Contents

Chapter 1

Introduction

1.1 Background

Are software developers responsible for the cyberattacks that has increased over the last decade? An analysis of over 9000 data breaches since 2005 have revealed that data breaches have contributed to the loss of 11,5 billion individual records with major financial and technical impact [1]. Countless attacks are allowed to happen as a consequence of software vulnerabilities that leave web applications, web servers, or websites exposed. Software developers increase the risk of a security vulnerability each time a new feature is added to an application, albeit many can be mitigated with a secure-by-design approach. However, software developers need to have adequate knowledge on how to mitigate common security vulnerabilities. Ergo the knowledge and skill to develop secure software needs to be taught to software developers, yet is commonly overlooked [2].

Several approaches to educate students on secure programming are commonly considered. The first, adding it as a material to existing classes. However, if the principles and practices are introduced as extra material in existing classes, the skill is atrophied, considering that whether the students write secure programs is not being examined. The second, adding a separate class that covers secure programming in an already packed curriculum. A third approach is the introduction of a "secure programming clinic" that facilitates students on the principles and practices of writing secure software [3]. The University of North Carolina at Charlotta introduced their own tool Educational Security in the IDE, which provides students with security warnings on assignment code for integrating secure programming education. The paper states that students awareness and knowledge on secure programming increases through the usage of ESIDE [2].

To learn secure programming practices, this thesis proposes to integrate the Security Knowledge Framework into the Software Development Life Cycle with the development of a social recipe sharing app, CheFEED. SKF is an open-source flagship project from the Open Web Application Security Project. It utilizes the OWASP Application Security Verification Standard to train developers in writing secure code by design. The ASVS provides a framework of security requirements and controls for designing, developing and testing modern web applications and web services. In addition they provide the Mobile Application Security Verification Standard that focuses security requirements and controls for mobile applications. SKF in addition has developed a security expert system that generates

these security requirements for a sprint during the SDLC.

The purpose of this thesis is to examine whether SKF is a good source for developers to learn about secure software development, without a security champion in the team. The thesis is conducted as a case study to see if skill and abilities in secure programming is enhanced through the development of the CheFEED application. The features of CheFEED will form the foundation to generate the security requirements needed to develop a secure application.

1.2 Scope

The thesis focuses on fundamental security concepts and principles. They will be applied by developing the CHEFEED application. CHEFEED will have an API and a hybrid native client. The application will go through a secure Software Development Life Cycle (SDLC). An important principle in information security is to find a good balance between security and productivity. Thus, a security maturity model and a standard for writing secure code will guide the SDLC. Adding security into the SDLC will allow to have secure code by design, instead of having security as an afterthought. Furthermore, security tests will be performed in the form of penetration tests and security code reviews.

CHEFEED is a group effort and the scope and responsibilities outside this thesis are summarized in Table 1.1.

Table 1.1: Member scope overview

Member	**Scope**	**Thesis**
Ikhsan Maulana	Sentiment Analysis, back-end development	Sentiment Analysis on Food Reviews
Stephanus Jovan Novarian	SDLC, back-end development	Implement Agile Software Development Life Cycle on CHEFEED

1.3 Aims and Benefits

1.3.1 Aims

Software code is the essence of each application. Applications that we use daily are processing important and sensitive assets. Therefore, the confidentiality, integrity, and availability of those assets are at risk of being compromised. However, adding security might be a challenging task. The aim of this thesis is to demonstrate how security can be added to the SDLC by constructing sensible security requirements for CHEFEED. The security requirements will be based on the OWASP ASVS and MASVS by utilizing OWASP SKF.

1.3.2 Benefits

Secure programming forces developers to improve coding standards as well as the overall coding architecture. Adding security to the SDL will save development

time, since there is no need to add new code after security audits are performed.

1.4 Structures

This section summarizes the general description of the coverage of each chapter.

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Chapter 2

Theoretical Foundation

2.1 Theoretical Foundation

2.1.1 Defining Information Security

It is important to understand what exactly is meant when discussing security. Therefore, we have to properly define it. For instance, the terms cybersecurity and information security are commonly used indistinguishably. The Cambridge Dictionary defines information security as "methods used to prevent electronic information from being illegally obtained or used", and defines cybersecurity as "things that are done to protect a person, organization, or country and their computer information against crime or attacks carried out using the internet". The definition for cybersecurity describes a much larger scope for security.

Solms et al [4] supports this argument and reasons that information security is solely about securing the information, generally referred to as the asset, from potential threats posed by inherent vulnerabilities. Furthermore, they outlined that cybersecurity goes beyond protecting assets. Cybersecurity includes the protection of those that function in cyberspace in addition to any of their assets that can be attained through cyberspace. Although the definition of information security and cybersecurity overlap each other, the latter is much more extensive in its definition. Overall, security is about securing assets against the most likely forms of attacks, to the best ability [5].

2.1.2 Challenges in Information Security

Computers have evolved drastically from the mechanical calculating machines they once were when first introduced in the 19th century. Today, computers are powerful machines that allows us to surf the internet, run games, and stream multimedia. Everything involves system and software technology that constantly is processing information. Important and sensitive information are controlled, managed or either form part of the cyberspace. Likewise, important and sensitive information can be exploited as well in this space [6]. Moreover, it has become very simple to stage an attack. A sophisticated attack can be constructed that affect millions of computers worldwide by an attacker with tutorials found online together with their own knowledge and resources.

In addition, many organizations perceive security as a hindrance to productiv-

ity. It is not uncommon for discussions concerning security to be avoided among business leaders and IT personnel. Errors made by developers might be costly and might endanger everyone who trust the software they build. Though the stakes in security are high, developers perceive security as a secondary concern [7].

Even so, security can be complicated without the appropriate approach. As the level of security is increased, the level of productivity is usually decreased. Therefore, the role of a security plan is to find the balance between protection, usability, and cost. Moreover, in what manner the level of security relates to the value of the item being secured needs to be taken into consideration, when an asset, system, or environment is secured.

Knowledge on how to stay secure changes at a much slower pace in contrast to the increasingly accelerated rate technology changes. As a consequence, security does not always keep up with the changes. At the same time, gaining a good understanding of the fundamentals of information security will provide the foundation needed to manage changes as they come along.

2.1.3 Secure By Design

The term secure can be defined as "freedom from risk and the threat of change for the worse". From the software engineers perspective security is about engineering software such that assets are free from risk and the threat of change for the worse, or at least to the best ability. Secure programming is about designing and implementing software with the minimal amount of vulnerabilities that an attacker can exploit [8]. However, modern software is complex and fragile. Despite professional engineers being capable of testing and debugging code, security is a different issue, because insecure code generally works without issues, given no attacker is exploiting the code.

2.1.4 Software Security Requirement

2.2 Theoretical Frameworks

2.2.1 Fundamentals of Security

The Confidentiality, Integrity, and Availability Triad

Security can be complicated as discussed in Section 2.1.2. Nonetheless, the confidentiality, integrity, and availability (CIA) triad, as shown in figure 2.1, provides a model to think about and discuss security concepts. It is commonly discussed in the information security literature [5] [9] [10]. The CIA triad is commonly referred to as tenets of information security. Information assets that are tied to an application can be associated to a specified CIA requirement represented as a number or values suchlike, high, medium, or low, which can be determined through risk analysis [9].

Confidentiality The first concept describes the ability to protect data from those who are authorized to view it. Confidentiality secures that the information is not exposed to unauthorized parties.

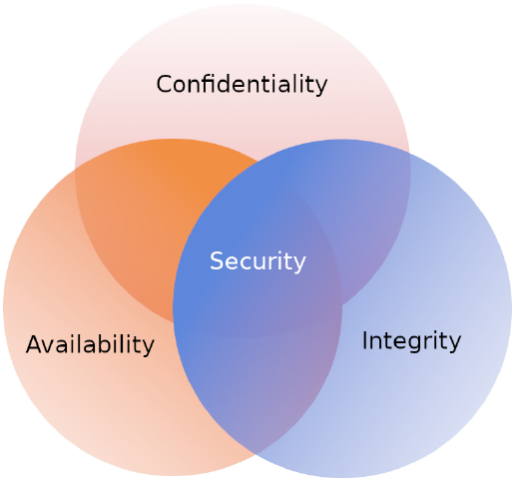

Figure 2.1: The CIA Triad

Integrity

The second concept describes the ability to prevent data from being changed in an unauthorized or undesirable manner. Integrity preserves the consistency of information both internally and externally.

Availability

The final concept describes the ability to access data when required.

The Gold Standard

Authentication Identification is the claim of identity by a person, process, or other entity without implying the authenticity of the claim or privileges that could be associated with the identity. It can be done through shortened versions of our names, images of ourselves, nicknames, account numbers, usernames, ID cards, fingerprint, DNA samples, and many more methods.

Authentication is the procedure used to validate whether the claim of identity is correct. A real world example of authentication would be the usage of a username and password combination inside an application. Depending on the security level required of an asset, more factors can be used for the authentication mechanism, also known as multifactor authentication.

Authorization Besides claiming an identity and confirming the validity of that claim, we need to decide what the party is allowed to do and if access to specific resources are allowed or denied. This can be accomplished through the concepts: authorization and access control.

Principle of least privilege The principle of least privilege is an important authorization concept which mandates that only the bare minimum of access to a party should be allowed to function. As an example, a user account is only granted the access needed to perform their routine work. Violation of the principle is the

heart of many security problems faced today. However, it is a very simple security measure that has little to no cost to develop, and it is very effective.

Access control At a high level, access control is about restricting access to a resource. Access control can be divided into two groups to either improve the design of physical security or cybersecurity. Generally, four basic actions can be carried out: allowing, access, denying access, limiting access, and revoking access. Through these actions, most access control issues or situations can be described. Furthermore, It is best practice to deny access by default, with the authorized users only being granted access.

There are two main methods that can be considered to implement access controls: access control lists (ACLs) and capabilities. ACLs, often referred to as "ackles", are a very common choice of access control implementation. Typically, ACLs are implemented in the file systems on which our operating systems run and to control the flow of traffic in the networks to which our systems are connected. A capability-based approach to security are oriented around the use of a token that controls our access. A good analogy would be the usage of a personal badge that grants access to certain doors inside a building. Notably, the right to access a resource is based completely on possession of the token, not who possesses it.

Auditing After going through the process of identification, authentication, and authorization, it is important to keep track of the activities that have occurred. Despite access being granted to the party, it is important that the party behaves according to the rules as it concerns to security, ethics, business conduct, and so on. With an abundance of information in digital form, including medical data, financial information, legal proceedings, trade secrets, and many more items, it has become a vital task to ensure that rules set forth are abided by.

Accountability Accountability depends on identification, authentication, and access control being present in order to trace activities and associate the transactions back to the source. Moreover, maintaining accountability ensures that organizations are in compliance with any laws or regulations associated with the type of data being handled or the industry in which the organization operates.

From a security perspective, accountability introduces several beneficial features. Without the information that is being monitored or logged, there would be no foundation to maintain a higher security posture.

Auditing A foremost approach to assure accountability through technical means is by ensuring that accurate records are registered of who did what when they did it. Without auditing there exists no ability to assess activities over a period of time. Consequently, there is no way to assist accountability on a large scale. Within the scope of information security, we primarily look at access to or from systems. Some common audited items are passwords and software licensing.

Cryptography

Cryptography is the science of keeping information secure in the sense of confidentiality and integrity. The foremost security measure allowing cryptography

is encryption, and often the terms are used interchangeably. Although in reality, encryption is a subset of cryptography. Encryption is the transformation of plaintext into ciphertext. Cryptanalysis defines the science of breaking through the encryption used to create the ciphertext. The study of cryptography and cryptanalysis is described as cryptology.

Cryptology is not a recent invention. At the very least cryptology can be traced back as far as 2500 years and was considered an obscure science. It was well established with both ancient Greeks and Romans who practiced different forms of cryptography. A classic example of ancient cryptography is the Ceasar cipher as seen in Figure 2.2. After the fall of the Roman Empire, cryptology was flourishing in the Arabic world [11].

Plaintext Alphabet	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext Alphabet	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Figure 2.2: Ceasar Cipher

Without cryptography, much of the internet-based activities we benefit from today would be at great risk. In fact, cryptography is essential in computing, networking and the great set of transactions that take place over such devices in everyday life. Cryptography has permitted us to become a very network-centric society. Data can be protected at rest, in motion, and to a certain extent, in use, because of cryptography. Thus allowing us to securely communicate and perform transactions when sensitive data is involved.

The process of encrypting plaintext and decrypting ciphertext is described as a cryptographic algorithm. In order to either encrypt or decrypt a message, cryptographic algorithms commonly use a key, or multiple keys, with a range of possible values for the key referred to as the keyspace. The harder the keyspace, the harder it is to decrypt the message. We will take a brief look at some popular cryptographic algorithms.

Symmetric cryptography Symmetric cryptography, also referred to as private key cryptography, utilizes a single key for both encryption of the plaintext and the decryption of the ciphertext as can be seen in Figure 2.3. A symmetric cipher only works if both the sender and the receiver are in possession of the same key to unlock the cipher. Therefore, everyone who uses a symmetric cipher must have the same set of keys and must use them in the correct order [11].

Asymmetric cryptography When a different key is used for encryption and decryption, we have an asymmetric system in place. Asymmetric cryptography can also be referred to as public key cryptography. Asymmetric cryptography relies on a public key to encrypt data from the sender, and a private key to decrypt data that arrives at the receiving end as seen in Figure 2.4. Due to the mathematical complexity of the operations to create the private and public keys, no method exist at present to reverse the private key from the public key.

Hash functions Unlike both symmetric and asymmetric cryptography, which relies on keys for encryption and decryption, there are algorithms that do not

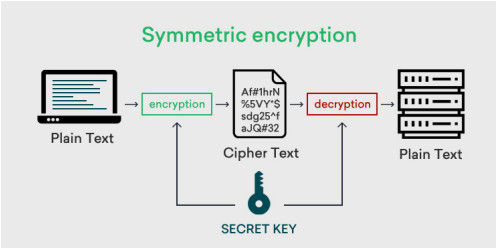

Figure 2.3: Symmetric encryption

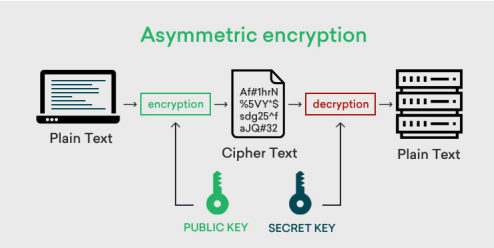

Figure 2.4: Asymmetric encryption

require keys, known as hash functions. Hash functions create a generally unique and fixed-length hash value, referred to as a hash, based on the original message as seen in Figure 2.5. Any form of change to the message will change the hash as well. Furthermore, hash functions do not allow for contents of the message to be read, though it can be utilized to determine the confidentiality of the message. Some hash algorithms include: Message-Digest 5 (MD5), MD2, MD4, SHA-2, and RACE.

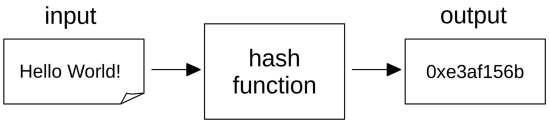

Figure 2.5: Hash function

Digital signatures A good example of where hash functions are utilized are digital signatures. To detect any changes to the content of the message, digital signatures make it possible to sign a message to ensure the authenticity from the sending party. This is accomplished by generating a hash of the message, and then use the senders private key to encrypt the hash, thereby creating a digital signature. The receiving party can use the sender's public key to decrypt the digital signature, thereby restoring the original hash of the message.

Digital signatures are now recognized as legally binding in many countries, allowing them to be used for certifying contracts or notarizing documents, for authentication of individuals or corporations, as well as components of more complex protocols. Broadly speaking, a digital signature is analogous to a handwritten signature, that provides much stronger security guarantees [12].

Certificates Another form of cryptography for message signing, is the usage of digital certificates, commonly known as certificates. Certificates link together a public key and an individual, typically by taking the public key and something to identify the individual, suchlike a name and address, and having them signed by a certificate authority (CA). A CA is a trusted entity that is responsible for digital certificates. The advantage of using a certificate is that it provides verification that a public key actually is associated with a particular individual.

Laws and Regulations

The Human Element

2.2.2 Designing Secure Software

Operation Security

Operation Security (OPSEC) is the analytical process as well a strategy utilized to identify information that potentially can be exploited by an attacker and used to collect vital information that could harm an organization's plans or reputation. Thus, OPSEC provides the means to design countermeasures to reduce or eliminate adversary exploitation.

During the Vietnam War, a team dubbed Purple Dragon observed how their adversaries were able to anticipate their strategies and tactics despite North Vietnam and the Viet Cong's incapability to decrypt the United States communications. They concluded that the U.S. was unknowingly exposing information to the enemy that was being grouped together and exploited. The same theory has been brought over from the military to the information security professionals [13].

The operation security process The OPSEC process is a five-step risk assessment that assists in identifying what information needs to be protected, analyzing the threats and vulnerabilities that might impact, and develop mitigations for those threats and vulnerabilities as shown in Figure 2.6.

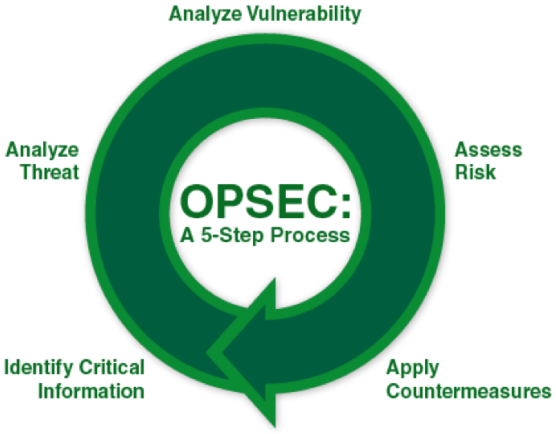

Figure 2.6: OPSEC model

Identification of critical information

Analysis of threats

Analysis of vulnerabilities

Assessment of risks

Application of countermeasures

Software Development Life Cycle

Software Assurance

The OWASP Security Knowledge Framework

Chapter 3

System Design

3.1 Project Description

This chapter describes the functional and technical requirements for developing the API and client for the application (CHEFEED) that will be developed as a case-study for this thesis. Therefore, the emphasis for the software design will be on the software's security. CheFeed will allow create, read, update and delete (CRUD) operations for posts, comments and accounts withing the native applications given the user is authenticated and authorized to do so.

The Application Programing Interface (API) will be developed with FastAPI, a modern, high-perfomace, web framework for building APIs with Python 3.6 and above with support for Python type hints. The client side will be developed with React Native, a JavaScript library for building user interfaces for both iOS and Android devices.

3.2 Overview

3.2.1 Purpose

CheFeed is social recipe sharing platform for any cooking enthusiast. In addition, the application can act as a catalog for anyone with less culinary experience. Cooking apps can contribute to cooking healthier meals, more convenient, and more fun.

3.2.2 Scope

This chapter explains the security requirements for the different components of CheFeed as they are gathered through SAMM and SKF.

3.2.3 Concepts

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] 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.

[3] Matt Bishop et al. "Evaluating Secure Programming Knowledge". In: *Information Security Education for a Global Digital Society*. Springer International Publishing, 2017, pp. 51–62. DOI: 10.1007/978-3-319-58553-6_5. URL: https://doi.org/10.1007%2F978-3-319-58553-6_5.

[4] Rossouw von Solms and Johan van Niekerk. "From information security to cyber security". In: *Computers Security* 38 (Oct. 2013), pp. 97–102. DOI: 10.1016/j.cose.2013.04.004. URL: https://doi.org/10.1016%2Fj.cose.2013.04.004.

[5] Jason Andress. *The basics of information security : understanding the fundamentals of InfoSec in theory and practice*. Waltham, MA: Syngress, 2014. ISBN: 0128007443.

[6] Yuchong Li and Qinghui Liu. "A comprehensive review study of cyber-attacks and cyber security Emerging trends and recent developments". In: *Energy Reports* 7 (Nov. 2021), pp. 8176–8186. DOI: 10.1016/j.egyr.2021.08.126. URL: https://doi.org/10.1016%2Fj.egyr.2021.08.126.

[7] Tamara Lopez et al. ""Hopefully We Are Mostly Secure": Views on Secure Code in Professional Practice". In: *2019 IEEE/ACM 12th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*. IEEE, May 2019. DOI: 10.1109/chase.2019.00023. URL: https://doi.org/10.1109%2Fchase.2019.00023.

[8] James Helfrich. *Security for software engineers*. Boca Raton: CRC Press, 2019. ISBN: 9781138583825.

[9] M. L. Srinivasan. *CISSP in 21 days : boost your confidence and get the competitive edge you need to crack the exam in just 21 days*. Birmingham, UK: Packt Publishing, 2016. ISBN: 9781785884498.

[10] Darren Death. *Information security handbook : develop a threat model and incident response strategy to build a strong information security framework*. Birmingham, UK: Packt Publishing, 2017. ISBN: 9781788478830.

[11] John Dooley. *History of cryptography and cryptanalysis : codes, ciphers, and their algorithms*. Cham, Switzerland: Springer, 2018. ISBN: 9783319904436.

[12] Jonathan Katz. *Digital signatures*. New York: Springer, 2010. ISBN: 0387277110.

[13] Abi Tyas Tunggal. *What is Operations Security (OPSEC)?* What is Operations Security (OPSEC)? Aug. 24, 2021. URL: `https://www.upguard.com/blog/opsec` (visited on 03/17/2022).