A Preliminary Structure of Software Security Assurance Model

Rafiq Ahmad Khan (author)
Software Engineering Research Group, Department of Computer Science & IT, University of Malakand, Khyber Paktunkhwa, Pakistan
rafiqahmadk@gmail.com

Siffat Ullah Khan (supervisor)
Software Engineering Research Group, Department of Computer Science & IT, University of Malakand, Khyber Pakhtunkhwa, Pakistan
siffatullah@uom.edu.pk

ABSTRACT

Software security is an important aspect that needs to be considered during the entire software development life cycle (SDLC). Integrating software security at each phase of SDLC has become an urgent need. To address software security, various approaches, techniques, methods, practices, and models have been proposed and developed. However, recent research shows that many software development methodologies do not explicitly include methods for incorporating software security during the development of software as it evolves from requirements engineering to its final disposal. The primary objective of this research is to study the state-of-the-art of security in the context of SDLC by following systematic mapping study (SMS). In the second phase, we will identify, through systematic literature review (SLR) and empirical study in the industry, the software security contributions, security challenges and their practices for global software development (GSD) vendors. The ultimate aim is to develop a Software Security Assurance Model (SSAM) to assist GSD vendor organisations in measuring their readiness towards the development of secure software.

KEYWORDS

Software Security, Software Development Life Cycle, Global Software Development, Vendors, Systematic mapping study, Systematic literature review, Empirical Study, Case study

1. INTRODUCTION

Nowadays software systems are becoming an important part in every domain of human society, such as electronics, telecommunications, shipping, home appliances, financial services, and more. Software security is the most important part of a system. Software security is the idea of engineering software so that it maintain to function properly under malicious attack [1]. In this emerging world, the most widespread security risks are encountered due to Internet-enabled software applications, and software are constantly in need of extensibility that may lead to complexity which adding up more fuel to the fire [2, 3].

Software engineers need to develop tools to support and manage the wide range of contradictions that facing during software development. These tools require to supply software engineers with behaviors to describe, notice, trace, present, interact and resolve complex contradictions between diverse visions of software artifacts, different developers and at different stages of software development [2].

Today, major problems in software security are Internet-enabled software applications, execution of bugs in the form of buffer overflow and inconsistent error handling [3, 4]. Every day, millions of people perform transactions through various applications such as Internet, ATM, mobile phone, email etc. Software are used by people bearing in mind that it is reliable and can be trusted upon and the operations they perform is secure. But if this software has ensemble security gaps, then how can they be considered secure? Security of software has become an integral part of everyday life. Because of the restriction of budget and release time of the software into market, many developers think security as an afterthought issue, thus resulting in poor quality of software [5]. In early days, software security was only be taken as a part of software testing, but later on, it has been proven that security is not afterthought issue and it is important to consider how developers can integrate security in the early phases of SDLC [6].

As a result, this research seeks to look at the software development process, from the viewpoint of each development stage of SDLC, and search to determine important security measures that must be employed throughout each phase to make highly secure applications.

We have organized this paper in different sections such as, Section 2 presents the background. In Section 3 the related limitations of software security in the context of SDLC have been presented. Objectives and research questions are discussed in Section 4. The research methodology is explained in Section 5. In Section 6, we have presented our proposed software security assurance model (SSAM). Finally, in Section 7 the conclusion and future work are presented.

2. BACKGROUND

The primary goal of the development of software is to manufacture high quality software within time and budget. The whole process of software development looks like a road map which directs project managers about the requirements to effectively complete the project [7].

To improve software application security, different security models and research have been conducted in recent decades in the field of software engineering, such models are MS DLC, BSIMM, and OWASPs SAMM [8-10]. Xinwen et al. [11] construct a static structure and dynamic behavior model for embedded software confidentiality and integrity by using Z/EVES tool and formal

ACM ISBN 978-1-4503-5717-3/18/05…$15.00
https://doi.org/10.1145/3196369.3196385

methods. This model provides semantics for the validation of embedded software security properties to assist in discovering early design errors and reduce the cost of testing and maintenance. Software security is a difficult task and it becomes even more tedious when developers have to develop, deploy or conscious of holding the security requirements of a specific organization [12]. To handle such type of problems, the static code analysis tools and techniques should be used to avoid software security challenges [12].

Idelia and Iskra [13] have proposed an approach for improving the security in SDLC in process control by using UML/SysML. This approach provides comprehensive stability in the syntax and underlying semantics; raises the possible likelihood of reuse; supports the whole SDLC in the field of process control from the requirements phase to the software implementation.

In order to better utilize the management of a software project, there is a need to focus on software risk management, because risk management encompasses people, software, hardware, cost, technology, and schedule [14]. Zhuobing et al. [25] present a software security assessment model, which gives a systematic approach to examine software security issues in three dimensions: technology, management, and engineering.

3. SECURITY ISSUES IN THE CONTEXT OF SDLC

The primary reason behind this study is due to the issues that are faced by vendor organisations at each phase of SDLC. A number of security issues have been reported in the literature, which are summarized as follow:

There is lack of defining security techniques, methods, models, processes, or may be solutions that provide security in each phase of SDLC [2].

Security is generally defined as non-functional requirement and due to this reason the security checks are normally applied during the final phase of the software development [5, 6]. However, attentions should be paid to software security at the early phase of the software development [5, 6].

Regardless of emergent attention in this area, there is need of systematic literature review to be conducted which covers the identification of challenges/risks, security contributions and its practices to develop a secure software. To address these, we plan to develop a model, SSAM which covers the research questions given in Section 4.

4. OBJECTIVES AND RESEARCH QUESTIONS

We aims to review the literature systematically to identify the state-of-the-art of software security to be considered by GSD vendor organisations during the development of a secure software as it evolves from requirements engineering to its final disposal. In order to improve security processes in the context of SDLC, we will develop Software Security Assurance Model (SSAM) to assist vendor organisations in measuring their readiness towards the development of secure software. The expected outcomes of this research will be a list of security measurements and their solutions to be incorporated by vendor organisations in each phase of the SDLC.

To achieve the afore mentioned objectives, our main aim is to tackle the following research questions (RQs):

RQ1: What is the state-of-the-art in secure software engineering (SSE)?

RQ2: What are the security contributions, as identified in the literature/industrial survey, to be pursued to develop secure software by Global software development (GSD) vendor organisations?

RQ3: What are the security risks, as identified in the literature/industrial survey, to be avoided to develop secure software by GSD vendor organisations?

RQ4: What are the best practices, as identified in the literature/industrial survey, to be adopted to develop secure software by GSD vendor organisations?

RQ5: Does the proposed SSAM model practically robust to assist software development vendor organisations in assessing their readiness towards the development of secure software?

5. RESEARCH METHODOLOGY

In this research, we plan to use various research methods as discussed in the following subsections:

5.1 Systematic Mapping Study

In the first phase of this research study, we plan to conduct Systematic Mapping Study to examine the state-of-the-art in the area of software security in the context of SDLC. Systematic mapping study (SMS) is a research strategy that is conducted to review topics in a broader sense and categorize the basic research articles in a specific area of interest [15, 16].

As compared to systematic literature review (SLR), systematic mapping study is conducted on a broader research questions in order to identify the gaps in a particular research domain. Therefore, SMS preserves huge significance to the researchers by giving a general idea about the literature in particular area [15].

The overreaching aim of the SMS in our research is to examine the state-of-the-art in the area of software security in the context of SDLC and also to capture the needs and directions for future research.

We designed a search string, given as follow, to examine the state-of-the-art in the area of software security in the context of GSD, however the results retrieved through different digital libraries, as shown in Table 1, were limited. We entitled this search string as Track 1.

Track 1: ((“software security” OR “software privacy” OR “secure software” OR “software protection” OR “software safety”) AND (“global software development” OR “GSD” OR “Distributed software development”))

Table 1: Search String Results per Database

S. No	Digital Libraries	Track 1 Search results	Track 2 Search results	Total Results
1	IEEE Xplore	14	1,759	1,773
2	Science Direct	14	599	613
3	ACM	26	375	401
4	Springer Link	17	1,656	1,673
5	Wiley Online Library	5	369	374
6	AIS Electronic Library (AiSel)	2	123	125
7	Google Scholar (Search Engine)	49	2,570	2,619
	Total	127	7, 451	7,578

We then decided to design another search string by naming it Track 2, given as follow, such as to examine the state-of-the-art in the area of software security in the context of SDLC, without restricting it to the GSD context. We therefore got significant results through different digital libraries, as shown in Table 1. We also presented the results of Track 2 to the members of the software engineering research group at University of Malakand (SERG_UOM), and it was concluded after a thorough discussion to follow and implement Track 2 for the conduction of the SLR, as shown in Table 1. Further a similar approach has been used by other researchers [25].

Track 2: ((“software security” OR “software privacy” OR “secure software” OR “software protection” OR “software safety”) AND (“Software Engineering” OR “Software Development lifecycle” OR “SDLC” OR “Software security Model”))

5.2 Systematic Literature Review

In the 2nd phase we plan to conduct systematic literature review (SLR). This is because SLR provides high level of strength in findings than ordinary literature reviews [19, 20]. SLR process provides a way to thoroughly estimating and understanding all the exiting research relevant to a specific research domain, question or occurrence of importance [17, 18]. The SLR is intended to identify the security contribution, risks and its best practices to reduce the risks in the development of secure software application, from the existing literature.

5.3 Empirical Study

In the third phase of our research an empirical study (questionnaire survey) will be conducted in GSD industry. We will use questionnaire survey to validate the SLR findings, concerning the security contributions, risks, and solutions, from GSD industry practitioners/experts whether these findings are applicable in the context of GSD or not. Additionally, in this phase we will find any new security contributions, risks, and solutions/practices in addition to the SLR findings.

The Software Security Assurance Model (SSAM) will be built based on the findings from the SLR and empirical study in the industry. Other researchers [22, 23] have also used a similar approach.

5.4 Case Study

Case study is an appropriate research methodology in the field of Software Engineering, because in natural circumstances it studies up-to-date phenomena [21, 24]. Case study is considered as the most powerful tool for the validation in Empirical Software Engineering [24]. We will use case study approach in the last stage of our research for validation of our proposed model (SSAM). For the validation of the SSAM, five case studies will be conducted at GSD vendor organisations. A feedback session will be arranged with the participants of the case studies, in order to get feedback about the SSAM.

6. PRELIMINARY STRUCTURE OF SOFTWARE SECURITY ASSURANCE MODEL

In order to identify, analyze, and mitigate security risks and to develop secure software, we have proposed a Software Security Assurance Model (SSAM). The proposed model will be developed in seven stages/phases as shown in Fig. 1 and the preliminary structure of the proposed model is presented in Fig. 2. As depicted in Fig. 1, the first stage/phase in the development of our proposed model is to conduct SMS in the area of software engineering to study the state-of-the-art of security in software engineering. This stage will give us direction about different sub-topics in the context of SDLC. In the second stage we will use Systematic Literature Review (SLR) process for identifying the security contributions, risks, and practices in the context of SDLC. In the 3rd stage/phase, empirical study (questionnaire survey) will be conducted in the GSD industry for the validation of the SLR findings and to find some new software security contributions, risks, and practices.

In the fourth stage, we will quantify the intensity (high, medium, low) of security contributions and risks by using mathematical models or formulas. In the fifth stage, we will develop different levels in SSAM by using the output of stage (2, 3, 4), supervisor inputs and guidance from existing models (CMMI, SAMM, MS DLC, BSIMM, SOP, SOVRM). In the sixth stage, we will validate our proposed model by conducting five case studies in GSD vendor companies. If the company gives some suggestions then we will mold the stage 5 accordingly. The software security assurance model will appear is a framework or assessment tool in the last stage. The preliminary structure of our propose model is shown in Fig. 2.

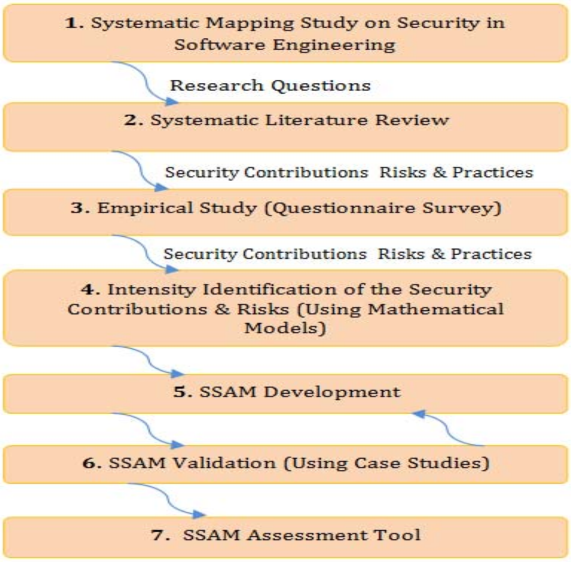
1. Systematic Mapping Study on Security in Software Engineering
Research Questions
2. Systematic Literature Review
Security Contributions Risks & Practices
3. Empirical Study (Questionnaire Survey)
Security Contributions Risks & Practices
4. Intensity Identification of the Security Contributions & Risks (Using Mathematical Models)
5. SSAM Development
6. SSAM Validation (Using Case Studies)
7. SSAM Assessment Tool

Figure 1: SSAM Development Stages

7. CONCLUSION AND FUTURE WORK

The probable outcome of this research will be a Software Security Assurance Model (SSAM) to assist GSD vendor organisations in measuring their readiness towards the development of secure software. This research will provide guidance about software security to be considered by GSD vendor organisations during the development of secure software as it evolves from requirements engineering to its final disposal. The SSAM is expected to assist software development vendor organisations in gauging their readiness towards to development of secure software. The model will generate different assessment reports such as a list of security measurements and their solutions to be incorporated by GSD vendor organisations in each phase of the SDLC.

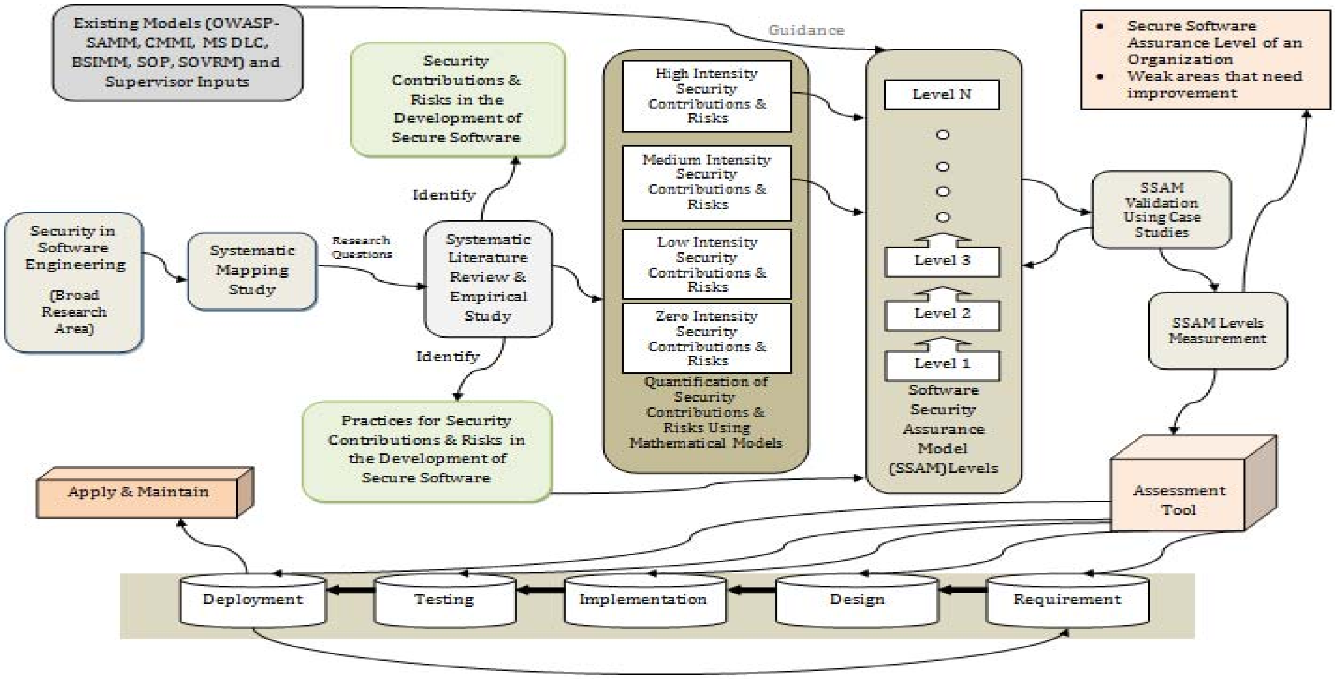
Existing Models (OWASP-SAMM, CMMI, MS DLC, BSIMM, SOP, SOVRM) and Supervisor Inputs
Guidance
Secure Software Assurance Level of an Organization
Weak areas that need improvement
Security Contributions & Risks in the Development of Secure Software
High Intensity Security Contributions & Risks
Medium Intensity Security Contributions & Risks
Low Intensity Security Contributions & Risks
Zero Intensity Security Contributions & Risks
Quantification of Security Contributions & Risks Using Mathematical Models
Level N
Level 3
Level 2
Level 1
Software Security Assurance Model (SSAM) Levels
Identify
Security in Software Engineering (Broad Research Area)
Systematic Mapping Study
Research Questions
Systematic Literature Review & Empirical Study
SSAM Validation Using Case Studies
SSAM Levels Measurement
Identify
Practices for Security Contributions & Risks in the Development of Secure Software
Assessment Tool
Apply & Maintain
Deployment
Testing
Implementation
Design
Requirement

Figure 2: Preliminary Structure of SSAM

ACKNOWLEDGMENT

The primary author (Rafiq Ahmad Khan) is thankful to his supervisor (Dr. Siffat Ullah Khan) for providing support and guidance at every step in the development of the proposed SSAM model.

REFERENCES:

[1] Z. Yan, “The performance cost of software obfuscation for Android applications,” Computers & Security, vol. 73, pp. 57-72, 2018.

[2] M. Phillips. Dewanne, A. Mazzuchi. Thomas, and S. Shahram, “An architecture, system engineering, and acquisition approach for space system software resiliency,” Information and Software Technology, vol. 94, pp. 150-164, 2018.

[3] Ammar. Mahmoud, Russello. Giovanni, and C. Bruno, “Internet of Things: A survey on the security of IoT frameworks,” Journal of Information Security and Applications, vol. 38, pp. 8-27, 2018.

[4] G. McGraw, “Software security,” IEEE Security & Privacy, vol. 2, no. 2, pp. 80 - 83, 2004.

[5] Sharma. Anuradha, and M. P. Kumar, “Aspects of Enhancing Security in Software Development Life Cycle,” Advances in Computational Sciences and Technology, vol. 10, no. 2, pp. 203-210, 2017.

[6] Abdul-Karim. Nor. Shahriza, Albuolayan. Arwa, Saba. Tanzila, and R. Amjad, “The practice of secure software development in SDLC:an investigation through existing model and a casestudy,” Security and Communication Network, vol. 9, no. 18, pp. 5333–5345, 2016.

[7] Roger. S. Pressman, and B. R. Maxim, Software Engineering: A Practitioner's Approach, Eight ed., pp 1-977, 2 Penn Plaza, New York, McGraw-Hill Education, 2014.

[8] C. Pravir, Software assurance maturity model: A guide to building security into software development, Version 1.0 ed., pp 1-96: OWASP Foundation, 2017.

[9] I. A. Tondel, M. G. Jaatun, and P. H. Meland, “Security Requirements for the Rest of Us: A Survey,” IEEE Software, vol. 25, no. 1, pp 20-27, 2008.

[10 Teodoro, Nuno, and C. Serrao, “Web application security: Improving critical web-based applications quality through in-depth security analysis,” in International Conference on Information Society (i-Society), London, UK, pp 457-462, 2011.

[11] Hu. Xinwen, Yi. Zhuang, Cao. Zining, Ye. Tong, and L. Mi, “Modeling and validation for embedded software confidentiality and integrity,” in 12th International Conference on Intelligent Systems and Knowledge Engineering (ISKE), Nanjing, China, pp 1-6, 2018,.

[12] Zhioua. Zeineb, Short. Stuart, and R. Yves, “Static Code Analysis for Software Security Verification: Problems and Approaches,” in EEE 38th International on Computer Software and Applications Conference Workshops (COMPSACW), Vasteras, Sweden, pp. 102-109, 2014.

[13] Batchkova. Idilia, and A. Iskra, “Improving the software development life cycle in process control using UML/SysML,” IFAC Proceedings Volumes, vol. 44, no. 1, pp. 14133-14138, 2011.

[14] Kumar. Chandan, and Y. D. Kumar, “A Probabilistic Software Risk Assessment and Estimation Model for Software Projects,” Procedia Computer Science, vol. 54, pp. 353-361, 2015.

[15] Kitchenham. Barbara , David. Budgen, and O. P. Brereton, “Using mapping studies as the basis for further research – A participant-observer case study,” Information and Software Technology, vol. 53, pp. 638–651, 2011.

[16] Petersen. Kai, Vakkalanka. Sairam, and K. Ludwik, “Guidelines for conducting systematic mapping studies in software engineering: An update,” Information and Software Technology, vol. 64, pp. 1–18, 2015.

[17] Kitchenham. Barbara, and S. Charters, Guidelines for Performing Systematic Literature Reviews in Software Engineering, Version 2.3, EBSE Technical Report, pp. 1-65, 2007.

[18] Velásquez. Ignacio, Caro. Angélica, and R. Alfonso, “Authentication schemes and methods: A systematic literature review,” Information and Software Technology, vol. 94, pp. 30-37, 2018.

[19] C. Lacerda. Thaísa, and W. G. Von, “Systematic literature review of usability capability/maturity models,” Computer Standards & Interfaces, vol. 55, pp. 95-105, 2018.

[20] Staples. Mark, and Niazi. Mahmood, “Experiences using systematic review guidelines,” Journal of Systems and Software, vol. 80, no. 9, pp. 1425-1437, 2007.

[21] Runeson. Per, and H. Martin, “Guidelines for conducting and reporting case study research in software engineering,” Empirical Software Engineering, vol. 14, pp. 131–164, 2009.

[22] Khan. Siffat. Ullah, Niazi. Mahmood, and A. Rashid, “A readiness model for software development outsourcing vendors,” in IEEE International Conference on Global Software Engineering, pp. 273-277, 2008.

[23] Ali. Sikandar, and Khan. Siffat. Ullah, “Software outsourcing partnership model: An evaluation framework for vendor organizations,” Journal of Systems and Software, vol. 117, pp. 402-425, 2016.

[24] Šmite. Darja, Wohlin. Claes, Gorschek. Tony, and F. Robert, “Empirical evidence in global software engineering: a systematic review,” Empirical Software Engineering, vol. 15, no. 1, pp. 91–118, 2010.

[25] Rashid. Nasir, and Khan. Siffat. Ullah, "Using agile methods for the development of green and sustainable software: Success factors for GSD vendors," Journal of Software: Evolution and Process, pp. 1-28, 2017.