

**A READINESS MODEL
FOR SECURE REQUIREMENTS ENGINEERING**

BY
YUSUF MUFTI MUZAMMIL

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES
KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

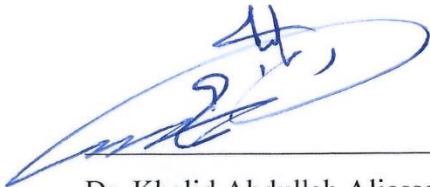
SOFTWARE ENGINEERING

DECEMBER 2017


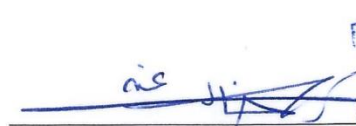
KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN- 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by Yusuf Mufti Muzammil under the direction his thesis advisor and approved by his thesis committee, has been presented and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the of **MASTER OF SCIENCE IN SOFTWARE ENGINEERING**.




Dr. Khalid Abdullah Aljasser
Department Chairman




Prof. Salam A. H. Zummo
Dean of Graduate Studies

8/2/2018
Date




8/11/2018

Dr. Mahmoud Niazi
(Advisor)



Dr. Mohammad Alshayeb
(Member)



Dr. Sajjad Mahmood
(Member)

This thesis is dedicated to Allah, Prophet Muhammad, and my family.

ACKNOWLEDGMENTS

Alhamdulillah, I was allowed by Allah to complete this thesis. This thesis would not exist without the help and support from several people. I am grateful to my advisor, Dr. Mahmood Niazi Khan, who teaches, motivates, and guides me to complete this thesis. Many thanks to Dr. Mohammad Alshayeb and Dr. Sajjad Mahmood, my committee members, who encourage me to improve the quality of this thesis.

I am indebted to King Fahd University of Petroleum and Mineral (KFUPM), College of Computer Science and Engineering (CCSE), and especially Information and Computer Science (ICS) department with the faculties that provide me the chance to learn software engineering knowledge in depth. I am also indebted to Mr. Basirudin, Mr. Daru, and Mr. Moustafa Al Saleh who have participated in my research.

Many thanks to my family in Indonesia, especially to Mr. Muzammil and Mrs. Zuhripah (my parents), Mr. Sadirman and Mrs. Rasiyah (my parents in law), Siti Fatimah (my wife), Qolbi and Qori (my sons), my brothers and my sisters, who always pray, motivate, and support me.

Last but not least, thanks to all of my friends, Indonesian student community (PPMI Dhahran), and KFUPM friends. Special thanks to my colleagues, Mr. Abdul Latif, Mr. Teguh Kurniawan, Mr. Rangga G. Noegraha, Mr. Iswan Pradiptya, and Mr. Aviandy W. Ismanto.

|

TABLE OF CONTENTS

ACKNOWLEDGMENTS	V
TABLE OF CONTENTS	VI
LIST OF TABLES	IX
LIST OF FIGURES	X
LIST OF ABBREVIATIONS.....	XI
ABSTRACT	XII
ملخص الرسالة.....	XIII
CHAPTER 1 INTRODUCTION.....	1
1.1 GENERAL	1
1.2 PROBLEM STATEMENT	2
1.3 OBJECTIVE.....	3
1.4 CONTRIBUTIONS	5
1.5 RESEARCH METHODOLOGY.....	6
1.6 THESIS OUTLINE	7
CHAPTER 2 BACKGROUND	8
2.1 SECURITY	8
2.1.1 DEFINITION	8
2.1.2 SECURITY STANDARDS	9
2.1.3 SECURITY OBJECTS	12
2.2 REQUIREMENTS ENGINEERING	20
2.3 SECURITY REQUIREMENTS	21
2.4 SECURITY REQUIREMENTS ENGINEERING	22
2.5 READINESS MODELS.....	23
2.6 RELATED WORKS ON SECURITY REQUIREMENTS ENGINEERING	23
2.7 MISSING WORK	26
CHAPTER 3 RESEARCH METHODOLOGY	27
3.1 INTRODUCTION	27
3.2 SYSTEMATIC MAPPING STUDY.....	27

3.2.1	DEFINING RESEARCH QUESTIONS.....	29
3.2.2	DEVELOPING PROTOCOL	30
3.2.3	SEARCH STRATEGY	30
3.2.4	RESEARCH DIGITAL LIBRARIES.....	30
3.2.5	SELECTION CRITERIA.....	31
3.2.6	QUALITY ASSESSMENT	31
3.2.7	EXTRACTION FORM	32
3.2.8	COLLECTING RELEVANT STUDIES	32
3.2.9	DATA EXTRACTION PROCESS	32
3.2.10	ANALYZING AND PRESENTING RESULT.....	33
3.3	READINESS MODEL DEVELOPMENT	33
3.4	CASE STUDY	33
CHAPTER 4 SYSTEMATIC MAPPING STUDY		35
4.1	RESEARCH QUESTIONS.....	35
4.2	REVIEW PROTOCOL.....	36
4.2.1	DETERMINING RESEARCH SOURCES	36
4.2.2	DEFINING SELECTION CRITERIA.....	37
4.2.3	DEVELOPING SEARCH STRATEGY	38
4.2.4	COLLECTING RELEVANT STUDIES.....	40
4.2.5	QUALITY ASSESSMENT CRITERIA	40
4.2.6	DATA EXTRACTION.....	41
4.3	FINDINGS	42
4.4	ANSWERING RESEARCH QUESTIONS	51
CHAPTER 5 READINESS MODEL		54
5.1	INTRODUCTION	54
5.2	STRUCTURE OF SRERM.....	55
5.2.1	PRELIMINARY LEVELS OF SRERM	56
5.2.2	COMPONENTS OF SRERM	57
5.3	ASSESSMENT TOOL	59
5.4	EVALUATION PROCESS OF SRERM	63
5.4.1	EVALUATION CRITERIA	64
5.4.2	EVALUATION ANALYSIS	64
CHAPTER 6 CASE STUDY		66

6.1	INTRODUCTION	66
6.2	RESULT.....	67
6.2.1	ORGANIZATION A	67
6.2.2	ASSESSMENT OUTCOMES OF ORGANIZATION A	68
6.2.3	ORGANIZATION B	69
6.2.4	ASSESSMENT OUTCOMES OF ORGANIZATION B	69
6.3	FEEDBACK SUMMARY	70
6.4	MODIFICATION OF SRERM	76
6.5	SECOND CASE STUDY	78
6.5.1	SECOND CASE STUDY OF ORGANIZATION A.....	78
6.5.2	ASSESSMENT OUTCOMES OF SECOND CASE STUDY OF ORGANIZATION A.....	79
6.5.3	SECOND CASE STUDY OF ORGANIZATION C.....	80
6.5.4	ASSESSMENT OUTCOMES OF SECOND CASE STUDY OF ORGANIZATION C.....	80
6.6	CASE STUDIES LESSON LEARNED	84
6.7	THREAT TO VALIDITY	85
CHAPTER 7 CONCLUSION		87
7.1	CONCLUSION	87
7.2	RECOMMENDATIONS.....	88
REFERENCES		89
APPENDICES		101
1.	APPENDIX 1: LIST OF PRIMARY STUDIES	101
2.	APPENDIX 2: THE PRACTICES OF SRERM	112
3.	APPENDIX 3: CASE STUDY FEEDBACK FORM	116
VITAE		120

LIST OF TABLES

Table 3.1 Example of Research Questions	29
Table 4.1 Research Questions for Systematic Mapping Study	35
Table 4.2 List of Research Sources.....	36
Table 4.3 Inclusion Criteria	37
Table 4.4 Exclusion Criteria	37
Table 4.5 Tailored Search String Based on Searching Rule in the Research Sources.....	39
Table 4.6 Quality Assessment Criteria	41
Table 4.7 Distribution of Primary Studies Based on Research Sources	43
Table 4.8 Distribution of Primary Studies Based on Publication Channel	44
Table 4.9 List of Active Authors in SRE Research	47
Table 4.10 Security Requirements Categories.....	48
Table 4.11 Security Requirements Engineering Techniques	49
Table 4.12 Security Requirements Engineering Activities.....	50
Table 5.1 Detail information of Preliminary SRERM Levels	58
Table 5.2 Motorola Assessment Tool	59
Table 5.3 The Example of Security Component Evaluation	62
Table 6.1 Implementation Score for SCs in Organization A	68
Table 6.2 Implementation Score for SCs in Organization B	70
Table 6.3 Ease of Learning Evaluation of Organization A and B	71
Table 6.4 User Satisfaction Evaluation of Organization A and B	72
Table 6.5 SRERM Structure Evaluation of Organization A and B	73
Table 6.6 Feedback Results of Organization A and B	74
Table 6.7 Detail Information of SRERM Modified Levels	77
Table 6.8 Implementation Score for SCs in Organization A	79
Table 6.9 Implementation Score for SCs in Organization C	81
Table 6.10 Ease of Learning Evaluation of Organization C	82
Table 6.11 User Satisfaction Evaluation of Organization C	83
Table 6.12 SRERM Structure Evaluation of Organization C	83

LIST OF FIGURES

Figure 2.1 The Relationship Among Software Security Knowledge Catalogues, Software Artifacts and The Best Practices of Software Security [1]	19
Figure 3.1 Systematic Mapping Study	28
Figure 4.1 Research Sources of Selected Studies	43
Figure 4.2 Selected Studies Based on Publication Channel	44
Figure 4.3 Selected Studies Based on Publication Year	45
Figure 5.1 SRERM Development	54
Figure 5.2 The Structure of SRERM	55
Figure 5.3 The Preliminary SRERM Levels	57
Figure 6.1 Modified Levels of SRERM.....	77

LIST OF ABBREVIATIONS

RE	:	Requirements engineering
SC	:	Security component
SDLC	:	Software development lifecycle
SMAPS	:	Systematic mapping study
SR	:	Security requirements
SRE	:	Security requirements engineering
SRERM	:	Security requirements engineering readiness model
SPIRM	:	Software process improvement readiness model
SOVRM	:	Software outsourcing vendor readiness model
SOPM	:	Software outsourcing partnership model

ABSTRACT

Full Name : Yusuf Mufti Muzammil
Thesis Title : A Readiness Model for Secure Requirements Engineering
Major Field : Software Engineering
Date of Degree : December 2017

The number of software vulnerabilities has been increasing with the growth of Internet-enabled software provide reference. Security awareness in the requirements engineering stage of software development is important in building secure software. Currently, there is no way to measure the readiness of security requirements engineering in an organization. The objective of this study is to develop a security requirements engineering readiness model (SRERM). Its purpose is to provide a model to assess security requirements engineering (SRE) readiness levels in organizations. In order to achieve this goal, a systematic mapping study was conducted to identify the relevant studies in the SRE domain. After analyzing 104 primary studies, 12 security requirements categories were identified and utilized to build a SRERM. Case studies were conducted into two software development organizations to validate the usability of the SRERM. Based on the case studies, the SRERM is applicable and has the ability to identify the readiness levels of SRE in the software organizations.

ملخص الرسالة

الاسم الكامل: يوسف مفتي مزمل

عنوان الرسالة: نموذج الجاهزية لأمن هندسه المتطلبات

التخصص: هندسة البرمجيات

تاريخ الدرجة العلمية: ديسمبر 2017

عدد الثغرات الامنية للبرمجيات تتزايد مع تطور البرامج التي تدعم الإنترنت توفير مرجعيه. إن الوعي الأمني في مرحلة هندسة المتطلبات لتطوير البرمجيات مهم في بناء البرمجيات الآمنة. حالياً، لا توجد طريقة لقياس الجاهزية لأمن هندسه المتطلبات في منظمة ما. والهدف من هذه الدراسة هو تطوير نموذج الجاهزية لأمن هندسه المتطلبات (SRERM). الغرض منه هو توفير نموذج لتقييم مستويات الجاهزية لأمن هندسه المتطلبات (SRE) في المنظمات. من أجل تحقيق هذا الهدف، تم إجراء دراسة منهجية لتحديد الدراسات ذات الصلة في مجال أمن هندسه المتطلبات. وبعد تحليل 104 دراسة أولية، تم تحديد 12 فئة من المتطلبات الأمنية وتم الإستفاده منها لبناء نموذج جاهزية لأمن هندسه المتطلبات. وقد أجريت دراسات حالة في منطمتين لتطوير البرمجيات للتحقق من صحة قابلية الاستخدام لنموذج الجاهزية لأمن هندسه المتطلبات (SRERM). وإستناداً إلى دراسات الحالة، فنموذج الجاهزية لأمن هندسه المتطلبات قابل للتطبيق ولديه القدرة على تحديد مستويات الجاهزية لأمن هندسه المتطلبات (SRE) في منظمات البرمجيات.

CHAPTER 1

INTRODUCTION

1.1 General

The number of vulnerabilities of software has been increasing with the growth of Internet-enabled software [1]. Security awareness in the requirements engineering (RE) stage of the software development lifecycle (SDLC) is important in building secure software. Currently, security issues gain more attention because of the popularity of social networking systems and cloud computing. Due to the increasing number of users around the world, both cloud computing and social networking systems have more challenges in securing the availability of the system, the integrity of transferred data and the confidentiality of information control [2], [3].

There are a number of common challenges to building secure software. Flaws, bugs, and defects in software are urgent issues and generally demand high attention. According to McGraw [4], it is motivated by the connectivity, complexity and extensibility of the software. Then, various attacks, such as buffer flows, race conditions and incomplete mitigation, could utilize software flaws to disclose access.

In addition, malware (malicious software) also becomes a challenge to building secure software. Stamp [5] lists various types of malware that are harmful to software, such as viruses, worms, Trojan Horses, trap doors, rabbits and spyware. Some solutions

are available for mitigating malware: signatures, changes, and anomaly detections. For example, to filter malware, users are encouraged to install antivirus software for desktops, network devices, mail gateways and network gateways [3]. However, these are not sufficient because software requirements are commonly changing over time, so various existing security mechanisms would not be relevant [2]. Therefore, security awareness in RE activity should be encouraged.

Integrating security awareness into the RE stage of the software development lifecycle (SDLC) is an active area of research and needs to be applied to the real-world software industry [6]–[8] . This topic is popularly known as security requirements engineering (SRE). For instance, capturing SR has been a popular area of research, discussed by dozens of researchers for more than two decades [9]–[12]. Recently, it is still applied to cloud computing [13] and Internet-of-things (IoT) [14] research.

In addition, based on Salini and Kanmani’s survey [15], some established frameworks can be considered to answer SRE integration challenges, such as SQUARE, SREP, Microsoft Trustworthy SDLC, CLASP, Secure Tropos, Charles Haley, McGraw, Appvrille and Pourzandi, Gustav Bostrom and Colleagues, Eduardo Fernandez, and Gunnar Peterson. Salini and Kanmani argued each framework has different advantages and disadvantages. However, the recommended SRE framework, in their opinion, is SQUARE due to its capabilities.

1.2 Problem Statement

Each software industry has its own approach to address SRE challenges. For example, one could hire a security expert or provide a workshop to train and encourage security awareness to the software developers. An organization also could hire a security

consultant to audit their SRE activities. Note that these approaches could only be applied to large-scale organizations due to the high costs.

Software technologies, types of vulnerabilities, and innovations in security mechanisms are frequently changing [16]. This has brought about a large amount of published research discussing SRE in term of techniques, guidelines and frameworks. Most publications discuss the techniques to perform SRE, while the rest tend to build an SRE framework. However, there is no study yet which provides a technique or tool for software organizations to identify their SRE readiness in software development. In other words, software organizations might perform SRE without evaluation.

The anticipated technique or tool should be validated in terms of usability and reliability in the real-world software industry. It should be implementable not only in large organizations, but also in smaller ones. In addition, to achieve high impact, it has to encompass most security requirements. After discovering the problems and challenges clearly, objectives can be defined.

1.3 Objective

Main objective of this study is developing a readiness model for security requirements engineering. This study aims to develop a readiness model that solves the problems of SRE, including the challenges, presented in the previous section. The readiness model is expected to have the ability to determine SRE readiness in an organization, to encompass relevant security requirements, and to be reliable in various software organizations.

There are two sub objectives required to support development of readiness model for SRE as follow:

1. To achieve a readiness model with high quality, we need to collect comprehensive information related to SRE. This study utilizes a SMAPS method [17], [18] to recognize security requirements engineering publications, including available techniques, which are readily accessible in the research electronic databases. A SMAPS is a powerful technique to discover relevant literature on SRE and comprehensively present extensive information. As a result, it could minimize missing important issues, various definitions, or recent improvements related to SRE. Thereafter, the obtained information could be used in constructing the readiness model.
2. To measure the applicability and usability of the readiness model, we need to conduct a case study. A case study in software organizations is required to evaluate the security requirements engineering readiness model (SRERM) usability [19], [20] . This approach could capture the missing perception between the literature and real-world software organizations. One common issue is that the suggested solution in published research cannot be implemented in the organization. The reason is different thinking about the environment and the software policy of the organization. For example, a software organization needs a solution which requires reduced cost and time, but the existing research requires a high level of effort. Consequently, the research recommendation will not be utilized. Therefore, a case study should be conducted properly in this research.

1.4 Contributions

Research will be valuable if it makes a contribution to knowledge. Typically, research contributions in the software engineering domain could propose methodology improvement, offer a new technique, build a framework, present a survey, or develop a model. Three contributions of this study are described below.

1. Systematic mapping study (SMAPS). This study conducted the SMAPS to gain extensive information related to SRE. The obtained result could help researchers in determining the current state of SRE and investigate the next interesting research. Various types of information resulted from the SMAPS, such as which security requirement category was discussed, which framework was developed, which digital sources were utilized, and which requirement activity was focused on in the publications. In addition, the most active researchers in SRE were also presented.
2. Security requirements engineering readiness model (SRERM). The SRERM utilizes the outcomes of the SMAPS, for example, security requirements categories, as an important part. To achieve the objectives, an iterative construction was performed in the SRERM development. As a result, this model has the ability to determine the readiness level of organizations in performing SRE. After applying the SRERM in the project, software developers are expected to have more security awareness in general, especially in SRE activities. It will also show them how to improve their SRE performance in future.

3. Case study. This study has conducted a case study to evaluate the usability of the SRERM in software organizations. Two organizations have participated to evaluate the SRERM based on set criteria, and have provided several constructive comments. An introduction to SRERM was essential before asking a participant to complete the post-case study questionnaire. Extracting and analyzing the outcome of case study was a challenge. As a result, based on the case study outcomes, the SRERM in general could be used in real-world software organizations.

1.5 Research Methodology

The research methodology consists of four following steps:

Phase 1: Systematic Mapping Study (SMAPS)

Five research digital sources were selected to obtain relevant studies. A specific protocol was defined to ensure the quality of the result. Then, the identification of security requirements and their practices were collected through a SMAPS on the SRE topic.

Phase 2: Developing a readiness model

The SRERM development was influenced by several published pieces of research that have presented a readiness model [19]–[21]. This study utilizes the outcomes of SMAPS to develop security requirements components, including specifying relevant practices, in constructing the SRERM. The Motorola assessment tool [22] was selected as the main part of the model due to various considerations.

Phase 3: Performing case study

In this phase, a case study was performed into two software organizations in order to evaluate the usability of the SRERM. The two organizations recommended changes, criticism, and modifications which were addressed in SRERM. This phase also had a modification section of the SRERM to accommodate feedback from respondents.

Phase 4: Performing evaluation and modification.

In the final phase, the evaluation of case study was performed to gain some feedbacks. The modification of the SRERM was performed based on respondents' suggestion.

1.6 Thesis Outline

The content of this paper is organized as follows. Chapter 2 introduces the background theories to provide a clear understanding of what this is discussed in this study, and to avoid confusion. In addition, the relevant literature that underpins this research are discussed in chapter 2. Chapter 3 comprehensively explains the research methodology applied in this research.

In chapter 4 the outcomes of the SMAPS are discussed, such as the identified security requirements, the digital research sources, and popular SRE techniques in the literature. Chapter 5 extensively describes the development of the SRERM. A case study is explained in chapter 6, including the result, feedback, and the SRERM modifications. The conclusion of the research and recommendations for future work are presented in Chapter 7.

CHAPTER 2

BACKGROUND

This chapter will present definitions of security, requirements engineering, SRE, and a readiness model, and will introduce the case study in detail. A number of SRE publications will be presented to recognize what researchers have done, and point out gaps in the research.

2.1 Security

Security has several meanings in the dictionary: things that are done to keep something, safe from danger or crime and protection from bad things. This study will examine the definition of security in the computer domain, several security standards, and various security objects.

2.1.1 Definition

In terms of computer and software, security has meant a way of thinking to protect the essential assets of the system, such as information, operating system, networking and program [3]. Its implementation has three types: defense, detection and deterrence. The most effective approach to include security into software development is donning a black hat and thinking like a bad guy [1]. However, software organizations commonly prefer to utilize existing security standards as a guideline to secure their system.

2.1.2 Security Standards

There are various security standards which are employed to assist information security management. COBIT, ISO 27001 and 27002, NIST and common criteria are the most widely discussed security standards in published studies. The reason is that these are produced by known organizations and obtain more security practitioners' attention than other types [3]. These security standards will be discussed concisely.

COBIT (Control Objective for Information and related Technology) is a well-established framework to support a company in information technology (IT) management and IT governance [23]. It was developed by ISACA (Information Systems Audit and Control Association). COBIT 5, recent version of COBIT, provides these security features: risk, information security and vulnerability management. The other features of COBIT are management of changing regulations and business goal management, which are clearly separated from the security domain.

ISO (International Organization for Standardization) 27001 and 27002 are frameworks which belong to the ISO 27000 series [24]. They specifically provide management services to develop a secure program. ISO 27001 is used for specifying the management of information security program, whereas ISO 27002 provides information security controls to support ISO 27001 [3]. To implement the ISO 27000 series, some steps, like the PDCA (Plan, Do, Check, Adjust) cycle in COBIT, should be executed. In short, ISO 27001 is close to the "Plan" concept in COBIT whereas ISO 27002 is close to the "Do" concept.

NIST (The National Institute of Standards and Technology) provides a document containing dozens of security practices to support software development in academic

organizations, software industry and government management [25]. It was named as the 800 series, which the point 800-53 specifically describes how to ensure security control [3]. It consists of 18 security control categories as listed below.

1. Access Control
2. Awareness and Training
3. Audit and Accountability
4. Security Assessment and Authorization
5. Configuration Management
6. Contingency Planning
7. Identification and Authentication
8. Incident Response
9. Maintenance
10. Media Protection
11. Physical and Environmental Protection
12. Planning
13. Personnel Security
14. Risk Assessment
15. System and Services Acquisition
16. System and Communication Protection
17. System and Information Integrity
18. Program Management.

Comparing to COBIT and the ISO 27000 series, NIST 800-53 is the only security standard which has not been updated since 2011. In contrast, COBIT was recently updated in 2017 and the ISO 27000 series was updated in 2016. While COBIT and ISO 27000 series are commercial products, NIST 800-53 is instead available for public usage. When some organizations need full support to implement security guidelines in software development, COBIT is more recommended.

The other popular free security standards are common criteria (ISO/IEC 15408) and W3C Security. The former is a security standard product developed by NIST and the National Security Agency (NSA) [26]. It focuses on providing security guidelines related to the requirements phase in SDLC. Its purpose is to assist organizations in developing security requirements to satisfy their needs. The latter security standard is W3C Security [27], which is supported by W3C community members. W3C Security provides various online web-security discussion groups, such as web authentication working, web application security, web payment, web cryptography working, privacy interests, XML security, web security interests, and hardware-based secure services groups. The resulting documents of each group contain recommended practices based on discussion to enhance security standards in the web domain.

In addition, there are other security standards which could be utilized for assisting security implementation, such as British Standard 7799 Part 3, ITIL (ISO/IEC 20000 series), SANS Security Policy Resource [28] and the security standard offered by Stanford University [29]. However, these are unpopular in SRE publications.

2.1.3 Security Objects

This section introduces various security objects in the software development lifecycle. These are computer security, network security, data security, physical security, and software security. By understanding security objects, every organization will be able to recognize the IT security management needs.

2.1.3.1 Computer Security

Computer security in this study is not limited only to explain security in the computer environment, but also the operating system, fundamental infrastructure, virtual machines, cloud computing and mobile devices security. All of these need to be discussed due to their function and support affecting the security of software. Failure in satisfying computer security will lead to various vulnerabilities.

Commonly, the kind of server operating systems for deploying software are Unix, Windows, and Linux [3], [30]. However, Peter Tsai [30] reported that the most popular server operating system in 2016 is Windows Server 2008 at 45.5 per cent. It is followed Windows Server 2012 at 23.6 per cent of, Virtual Machine at 17.9 per cent, and Linux at 11.7 per cent.

People's selection of server operating systems vary due to a number of factors: the provided administration tools, security support, stability, features, performance, hardware requirements, scalability, TCO (cost of production, administration, and downtime), and available third-party applications [31]. In addition, some practices are required to improve the security of the selected operating system as listed below [3].

1. Remove the unnecessary program to minimize attack objects.

2. Install the appropriate security software.
3. Enhance the authentication processes.
4. Limit the number of administrators with privileges.
5. Utilize firewalls to protect the services.
6. Modify the configuration of software settings.
7. Patch the system in periodically.

Fundamental infrastructure security, an important part of computer security, encompasses various items: email, web server, proxy server, and DNS [3]. Software developers need to consider the security of email. They must guard against security attacks that utilize email to attach malware, such as a fake document. The recommended practices to secure email are enhancing spam control, email protocol and malware control.

The other object of fundamental infrastructure security is the web server, which has vulnerabilities such as buffer overflow, directory traversal, script permissions, directory browsing, and old default sample web code [3]. The recommended practices are utilizing firewalls, antiviruses, secure logs, feedback analyzers, input validation and vulnerability scanners. DNS (Domain Name Service) is an object for satisfying fundamental infrastructure security, and the latest updated version needs to be installed to secure the system from DoS (Denial of Service) [3]. A proxy server, the last object of fundamental infrastructure security, needs to be provided to ensure the transferring-data process between client and server is protected.

Virtual machines (VMs), a famous term in computer security, is software that provides people an authority to install various operating systems in one single computer

hardware [3]. It works by utilizing a hypervisor to manage all guest operating systems (OSs). The best practices for securing an operating system also should be applied to VMs. However, some additional security attention is required for VMs such as managing the security control of data storage and securing the hypervisor. The detail best practices to protect VMs are listed below.

1. Utilize security standard NIST 800-125, which offers how to design and secure VMs.
2. Protect the hypervisor by installing a firewall and updating the security control configuration. Minimizing the number of administration accounts is highly recommended.
3. Protect the guest OSs by utilizing partitioning that will limit access of attack from one guest OS to another one. Another practice is by empowering the intrusion detection system (IDS) or intrusion prevention systems (IPS).
4. Protect the virtual storage by improving the configuration of files control.
5. Protect the virtual network by integrating IDS or IPS to the network configurations.

Cloud computing, a recent popular technology, is considered as a part of computer security. There are various services provided by cloud computing: infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS), utility computing, web services in the cloud, managed service providers (MSP), service commerce platforms, and internet integration [3]. The benefits of cloud computing are minimizing the cost of building new infrastructure, educating the employee, licensing additional software, and improving security. However, since 2009

the security challenges of cloud computing services are growing, such as outage, data loss, and attacks. For example, Amazon Web Services (AWS), a provider of cloud computing services, had an outage problem to their server in 2011, which meant their customers could not access the service.

Two recommended security practices for cloud computing are performing a vendor security review and analyzing the risks [3]. Discovered risks in cloud computing can be categorized into confidentiality, integrity and availability risks. For confidentiality risks, there are data theft, espionage activity, uncontrolled administration authorization, storage stability, storage platform attacks, hijacking and misuse of data. Thereafter, integrity risks encompass data loss, data tampering, accidental modification, computer failure, and phishing. Lastly, availability risks include outage, application failure, backup failure, and slowness.

Mobile device security discusses security in various existing devices such as smartphones and tablets [3]. These devices are considered as computers due to the existence of an operating system, management of files and data, and application management. Similarly, mobile devices have some risks such as file and data theft, Wi-Fi hijacking, open hotspot features, hidden Trojan applications and phishing. The recommended security practice is utilizing mobile device management (MDM) such as controlling the allowed features and applications.

2.1.3.2 Network Security

Network security discusses some solutions to secure the connection between server and client devices. The existing solutions utilize a virtual private network (VPN),

implementing an intrusion detection system (IDS) and an intrusion prevention system (IPS), and installing firewalls. However, attention is required for each solution due to various challenges.

A virtual private network (VPN) aims to virtualize the Internet connection between a particular server and client by empowering encryption and a traffic isolation technique [3]. The benefit of a VPN is allowing a system to mitigate the person in the middle and identify the suspected packages. The challenge of a VPN is how to ensure remote access is used properly by users. In addition, administrator access is prohibited for suspected emails or malware-infected websites.

Firewalls are utilized to monitor the network activities and block unauthorized access of some applications in the network [3]. This includes network address translation (NAT) to convert the IP address and records the traffic log. The challenge in implementation is when the applications encrypt their traffic, so the firewall is unable to determine whether it is allowed or not.

An intrusion detection system (IDS) and an intrusion prevention system (IPS) are techniques to notify people when strange traffic activities occur in the network [3]. Commonly, both are deployed after installing the firewall and antivirus. While the basic concept of IDS is logging the malicious activity, and alerting the administrator if unknown activity occurs, an IPS instead will block it. The management of both an IDS and an IPS will be the challenge.

2.1.3.3 Physical Security

Physical security has a number of considerations for enhancing system security [3]. As a security object, it encourages people to divide their security attention into various assets such as computer, communication, technical, storage, furniture and fixtures assets. The recommended practices to enhance physical security are listed below.

1. Ensure the doors and windows of a building's assets are locked properly.
2. Ensure the computer assets are secured by physical lock, and protected by BIOS, access to server room is limited, and enable a tracking system.
3. Ensure the location of the server is not in a disaster and war zone.
4. Ensure the location satisfies accessibility, lighting, and other required facilities.
5. Provide a closed-circuit television (CCTV) and alarm for an unexpected case.

These practices are examples of ways to maintain physical security. Paying attention to them will achieve higher-level security in the future.

2.1.3.4 Data Security

Data security covers some important topics such as database security, storage security, and data encryption. Understanding data security will encourage the awareness of information assets. While computer security and network security are the medium, data security is the object transferred. Failure to satisfy data security, the benefits of other security objects will be lost.

Storage is hardware which data reside in. There have been numerous improvements in storage over the years: floppy disk, compact disc (CD) or digital video disc (DVD), flash drive, hard drive, and currently solid-state drive (SSD) [3]. The risk

will occur in the uncontrolled storage. The recommended technique to achieve storage security is by utilizing data encryption and access control management. Administrator access must be limited properly to avoid further vulnerabilities.

A database is a system for storing and managing information such as transaction records and human identity records [3]. Paying attention to the security of database is important to secure secret information. Data encryption is a common technique to store sensitive data into a database system. Managing the people who have administrator privileges is also mandatory to ensure the security of the database. Backup and recovery are required in order to establish database security. The first challenge of database security is how to determine which data will be backed up. When the data backup is huge, the technique for recovery also needs consideration. The second challenge is how to monitor and ensure the database will survive for longer time. Both challenges require more attention to satisfy the security of data as an important asset.

2.1.3.5 Software Security

Software security has various definitions in the literature with a similar meaning. “Software security is about building secure software: designing software to be secure, making sure that software is secure, and educating software developers, architects, and users about how to build secure things” [32]. It is different with application security, which focuses on protecting the application after development [17].

Software security has three pillars (risk management, software security touchpoints, and knowledge) to encourage security awareness among team members [1]. Risk management will motivate the team how to understand the business context, how to identify the risks, how to rank the risks, how to define the mitigation strategy and validate

the solution. Software security touchpoints will encompass analyzing the architectural risk, penetration testing, abuse cases, security requirements, security operations, code review and risk-based security testing. Knowledge, in terms of software security, will provide comprehensive information such as vulnerabilities and attack patterns to enable building secure software.

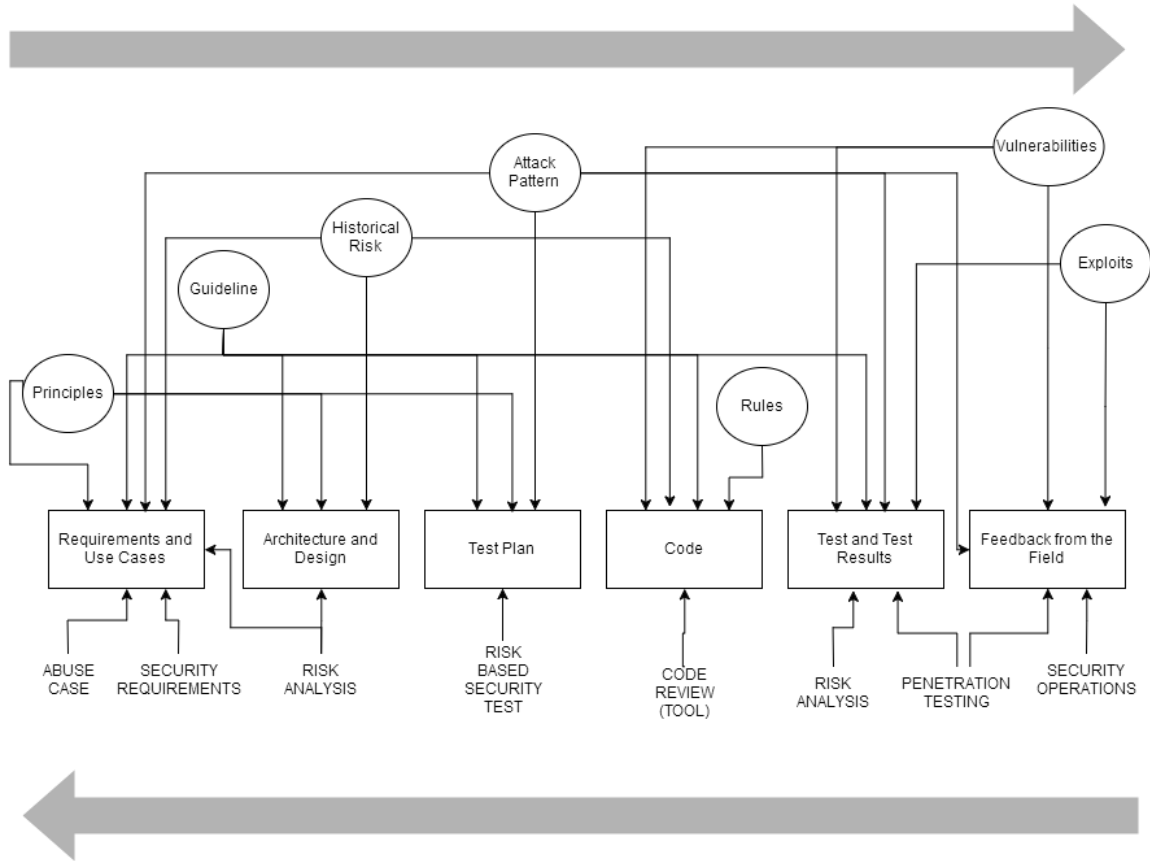


Figure 2.1 The Relationship Among Software Security Knowledge Catalogues, Software Artifacts and The Best Practices of Software Security [1]

2.2 Requirements Engineering

Requirements engineering (RE) is a beginning process in the software development lifecycle to collect and document some conditions as a reference for satisfying users and solving their problems [33]. This stage should be carefully undertaken by the project team. Failure to avoid the requirements error will lead high cost to fix it in the future. There are several popular factors that can be a challenge in the requirements engineering such as lack of user input, incomplete requirements and specifications, and changing requirements and specifications [34].

RE has three core activities: elicitation, documentation, and negotiation. These are performed iteratively to establish the software as requested by the stakeholder. Validation and management, as additional activities, support the core activities and secure the outcomes of RE [33].

In the field of software requirements, there are three types of requirements: functional requirements, nonfunctional requirements, and constraints [34].

1. Functional requirements. These are system action-oriented requirements that provide an interaction of system to the user through the inputs, outputs, and functions. One of best practices to express functional requirements is utilizing use cases.
2. Nonfunctional requirements. These requirements provide additional attribute to the system. There are four categories in this requirement type: usability, reliability, performance, and supportability.
3. Constraints. These are restrictions on the development of system that must be completed but should not affect the external behavior of the system.

2.3 Security Requirements

There are two popular definitions of security requirements (SR) in published studies. The first definition states SR is a constraint on the functions of the system, whose purpose is to satisfy one or more security goals [9], [15], [35]. SR as a constraint will specify urgent notes or restrictions of relevant security concerns to the functional requirements. For example, a functional requirement states a user's need to insert their username and password to log in to the system. SR would then have the system verify the inserted information before allowing them to access the system.

The second definition argues that SR should be considered as a functional requirement [35], [36]. This meaning is similar to the common criteria concept [37], which recommends some security mechanisms as a requirement, and provides a particular section to discuss the reasons behind them. For example, there is consideration that "the user is authenticated by using biometric devices" as a requirement. When this is documented in software requirement specifications (SRS), it will encourage people to focus on the technical security architectural mechanism and design, rather than the foundation why biometric devices are selected.

In this study, the definition of SR as a constraint is adopted, rather than as a functional requirement. In other words, security requirements will document various important assets linked to running software such as the information, the communication data and the software itself.

2.4 Security Requirements Engineering

Typically, SRE is performed in the first stage of the software development lifecycle. The main activities of SRE include eliciting, analyzing and specifying the security requirements. To support the main activities, SRE also talks about validating and managing the collected security requirements. The outcomes of SRE are a security requirement specification, which describes identified assets, detected threats, potential vulnerabilities, analyzed risks and the practices [15], [33].

Salini and Kanmani [15] state there are some published SRE methods in real software development. Some of these are McGraw's SSDL process, Microsoft's Trustworthy Computing SDLC, Aprville and Purzandi's SDLC, CLASP (Comprehensive, Lightweight Application Security Process), SQUARE (Security Quality Requirement Engineering), Haley and his colleague's framework, Security Requirement Engineering Process (SREP), and Secure Tropos. One difference among SRE above is the number of activities covered. For example, SQUARE has misuse modeling activity while Secure Tropos and CLASP do not have it. Thereafter, SREP performs asset identification activity while SQUARE does not. SREP has validation activities while Trustworthy Computing SDLC does not. In the authors' opinion, the most recommended SRE method is SREP because it covers most activities of SRE.

In addition, SRE will heighten people's awareness to improve and ensure the security of software since beginning development. It can be interpreted by analyzing the potential threats, such as abuser, attack, malware and theft. As a result, it will lead to protecting the confidentiality, integrity and availability of software and its information.

2.5 Readiness Models

In software engineering research, a readiness model was utilized by several studies. It was used by Niazi et al. [20] to assess organizational readiness in terms of software process improvement. Their readiness model has several levels: aware, defined, and optimizing. Each level is supported by some critical factors and barriers. The researchers validated their readiness model by performing case studies in three software organizations.

Similarly, Ali and Khan [38] presented a model to measure the readiness of a software organization to forming outsourcing relationships. To develop a readiness model, they utilized critical partnership factors and their practical implementation. Their readiness model also has several levels: contract, success, readiness, conversion and maturity. By utilizing case studies in two software organizations, they argue that their readiness model has the ability to assist software development outsourcing.

As a result, a readiness model can be defined as a technique to assess an organization or team based on the specified criteria to represent their level of readiness. The above studies utilize the Motorola assessment tool and a case study to show the usability of their readiness model. The challenges learned from the literature is how to construct the levels with practices that can be applied to real software organizations.

2.6 Related Works on Security Requirements Engineering

Exploring SRE will not be complete until we understand that it is a part of software security. While software security covers overall practiced security knowledge and how to integrate it in the software development lifecycle, SRE focuses only on the

early phase [1]. This part will describe some published research which motivates this study.

Capturing security requirements is a popular topic in the elicitation step of SRE. There are several studies that describe a technique to elicit security requirements in a systematic way. El-Hadary and El-Kassas [9] have proposed a technique for eliciting security requirements based on problem frames and abuse frames. They used problem frames to build a security catalog and to represent security requirements, while abuse frames are used for threats modeling. Abuse frames and problem frames were previously also utilized by Lin et al [39], [40] to collect threats and vulnerabilities for enhancing security requirements engineering.

Another technique for eliciting security requirements is misuse cases. Sindre and Opdahl [41] have proposed misuse cases to capture security threats and requirements. Misuse cases provide a visualization of the connection between use cases and misuse cases. Although misuse cases have the trustable capability for analyzing threats of functional requirements, there are some weaknesses, such as requiring the developer to have a high level of understanding to know how to improve the misuse case, and it does not cover some kinds of threats.

Tondel et al. [42] highlighted the high potential of combining misuse cases with attack trees [43] to improve security requirements elicitation. They argue attack trees can provide references of threats more detail to support the misuse cases. Gandotra et al. [44] have a similar consideration to combine the strength of misuse case and attack trees.

Similar to misuse cases, abuse cases previously have been proposed by McDermott and Fox [45]. Although both misuse cases and abuse cases employ the

concept of use case, they have an essential difference. While misuse cases are visualized in one single diagram with the use case, abuse cases instead are separated.

Recently, some research has offered a framework to overcome some activities of SRE. For instance, Dalpiaz et al. [46] proposed a SecCo framework, which focuses on elicitation and specification activity to document security requirements. SecCo works by utilizing a commitment view between actors. In addition, Saleem et al. [47] presented the framework for eliciting and modeling the security requirements from the business process model. They stated their framework is able to model the security requirements on SOA-based applications.

Furthermore, Salini and Kanmani [48] presented model oriented security requirements engineering (MOSRE) framework. They utilized a use case diagram for eliciting security requirements. MOSRE has been applied to E-Health web applications. To determine security requirements, it has the ability to identify, quantify and rank the risks of the security threats and vulnerabilities.

Mellado et al. [49] proposed SRE process for software product line (SREPPLine) framework. They utilized XML grammar and security reference model in their framework. They argued their framework conforms to ISO/IEC 27001 and common criteria linked to security requirements management concerns. In addition, common criteria [37] as a standardized guideline for eliciting, specifying, and analyzing SR, was also utilized in research by Ware et al [50]. They utilize it combined with use cases for eliciting SR.

In order to help people understand and determine which SRE method satisfies their needs, Salini and Kanmani [15] provide a comparison among SRE methods based

on activities covered, the usage in the organizations, and the techniques utilized. They analyzed and compared SQUARE, SREP, Microsoft Trustworthy SDLC, CLASP, Secure Tropos, Charles Haley, McGraw, Appvriile and Pourzandi, Gustav Bostrom and Colleagues, Eduardo Fernandez, and Gunnar Peterson.

Recently, some popular studies have discussed how to build a framework for SRE [14], [51], [52]. Other fruitful discussions talk about how to implement SRE in cloud system development [53]–[55]. In general, every new technology such as the Internet of Things (IoT) has its own security challenges. As a result, after discussing the published studies above, SRE can be considered as an active area of research.

2.7 Missing Work

Much research has been published discussing SRE in term of techniques, guidelines, and frameworks. However, it raises a challenge of how to assess the strength of SRE implementation in the software industry. There is still no study which provides a solution to identify which security area is overlooked in software development. Due to the high number of technology challenges and security threats in the future, the software industry needs an assistant or tool to indicate the readiness level of their SRE process.

The readiness model is one of recommended solution to fill the gap described above. It can be used as a long-term evaluation tool for assessing the readiness level of SRE in the organization. In addition, it can be a trigger to encourage security awareness of project team in software development, especially requirements analysis stage.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

This chapter describes the research methodology to develop a readiness model for SRE. In order to achieve the objectives, there are three phases which need to be performed. First, a SMAPS is utilized for identifying the security requirements categories and their practices. Second, a readiness model for SRE is iteratively developed by considering the outcome of the SMAPS. In the third phase, case studies are conducted to evaluate the readiness model based on software organizations' perspectives.

3.2 Systematic Mapping Study

Systematic mapping study (SMAPS) is a kind of advanced literature review [18]. SMAPS is different to a systematic literature review in term of purposes, broadness of research area, and validity issues. SMAPS will provide analysis of a specific research field and investigate the portion and category of published research and existing results in the selected field. Based on this methodology, several research questions will be used for initializing the selection criteria and data extraction form. All primary studies must satisfy the criteria to be incorporated.

There are some steps for conducting SMAPS which will be used in this thesis. The following steps refers to systematic mapping study research conducted by Petersen et al. [18] which specifically in software engineering domain.

1. Defining the research questions.
2. Developing the protocol of systematic mapping study.
3. Collecting the relevant studies by applying the search string to the different research databases.
4. Implementing the selection process into collected studies based on provided criteria.
5. Applying quality assessment into selected findings.
6. Extracting the data for each selected finding by using provided form.
7. Analyzing and presenting the collected data.

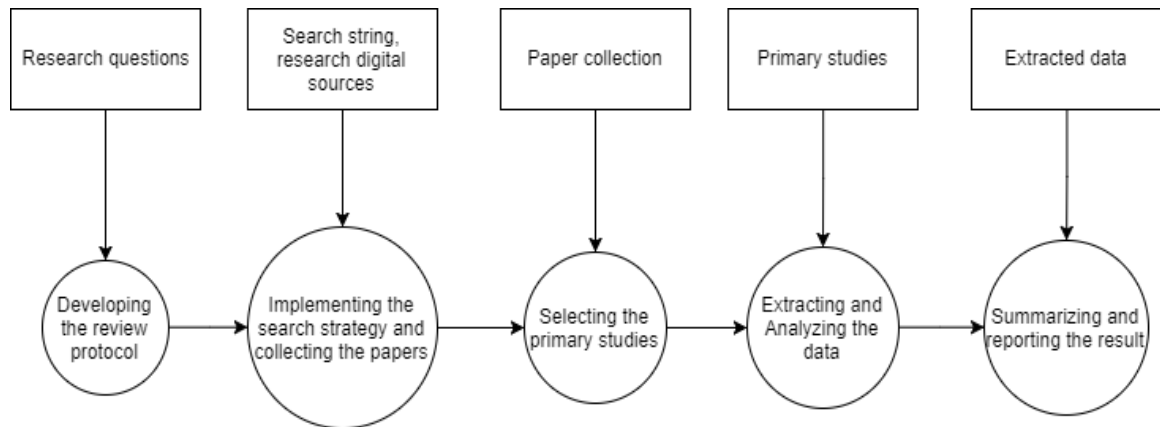


Figure 3.1 Systematic Mapping Study

3.2.1 Defining Research Questions

Research questions should be defined properly before developing the review protocol. From research questions, some relevant keywords will be identified. Since systematic mapping is time consuming, it is recommended to determine the research questions carefully. When a piece of research lacks a research question, some collected studies might be not relevant to the objectives.

The strategy to define the research question focuses on the research objectives. Since this research has objectives to develop a readiness model for SRE, the research questions are linked to the SRE topic. Commonly, the research questions in existing SMAPS research include identifying the key journals that publish relevant research. Another one will identify the relevant research in terms of the technique or method applied, the research type, the objections, and the contributions. Examples of research questions in several systematic mapping studies are shown in Table 3.1.

Table 3.1 Example of Research Questions

No.	Research questions example
1.	“What areas in software product line variability are addressed and how many articles cover the different areas?” [18]
2.	“What types of papers are published in the area and in particular what type of evaluation and novelty do they constitute?” [18]

3.2.2 Developing Protocol

The outcome of the SMAPS depends on the constructed protocol. Several important points need to be defined before collecting primary studies. These are developing a search strategy, deciding the appropriate research digital libraries as sources, determining the selection criteria, and specifying the quality assessment criteria.

3.2.3 Search Strategy

The technique to develop a search strategy contains three steps [17]. First, build the search string based on population, intervention, outcome of relevance, and experimental design. Second, find the synonym of the obtained term from the first step and improve it with Boolean operators. The final step is to combine and verify the terms previously collected.

3.2.4 Research Digital Libraries

Collecting qualified studies relies on the quality of research digital libraries. In other words, determining which research digital libraries are used is important. In the software engineering context, there are a number of popular research digital libraries, such as ACM, IEEE Xplore, Science Direct, Springer Link, and John Wiley.

Generally, these research digital libraries provide an advanced search service. However, the rule of syntax of search string in one research digital library may be different from others. For example, applying a search string in Springer will be simpler than in ACM. Based on author experience, the most challenging to apply a search string in is IEEE Xplore, which needs more iterations to improve the syntax.

3.2.5 Selection Criteria

Selection criteria in the SMAPS are employed to eliminate non-relevant studies. There are two kinds of selection criteria: inclusion criteria and exclusion criteria. Collecting studies from research digital libraries typically will obtain a large amount of research; either it is relevant or not. After defining the inclusion and exclusion criteria in the protocol design, people can obtain the relevant studies to for research questions. The implementation of selection criteria is by reading the title and abstract of the research.

Below some examples of inclusion and exclusion criteria which have been applied in published SMAPS research [17].

1. “Publications which focus on motivation factors or de-motivation factors” are inclusion criteria.
2. “Studies in other domains of knowledge, for example, electrical engineering projects” are exclusion criteria.

3.2.6 Quality Assessment

Gauging whether research is relevant cannot be done by analyzing the title and abstract only. To indicate the research has high quality, some quality assessment criteria should be applied. These outcomes of quality assessment will recommend whether the research is useful or not. This is an example of quality assessment criteria: “Are the findings and results clearly stated in the paper?” [17].

3.2.7 Extraction Form

The last part of protocol design is defining the extraction form. This will be utilized in the data extraction process. The fields of the extraction form are the requested information in order to answer research questions. Examples of extraction form fields include the publication year, the channel sources, the type of contribution, and the research type.

3.2.8 Collecting Relevant Studies

The activity of collecting relevant studies is performed after the protocol has been completed. As described in the research digital libraries section, the challenge of this process is how to tailor the search string to be accepted syntax for each research digital library. Well-developed search syntax will produce more accurate results. The researcher needs to attempt several types of syntax and select the appropriate one.

3.2.9 Data Extraction Process

The process of data extraction is the most time-consuming part. The selected studies will be analyzed by using the data extraction form. Typically, reading a selected study is not enough one-time due to its structure or language. There are two types of possible software that can help people to extract the data: Microsoft Excel or State of the Art through Systematic Review (StArt). In addition, the result must be organized properly to simplify the next process, which is analyzing and presenting the data.

3.2.10 Analyzing and Presenting Result

The last process of the systematic mapping study is analyzing and presenting the results. Various information will be obtained and categorized based on the data extraction fields. However, this process should carry on analysis which is required by the research questions. Then the presented results need to satisfy and answer defined research questions. The recommended technique to present the result of the SMAPS is providing the table and chart.

3.3 Readiness Model Development

Readiness model development is the main process of this thesis. Adapting the readiness model concept from several published studies, readiness model development in this thesis will utilize the output of the systematic mapping study. It will determine which information is used as a list of components. Every component of the readiness model will have some practices which are integrated with the Motorola assessment tool. The development of the readiness model is not straightforward because it needs iterative reviews.

3.4 Case Study

A case study has the ability to gain more information based on real-world perspectives. In other words, a case study is beneficial to investigate unknown information. Therefore, determining the topic is important to optimize the results. According to Tellis [56], a case study is utilized to compare the voice of author with a

selected group. In addition, a case study can answer the issue of generalization whether the contribution of research is applicable in the real world or not.

Yin [57] suggested that to properly conduct a case study the practitioners should have several capabilities such as the skill when proposing a question and the skill when interpreting the response. Also, the practitioners have to be good listeners and flexible when unpredictable situations arise.

Basically, some organizations or practitioners will be invited to attempt the offered readiness model. There is a qualification criterion to determine whether the respondent is appropriate or not. Technically, a case study can be implemented by meeting face to face or through an online form. One case study challenge is to ensure the respondent understands what our research is talking about and how to utilize the assessment tool properly.

A case study is considered in this research for a number of reasons:

1. Demonstrate that the readiness model can be adapted to real software development.
2. Spotlight the part where the readiness model requires improvement.
3. Demonstrate the benefit of applying the readiness model.

The expected outcome after applying a case study is that the weaknesses of the readiness model will be identified. This can be used for improving the next readiness model.

CHAPTER 4

SYSTEMATIC MAPPING STUDY

This chapter explains the conducted systematic mapping study (SMAPS) as a method to obtain comprehensive information about SRE. The following steps have been described in Chapter 3. A summary will be provided at the end of this chapter.

4.1 Research Questions

Before developing design review protocol, we built five research questions to initiate the development of SMAPS protocol. Research questions along with their motivation are listed in Table 4.1. It directed the analysis process of the SMAPS. It was also utilized to limit the scope of the anticipated outcome. The outcome of the SMAPS was analyzed to answer these research questions.

Table 4.1 Research Questions for Systematic Mapping Study

No.	Research Question	Motivation
RQ1	What approaches, techniques and tools are available for SRE?	Identifying the existing solution which aims to implement SRE.
RQ2	What is the limitation of identified SR approaches, technique, and tools?	Identifying the weaknesses for existing solutions due to SRE.
RQ3	Which researchers have produced most of the publications in the SRE field?	Identifying the most active researcher in the SRE field.
RQ4	Which database contains large number of publication in SRE field?	Identify the high interest research database for SRE.
RQ5	What SR categorizations are available?	Identify the security requirements categories in SRE publications.

4.2 Review Protocol

This section explains the rule of systematic mapping study implementation. Some important points in review protocol are determining the research sources, defining the selection criteria, developing the search string, collecting studies, assessing the collected studies by quality assessment, and extracting the required data. Every point must be undertaken carefully to obtain an appropriate result.

4.2.1 Determining Research Sources

The selected database sources are IEEE Xplore, ACM, Springer Link, Wiley Online Library, and Science Direct. These research sources were selected because they provide a large number of software development research, especially security requirements engineering. They also provide an advanced searching tool which is suitable for a systematic mapping study. The addresses of each research source are listed below.

Table 4.2 List of Research Sources

Research sources	URL of advance search
IEEE Xplore	http://ieeexplore.ieee.org/search/advsearch.jsp
ACM	https://dl.acm.org/advsearch.cfm
Springer Link	https://link.springer.com/advanced-search
Wiley Online	http://onlinelibrary.wiley.com/mrw/advanced/search
Science Direct	http://www.sciencedirect.com/science/search

4.2.2 Defining Selection Criteria

Selection criteria were utilized to determine whether the collected studies from research sources could be selected or not. It contains inclusion and exclusion criteria. A study was selected when satisfying all the inclusion criteria as shown in Table 3. When the study was detected as having exclusion criteria in Table 4, it was then rejected. The purpose of selection criteria was to ensure the studies are relevant to the research objectives and have appropriate qualifications. This research adapted the inclusion and exclusion criteria from published research of systematic mapping studies [17].

Table 4.3 Inclusion Criteria

No.	Inclusion Criteria
1.	Related to secure software engineering domain.
2.	Discussing secure requirement engineering evidence.
3.	Published after 1980 since the Internet appears after that year.

Table 4.4 Exclusion Criteria

No.	Exclusion Criteria
1.	The language is other than English
2.	Papers without sufficient bibliographic information.
3.	Not peer-reviewed publications.
4.	A different domain of knowledge.
5.	Duplicate publication. A complete version will be selected.
6.	Technical reports, white papers, master thesis, Ph.D. dissertation, and textbooks are eliminated
7.	Not relevant to the defined research questions

4.2.3 Developing Search Strategy

This section describes how a search string was applied to the research sources. Three steps will be explained. In the first step, we built the search terms by defining the population, the intervention, the outcome of relevance, and the experimental design that is suitable for our research.

- Population: secure requirements engineering in software development
- Intervention: available technique, model, approaches to satisfy secure requirements engineering
- The outcome of relevance: secure requirement engineering technique, SRE model, SRE approaches.
- Experimental design: case study, empirical studies, theoretical studies.

Based on the results above, the search string was configured by using some keywords such as “secure”, “security”, “requirement”, “software”, “engineering”, and “approach”.

In the second step, we looked for the synonyms of the obtained keywords to enhance the quality of the search string. We performed this step due to studies often utilizing different words with the same meaning.

- Secure Requirements Engineering: “Security Requirements” OR “Securing Requirements” OR “Secured Requirements Engineering”
- Approaches:” guideline” OR “technique” OR “technology” OR “tool” OR “model” OR “framework” OR “approach”

The word “secure” has a similar meaning with “security” and “secured” in terms of requirements, whereas the word “approaches” contains many potential meanings, such as “technique”, “guideline”, “model”, “tool”, and “framework”.

In the final step, after identifying the synonym of each keyword, we then described a general search string that has been applied in research sources. The full search string is defined below.

Software AND requirement AND (secure OR security OR securing OR secured) AND (technique OR method OR technology OR tool OR model OR diagram OR approach OR framework OR guideline)

This search string was tailored to correspond to each research source due to different mechanisms. If the accuracy of the search string was low, then the number of studies collected was too large. Thereafter, it required greater effort to identify the relevant studies. Details of the tailored search strings are listed in Table 5.

Table 4.5 Tailored Search String Based on Searching Rule in the Research Sources

Sources	Search String
IEEE	TITLE-ABSTR KEY (("Secure Requirement" OR "Security Requirement" OR "Trust Requirement")) and TITLE_ABSTR-KEY (("approach" OR "method" OR "technique" OR "technology" OR "model" OR "diagram" OR "framework" OR "guideline"))
ACM	("Security Requirement" +OR + "Secure +Requirement") +AND + ("method" +OR + "technique" +OR + "technology" +OR + "model" +OR + "diagram" +OR + "framework" +OR + "guideline" +OR + "approach")
Science Direct	("Secure Requirement" OR "Security Requirement") AND ("approach" OR "method" OR "technique" OR "technology" OR "model" OR "diagram" OR "framework" OR "guideline")
Springer	"*secur* requirement*" "*trust* requirement"
Wiley	("secure requirement" OR "security requirement" OR "secured requirement" OR "securing requirement") in All Fields AND ("approach" OR "technique" OR "technology" OR "method" OR "diagram" OR "framework" OR "guideline") in All Fields

4.2.4 Collecting Relevant Studies

After applying the search string into research sources, a list of potential studies was generated. It was important to ensure that the result obtained is acceptable. We applied the search string into a research source several times due to the low accuracy of the results. The number of studies generated was more than a thousand, and most of them were not related to research domain. The recommended solution is to improve the search string syntax and ensure it follows the rules of each research source.

There are two steps to identify the study, which will be considered as selected studies:

- Reading the title, keyword, and abstract.
- Reading the whole publication.

Reading the title, keyword, and abstract were performed to determine whether it is in the domain of our research or not. Inclusion and exclusion criteria were also utilized in this step. Duplicate articles and those not relevant to the research topic were eliminated. Reading all the content of the collected studies is required when the title and abstract need more description.

4.2.5 Quality Assessment Criteria

Quality assessment criteria was utilized for measuring the selected studies based on the quality of content. Indeed, quality assessment of studies is applied in a systematic literature review, but it is not mandatory in a systematic mapping study. The purpose of the quality assessment process in a SMAPS is only to support the selection criteria. Note

that the SMAPS aims to discover more relevant studies and to generate some essential categorizations of information rather than focus on the specific issue.

We adapted the quality assessment criteria proposed by Nabil et al. [17]. A study which obtains a score lower than 4 is then rejected from the selected studies. The detail of quality assessment criteria is listed in Table 4.6.

Table 4.6 Quality Assessment Criteria

Criteria	Notes
Are the purposes of the research clearly described?	Yes = 1, No = 0
Are the findings or results clearly stated in the paper?	Yes = 1, No = 0
Does the research create or add contribution to the academia or industry?	Yes = 1, No = 0
Is the proposed technique clearly explained?	Yes = 1, No = 0
Is the paper well referenced (i.e. article references from various journals and peers reviewed conferences)?	Yes = 1, Partially = 0.5, No = 0

4.2.6 Data Extraction

In the beginning, we utilized a systematic literature review tool, namely StArt (State of the Art through Systematic Review). It has the capability to manage the required steps in a systematic mapping study, starting from the defining protocol step until the extracting data step. However, in our case, this tool has an error when managing hundreds of studies. Finally, for managing the extracting data process we continued by using Microsoft Excel software.

In essence, we collected the research information such as the title, publication year, authors, and the publisher. There are other classifications which were used for extracting data from primary studies as listed below.

- Paper channel (journal, workshop, conference, symposium)
- Empirical type (case study, experiment, survey, other)
- Approach (framework, method, model, tool, guideline)
- Requirement activities (elicitation, analysis, specification, verification, management)
- Security requirement type (identification security requirement, authentication security requirement, authorization security requirement, etc.)
- SRE techniques (Misuse case, Problem Frames, CLASP, etc.)

The outcome of the data extraction process was utilized to answer defined research questions.

4.3 Findings

This section presents some categorization of data from the extracting process. We present the distribution of primary studies based on research sources, publication channel type, and year. In addition, there are some results describing security requirements engineering, such as security requirements categories, security requirements techniques, and the list of active researchers in security requirements engineering topic. It's detail information were provided in the tables and the figures.

First, the distribution of primary studies based on research sources was listed in Table 4.7 and depicted in Figure 4.1. The most relevant publications for SRE are IEEE Xplore, followed by Science Direct, ACM, Springer Link and Wiley Online. Comparing the results of the searching process, the initial selection and the final selection, there is a higher concentration in the ACM and Science Direct libraries. For example, in the ACM

library, the collected studies numbered 371. After performing the selection criteria, it was reduced to be 82. Thereafter, based on relevancy and quality assessment criteria, it was further reduced to be 14. The reason for this might be due to the search string quality or the accuracy of the searching algorithm inside each research digital library.

Table 4.7 Distribution of Primary Studies Based on Research Sources

Research sources	Total result	Initial Selection	Final Selection
IEEE Xplore	146	92	65
Science Direct	315	35	19
Springer Link	72	18	2
ACM	371	82	14
Wiley Online	20	7	4
Total	924	234	104

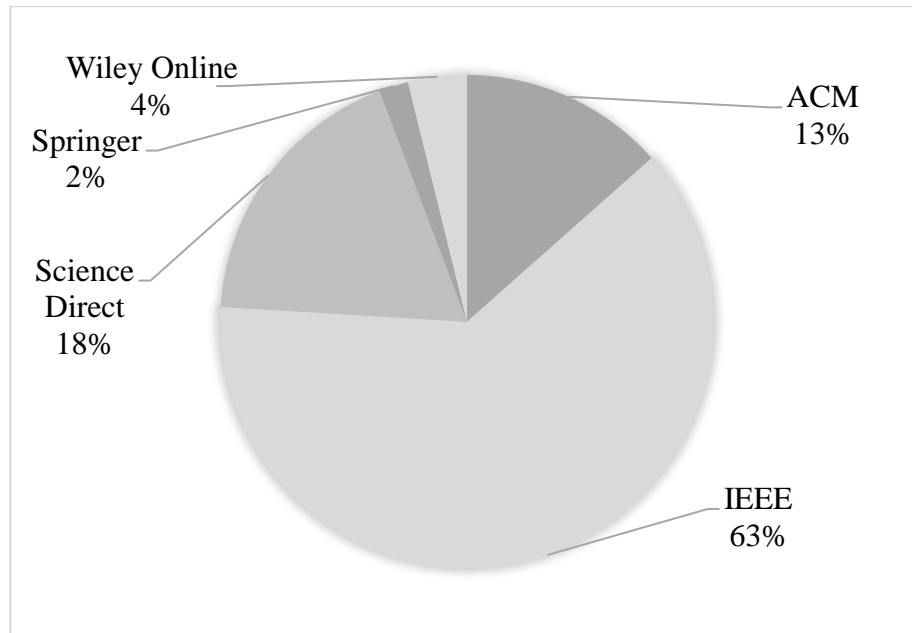


Figure 4.1 Research Sources of Selected Studies

Table 4.8 Distribution of Primary Studies Based on Publication Channel

Publication Channel	Amount	%
Journal	31	29.82
Conference	61	58.65
Workshop	9	8.65
Symposium	3	2.88
Total primary studies	104	100

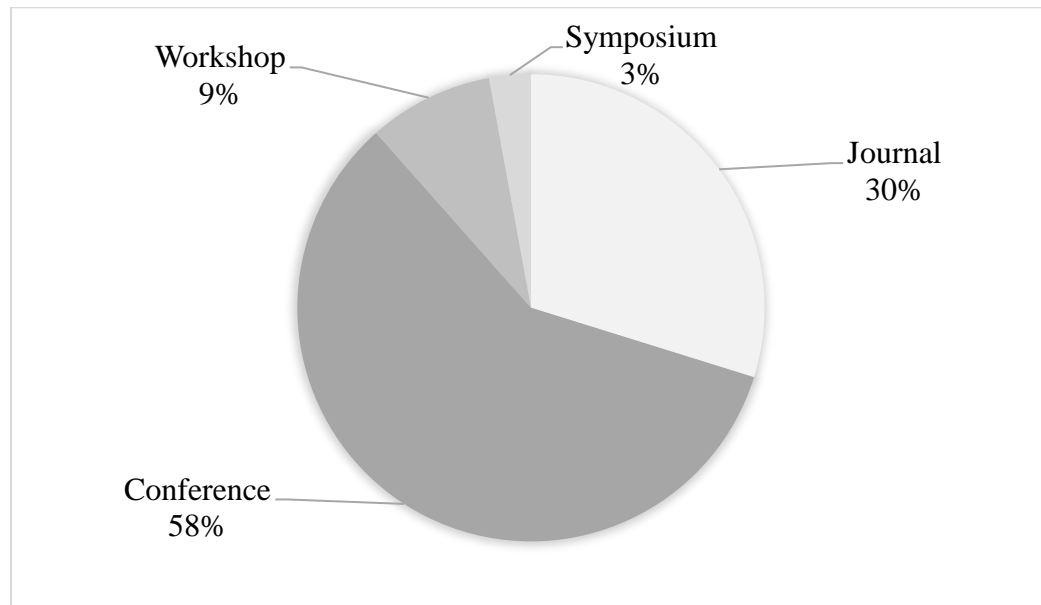
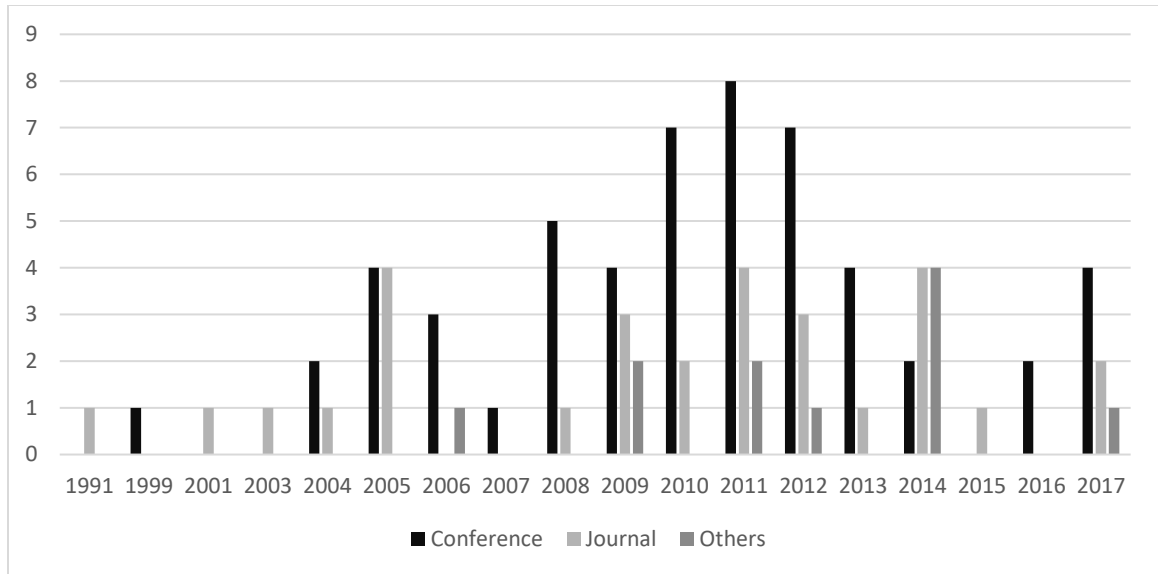


Figure 4.2 Selected Studies Based on Publication Channel

Second, we present the primary studies based on the publication channel. As shown in Table 4.8 and Figure 4.2, most of the primary studies have been published in conferences, followed by journals, workshops and symposiums. It indicates that researchers who are interested in SRE have an opportunity to publish in the conference or journal.



[Figure 4.3 Selected Studies Based on Publication Year]

Figure 4.3 shows the selected studies based on publication year. SRE research was begun in 1991. The most active studies were published 10 years later, in 2011. In 2017, we found seven relevant papers to the SRE studies. As a result, it indicates the SRE field is still interesting for research.

After extracting the data from each selected study, some hidden information was obtained. For instance, this research discovers various research types and empirical result categories utilized in SRE research. We utilized the classification of research types that available in a research by Ouhbi et.al.[58].

- Evaluation Research: This research will evaluate or investigate the conducted approaches. The problems in SRE also were identified in this research.
- Solution Proposal: This research proposed a solution to SRE problems. This solution may be novel or a significant extension of a published approach. The potential advantages and the applicability of the solution are indicated with a small example or a good argumentation.

- Experience Paper: This paper should show the author's experience and describe what has been done and how it was conducted in practice.
- Other: e.g., Theoretical papers, opinion papers, reviews.

Ouhbi et.al [58] also utilized the classification empirical research type as follows to represent their systematic mapping study outcomes.

- Case study: An empirical inquiry to investigate the impact of approach within real-life situations.
- Survey: A method for collecting information from selected respondent to gain quantitative data.
- Experiment: An empirical method applied under controlled conditions.

As depicted in Figure 4.4, most SRE publications use the solution paper as their research type and utilize the case study technique for providing empirical evidence. Thereafter, the information in Figure 4.4 could be essential for other researchers to determine the suitable research type and the empirical result category for their research in future.

In addition, this research recognizes the most active researchers in SRE publications as listed in Table 4.9. Generally, they worked in a collaboration for several papers. For example, Eduardo, Mario Piattini, and Daniel Melado have been involved in same publication. If we are interested on SRE topic, we may follow or communicate with them to obtain an update about this topic. Moreover, we also may invite them to supervise or criticize our SRE research.

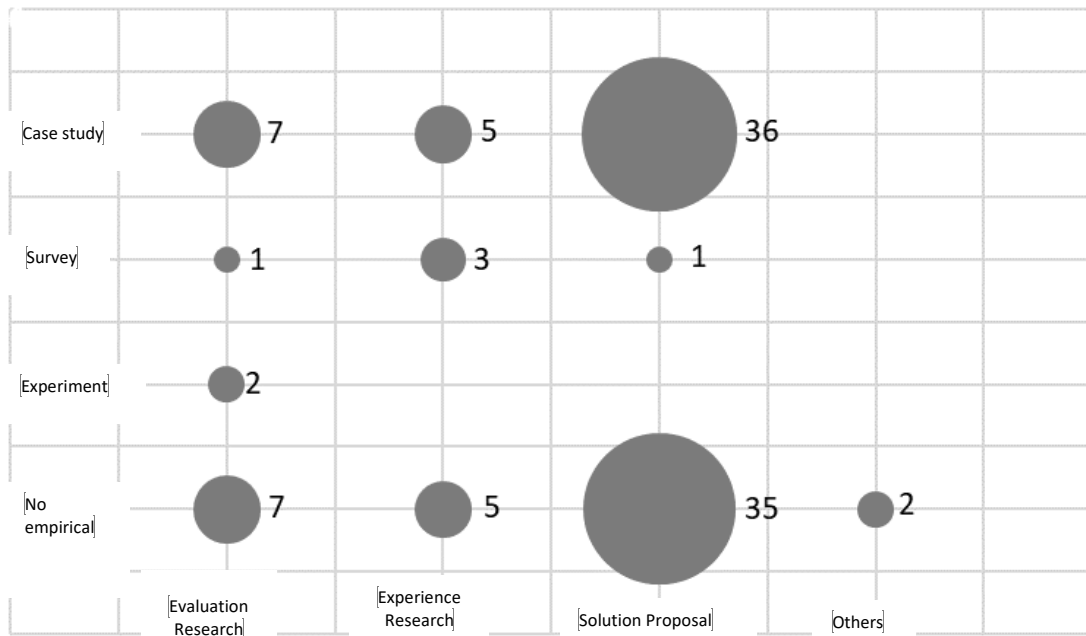


Figure 4.4 Research Type and Empirical Result Type

Table 4.9 List of Active Authors in SRE Research

Author	# Papers
Eduardo Fernández-Medina	10
Mario Piattini	10
Daniel Mellado	7
S. Kanmani	6
P. Salini	6
Giorgini, Paolo	4
Dalpiaz, Fabiano	3
Massacci, Fabio	3
Opdahl, Andreas L.	3
Paja, Elda	3
Sindre, Guttorm	3
Yoshioka, Nobukazu	3

Table 4.10 Security Requirements Categories

Security Requirement Category	Number of Studies (n=104)	%
Identification security requirements	5	4.81
Authentication security requirements	30	28.85
Authorization security requirements	35	33.65
Immunity security requirements	3	2.88
Privacy security requirements	20	19.23
Integrity security requirements	27	25.96
Physical protection security requirements	8	7.69
Non-repudiation security requirements	14	13.46
Intrusion detection security requirements	8	7.69
System maintenance security requirements	8	7.69
Secure auditing security requirements	10	9.62
Survivability security requirements	15	14.42
Not specific	40	38.46

We have collected various SR categories from selected studies as listed in Table 4.10. Not every selected study in our SMAPS discusses the category of security requirements in detail. Authorization security requirements is the most discussed by primary studies, which occurs in 35 publications, followed by authentication, integrity and privacy security requirements.

There are some publications which described various SR categories in SRE, such as research by Al-Shorafat [59], Zafar et al. [60], and Felderer et al. [61]. However, due to the usability and popularity, this study selected the SR categories from research by Firesmith [62], which is further emphasized by P. Salini and S. Kanmani [15].

Table 4.11 Security Requirements Engineering Techniques

Security Requirements Engineering Technique	Number of Studies (n=104)	%
Misuse case	23	22.12
UML	8	8.25
Common criteria	8	7.69
Attack tree	7	6.73
I* framework	6	5.77
Secure Tropos	6	5.77
Problem frame	5	4.81
UMLSec	5	4.81
SREP	5	4.81
SQUARE	4	3.85
ISO/IEC 270001	4	3.85
Security use case	4	3.85
CLASP	2	1.92
MOSRE	2	1.92

There are various techniques applied in SRE research. The common techniques in SRE research are listed in Table 4.11. The misuse case is the most utilized in SRE research, followed by UML, common criteria, attack tree, I* framework and Secure Tropos. Some studies used the misuse case as a technique that collaborated with other techniques, such as research by P. Salini and S. Kanmani [63]. Whereas Sindre and Opdhal [41] utilized the misuse case as a single technique.

Table 4.12 Security Requirements Engineering Activities

Security Requirement Engineering Activity	# Research (n=104)	%
Elicitation	44	42.31
Negotiation/ Analysis	29	41.35
Documentation	43	27.88
Verification	9	8.65
Management/ Change Management	4	3.85
Not specific	12	11.54

Based on Table 4.12, most of selected studies discuss SRE in elicitation and analysis activity. Some publications have concern specifically in one activity, such as research by Shaman and Ivan [64], which focused on elicitation, and research by Nauman et al. [60], which focused on analysis activity. The research has more than one SRE activity for a number of reasons, such as the need to develop a complete framework [10], there is a need for obtaining a comprehensive result [65], and the linkage between activities to execute the technique completely [66].

The terms of negotiation, analysis, and modeling are classified within one activity. These terms existed in different publications with same purpose, which is investigating the elicited SR. Similarly, the terms documentation and specification are activities in selected studies that have the same objective. Few studies discuss research in management activities. The reason for this is because of the high effort required. For example, a researcher should be involved in practical software project development in the industry to monitor changes in security requirements in the set period of time.

4.4 Answering Research Questions

RQ1. What approaches, techniques and tools are available for SRE?

Table 4.11 presents some approaches which are popular in SRE research. Misuse cases are the most utilized by researchers. From 104 selected studies, 22.12 % conducted research by using the misuse case. This is followed by 7.69 % studies using UML, 7.69 % studies which discussed common criteria, 6.73 % studies which performed attack tree, 5.77 % studies which utilized I* framework, and 5.77 % studies which are interested in Secure Tropos.

The misuse cases is considered a technique which is an extension of the use case and has the capability to identify threats [67]. It offers the misuse case as a negation of the use case to recognize the behavior of a misuser. A misuser is actor whose behavior is harmful to the system. By utilizing the misuse case, various mitigations can be developed as security requirements to prevent a potential threat to the system. This is the reason that misuse cases are popular in SRE research, especially in elicitation activity.

Common criteria (CC) provides updated security mechanisms which are utilized by some selected studies. First, CC is utilized as complementary, as suggested by research by Mellado et al. [68], which integrated CC with ISO/IEC 17799. Second, CC is utilized as a main technique in SRE research, such as a study by Ware et al [69] to elicit security requirements.

Several tools are offered by several selected studies, such as SREPPLine tool [49] for product line topic, SSC4Cloud tool [70] for cloud environment, and ST-Tool [71] for data and privacy assessment in web technology. Although UML has a capability for visualizing, it cannot be considered as a tool. Instead it is considered as a technique.

RQ2. What are the limitations of the identified security requirements approaches, techniques, and tools?

Based on analyzing the selected studies, we obtained some limitations of presented techniques which are described by researchers in order to propose new techniques. The misuse case as a popular technique in SR elicitation has some weaknesses. First, misuse cases are considered not comprehensive for identifying threats and the malicious actor. Second, the outcome of misuse cases is difficult to be validated. For large and complex systems having more threats, misuse case diagram is not recommended as it will be a challenge to read so many cross edges in the diagram [72]. Although the misuse case diagram shows attacks or threats, it does not have the ability to explain in detail on how these attacks can be detected.

The attack tree is other popular technique in SRE research. Despite the attack trees have some benefits over the misuse cases, it does not offer information about preconditions and mitigation policies corresponding to the threats [72]. Therefore, it is recommended to overlap the two techniques to meet the aforementioned deficiencies.

RQ3. Which researchers have the most publications in the SRE field?

Based on Table 4.9, the most active researchers are Eduardo Fernández-Medina and Mario Piattini. They collaborated with Daniel Mellado to focus on improvement security requirements engineering in the software product line [49], [73]. P. Salini, and S. Kanmani also worked together in SRE in order to provide the MOSRE framework to be applied in several projects [10], [74]. Unfortunately, Eduardo et al. published their latest SRE research in 2014, while P. Salini and S. Kanmani published their SRE research in 2012.

RQ4. Which database contains an appropriate number of publications in the SRE field?

Database sources of research are listed in Table 4.7 and indicate that we can get relevant publications of SRE in IEEE Xplore, followed by Science Direct and ACM. Most published studies in these database sources are categorized as conference papers.

RQ5. What SR categorization are available?

According to the Table 4.10, there are various SR categories as listed below.

- Identification security requirement
- Authentication security requirement
- Authorization security requirement
- Immunity Security Requirement
- Privacy Security Requirement
- Integrity Security Requirement
- Physical Protection Security Requirement
- Non-repudiation Security Requirement
- Intrusion Detection Security Requirement
- System Maintenance Security Requirement
- Secure Auditing Security Requirement
- Survivability Security Requirement

CHAPTER 5

READINESS MODEL

5.1 Introduction

This chapter summarizes the development process of a security requirements engineering readiness model (SRERM). Before applying this SRERM in the real software industry, we need to determine the structure of the proposed readiness model, the proper measurement instruments, and the suitable assessment tool. In addition, a feedback evaluation form was developed to gain comments from respondents. Internal reviews and iterative changes were performed before external evaluation in the real software industry. The flow process of SRERM development is depicted in Figure 5.1.

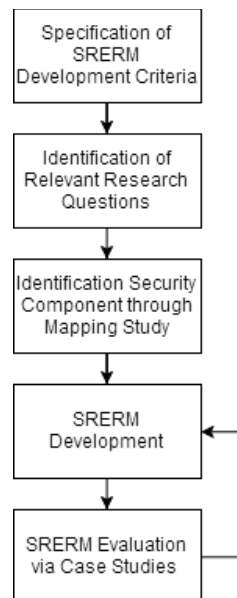


Figure 5.1 SRERM Development

5.2 Structure of SRERM

This section explains SRERM in terms of structure. SRERM is purposed to assist organizations in quantifying their readiness corresponding to the SRE activities. SRERM structure is motivated by the software improvement process readiness model (SPIRM) [20], the software outsourcing vendor readiness model (SOVRM) [75], and the software outsourcing partnership model (SOPM) [38] concept. Some parts of these models are utilized in this research such as the measurement level concept, the assessment tool and collecting feedback from respondents. The difference with SRERM is the content of the levels. While the above models utilized critical success factors (CSFs), this research instead uses security requirements categories (SRCs).

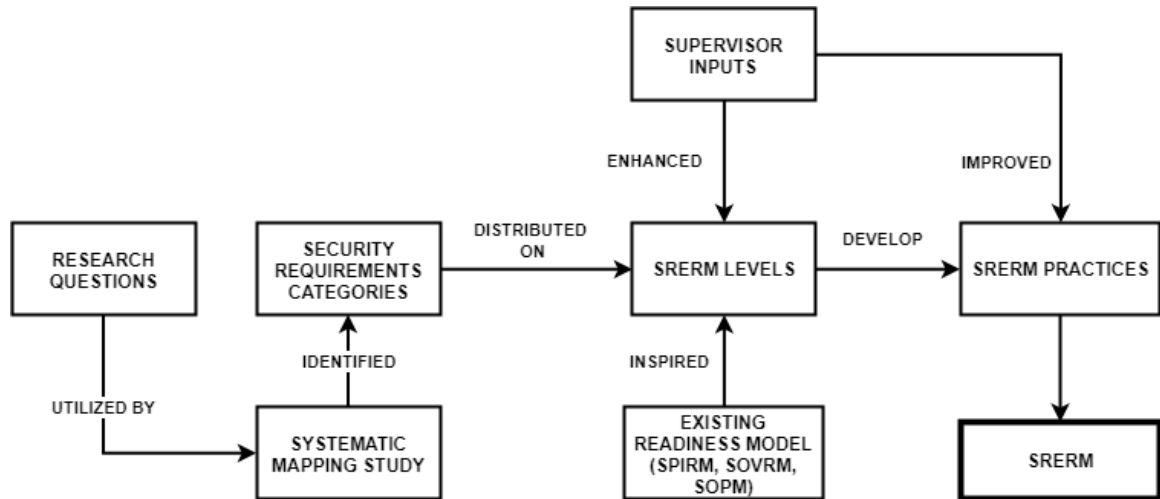


Figure 5.2 The Structure of SRERM

Figure 5.2 shows the flow process of SRERM development. The results of the SMAPS, which are security requirements categories, are utilized to construct the SRERM structure. The preliminary SRERM structure has three dimensions: levels, security components (SCs), and SRE practices.

5.2.1 Preliminary Levels of SRERM

The levels of SRERM are purposed to represent the readiness achievement degree of software organizations in undertaking SRE. Following are the five preliminary levels of the SRERM for software development organizations.

1. **Initial:** This readiness level can be recognized as having a confused status. At this level, the organization does not provide any preparation for security requirements engineering.
2. **Basic:** This readiness level indicates the concern on developing basic security requirements for software development. At this level, organizations realize security requirements is mandatory in software development.
3. **Protected:** This level analyzes security requirements related to the information and assets.
4. **Anticipated:** At this level prevention and greater awareness are emphasized.
5. **Monitored:** This is the highest readiness level. At this level, organizations have a high focus on maintaining security requirements built at the previous level.

The levels classification was developed based on the main purpose of each security requirements and the prerequisite. For instance, the basic level shall contain the components required as a prerequisite to support the components at the protected level.

Overall readiness levels in the SRERM require evaluation and feedback to be sufficient for analyzing the SRE readiness in the organizations. When a conflict among security requirements components is found, or some suggestions relating to the representation of the readiness model are received, correction and improvement should then be rapidly undertaken. Figure 5.3 represents how SRERM levels recognize the

organization's performance and how the discoveries from the SMAPS are distributed into the SRERM levels.

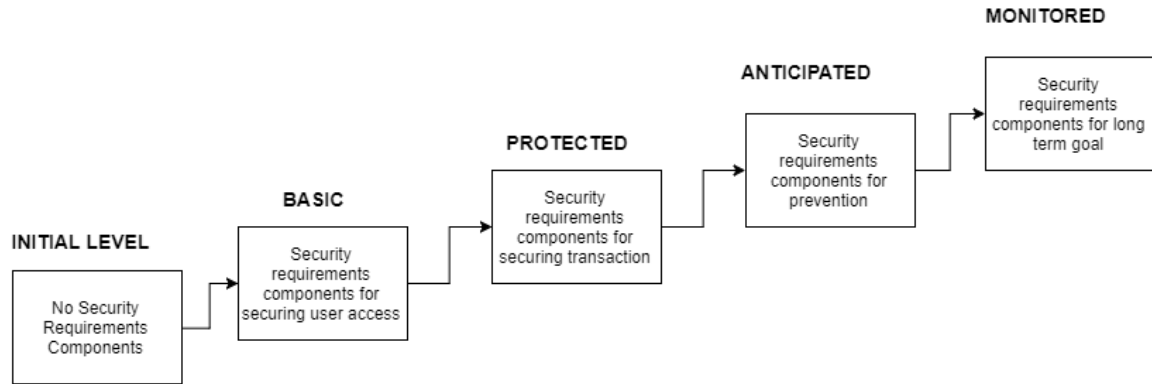


Figure 5.3 The Preliminary SRERM Levels

5.2.2 Components of SRERM

The SPI readiness model [20] was used to distribute the critical success factors and the barriers of software process improvement to each level. The SOPM [38] utilized the critical success factor of the outsourcing relationship. This research follows these concepts in order to develop the levels of the SRERM which distribute the security requirement components. The security requirements components in our research referred to the security requirements categories, which were collected through the SMAPS. Some SRE practices were also developed based on outcomes of the SMAPS and RE activities. The detail of the SRE practices are available in the appendix B.

Twelve security requirements (SR) categories, identified via mapping study (RQ5), were distributed into five preliminary readiness levels as depicted in Table 5.1. The distribution of these security categories was based on the prioritization which was obtained from the SMAPS. Each level contains some security requirements categories

except the initial one. A focus column is added in Table 5.1 to describe the motivation or situation of each readiness level.

Identification SR, authentication SR, and authorization SR were distributed in the basic level because they are mandatory for each system. Immunity SR, privacy SR, and integrity SR were placed in the protected level because they are suitable for protecting important assets. Consideration for physical protection SR, non-repudiation SR, and intrusion SR in the anticipated level because it will require high effort. System maintenance SR, secure auditing SR, and survivability SR were also placed in the monitored level because the cost of these SRs is very high and commonly purposed for the sustainability of the organizations' long-term goals.

Table 5.1 Detail information of Preliminary SRERM Levels

No	SRERM Level	Focus	Security Requirement Categories
1	Initial	The situation without security requirement component included	Nil
2	Basic	Securing user's access	Identification Security Req.
			Authentication Security Req.
			Authorization Security Req.
3	Protected	Securing transactions	Immunity Security Req.
			Privacy Security Req.
			Integrity Security Req.
4	Anticipated	Prevention and high-quality security	Physical Protection Security Req.
			Non-repudiation Security Req.
			Intrusion Detection Security Req.
5	Monitored	Security for long term goals	System Maintenance Security Req.
			Secure Auditing Security Req.
			Survivability Security Req.

5.3 Assessment Tool

The Motorola assessment tool [22] is the measurement tool used in the SRERM. As shown in Table 5.2, it is utilized to assess the practices for each security requirements component. This tool has been used in SPIRM [75], SOVRM [38] and SOPM [38]. The Motorola assessment tool requires three assessment aspects [22]:

- **Approach:** This aspect focuses on the support of management and the commitment of the organization relating to the practice.
- **Deployment:** This aspect focuses on the comprehensiveness and consistency of the practice deployment.
- **Results:** This aspect focuses on the positive results in term of the effect scale in the project.

Table 5.2 Motorola Assessment Tool

Score	Approach (A)	Deployment (D)	Results (R)
Poor (0)	<ul style="list-style-type: none">• No management recognition of need (OR)• No organizational ability (OR)• No organizational commitment (OR)• Practice not evident	<ul style="list-style-type: none">• No part of the organization uses the practice (OR)• No part of the organization shows interest	<ul style="list-style-type: none">• Ineffective
Weak (2)	<ul style="list-style-type: none">• Management begins to recognize need (OR)• Support items for the practice start to	<ul style="list-style-type: none">• Fragmented use (OR)• Inconsistent use (OR)• Deployed in	<ul style="list-style-type: none">• Spotty result (OR)• Inconsistent result (OR)• Some evidence

	be created (OR) <ul style="list-style-type: none"> • A few parts of organization are able to implement the practice 	some parts of the organization (OR) <ul style="list-style-type: none"> • Limited to monitoring/verification of use 	of effectiveness for some parts of the organization
Fair (4)	<ul style="list-style-type: none"> • Wide but not complete commitment by management (OR) • Road map for practice implementation defined (OR) • Several supporting items for the practice in place 	<ul style="list-style-type: none"> • Less fragmented use (OR) • Some consistency in use (OR) • Deployed in some major parts of the organization (OR) • Monitoring/verification of use for several parts of the organization 	<ul style="list-style-type: none"> • Reliable and positive results for several parts of the organization (OR) • Inconsistent result for several parts of the organization
Marginally qualified (6)	<ul style="list-style-type: none"> • Some management commitment (OR) • Some management becomes proactive (OR) • Practice implementation well underway across parts of the organization (OR) • Supporting items in 	<ul style="list-style-type: none"> • Deployed in some parts of the organization (OR) • Mostly consistent use across many parts of the organization (OR) • Monitoring/verification of use for 	<ul style="list-style-type: none"> • Positive measurable results in most parts of the organization (OR) • Consistently positive results overtime across many parts of the organization

	place	many parts of the organization	
Qualified (8)	<ul style="list-style-type: none"> • Total management commitment (OR) • Majority of management is proactive (OR) • Practice established as an integral part of the process (OR) • Supporting items encourage and facilitate the use of practice 	<ul style="list-style-type: none"> • Deployed in almost all parts of the organization (OR) • Consistent use across almost all parts of the organization (OR) • Monitoring/verification of use for almost all parts of the organization 	<ul style="list-style-type: none"> • Positive measurable results in almost all parts of the organization (OR) • Consistently positive results over time across almost all parts of the organization
Outstanding (10)	<ul style="list-style-type: none"> • Management provide enthusiastic leadership commitment (OR) • Organizational excellence in the practice recognized even outside the organization 	<ul style="list-style-type: none"> • Universal and constant deployed in all parts of the organization (OR) • Consistent use over time across all parts of the organization (OR) • Monitoring/verification for all parts of the organization 	<ul style="list-style-type: none"> • Requirement exceeded (OR) • Consistently world-class results (OR) • Guidance sought by others

For each aspect, we select a value (0, 2, 4, 6, 8, and 10) which can be determined referring to the criteria provided in Table 5.2. Here, we explain how to utilize the Motorola assessment tool by assuming the scores have been computed as shown in the Table 5.3.

- First, for each practice calculate the total score of three aspects (approach, deployment, and result), then divide the total by three to find the average and round to a whole number.
- Second, repeat the first step to overall practice in one security component.
- Third, sum the average of every practice and divide by the number of practices for each security component.

Fourth, repeat the third step and find the average for each level. If the level gains average score is less than seven it is regarded as weak, whereas higher than or equal to seven is strong.

Table 5.3 The Example of Security Component Evaluation

No.	Practices	Approach 0,2,4,6,8,10	Deployment 0,2,4,6,8,10	Result 0,2,4,6,8,10	Average
1.	Utilize brainstorming technique to aggregate identification security requirements	10	10	10	10
2.	Identify system stakeholders to improve identification security requirements	8	8	8	8
3.	Plan for conflicts and conflict resolution for	8	8	8	8

	identification security requirements in term of stakeholders				
4.	Define standard templates for describing identification security requirements	8	8	8	8
5.	Use languages simply and concisely to explain identification security requirements	8	8	8	8
6.	Check that identification security requirements meets your standard	8	8	8	8
7.	Define change management policies for identification security requirements	4	4	4	4
Total of average scores (Calculate all scores in Average column)					46
Final score (Total of average scores divided by number of practices) $= 46/7 = 7.7$					8

5.4 Evaluation Process of SRERM

The evaluation step for the SRERM is an important stage to validate and improve the applicability of the SRERM for the real software industry. Two case studies in the software industry were rigorously carried out. The respondents of the evaluation were selected once they were determined to have the capability and experience to answer a question in the SRERM. In the evaluation agreement, it is also stated for privacy and business considerations that their affiliated organizations will not be published. Finally, they completed both the SRERM and the feedback section.

Once the case studies were completed, respondents were requested to complete the questionnaire for assessing the quality of the SRERM. Overall criteria were explained in the evaluation criteria section. The outcomes of the SRERM evaluation were utilized for analyzing the weaknesses. The respondent suggested some changes to aid future improvement of the object.

5.4.1 Evaluation Criteria

Criteria in the feedback section are described below.

1. **Ease of use:** This criterion assesses and evaluates the usability of the SRERM structure. It requires the SRERM structure to have flexibility and be unambiguous because complex models will require a higher effort and training.
2. **Satisfaction of user:** This criterion assesses and evaluates users' satisfaction according to the outcomes of the SRERM. They should have a chance to utilize the SRERM without any misunderstanding or difficulties to achieve the goals related to the SRE domain.
3. **The structure of the SRERM:** This criterion's purpose is to recognize any gaps in the SRERM structure and how to make improvements for these.

The complete detail of the evaluation form of the SRERM is provided in appendix.

5.4.2 Evaluation Analysis

The outcome of the conducted SRERM will be analyzed in terms of the obtained score, feedback information, and suggestions from the respondent. The score of each organization certainly will not be the same due to different criteria. The plan is to conduct the SRERM in a well-established and growing organization. This study will identify the interesting area of SRE for each respondent.

The feedback section will be used as consideration whether the SRERM achieves satisfaction of the user or not. It also will indicate whether there is any suggestion for improving the SRERM. When a curious result is discovered, or an urgent suggestion received, it is likely to cause a modification to the SRERM.

CHAPTER 6

CASE STUDY

6.1 Introduction

A case study has the ability to gain more information based on real world-perspectives. This advantage is suited with our need that the SRERM requires an evaluation from a practitioner in the software industry. We utilized a case study in this research for the following reasons:

- To demonstrate that the SRERM can be adapted to real software development.
- To highlight the areas where the SRERM requires improvement.
- To demonstrate the benefit of applying the SRERM.

To achieve confidence in the evaluation, this research conducted two case studies on two different software development organizations. The selected organizations have a clear software development processes. They also allow the research to be released with their identity disguised.

Initially, we personally communicated to each respondent from the different organizations, introducing the concept of the SRERM and inviting them to participate in our case study. Considering the quality of the respondents' feedback, training and introductory discussion were carried out at the beginning. Although they are unfamiliar with security requirements engineering research, due to their knowledge of security mechanisms they may rapidly learn how to utilize the SRERM.

6.2 Result

We have conducted the case study in two organizations: organization A and organization B. Both organizations are providing software development service for their customers. We selected these organizations by considering the maturity, employee number, customer number, and number of branches. For each organization, we selected the respondent who has strong role in the software development process.

The outcomes of each organization assessment are then collected in Tables 6.1 and 6.2. Each respondent was required to utilize their experience in completed projects to undertake the assessment. Due to quality concerns and independent feedback, the respondent was requested to complete the questionnaire at their place of business. In a short period of time, they submitted the assessment outcomes including the SRERM evaluation form through email. The assessment outcomes and SRERM evaluation were reviewed and utilized to produce an analysis report.

6.2.1 Organization A

Organization A is a well-established software development organization working for customers around the world. They have branches in Asia, Australia, Europe, and America. The number of their employee is around 700 people. They support a number of oil companies by developing services such as real-time monitoring, data analytics, and reporting. The respondent of organization A has 17 years of experience in developing software. Currently, his position is a software development manager. One of his responsibility is ensuring the requirements analysis process is implemented and can satisfy the customer's need.

6.2.2 Assessment Outcomes of Organization A

Organization A obtained a high score at the basic level. This indicated their security awareness is established from the earliest point of the software development lifecycle. At the protected level, they achieved a high score in two SRs (the immunity SR and privacy SR). It shows that they have awareness of the threat of malware and of privacy issues. However, their integrity SR score and the SRs at the monitored level are very low as they are still planning and discussing these concerns. This information was used to evaluate and improve the SRERM.

Table 6.1 Implementation Score for SCs in Organization A

SRERM Levels		Security Component	Organization A	
No.	Level		Score	Status
1.	Initial	Nil		
2.	Basic	Identification Security Requirements	7.5	strong
		Authentication Security Requirements	8.2	strong
		Authorization Security Requirements	8.5	strong
3.	Protected	Immunity Security Requirements	7.6	strong
		Privacy Security Requirements	7.1	strong
		Integrity Security Requirements	0.8	weak
4.	Anticipated	Physical Protection Security Requirements	3.2	weak
		Non-repudiation Security Requirements	2.1	weak
		Intrusion Detection Security Requirements	5.1	weak
5.	Monitored	System Maintenance Security Requirements	0	weak
		Secure Auditing Security Requirements	1	weak
		Survivability Security Requirements	0	weak

6.2.3 Organization B

Organization B is a growing software development organization working for a university. They develop various integrated software such as class a registration system, an e-learning system, a payment system, a graduation system, an attendance system, a network settings system, and a library system. The main core of the organization's service is data center management and software development.

We selected this organization to represent a non-international organization with fewer employees. The number of their employee is less than 100 people. In addition, this organization has only one customer, which is a university. As a result, organization B was expected could add more usability value to the SRERM.

The selected respondent is the senior developer in the organization B. His experience in developing the system is around 5 years. He has a strong role in the software development, especially the requirement analysis process. He has a responsibility to analyze the customer's need and develop the system.

6.2.4 Assessment Outcomes of Organization B

Organization B has been in the initial level because they have not completed the security component at the basic level. For a growing organization, this level achievement shows that their awareness of security requirements engineering has not been started. They need more support and commitment from management to encourage their team to achieve a higher level of SRE readiness.

Table 6.2 Implementation Score for SCs in the Organization B

SRERM Levels		Security Component	Organization B	
No.	Level		Score	Status
1.	Initial	Nil		
2.	Basic	Identification Security Requirements	5.7	weak
		Authentication Security Requirements	8	strong
		Authorization Security Requirements	7.7	strong
3.	Protected	Immunity Security Requirements	2.4	weak
		Privacy Security Requirements	4.4	weak
		Integrity Security Requirements	2.1	weak
4.	Anticipated	Physical Protection Security Requirements	7.3	strong
		Non-repudiation Security Requirements	2.7	weak
		Intrusion Detection Security Requirements	1.3	weak
5.	Monitored	System Maintenance Security Requirements	3.3	weak
		Secure Auditing Security Requirements	3.1	weak
		Survivability Security Requirements	4.7	weak

The result of organization B also indicates there is concern enough for the physical protection SR and the survivability SR. In contrast, they obtained a low score in identification SR. One of the possible reasons because they have only several stakeholders for their systems. As a result, they are able to describe the identification security requirements without completing all the provided practices in the SRERM. These findings were used for improving the SRERM.

6.3 Feedback Summary

Both respondents from organizations A and B completed the feedback forms to evaluate various aspects of the SRERM. As we described in the section of SRERM development, there are three key aspects (ease of use, the satisfaction of the user, and the

structure of the SRERM). It was evaluated by using quantitative measurement. In addition, some questions were provided to collect their reviews, suggestions, or constructive corrections for improving the SRERM.

First, they were asked to evaluate the ease of the learning aspect. Based on Table 6.3, organizations A and B positively agreed that the form of SRERM is clear and easy to learn. However, training is still required to understand how to utilize the SRERM properly. Although they are familiar with the requirements engineering process and security mechanisms, they recently learnt about SRE.

Table 6.3 Ease of Learning Evaluation of Organization A and B

Ease of Learning	Organizations' perception (n=2)							
	Positive			Negative			Neutral	
	SA	A	%	SD	D	%	N	%
SRERM representation is clear	0	2	100%	0	0	0	0	0
A little knowledge of security requirements engineering is required to learn how to use SRERM	0	2	100%	0	0	0	0	0
It is applicable to learn the practices arranged for each security requirements component	0	2	100%	0	0	0	0	0
It is applicable to learn the assessment method	0	2	100%	0	0	0	0	0
It is applicable to utilize the SRERM to measure organizations readiness for security requirements engineering.	0	2	100%	0	0	0	0	0
It is applicable to utilize distribution of security requirements components among various levels, e.g. Identification, Authentication, and	0	2	100%	0	0	0	0	0

Authorization in Basic Level								
Some trainings should be accommodated for the utilization of SRERM	0	2	100%	0	0	0	0	0

Second, they assessed the user satisfaction aspect. As described in the evaluation criteria section, this criterion assesses and evaluates users' satisfaction corresponding to the results of the SRERM. As Table 6.4 shows, both organizations agreed that the SRERM could be useful in other organizations. They were interested to utilize this SRERM in their work if it is available in their organizations. They were satisfied with the capability of the SRERM to recognize the area of their SRE which needs further improvement.

Table 6.4 User Satisfaction Evaluation of Organization A and B

User Satisfaction	Organizations' perception (n=2)							
	Positive			Negative			Neutral	
	SA	A	%	SD	D	%	N	%
SRERM is can be executed to the most organizations	0	2	100%	0	0	0	0	0
Every practice is obvious to learn and clear	0	2	100%	0	0	0	0	0
Utilizing the SRERM would distinguish strong and weak areas in the organizations corresponding to the security requirements engineering	0	2	100%	0	0	0	0	0
Using the SRERM would improve our security requirements engineering	0	2	100%	0	0	0	0	0
When SRERM were accessible for my occupation, I anticipate that I would utilize it later on.	0	2	100%	0	0	0	0	0

I am fulfilled and approved with the readiness issues recognized by SRERM.	0	2	100%	0	0	0	0	0
It is critical to actualizing SRERM as an automated software tool to encourage security requirements engineering in measuring organization's readiness.	0	2	100%	0	0	0	0	0

Third, the structure aspect of the SRERM was evaluated by two organizations as shown in Table 6.5. They agreed the arrangement of the SRERM structure is suited and obvious. The creation of levels and the distribution of security requirement categories have no confusion or ambiguity. Based on their evaluation outcomes, SRERM can be utilized to effectively measure the SRE readiness of software development organizations.

Table 6.5 SRERM Structure Evaluation of Organization A and B

Structure of SRERM	Organizations' perception (n=2)							
	Positive			Negative			Neutral	
	SA	A	%	SD	D	%	N	%
Every component of the SRERM are self-explanatory and require no further clarification to be utilized adequately	0	2	100%	0	0	0	0	0
Every component of the SRERM are feasible and are suited in security requirement engineering process	0	2	100%	0	0	0	0	0
The SRERM can be used effectively to identify security requirements engineering readiness issues with a goal of increasing organizations readiness for security requirement	0	2	100%	0	0	0	0	0

engineering.								
The distribution of security component among various readiness levels (e.g. identification, authentication, and authorization) is valuable	0	2	100%	0	0	0	0	0
Five readiness levels of SRERM are valuable	0	2	100%	0	0	0	0	0

Fourth, we received a suggestion and criticism from organization A only. The respondent of organization A suggested a modification related to the levels of the SRERM. The SRERM is recommended to have four levels instead of five levels. A criticism was made that the document needed improvement because the respondent already utilizes a requirement engineering template from JIRA. In addition, the design of the questionnaire is advised to be improved in future. This feedback was utilized in the next section, which discusses the modification of the SRERM.

Table 6.6 Feedback Results of Organization A and B

Question	Response		
	Organization A	Organization B	
Do you suggest any correction or enhancement to the SRERM?	It would be good if SRERM can have a level of the appliance. i.e. Level 1 covers Initial and Basic state, Level 2 covers additionally Protected and highest level – 3 includes the last two level.	No	Positive

Do you suggest any new components to the SRERM in the future and please provide the reason?	No	No	Very Positive
Opinion corresponding to the assessment method.	Perhaps need to make a clear definition which part is a document is required which one is the only statement. Or the implementation in another format, I.e. When a company doesn't use document but tool instead (i.e. Jira)	No	Positive
Opinion corresponding to the distribution of various security requirements practices	No	No	Very Positive
Opinion corresponding to the SRERM usability with respect to time it takes the respondent to quantify security requirements engineering readiness.	Probably it is better to redesign the format when the parameters are same for each level.	No	Positive
Practices Review	No need of changes	No changes suggested	Very Positive

6.4 Modification of SRERM

Based on the outcomes of two case studies, we then applied some modifications to the SRERM. The modifications were purposed to achieve higher usability of the SRERM in the software industry. When the usability value of the SRERM is high, it will be correspondingly easier to attract more software organizations to utilize the SRERM. The changes were related to moving the position of a security requirements component across levels and merging one level into another.

First, the physical protection security requirements component was moved from the anticipated level (third level) to the protected level (second level). This was motivated by the outcome of organization B that indicated high interest to secure the physical protection SR. They considered the physical properties of the server as important as the security of information. The physical server should be protected from any challenges such as theft, vandalism, fire, and natural disasters.

Second, due to the suggestion by the respondent of organization A, we merged the anticipated level with the monitored level. Therefore, non-repudiation SR and intrusion detection SR were distributed at the monitored level. We considered non-repudiation SR to be at the monitored level because it had the purpose of advancing the quality of software security. In addition, intrusion detection SR was considered to be at the monitored level due to its providing high-quality security.

Finally, these modifications were updated in the SRERM as shown in Figure 6.1 and Table 6.7. The modifications were motivated by the assessment results obtained in the case studies of organizations A and B. The modifications also affected the practices

for each security component. We ensured the modification of the SRERM does not reduce the usability.

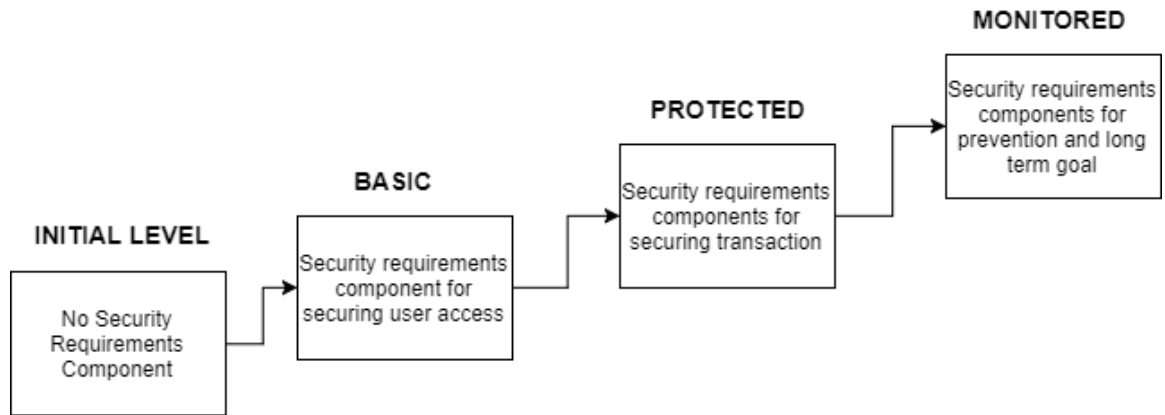


Figure 6.1 Modified Levels of SRERM

Table 6.7 Detail Information of SRERM Modified Levels

No	SRERM Levels	Focus	Security Components
1	Initial	The situation without any SR component included	Nil
2	Basic	Securing user's access	Identification Security Req.
			Authentication Security Req.
			Authorization Security Req.
3	Protected	Securing transactions and another important asset	Immunity Security Req.
			Privacy Security Req.
			Integrity Security Req.
			Physical Protection Security Req.
4	Monitored	Prevention, providing high-quality security, and support long-term goals	Survivability Security Req.
			Intrusion Detection Security Req.
			Non-repudiation Security Req.
			Secure Auditing Security Req.
			System Maintenance Security Req.

Based on Table 6.7, the modified SRERM has four levels. The initial level will indicate the organization has no interest to implement SRE in software development. The basic level will indicate the organization has established the awareness of SRE for the mandatory security components. The protected level will indicate the organization has a high concern to implement SRE by ensuring the security of their data. Lastly, the monitored level will indicate the organizations are motivated to implement SRE by adding advanced services.

6.5 Second Case Study

We conducted second case study on organization A and C. Generally, it was purposed to evaluate the modified SRERM. As previous case study, one of the anticipated advantages conducting case study is assessing its applicability and usability in the real software organizations. The detail explanation of the specific objectives and the outcomes are described in the following sections.

6.5.1 Second Case Study of Organization A

There are two main objectives of second case study on organization A. First, we tried to observe the improvement of SRE in this organization after six months. We would compare the previous case study outcomes with the outcome of second case study. Second, we aimed to check the user satisfaction of organization A to the improved SRERM. Since we incorporated several recommendations from organization A to improve the SRERM, their reviews became essential in the second case study.

6.5.2 Assessment Outcomes of Second Case Study of Organization A

Table 6.8 Implementation Score for SCs in Organization A

SRERM Level		Security Component	Organization A	
No.	Level		Previous Score	New Score
1.	Initial	Nil		
2.	Basic	Identification Security Requirements	7.5	7.5
		Authentication Security Requirements	8.2	8.2
		Authorization Security Requirements	8.5	8.5
3.	Protected	Immunity Security Requirements	7.6	7.6
		Privacy Security Requirements	7.1	7.1
		Integrity Security Requirements	0.8	5.4
		Physical Protection Security Requirements	3.2	4.3
5.	Monitored	Non-repudiation Security Requirements	2.1	4.2
		Intrusion Detection Security Requirements	5.1	6.1
		System Maintenance Security Requirements	0	2.3
		Secure Auditing Security Requirements	1	2.7
		Survivability Security Requirements	0	2.6

Based on Table 6.8, comparing the result of previous case study with the second case study, organization A has improvement in several security requirements categories. For example, the integrity SR is increasing from previously 0.8 to be 5.4. Although their SRE position is still in the basic level, but it indicates there is an improvement in the organization. In addition, various security requirements categories in this organization have been motivated such as physical protection, non-repudiation, intrusion detection, system maintenance, secure auditing, and survivability security requirements categories.

The respondent of organization A recognized and agreed with the modification of SRERM, especially the modified levels. In the feedback section, he did not put any

comments or suggestions to the modified SRERM. However, he agreed that the modified SRERM could satisfy the ease of learning evaluation, user satisfaction evaluation, and structure evaluation.

6.5.3 Second Case Study of Organization C

We conducted a case study in organization C to evaluate the modified SRERM. Organization C has main responsibility to provide IT services for a university in Saudi Arabia. The number of employee is around 160 people. They are developing several education systems such as student registration system, academic portal system, room booking system, and library system.

The selected respondent of organization C is a senior developer. He has experiences in software development more than 5 years. In addition, he has conducted a research in the field of software engineering when he was a graduate student in the university. As a result, he could understand the purpose and the essential of the SRERM.

6.5.4 Assessment Outcomes of Second Case Study of Organization C

Similar to the previous case studies, we provided concisely an introduction of the SRERM in a meeting with the respondent. An online form service was utilized to establish the questionnaire and to collect the respondent's answers. He was requested to complete the questionnaire that contains assessment of SRERM, easy of learning evaluation, user satisfaction evaluation, structure evaluation, and feedback form.

Once respondent completed the questionnaire, the inserted answers were extracted and analyzed based on the defined evaluations. Following information describe the case study outcomes that was completed by respondent of organization C.

Table 6.9 Implementation Score for SCs in Organization C

SRERM Level		Security Component	Organization B	
No.	Level		Score	Status
1.	Initial	Nil		
2.	Basic	Identification Security Requirements	5.1	weak
		Authentication Security Requirements	4.7	weak
		Authorization Security Requirements	5.1	weak
3.	Protected	Immunity Security Requirements	8.0	strong
		Privacy Security Requirements	6.1	weak
		Integrity Security Requirements	5.3	weak
		Physical Protection Security Requirements	7.3	strong
4.	Monitored	Non-repudiation Security Requirements	2.3	weak
		Intrusion Detection Security Requirements	3.0	weak
		System Maintenance Security Requirements	4.0	weak
		Secure Auditing Security Requirements	4.6	weak
		Survivability Security Requirements	4.2	weak

Based on Table 6.9, organization C was at the initial level because they did not achieve standard score for security requirement categories in the basic level. However, they had achieved high score in two security requirements categories: the immunity and physical protection. One reason they had many low scores due to unspecified format of security requirements. They have not provided a clear format and policy how to manage security requirements in the software development process although they have high concern about security requirements. They might improve their readiness level of SRE utilizing the template provided in several security requirements engineering frameworks.

The evaluation results of the SRERM were shown at Table 6.10, Table 6.11, and Table 6.12. The respondent of organization C positively agreed that the form of SRERM is clear and easy to learn. He agreed the arrangement of the SRERM structure, including

the creation of levels and the distribution of security requirement categories, have no confusion or ambiguity. In addition, SRERM can be utilized to effectively measure the SRE readiness of software development organizations. He agreed that the SRERM could be useful in other organizations. He leaves no suggestion or correction to the SRERM.

Table 6.10 Ease of Learning Evaluation of Organization C

Ease of Learning	Organizations' perception (n=1)							
	Positive			Negative			Neutral	
	SA	A	%	SD	D	%	N	%
SRERM representation is clear	0	1	100%	0	0	0	0	0
A little knowledge of security requirements engineering is required to learn how to use SRERM	1	0	100%	0	0	0	0	0
It is applicable to learn the practices arranged for each security requirements component	0	1	100%	0	0	0	0	0
It is applicable to learn the assessment method	0	1	100%	0	0	0	0	0
It is applicable to utilize the SRERM to measure organizations readiness for security requirements engineering.	0	1	100%	0	0	0	0	0
It is applicable to utilize distribution of security requirements components among various levels, e.g. Identification, Authentication, and Authorization in Basic Level	0	1	100%	0	0	0	0	0
Some trainings should be accommodated for the utilization of SRERM	1	0	100%	0	0	0	0	0

Table 6.11 User Satisfaction Evaluation of Organization C

User Satisfaction	Organizations' perception (n=1)							
	Positive			Negative			Neutral	
	SA	A	%	SD	D	%	N	%
SRERM is can be executed to the most organizations	1	0	100%	0	0	0	0	0
Every practice is obvious to learn and clear	0	1	100%	0	0	0	0	0
Utilizing the SRERM would distinguish strong and weak areas in the organizations corresponding to the security requirements engineering	1	0	100%	0	0	0	0	0
Using the SRERM would improve our security requirements engineering	1	0	100%	0	0	0	0	0
When SRERM were accessible for my occupation, I anticipate that I would utilize it later on.	1	0	100%	0	0	0	0	0
I am fulfilled and approved with the readiness issues recognized by SRERM.	1	0	100%	0	0	0	0	0
It is critical to actualizing SRERM as an automated software tool to encourage SRE in measuring organization's readiness.	0	1	100%	0	0	0	0	0

Table 6.12 SRERM Structure Evaluation of Organization C

Structure of SRERM	Organizations' perception (n=1)							
	Positive			Negative			Neutral	
	SA	A	%	SD	D	%	N	%
Every component of the SRERM are self-explanatory and require no further	0	1	100%	0	0	0	0	0

clarification to be utilized adequately								
Every component of the SRERM are feasible and are suited in security requirement engineering process	0	1	100%	0	0	0	0	0
The SRERM can be used effectively to identify SRE readiness issues with a goal of increasing organizations readiness for security requirement engineering.	0	1	100%	0	0	0	0	0
The distribution of security component among various readiness levels (e.g. identification, authentication, and authorization) is valuable	0	1	100%	0	0	0	0	0
Four readiness levels of SRERM are valuable	0	1	100%	0	0	0	0	0

6.6 Case Studies Lesson Learned

There are several lessons we obtained through the case study of SRERM. First, we learned how to prepare a well-designed questionnaire. The structure and the design of the questionnaire, including the proper description, indirectly contribute to the comfort of respondents in completing all sections. It will motivate them to finish the SRERM by providing an appropriate answer. In addition, it is essential by respecting the value of our respondents' time. When a questionnaire is not properly designed, the respondent will probably need a longer time to finish the SRERM.

Second, we ascertained a suitable strategy for transferring SRERM knowledge. Although the document of the SRERM has been reviewed several times, the respondents

may encounter some ambiguities. We provide an introduction to the SRERM for the respondents and how to complete before asking them to undertake it. If we ask them look directly at the document and to complete it, we probably have to assist them and attend to their confusion.

Third, we identified how to analyze the results of the SRERM. The respondents of our case study have differing characteristics, such as the duration of experience, number of branches, number of customers and number of the team members. Due to the mentioned differences, we carefully extract the information, analyze the reason behind the result and propose the summary.

Lastly, we determined how to improve the SRERM. The feedback from our respondents is beneficial for enhancing the usability of the SRERM. We noticed that the comments and recommendations from the respondents are essential to ensure the applicability of the SRERM in the software organizations.

6.7 Threat to Validity

This research has a limitation regarding the outcomes of the conducted SMAPS which is utilized for developing the SRERM. When selecting primary studies and extracting the data, subjective decisions may occur. A reason for this is that some primary studies do not have enough clear description, discussion and contributions. Another potential issue is the SMAPS was performed by one individual reviewer. Consequently, we mitigated these limitations by utilizing mapping study assistant software, undertaking an iterative selection process, and extracting the data comprehensively.

Another limitation is that this study retrieves publications from five research electronic databases only. Some relevant publications may exist in other research

electronic databases which are not included in this research. Studies which were published since this research was undertaken could have been missed. Nevertheless, we believe our outcomes cover the most relevant published literature.

Realizing this research conducted a case study only on two organizations, it has external validity. How the findings can generalize the applicability of the SRERM into other organizations will be a challenge. The SRERM was evaluated by two organizations which have different characteristics, so generalization of the findings into other organizations needs careful consideration.

CHAPTER 7

CONCLUSION

7.1 Conclusion

This study developed a security requirements engineering readiness model (SRERM). The purpose of the SRERM is to provide a model which has ability to measure the readiness level of SRE activities in software development organizations. The organizations are expected to be able to reduce their vulnerabilities in terms of SRE in order to produce secure software.

A systematic mapping study (SMAPS) was an essential part of this research. It was comprehensively conducted at the beginning of research. It was purposed to provide more information and the current state of SRE. In addition, 104 primary studies were analyzed to uncover gaps in the research. Eventually, the security requirement categories were utilized in the SRERM development.

The SRERM has a structure which consists of levels, components, and practices. This study presented the SRERM with five levels before conducting a case study, and it was changed into four levels after analyzing the respondents' feedback. Each level contains various security components which are referred to security requirement categories in the SMAPS. This study utilized the Motorola assessment tool [22] as an assessment for each practice in the SRERM. The calculated result for each practice will define the level of organization readiness in terms of SRE.

In order to assess the usability of the SRERM, case studies were conducted in two software organizations. The first is an international organization, which has several branches and customers around the world. The second organization is a growing one, which provides a support system for a university. Due to the difference in characteristics, the results of both case studies were carefully analyzed.

The outcomes of the case studies and the respondents' feedback motivated some modifications to the SRERM. The changes include moving the security requirements component from one level to another level and merging the anticipated level with the monitored level. The modified SRERM has been investigated by conducting second case studies and we obtained that its feedback is positive.

7.2 Recommendations

Considering the trend of software needs is growing, this research offers some potential suggestions for future research.

- To continue this work, the SRERM still needs to generate comprehensive outcomes due to different security mechanisms and facilities in various organizations. It needs more collaboration with several software development organizations. Some organizations which have security third parties or a large number of security experts will have a different priority for implementing SRE from the growing organizations.
- There are various recent technologies such as cloud computing, Internet of Things (IoT) and Virtual Reality, which have special characteristics and were not covered in this research. However, there is a challenge to find suitable organizations to carry out these kinds of studies.

REFERENCES

- [1] G. McGraw, *Software Security Building Security In*. Boston: Pearson Education Ltd., 2006.
- [2] C. Wong, *Security Metrics: A Beginner's Guide*. New York: McGraw-Hill Companies, 2012.
- [3] M. Rodhes, *Infomation Security: The Complete Reference, Second Edition*. New York: McGraw-Hill Companies, 2013.
- [4] G. McGraw, "Software Security," *Datenschutz und Datensicherheit - DuD*, vol. 36, no. 9, pp. 662–665, 2012.
- [5] M. Stamp, *Information Security : Principles and Practice*. New Jersey: John Wiley & Sons, Inc, 2011.
- [6] I. El Kassmi and Z. Jarir, "Security Requirements in Web Service Composition: Formalization, Integration, and Verification," in *Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE*, pp. 179–184, 2016.
- [7] R. Jindal, R. Malhotra, and A. Jain, "Automated Classification of Security Requirements," in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 2027–2033, 2016.
- [8] N. Rjaibi and L. Ben Arfa Rabai, "Developing a Novel Holistic Taxonomy of Security Requirements," *Procedia Comput. Sci.*, vol. 62, pp. 213–220, 2015.
- [9] H. El-Hadary and S. El-Kassas, "Capturing Security Requirements for Software Systems," *J. Adv. Res.*, vol. 5, no. 4, pp. 463–472, Jul. 2014.
- [10] P. Salini and S. Kanmani, "Elicitation of Security Requirements for E-Health System by Applying Model Oriented Security Requirements Engineering (MOSRE) Framework," *ACM Second Int. Conf. Comput. Sci. Eng. Inf. Technol. CCSEIT*, pp. 126–131, 2012.
- [11] S. Faily and I. Fl, "Eliciting Usable Security Requirements with Misusability Cases," in *19th IEEE InternationalRequirements Engineering Conference (RE)*, pp. 339–340, 2011.
- [12] C. Gutiérrez, D. G. Rosado, and E. Fernández-medina, "The practical application of a process for eliciting and designing security in web service systems," *Inf. Softw. Technol.*, vol. 51, no. 12, pp. 1712–1738, 2009.
- [13] K. Beckers, I. Côté, and L. Goeke, "A Catalog of Security Requirements Patterns for The Domain of Cloud Computing Systems," *29th Symp. Appl. Comput.*, pp.

337–342, 2014.

- [14] I. Alqassem, “Privacy and Security Requirements Framework for the Internet of Things (IoT),” in *36th International Conference on Software Engineering*, pp. 739–741, 2014.
- [15] P. Salini and S. Kanmani, “Survey and Analysis on Security Requirements Engineering,” *Comput. Electr. Eng.*, vol. 38, no. 6, pp. 1785–1797, 2012.
- [16] A. Rashid, S. A. A. Naqvi, R. Ramdhany, M. Edwards, R. Chitchyan, and M. A. Babar, “Discovering ‘Unknown Known’ Security Requirements,” *38th Int. Conf. Softw. Eng. - ICSE*, pp. 866–876, 2016.
- [17] N. M. Mohammed, M. Niazi, M. Alshayeb, and S. Mahmood, “Exploring software security approaches in software development lifecycle: A systematic mapping study,” *Comput. Stand. Interfaces*, vol. 50, pp. 107–115, 2017.
- [18] K. Petersen, R. Feldt, S. Mujtaba, and M. Mattsson, “Systematic mapping studies in software engineering,” *EASE’08 Proc. 12th Int. Conf. Eval. Assess. Softw. Eng.*, pp. 68–77, 2008.
- [19] S. Khan, M. Niazi, and R. Ahmad, “A Readiness Model for Software Development Outsourcing Vendors,” in *2008 IEEE International Conference on Global Software Engineering*, pp. 273–277, 2008.
- [20] M. Niazi, D. Wilson, and D. Zowghi, “Organisational Readiness and Software Process Improvement,” *Prod. Softw. Process Improv.*, vol. 4589, pp. 96–107, 2007.
- [21] C. Wu, “A Readiness Model for Adopting Web services,” *J. Enterp. Inf. Manag.*, vol. 17, no. 5, pp. 361–371, 2004.
- [22] M. K. Daskalantonakis, “Achieving Higher SEI Levels,” *IEEE Softw.*, vol. 11, no. 4, pp. 17–24, 1994.
- [23] “Cobit Overview.” [Online]. Available: <http://www.isaca.org/cobit/pages/default.aspx>. [Accessed: 11-Sep-2017].
- [24] “ISO/IEC 27001:2013,” 2013. [Online]. Available: <https://www.iso.org/standard/54534.html>. [Accessed: 11-Sep-2017].
- [25] “NIST SP 800-53 rev 3 - Recommended Security Controls for Federal Information Systems,” 2011. [Online]. Available: http://www.nist.org/nist_plugins/content/content.php?content.18. [Accessed: 11-Sep-2017].
- [26] W. Stallings, “Standards for Information Security Management.” [Online]. Available: <https://www.cisco.com/c/en/us/about/press/internet-protocol->

- journal/back-issues/table-contents-38/104-standards.html. [Accessed: 11-Sep-2017].
- [27] “W3C Security Activity,” 2016. [Online]. Available: <https://www.w3.org/Security/>. [Accessed: 11-Sep-2017].
 - [28] Infosec, “IT Security Standards and Best Practices.” [Online]. Available: <https://www.infosec.gov.hk/english/technical/standards.html>. [Accessed: 11-Sep-2017].
 - [29] S. University, “Minimum Security Standards.” [Online]. Available: <https://uit.stanford.edu/guide/securitystandards>. [Accessed: 11-Sep-2017].
 - [30] P. Tsai, “Server Virtualization and OS Trends,” 2016. [Online]. Available: <https://community.spiceworks.com/networking/articles/2462-server-virtualization-and-os-trends>. [Accessed: 11-Sep-2017].
 - [31] “Server Operating Systems,” 2002. [Online]. Available: https://www.pcworld.idg.com.au/article/151491/server_operating_systems/. [Accessed: 11-Sep-2017].
 - [32] G. McGraw, “Software security,” *IEEE Secur. Priv.*, vol. 2, no. 2, pp. 80–83, Mar. 2004.
 - [33] Klaus Pohl, *Requirement Engineering: Fundamentals, Principles, and Techniques*. Germany, 2010.
 - [34] D. Leffingwell and D. Widrig, *Managing Software Requirements: A Use Case Approach, Second Edition*, Second Edi. Addison Wesley, 2003.
 - [35] C. Haley, “Arguing Security: A Framework for Analyzing Security Requirements,” *the-Haleys.Com*, March, 2007.
 - [36] P. Jaferian, G. Elahi, M. R. A. Shirazi, and B. Sadeghian, “RUPSec: Extending Business Modeling and Requirements Disciplines of RUP for Developing Secure Systems,” in *Software Engineering and Advanced Applications. 31st EUROMICRO Conference*, pp. 232–239, 2005.
 - [37] Common Criteria Implementation Board, *Common Criteria for Information Technology Security Evaluation Part 2 : Security functional components*, 2012.
 - [38] S. Ali and S. U. Khan, “Software outsourcing partnership model: An evaluation framework for vendor organizations,” *J. Syst. Softw.*, vol. 117, pp. 402–425, 2016.
 - [39] L. Lin, B. Nuseibeh, and I. D., “Using Abuse Frames to Bound the Scope of Security Problems,” in *12th IEEE International on Requirements Engineering Conference*, 2004.

- [40] L. Lin, B. Nuseibeh, and D. Ince, “Introducing Abuse Frames for Analysing Security Requirements,” in *11th International Conference on Requirements Engineering Conference*, 2003.
- [41] G. Sindre and A. L. Opdahl, “Eliciting security requirements with misuse cases,” *Requir. Eng.*, vol. 10, no. 1, pp. 34–44, 2005.
- [42] I. A. Tøndel, J. Jensen, and L. Røstad, “Combining Misuse Cases with Attack Trees and Security Activity Models,” *5th Int. Conf. Availability, Reliab. Secur.*, pp. 438–445, 2010.
- [43] B. Schneier, “Attack Trees,” *Dr. Dobb’s Portal*, no. December 1999, pp. 21–23, 2001.
- [44] V. Gandotra, A. Singhal, and P. Bedi, “Identifying Security Requirements Hybrid Technique,” *4th Int. Conf. Softw. Eng. Adv. ICSEA*, pp. 407–412, 2009.
- [45] J. McDermott and C. Fox, “Using Abuse Case Models for Security Requirements Analysis,” *15th Annu. Comput. Secur. Appl. Conf.*, pp. 55–64, 1999.
- [46] F. Dalpiaz, E. Paja, and P. Giorgini, “Security Requirements Engineering via Commitments,” *Proc. Fifth Int. i* Work.*, pp. 1–8, 2011.
- [47] M. Q. Saleem, J. Jaafar, and M. F. Hassan, “Model Driven Security Framework for Definition of Security Requirements for SOA Based Applications,” in *International Conference on Computer Applications and Industrial Electronics, ICCAIE*, pp. 266–270, 2010.
- [48] P. Salini, “Elicitation of Security Requirements for E-Health System by applying Model Oriented Security Requirements Engineering (MOSRE) Framework,” pp. 126–131, 2012.
- [49] D. Mellado, E. Fernández-Medina, and M. Piattini, “Security Requirements Engineering Framework for Software Product Lines,” *Inf. Softw. Technol.*, vol. 52, no. 10, pp. 1094–1117, 2010.
- [50] M. S. Ware, J. B. Bowles, and C. M. Eastman, “Using the Common Criteria to Elicit Security Requirements with Use Cases,” in *SoutheastCon*, pp. 273–278, 2005.
- [51] D. Mellado, H. Mouratidis, and E. Fernández-Medina, “Secure Tropos Framework for Software Product Lines Requirements Engineering,” *Comput. Stand. Interfaces*, vol. 36, no. 4, pp. 711–722, 2014.
- [52] M. Riaz, J. Stallings, M. P. Singh, J. Slankas, and L. Williams, “DIGS – A Framework for Discovering Goals for Security Requirements Engineering,” *Emprical Softw. Eng. Meas.*, 2016.

- [53] K. Beckers, M. Heisel, I. Cote, L. Goeke, and S. Guler, "Structured Pattern-Based Security Requirements Elicitation for Clouds," in *International Conference on Availability, Reliability and Security, ARES*, pp. 465–474, 2013.
- [54] A. Akinbi and E. Pereira, "Mapping Security Requirements to Identify Critical Security Areas of Focus in PaaS Cloud Models," in *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM)*, pp. 789–794, 2015.
- [55] S. Islam, M. Ouedraogo, C. Kalloniatis, H. Mouratidis, and S. Gritzalis, "Assurance of Security and Privacy Requirements for Cloud Deployment Model," *IEEE Trans. Cloud Comput.*, 2015.
- [56] W. M. Tellis, "Application of a Case Study Methodology," *Qual. Rep.*, vol. 3, pp. 1–19, 1997.
- [57] R. Yin, *Case study research: Design and methods (2nd ed.)*. Sage Publishing, 1994.
- [58] S. Ouhbi, A. Idri, J. L. Fernández-Alemán, and A. Toval, "Requirements engineering education: a systematic mapping study," *Requir. Eng.*, vol. 20, no. 2, pp. 119–138, 2015.
- [59] W. S. Al-Shorafat, "Security in Software Engineering Requirement.," in *International Conference for Internet Technology and Secured Transactions (ICITST)*, pp. 666–673, 2013.
- [60] N. Zafar, E. Arnautovic, A. Diabat, and D. Svetinovic, "System Security Requirements Analysis:A Smart Grid Case Study," *Syst. Eng.*, vol. 14, no. 3, pp. 305–326, 2011.
- [61] M. Felderer, B. Agreiter, and R. Breu, "Evolution of security requirements tests for service-centric systems," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6542 LNCS, pp. 181–194, 2011.
- [62] F. Donald, "Engineering security requirements.," *J Object Technol*, vol. 2, no. 1, p. 53–68., 2003.
- [63] P. Salini and S. Kanmani, "A Model Based Security Requirements Engineering Framework Applied for Online Trading System," in *International Conference on Recent Trends in Information Technology (ICRTIT)*, pp. 1195–1202, 2011.
- [64] S. Faily and I. Fl, "Eliciting Usable Security Requirements with Misusability Cases," pp. 339–340, 2011.
- [65] F. Z. Jorshari, H. Mouratidis, and S. Islam, "Extracting Security Requirements from Relevant Laws and Regulations," in *Sixth International Conference on*

- [66] Y. Roudier, M. S. Idrees, and L. Apvrille, "Towards the Model-driven Engineering of Security Requirements for Embedded Systems," *Int. Work. Model. Requir. Eng.*, pp. 55–64, 2013.
- [67] N. Ikram, S. Siddiqui, and N. F. Khan, "Security Requirement Elicitation Techniques: The Comparison of Misuse Cases and Issue Based Information Systems," in *IEEE Fourth International Workshop on Empirical Requirements Engineering (EmpiRE)*, pp. 36–43, 2014.
- [68] D. Mellado, E. Fernández-Medina, and M. Piattini, "Towards Security Requirements Management for Software Product Lines: A Security Domain Requirements Engineering Process," *Comput. Stand. Interfaces*, vol. 30, no. 6, pp. 361–371, 2008.
- [69] M. S. Ware, J. B. Bowles, and C. M. Eastman, "Using the Common Criteria to Elicit Security Requirements with Use Cases," in *SoutheastCon*, pp. 273–278, 2006.
- [70] F. Lins, R. Medeiros, B. Silva, A. Souza, D. Arag??o, J. Damasceno, P. MacIel, N. Rosa, B. Stephenson, and J. Li, "SSC4Cloud Tooling: an Integrated Environment for the Development of Business Processes with Security Requirements in the Cloud," in *IEEE World Congress on Services, SERVICES*, pp. 53–60, 2011.
- [71] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone, "ST-Tool: A CASE Tool for Security Requirements Engineering * ," pp. 3–4, 2005.
- [72] V. Gandotra, A. Singhal, and P. Bedi, "Identifying Security Requirements Hybrid Technique," in *Fourth International Conference on Software Engineering Advances*, pp. 407–412, 2009.
- [73] D. Mellado, E. Fernandez-Medina, and M. Piattini, "Automated Support for Security Requirements Engineering in Software Product Line Domain Engineering," in *Availability, Reliability and Security*, 2009, vol. 6, no. 3, pp. 298–305.
- [74] P. Salini and S. Kanmani, "Application of Model Oriented Security Requirements Engineering Framework for secure E-Voting," *6th Int. Conf. Softw. Eng. CONSEG*, 2012.
- [75] S. U. Khan, M. Niazi, and R. Ahmad, "Critical Success Factors for Offshore Software Development Outsourcing Vendors: A Systematic Literature Review," *IEEE Int. Conf. Glob. Softw. Eng. ICGSE*, pp. 207–216, 2009.
- [76] S. Lawrence, "A Framework for Security Requirements," vol. 10, pp. 515–523, 1991.

- [77] K. Muralidhar, R. Sarathy, and R. Parsa, “An Improved Security Requirement for Data Perturbation with Implications for E-Commerce,” *Decis. Sci.*, vol. 32, no. 4, pp. 683–698, 2001.
- [78] C. Lambrinoudakis, S. Gritzalis, F. Dridi, and G. Pernul, “Security Requirements for e-Government Services: A Methodological Approach for Developing a Common PKI-based Security Policy,” *Comput. Commun.*, vol. 26, no. 16 SPEC., pp. 1873–1883, 2003.
- [79] A. Zuccato, “Holistic Security Requirement Engineering for Electronic Commerce,” *Comput. Secur.*, vol. 23, no. 1, pp. 63–76, Feb. 2004.
- [80] D. Cotroneo, A. Graziano, and S. Russo, “Security Requirements in Service Oriented Architectures for Ubiquitous Computing,” pp. 172–177, 2004.
- [81] A. van Lamsweerde, “Elaborating Security Requirements by Construction of Intentional Anti-Models,” *26th Int. Conf. Softw. Eng.*, vol. 26, no. May, pp. 148–157, 2004.
- [82] G. T. Sigrid, P. Ochsenschl, C. Rudolph, S. Gürgens, P. Ochsenschläger, and C. Rudolph, “On a Formal Framework for Security Properties,” *Comput. Stand. Interfaces*, vol. 27, no. 5, pp. 457–466, 2005.
- [83] P. Giorgini, F. Massacci, J. Mylopoulos, and N. Zannone, “Modeling Security Requirements Through Ownership, Permission and Delegation,” in *13th IEEE International Conference on Requirement Engineering*, 2005.
- [84] E. Anderson, J. Choobineh, and M. R. Grimaila, “An Enterprise Level Security Requirements Specification Model,” in *38th Annual Hawaii International Conference on System Sciences*, 2005.
- [85] M. N. Kreeger and I. Duncan, “Engineering Secure Software by Modelling Privacy and Security Requirements,” in *39th Annual International Carnahan Conference on Security Technology.*, 2005.
- [86] F. Massacci, M. Prest, and N. Zannone, “Using a Security Requirements Engineering Methodology in Practice: The Compliance With the Italian Data Protection Legislation,” *Comput. Stand. Interfaces*, vol. 27, no. 5, pp. 445–455, 2005.
- [87] J. Viega, “Building Security Requirements with CLASP,” *ACM SIGSOFT Softw. Eng. Notes*, vol. 30, p. 1, 2005.
- [88] J. D. Moffett and B. A. Nuseibeh, “A Framework for Security Requirements Engineering,” *Australas. Inf. Secur. Conf.*, 2006.
- [89] N. Hallberg and J. Hallberg, “The Usage-Centric Security Requirements Engineering (USeR) Method,” *Work. Inf. Assur.*, pp. 34–41, 2006.

- [90] J. Pauli, "Integrating Functional and Security Requirements with Use Case Decomposition," in *11th IEEE International Conference on Engineering of Complex Computer Systems (ICECCS'06)*, 2006.
- [91] A. Rodríguez, E. Fernández-Medina, M. Piattini, B. Bio, E. Fernández-Medina, M. Piattini, and C. Real, "Security Requirement with a UML 2.0 Profile," in *The First International Conference on Availability, Reliability and Security*, 2006.
- [92] Y. Luo, G. Antoniou, and L. Sterling, "Incorporating Security Requirements into Communication Protocols in Multi-Agent Software Systems," in *Parallel and Distributed Computing, Applications and Technologies*, pp. 159–160, 2007.
- [93] J.-S. Cui and D. Zhang, "The Research and Application of Security Requirements Analysis Methodology of Information Systems," in *2nd International Conference on Anti-counterfeiting, Security and Identification*, pp. 30–36, 2008.
- [94] D. Mellado, E. Fernández-Medina, and M. Piattini, "Security Requirements Engineering process for Software Product Lines: A Case Study," in *3rd International Conference on Software Engineering Advances, ICSEA*, pp. 1–6, 2008.
- [95] K. Saleh and M. Habil, "The Security Requirements Behavior Model for Trustworthy Software," in *International MCETECH Conference on e-Technologies, MCETECH*, pp. 235–238, 2008.
- [96] E. Soler, V. Stefanov, J.-N. Mazon, J. Trujillo, E. Fernandez-Medina, and M. Piattini, "Towards Comprehensive Requirement Analysis for Data Warehouses: Considering Security Requirements," *Third Int. Conf. Availability, Reliab. Secur. ARES*, pp. 104–111, 2008.
- [97] D. Mellado, E. Fernández-Medina, and M. Piattini, "Towards Security Requirements Management for Software Product Lines: A Security Domain Requirements Engineering Process," *Comput. Stand. Interfaces*, vol. 30, no. 6, pp. 361–371, 2008.
- [98] A. Gouglidis and I. Mavridis, "A Foundation for Defining Security Requirements in Grid Computing," in *13th Panhellenic Conference on Informatics*, 2009, pp. 180–184.
- [99] N. R. Mead, D. Shoemaker, and J. Ingalsbe, "Teaching Security Requirements Engineering Using SQUARE," *Fourth Int. Work. Requir. Eng. Educ. Train.*, pp. 20–27, 2009.
- [100] H. Belani, Ž. Car, and A. Carić, "RUP-Based Process Model for Security Requirements Engineering in Value-Added Service Development," *Work. Softw. Eng. Secur. Syst. ICSE Work.*, pp. 54–60, 2009.
- [101] J. Trujillo, E. Soler, E. Fernández-Medina, and M. Piattini, "A UML 2.0 Profile to

- Define Security Requirements for Data Warehouses,” *Comput. Stand. Interfaces*, vol. 31, no. 5, pp. 969–983, 2009.
- [102] A. L. Opdahl and G. Sindre, “Experimental Comparison of Attack Trees and Misuse Cases for Security Threat Identification,” *Inf. Softw. Technol.*, vol. 51, no. 5, pp. 916–932, 2009.
 - [103] M. Menzel, I. Thomas, and C. Meinel, “Security Requirements Specification in Service-Oriented Business Process Management,” *Int. Conf. Availability, Reliab. Secur.*, pp. 41–48, 2009.
 - [104] L. Langer, A. Schmidt, J. Buchmann, M. Volkamer, and A. Stolfik, “Towards a Framework on the Security Requirements for Electronic Voting Protocols,” *1st Int. Work. Requir. Eng. e-Voting Syst. RE-VOTE*, 2009.
 - [105] L. Yin and F.-L. Qiu, “A Novel Method of Security Requirements Development Integrated Common Criteria,” *Int. Conf. Comput. Des. Appl.*, vol. 5, pp. 531–535, 2010.
 - [106] M. Ficco, F. Palmieri, and A. Castiglione, “Modeling Security Requirements for Cloud-based System Development,” *Concurr. Comput. Pract. Exp.*, vol. 27, pp. 2107–2124, 2015.
 - [107] K. Taguchi, N. Yoshioka, T. Tobita, and H. Kaneko, “Aligning Security Requirements and Security Assurance Using the Common Criteria,” in *4th International Conference on Secure Software Integration and Reliability Improvement*, pp. 69–77, 2010.
 - [108] L. Cheng, Y. Zhang, and D. Feng, “A Language For Secure Requirement Description Based on Information Flow,” in *IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS)*, 2010.
 - [109] R. Warschofsky, M. Menzel, and C. Meinel, “Transformation and Aggregation of Web Service Security Requirements,” in *8th European Conference on Web Services, ECOWS*, pp. 43–50, 2010.
 - [110] H. Schmidt, “Threat and Risk Analysis during Early Security Requirements Engineering,” *5th Int. Conf. Availability, Reliab. Secur.*, pp. 188–195, 2010.
 - [111] S. Al-Fedaghi and K. A. Al-Enazi, “Extracting Security Requirements from Reality,” *3rd Int. Conf. Comput. Res. Dev.*, vol. 1, pp. 221–228, 2011.
 - [112] R. K. Abercrombie, F. T. Sheldon, and A. Mili, “Validating Cyber Security Requirements: A Case Study,” *44th Hawaii Int. Conf. Syst. Sci.*, pp. 1–10, 2011.
 - [113] P. Karpati, G. Sindre, and A. L. Opdahl, “Characterising and Analysing Security Requirements Modelling Initiatives,” in *6th International Conference on Availability, Reliability and Security, ARES*, no. 1, pp. 710–715, 2011.

- [114] K. K. Fletcher and X. Liu, "Security Requirements Analysis, Specification, Prioritization and Policy Development in Cyber-Physical Systems," in *5th International Conference on Secure Software Integration and Reliability Improvement - Companion*, pp. 106–113, 2011.
- [115] K. Khajaria and M. Kumar, "Modeling of Security Requirements for Decision Information Systems," *SIGSOFT Softw. Eng. Notes*, vol. 36, no. 5, p. 1, 2011.
- [116] F. Dalpiaz, E. Paja, and P. Giorgini, "Security Requirements Engineering via Commitments," in *Socio-Technical Aspects in Security and Trust (STAST), 1st Workshop*, pp. 1–8, 2011.
- [117] G. R. Haron and N. K. Siong, "Extrapolating Security Requirements to an Established Software Process: Version 1.0," in *International Conference for Internet Technology and Secured Transactions*, pp. 752–757, 2011.
- [118] G. Fan, H. Yu, L. Chen, and D. Liu, "An Approach to Modeling and Analyzing Security Requirements of Service Composition," *Serv. Comput. Conf. (APSCC), 2011 IEEE Asia-Pacific*, pp. 456–463, 2011.
- [119] T. D. Breaux and D. L. Baumer, "Legally 'reasonable' Security Requirements: A 10-year FTC Retrospective," *Comput. Secur.*, vol. 30, no. 4, pp. 178–193, 2011.
- [120] Y. Wen, Haihong Zhao, and L. Liu, "Analysing Security Requirements Patterns Based on Problems Decomposition and Composition," *1st Int. Work. Requir. Patterns, RePa*, pp. 11–20, 2011.
- [121] P. Salini and S. Kanmani, "Security Requirements Engineering Process for Web Applications," *Procedia Eng.*, vol. 38, pp. 2799–2807, 2012.
- [122] S. Tejas R and P. S. V, "Security, Privacy and Trust Oriented Requirements Modeling for Examination System," in *Nirma University International Conference on Engineering (NUIcone)*, pp. 6–8, 2012.
- [123] T. Okubo, H. Kaiya, and N. Yoshioka, "Mutual Refinement of Security Requirements and Architecture using Twin Peaks Model," *Int. Comput. Softw. Appl. Conf.*, pp. 367–372, 2012.
- [124] D. Mougouei, W. N. W. A. Rahman, and M. Moein Almasi, "Evaluating Fault Tolerance in Security Requirements of Web Services," in *International Conference on Cyber Security, Cyber Warfare and Digital Forensic, CyberSec*, pp. 111–116, 2012.
- [125] S. Almutairi, G. Bella, and A. Abu-samaha, "Specifying Security Requirements of Context Aware System Using UML," in *Seventh International Conference on Digital Information Management (ICDIM)*, pp. 259–265, 2012.
- [126] G. Bibu, N. Yoshioka, and J. Padget, "System Security Requirements Analysis

- with Answer Set Programming,” in *Second International Workshop on Requirements Engineering for Systems, Services, and Systems-of-Systems (RESS)*, no. ii, pp. 10–13, 2012.
- [127] E. Paja, F. Dalpiaz, M. Poggianella, P. Roberti, and P. Giorgini, “STS-tool: Socio-Technical Security Requirements through Social Commitments,” in *20th IEEE International Requirements Engineering Conference*, pp. 331–332, 2012.
 - [128] M. Borek, N. Moebius, K. Stenzel, and W. Reif, “Security Requirements Formalized with OCL in a Model-Driven Approach,” *Int. Work. Model. Requir. Eng. MoDRE*, pp. 65–73, 2013.
 - [129] A. Guesmi and P. Clemente, “Access Control and Security Properties Requirements Specification for Clouds’ SecLAs,” in *International Conference on Cloud Computing Technology and Science, CloudCom*, vol. 1, pp. 723–729, 2013.
 - [130] S. T. Lai, F. Y. Leu, and W. C. C. Chu, “A Software Security Requirement Quality Improvement Procedure to Increase E-commerce Security,” in *8th International Conference on Broadband, Wireless Computing, Communication and Applications, BWCCA*, pp. 366–371, 2013.
 - [131] S. Faßbender, M. Heisel, and R. Meis, “Functional Requirements Under Security PresSuRE,” in *9th International Conference on Software Paradigm Trends (ICSOFT-PT)*, pp. 5–16, 2014.
 - [132] R. Hassan, S. Bohner, and S. El-Kassas, “Formal derivation of security design specifications from security requirements,” *Proc. 4th Annu. Work. Cyber Secur. Informait. Intell. Res. Dev. Strateg. to meet cyber Secur. Inf. Intell. challenges ahead - CSIIRW*, p. 1, 2008.
 - [133] M. Riaz, J. Slankas, J. King, and L. Williams, “Using Templates To Elicit Implied Security Requirements From Functional Requirements - A Controlled Experiment,” in *8th International Symposium on Empirical Software Engineering and Measurement, ESEM*, p. 22, 2014.
 - [134] M. Giacalone, F. Paci, R. Mammoliti, R. Perugino, F. Massacci, and C. Selli, “Security Triage: An Industrial Case Study on the Effectiveness of a Lean Methodology to Identify Security Requirements,” *8th ACM/IEEE Int. Symp. Empir. Softw. Eng. Meas.*, pp. 1–8, 2014.
 - [135] Y. Aoki and S. Matsuura, “Verifying Security Requirements Using Model Checking Technique for UML-Based Requirements Specification,” in *International Workshop on Requirements Engineering and Testing*, pp. 18–25, 2014.
 - [136] Y. A. Younis, K. Kifayat, and M. Merabti, “An Access Control Model for Cloud Computing,” *J. Inf. Secur. Appl.*, vol. 19, no. 1, pp. 45–60, 2014.

- [137] C. Schmitt and P. Liggesmeyer, “Instantiating a Model for Structuring and Reusing Security Requirements Sources,” *Int. Work. Evol. Secur. Priv. Requir. Eng. ESPRE*, pp. 25–30, 2015.
- [138] E. Paja, F. Dalpiaz, and P. Giorgini, “Modelling and Reasoning About Security Requirements in Socio-Technical Systems,” *Data Knowl. Eng.*, vol. 98, pp. 123–143, 2015.
- [139] I. Maskani, “Student Research Abstract: A New Comprehensive Approach to Security Requirements Engineering,” in *Symposium on Applied Computing*, pp. 1136–1137, 2017.
- [140] E. Ungan, “Using FSM Patterns to Size Security Non-Functional Requirements with COSMIC,” in *International Workshop on Software Measurement and International Conference on Software Process and Product Measurement*, 2017.
- [141] S. G. Yoo, H. P. Vaca, and J. Kim, “Enhanced Misuse Cases for Prioritization of Security Requirements,” in *International Conference on Information Management and Engineering*, pp. 1–10, 2017.
- [142] S. T. Bulusu, R. Laborde, A. S. Wazan, F. Barrère, and A. Benzekri, “Which Security Requirements Engineering Methodology Should I Choose?,” *Int. Conf. Availability, Reliab. Secur.*, pp. 1–6, 2017.
- [143] B. O. Emeka and S. Liu, “Security Requirement Engineering Using Structured Object-oriented Formal Language for M-banking Applications,” *IEEE Int. Conf. Softw. Qual. Reliab. Secur.*, pp. 176–183, 2017.
- [144] A. Ouaddah, H. Mousannif, A. Abou Elkalam, and A. Ait Ouahman, “Access Control in the Internet of Things: Big Challenges and New Opportunities,” *Comput. Networks*, vol. 112, pp. 237–262, 2017.
- [145] A. Yasin, L. Liu, T. Li, J. Wang, and D. Zowghi, “Design and Preliminary Evaluation of a Cyber Security Requirements Education Game (SREG),” *Inf. Softw. Technol.*, no. April, 2017.

APPENDICES

1. Appendix 1: List of Primary Studies

No.	Primary studies	Year	Research type	Empirical type	Approach	Publication channel
1.	A Framework for Security Requirements [76]	1991	Solution proposal	No	Framework	journal
2.	Using Abuse Case Models for Security Requirements Analysis [45]	1999	Solution proposal	No	Method	conference
3.	An Improved Security Requirement for Data Perturbation with Implications for E-Commerce [77]	2001	Solution proposal	Case study	Method	journal
4.	Security Requirements for e-Government Services: A Methodological Approach for Developing a Common PKI-based Security Policy [78]	2003	Solution proposal	Case study	Method	journal
5.	Holistic Security Requirement Engineering for Electronic Commerce [79]	2004	Solution proposal	No	Method	journal
6.	Security Requirements in Service Oriented Architectures for Ubiquitous Computing [80]	2004	Evaluation research	No	Method	conference
7.	Elaborating Security Requirements by Construction of Intentional Anti-Models [81]	2004	Solution proposal	Case study	Framework	conference
8.	On a Formal Framework for Security	2005	Solution proposal	No	Framework	journal

	Properties [82]					
9.	RUPSec: Extending Business Modeling and Requirements Disciplines of RUP for Developing Secure Systems [36]	2005	Solution proposal	No	Method	conference
10.	Modeling Security Requirements Through Ownership, Permission and Delegation [83]	2005	Solution proposal	No	Framework	conference
11.	An Enterprise Level Security Requirements Specification Model [84]	2005	Experience paper	No	Method	conference
12.	Engineering Secure Software by Modelling Privacy and Security Requirements [85]	2005	Other	No	Method	conference
13.	Using a Security Requirements Engineering Methodology in Practice: The Compliance With the Italian Data Protection Legislation [86]	2005	Evaluation research	Case study	Method	journal
14.	Eliciting security requirements with misuse cases [41]	2005	Solution proposal	No	Method	journal
15.	Building Security Requirements with CLASP [87]	2005	Experience paper	No	Framework	journal
16.	Framework for Security Requirement Engineering [88]	2006	Experience paper	No	Framework	conference
17.	Using the Common Criteria to Elicit Security Requirements with Use Cases [69]	2006	Solution proposal	No	Tool	conference
18.	The Usage-Centric Security Requirements Engineering (USEr) Method [89]	2006	Experience paper	No	Method	others
19.	Integrating Functional and Security	2006	Evaluation research	No	Method	conference

	Requirements with Use Case Decomposition [90]					
20.	Security Requirement with a UML 2.0 Profile [91]	2006	Experience paper	Case study	Method	conference
21.	Incorporating Security Requirements into Communication Protocols in Multi-Agent Software Systems [92]	2007	Solution proposal	No	Method	conference
22.	The research and application of security requirements analysis methodology of information systems [93]	2008	Evaluation research	Case study	Method	conference
23.	Security Requirements Variability for Software Product Lines [94]	2008	Solution proposal	No	Method	conference
24.	A Systematic Framework for Structured Object-Oriented Security Requirements Analysis in Embedded Systems [95]	2008	Solution proposal	No	Framework	conference
25.	The Security Requirements Behavior Model for Trustworthy Software [96]	2008	Solution proposal	No	Framework	conference
26.	Towards Comprehensive Requirement Analysis for Data Warehouses: Considering Security Requirements [94]	2008	Solution proposal	Case study	Model	conference
27.	Security Requirements Engineering process for Software Product Lines: A Case Study [93]	2008	Solution proposal	Case study	Framework	conference
28.	The Research and Application of Security Requirements Analysis Methodology of	2008	Evaluation research	Case study	Method	conference

	Information Systems [97]					
29.	Towards Security Requirements Management for Software Product Lines: A Security Domain Requirements Engineering Process [98]	2008	Solution proposal	Case study	Method	journal
30.	A Foundation for Defining Security Requirements in Grid Computing [99]	2009	Evaluation research	No	Method	conference
31.	Teaching Security Requirements Engineering Using SQUARE [99]	2009	Evaluation research	Case study	Method	journal
32.	RUP-Based Process Model for Security Requirements Engineering in Value-Added Service Development [100]	2009	Solution proposal	No	Model	others
33.	Automated Support for Security Requirements Engineering in Software Product Line Domain Engineering [73]	2009	Solution proposal	No	Tool	conference
34.	Identifying Security Requirements Hybrid Technique [44]	2009	Solution proposal	Case study	Method	conference
35.	A UML 2.0 Profile to Define Security Requirements for Data Warehouses [101]	2009	Solution proposal	Case study	Method	journal
36.	Experimental Comparison of Attack Trees and Misuse Cases for Security Threat Identification [102]	2009	Experience paper	Experiment	Method	journal
37.	Security Requirements Specification in Service-Oriented Business Process	2009	Solution proposal	No	Model	conference

	Management [103]					
38.	Towards a Framework on the Security Requirements for Electronic Voting Protocols [104]	2009	Evaluation research	No	Model	others
39.	Security requirements engineering framework for software product lines [49]	2010	Solution proposal	Case study	Framework	journal
40.	A novel method of security requirements development integrated common criteria [105]	2010	Solution proposal	No	Framework	conference
41.	Security Requirements Engineering Framework for Software Product Lines [49]	2010	Solution proposal	Case study	Framework	journal
42.	HPCTOOLKIT: Tools for performance analysis of optimized parallel programs [106]	2010	Solution proposal	No	Method	journal
43.	Aligning Security Requirements and Security Assurance Using the Common Criteria [107]	2010	Solution proposal	Case study	Framework	conference
44.	Model Driven Security Framework for Definition of Security Requirements for SOA Based Applications [47]	2010	Solution proposal	Case study	Framework	conference
45.	Combining Misuse Cases with Attack Trees and Security Activity Models [42]	2010	Solution proposal	No	Method	conference
46.	A Language For Secure Requirement Description Based on Information Flow [108]	2010	Solution proposal	No	Tool	conference
47.	Transformation and Aggregation of Web Service Security Requirements [109]	2010	Solution proposal	No	Model	conference

48.	Threat and Risk Analysis during Early Security Requirements Engineering [110]	2010	Solution proposal	Case study	Method	conference
49.	A model based security requirements engineering framework applied for online trading system [63]	2011	Solution proposal	No	Framework	conference
50.	Extracting Security Requirements from Reality [111]	2011	Solution proposal	No	Method	conference
51.	Validating Cyber Security Requirements: A Case Study [112]	2011	Solution proposal	Case study	Method	conference
52.	Characterizing and Analyzing Security Requirements Modelling Initiatives [113]	2011	Solution proposal	No	Guideline	conference
53.	Security Requirements Analysis, Specification, Prioritization and Policy Development in Cyber-Physical Systems [114]	2011	Experience paper	Case study	Method	conference
54.	Modeling of Security Requirements for Decision Information Systems [115]	2011	Solution proposal	Case study	Model	journal
55.	Security Requirements Engineering via Commitments [116]	2011	Solution proposal	No	Framework	others
56.	SSC4Cloud Tooling: an Integrated Environment for the Development of Business Processes with Security Requirements in the Cloud [70]	2011	Solution proposal	Case study	Tool	conference
57.	Extrapolating Security Requirements to an Established Software Process: Version 1.0 [117]	2011	Experience paper	No	Model	conference

58.	An Approach to Modeling and Analyzing Security Requirements of Service Composition [118]	2011	Experience paper	Experiment	Model	conference
59.	Legally "reasonable" Security Requirements: A 10-year FTC Retrospective [119]	2011	Evaluation research	Case study	Method	journal
60.	Eliciting Usable Security Requirements with Misusability Cases [11]	2011	Solution proposal	Case study	Method	conference
61.	Evolution of Security Requirements Tests for Service – Centric Systems [61]	2011	Solution proposal	Case study	Method	journal
62.	System Security Requirements Analysis: A Smart Grid Case Study [60]	2011	Solution proposal	Case study	Method	journal
63.	Analyzing Security Requirements Patterns Based on Problems Decomposition and Composition [120]	2011	Solution proposal	No	Framework	others
64.	Security Requirements Engineering Process for Web Applications [121]	2012	Solution proposal	No	Method	conference
65.	Security, Privacy and Trust Oriented Requirements Modeling for Examination System [122]	2012	Solution proposal	Case study	Framework	conference
66.	Elicitation of Security Requirements for E-Health System by Applying Model Oriented Security Requirements Engineering (MOSRE) Framework [10]	2012	Solution proposal	Case study	Framework	journal

67.	Mutual Refinement of Security Requirements and Architecture using Twin Peaks Model [123]	2012	Solution proposal	Case study	Model	conference
68.	Evaluating Fault Tolerance in Security Requirements of Web Services [124]	2012	Evaluation research	Case study	Model	conference
69.	Survey and Analysis on Security Requirements Engineering [15]	2012	Evaluation research	Survey	Method	journal
70.	Specifying Security Requirements of Context Aware System Using UML [125]	2012	Solution proposal	Case study	Method	conference
71.	Extracting Security Requirements from Relevant Laws and Regulations [65]	2012	Solution proposal	Case study	Method	conference
72.	Application of Model Oriented Security Requirements Engineering Framework for secure E-Voting [74]	2012	Solution proposal	Case study	Framework	conference
73.	System Security Requirements Analysis with Answer Set Programming [126]	2012	Solution proposal	Case study	Framework	others
74.	STS-tool: Socio-Technical Security Requirements through Social Commitments [127]	2012	Solution proposal	No	Tool	conference
75.	Security Requirements Engineering Process for Web Applications [121]	2012	Solution proposal	No	Model	journal
76.	Security requirements formalized with OCL in a model-driven approach [128]	2013	Solution proposal	No	Method	conference

77.	Access Control and Security Properties Requirements Specification for Clouds' SecLAs [129]	2013	Solution proposal	No	Method	conference
78.	Structured Pattern-Based Security Requirements Elicitation for Clouds [53]	2013	Solution proposal	Case study	Tool	conference
79.	A Software Security Requirement Quality Improvement Procedure to Increase E-commerce Security [130]	2013	Evaluation research	No	Model	conference
80.	Security in Software Engineering Requirement. [59]	2013	Other	No	Guideline	conference
81.	Towards the Model-driven Engineering of Security Requirements for Embedded Systems [66]	2013	Solution proposal	Case study	Method	journal
82.	Capturing security requirements for software systems [9]	2014	Solution proposal	No	Method	journal
83.	A Catalog of Security Requirements Patterns for The Domain of Cloud Computing Systems [13]	2014	Solution proposal	Case study	Method	journal
84.	Privacy and Security Requirements Framework for the Internet of Things (IoT) [14]	2014	Solution proposal	No	Framework	conference
85.	Secure Tropos Framework for Software Product Lines Requirements Engineering [51]	2014	Solution proposal	Case study	Framework	journal
86.	Security Requirement Elicitation Techniques: The Comparison of Misuse Cases and Issue Based Information	2014	Evaluation research	Experiment	Method	others

	Systems [67]					
87.	Functional Requirements Under Security PresSuRE [131]	2014	Solution proposal	Case study	Method	conference
88.	From Goal-Driven Security Requirements Engineering to Secure Design [132]	2014	Solution proposal	No	Method	journal
89.	Using Templates To Elicit Implied Security Requirements From Functional Requirements - A Controlled Experiment [133]	2014	Solution proposal	Experiment	Method	others
90.	Security Triage: An Industrial Case Study on the Effectiveness of a Lean Methodology to Identify Security Requirements [134]	2014	Experience paper	Case study	Method	others
91.	Verifying Security Requirements Using Model Checking Technique for UML-Based Requirements Specification [135]	2014	Solution proposal	No	Tool	others
92.	An Access Control Model for Cloud Computing [136]	2014	Solution proposal	Case study	Model	journal
93.	Instantiating a Model for Structuring and Reusing Security Requirement Sources [137]	2015	Solution proposal	No	Model	others
94.	Developing a Novel Holistic Taxonomy of Security Requirements [8]	2015	Evaluation research	No	Method	conference
95.	Modeling and Reasoning About Security Requirements in Socio-Technical Systems [138]	2015	Experience paper	Case study	Tool	journal

96.	Automated Classification of Security Requirements [7]	2016	Solution proposal	Case study	Model	conference
97.	Security Requirements in Web Service Composition: Formalization, Integration, and Verification [6]	2016	Solution proposal	Case study	Model	conference
98.	Student Research Abstract: A New Comprehensive Approach to Security Requirements Engineering [139]	2017	Evaluation research	No	Guideline	others
99.	Using FSM Patterns to Size Security Non-Functional Requirements with COSMIC [140]	2017	Experience paper	Case study	Method	conference
100.	Enhanced Misuse Cases for Prioritization of Security Requirements [141]	2017	Solution proposal	Case study	Method	conference
101.	Which Security Requirements Engineering Methodology Should I Choose? [142]	2017	Evaluation research	Case study	Method	conference
102.	Security Requirement Engineering Using Structured Object-oriented Formal Language for M-banking Applications [143]	2017	Solution proposal	Case study	Framework	conference
103.	Access Control in the Internet of Things: Big Challenges and New Opportunities [144]	2017	Evaluation research	Survey	Method	journal
104.	Design and Preliminary Evaluation of a Cyber Security Requirements Education Game (SREG) [145]	2017	Experience paper	Experiment	Tool	journal

2. Appendix 2: The Practices of SRERM

2.1 Identification Security Requirements Practices

No.	Practices
1.	Utilize brainstorming technique to aggregate identification security requirement
2.	Identify system stakeholders to improve identification security requirement
3.	Plan for conflicts and conflict resolution for identification security requirement in term of stakeholders
4.	Define standard templates for describing identification security requirement
5.	Use languages simply and concisely to explain identification security requirement
6.	Check that identification security requirement meets your standard
7.	Define change management policies for identification security requirement

2.2 Authentication Security Requirements Practices

No.	Practices
1.	Use scenarios to elicit authentication security requirement
2.	Plan for conflicts and conflict resolution for authentication security requirement in term of multiple accounts
3.	Define standard templates for describing authentication security requirement
4.	Use languages simply and concisely to explain authentication security requirement
5.	Check that authentication security requirement meets your standard
6.	Define change management policies for authentication security requirement

2.3 Authorization Security Requirements Practices

No.	Practices
1.	Use scenarios to elicit the roles of stakeholders in term of authorization SR
2.	Plan for conflicts and conflict resolution for authorization SR in term of multiple roles
3.	Define standard templates for describing authorization SR
4.	Use languages simply and concisely to explain authorization SR
5.	Check that authorization SR meets your standard
6.	Define change management policies for authorization SR

2.4 Immunity Security Requirements Practices

No.	Practices
1.	Define the system's operation environment to gain immunity SR
2.	Assess immunity SR in term of undesirable programs
3.	Define standard templates for describing immunity SR
4.	Use languages simply and concisely to explain immunity SR
5.	Check that immunity SR meets your standard
6.	Define change management policies for immunity SR

2.5 Privacy Security Requirements Practices

No.	Practices
1.	Use scenarios to elicit the sensitive data and communication in term of privacy SR
2.	Define the system boundaries in term of privacy SR such as sensitive data and communication.
3.	Define standard templates for describing privacy SR
4.	Use languages simply and concisely to explain privacy SR
5.	Check that privacy SR meets your standard
6.	Define change management policies for privacy SR

2.6 Integrity Security Requirements Practices

No.	Practices
1.	Define the operational processes to gain integrity SR
2.	Assess integrity SR risks
3.	Define standard templates for describing integrity SR
4.	Use languages simply and concisely to explain integrity SR
5.	Check that integrity SR meets your standard
6.	Define change management policies for integrity SR

2.7 Physical Protection Security Requirements Practices

No.	Practices
1.	Be sensitive to organizational and political consideration in gaining physical protection of SR
2.	Assess physical protection SR risks
3.	Define standard templates for describing physical protection SR
4.	Use languages simply and concisely to explain physical protection SR
5.	Check that physical protection SR meets your standard
6.	Define change management policies for physical protection SR

2.8 Non-repudiation Security Requirements Practices

No.	Practices
1.	Define operational processes to gain non-repudiation SR
2.	Plan for conflicts and conflict resolution in term of non-repudiation SR
3.	Define standard templates for describing non-repudiation SR
4.	Use languages simply and concisely to explain non-repudiation SR
5.	Check that non-repudiation SR meets your standard
6.	Define change management policies for non-repudiation SR

2.9 Intrusion Detection Security Requirements Practices

No.	Practices
1.	Define operational processes to gain intrusion detection SR
2.	Use interaction matrices to find conflicts and overlaps in term of intrusion detection SR
3.	Define standard templates for describing intrusion detection SR
4.	Use languages simply and concisely to explain intrusion detection SR
5.	Check that intrusion detection SR meets your standard
6.	Define change management policies for intrusion detection SR

2.10 System Maintenance Security Requirements Practices

No.	Practices
1.	Use scenarios to elicit system maintenance SR
2.	Define system boundaries in term of system maintenance SR
3.	Define standard templates for describing system maintenance SR
4.	Use languages simply and concisely to explain system maintenance SR
5.	Check that system maintenance SR meets your standard
6.	Define change management policies for system maintenance SR

2.11 Secure Auditing Security Requirements Practices

No.	Practices
1.	Define operational processes in order to gain secure auditing SR
2.	Use checklists for secure auditing SR
3.	Assess security requirement risks to support secure auditing SR
4.	Define standard templates for describing secure auditing SR
5.	Use languages simply and concisely to explain secure auditing SR
6.	Check that secure auditing SR meets your standard
7.	Define change management policies for secure auditing SR

2.12 Survivability Security Requirements Practices

No.	Practices
1.	Define the system's operation environment to gain survivability SR
2.	Assess system feasibility in term of survivability SR
3.	Plan for conflicts and conflict resolution in term of survivability SR
4.	Assess survivability SR risk
5.	Define standard templates for describing survivability SR
6.	Use languages simply and concisely to explain survivability SR
7.	Check that survivability SR meets your standard
8.	Define change management policies for survivability SR

3. Appendix 3: Case Study Feedback form

Please copy this check list icon to your answer (✓)

	Very High	High	Neutral	Low	Very Low
How do you rate your knowledge security requirement engineering?					
How do you rate your practical experience of security requirement engineering?					

3.1 Ease of Learning

No.	Question	Strongly Agree	Agree	Disagree	Strongly Disagree	Not Sure
1.	SRERM representation is very clear					
2.	A little knowledge of security requirement engineering is required to learn how to use SRERM					
3.	It is easy to understand the practices designed for each security requirement component					
4.	It is easy to understand the assessment method					
5.	It is easy to use the SRERM to assess organizations readiness for security requirement					

	engineering.					
6.	It is easy to use distribution of security requirement among different levels, e.g. Identification, Authentication, and Authorization in Initial Level					
7.	Some training needs to be provided for the use of SRERM					

8.	How confident are you in the ratings that you have made in this section	Very	Confident	Little Confident	Not confident at all	No sure

3.2 User Satisfaction

No.	Question	Strongly Agree	Agree	Disagree	Strongly Disagree	Not Sure
9.	SRERM is general and can be applied to most companies					
10.	Each individual practice is easy to understand and unambiguous					
11.	Using the SRERM would identify strong and weak areas in the company regarding security requirement engineering					
12.	Using the SRERM would improve our security requirement engineering					
13.	If the SRERM were available for my job, I predict that I would use it on regular basis in the future.					
14.	I am satisfied and agreed with the readiness issues identified by SRERM.					
15.	It is important to implement SRERM in the form of an automated software tool to facilitate security requirement					

	engineering in assessing organization's readiness.					
16.	The SRERM is self-contained					
17.	The SRERM is a useful readiness tool for security requirement engineering.					
18.	The assessment method is useful.					

		Very	Confident	Little Confident	Not confident at all	No sure
19.	How confident are you in the ratings that you have made in this section					

3.3 Structure of SRERM

No.	Question	Strongly Agree	Agree	Disagree	Strongly Disagree	Not Sure
20.	All the components of the SRERM are self-explanatory and require no further explanation to be used effectively					
21.	The components of the SRERM are practical and are applicable in security requirement engineering process					
22.	The SRERM can be used effectively to identify security requirement engineering readiness issues with a goal of increasing organizations readiness for security requirement engineering.					
23.	The distribution of security component among different readiness levels (e.g. identification, authentication, and authorization) is useful					

24.	The 4 readiness levels of SRERM are useful					
-----	--	--	--	--	--	--

25.	How confident are you in the ratings that you have made in this section	Very	Confident	Little Confident	Not confident at all	No sure

26. Are there any modifications or improvements to the SRERM that you may suggest?

27. Are there any components that you may suggest adding to the SRERM in the future, please also give the reasons?

28. Please provide any comments relating to the assessment method.

29. Please provide any comments relating to the distribution of practices across various security requirement practices.

30. Please provide any comments relating to the usability of SRERM with respect to time it takes users to measure vendors' readiness.

Vitae

Name : Yusuf Mufti Muzammil (alias Yusuf Mufti)

Nationality : Indonesia

Date of Birth : 1/15/1989

Email : yusufmufti@outlook.com

Address : Desa Kuwayuhan RT 01/05, Pejagoan, Kebumen, Jawa
Tengah, Indonesia, Postal code 54361.

Academic Background : Bachelor of Informatic Engineering, UIN Sunan
Kalijaga, Yogyakarta, Indonesia

Experiences : 1. Top 10 Winners in Prototype Competition at
KFUPM 50K Challenge, 2016

2. Top 10 Winners in Business Model Competition at
Sixth Student Conference of KSA, 2015

3. Author of book Apps Development use Google Map
Android, published in Indonesia language, 2015

4. Part time job in ITC KFUPM as IT Support, CCSE
KFUPM as Oracle web service developer, ICS
Department as a grader of Software Requirements
course.