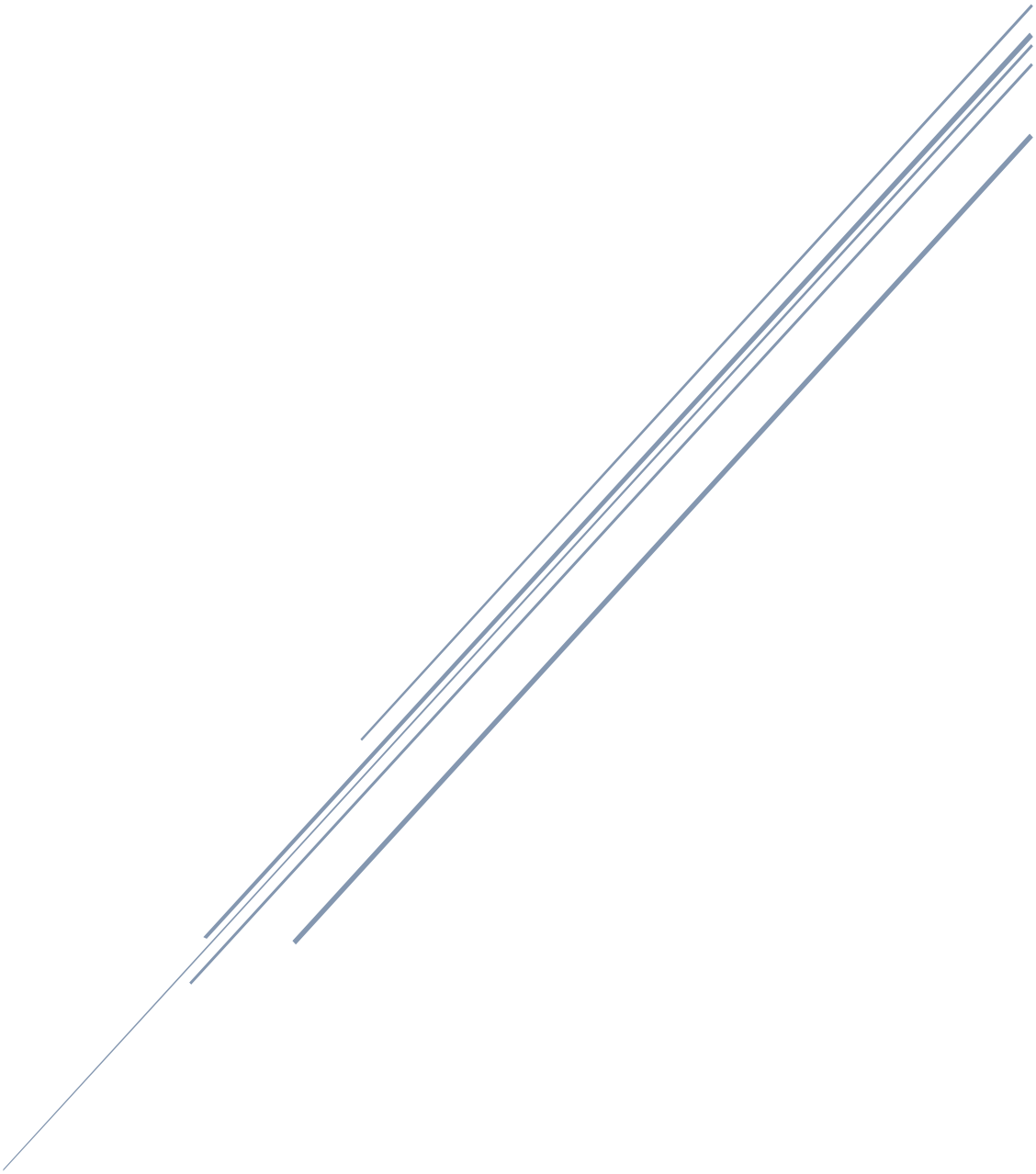
OWASP SECURITY REPORT

Project: Public transport lookup application



FONTYS UNIVERSITY OF APPLIED SCIENCES
Mertan Rasim, S3-CB03, 2022-12-13

Contents

Introduction

Probable vulnerabilities

Vulnerabilities	Likelihood	Impact	Risk	Actions possible	Planned
A01:2021-Broken Access Control	Very unlikely	Moderate	Low	Reevaluate access control	Yes
A02:2021-Cryptographic Failures	Very unlikely	Severe	Low	Use OAuth 2.0 to not store passwords	Yes
A03:2021-Injection	Highly likely	Severe	High	Implement user input sanitization. Use prepared statements.	No
A04:2021-Insecure Design	Unlikely	Moderate	Low	Reevaluate the level of security required	Yes
A05:2021-Security Misconfiguration	Likely	Severe	High	Research & review security configuration	No
A06:2021-Vulnerable and Outdated Components	Very unlikely	Moderate	Low	Update/resolve dependencies	Yes
A07:2021-Identification and Authentication Failures	Likely	Severe	High	Integrate MFA; complex passwords; defense mechanism against brute force;	No
A08:2021-Software and Data Integrity Failures	Likely	Moderate	Moderate	Reevaluate source validation mechanisms	No
A09:2021-Security Logging and Monitoring Failures	Highly likely	Moderate	High	Implement more frequent logging	No
A10:2021-Server-Side Request Forgery	Likely	Severe	High	More information hiding. Sanitize and validate all client-supplied input data.	No

Reasoning

- A01 – Broken access control is highly unlikely to happen, due to the token-based authentication & authorization. Nevertheless, the impact would be a full control over the city traffic data and most importantly, registered users' emails. Passwords are secured under modern hashing algorithms.
- A02 – Cryptographic failures depend on the provided implementation from imported libraries, not from the project's own code. The project specifically uses Argon2 password hashing algorithm which is battle-tested. [1]
- A03 – The scope of the project did not cover SQL injection, which would be that it is highly likely exploitable. Even though the project uses Hibernate ORM, it does not defend against such attacks. Using prepared statements would be the best option to prevent such attacks.
- A04 – Insecure design would be unlikely, due to no severe vulnerabilities being found. In the case that the statement is not true, reevaluating design decisions would be the correct way to handle the issue.
- A05 – Security misconfiguration would be quite likely, due to the scope being oriented around functional requirements delivery and learning goals, and not security. Nevertheless, basic concepts have been applied to the project such that it does not deviate from the practical perspective. More research and vulnerability analysis would mitigate this issue.
- A06 – Vulnerable and outdated components are very unlikely to be present, due to the task being inside the scope (See issue #3). A solution to the problem would be to update/resolve the project's dependencies.
- A07 – The main authentication weakness of the project is the lack of defense mechanisms. Contemporary strategies to mitigate such attacks include multi-factor authentication, strict password complexity requirement, option to reset password, logging, etc.
- A08 - Software and data integrity failures are likely to happen, due to not being revised. However, data integrity primarily depends on Gradle/NPM servers to provide non-malicious modules. Validating can be done in the form of cryptographic keys to prove the origin of dependencies.

- A09 – Security and logging failures are highly likely to happen, being out of the scope. Security breaches would be harder to track and prevent without logging, therefore requiring it.
- A10 – Server-side request forgery is highly likely because of no user data sanitization, no advanced information hiding, and no security-wise decoupling of the project.

Conclusion

Overall, 4 low risk, 1 moderate, and 5 severe security issues have been reviewed. Considering the assignment's scope not focusing primarily on security, I conclude that the results are as expected from the planning.

References

[1] J. Aumasson, "Password hashing competition," 2022. [Online]. Available: https://www.password-hashing.net/. [Accessed 13 12 2022].