The European Proceedings of
Social & Behavioural Sciences
EpSBS

ISSN: 2357-1330

https://doi.org/10.15405/epsbs.2019.05.02.36

AIMC 2018
Asia International Multidisciplinary Conference

SYSTEMATIC MAPPING STUDY PROTOCOL FOR SECURE SOFTWARE ENGINEERING

Rafiq Ahmad Khan (a)*, Siffat Ullah Khan (b), Mohd Yazid Idris (c)
*Corresponding author

(a) Software Engineering Research Group, Department of Computer Science & IT, University of Malakand, Khyber Pakhtunkhwa, Pakistan, rafiqahmadk@gmail.com
(b) Software Engineering Research Group, Department of Computer Science & IT, University of Malakand, Khyber Pakhtunkhwa, Pakistan, siffatullah@uom.edu.pk
(c) Department of Software Engineering, Faculty of Computing, Universiti Teknologi Malaysia, yazid@utm.my

Abstract

In today's world, software security has become essential for protecting the overall organization's operations. Mostly software development companies are adopting various strategies to build secure software to cope with the challenges of the client organization. Information become threatened due to connection with the cyber world, and it requires better security mechanisms. Software development organizations are receiving pressure from their clients to focus on the enhancement of security during the whole software development life cycle (SDLC). This protocol aims to review the literature in a systematic way to identify the state-of-the-art of software security to be considered by Global Software Development (GSD) vendor organizations 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 software development, our current research developed a Systematic Mapping Study (SMS) protocol. Presently, we are in the implementation phase of the protocol for the development of secure software. The expected outcomes of this SMS will be a list of security measurements and their solutions to be incorporated by GSD vendor organizations in each phase of the SDLC. This will also give a direction for new research in this area.

Keywords: Software security, secure software, SDLC; systematic mapping study, global software development, vendors.

https://doi.org/10.15405/epsbs.2019.05.02.36
Corresponding Author: Rafiq Ahmad Khan
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

1. Introduction

Due to the advancement of information and communication technologies (ICTs), software security is becoming more and more critical. 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. For developing any software system, the main important part is to consider its security. Software security is the most important part of a system. Literature reveals that a number of strategies have been developed for building secure software, however no such tools are adapted to handle the security of a complex environment, generally at each phase of SDLC and in large distributed software development teams (Pan, Zhu, & Mao, 2016; Tung, Lo, Shih, & Lin, 2016).

The critical question is arising that how could a software will be secure to security threats? Obviously, majority of software products undergoing live testing are vulnerable to threats and mostly fail to provide a secure and safe environment to clients and users. This is due to the lack of systematic evaluations such as systematic reviews, procedures, approaches, or frameworks as these evaluations could help project managers and software engineers to ensure that security processes are continuously followed throughout the software development process, according to a set of predefined procedures or rules (Karim, Albuolayan, Saba, & Rehman, 2016; Mundher, Muhamad, Rehman, Saba, & Kausar, 2014). To efficiently grip the security problems, that are present in during the development of applications, it is necessary to consider security-minded thinking throughout the development processes, which reduces the threats of missing essential security requirements or creating vital faults in software design (Mohammed, Niazi, Alshayeb, & Mahmood, 2017).

Traditionally, most software developers ignore taking care of early stage vulnerabilities and security threats. These defects in software are rising due to poor development approach, bad design and bad requirement analysis, which can lead to easy exploitation by the cybercriminal (Garousi, Coşkunçay, Demirörs, & Yazici, 2016). Security in software development life cycle are mostly treated by adding security features, using firewalls by security experts, imposition avoidance systems, proxies, platform security and antivirus (Jürjens, 2005). Today, the most widespread security risks are encountered due to Internet enabled software applications, and software are rapidly growing extensibility and complexity which adding up more fuel to the fire. As a result of every assessment, security risks are very common in software applications and the problems are rising day-by-day (McGraw, 2004).

Software Security plays an essential role in the software development life cycle, and software security testing is very significant means to achieve the goal of building secure software. Considerable research has been focused on software security in the context of SDLC. In order to develop a secure software and improve software development processes, several standards and models have been initiated:

A systematic mapping study performed by (Mohammed, Niazi, Alshayeb, & Mahmood, February 2017), has identified various approaches that are used for detection of security defects in SDLC. After thorough analysis they argued that dynamic code analysis and static code analysis are the most frequently used approaches for checking security risks mainly in coding phase. The study shows that most of the security checks are approximately applied only to coding stage of SDLC. In order to construct a secure software, (Tung et al., 2016), suggested an integrated security testing framework, which assume security activities and practices of SDLC to generate security guidelines and test cases.

https://doi.org/10.15405/epsbs.2019.05.02.36
Corresponding Author: Rafiq Ahmad Khan
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

Velmourougan, Dhavachelvan, Baskaran, and Ravikumar, (2014), presents a maintainable software development life cycle model (MSDLC), which establishes different maintainability tasks to be followed during the software development life cycle phases. They propose various actions and practices to be incorporated at different stages of SDLC. (Hashimi, Hafez, and Beraka, (2012) have determined risk management processes and related techniques, and have defined the critical risks that may face the software. To attain that, they intend a wheel model which shows the relation among phases of SDLC and risk management. Gilliam, Wolfe, Sherif, and Bishop, (2003), have worked on the SDLC and proposed software development security checklists. They proposed 26 software security checklists that should be followed during the critical areas of requirements collection and its specification, design and code issues, maintenance and decommissioning of software and systems. As in traditional SDLC, security testing is done at the end however, it should consider security and risks at each phase to avoid major threats at the later phases.

The Software Assurance Maturity Model (SAMM) is an open framework to facilitate organizations in assuring the organization's maturity towards software security (Chandra, 2009). This model was defined with flexibility in mind which could be developed by small, medium, and large organizations using any approach of development. (Karim et al., 2016) conducted a case study and have argued to introduce security measures at the early stages in the SDLC. They present different essential basics as a security standards, practices, approaches, tools and techniques used within SDLC. In this manner, they gathered various recommendations and verifications to obtain the real activities that are suitable to be performed at each stage of SDLC. Recently, complicated and inventive attacks have been reported which are constantly increasing software security risks (Bedi, Gandotra, Singhal, Narang, & Sharma, 2013). Bedi et al., (2013) proposed a three phased threat-oriented security model, gathering complicated and inventive attacks as a component, and then suggesting practices to handle it. Felderer, Zech, Breu, Büchler, and Pretschner, (2016), systematically extracted a categorization of testing approaches for model based security. This model contains filter criterion (i.e. model of system security, security model of the environment and precise test collection criteria) and indication criteria (i.e. maturity of evaluated system, evidence measures and evidence level).

2. Problem Statement

From the above literature in the field of Software Engineering, we overreach to this point that there is a growing interest and need of research in software security in the context of SDLC. Regardless of emergent attention in this area, there is lack of systematic literature review or systematic mapping study to be conducted which covers the identification of challenges/risks, security configurations and its practices to develop a secure software for Global Software Development (GSD) vendors. It is important to note that the limit of our systematic analysis is constrained to the area of software security in the context of SDLC, which is defined in our search plan and terms.

https://doi.org/10.15405/epsbs.2019.05.02.36
Corresponding Author: Rafiq Ahmad Khan
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

3. Research Questions

The overreaching goal of this systematic mapping study 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. To achieve this, our main aim is tackle the following research questions (RQs):

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

In order to answer RQ1, we will analyze the literature on the basis of the following sub questions:

RQ1.1: Which topics/techniques/challenges that are related to secure software engineering are covered?

RQ1.2: What are the solutions that should be followed at each phase of the SDLC to deliver a secure software?

4. Purpose of the Study

The overreaching goal of this systematic mapping study 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

5. Research Methods

In this paper, a systematic mapping study protocol is discussed. We have followed the SMS guidelines (Budgen, Turner, Brereton, & Kitchenham, 2008) for developing this protocol. Protocol development is the first phase of a system mapping study. SMS reviews topics in a broader sense and categorize the basic research articles in a specific area of interest (Kitchenham, Budgen, & Brereton, 2011). 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 preserve huge approaching significance to the field of software engineering researchers by giving a general idea about the literature in particular area.

5.1. Search Strategy

The first step for building the search string, the PICO criteria have been reported in the literature (Budgen et al., 2008). PICO (Population, Intervention, Comparison, and Outcomes) to make out basic keywords and prepare search strings from the research questions.

Population: Software Security, Software Development Life Cycle (SDLC), Global Software Development

Interventions: Software security strategies/techniques/models, solutions

Comparison: The current study do not perform any comparison

Outcomes: Secure software

5.2. Study Selection and Quality Assessment

We will select and include the articles based on title, abstracts, and those which have full text reading are available, as well follow the quality assessment criteria. We are doing this because to retrieve a set of

https://doi.org/10.15405/epsbs.2019.05.02.36
Corresponding Author: Rafiq Ahmad Khan
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

relevant papers based on applying the inclusion and exclusion criteria. We will apply the following inclusion and exclusion criteria to the titles and abstracts of the relevant papers:

- Papers in the area of software security
- Papers related to software development life cycle (SDLC).

The following exclusion criteria will be applied to the relevant papers.

- The studies which are not written in English language.
- The studies that are not in the domain of Software Engineering.
- The studies which occurs several times in the final set.
- The studies whose full text are not available.
- The papers that are not published in any peer reviewed journal or conference proceedings.
- Books and magazine's articles will be excluded.

We will follow the following quality assessment criteria's as shown in Table 01, in order to included papers in the final selection.

Table 01. Quality Assessment Criteria's

S. No	Quality Assessment Criteria's	Choices
1	Are the findings and results are clearly stated in the study?	Yes = 1, No = 0
2	Are the findings based on empirical method?	Yes = 1, No = 0
3	Is the paper well referenced (cited) in Journals or conference proceedings?	Yes = 1, No = 0
4	Are the arguments well obtainable and justified?	Yes = 1, No = 0

5.3. Data Extraction

In this phase we will extract the data by studying the selected studies. In order to answer the research questions mention in section 3.1, we will study the full text of each selected paper. We will extract the following data from each study:

- Title, Year, and Author(s) of the paper
- First author's affiliation: Country, Institute
- Publisher: IEEE Xplore, Science Direct, ACM, and others
- Venue of the Paper: Journal/Conference/Workshop
- Area in Software Engineering
- Security checks in each stage of SDLC
- Software security approaches, tools, methods
- Practices for secure software
- Software security contributions and risks
- SDLC phase covered
- Company type: Small, medium, large
- Company level: National, multinational or both
- Research methodology used in the paper

https://doi.org/10.15405/epsbs.2019.05.02.36
Corresponding Author: Rafiq Ahmad Khan
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

5.4. Analysis and Classifications

The information for each data item extracted will be tabulated and given a theme during analysis. The papers relating to each theme will be analyzed and counted. All the extracted data will be saved in Microsoft Excel sheets. The data will be analyzed using statistical tool, SPSS.

5.5. Validation of the Protocol

The protocol was jointly developed by the authors and then presented to Software Engineering Research Group (SERG_UOM) for validation. The suggestions/review comments were incorporated and the protocol was revised accordingly.

6. Findings

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 02, were limited. We entitled this search string as Track 1. The search string was constructed on connecting the features of PICO by Boolean AND connector:

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”)

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 02. 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 02.

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”))

This search string was run in IEEE Xplore, Science Direct, ACM, Springer Link, Wiley Online Library, and AIS Electronic Library (AiSel) digital libraries. We also run this search string in Google Scholar an online search engine. Table 02 shows the number of search results per database and search engine.

Table 02. 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

https://doi.org/10.15405/epsbs.2019.05.02.36
Corresponding Author: Rafiq Ahmad Khan
Selection and peer-review under responsibility of the Organizing Committee of the conference
eISSN: 2357-1330

6	AIS Electronic Library (AiSel)	2	123	125
7	Google Scholar (Search Engine)	49	2,570	2,619
Total		127	7,451	7, 578

7. Conclusion

It is evident from the findings of the search phase of the systematic mapping study that no systematic mapping study or systematic literature review is published so far that can identify the GSD security challenges and its solutions at each phase of the SDLC. In this paper, we only present findings of the one component of our proposed study in the form of SMS protocol. Currently, we are in the implementation phase of the SMS protocol.

The ultimate aim in future, we plan the following in future:

- Identification of the software security challenges, security contributions and their practices for GSD vendors through SLR and empirical study in the industry.
- To develop Software Security Assurance Model (SSAM) to assist GSD vendor organizations in measuring their readiness towards the development of secure software.

Acknowledgments

We are very thankful to Software Engineering Research Group, University of Malakand (SERG_UOM) for their valuable feedbacks/reviews in validation of the SMS protocol. We are also grateful to COE Grant Universiti Teknologi Malaysia (Q.J130000.2428.03G94) for financial supporting.

References

Bedi, P., Gandotra, V., Singhal, A., Narang, H., & Sharma, S. (2013). Threat-oriented security framework in risk management using multiagent system. *Software: Practice and Experience, 43*(9), 1013-1038.

Budgen, D., Turner, M., Brereton, P., & Kitchenham, B. A. (2008). *Using Mapping Studies in Software Engineering.* Paper presented at the PPIG.

Chandra, P. (2009). Software assurance maturity model. *A guide to building security into software development v1. 0.*

Felderer, M., Zech, P., Breu, R., Büchler, M., & Pretschner, A. (2016). Model-based security testing: a taxonomy and systematic classification. *Software Testing, Verification and Reliability, 26*(2), 119-148.

Garousi, V., Coşkunçay, A., Demirörs, O., & Yazici, A. (2016). Cross-factor analysis of software engineering practices versus practitioner demographics: An exploratory study in Turkey. *Journal of Systems and Software, 111*, 49-73.

Gilliam, D. P., Wolfe, T. L., Sherif, J. S., & Bishop, M. (2003). *Software security checklist for the software life cycle.* Paper presented at the WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.

Hashimi, H., Hafez, A., & Beraka, M. (2012). *A novel view of risk management in software development life cycle.* Paper presented at the 2012 12th International Symposium on Pervasive Systems, Algorithms and Networks.

Jürjens, J. (2005). *Secure systems development with UML*: Springer Science & Business Media.

Karim, N. S. A., Albuolayan, A., Saba, T., & Rehman, A. (2016). The practice of secure software development in SDLC: an investigation through existing model and a case study. *Security and Communication Networks, 9*(18), 5333-5345.

Kitchenham, B. A., Budgen, D., & Brereton, O. P. (2011). Using mapping studies as the basis for further research–a participant-observer case study. *Information and Software Technology*, *53*(6), 638-651.

Mohammed, N. M., Niazi M., Alshayeb, M., Mahmood, S. (February 2017). Exploring Software Security Approaches in Software Development Lifecycle: A Systematic Mapping Study. *Computer Standards & Interfaces, 50*, 107-115.

McGraw, G. (2004). Software security. *IEEE Security & Privacy*, *2*(2), 80-83.

Mohammed, N. M., Niazi, M., Alshayeb, M., & Mahmood, S. (2017). Exploring software security approaches in software development lifecycle: A systematic mapping study. *Computer Standards & Interfaces*, *50*, 107-115.

Mundher, M., Muhamad, D., Rehman, A., Saba, T., & Kausar, F. (2014). Digital watermarking for images security using discrete slantlet transform. *Applied Mathematics & Information Sciences*, *8*(6), 2823.

Pan, P., Zhu, X., & Mao, X. (2016). *Security Test for Application Software Based on SPN*. Paper presented at the First International Conference on Real Time Intelligent Systems.

Tung, Y.-H., Lo, S.-C., Shih, J.-F., & Lin, H.-F. (2016). *An integrated security testing framework for Secure Software Development Life Cycle*. Paper presented at the 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS).

Velmourougan, S., Dhavachelvan, P., Baskaran, R., & Ravikumar, B. (2014). *Software development life cycle model to improve maintainability of software applications.* Paper presented at the 2014 Fourth International Conference on Advances in Computing and Communications.