Chapter 1

Introduction

1.1 Background

Throughout the last ten years, mass cyberattacks against major organizations have amplified. Security breaches are the most prominent cause for attacks being allowed to happen. Although different types of organizations become victim of cyberattacks, an analysis of data breaches experienced in multiple organizations established that medical organizations and BSOs are the least prepared against attacks [1]. Furthermore, the latest reports confirm of almost half of businesses have been confronted with cyberattacks since companies are settling into the new normal. Cyberattacks have grown in frequency and severity since the pandemic. Lallie, Harjinder Singh et al observed that there appears to be a loose correlation between the announcement and a corresponding cyber-attack campaign that utilizes the event as a hook to increase the likelihood of success [2]. Thus, to minimize or eliminate the damage caused by cyberattacks, software security has to be addressed during software development.

Security breaches are caused by security vulnerabilities in source code introduced by software developers when creating software, and therefore developers are often blamed for vulnerabilities [3]. However, application security is primarily performed by security experts causing a separation between security and development. As a result, the probability of insecure software is increased [4].

Writing secure code is critical with the prevalence of security vulnerabilities. To achieve this, developers need to be aware of the potential vulnerabilities they might introduce when developing software features and understand how to mitigate them. Still, the knowledge and skills to produce secure software are lacking and often are not taught in computing curricula despite the existence of secure coding guidelines [5] [6] [7].

Such secure coding guidelines are provided by the Open Web Application Security Project (OWASP). For instance, the OWASP Top Ten brings awareness about the most common security vulnerabilities found in software. Furthermore, security standards such as the OWASP Application Security Verification Standard (ASVS) and the Mobile Application Security Verification Standard (MASVS) enable organizations and developers to produce and maintain secure software.

To address security during software development, this study will implement a secure agile Software Development Life Cycle (SDLC) with the OWASP Security Knowledge Framework (SKF) to develop CheFeed, a mobile application developed for this thesis. SKF is an expert system application that uses secure coding guidelines such as the OWASP ASVS and OWASP MASVS to assist developers in pre-development and post-development phases to create secure-by-design software.

1.2 Aims and Benefits

1.2.1 Aim

Security vulnerabilities found in applications introduced by software developers are the cause of many security breaches and data theft. To reduce the introduction of security vulnerabilities security must be addressed during software development. This study aims to implement a secure agile SDLC with testable security controls configured through SKF.

1.2.2 Benefits

This study will provide insights into the process of developing software in an agile SDLC where security is addressed from the beginning. It provides a clear presentation of the benefits of using SKF to configure feature sprints where security should be in place. Students and software developers may benefit from understanding how secure code guidelines can be used. Furthermore, the implementation of SKF for a secure SDLC could serve as a tool for future studies to build upon the process presented in this study.

1.3 Scope

The scope of this study is limited to ~~configuring~~ MASVS Level 1 (MASVS-L1) security controls with SKF for the development of CheFeed. MASVS-L1 ensures that mobile applications cohere to the best security practices concerning code quality, handling of sensitive data, and interaction with the mobile environment.

This project is a group effort where two fellow students will contribute to the development of CheFeed. Briefly their study contributes to the following:

- The development of the API and database design by Stephanus Jovan Novarian in his thesis "CheFeed: Development of CRUD backend services on recipes".
- The development of a sentiment analysis model which uses recipe reviews as its input by Ikshan Maulana in his thesis "CheFeed - The Implementation of different RNN Architectures".

Bibliography

[1] Hicham Hammouchi et al. “Digging Deeper into Data Breaches: An Exploratory Data Analysis of Hacking Breaches Over Time”. In: *Procedia Computer Science* 151 (2019), pp. 1004–1009. DOI: 10.1016/j.procs.2019.04.141. URL: https://doi.org/10.1016%2Fj.procs.2019.04.141.

[2] Harjinder Singh Lallie et al. “Cyber security in the age of COVID-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic”. In: *Computers amp Security* 105 (June 2021), p. 102248. DOI: 10.1016/j.cose.2021.102248. URL: https://doi.org/10.1016%2Fj.cose.2021.102248.

[3] Hala Assal and Sonia Chiasson. “*less*iThink secure from the beginning*less*/i”. In: *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, May 2019. DOI: 10.1145/3290605.3300519. URL: https://doi.org/10.1145%2F3290605.3300519.

[4] Tyler W. Thomas et al. “Security During Application Development”. In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, Apr. 2018. DOI: 10.1145/3173574.3173836. URL: https://doi.org/10.1145%2F3173574.3173836.

[5] Madiha Tabassum et al. “Evaluating Two Methods for Integrating Secure Programming Education”. In: *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*. ACM, Feb. 2018. DOI: 10.1145/3159450.3159511. URL: https://doi.org/10.1145%2F3159450.3159511.

[6] Huiming Yu et al. “Teaching secure software engineering: Writing secure code”. In: *2011 7th Central and Eastern European Software Engineering Conference (CEE-SECR)*. IEEE. 2011, pp. 1–5.

[7] Tiago Espinha Gasiba et al. “Is Secure Coding Education in the Industry Needed? An Investigation Through a Large Scale Survey”. In: *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*. IEEE, May 2021. DOI: 10.1109/icse-seet52601.2021.00034. URL: https://doi.org/10.1109%2Ficse-seet52601.2021.00034.