Fontys University of Applied Sciences

# Security Risk Assessment

GetawayGo

Anna Kadurina
11/11/2024

## Table of Contents



# Introduction

The purpose of this document is to identify and address potential security risks across each phase of the GetawayGo application development cycle. By analyzing risks during Analysis, Design, Implementation, Testing, Deployment, and Monitoring phases, this document aims to ensure that all aspects of the project are designed and built to be secure.

# Analysis Phase

In this phase, I am going to identify potential security requirements that are going to guide design and implementation.

### Risks

- Incomplete or vague security requirements – Failing to define comprehensive security requirements. To mitigate this, I have created 2 guideline documents to check all the security requirements that are relevant for GetawayGo – one for OWASP Top 10 and one for GDPR.

## Design Phase

In this phase, I am going to ensure that the system architecture supports robust security.

### Risks

- Inadequate access control design – Poorly defined access control could lead to unauthorized access, exposing sensitive data. To mitigate this, I have implemented role-based access control (RBAC) for restricting access based on user roles.
- Unprotected data flows – Data transmitted without encryption may be intercepted, leading to information leaks. To avoid this, I am using HTTPS/TLS for all data transfers.

## Implementation Phase

I am going to code the application with secure practices.

### Risks

- Insecure Coding Practices – Poor coding practices may introduce vulnerabilities like SQL injection. To mitigate this, I am using secure coding practices, SonarQube to detect common vulnerabilities, and Snyk to detect vulnerable packages.
- Improper input validation – Failure to validate user input can lead to injection attacks. To mitigate this, I will implement robust input validation across all user inputs.
- Unencrypted storage of sensitive data – Storing sensitive data in plain text increases rhe risk for the data to be compromise. To avoid this risk, I am encrypting sensitive data such as user passwords.

# Testing Phase

In this phase, I am going to validate that the system behaves securely under various conditions.

## Risks

- Lack of security-focused testing – Without specific tests for vulnerabilities, the application might miss critical flaws. To ensure I am mitigating this, I have implemented OWASP ZAP scan and Snyk in my pipelines.
- Failed tests – Failed tests may indicate there is something wrong with the application. The tests can be the unit and integration tests, Snyk vulnerability testing, SonarQube or the OWASP ZAP scan. Whichever of these tests fail, requires manual check on what went wrong. To avoid further issues, the pipelines are programmed to stop in case of a failure. Furthermore, the deployment needs approval.

# Deployment Phase

I am going to deploy the application with secure configurations and settings.

## Risks

- Exposure of secrets in configuration files – Secrets that are hardcoded are not secure and can lead to serious risks. To mitigate this, I am using variable groups that contain the connection strings, URLs, etc. and this information is filled in during the pipleines.
- Misconfigured security settings – Configuration errors could expose the system to attacks such as unauthorized access. To mitigate this, I am ensuring the all environments are configured correctly before deployment,.
- Exposure of sensitive information in logs or error messages – Exposing stack traces or sensitive information in logs can reveal vulnerabilities to attackers. To mitigate this, I am sanitizing the logs to ensure no sensitive information is logged. Furthermore, logging access is limited to authorized personnel only (me).

# Monitoring Phase

The application should be continuously monitored after deployment.

### Risks

- Unmonitored security incidents – A lack of monitoring can result in undetected breaches and slow response times. To mitigate this, I have setup a monitoring tool called Sentry.io that is alerting me on any unusual behaviour.
- Outdated dependencies – Outdated software libraries or frameworks may contain vulnerabilities. To solve this issue, I have included Snyk scan in my pipelines that checks the third-party libraries prior to deployment. If any of them are a risk to the application, the Build should fail.

## Conclusion

This security risk assessment document provides a comprehensive plan for identifying and mitigating security risks throughout the lifecycle of GetawayGo. By following these measures, I aim to protect sensitive data, maintain system integrity, and uphold compliance with security best practices.