



**Date:** May 2025

**Prepared By:** Sahithi Gudigopuram

#### **Assessment Methodology:**

S.No	Test Methodology	Scope
01	Penetration testing execution standards	It is a structured approach to identifying, exploiting, and reporting security vulnerabilities in systems, networks, or applications.
02	OWASP Web Security Testing Guide v4.2	Testing Juice shop vulnerabilities across the login user profile and APIs using tools like Burp Suite

## Table of Contents

S.No	Title	Page Number
1.	Executive Summary	04
1.1	Summary	04
1.2	Strength	04
1.3	Weakness	05
1.4	Business Impact	05
2.	Technical Summary	06
2.1	Scope & Objective	06
2.2	Testing environment	06
2.3	Testing Environment Overview	09
2.4	Technical Impact	09
2.5	Table of Findings	10
3.	Passive Reconnaissance	11
3.1	Whois	11
3.2	nslookup	11
3.3	Wappalyzer	12
3.4	Whatweb	12
4.	Active Reconnaissance	14
4.1	Nikto	14
4.2	Nmap	15
4.3	Cross-Site Scripting(xss)	16
4.4	SQL Injection	19

4.5	CSRF	22
4.6	Broken Access Control	24
4.7	Security Misconfiguration	26
4.8	Unprotected API	27
5.	Appendix	27
6.	Key Observations	28
7.	Conclusion	28

## 1. Executive summary

---

### 1.1 Summary

A comprehensive penetration test was conducted on the OWASP Juice Shop application, using manual techniques and industry-standard tools. The assessment focused on identifying web vulnerabilities outlined in the OWASP framework. Several high-impact issues were uncovered, including authentication flaws, exposed APIs, and input validation weaknesses, revealing exploitable vectors that threaten application confidentiality and integrity.

### 1.2. Strength

1. **Basic Security Headers Implemented:** Headers like X-Content-Type-Options: nosniff and X-Frame-Options: SAMEORIGIN are in place, helping defend against MIME-type sniffing and clickjacking.
2. **Access Control Enforcement on Admin Routes:** Attempts to access privileged sections like the admin panel were met with proper HTTP 403 responses, showing baseline role-based access control.
3. **HTTPS Ready & No Sensitive Data in Transit:** No indications of sensitive data (passwords, tokens) being transmitted insecurely via GET parameters or exposed in URLs.
4. **No Critical Customer Data Exposure:** During API testing, no personal financial data, credit card numbers, or identity information were leaked.
5. **Modern Tech Stack & Feature Flags:** The app leverages Angular, Node.js, and includes modern headers like feature-policy, demonstrating up-to-date technology use.
6. **Stable Session Management Logic:** While session sharing was discovered, the core login/logout flow was responsive and stable across sessions.
7. **Rich Developer Functionality:** Structured, RESTful API endpoints make the app test-friendly and modular, simplifying debugging and future hardening.

### **1.3. Weakness**

1. **Unprotected API Endpoint:** /api/Challenges exposed sensitive challenge data without authentication.
2. **Weak Password Reset Logic:** Tokens weren't clearly validated same reset flow was shown to all users.
3. **IDOR (Insecure Direct Object Reference):** Manipulating object IDs (like basket IDs) gave access to other users' data.
4. **Cross-Site Scripting (XSS):** Reflected, stored, and DOM-based XSS all successfully executed.
5. **SQL Injection:** Auth bypass via ' OR 1=1 -- and error-based injection vectors confirmed.
6. **CSRF Vulnerability:** Lack of CSRF protection allowed silent password changes via crafted HTML.
7. **Missing Security Headers:** Absence of key headers like Content-Security-Policy, Strict-Transport-Security, etc.

### **1.4. Business Impact**

1. **Unauthorised Access:** Exploitable flaws like IDOR and SQL Injection can allow attackers to bypass authentication and access other users' data.
2. **Data Leakage & Tampering:** Sensitive information and session tokens can be exposed or manipulated, compromising data integrity and confidentiality.
3. **Account Takeover Risks:** CSRF and weak session management may lead to full user account compromise without their awareness.
4. **Brand & Reputation Damage:** Persistent client-side attacks like XSS can be leveraged for phishing, reducing user trust and confidence.
5. **Regulatory Non-Compliance:** Failure to implement security best practices may lead to violations of data protection standards (e.g., GDPR, PCI-DSS)

## 2. Technical Summary

---

### 2.1. Scope & Objective

The assessment focused on evaluating the security posture of the OWASP Juice Shop application hosted at <http://10.0.0.212:3000>. Testing included critical web components such as login, search, feedback forms, and exposed API endpoints. The internal infrastructure and systems beyond this application were not included in scope. The objective was to identify vulnerabilities through simulated attack techniques, with emphasis on input handling, session management, access control, and authentication mechanisms. Findings were used to recommend actionable security improvements.

### 2.2. Testing Environment

#### Setting up OWASP Juice shop( Kali linux)

- Step 1: Open the terminal and update package repositories “sudo apt update”
- Step 2: Install Node.js (JavaScript runtime) “sudo apt install nodejs npm -y”

```
(root@kali:~/home/kali]
# sudo apt install nodejs npm -y

The following packages were automatically installed and are no longer required:
libbdnnl3 libxmpack0
Use 'sudo apt autoremove' to remove them.

Upgrading:
bulk-extractor      gstreamer1.0-plugins-base  icu-devtools      libgstreamer-plugins-bad1.0-0  libgstreamer1.0-0  libopenexr-3.1-30  libsbcl1  openssl
gstreamer1.0-plugins-bad  gstreamer1.0-plugins-good  libgstreamer-gl1.0-0  libgstreamer-plugins-base1.0-0  libicu-dev    libre2-11   libssl3t64  openssl-provider-legacy

Installing:
nodejs  npm

Installing dependencies:
eslint          node-commandir        node-growl           node-move-concurrently  node-set-blocking
gyp             node-concat-stream    node-gyp            node-ms              node-set-immediate-shim
handlebars       node-console-control-strings  node-has-flag       node-mute-stream
libbdnnl20240722 node-convert-source-map  node-has-unicode     node-n3              node-set-value
libbdnnl3.6      node-copy-concurrently  node-has-value      node-negotiator
libflac14        node-core-js          node-has-values    node-neo-async
libicu76         node-core-js-compat   node-hosted-git-info  node-nopt
libjs-events     node-core-js-pure    node-http-proxy-agent  node-normalize-package-data
libjs-inherits   node-core-util-is   node-https-proxy-agent  node-normalize-path
libjs-is-typedarray node-corepack        node-iconv-lite     node-npm-bundled
node-coveragealls  node-coveralls      node-icss-utils     node-npm-package-arg
node-regenerate   node-css-loader     node-ieee754        node-npm-run-path
node-source-map   node-css-selector-tokenizer  node-iferr          node-npmlog
node-sprintfjs    node-data-uri-to-buffer  node-ignore         node-object-assign
node-typedarray-to-buffer  node-debounce-equals-to-primitive  node-imurmurhash  node-object-inspect
node-util         node-debug          node-indent-string  node-object-visit
node-dev         node-decamelize      node-inflight      node-once
node-node115      node-decompress-response  node-inherits      node-opener
node-bonxnruntime1.21  node-deep-equal    node-ini            node-optimist
node-bonxnzh264-8  node-deep-is        node-interpret     node-operatorator
node-dev         node-defaults      node-ip            node-osenv
node-node115      node-define-properties  node-ip-regex     node-p-cancelable
node-abrev        node-define-property  node-is-arrayish  node-p-limit
node-acorn        node-defined      node-is-binary-path  node-p-locate
node-agent-base   node-del          node-is-buffer     node-p-map
node-ajv          node-delegates    node-is-descriptor  node-parse-json
node-ajv-keywords  node-depd         node-is-extensible  node-pascalcase
node-amiproject-remapping  node-diff        node-is-extglob    node-path-dname
node-ansi-escapes  node-doctrine    node-is-eqlsh     node-path-exists
node-nanparser

The menu bar points outside or press Ctrl+Alt]
```

- Step 3: npm (Node package manager)“sudo apt install npm -y”

```
(root㉿kali)-[~/home/kali]
└─# sudo apt install npm -y

npm is already the newest version (9.2.0~ds1-3).
The following packages were automatically installed and are no longer required:
  libdnnl3  libxnnpack0
Use 'sudo apt autoremove' to remove them.

Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 1973

(root㉿kali)-[~/home/kali]
└─# npm -v

9.2.0
```

- Step 4: Copy the official repository URL from GitHub and clone it
- Now OWASP JuiceShop is installed

```
(root㉿kali)-[~/home/kali]
└─# git clone https://github.com/juice-shop/juice-shop.git
Cloning into 'juice-shop'...
remote: Enumerating objects: 138115, done.
remote: Total 138115 (delta 0), reused 0 (delta 0), pack-reused 138115 (from 1)
Receiving objects: 100% (138115/138115), 246.54 MiB | 26.56 MiB/s, done.
Resolving deltas: 100% (107977/107977), done.

(root㉿kali)-[~/home/kali]
└─# ls
create_jwt.py  Desktop  Downloads  juice-shop  Music  Pictures  requesttxt
csrf           Documents  DVWA      jwt-demo.html  myenv  Public    splunk-9.2.1-ae6821b2f2ec-linux-2.6-amd64.deb  Templates  testfile.txt
splunk-9.2.1-deb  splunk-9.4.2.deb  testdir  Videos

(root㉿kali)-[~/home/kali]
└─#
```

- Step 5: Change to the Juice Shop directory “cd juice-shop”
- Step 6: Now, install all project dependencies “npm install”

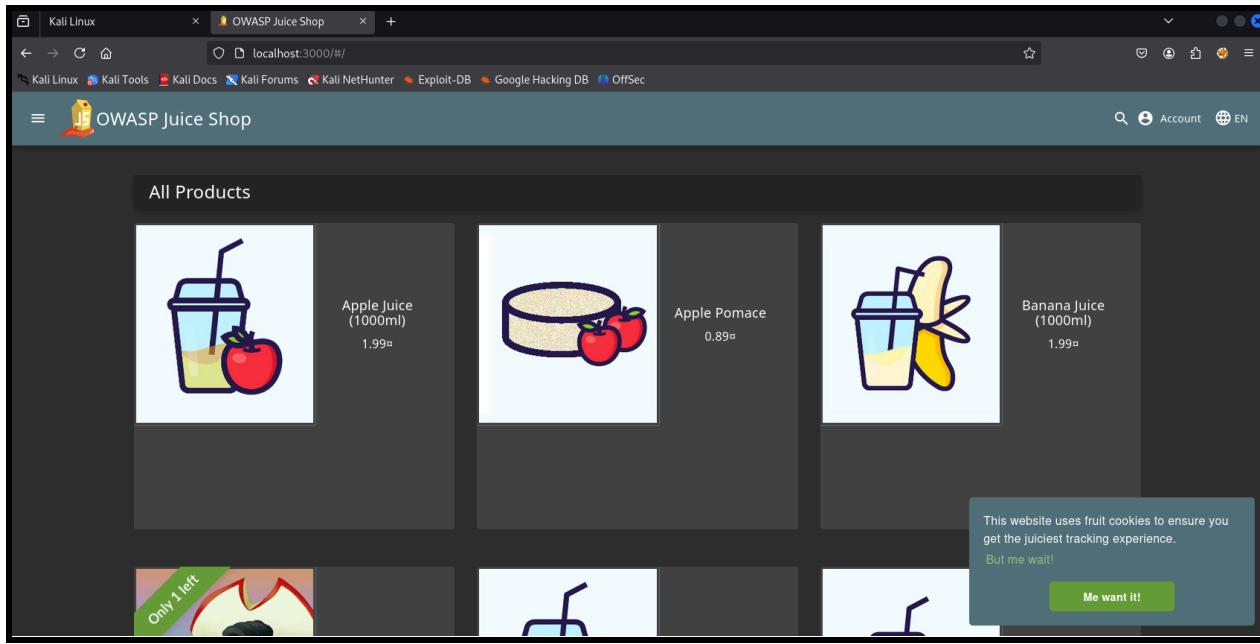
```
(root㉿kali)-[~/home/kali/juice-shop]
└─# npm install
npm WARN EBAOENGINE Unsupported engine {
npm WARN   EBAOENGINE   package: "libxmljs@0.37.0",
npm WARN   EBAOENGINE   required: { node: '>=22' },
npm WARN   EBAOENGINE   current: { node: 'v28.19.2', npm: '9.2.0' }
npm WARN EBAOENGINE } deprecated inflight@1.0.0: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, which is more comprehensive and powerful.
npm WARN deprecated @npmcli/move-file@1.1.2: This functionality has been moved to @npmcli/fs
npm WARN deprecated formatio@1.1.1: This package is unmaintained. Use @sinonjs/formatio instead
npm WARN deprecated evinidjected@0.1.1: Use @evifly/evinid instead
npm WARN deprecated rmsmash@1.2: This package has been deprecated in favour of @sinonjs/samsam
npm WARN deprecated source-map-url@0.4.1: See https://github.com/lydell/source-map-url#deprecated
npm WARN deprecated urix@1.0: Please use https://github.com/lydell/urix#deprecated
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated lodash.get@4.4.2: This package is deprecated. Use the optional chaining (?) operator instead.
npm WARN deprecated source-map-resolve@0.5.3: See https://github.com/lydell/source-map-resolve#deprecated
npm WARN deprecated fstream@0.0.12: This package is no longer supported.
npm WARN deprecated are-we-there-yet@1.7: This package is no longer supported.
npm WARN deprecated gauge@2.7.4: This package is no longer supported.
npm WARN deprecated jws@0.2.6: Security update: Versions below 3.0.0 are deprecated.
npm WARN deprecated npmlog@4.1.2: This package is no longer supported.
npm WARN deprecated vm2@3.0.17: The library contains critical security issues and should not be used for production! The maintenance of the project has been discontinued. Consider migrating your code to ed-vm.
npm WARN deprecated rimraf@3.0.2: Rimraf versions prior to v4 are no longer supported
npm WARN deprecated messageformat@2.3.0: Package renamed as '@messageformat/core', see messageformat.github.io for more details. '@messageformat@4' will eventually provide a polyfill for Intl.MessageFormat & it's been defined by Unicode & ECMA.
npm WARN deprecated @humanahocodes/config-array@0.13.0: Use @eslint/config-array instead
npm WARN deprecated @humanahocodes/object-schema@0.3: Use @eslint/object-schema instead
npm WARN deprecated ectstatic@3.3.2: This package is unmaintained and deprecated. See the GH Issue 259.
npm WARN deprecated osenv@0.1.5: This package is no longer supported.
npm WARN deprecated @types/socket.io-parser@3.0.0: This is a stub types definition. socket.io-parser provides its own type definitions, so you do not need this installed.
npm WARN deprecated @types/express-unless@0.3: This is a stub types definition. express-unless provides its own type definitions, so you do not need this installed.
npm WARN deprecated eslint-config-standard-with-tvnescript@3.0.1: Please use eslint-config-love. instead.
```

```
npm WARN deprecated eslint@8.57.1: This version is no longer supported. Please see https://eslint.org/version-support for other options.  
added 1465 packages, and audited 1466 packages in 1m  
344 packages are looking for funding  
  run `npm fund` for details  
18 vulnerabilities (15 moderate, 3 high)  
To address issues that do not require attention, run:  
  npm audit fix  
To address all issues (including breaking changes), run:  
  npm audit fix --force  
Run `npm audit` for details.  
> juice-shop@18.0.0 build:frontend  
> cd frontend && npm run build  
  
> frontend@18.0.0 build  
> ng build --configuration production
```

- Compile the frontend and backend (TypeScript → JavaScript) “npm run build: frontend  
npm run build: server”
- Launch the application “npm start”

```
└─(root㉿kali)-[~/home/kali/juice-shop]  
# npm run build:server  
  
> juice-shop@18.0.0 build:server  
> tsc  
  
└─(root㉿kali)-[~/home/kali/juice-shop]  
# npm start  
  
> juice-shop@18.0.0 start  
> node build/app  
  
info: Detected Node.js version v20.19.2 (OK)  
info: Detected OS linux (OK)  
info: Detected CPU x64 (OK)  
info: Configuration default validated (OK)  
info: Entity models 19 of 19 are initialized (OK)  
info: Required file server.js is present (OK)  
info: Required file index.html is present (OK)  
info: Required file styles.css is present (OK)  
info: Required file main.js is present (OK)  
info: Required file tutorial.js is present (OK)  
info: Required file runtime.js is present (OK)  
info: Required file vendor.js is present (OK)  
info: Port 3000 is available (OK)  
info: Chatbot training data botDefaultTrainingData.json validated (OK)  
info: Domain https://www.alchemy.com/ is reachable (OK)  
info: Server listening on port 3000
```

- Open browser “<http://localhost:3000>”



## 2.3. Testing Environment Overview

Component	Details
Target Application	The OWASP Juice Shop host is located on Kali Linux IP 10.0.0.212
Attacker Machine	An Ubuntu-based system used to perform reconnaissance and exploitation
Network setup	Internal lab using Bridged Adapter
Browsers & tools	Firefox, Burp suite, Nmap, Nikto, wappalzer..etc
Isolation	The lab was segmented from the external network to simulate a secure testing scope.

## 2.4. Technical Impact

The vulnerabilities identified during the assessment pose significant technical risks to the confidentiality, integrity, and availability of the application. Exploits such as SQL Injection can allow unauthorised database access or manipulation, while XSS and CSRF attacks enable remote attackers to hijack sessions and perform unauthorised actions on behalf of users. IDOR and broken access controls further permit exposure or modification of user-specific resources. Together, these flaws could undermine secure session management, leak sensitive data, and facilitate attacker persistence within the application.

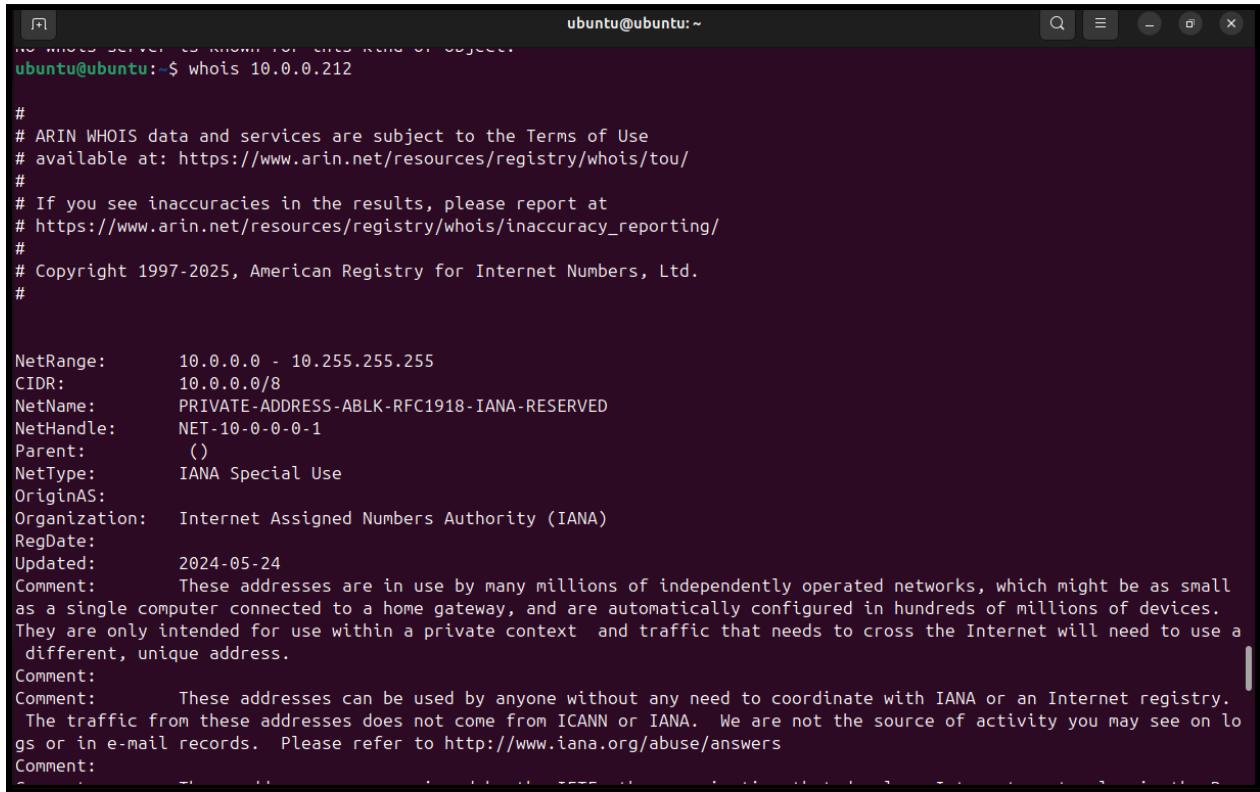
## **2.5. Table of Findings**

S.No	Category	Vulnerability	Risk level
1	Cross-site scripting(XSS)	Reflected, stored, DOM-based XSS	High
2	Injection	SQL Injection	High
3	Cross-site request forgery	CSRF on password change	High
4	Access Control	IDOR(Insecure direct object references)	Hight
5	Security Misconfiguration	Missing Security Headers	Medium
6	Unprotected API	/api/challenges exposed	Medium
7	Web Server Profiling	Nikto scan	Medium
8	Technology Fingerprinting	Wapplyzer	Low
9	Technology Fingerprinting	Whatweb	Low
10	Network mapping	Nmap scan	Info

### 3. Passive Reconnaissance

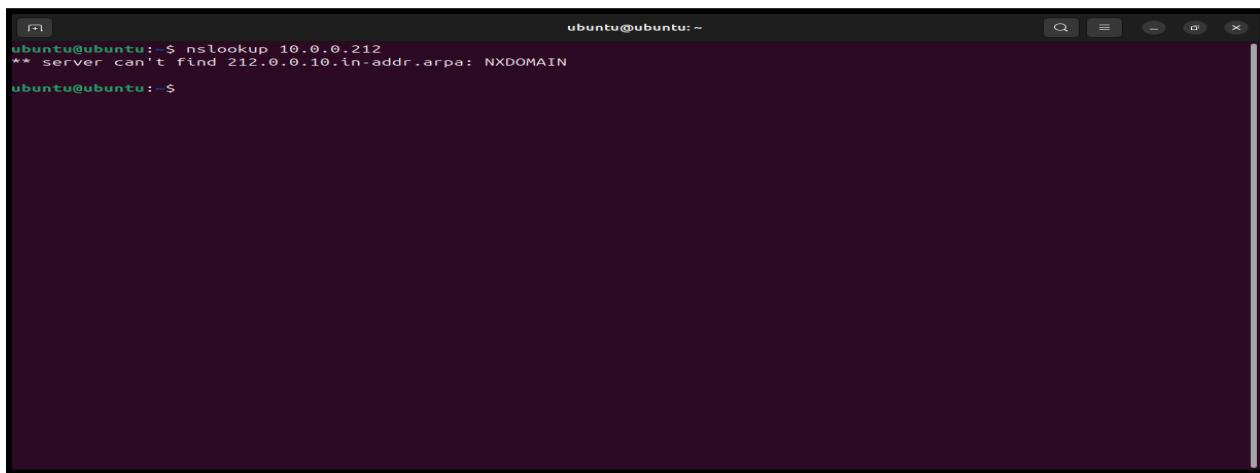
---

**3.1. Whois:** The whois result shows that IP 10.0.0.212 belongs to the private IP range 10.0.0.0/8, reserved by IANA for internal network use. This means it's not assigned to any public organisation and is used in local environments.



```
ubuntu@ubuntu:~$ whois 10.0.0.212
#
# ARIN WHOIS data and services are subject to the Terms of Use
# available at: https://www.arin.net/resources/registry/whois/tou/
#
# If you see inaccuracies in the results, please report at
# https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
# Copyright 1997-2025, American Registry for Internet Numbers, Ltd.
#
NetRange:      10.0.0.0 - 10.255.255.255
CIDR:         10.0.0.0/8
NetName:       PRIVATE-ADDRESS-ABLK-RFC1918-IANA-RESERVED
NetHandle:     NET-10-0-0-0-1
Parent:        ()
NetType:       IANA Special Use
OriginAS:
Organization:  Internet Assigned Numbers Authority (IANA)
RegDate:
Updated:       2024-05-24
Comment:       These addresses are in use by many millions of independently operated networks, which might be as small as a single computer connected to a home gateway, and are automatically configured in hundreds of millions of devices. They are only intended for use within a private context and traffic that needs to cross the Internet will need to use a different, unique address.
Comment:       These addresses can be used by anyone without any need to coordinate with IANA or an Internet registry. The traffic from these addresses does not come from ICANN or IANA. We are not the source of activity you may see on logs or in e-mail records. Please refer to http://www.iana.org/abuse/answers
Comment:
```

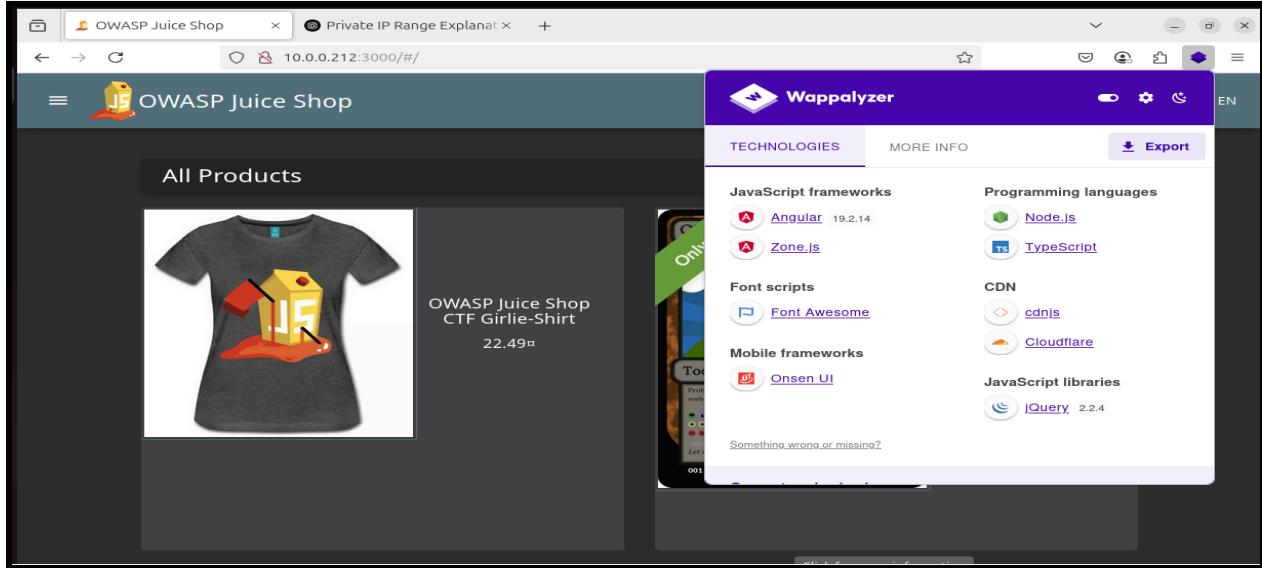
**3.2. nslookup:** This lookup attempted a reverse DNS resolution and still returned NXDOMAIN, which is also expected in most internal or lab setups unless you've configured reverse DNS.



```
ubuntu@ubuntu:~$ nslookup 10.0.0.212
** server can't find 212.0.0.10.in-addr.arpa: NXDOMAIN
ubuntu@ubuntu:~$
```

**3.3. Wappalyzer:** The target web application is **OWASP Juice Shop**, built with **Angular**

**19.2.14, Node.js**, and **TypeScript**, and includes outdated components such as **jQuery 2.2.4**, presenting potential vectors for exploitation and security testing.



#### **3.4. WhatWeb:**

It is a passive reconnaissance tool used to identify technologies powering a web application by analysing HTTP headers and page content. It detected components such as HTML5, jQuery 2.2.4, and headers like X-Frame-Options and Feature-Policy. This non-intrusive scan provided insights into the application's framework and potential outdated libraries. It helped establish the technology stack and informed later vulnerability assessments.

```

Processing triggers for libc-bin (2.39-0ubuntu8.4) ...
root@sahithi-VMware-Virtual-Platform:/home/sahithi# whatweb http://10.0.0.212:3000
http://10.0.0.212:3000 [200 OK] Country[RESERVED][ZZ], HTML5, IP[10.0.0.212], JQuery[2.2.4], Script[module], Title[OWASP Juice Shop], UncommonHeaders[access-control-allow-origin,x-content-type-options,feature-policy,x-recruiting], X-Frame-Options[SAMEORIGIN]
root@sahithi-VMware-Virtual-Platform:/home/sahithi# whatweb -v http://10.0.0.212:3000
WhatWeb report for http://10.0.0.212:3000
Status : 200 OK
Title : OWASP Juice Shop
IP : 10.0.0.212
Country : RESERVED, ZZ

Summary : HTML5, JQuery[2.2.4], Script[module], UncommonHeaders[access-control-allow-origin,x-content-type-options,feature-policy,x-recruiting], X-Frame-Options[SAMEORIGIN]

Detected Plugins:
[ HTML5 ]
    HTML version 5, detected by the doctype declaration

[ JQuery ]
    A fast, concise, JavaScript that simplifies how to traverse
    HTML documents, handle events, perform animations, and add
    AJAX.

    Version : 2.2.4
    Website : http://jquery.com/

[ Script ]
    This plugin detects instances of script HTML elements and
    returns the script language/type.

    String : module

[ UncommonHeaders ]
    Uncommon HTTP server headers. The blacklist includes all

```

```

the standard headers and many non standard but common ones.
Interesting but fairly common headers should have their own
plugins, eg. x-powered-by, server and x-aspmx-version.
Info about headers can be found at www.http-stats.com

String : access-control-allow-origin,x-content-type-options,feature-policy,x-recruiting (from headers)

[ X-Frame-Options ]
    This plugin retrieves the X-Frame-Options value from the
    HTTP header. - More Info:
    http://msdn.microsoft.com/en-us/library/cc288472%28VS.85%29.
    aspx

    String : SAMEORIGIN

HTTP Headers:
    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
    Accept-Ranges: bytes
    Cache-Control: public, max-age=0
    Last-Modified: Tue, 24 Jun 2025 01:53:33 GMT
    ETag: W/"138f5-1979fa3fb0F"
    Content-Type: text/html; charset=UTF-8
    Vary: Accept-Encoding
    Content-Encoding: gzip
    Date: Tue, 24 Jun 2025 02:13:28 GMT
    Connection: close
    Transfer-Encoding: chunked

root@sahithi-VMware-Virtual-Platform:/home/sahithi# whatweb -i urls.txt
No targets selected
root@sahithi-VMware-Virtual-Platform:/home/sahithi# █

```

## 4. Active Reconnaissance

---

### 4.1. Nikto

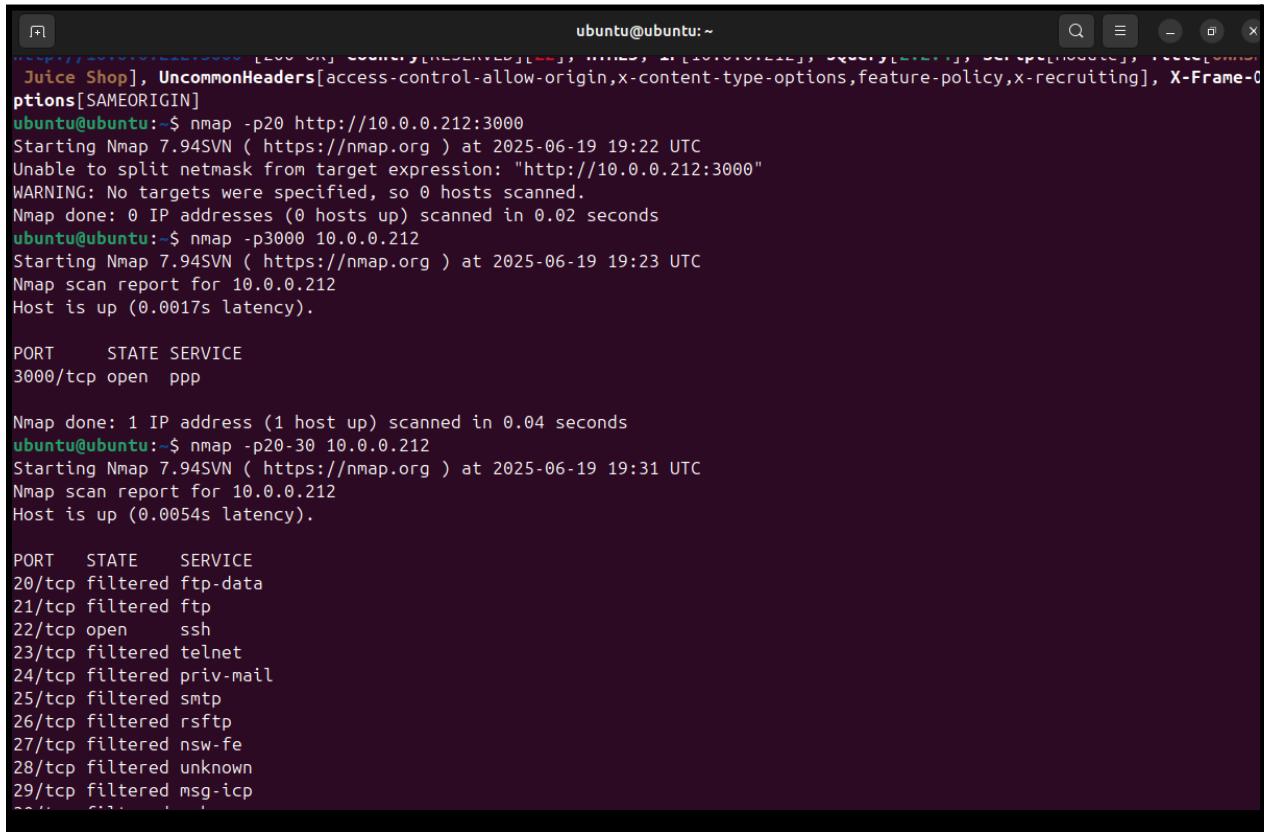
This scan on `http://10.0.0.212:3000` uncovered multiple signs of insecure configuration and information exposure. Publicly accessible paths like `/ftp/`, `/css/`, and `/public/`—some even indexed in `robots.txt`—may unintentionally leak sensitive content or support files. Headers such as `x-recruiting` can expose internal tooling or company tech stacks if linked to job platforms or developer resources, while others like `feature-policy` suggest modern browser behaviour controls are in place. The server discloses inode information through weakly configured ETag headers, aiding potential fingerprinting attempts. Additionally, the permissive **CORS** setting (`Access-Control-Allow-Origin: *`) and absence of a server banner reflect a mix of basic security awareness alongside overlooked risks, warranting further manual inspection of exposed endpoints.

```
Processing triggers for man-db (2.12.0-4build2) ...
ubuntu@ubuntu:~$ nikto -h http://10.0.0.212:3000
- Nikto v2.1.5
-----
+ Target IP:          10.0.0.212
+ Target Hostname:    10.0.0.212
+ Target Port:        3000
+ Start Time:         2025-06-19 20:24:22 (GMT0)
-----
+ Server: No banner retrieved
+ Server leaks inodes via ETags, header found with file /, fields: 0xW/138f5 0x197898f9133
+ Uncommon header 'x-recruiting' found, with contents: #/jobs
+ Uncommon header 'x-content-type-options' found, with contents: nosniff
+ Uncommon header 'feature-policy' found, with contents: payment 'self'
+ Uncommon header 'access-control-allow-origin' found, with contents: *
+ Uncommon header 'x-frame-options' found, with contents: SAMEORIGIN
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ File/dir '/ftp/' in robots.txt returned a non-forbidden or redirect HTTP code (200)
+ "robots.txt" contains 1 entry which should be manually viewed.
+ Uncommon header 'access-control-allow-methods' found, with contents: GET,HEAD,PUT,PATCH,POST,DELETE
+ OSVDB-3092: /css: This might be interesting...
+ OSVDB-3092: /ftp/: This might be interesting...
+ OSVDB-3092: /public/: This might be interesting...
■
```

- **Remediation:** Remove or restrict access to sensitive directories like `/ftp/` and `/public/`, and implement missing headers such as **Content-Security-Policy** and **Strict-Transport-Security** to reduce exposure.

## 4.2. Nmap

I discovered two open ports on the target system (10.0.0212), port 22 running SSH, which could allow secure remote access if valid credentials are found, and port 3000, hosting an unknown service labelled "ppp" that requires further investigation. These findings will guide the next steps in enumeration and potential exploitation.

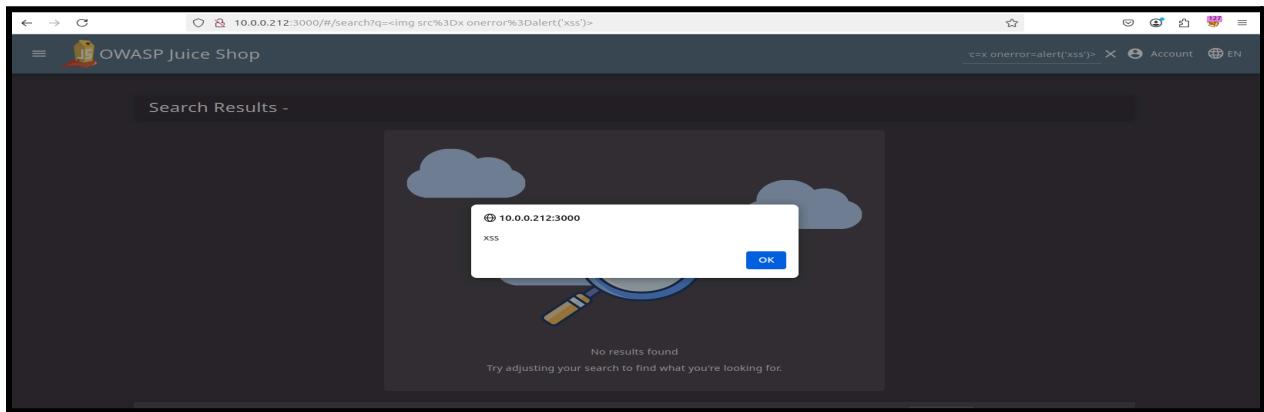


```
ubuntu@ubuntu:~  
...  
Juice Shop], UncommonHeaders[access-control-allow-origin,x-content-type-options,feature-policy,x-recruiting], X-Frame-Options[SAMEORIGIN]  
ubuntu@ubuntu:~$ nmap -p20 http://10.0.0.212:3000  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-19 19:22 UTC  
Unable to split netmask from target expression: "http://10.0.0.212:3000"  
WARNING: No targets were specified, so 0 hosts scanned.  
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.02 seconds  
ubuntu@ubuntu:~$ nmap -p3000 10.0.0.212  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-19 19:23 UTC  
Nmap scan report for 10.0.0.212  
Host is up (0.0017s latency).  
  
PORT      STATE SERVICE  
3000/tcp  open   ppp  
  
Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds  
ubuntu@ubuntu:~$ nmap -p20-30 10.0.0.212  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-06-19 19:31 UTC  
Nmap scan report for 10.0.0.212  
Host is up (0.0054s latency).  
  
PORT      STATE SERVICE  
20/tcp    filtered  ftp-data  
21/tcp    filtered  ftp  
22/tcp    open     ssh  
23/tcp    filtered telnet  
24/tcp    filtered priv-mail  
25/tcp    filtered smtp  
26/tcp    filtered rsftp  
27/tcp    filtered nsw-fe  
28/tcp    filtered unknown  
29/tcp    filtered msg-icp  
2000/tcp  filtered  ...
```

- **Remediation:** Restrict unnecessary open ports (e.g., port 22 if not needed) using firewall rules. Ensure only required services are exposed to trusted networks

### 4.3. Cross-site scripting(xss)

- Reflected XSS in search functionality using <img src=x onerror=alert('xss')> payload.  
**Impact:** Arbitrary JavaScript execution in the user's browser, allowing session hijacking or phishing attacks.



A screenshot of the Burp Suite interface. The 'Repeater' tab is selected. On the left, the 'Request' pane shows a GET request to /products/search?q=. The request includes various headers such as Host, User-Agent, Accept, Accept-Encoding, and Connection. The 'Response' pane shows the corresponding HTTP response with status code 200 OK. The 'Inspector' pane on the right displays the request attributes, query parameters, body parameters, cookies, and headers. The response body contains the reflected XSS payload: &lt;img src=x onerror=alert('xss')&gt;.

- Stored XSS via feedback input using <a href=" javascript: alert('xss')">clickme</a> payload. **Impact:** Malicious JavaScript executes when the victim clicks the injected link, leading to session hijacking

The screenshot shows the OWASP Juice Shop application's 'Customer Feedback' form. The 'Comment' field contains the following XSS payload:

```
<a href="javascript('xss')">clickme</a>
```

The 'Rating' slider is set to 5. The CAPTCHA question is "What is 1+7-2?" and the answer is "5". The 'Submit' button is visible at the bottom.

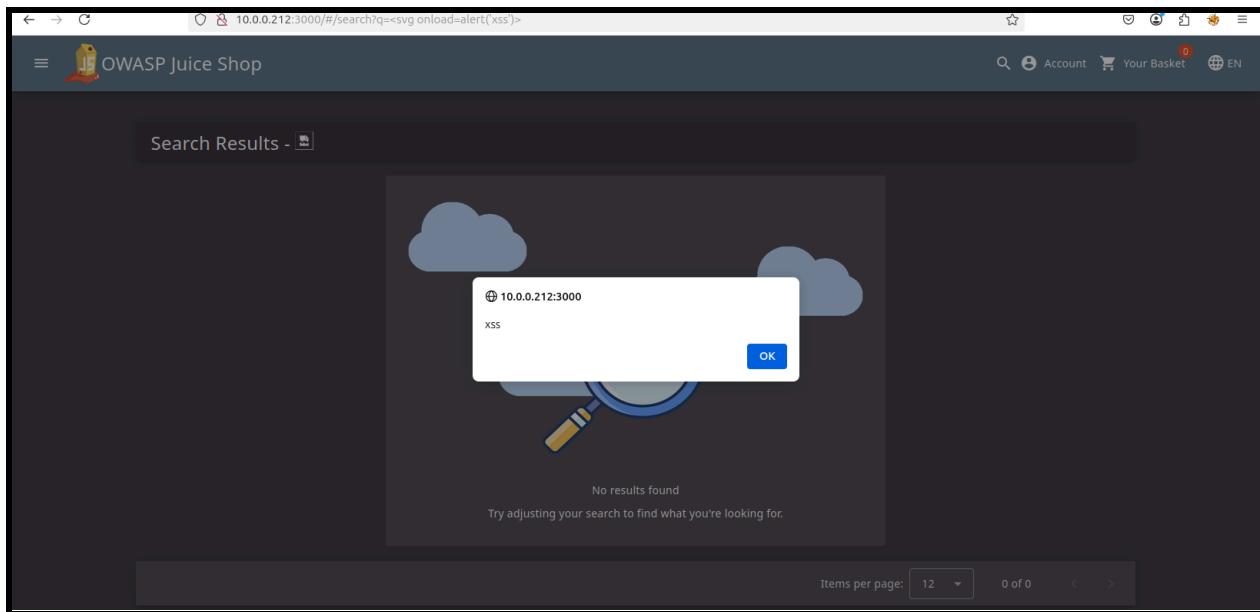
The screenshot shows the Burp Suite interface with the 'Repeater' tab selected. The 'Request' pane displays a POST request to '/api/Feedbacks/' with the following payload:

```
POST /api/Feedbacks/ HTTP/1.1
Host: 10.0.0.212:3000
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:135.0) Gecko/20100101
Accept: application/json, text/plain, */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/json
Content-Length: 97
Origin: http://10.0.0.212:3000
Connection: keep-alive
Referer: http://10.0.0.212:3000/
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss;
continueCode=a40D04KyQoPJ7j2n0vp9EQ38gYVAJLGMlwxa!NDSreZRLzmXk6BbmzPb3
Priority: ue0
{
  "captchaId":0,
  "captcha":"5",
  "comment": "<a href=\"javascript('xss')\">clickme</a> ()",
  "rating":3
}
```

The 'Response' pane shows the JSON response received from the server:

```
HTTP/1.1 201 Created
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Content-Security-Policy: frame-ancestors 'self'
X-Recruiting: #/jobs
Location: /api/Feedbacks/12
Content-Type: application/json; charset=utf-8
Content-Length: 197
ETag: W/"c5-G9!+BD/40BzFJmLG6usyC2NsuIY"
Vary: Accept-Encoding
Date: Sat, 21 Jun 2025 03:31:45 GMT
Connection: keep-alive
Keep-Alive: timeout=5
{
  "status": "success",
  "data": {
    "id": 12,
    "comment": "<a href=\"javascript('xss')\">clickme</a> ()",
    "rating": 3,
    "updatedAt": "2025-06-21T03:31:45.998Z",
    "createdAt": "2025-06-21T03:31:45.998Z",
    "userId": null
  }
}
```

- DOM-based XSS in search feature via q parameter using `<svg onload=alert('xss')>`.  
**Impact:** Payload executes in the browser through client-side injection, allowing arbitrary JavaScript execution.

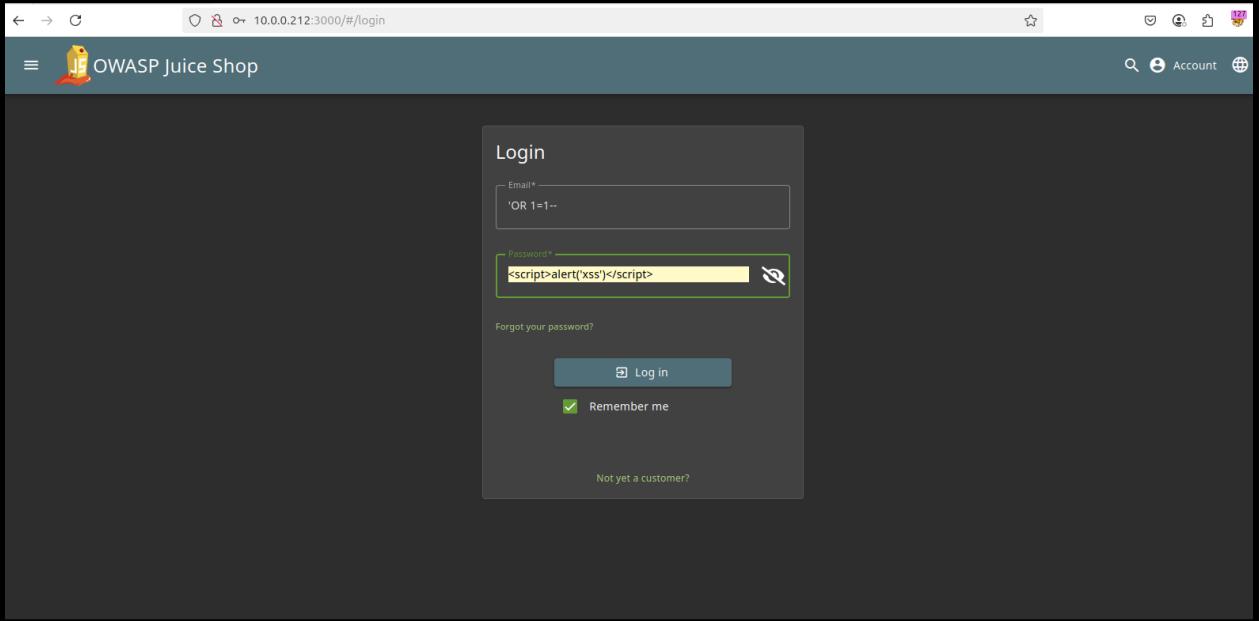


- **Remediation:**

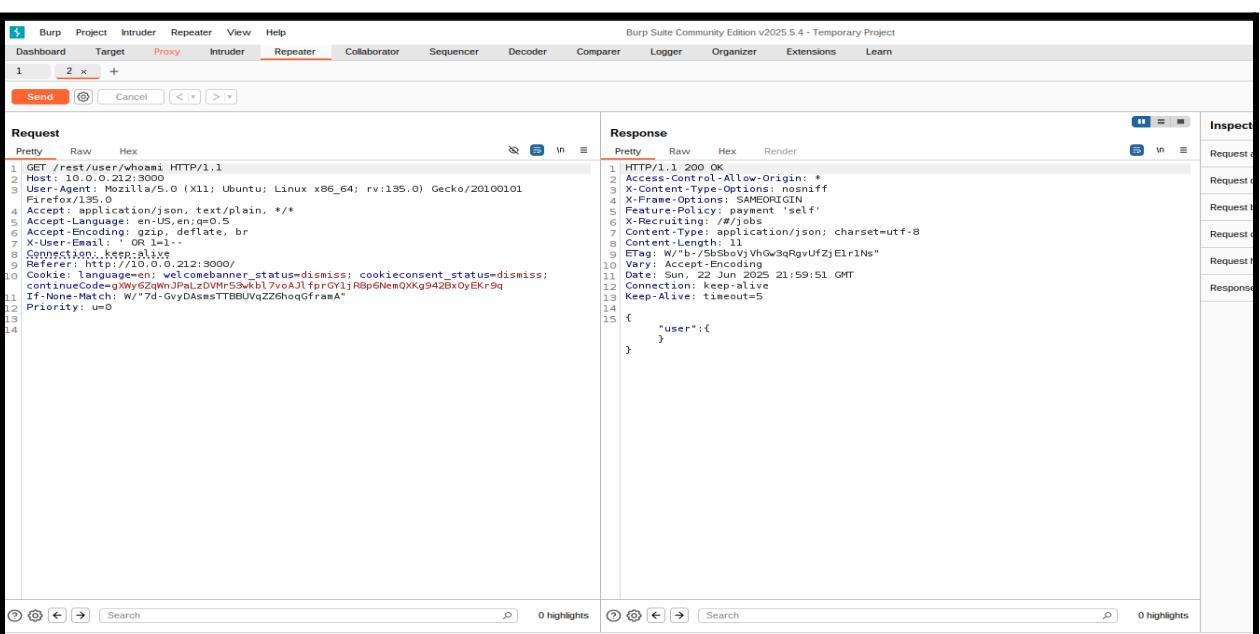
1. **Sanitise and validate all user inputs** on both client and server sides—reject or escape HTML, JavaScript, and script-related characters.
2. **Implement context-aware output encoding** (e.g., HTML, URL, or JavaScript encoding) and deploy a strong Content Security Policy (CSP) to reduce script execution risk.

## 4.4. SQL injection

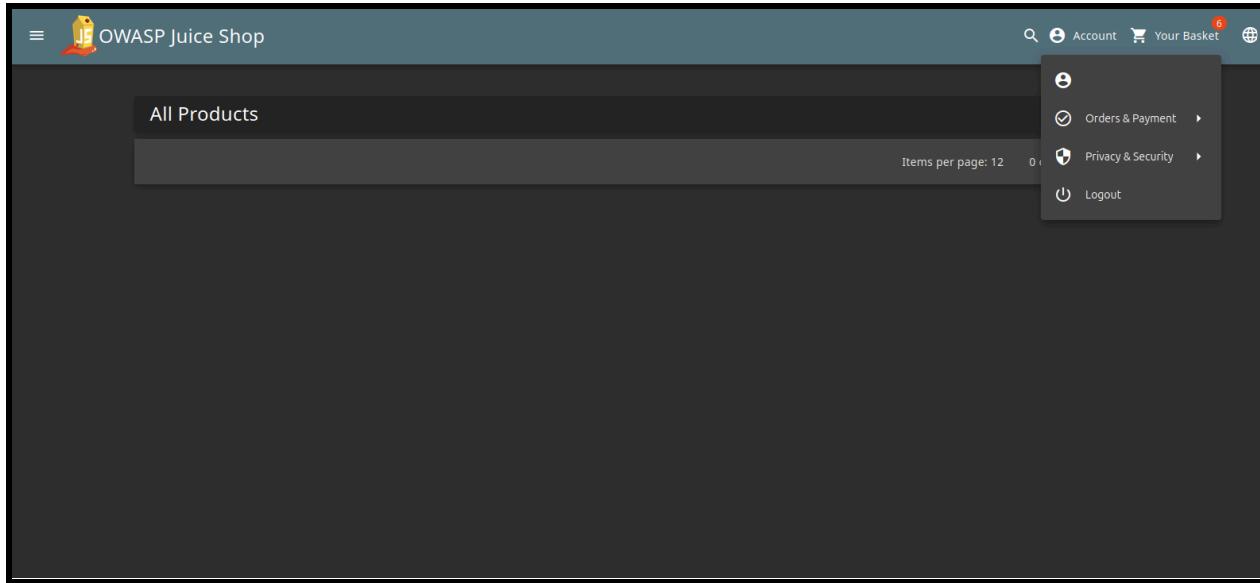
SQL Injection in login form using ' OR 1=1 -- payload to bypass authentication. **Impact:** User gained unauthorised access without valid credentials, demonstrating critical backend query manipulation.



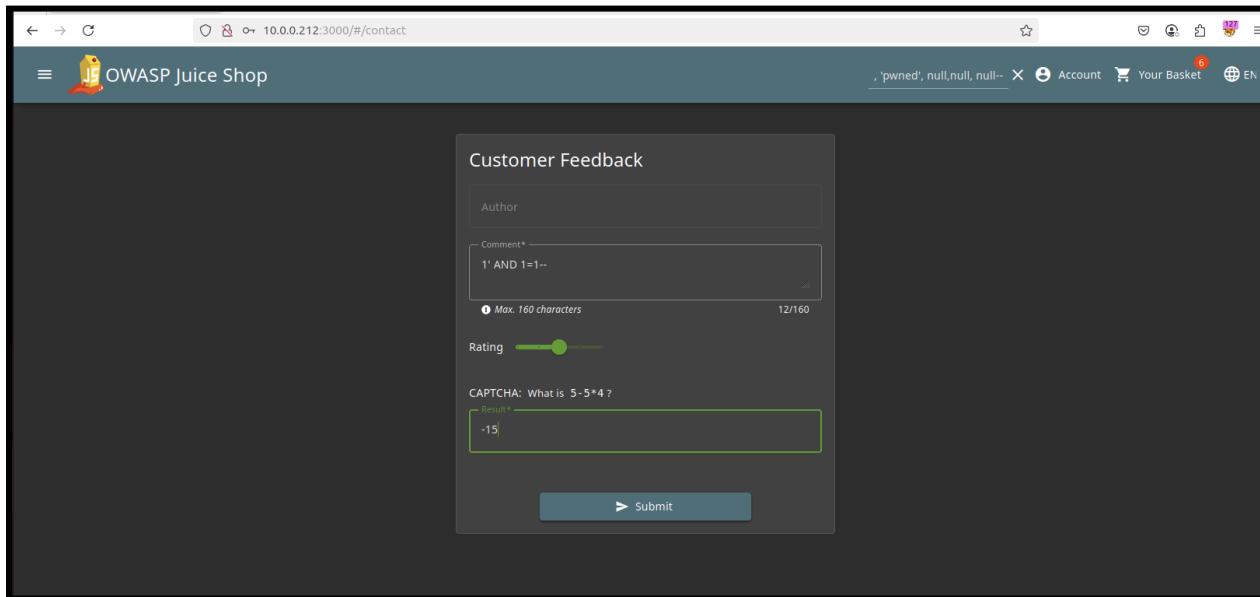
The screenshot shows the OWASP Juice Shop login interface. In the 'Email\*' field, the user has entered "'OR 1=1--". In the 'Password\*' field, they have entered '<script>alert('xss')</script>'. Below the password field, a green status bar indicates "Validating..." and "No XSS detected". The 'Log in' button is visible, along with a 'Remember me' checkbox and a 'Forgot your password?' link.

The screenshot shows the Burp Suite interface with the 'Proxy' tab selected. The 'Request' pane displays a captured GET request to '/rest/user/whoami'. The 'Response' pane shows the server's JSON response, which includes a user object with an email field set to 'OR 1=1--'. The 'Burp Suite Community Edition v2025.5.4 - Temporary Project' header is visible at the top.



- Error-Based SQL Injection Attempt in Feedback API **Payload Used:** '1 AND 1=1-- ()  
**Injection Point:** Comment field in Feedback form **Request Method:** POST to /api/Feedbacks/  
**Observation:** Payload accepted (HTTP 201), echoed back without triggering SQL errors or altering behaviour.



The screenshot shows the Burp Suite NetworkMiner interface. A POST request is captured to the endpoint `/api/Feedbacks`. The response is a `201 Created` status with a JSON payload containing an ID, rating, and user ID. The response body is as follows:

```
HTTP/1.1 201 Created
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Location: /api/Feedbacks/1
Content-Type: application/json; charset=utf-8
Content-Length: 165
ETag: "a5d492d4c1e190spkXGVTWm7cw"
Date: Sun, 22 Jun 2025 22:19:34 GMT
Connection: keep-alive
Keep-Alive: timeout=5
{
    "status": "success",
    "data": {
        "id": 12,
        "rating": 1 AND l=1-- (),
        "userId": 1,
        "updatedAt": "2025-06-22T22:19:34.049Z",
        "createdAt": "2025-06-22T22:19:34.049Z"
    }
}
```

OWASP Juice Shop

10.0.0.212:3000/#/contact

OWASP Juice Shop

Customer Feedback

Author

Comment\*

Max. 160 characters  
0/160

Rating

CAPTCHA: What is  $5 \cdot 5 + 4$  ?

Result\*

Submit

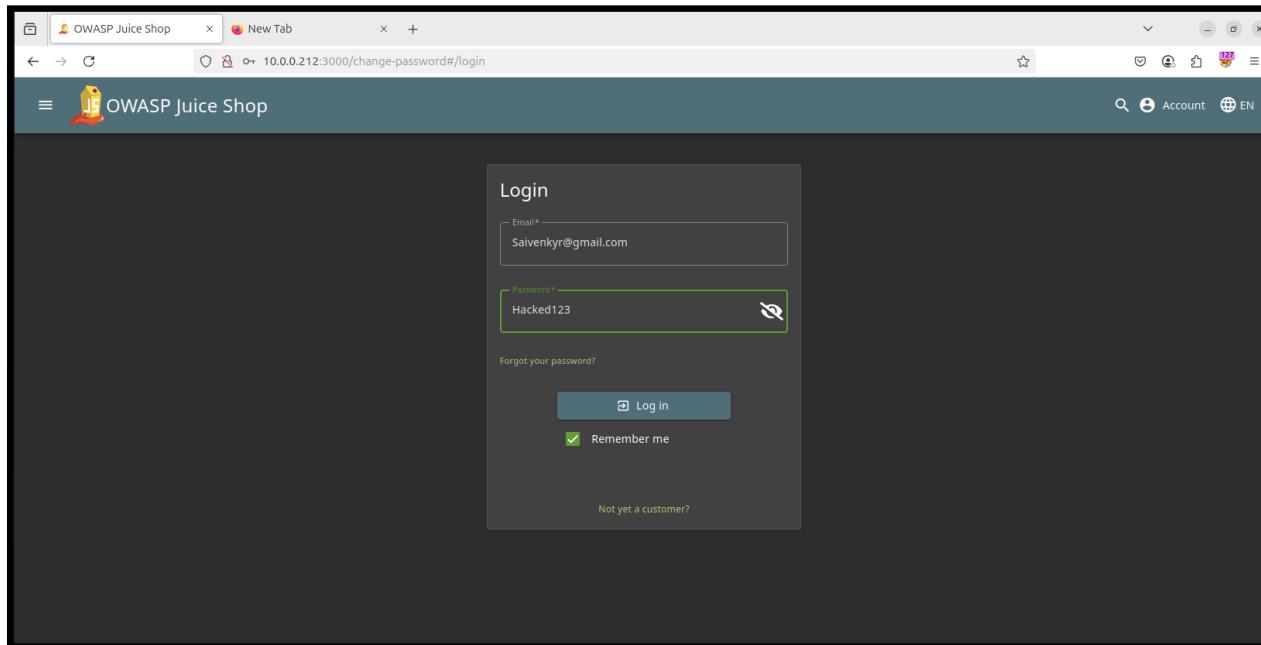
Thank you for your feedback.

- **Remediation:**

1. Use parameterised queries or prepared statements across all database interactions to prevent injection of malicious SQL code. Avoid dynamic query construction with direct user input.
  2. Implement input validation and least privilege principles—ensure only expected input is accepted, and the database account used by the application has limited permissions.

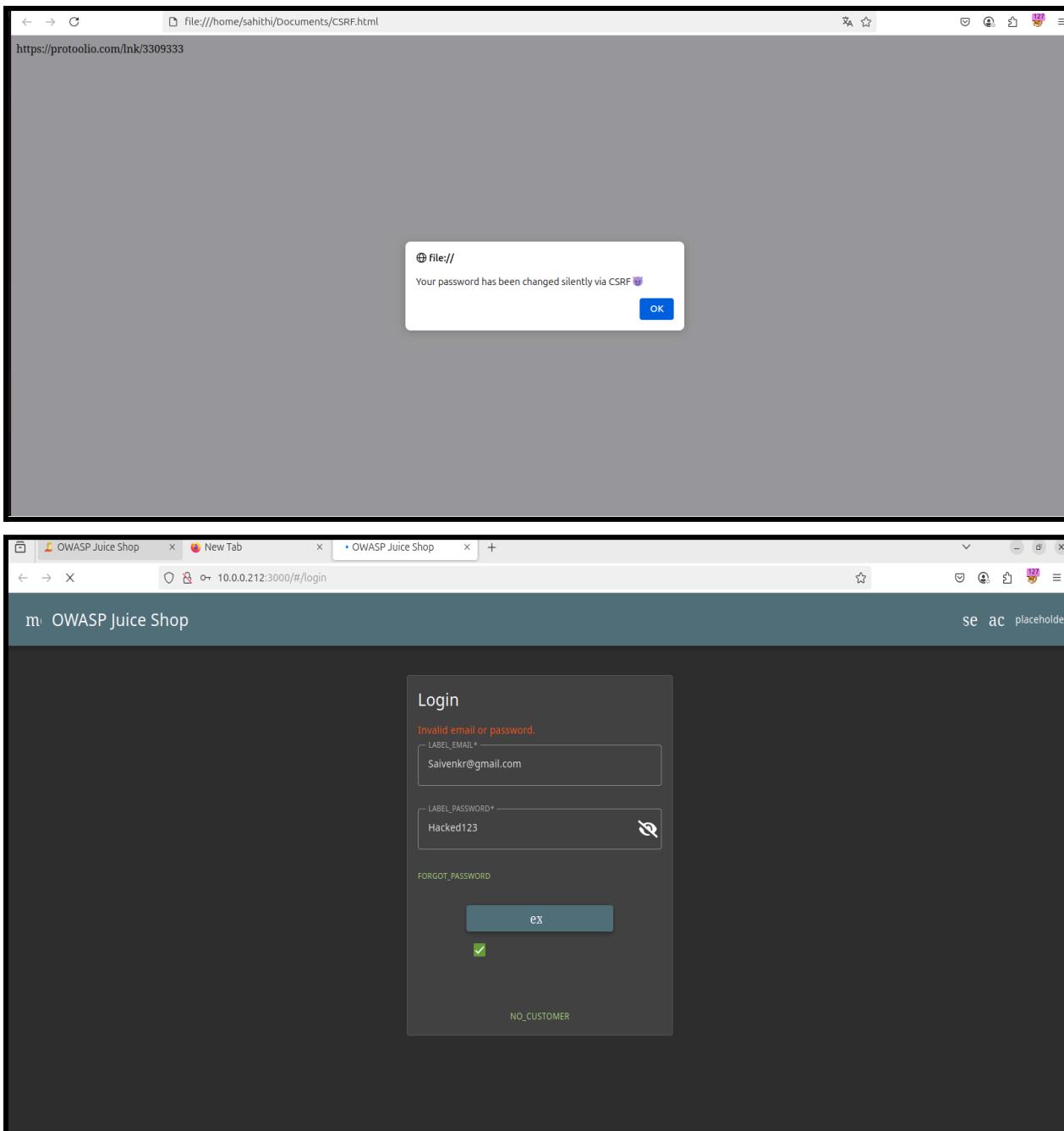
## 4.5. CSRF

The password change endpoint in Juice Shop is vulnerable to CSRF, allowing unauthorised modification of a user's credentials without their knowledge. A crafted local HTML file (CSRF.html) silently triggers the exploit when opened during an active session, confirming a lack of CSRF protection on sensitive functionality.



A screenshot of the Burp Suite interface. The 'Proxy' tab is selected, showing a captured request and response. The request is a GET to /rest/user/change-password?current=Hacked123&amp;new=Hacked@123&amp;repeat=Hacked@123. The response is a 200 OK status with various headers and a JSON payload containing a user object with email, password, role (customer), and other details. The 'Raw' tab shows the full JSON response.

If I clicked on this path, it is showing me “file:///home/sahithi/documents/CSRF.html”



- **Remediation:**

1. Implement anti-CSRF tokens in all state-changing requests (e.g., password updates, profile changes), and ensure tokens are unique per session and validated on the server side.
2. Enforce SameSite cookie attributes (e.g., SameSite=Strict or Lax) and require user re-authentication for sensitive actions to reduce the chance of unauthorised cross-origin requests.

## 4.6. Broken Access Control

During the security testing of Juice Shop, I created two user accounts to evaluate access control. While logged in as User 1, I was able to access User 2's basket data by manipulating the basketId in the API request. The application responded with a 200 OK, confirming unauthorised access to another user's private information. This reveals a Broken Access Control vulnerability, specifically an Insecure Direct Object Reference (IDOR). It highlights the absence of proper backend validation tied to resource ownership.

The screenshot shows a web browser window for the OWASP Juice Shop application. The URL in the address bar is 10.0.0.212:3000/#/basket. The page displays "Your Basket (User2@example.com)" with two items: "Apple Pomace" and "Banana Juice (1000ml)". Both items have quantity dropdowns set to 1. The total price is 0.89 and 1.99 respectively. On the right, a user menu for "User1@example.com" is open, showing options like Account, Orders & Payment, Privacy & Security, and Logout.

Below the main content, the browser's developer tools Network tab is visible, showing a list of network requests. The table includes columns for Status, Method, Domain, File, Initiator, Type, Transferred, Size, and Time. Key entries include:

Status	Method	Domain	File	Initiator	Type	Transferred	Size	Time
304	GET	10.0.0.212:3000	permfrost.jpg	img	jpeg	cached	93.64 kB	0 ms
304	GET	10.0.0.212:3000	lemon_juice.jpg	img	jpeg	cached	17.04 kB	4 ms
304	GET	10.0.0.212:3000	melon_bike.jpeg	img	jpeg	cached	21.52 kB	3 ms
304	GET	10.0.0.212:3000	fan_facemask.jpg	img	jpeg	cached	26.93 kB	2 ms
304	GET	10.0.0.212:3000	8	polyfills.js:1 (xhr)	json	cached	1.02 kB	0 ms
304	GET	10.0.0.212:3000	whoami	polyfills.js:1 (xhr)	json	cached	132 B	64 ms

At the bottom of the developer tools, it shows 47 requests, 1.36 MB / 20.78 kB transferred, Finish: 2.24 min, DOMContentLoaded: 137 ms, and load: 1.06 s.

Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms	≥ 40 ms	≥ 128 s	≥ 192 s
200	GET	10.0.0.212:3000	/socket.io/?EIO=4&transport=polling&t=PUQXuh9&sid=GETwKs_9n8Ed3cMAAAw	polyfills.js:1 (xhr)	plain	262 B	32 B		2ms		
101	GET	10.0.0.212:3000	/socket.io/?EIO=4&transport=websocket&sid=GETwKs_9n8Ed3cMAAAw	vendor.js:1 (websocket)	plain	129 B	0 B		3 ms		
200	GET	10.0.0.212:3000	Favicon.ico	FaviconLoader.sys.mjs-175 (img)	x-icon	cached	15.09 kB		0 ms		
0	GET	10.0.0.212:3000	apple_pressings.jpg	vendor.js:1 (img)	jpeg	NS_BINDING_ABORTED	29.16 kB		3 ms		
0	GET	10.0.0.212:3000	banana_juice.jpg	vendor.js:1 (img)	jpeg	NS_BINDING_ABORTED	19.83 kB		3 ms		
0	GET	10.0.0.212:3000	/socket.io/?EIO=4&transport=polling&t=PUQXumy&sid=GETwKs_9n8Ed3cMAAAw	polyfills.js:1 (xhr)	plain	230 B	1 B		105 ms		

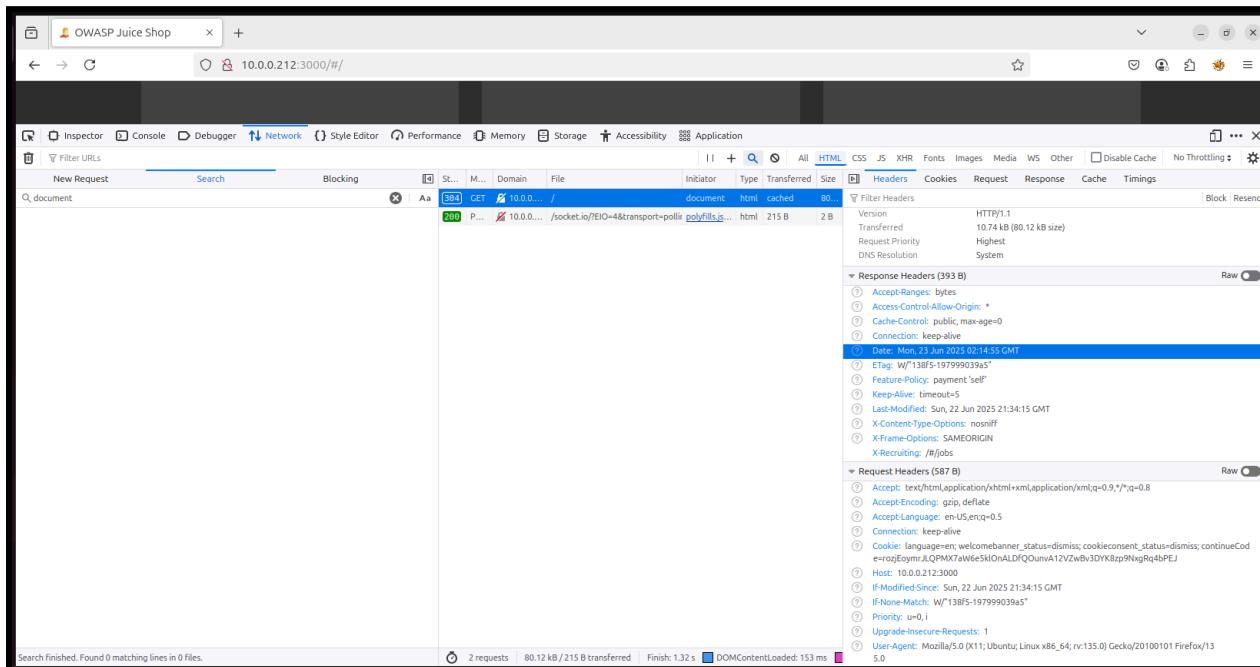
31 requests | 1.01 MB / 22.55 kB transferred | Finish: 1.71 s | DOMContentLoaded: 164 ms | load: 1.19 s

- **Remediation:**

1. Implement robust access control checks on the server side—every request should be verified to ensure the authenticated user is authorised to access or modify the resource (e.g., Basket ID).
2. Avoid relying on client-side identifiers alone—use session-bound tokens or role-based access enforcement to prevent unauthorised resource manipulation.

## 4.7. Security Misconfiguration

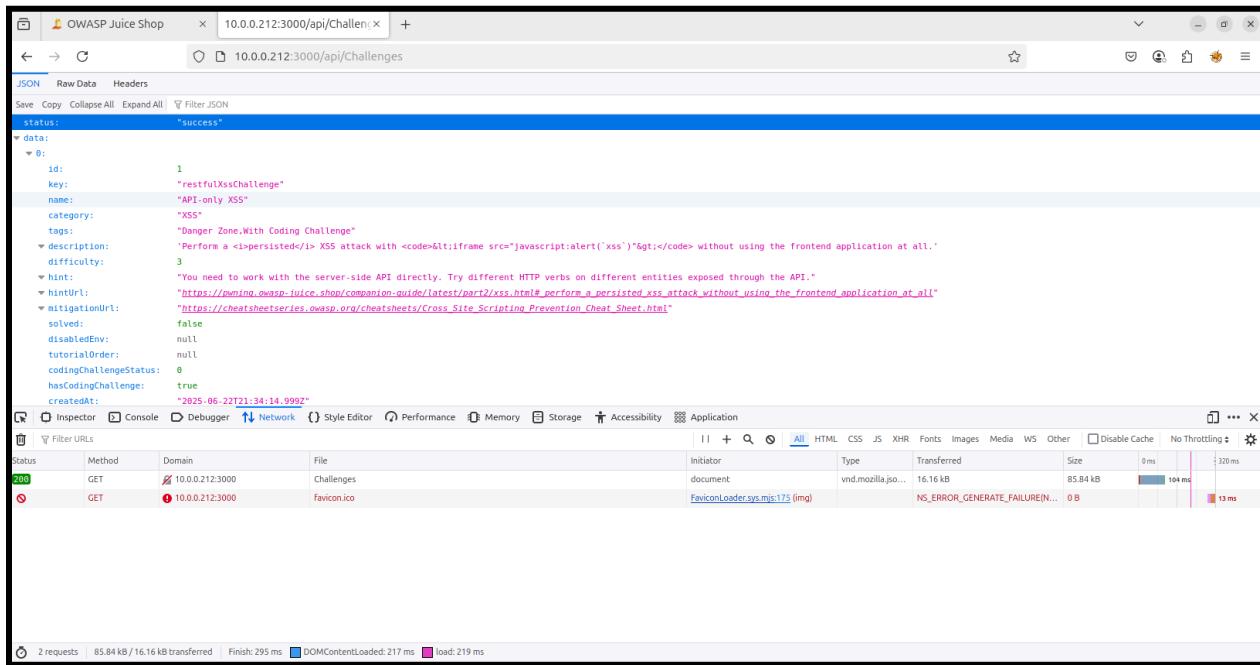
**HTTP Headers:** The application properly sets **X-Content-Type-Options**: nosniff and **X-Frame-Options**: SAMEORIGIN, offering basic protection. However, essential headers like Content-Security-Policy, Strict-Transport-Security, Referrer-Policy, and Permissions-Policy are missing. These headers defend against XSS, clickjacking, insecure transport, and data leakage. Their absence indicates a server-side security misconfiguration. Strengthening the header configuration is necessary to improve client-side security.



- **Remediation:** Configure the web server to include essential HTTP security headers such as **Content-Security-Policy**, **Strict-Transport-Security**, **X-Content-Type-Options**, and **Referrer-Policy**. These headers help mitigate XSS, clickjacking, and man-in-the-middle attacks

## 4.8. Unprotected API

The endpoint /api/Challenges is accessible without authentication, allowing public exposure of internal challenge data. This represents an **unprotected API** and a **security misconfiguration**. Sensitive metadata is returned with a 200 OK, violating access control principles.



The screenshot shows the OWASP Juice Shop application interface. In the browser's address bar, the URL is 10.0.0.212:3000/api/Challenges. The page content displays a JSON response for a challenge:

```
status: "success"
data:
  0:
    id: 1
    key: "restfulXssChallenge"
    name: "API-only XSS"
    category: "XSS"
    tags: "Danger Zone,With Coding Challenge"
    description: "Perform a <i>persistent</i> XSS attack with <code><iframe src='javascript:alert('xss')'></code> without using the frontend application at all."
    difficulty: 3
    hint: "You need to work with the server-side API directly. Try different HTTP verbs on different entities exposed through the API."
    hintUrl: "https://portswigger.net/www-enum/vulnerabilities/persistent-xss"
    mitigationUrl: "https://cheatsheetsseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html"
    solved: false
    disabledEnv: null
    tutorialOrder: null
    codingChallengeStatus: 0
    hasCodingChallenge: true
    createdat: "2025-06-22T21:34:14.999Z"
```

Below the JSON response, the developer tools Network tab shows two requests:

Status	Method	Domain	File	Initiator	Type	Transferred	Size	Time
200	GET	10.0.0.212:3000	Challenges	document	script	85.84 kB	0 ms	320 ms
0	GET	10.0.0.212:3000	favicon.ico	FaviconLoader.sys.mjs [175 (img)]	image	NS_ERROR_GENERATE_FAILURE(0...)	0 B	13 ms

- **Remediation:** Implement authentication and authorisation checks on all API endpoints. Ensure that sensitive endpoints (e.g., /api/Challenges) are not publicly accessible without valid user credentials, and follow the principle of least privilege for API access.

## 5. Appendix

S.no	Tool	Purpose
1	Whois	Gather domain and IP ownership details
2	nslookup	Perform DNS and reverse DNS lookup
3	Wappalyzer	Identifying front-end and back-end technologies
4	Whatweb	Analysing HTTP responses and detecting tech stack
5	Nmap	Discovering open ports and services on the target system
6	Nikto	Enumerating web server misconfigurations and exposures
7	Burpsuite	Intercept test and manipulate web traffic and payloads

## **6. Key Observations**

---

1. The application revealed multiple OWASP Top 10 vulnerabilities, including SQL Injection, Cross-Site Scripting (XSS), IDOR, CSRF, and Security Misconfigurations.
2. Lack of security headers and exposed API endpoints indicated potential for client-side exploitation and unauthorised access.
3. Technologies like jQuery 2.2.4 and unprotected directories suggested outdated components and misconfigured web servers.
4. Authentication and session management were implemented securely, and no token reuse or session fixation was observed.
5. Enumeration using Nmap, Nikto, and WhatWeb aided in mapping out the infrastructure and understanding the attack surface.

## **7. Conclusion**

---

The assessment confirmed that while the application is functionally sound, it contains critical security flaws that could be exploited by malicious users. Remediation of identified issues—especially those involving input validation, access control, and secure server configurations—is essential. Implementing secure development practices and regular vulnerability assessments will greatly reduce the attack surface and improve overall resilience.