

OWASP report

John Ooi



**Table of contents**

[Analysis 2](#_Toc137162658)

[Reasoning 3](#_Toc137162659)

[Conclusion 3](#_Toc137162660)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Likelihood | Impact | Risk | Actions possible | Planned |
| A1: Broken Access Control | Low | Severe | Low | Implement proper access control measures | Yes |
| A2: Cryptographic failure | Unlikely | High | Low | Use strong encryption algorithms and key management | Yes |
| A3: Injection | Moderate | Severe | Medium | Apply input validation and parameterized queries | Parameterized queries are used, input validation not present for some. |
| A4: Insecure design | Low | Medium | Low | Using HTTPS | No, risk accepted |
| A5: Security misconfiguration | Medium | Medium | Medium | Configure security settings correctly | Yes |
| A6: Vulnerable and outdated components | Medium | Medium | Medium | Regularly update and patch components | No, risk accepted |
| A7: Identification and authentication failures | Unlikely | High | Medium | Implement proper authentication and session management | Yes |
| A8: Software and data integrity failures | Low | High | Medium | Verify software integrity and ensure secure data storage | Somewhat. |
| A9: Security logging and monitoring failures | Medium | Medium | Medium | Implement comprehensive logging and monitoring | No, risk accepted |
| A10: Server-side request forgery | High | Moderate | Moderate | Validate and sanitize user-supplied URLs | No, risk accepted |

# Analysis

# Reasoning

A1: Broken Access Control: This security risk has a low likelihood of occurrence, but if it happens, the impact can be severe. Implementing proper access control measures is crucial to mitigate the risk of unauthorized access and protect sensitive resources.

A2: Cryptographic failure: While the likelihood of this risk is unlikely, the impact can be high. Using strong encryption is essential to ensure the confidentiality and integrity of sensitive data.

A3: Injection: The likelihood of injection attacks is moderate, but the impact can be severe. Applying input validation and parameterized queries helps prevent these attacks, but some input validation is missing in the current setup.

A4: Insecure design: With a low likelihood and medium impact, the risk associated with insecure design can be mitigated by using HTTPS to secure communication. However, the decision not to implement secure design principles indicates acceptance of the risk.

A5: Security misconfiguration: This risk has a medium likelihood and impact. Configuring security settings correctly is necessary to avoid potential vulnerabilities arising from misconfigurations.

A6: Vulnerable and outdated components: The likelihood of vulnerabilities due to outdated components is medium, with a medium impact. Regularly updating and patching components is vital to minimize the risk, but currently, the decision is made to accept the risk.

A7: Identification and authentication failures: While the likelihood is unlikely, the impact can be high. Proper implementation of authentication and session management mechanisms is crucial to mitigate this risk.

A8: Software and data integrity failures: This risk has a low likelihood but a high impact. Ensuring software integrity and secure data storage are important measures to prevent unauthorized modifications and protect data integrity.

A9: Security logging and monitoring failures: The likelihood of such failures is medium, with a medium impact. Implementing comprehensive logging and monitoring systems is necessary to detect and respond to security incidents effectively. However, the decision is made to accept the risk in the current setup.

A10: Server-side request forgery: With a high likelihood and moderate impact, validating and sanitizing user-supplied URLs is essential to prevent potential exploits targeting server-side request forgery vulnerabilities. However, the decision is made to accept the risk in the current setup.

# Conclusion

In conclusion, the application has implemented several security measures that address potential risks. Proper access control measures, strong hashing algorithms, input validation, and parameterized queries have been implemented for relevant risks. Security settings have been configured correctly, and authentication and session management mechanisms are in place. However, there are areas that require attention for future security improvements. Insecure design could be further addressed by considering additional security measures beyond HTTPS. Regular updates and patching of components, as well as verifying software integrity and secure data storage, should be prioritized. Enhancements in security logging and monitoring would provide better incident detection and response capabilities. While the application has a solid foundation, continuous evaluation and improvement are recommended to maintain a high level of security.