot your password?", "Log in", "Remember me", and "Log in with Google". A "Not yet a customer?" link is also present.

The bottom-left window is another instance of Burp Suite showing the "Proxy" tab. The "Intercept" dropdown is set to "HTTP history". The "Request" pane shows a POST request to /rest/user/login with parameters: email=admin@juice-shop.op and password=felfvrg. The "Inspector" pane shows the raw request and response headers and bodies.

The bottom-right window is another instance of Burp Suite showing the "Repeater" tab. The target is set to https://juice-shop.herokuapp.com/. The "Request" pane shows the same POST /rest/user/login request. The "Response" pane shows the raw response body, which is a JSON object containing session information and a token.

The screenshot displays the Burp Suite interface during a penetration testing session against the OWASP Juice Shop application. The main window shows the 'OWASP Juice Shop' login page with fields for 'Email' (admin@juice-shop.com) and 'Password' (fefgvrg). The 'Inspector' tool is open, showing the request and response details for this login attempt. The 'Request' tab in the Inspector shows the POST data sent to the /rest/user/login endpoint, which includes the email and password. The 'Response' tab shows the JSON response from the server, which includes a token and other session-related information. The 'Burp' tab in the top navigation bar is selected, indicating the current context of the session.

4- login admin

we use it to log in with the administration user account, its category is injection

we will go to the GitHub repo it is sql injection paid and we tried to search for the right user name and it is 'OR 1 = 1-- - and the password is the same

The screenshot displays four Firefox browser tabs:

- Top Left Tab:** A GitHub repository page for "payloadbox/sql-injection-payload-list". It shows a "README.md" file with various SQL injection payload examples, such as "explo - Human And Machine Readable Web Vulnerability Testing Format", "Blind-Sql-Bitshifting - Blind SQL-Injection via Bitshifting", and "Leviathan - Wide Range Mass Audit Toolkit". Below this is a section titled "Generic SQL Injection Payloads" containing many more complex payload examples.
- Top Right Tab:** The OWASP Juice Shop application interface. The URL is 192.168.47.134:3000/#/search. A green success message at the top states: "You successfully solved a challenge: Login Admin (Log in with the administrator's user account.)". Below this is a grid of product cards labeled "All Products".
 - Apple Juice (1000ml) - Price: 1.99€
 - Apple Pomace - Price: 0.89€
 - Banana Juice (1000ml) - Price: 1.99€
 - Best Juice Shop Salesman Artwork - Price: 5000€ (with a cartoon character holding a juice glass)
 - Carrot Juice (1000ml) - Price: 2.99€
 - Eggfruit Juice (500ml) - Price: 8.99€
 - Fruit Press - Price: 89.99€
 - Green Smoothie - Price: 1.99€
- Bottom Left Tab:** The OWASP Juice Shop application interface. The URL is 192.168.47.134:3000/#/login. It shows a "Login" form with fields for "Email" and "Password". The "Email" field has the value "'OR 1 = 1-- -".
- Bottom Right Tab:** The OWASP Juice Shop application interface. The URL is 192.168.47.134:3000/#/login. It shows the same "Login" form as the bottom-left tab, but the "Email" field now contains the value "admin' OR 1 = 1-- -".

5- 5 stars feedback

we went to the administration section and then clicked on the only 5 stars feedback
and then it is successfully done

The screenshot shows two Firefox browser windows side-by-side. Both windows are displaying the OWASP Juice Shop administration interface at the URL `https://www.owasp.org/juice-shop/administration`.

Top Window (Right): This window shows a success message: "You successfully solved a challenge: Five-Star Feedback (Get rid of all 5-star customer feedback.)". Below this, the "Administration" page is visible, featuring sections for "Registered Users" and "Customer Feedback".

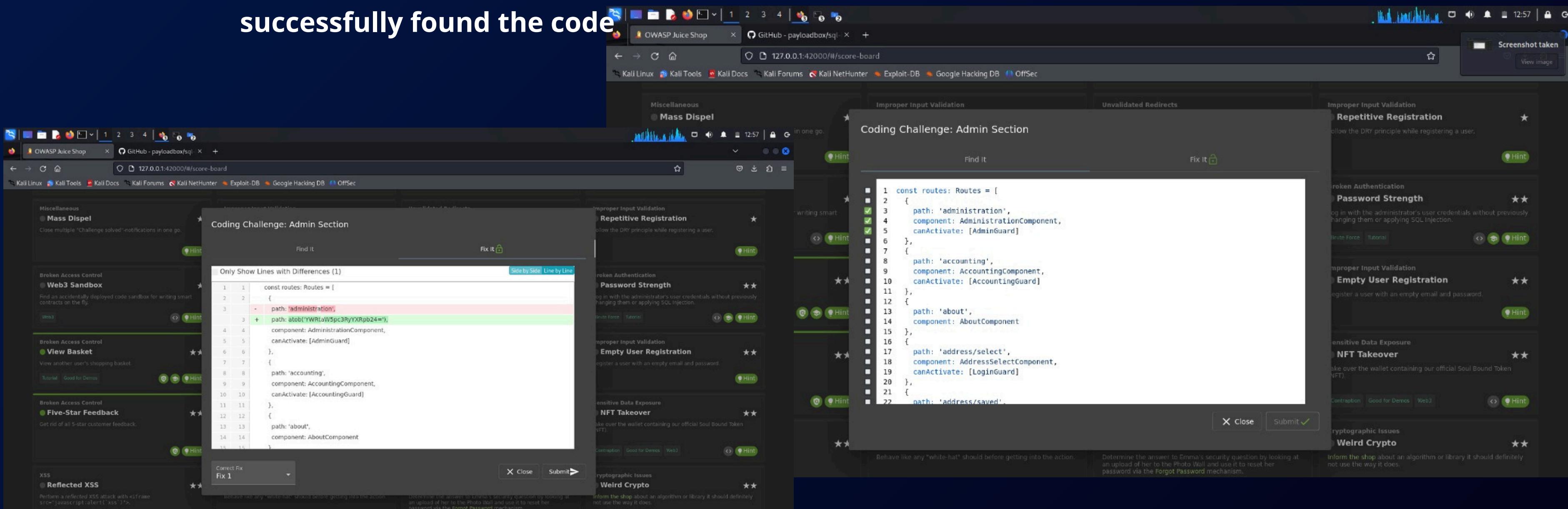
User	Feedback	Rating
admin@juice-sh.op	I love this shop! Best products in town! Highly recommended! (**in@juice-sh.op)	★★★★★
jim@juice-sh.op	Great shop! Awesome service! (**@juice-sh.op)	★★★★★
bender@juice-sh.op	Nothing useful available here! (**der@juice-sh.op)	★
bjoern.kimmich@gmail.com	Incompetent customer support! Can't even upload photo of broken purchase!...	★★
ciso@juice-sh.op	This is the store for awesome stuff of all kinds! (anonymous)	★★★★★
support@juice-sh.op	Never gonna buy anywhere else from now on! Thanks for the great service! (anonymous)	★★★★★
morty@juice-sh.op	Keep up the good work! (anonymous)	★★★
mc.safesearch@juice-sh.op		
J12934@juice-sh.op		
wurstbrot@juice-sh.op		

Bottom Window (Left): This window shows the same "Administration" page, but the "Customer Feedback" section is empty, indicating that the 5-star feedback has been removed.

User	Feedback	Rating
admin@juice-sh.op	I love this shop! Best products in town! Highly recommended! (**in@juice-sh.op)	★★★★★
jim@juice-sh.op	Great shop! Awesome service! (**@juice-sh.op)	★★★★★
bender@juice-sh.op	Nothing useful available here! (**der@juice-sh.op)	★
bjoern.kimmich@gmail.com	Incompetent customer support! Can't even upload photo of broken purchase!...	★★
ciso@juice-sh.op	This is the store for awesome stuff of all kinds! (anonymous)	★★★★★
support@juice-sh.op	Never gonna buy anywhere else from now on! Thanks for the great service! (anonymous)	★★★★★
morty@juice-sh.op	Keep up the good work! (anonymous)	★★★
mc.safesearch@juice-sh.op		
J12934@juice-sh.op		
wurstbrot@juice-sh.op		

6- admin section

we login as an admin and search for administration and we will have the admin section and it will allow us to see all registered user and then we will choose the code in admin section and we will select line 3 4 5 and submit it will tell us that it is we had successfully found the code



The screenshot shows the OWASP Juice Shop application interface. In the center, there is a modal window titled "Coding Challenge: Admin Section". The challenge description states: "Close multiple 'Challenge solved'-notifications in one go." Below this, there is a code editor with the following content:

```
1 const routes: Routes = [
2   {
3     - path: 'administration',
4     + path: atob('YWRTaW5pc3RyXRp24='),
5       component: AdministrationComponent,
6       canActivate: [AdminGuard]
7     },
8     {
9       path: 'accounting',
10      component: AccountingComponent,
11      canActivate: [AccountingGuard]
12    },
13    {
14      path: 'about',
15      component: AboutComponent
16    },
17    {
18      path: 'address/select',
19      component: AddressSelectComponent,
20      canActivate: [LoginGuard]
21    },
22    {
23      path: 'address/saved'.
24    }
25 ]
```

At the bottom of the code editor, there are "Close" and "Submit" buttons. The "Submit" button has a green checkmark icon next to it. To the right of the code editor, there is a list of other challenges, each with a star rating and a brief description.

OWASP Juice Shop GitHub - payloadbox/sqli 127.0.0.1:42000/#/score-board

CPU usage: 8.0%

Miscellaneous
Mass Dispel
Close multiple "Challenge solved"-notifications in one go.

Broken Access Control
Web3 Sandbox
Find an accidentally deployed code sandbox for writing smart contracts on the fly.

Broken Access Control
View Basket
View another user's shopping basket.

Broken Access Control
Five-Star Feedback
Get rid of all 5-star customer feedback.

XSS
Reflected XSS
Perform a reflected XSS attack with <iframe src="javascript:alert('xss')">.

Coding Challenge: Admin Section
Only Show Lines with Differences (1)
Side by Side Line by Line

```
1 1 const routes: Routes = [
2 2 {
3 3   path: 'administration',
4 4     component: AdministrationComponent,
5 5       canActivate: [AdminGuard]
6 6   },
7 7   {
8 8     path: 'accounting',
9 9       component: AccountingComponent,
10 10      canActivate: [AccountingGuard]
11 11   },
12 12   {
13 13     path: 'about',
14 14       component: AboutComponent
15 15   }
16 16 ]
```

Find It Fix It

Improper Input Validation
Repetitive Registration
Follow the DRY principle while registering a user.

Broken Authentication
Password Strength
Log in with the administrator's user credentials without previously changing them or applying SQL Injection.

Improper Input Validation
Empty User Registration
Register a user with an empty email and password.

Sensitive Data Exposure
NFT Takeover
Take over the wallet containing our official Soul Bound Token (SFT).

Cryptographic Issues
Weird Crypto
Inform the shop about an algorithm or library it should definitely not use the way it does.

OWASP Juice Shop

You successfully solved a challenge: View Basket (View another user's shopping basket.)

5% Hacking Challenges

2% Coding Challenges

6/168 Challenges Solved

admin@juice-shop

Orders & Payment Privacy & Security Logout

1/28 5/22 0/43

0/37 0/24 0/14

Firefox ESR GitHub - payloadbox/sqli 127.0.0.1:42000/#/score-board

Miscellaneous
Mass Dispel
Close multiple "Challenge solved"-notifications in one go.

Broken Access Control
Web3 Sandbox
Find an accidentally deployed code sandbox for writing smart contracts on the fly.

Broken Access Control
View Basket
View another user's shopping basket.

Broken Access Control
Five-Star Feedback
Get rid of all 5-star customer feedback.

XSS
Reflected XSS
Perform a reflected XSS attack with <iframe src="javascript:alert('xss')">.

Coding Challenge: Admin Section
Find It Fix It

```
1 const routes: Routes = [
2   {
3     path: 'administration',
4       component: AdministrationComponent,
5         canActivate: [AdminGuard]
6   },
7   {
8     path: 'accounting',
9       component: AccountingComponent,
10      canActivate: [AccountingGuard]
11   },
12   {
13     path: 'about',
14       component: AboutComponent
15   },
16   {
17     path: 'address/select',
18       component: AddressSelectComponent,
19         canActivate: [LoginGuard]
20   },
21   {
22     path: 'address/saved'.
23   }
24 ]
```

Find It Close Submit ✓

Improper Input Validation
Unvalidated Redirects
Improper Input Validation
Repetitive Registration
Follow the DRY principle while registering a user.

Broken Authentication
Password Strength
Log in with the administrator's user credentials without previously changing them or applying SQL Injection.

Improper Input Validation
Empty User Registration
Register a user with an empty email and password.

Sensitive Data Exposure
NFT Takeover
Take over the wallet containing our official Soul Bound Token (SFT).

Cryptographic Issues
Weird Crypto
Inform the shop about an algorithm or library it should definitely not use the way it does.

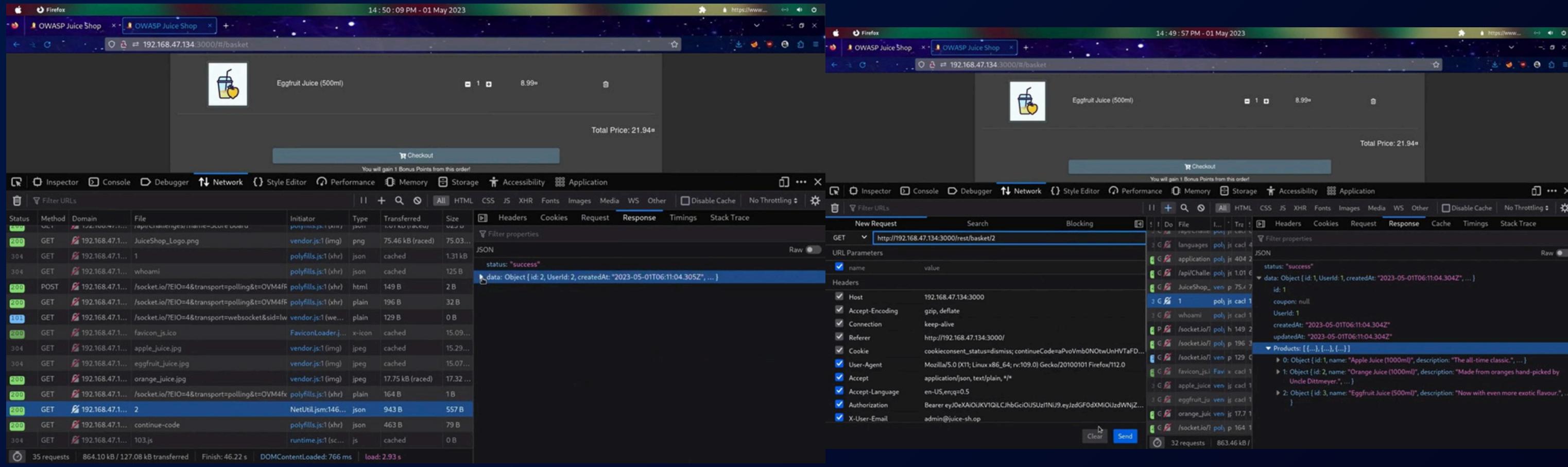
Score Board
XSS
DOM XSS
Find the carefully hidden 'Score Board' page.

Bonus Payload
XSS
Privacy Policy
Read our privacy policy.

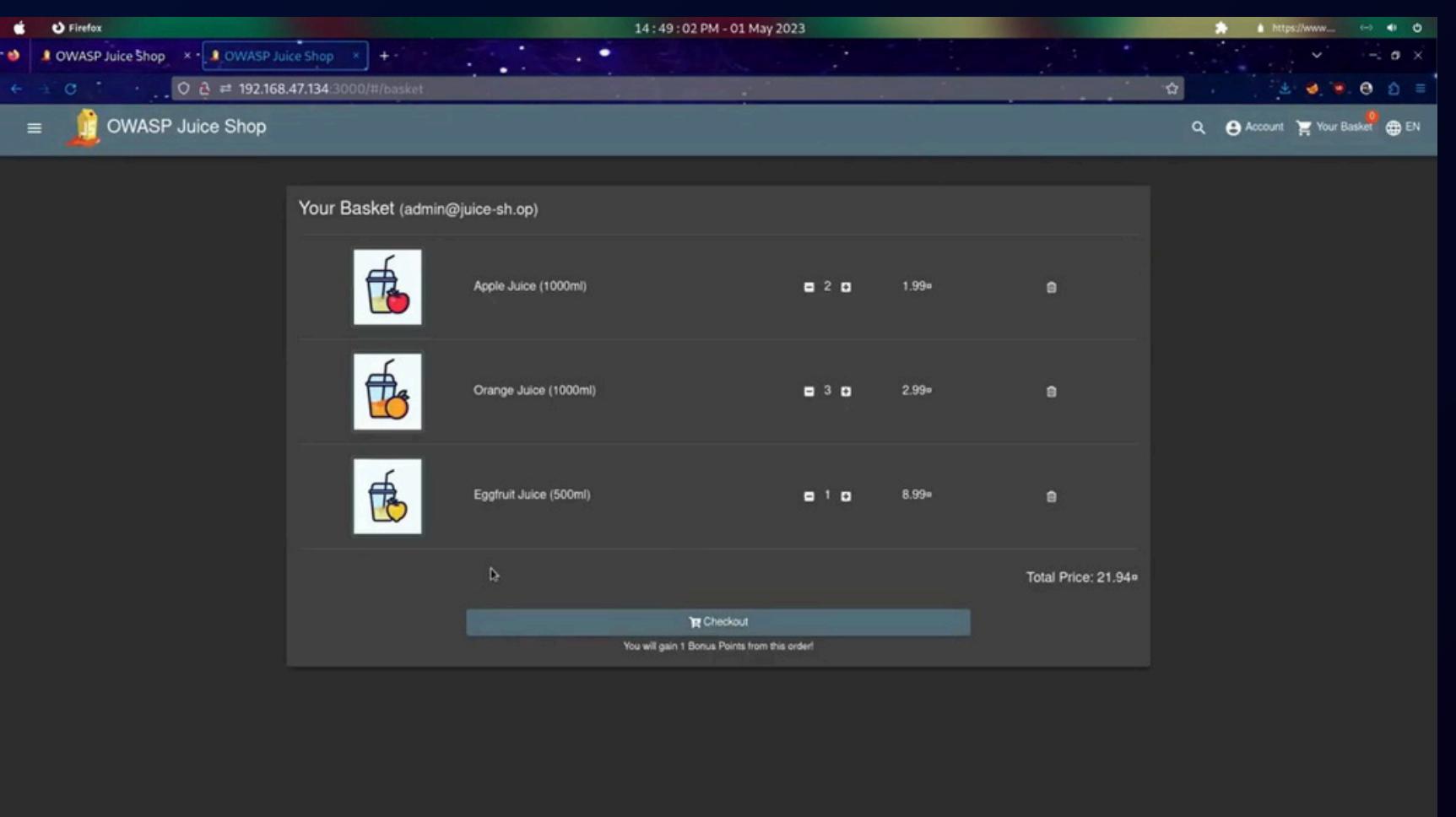
This is the new Score Board! If you notice any bugs or have any feedback, please let us know! Reach out via our community channels

Switch to the legacy Score Board

7- shop view basket



we viewed our shop basket and then we clicked on inspect Q then we clicked on network then reload, and we chose 192.168.47.1 1 and we edited it as 2 and then send and it is done successfully



- Conclusion

Based on the findings, the OWASP Juice Shop application has a high-risk security posture. The identified vulnerabilities highlight the need for immediate remediation and implementation of robust security measures. Regular testing and adherence to security best practices are essential to mitigate future risks.

Next Steps:

1. Prioritize remediation of critical vulnerabilities.
2. Conduct a follow-up penetration test post-remediation.
3. Implement a continuous monitoring strategy to ensure ongoing security