AvdB Security & Protection

Customer: BrightWave Solutions B.V.

Website: https://app.brightwavesolutions.nl

Test-type: Light Web Application Penetration Test

Date: 18-03-2025

Report version: 1.0

Performed by: Alwin van der Bildt → **AvdB Security & Protection**

Confidentiality Statement

This report contains confidential information related to the security of the tested application.

The document is intended exclusively for the client and may not be shared with third parties without the prior written permission of the author.

Table of contents

Executive Summary

On behalf of the client, a light web application penetration test was performed on the application mentioned above.

The goal of this test was to identify common security weaknesses within the web application, with a focus on vulnerabilities that arise due to improper implementation of authentication, authorization, and input processing.

The test was performed with limited depth and low impact, to ensure the continuity and stability of the application. Aggressive attack techniques and exploit chaining have been deliberately refrained from.

Important note:

This test is not a substitute for a full penetration test conducted by a senior security team.

Advanced attack scenarios, complex exploit chains, and in-depth infrastructure testing are outside the scope of this assessment

In scope

The following components are in scope for this test:

- Web application accessible via HTTP / HTTPS
- Functionality available to end users
- API endpoints linked to the application (if publicly accessible)
- Login and registration forms

Out of scope

The following components are explicitly excluded from this test:

- Denial-of-Services (DoS) and stress tests
- Brute-force attacks
- Exploit chaining and privilege escalation
- Infrastructural penetration testing
- Social engineering
- Complete source code review
- Post-exploitation activities

Methodology

This test was performed according to a structured approach consisting of the phases below.

Reconnaissance

- Passive and active information collection
- Inventory of subdomains and endpoints
- Browser-based analysis of application behaviour

Manual security testing

- Authentication and session management
- Authorization and access control
- Insecure Direct Object References (IDOR)
- Cross-Site Scripting (limited)
- SQL Injection (detection-oriented)
- File path handling and directory traversal
- Weak or default credentials (limited)

No automated exploit frameworks or high-impact scanning were used.

Limitations of the test

This Light Web Application Penetration Test was performed within the pre-agreed time and scope constraints.

As a result:

- Not all vulnerabilities can be ruled out
- Only tested functionality has been reviewed
- Changes after the test date are not included in this assessment

This test does not guarantee that the application is free of security risks.

Summary and Findings

ID	FINDING	COLOR OF SEVERITY	RISK
f-01	Insufficient Authorization Controls (IDOR)	🔴	High
f-02	Reflected Cross-Site Scripting (XSS)	🟠	Medium
f-03	Missing Security Headers	🟠	Medium
f-04	Session Token Not Invalidated After Logout	🟡	Low
f-05	Verbose Error Messages Expose Internal Paths	🟡	Low

Findings in detail:

F-01 — Insufficient Authorization Controls (IDOR)

Risk Level: High
Category: Access Control (OWASP A01:2021)

Description
During the test, it was discovered that the API endpoint /api/v1/users/{id}/profile does not verify whether the authenticated user is authorized to access the requested resource. By modifying the user ID parameter in the URL, it was possible to retrieve personal data belonging to other users, including email addresses, phone numbers, and account settings.

Impact
An attacker can gain unauthorized access to sensitive personal data of other users. This may lead to privacy violations, identity theft, and potential regulatory consequences under the GDPR.

Evidence

- GET /api/v1/users/1042/profile returned data for user ID 1042 while authenticated as user ID 1097
- Response contained full name, email, phone number, and billing address of the target user

Recommended Mitigation
Implement server-side authorization checks that verify the authenticated user's identity matches the requested resource. Use access control middleware that enforces ownership validation on all user-specific endpoints.

F-02 — Reflected Cross-Site Scripting (XSS)

Risk Level: Medium
Category: Injection (OWASP A03:2021)

Description
The search functionality at /search?q= does not properly sanitize user input before reflecting it in the HTML response. A crafted URL containing JavaScript code is executed in the browser of any user who clicks the link.

Impact
An attacker can craft a malicious link that, when clicked by a victim, executes arbitrary JavaScript in their browser session. This could be used to steal session cookies, redirect users to phishing pages, or perform actions on behalf of the victim.

Evidence

- Input <script>alert(document.cookie)</script> in the search parameter was reflected and executed
- The response Content-Type was text/html with no output encoding applied

Recommended Mitigation
Implement proper output encoding (HTML entity encoding) for all user-supplied data reflected in HTML responses. Additionally, deploy a Content Security Policy (CSP) header to mitigate the impact of any remaining XSS vulnerabilities.

F-03 — Missing Security Headers

Risk Level: Medium
Category: Security Misconfiguration (OWASP A05:2021)

Description
The application does not include several recommended HTTP security headers in its responses. Missing headers include Content-Security-Policy, X-Content-Type-Options, Strict-Transport-Security, and Referrer-Policy.

Impact
Without these headers, the application is more susceptible to various client-side attacks including clickjacking, MIME-type sniffing attacks, and protocol downgrade attacks. The absence of HSTS also means users connecting over HTTP are not automatically redirected to HTTPS.

Evidence

- HTTP response headers inspected on all tested endpoints showed no security-related headers present
- The application was accessible over plain HTTP without redirection to HTTPS

Recommended Mitigation
Configure the web server to include security headers on all responses: Content-Security-Policy with a restrictive policy, X-Content-Type-Options: nosniff, Strict-Transport-Security with a minimum max-age of 31536000, Referrer-Policy: strict-origin-when-cross-origin, and X-Frame-Options: DENY.

F-04 — Session Token Not Invalidated After Logout

Risk Level: Low
Category: Authentication (OWASP A07:2021)

Description
After a user logs out of the application, the session token (JWT) remains valid and can be

reused to access protected endpoints. The logout action only removes the token from the client-side storage but does not invalidate it on the server.

Impact
If an attacker obtains a session token (e.g., through XSS, network sniffing, or access to a shared device), they can continue using it even after the legitimate user has logged out, until the token naturally expires.

Evidence

- After logout, replaying the Authorization header with the previous JWT token still returned HTTP 200 on /api/v1/users/me
- Token remained valid for the full expiration period (24 hours) after logout

Recommended Mitigation
Implement a server-side token blacklist or revocation mechanism. Upon logout, add the token to a revocation list and validate against this list on each request. Consider reducing the JWT expiration time to limit the window of exposure.

F-05 — Verbose Error Messages Expose Internal Paths

Risk Level: Low
Category: Security Misconfiguration (OWASP A05:2021)

Description
When the application encounters an unhandled exception, it returns detailed error messages including internal file paths, stack traces, and framework version information. This was observed on various endpoints when supplying malformed input.

Impact
Internal path disclosure and stack traces provide an attacker with valuable information about the application's internal structure, technology stack, and potential attack surfaces. This information can be used to craft more targeted attacks.

Evidence

- Sending a malformed JSON body to /api/v1/orders returned a full Python traceback including /opt/app/brightwaveapi/views/orders.py
- The error response also revealed Django version 4.2.8 and Python 3.11.6

Recommended Mitigation
Implement a global error handler that catches all unhandled exceptions and returns a generic error message to the client. Log detailed error information server-side only. Ensure debug mode is disabled in the production environment.

General Recommendations

- Resolve high-risk findings with priority
- Implement structural server-side validation
- Consider periodic re-tests after changes
- In case of growth or increased risk, have a more in-depth penetration test performed

Limitation of Liablity & Indemnification

This security assessment has been performed to the best of our knowledge and ability, within the pre-agreed scope, methodology and limitations as described in this report.

The test performed is a light web application penetration test and does not guarantee that the tested application, including the specifically assessed components, is or will remain free of security vulnerabilities or security incidents.

The client acknowledged and accepted beforehand that security incidents may occur before, during, or after the performance of this test;

Vulnerabilities, misconfigurations or attack scenarios may exist despite the performance of this test, even within components that have been explicitly tested.

Security is an ongoing process and not a one-time assessment.

The author of this report shall not be held liable for any damage, loss or consequential loss arising out of, or in connection with:

- Security incidents, compromise or hacking of (parts of) the application, regardless of whether these parts were part of the test.
- Abuse or attacks by third parties, regardless of timing, cause or technique used.
- Vulnerabilities that were not identified during the test, regardless of reason.
- Changes to the application, configuration, or infrastructure before or after the test period.
- Decisions or actions taken based on the results of this assessment.

By conducting this assessment, the client fully indemnifies the author against any liability, claims, damages or legal proceedings arising from security incidents in relation to the tested application.

Conclusion

This report provides a snapshot of the security status of the tested application within the agreed scope.

A total of five findings were identified: one high-risk, two medium-risk, and two low-risk vulnerabilities. It is recommended to address the high-risk finding (IDOR) with priority.

For additional assurance, it is recommended to periodically evaluate security and have more in-depth testing performed where necessary.

Thank you for choosing AvdB Security & Protection

Alwin van der Bildt

Junior Web Application Tester
Chamber of Commerce number: 99972654