Faculty of Information and Communication Technology

A TEMPLATE-BASED APPROACH TO WRITE COMPLETE SECURITY REQUIREMENTS FOR SOFTWARE DEVELOPMENT ENVIRONMENT

Nuridawati binti Mustafa

Doctor of Philosophy

2020

A TEMPLATE-BASED APPROACH TO WRITE COMPLETE SECURITY REQUIREMENTS FOR SOFTWARE DEVELOPMENT ENVIRONMENT

NURIDAWATI BINTI MUSTAFA

A thesis submitted
in fulfillment of the requirements for the degree of Doctor of Philosophy

Faculty of Information and Communication Technology

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2020

DECLARATION

I declare that this thesis entitled "A Template-based Approach to Write Complete Security Requirements for Software Development Environment" is the result of my own research except as cited in the references. The thesis has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature : ……..

Name : Nuridawati Binti Mustafa

Date : ……..

APPROVAL

I hereby declare that I have read this thesis and in my opinion this thesis is sufficient in terms of scope and quality for the award of Doctor of Philosophy.

Signature	:	……..**.....**
Supervisor Name	:	Associate Professor Ts. Dr. Massila Kamalrudin
Date	:	……..…….**.....**

DEDICATION

I dedicate this thesis to

My darling husband Mohd Izani, my lovely daughter Nur Aisha, my beloved mother Zabidah, families and my in-laws.

ABSTRACT

Writing quality security requirements contributes to the success of secure software development. It has been a common practice to include security requirements in a software system after the system is defined. Thus, incorporating security requirements at a later stage of software development will increase the risks of security vulnerabilities in software development. However, the process of writing security requirements is tedious and complex. There are a few gaps found in the existing works, categorized into method-related and people-related issues. The method-related issues include the lack of checking on security requirements completeness, security requirements templates, security standards used as reference and automated tool for validation. While, the people-related issues consist of inexperienced requirements engineers, minimal involvement of technical team in defining security requirements and language barriers. Motivated from these gaps, the main objective of this study is to propose a template-based approach to write complete security requirements. This study proposes a new template-based approach to assist the requirements engineers and client-stakeholders for writing complete security requirements. For this, we integrate the template-based approach with security requirements density using probability ratio, syntax-based density using lexical density and security requirements completeness prioritization using numerical assignment. We also developed two new pattern libraries, SecLib and SRCLib to validate the syntax and the completeness of security requirements. Additionally, an automated tool support called SecureMEReq was also developed to realize the approach. Finally, a comprehensive evaluation of the approach, comprising the comparison study between manual and automated tool as well as usability test were conducted. In summary, the findings of the evaluations show that our approach can contribute to the body of knowledge of requirements engineering, especially in enhancing the completeness of writing security requirements. It is found that the approach is able to enhance the completeness level of security requirements compared to the manual approach and produce a complete generation of security requirements. The results of the usability tests show that the approach is useful and helpful in eliciting complete security requirements of software development and able to ease the security requirements elicitation process.

PENDEKATAN BERASASKAN TEMPLAT UNTUK MENULIS KEPERLUAN KESELAMATAN YANG LENGKAP BAGI PERSEKITARAN PEMBANGUNAN PERISIAN

ABSTRAK

Penulisan keperluan keselamatan yang berkualiti menyumbang kepada pembangunan perisian keselamatan yang berjaya. Ianya merupakan amalan umum untuk memasukkan keperluan keselamatan dalam sistem perisian selepas sesebuah sistem ditakrifkan. Oleh itu, penggabungan keperluan keselamatan dalam peringkat yang terkemudian dalam pembangunan perisian akan meningkatkan risiko dalam pengenalan serangan keselamatan ke dalam pembangunan perisian. Walau bagaimanapun, proses untuk menulis keperluan keselamatan adalah rumit dan kompleks. Terdapat beberapa jurang yang dijumpai di dalam kerja yang sedia ada, dikategorikan sebagai isu yang berkaitan dengan kaedah dan isu yang berkaitan dengan manusia. Isu-isu yang berkaitan dengan kaedah termasuklah kurang semakan keatas kesempurnaan keperluan keselamtan, keperluan keselamatan, piawai keselamatan digunakan sebagai rujukan dan alatan sokongan automatik untuk pengesahan. Manakala, isu-isu berkait-orang terdiri daripada jurutera keperluan yang tidak berpengalaman, penglibatan pasukan teknikal yang minimal dalam mentakrifkan keperluan keselamatan dan batasan bahasa. Motivasi kepada jurang ini, objektif utama kajian ini adalah untuk membangunkan pendekatan berasaskan templat untuk menulis keperluan keselamatan yang lengkap. Kajian ini mencadangkan pendekatan baru berasaskan templat untuk membantu jurutera keperluan dan pihak berkepentingan - pelanggan bagi penulisan keperluan keselamatan yang lengkap. Oleh itu, kami menggabungkan pendekatan berasaskan templat dengan kepadatan keperluan keselamatan menggunakan nisbah kebarangkalian, kepadatan berasaskan sintaks menggunakan kepadatan leksikal dan keutamaan kesempurnaan keperluan keselamatan menggunakan umpukan berangka. Kami juga membangunkan dua pustaka corak yang baru SecLib dan SRCLib untuk mengesahkan sintaks dan kesempurnaan bagi keperluan keselamatan. Tambahan, satu alatan sokongan automatik dipanggil SecureMEReq telah dibangunkan untuk merealisasikan pendekatan tersebut. Akhir sekali, satu penilaian menyeluruh bagi pendekatan, merangkumi perbandingan kajian diantara manual dan alatan automatik dan juga ujian kebolehgunaan telah dijalankan. Kesimpulannya, dapatan daripada penilaian menunjukkan pendekatan kami mampu menyumbang kepada badan pengetahuan kejuruteraan keperluan terutamanya dalam meningkatkan kesempurnaan dalam penulisan keperluan keselamatan. Didapati bahawa pendekatan ini mampu untuk meningkatkan aras kesempurnaan bagi keperluan keselamatan berbanding dengan pendekatan manual dan menghasilkan satu penjanaan keperluan keselamatan yang lengkap. Keputusan ujian kebolehgunaan menunjukkan bahawa pendekatan ini berguna dan membantu dalam mencungkil keperluan keselamatan yang lengkap bagi pembangunan perisian dan mampu untuk memudahkan proses pencungkilan keperluan keselamatan.

ACKNOWLEDGEMENTS

All praise and thanks belong to Allah the Most Gracious, the Most Merciful for choosing me to experience this wonderful journey. I was blessed with good health, strength, and ability to complete this study.

I would like to thank all who contributed in the completion of this thesis. My sincere thanks go to my two supervisors, Associate Professor Ts. Dr. Massila Kamalrudin and Associate Professor Dr. Safiah Sidek for the patience guidance, encouragement and advices toward the completion of this thesis. I am extremely lucky to have committed supervisors who were willing to spare their precious time to help me with my study; for many fruitful discussions and constructive suggestion for improvement.

My deepest thanks go to my darling husband Mohd Izani, my lovely daughter Nur Aisha, my beloved mother Zabidah and the rest of my family who always be with me through thick and thin and especially to my in-laws family for their incomparable patience, support and understanding.

In addition, I would like to thank everyone from my research group and friends for their continuous assistance and support.

Thanks for all your encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS

BoK	-	Body of Knowledge
CC	-	Common Criteria
CD	-	Cognitive Dimension
CLASP	-	Comprehensive, Lightweight Application Security Process
DIGS	-	Discovering Goals for Security
EUC	-	Essential Use Case
EUI	-	Essential User Interface
FBI	-	Federal Bureau of Investigation
GBRAM	-	Goal-Based Requirements Analysis Method
GUI	-	Graphical User Interface
IC3	-	Internet Crime Complaint Cente
ISO/IEC	-	International Organization for Standardization/ International Electrotechnical Commission
ISMS	-	Information Security Management System
IS	-	Information System
IT	-	Information Technology
KAOS	-	Keep All Objectives Satisfied
MCOQR	-	Misuse Case Oriented Quality Requirements
MSRA	-	Multilateral Security Requirements Analysis
PICOC	-	Population, Intervention, Comparison, Outcomes and Context

PBSE	-	Pattern-based System and Software Engineering
QA	-	Quality Assessments
RE	-	Requirement Engineer
RQ	-	Research Question
SDLC	-	System Development Life Cycle
SecureMEReq	-	Security Requirements Tool
SLR	-	Systematic Literature Review
SQUARE	-	Security Quality Requirements Engineering Methodology
SR-CP	-	Security Requirements Completeness Prioritization
SREP	-	Security Requirements Engineering Process
SR-PD	-	Security Requirements Probability Density
SRS	-	Software Requirement Specification
SR-SD	-	Security Requirements Syntax Density
TBAT	-	Template-Based Authoring Tool
TDD	-	Test Driven Development
UML	-	Unified Modelling Language
UTeM	-	Universiti Teknikal Malaysia Melaka

LIST OF PUBLICATIONS

1. Mustafa, N., Kamalrudin, M., and Sidek, S., 2019. SecureMEReq: A Tool Support to Check for Completeness of Security Requirements. *International Journal of Recent Technology and Engineering (IJRTE)* ISSN: 2277-3878, 8(2S11), pp. 768-771.

2. Mustafa, N., Kamalrudin, M., and Sidek, S., 2018. Writing Good Security Requirements. *The Turkish Online Journal of Design, Art and Communication (TOJDAC)* ISSN: 2146-5193, September 2018 Special Edition pp. 2503-2511, In: International Symposium on Research in Innovation and Sustainability (ISoRIS 2018).

3. Mustafa, N., Kamalrudin, M., and Sidek, S., 2018. Security Requirements Elicitation And Consistency Validation: A Systematic Literature Review. *Journal of Theoretical and Applied Information Technology (JATIT)*, 96(16), pp. 5413-5424.

4. Kamalrudin, M., Mustafa, N., and Sidek, S., 2017. A Preliminary Study: Challenges in Capturing Security Requirements and Consistency Checking By Requirement Engineers. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 10(1-7), pp. 5-9.

5. Mustafa, N. and Kamalrudin, M., 2017. A New Consistency Validation Approach To Enhance The Quality Of Functional Security Requirements For Secure Software. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 10(2-2), pp. 73-76.

6. Kamalrudin, M., Mustafa, N., and Sidek, S., 2017. A Template for Writing Security Requirements. *Springer Communications in Computer and Information Science Book Series (CCIS)*, 809, pp. 73-86.

7. Mustafa, N. and Kamalrudin, M., 2017. Consistency Validation of Functional Security Requirements for Secure Software. *Proceedings of International Symposium on Research in Innovation and Sustainability (ISORIS)* , pp. 347-350.

CHAPTER 1

INTRODUCTION

1.1 Introduction

This chapter presents an overview of this thesis. First, it describes the background of the research and introduces the motivation of the research. The next section presents the research questions as well as the objectives of the research, followed by the description of the contribution of the study in relation to the field of Requirements Engineering. Finally, the chapter concludes with the outline of the thesis structure.

1.2 Research background

Secure software practices are gradually gaining relevance among software practitioners and researchers. This is happening because today, more than ever software is becoming part of our lives and cybercrimes are constantly appearing (Sánchez-Gordón et. al., 2017).

In 2015, cybercrime victims forked over $24 million across nearly 2,500 ransomware cases reported to the Federal Bureau of Investigation (FBI)'s Internet Crime Complaint Center (IC3). Meanwhile, Cybersecurity Ventures predicts global annual cybercrime costs will grow from $3 trillion in 2015 to $6 trillion annually by 2021 (Morgan, 2016). Here, attackers exploit software vulnerabilities and cause threats to the systems (El-Hadary and El-Kassas, 2014). It includes damage and destruction of data, stolen money, lost productivity, theft of intellectual property, theft of personal and financial data, embezzlement, fraud, post-attack disruption to the normal course of

business, forensic investigation, restoration and deletion of hacked data and systems, and reputational harm (Morgan, 2016). Therefore, security becomes an important issue and a crucial requirement for software systems due to the large number of incidents and attacks targeting software systems (Daley, 2017). Security ensures that application works in a desired manner and to provide defense against security threats (Daud, 2010). The common approach towards the inclusion of security within a software system is to identify security requirements after the definition of a system in software development. Thus, incorporating security in later stages of software development will increase the risks of introducing security vulnerabilities into software (Sánchez-Gordón et. al., 2017).

Contextualized within this scenario, a better way to develop secure software is to incorporate security from the very beginning of software development. When building a secure software, it is helpful to take into account the security concerns right from the beginning of the development process (Salini and Kanmani, 2012a). Early realization of the security is important so that security problems can be tackled early enough before proceeding further in the process; hence, any rework can be avoided (Yu, 1997; Mellado et. al., 2010). Therefore, having quality security requirements is essential in contributing to the success of developing a secure software.

Capturing complete security requirements is important to the development of secure software. It needs to be completely defined because poor elicited security requirements could cause failure to the development and consume high cost (Schneider et. al., 2012). Further, incomplete security requirements could lead to incorrect generation of non-functional security requirements (Firesmith, 2007b).

Security requirements can be defined as a system specification of its required security, such as the specification towards types and levels of protection that necessary for the data, information, and application of the systems. Examples of security

requirements are authentication requirements, authorization requirements, intrusion detection requirements, and many others (Firesmith, 2003a). Security requirements are also divided into functional and non-functional requirements (Slankas et. al., 2014).

However, one of the most common problems of requirement engineering in the industry is poor requirements quality. This relates to ambiguous, incomplete, inconsistent, incorrect, infeasible, unusable, or not verifiable requirements (Firesmith, 2007a; Talha, 2018). Hence, the quality of software product and overall subsequent phases is influenced by the requirement phase quality (Davis and Zowghi, 2006; Alshazly et. al., 2014). According to Matsugu (2018), the delivery of late product, poor quality of product, degraded design and documentation integrity, and delivery of invalid features caused by poor requirements are very real and give significant impacts. Research by Anuar et. al. (2015) agreed that most documented requirement specification were in poor quality. These constraints are also affected by the quality of security requirements. This is due to the elicitation of incomplete security requirements and low clarity security requirements.

It is also found that, most of the requirements engineers faced problems to elicit security requirements from the clients-stakeholders as there are instances of mismatch between the real needs and the security terms used (Houmb et. al., 2010; Banerjee et. al., 2015). In addition, the process of eliciting security requirements is complex and requires Requirement Engineer (RE) to have security experience in the process of eliciting consistent security requirements from the clients-stakeholders. Therefore, these resulted in the elicitation of incomplete security requirements.

At present, when capturing security requirements from clients, RE often uses some forms of natural language, written either by clients or themselves. These requirements are captured from the discussion and negotiation between both parties; clients and the RE. However, due to the ambiguities and complexities of natural language

(Kamsties and Paech, 2000; Bano, 2016) and the process of capturing, these requirements often have incompleteness which finally lead to the development of insecure software. Besides, RE also faced problems in eliciting consistent security compliance requirements from the clients-stakeholders as they misunderstood the real needs and the security terms used (Kamalrudin et. al., 2017a).

1.3 What are requirements?

Requirements are the main element of a software development project that must be well-defined to ensure they correctly represent the users' need. This is to avoid any misinterpretation, misconception or misunderstanding among client-stakeholders. Poor qualities of requirements, such as incompleteness, inconsistency or ambiguous requirements have a critical impact on the quality of the developed software as well as the success of the project (Boota et. al., 2014).

Requirement is a property that must be exhibited by something in order to solve some problem in the real world. It may aim to automate part of a task for someone to support the business processes of an organization, to correct shortcomings of existing software, or to control a device—to name just a few of the many problems for which software solutions are possible (SWEBOK, 2019). They are captured at the first stage of Requirements Engineering process. It is the basic element of a project that contains the formal expression of client-stakeholders' needs and expectations of a system to satisfy their business objectives (Wen et. al., 2012; Azadegan et. al., 2013; Marques-lucena et. al., 2015). Subsequently, it describes "*how the system should behave, constraints on the system's application domain information, constraints on the system operation or specification of a system property or attribute*" (Kotonya and Sommerville, 1998; Kamalrudin, 2009).