Software Security in Open Source Development: A Systematic Literature Review

Shao-Fang Wen
Norwegian University of Science and Technology
Gjøvik, Norway
shao-fang.wen@ntnu.no

Abstract—**Despite the security community's emphasis on the importance of building secure open source software (OSS), the number of new vulnerabilities found in OSS is increasing. In addition, software security is about the people that develop and use those applications and how their vulnerable behaviors can lead to exploitation. This leads to a need for reiteration of software security studies for OSS developments to understand the existing security practices and the security weakness among them. In this paper, a systematic review method with a socio-technical analysis approach is applied to identify, extract and analyze the security studies conducted in the context of open source development. The findings include: (1) System verification is the most cited security area in OSS research; (2) The socio-technical perspective has not gained much attention in this research area; and (3) No research has been conducted focusing on the aspects of security knowledge management in OSS development.**

I. Introduction

It is indisputable that open source software (OSS) development has earned a key position standing in today's software engineering. Due to the uniqueness of the OSS model, the software security of OSS product has been widely discussed in security communities. However, the number of new vulnerabilities keeps increasing in today's OSS systems. According to the National Vulnerability Database (NVD), over 11,500 new vulnerabilities in OSS have been uncovered since 2012 [8]. These vulnerabilities open some of the most critical OSS projects to potential exploitation: Heartbleed and Logjam (in OpenSSL); Quadrooter (in Android); Glibc Vulnerability (in Linux servers and web frameworks); NetUSB (in Linux kernel), and many others [39, 51]. With increasing importance and complexity of OSS, the ineffective security practices to secure OSS development will result in more breaches that are serious in the future.

On the other hand, open source software is developed collectively by the online community of practices with a strong relationship between the technical and social interactions in a knowledge intensive process. There are unique characteristics of OSS, such as community-based distributed development, volunteer workers, on-line information exchange, and informal integration of new contributors. These characteristics contribute the high socio-technical complexity of OSS security, influence the applicability of software security practices in OSS development, and result in a need to manage the security practices and knowledge efficiently within the OSS communities. Moreover, the trustworthiness of the open source depends on socio-technical aspects of the software security practices [19], [23], [44], [66], which include the expertise of the developers in the communities to produce secure code, quality of tools used in development, the level of testing carried out before releasing the product, and the collaborative practices followed throughout the development cycle, etc. These aspects need careful investigation from a socio-technical perspective as well [37].

Many studies have been conducted by both researchers and practitioners on the mechanisms of building security in OSS development. The overarching objective of this research is to summarize what we know about these security studies and to offer suggestions for research in OSS security. In this research, we carried out a systematic review of the existing literature to identify and classify the software security practices in securing the software products that are developed by the open source communities. In addition, to investigate the security studies that are conducted in two aspects: socio-technical security and security knowledge management.

The rest of this paper is organized as follows. Section II describes the related works. The classification frameworks used in this SLR research is explained in section III. The research method is explained in section IV. Section V describes each step in selection execution. In section VI, we give an overview of the literature review results. Section VII provides a discussion based on the result. Section VIII states the limitation of the study. Finally, we describe the conclusion in section IX.

II. Related works

In the open source research, there are few examples of the literature review. Hauge et al. [32] seek to identify how organizations adopt OSS. They classified the literature according to the ways of adopting OSS and evaluated the research on adoption of OSS in organizations. Stol and Babar [57] aims to gain insights into the state of the practice of reporting empirical studies of OSS in order to identify the gaps to be filled for improving the quality of evidence being provided for OSS. Feller et al. [26] review 155 research papers to identify the kinds of open source project communities that have been researched and the kinds of research questions that have been asked.

In an introduction to a special issue, Scacchi et al. [56] provide an overview of the research on the development processes found in OSS projects. Crowston et al. [18] also present a quantitative summary of the literature of OSS development selected for the review and discuss findings of this literature categorized into issues pertaining to inputs, processes, emergent states, and outputs. Von Krogh and von Hippel [62] give an overview of some of the research on OSS and organize it into three categories: motivations of contributors, innovation processes, and competitive dynamics.

III. CLASSIFICATION FRAMEWORK

A. *Software security areas*

To identify the security practices in OSS development, we adopt OWASP Software Assurance Maturity Model (SAMM) [13] as the guidance of the classification. The foundation of the model is built upon the core business functions of software development with security practices tied to each (see Fig. 1). The building blocks of the model are the three maturity levels defined for each of the twelve security practices.

B. *Socio-technical perspectives*

Software development process is not purely a technical task, but also a social process embedded within organizational and cultural structures [31]. The socio-technical perspective provides a deeper analysis of the relationship between the methods, techniques, tools, development environment and organizational structure [20], [21].

Our research is based on the Socio-Technical System (STS) and the Security-By-Consensus model (SBC) developed by Kowalski [37]. The STS model is depicted in Fig. 2. This has two sub-systems include social aspects (culture and structures) and technical aspects (methods and machines). The SBC model is applied to define the detailed parts of STS subsystem controls, illustrated in Fig. 3.

IV. RESEARCH METHOD

The design of this literate review is based on the original guidelines of systematic literature review provided by Kitchenham [35], [36] while also being guided by other systematic literature review articles in the area of open source software, such as Crowston et al.[18] and Hauge et al. [32]. The steps of the review include definition of the research questions and the research protocol, conduct search for studies, screening of papers, data extraction, and data synthesis.

A. *Research questions*

The aim of this SLR is to understand and summarize the empirical proofs as regard the software security literatures in the context of open source development. In addition, to investigate the security studies that are conducted in two aspects: socio-technical security and security knowledge management. To achieve this aim, the research question addressed by our research is formulated as presented below:
RQ1: What research has been conducted on the security practices and behaviors in the context of OSS development?

RQ2: What research has been conducted on the socio-technical security aspects associated with OSS development?

RQ3: What research has been conducted focusing on aspects of security knowledge management in OSS development?

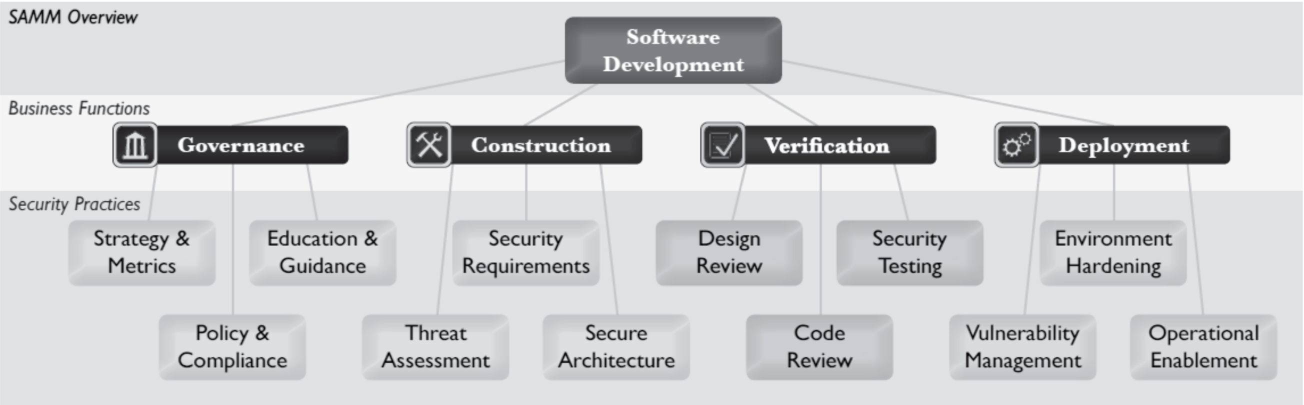

Fig. 1. Software Assurance Maturity Model (Chandra [13])

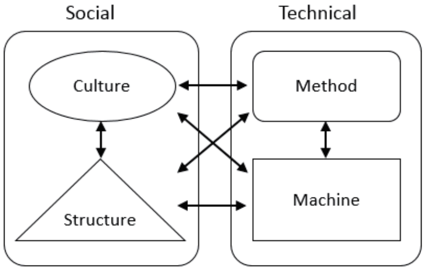

Fig. 2. Socio-technical system (Kowalski [37], page 10)

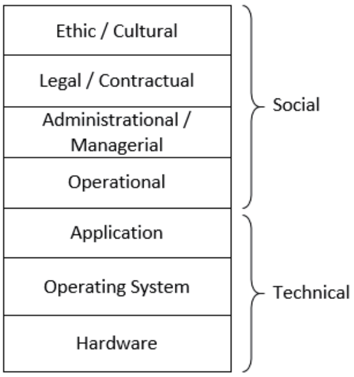

Fig 3. SBC Model (Kowalski [37], page 19)

B. *Search Strategy*

The search strategy is used to search for primary studies including search strings and resources to be searched. The detailed description of the search strategies utilized in this research as explained below:

1) Search term

To avoid overlooking relevant studies, all searches will be conducted using the combination of two categories of keywords in relation to “Open Source” (S1) and “Security” (S2), defined as follows:

- S1 is a string made of keywords related open source, such as “open source”, “free software”, “free/libre software”, “OSS”, “FOSS”, “FLOSS”.
- S2 is a string made up of keywords related to security, such as “security”, “secure”, “insecure”, “vulnerability”, “virus”, “malware”, “exploits”, “threat” and “hack”.

An example of a search done in the electronic data is described as follows:

(“security” OR “secure” OR “insecure” OR “vulnerability”) AND (“open source” OR “open-source” OR “free software” OR “free/libre software” OR “OSS” OR “FLOSS”)

2) Literature resources

Six primary electronic database resources were used to extract data for synchronizations in this research.

- ACM Digital Library (https://dl.acm.org).
- IEEExplore (http://ieeexplore.ieee.org).
- Springerlink (http://link.springer.com).
- Science Direct (http://www.sciencedirect.com).
- Scopus (https://www.scopus.com).
- Google Scholar (http://scholar.google.com/)

C. *Study Selection Criteria*

The main inclusion criterion for this study is to include the software security studies that have been conducted in the context of open source development. The literature published during 2000-2016 are taken into consideration for the inclusion in search criteria. The detail inclusion criteria included are:

- Studies that describe security practices of OSS development.Studies that investigate security issues of OSS development.
- Studies that discuss the socio-technical characteristics of OSS security.
- Studies that discuss knowledge issues of OSS security.

Articles on the following criteria are excluded

- Papers that are not written in English.
- Studies that do not focus explicitly in OSS context, such as making use of OSS repositories as the study reference.
- Studies that only address OSS security concepts, such as comparing open source and proprietary (closed) software, and the use of OSS.

Studies that focus on a specific open source platform or product.

V. SELECTION EXECUTION

The search on the digital libraries initially identified 2942 papers. The selection execution was composed by four filter stages as shown in Fig. 4. In stage 2, we individually reviewed the papers from the previous stage based on their titles and abstracts, and if necessary by skimming the full text and resulted in 167 papers. Next, in stage 3, to identify publications on security practices in OSS development, we individually went through the output of the second stage and evaluated the papers' topics by skimming the papers. Publications on the discussion of software security in open source were included, while those do not focus explicitly on software security (only refer to software security as a side topic) and OSS context (only make use of OSS project data as the study reference) were rejected. Moreover, papers that focus on examining specific platform without contributing to OSS development were also excluded. Through stage 3, we discarded 74 of the 167 papers and selected 93 papers for further analysis.

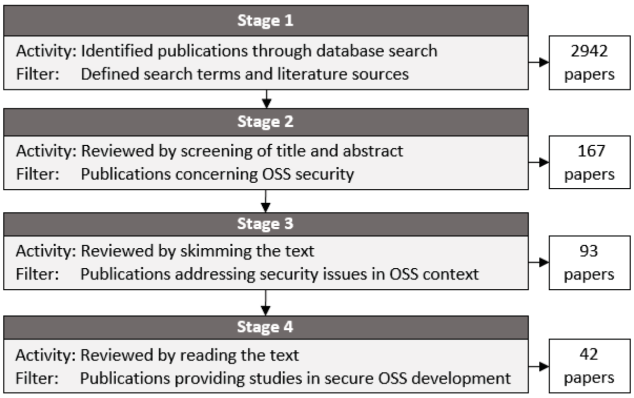
Stage 1
Activity: Identified publications through database search
Filter: Defined search terms and literature sources
2942 papers
Stage 2
Activity: Reviewed by screening of title and abstract
Filter: Publications concerning OSS security
167 papers
Stage 3
Activity: Reviewed by skimming the text
Filter: Publications addressing security issues in OSS context
93 papers
Stage 4
Activity: Reviewed by reading the text
Filter: Publications providing studies in secure OSS development
42 papers

Fig. 4. Paper screening process of SLR

Then we classified the publications from stage 3 into three categories: OSS concept where the authors discuss (debate) software security between open source and closed source, OSS adoption where authors present the security concerns in the use of OSS and OSS development. Of the 93 included papers, 27 were classified as open source concept papers, 24 as open source adoption paper, and 42 as OSS development papers. The OSS concept papers and OSS adoption papers may expand the understanding of OSS security issues but they are not providing any practical study to secure open source development. Hence, these papers were not included. Accordingly, the final stage of the review included 42 papers.

VI. RESULT

This section presents an overview of the selected studies.

A. *Publications by year*

The Table VI shows the results of the research sources that have been found during SLR. Fig. 4 illustrates the number of selected studies from the years 2000-2016. There are no significant studies related to our research topic in the year 2000 and 2001, and just a few papers were published between 2002 and 2005 (total five papers in four years). This results from most studies of open source security in this period focus on the general discussion, such as concepts of open source security and debate on open vs. closed source security, etc. instead of security practices in open source development. The highest number of publications happened in the year 2014 (6 papers).

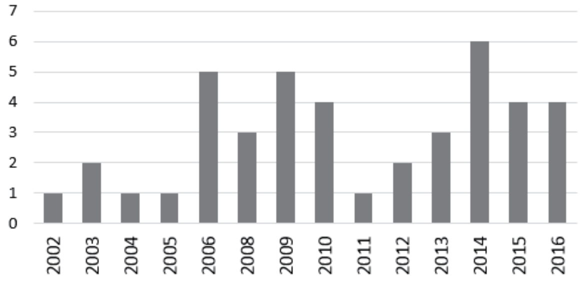

Fig. 5. Number of publications versus year

B. *Publication venues and sources types*

Table I presents the distribution of the studies' publication sources. Of the 42 studies, 70% (29 of them) were published in conferences, 16% (7 of them) in journals, 14% (6 of them) are distributed in books, thesis, and research white papers.

Table II presents the top five publication venues of some the selected studies, and the number of studies. Overall 34 publications venues are identified the cover different areas of computer science, such as software engineering, information system, and security, etc.; which means this study topic has received wide attention in the research community. One observation that can be made is that the leading publication venues are the type of conference proceedings, which are in the field of software engineering. This demonstrates the importance of OSS security research in software engineering and other related fields.

TABLE I. DISTRIBUTION OF STUDIES ACCORDING TO THE PUBLICATION VENUES

Type	Frequency	%
Conference Proceeds	29	70%
Journal	7	16%
Others (Book, Thesis, White paper)	6	14%

TABLE II. TOP FIVE PUBLICATION VENUES OF IDENTIFIED ARTICLES

Source	Acronym	No.
International Conference on Open Source Systems	OSS	3
International Symposium on Empirical Software Engineering and Measurement	ESEM	3
International Symposium on Software Reliability Engineering	ISSRE	3
ACM Conference on Computer and Communications Security	ACM CCS	2
International Conference on Engineering and MIS	ICEMIS	2

VII. DISCUSSION

This section describes and discusses the findings from the data extraction and analysis activities. The findings are presented in a graphical view and are organized by research question mentioned in section IV (A).

RQ1: What research has been conducted on the security practices and behaviors in the context of OSS development?

Table III shows the categorization of security areas and related publications that fit the areas using OWASP SAMM presenting in section III (A). Based on our review, the focus in the OSS development varies in different papers. Fig. 5 shows that 'Verification' is the most cited category in our SLR study (47%). This is due to the facts that open source development generally lack formal system verification. The other reason is that as vulnerabilities introduced in the design or construction stage will manifest themselves in code review or security testing if not detected earlier.

As shown in Fig. 6, ‘Construction’ received the second highest attention (29 %) in which sub-category of ‘Secure Architecture’ has significantly higher numbers of studies (10 out of 14). The topics discussed in this area include the characteristics of security bugs [40], [58], vulnerable code change in OSS, [9], [11], [12], secure system design [15], [47], [55] and adoption of security tools [16], [33].

‘Deployment’ and ‘Governance’ are the two areas that receive the least attentions in the research, 14% and 10 %, respectively. This may be due to open source projects do not typically have a corporate management staff to organize, lead, monitor, and improve the software development processes, which explains how hard the project management functions are in these two areas, such as strategic management, policy management, training and operational enhancement, etc.

TABLE III. SECURITY AREAS OF THE SELECTED STUDIES

Category	Subcategory	Publications
Governance	Strategy & Metrics	[28, 38, 60, 67]
	Policy & Compliance	[67]
	Education & Guidance	n/a
Construction	Threat Assessment	[14]
	Security Requirement	[22, 40, 58]
	Secure Architecture	[9, 11, 12, 15, 16, 33, 40, 47, 55, 58]
Verification	Design Review	[27]
	Code Review	[1-3, 9, 10, 12, 24, 25, 41-43, 48]
	Security Testing	[16, 17, 30, 34, 45, 46, 49, 61, 64, 67]
Deployment	Vulnerability Management	[4, 6, 52, 54, 63]
	Environmental Hardening	[7]
	Operational Enhancement	[5]

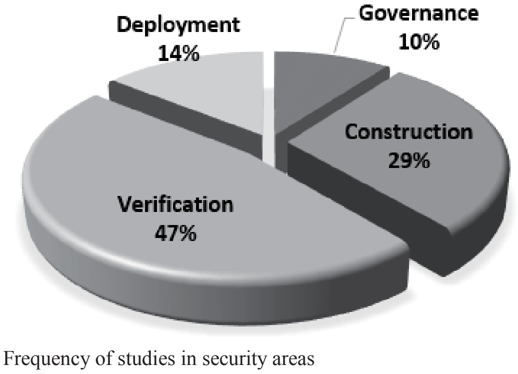

Fig. 6. Frequency of studies in security areas

RQ2: What research has been conducted on the socio-technical security aspects associated with OSS development?

Our second focus is to investigate the socio-technical perspectives of OSS security revealed in these studies. Among the selected 42 studies, only two studies applied socio-technical approaches to address software security in the context of open source development [41], [54]: Study [41] proposed socio-technical metrics to describe the code review collaboration; study [54] analyzed socio-technical aspects of software problem management in OSS communities. Despite that, we performed a socio-technical analysis on these papers to understand what social and technical elements are highlighted in them, which was based on the socio-technical models mentioned in section III (B). The analysis result is presented in Table IV.

From Fig. 7, we see that the discussion of technical aspects has happened in 98% of the selected studies (41 out of 42). However, less than 50% of studies talked about the social-sector of OSS security (cultural, structural, legal, managerial and operational), and the average value is only 16%.

Looking at the information in more detailed, ‘Operational’ security has the higher frequency of discussion (45%, 19 papers). This is because that the technical methods in software security are always accompanied with the certain process to have a successful implementation, especially at the working level. Compared with the significant portion of ‘Operational’ security, other social elements (cultural, structural, legal, and administrational) of OSS security have not been given enough attentions. They are noted in 7% (2 studies), 7% (2 studies), 2% (1 study) and 14% (7 studies) of selected studies, respectively.

RQ3: What research has been conducted focusing on aspects of security knowledge management in OSS development?

According to Table III, there is no OSS security practice categorized in ‘Education/Guideline’ in which the security training and knowledge management are major activities. However, some papers did address knowledge problems in relation to OSS security, which are summarized in Table V.

As we can see, lack of security knowledge is the common problem that the research usually deal with. Among these papers, only [1] and [34] (2 out of 6) have proposed systematic solutions to tackle security knowledge issues, which aim to minimize the human efforts in software verification.

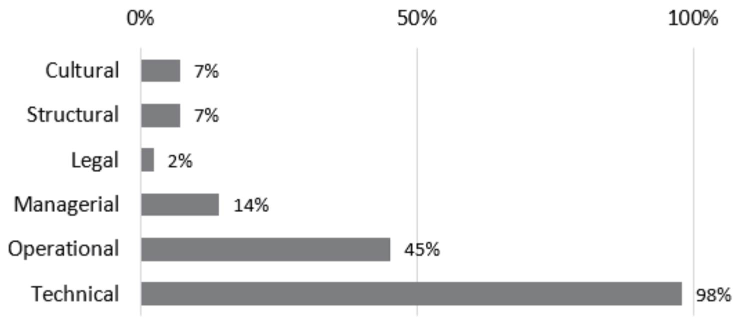

Fig. 7. Coverage rate of socio-technical aspects

TABLE IV. SOCIO-TECHNICAL ASPECTS OF THE SELECTED STUDIES

Social-Technical Aspects		Publications
Cultural	Incentive of OSS participants	[67]
	Developer reputation	[10]
	Testing culture	[49]
Structural	Onion model vs. Source code maintenance	[15]
	Core-periphery structure vs. Code review outcome	[10]
	Distributed team vs. Developing a shared model in bug fixing	[17]
Legal	Governments policies	[67]
Managerial	Software repository management (Malware prevention)	[15]
	Risk analysis	[28]
	Coordination and communication mechanisms (Code review and security testing)	[10, 17, 45, 49]
Operational	Vulnerability handling behavior	[4, 5, 17]
	Secure design process	[14]
	Coding behaviors	[3, 9, 11, 40, 41, 43, 58]
	System testing behaviors	[49, 67]
	Security practices and tools adoption	[28, 33, 55]
	Code review behaviors	[10, 24]
	Quality assurance process	[45]
Technical	[1-7, 9-12, 14-17, 22, 24, 25, 27, 28, 30, 33, 34, 40-43, 45-49, 52, 54, 55, 58, 60, 61, 63-65, 67]	

TABLE V. KNOWLEDGE PROBLEMS ADDRESSED IN THE SELECTED SECURITY STUDIES

Publication	Knowledge problems addressed in the study	Suggestions in the study
[1]	Lack of security knowledge in secure coding	Vulnerability prediction technique can provide a great help to OSS projects to deal with vulnerability flaw on a timely basis and with sufficient effort.
[34]	Lack of security knowledge in secure coding	Proposed an exploitable automatic verification system for secure open source software
[3]	Lack of security knowledge in secure coding	The OSS project should emphasize secure programming standards and reduce the use of unsafe statements.
[55]	Lack of knowledge in adoption of security tactics	The OSS project should identify more practical security tactics and systematically incorporate them into the development process.
[9, 11]	Study the characteristics of the vulnerable changes and found that differences among developers' knowledge and experience affect their likelihood of authoring vulnerable code change.	The OSS project should (a) create or adopt secure coding guidelines, (b) create a dedicated security review team, (c) ensure detailed comments during review to help knowledge dissemination and (d) encourage developers to make small, incremental changes.

VIII. Limitation of the Study

Even though this systematic literature review has been supported by a rigorous review methodology, well-defined study protocol, and a close-knit paper screening process, it has some limitation.

A. Missing relevant publications

Our results depend on the used keywords and the limitations of the selected search engines. This approach misses the papers that are not indexed by the search engine and the papers that are not indexed with the keywords we used. We note that keywords are both discipline and language specific and are not standardized. In order to limit the risk of incompleteness in keywords lists, we used alternative spellings and synonyms to build the search terms. Furthermore, by basing the search on a defined set of digital database and the publication date, we excluded certain types of publications, work published through other channels or outside the defined timeframe. We can therefore not claim to have included all relevant publications. However, we adopted six popular digital databases with the full-text search to reduce inherent limitations of search engines. We believe that our preliminary results cover the most relevant published literature.

B. Bias in the selection of relevant studies

Another potential limitation of the study is that subjective decisions can occur during the paper selection phases that causes the bias in the selection execution. This is due to lack of clear description of context, objective, and results of the selected studies. In order to mitigate this limitation, the selection process was carried out in an iterative way and the data extraction was realized. The selection execution in each paper screening stage was validated through an internal review process, which also helps to reduce the bias in the selection of studies.

IX. Conclusion

This paper presents the systematic literature review that was conducted to identify open source studies with respect to the research practitioners for further work on open source security.

A total of 42 papers were selected in the SLR that met our inclusion criteria. The selected studies were analyzed and extracted data was classified into four main categories namely Governance, Construction, Verification, and Deployment. The result shows that security areas in Construction and Verification (Secure Architecture, Code Review, and Security Testing) are followed by researchers with more interests than other areas in Governance and Deployment.

Next, based on our research, the security studies in OSS development are mostly technical driven. The socio-technical perspective has not gained much attention in this research area (2 out of 42 papers). According to the result of socio-technical analysis on the selected papers, the discussions between technical and social aspects seem quite unbalanced, either (Coverage rate: 98% versus 16% in average). The socio-technical perspective has as the main target to blend both the technical and the social systems in an organization. This can be viewed as a necessary condition within a security management framework as both aspects are of equal importance [29]. Technical security practice considering different social aspects (e.g., culture and structure) of open source development will assure the effectiveness and efficiency of the implementation of the tool.

Furthermore, the result of this SLR study also shows the gap that there is a lack of knowledge management aspects of open source security. Several researchers did mention the knowledge problems in securing OSS development, however, we cannot identify any study tackle this security issue from knowledge management perspectives.

Based on the finding of this research, we have come to the conclusion that the existing software security practices have limitations in supporting secure open source development. Secure architecture, code review, and security testing do help secure OSS products. However, as there is less research on socio-technical security aspects and no discussion of security knowledge management in the context of OSS development, these practices, and software security knowledge cannot be effectively spread within the open source community. Since OSS participants are not experts on security in general and the domain knowledge of software security is vast and extensive, it is suggested that future research should explore socio-technical approaches in helping OSS developers learn the necessary security knowledge to fulfill the need of their work, further, to reinforce their behaviors towards OSS security.

The contribution of this work is to supply researchers with a summary of existing information about software security in open source development in a thorough manner, so as to provide a context in which to operate. It can also provide other researchers with a firm basis on which to develop new security approaches for open source development and address any of the identified limitations.

Acknowledgement

The author would like to thank Professor Dr. Stewart Kowalski and Professor Dr. Rune Hjelsvold of Faculty of Information Technology and Electrical Engineering at Norwegian University of Science and Technology, who have made comments and suggestions in this paper.

References

[1] Abunadi, I. and M. Alenezi (2015). "Towards cross project vulnerability prediction in open source web applications". Proceedings of the The International Conference on Engineering & MIS 2015, ACM.

[2] Alenezi, M. and Y. Javed (2016). "Open source web application security: A static analysis approach". Engineering & MIS (ICEMIS), International Conference on, IEEE.

[3] Alnaeli, S. M., M. Sarnowski, M. S. Aman, K. Yelamarthi, A. Abdelgawad and H. Jiang (2016). "On the evolution of mobile computing software systems and C/C++ vulnerable code: Empirical investigation". Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual, IEEE.

[4] Altinkemer, K., J. Rees and S. Sridhar (2008). "Vulnerabilities and patches of open source software: an empirical study." Journal of Information System Security. volume 4, issue 2, pp. 3-25.

[5] Anbalagan, P. and M. Vouk (2009). "Towards a unifying approach in understanding security problems". ISSRE'09. 20th International Symposium on Software Reliability Engineering, IEEE.

[6] Anbalagan, P. and M. Vouk (2010). "Towards a bayesian approach in modeling the disclosure of unique security faults in open source projects". IEEE 21st International Symposium on Software Reliability Engineering (ISSRE), IEEE.

[7] Banday, M. T. (2011). "Ensuring Authentication and Integrity of Open Source Software using Digital Signature." International Journal of Computer Application (IJCA), Special Issue on “Network Security and Cryptography”, 2011

[8] Black Duck Software (2016). "Security in the age of open source " Web: https://www.slideshare.net/blackducksoftware/september-13-2016-security-in-the-age-of-open-source.

[9] Bosu, A. (2014). "Characteristics of the vulnerable code changes identified through peer code review". Companion Proceedings of the 36th International Conference on Software Engineering, ACM.

[10] Bosu, A. and J. C. Carver (2014). "Impact of developer reputation on code review outcomes in OSS projects: an empirical investigation". Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ACM.

[11] Bosu, A., J. C. Carver, M. Hafiz, P. Hilley and D. Janni (2014). "Identifying the characteristics of vulnerable code changes: An empirical study". Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering, ACM.

[12] Bosu, A., J. C. Carver, M. Hafiz, P. Hilley and D. Janni (2014). "When are OSS developers more likely to introduce vulnerable code changes? A case study". IFIP International Conference on Open Source Systems, Springer.

[13] Chandra, P. (2009). "The Software Assurance Maturity Model-A guide to building security into software development." Web: http://www.opensamm.org/

[14] Chehrazi, G., I. Heimbach and O. Hinz (2016). "The impact of security by design on the success of open source software". Research Papers. ECIS 2016 Proceedings, Paper 179.

[15] Colomina, I., J. Arnedo-Moreno and R. Clarisó (2013). "A study on practices against malware in free software projects". 2013 27th International Conference on Advanced Information Networking and Applications Workshops, IEEE.

[16] Cowan, C. (2003). "Software security for open-source systems." IEEE Security & Privacy. volume 99, issue 1, pp. 38-45.

[17] Crowston, K. and B. Scozzi (2008). "Bug fixing practices within free/libre open source software development teams." Journal of Database Management, Volume 19, Issue 2, Number 2, pp. 1–30.

[18] Crowston, K., K. Wei, J. Howison and A. Wiggins (2012). "Free/Libre open-source software development: What we know and what we do not know." ACM Computing Surveys (CSUR). volume 44, issue 2, pp. 7.

[19] Dabbish, L., C. Stuart, J. Tsay and J. Herbsleb (2012). "Social coding in GitHub: transparency and collaboration in an open software repository". Proceedings of the ACM 2012 conference on Computer Supported Cooperative Work, ACM.

[20] Damaševičius, R. (2007). "Analysis of software design artifacts for socio-technical aspects." INFOCOMP Journal of Computer Science. volume 6, issue 4, pp. 7-16.

[21] Damaševičius, R. (2009). On the human, organizational, and technical aspects of software development and analysis. Information Systems Development, Springer: 11-19.

[22] Damiani, E., C. A. Ardagna and N. El Ioini (2009). OSS security certification. Open Source Systems Security Certification, Springer: 1-36.

[23] Ducheneaut, N. (2005). "Socialization in an open source software community: A socio-technical analysis." Computer Supported Cooperative Work (CSCW). volume 14, issue 4, pp. 323-368.

[24] Edwards, N. and L. Chen (2012). "An historical examination of open source releases and their vulnerabilities". Proceedings of the 2012 ACM conference on Computer and communications security, ACM.

[25] Erturk, E. (2012). "A case study in open source software security and privacy: Android adware". World Congress on Internet Security (WorldCIS-2012), IEEE.

[26] Feller, J., P. Finnegan, D. Kelly and M. MacNamara (2006). Developing open source software: a community-based analysis of research. Social Inclusion: Societal and Organizational Implications for Information Systems, Springer: 261-278.

[27] Feng, Q., R. Kazman, Y. Cai, R. Mo and L. Xiao (2016). "Towards an architecture-centric approach to security analysis". 2016 13th Working IEEE/IFIP Conference on Software Architecture (WICSA), IEEE.

[28] Fortify's Security Research Group (2008). "Open Source Security Study: How Are Open Source development communities embracing Security Best practices?".

[29] Fox, W. M. (1995). "Sociotechnical system principles and guidelines: past and present." The Journal of Applied Behavioral Science. volume 31, issue 1, pp. 91-105.

[30] Groven, A.-K., K. Haaland, R. Glott and A. Tannenberg (2010). "Security measurements within the framework of quality assessment models for free/libre open source software". Proceedings of the 4th European conference on Software Architecture, ACM.

[31] Hales, D. and C. Douce (2002). "Modelling Software Organisations". Proc. of PPIG.

[32] Hauge, Ø., C. Ayala and R. Conradi (2010). "Adoption of open source software in software-intensive organizations–A systematic literature review." Information and Software Technology. volume 52, issue 11, pp. 1133-1154.

[33] Jordan, T. B., B. Johnson, J. Witschey and E. Murphy-Hill (2014). "Designing Interventions to Persuade Software Developers to Adopt Security Tools". Proceedings of the 2014 ACM Workshop on Security Information Workers, ACM.

[34] Kim, B., J.-h. Song, J.-P. Park and M.-s. Jun (2015). Design of Exploitable Automatic Verification System for Secure Open Source Software. Advances in Computer Science and Ubiquitous Computing, Springer: 275-281.

[35] Kitchenham, B. (2004). "Procedures for performing systematic reviews." Keele, UK, Keele University. volume 33, issue 2004, pp. 1-26.

[36] Kitchenham, B. (2007). "Guidelines for performing systematic literature reviews in software engineering." Tech. rep., Software Engineering Group, School of Computer Science and Mathematics, Keele University, and Department of Computer Science, University of Durham, eBSE Technical Report, EBSE-2007-01.

[37] Kowalski, S. (1994). "IT insecurity: a multi-discipline inquiry." PhD Thesis, Department of Computer and System Sciences, University of Stockholm and Royal Institute of Technology, Sweden. ISBN: 91-7153-207-2.

[38] Krishnamurthy, S. and A. K. Tripathi (2006). "Bounty programs in free/libre/open source software." BITZER Jurgen, The Economics of Open Source Software Development, Lavoisier, Paris. volume, issue, pp. 165-183.

[39] Levy, J. (2016). "Top Open Source Security Vulnerabilities." WhiteSource Blog. Web: https://www.whitesourcesoftware.com/whitesource-blog/open-source-security-vulnerability/.

[40] Li, Z., L. Tan, X. Wang, S. Lu, Y. Zhou and C. Zhai (2006). "Have things changed now?: an empirical study of bug characteristics in

modern open source software". Proceedings of the 1st workshop on Architectural and system support for improving software dependability, ACM.

[41] Meneely, A., A. C. R. Tejeda, B. Spates, S. Trudeau, D. Neuberger, K. Whitlock, C. Ketant and K. Davis (2014). "An empirical investigation of socio-technical code review metrics and security vulnerabilities". Proceedings of the 6th International Workshop on Social Software Engineering, ACM.

[42] Meneely, A. and L. Williams (2009). "Secure open source collaboration: an empirical study of linus' law". Proceedings of the 16th ACM conference on Computer and communications security, ACM.

[43] Meneely, A. and L. Williams (2010). "Strengthening the empirical analysis of the relationship between Linus' Law and software security". Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, ACM.

[44] Meneely, A. and L. Williams (2011). "Socio-technical developer networks: Should we trust our measurements?". Proceedings of the 33rd International Conference on Software Engineering, ACM.

[45] Michlmayr, M., F. Hunt and D. Probert (2005). "Quality practices and problems in free software projects". Proceedings of the First International Conference on Open Source Systems.

[46] Mockus, A., R. T. Fielding and J. D. Herbsleb (2002). "Two case studies of open source software development: Apache and Mozilla." ACM Transactions on Software Engineering and Methodology (TOSEM). volume 11, issue 3, pp. 309-346.

[47] Mourad, A., M.-A. Laverdière and M. Debbabi (2006). "Security hardening of open source software". Conference on Privacy, Security and Trust.

[48] Nagy, C. and S. Mancoridis (2009). "Static security analysis based on input-related software faults". CSMR'09. 13th European Conference on Software Maintenance and Reengineering, IEEE.

[49] Pham, R., L. Singer, O. Liskin, F. Figueira Filho and K. Schneider (2013). "Creating a shared understanding of testing culture on a social coding site". 35th International Conference onSoftware Engineering (ICSE), IEEE.

[50] Pham, R., L. Singer, O. Liskin, F. Figueira Filho and K. Schneider (2013). "Creating a shared understanding of testing culture on a social coding site". Software Engineering (ICSE), 2013 35th International Conference on, IEEE.

[51] Pittenger, M. (2016). "Know your open source code." Network Security. volume 2016, issue 5, pp. 11-15.

[52] Ransbotham, S. (2010). "An Empirical Analysis of Exploitation Attempts Based on Vulnerabilities in Open Source Software". Proceedings of the 9th Workshop on Economics of Information Security, Cambridge, MA, June 2010.

[53] Ransbotham, S. (2010). "An Empirical Analysis of Exploitation Attempts Based on Vulnerabilities in Open Source Software". WEIS.

[54] Ripoche, G. and L. Gasser (2003). "Scalable automatic extraction of process models for understanding F/OSS bug repair". Proceedings of ICSSEA’03.

[55] Ryoo, J., B. Malone, P. A. Laplante and P. Anand (2016). "The use of security tactics in open source software projects." IEEE Transactions on Reliability. volume 65, issue 3, pp. 1195-1204.

[56] Scacchi, W., J. Feller, B. Fitzgerald, S. Hissam and K. Lakhani (2006). "Understanding free/open source software development processes." Software Process: Improvement and Practice. volume 11, issue 2, pp. 95-105.

[57] Stol, K.-J. and M. A. Babar (2009). "Reporting empirical research in open source software: the state of practice". IFIP International Conference on Open Source Systems, Springer.

[58] Tan, L., C. Liu, Z. Li, X. Wang, Y. Zhou and C. Zhai (2014). "Bug characteristics in open source software." Empirical software engineering. volume 19, issue 6, pp. 1665-1705.

[59] Tawileh, A., J. Hilton and S. Mcintosh (2006). Modelling the Economics of Free and Open Source Software Security. ISSE 2006—Securing Electronic Busines Processes, Springer: 326-335.

[60] Tawileh, A., J. Hilton and S. Mcintosh (2006). "Modelling the Economics of Free and Open Source Software Security". ISSE 2006 - Securing Electronic Business Processes: Highlights of the Information Security Solutions Europe Conference, Springer.

[61] Vangaveeti, A. (2015). "An Assessment of Security Problems in Open Source Software.".

[62] Von Krogh, G. and E. Von Hippel (2006). "The promise of research on open source software." Management Science. volume 52, issue 7, pp. 975-983.

[63] Vouk, M. and L. Williams (2013). "Using software reliability models for security assessment—Verification of assumptions". 2013 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), IEEE.

[64] Walden, J., M. Doyle, G. A. Welch and M. Whelan (2009). "Security of open source web applications". Proceedings of the 2009 3rd international Symposium on Empirical Software Engineering and Measurement, IEEE Computer Society.

[65] Witschey, J. (2013). "Secure development tool adoption in open-source". Proceedings of the 2013 companion publication for conference on Systems, programming, & applications: software for humanity, ACM.

[66] Xiao, S., J. Witschey and E. Murphy-Hill (2014). "Social influences on secure development tool adoption: why security tools spread". Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing, ACM.

[67] Xiong, M., L. Huang, A. Tolba, W. Wong, E. Vandenberg and M. El-Gammal (2004). "Perspectives on the Security of Open Source Software." eBook, ISBN: 9780262345774.

TABLE VI. LIST OF SELECTED STUDIES

Author	Year	Title	ID
Abunadi, I. & Alenezi, M.	2015	Towards cross project vulnerability prediction in open source web applications	[1]
Alenezi, M. & Yasir, J.	2016	Open source web application security: A static analysis approach	[2]
Alnaeli, S. M., et al.	2016	On the Evolution of Mobile Computing Software Systems and C/C++ Vulnerable Code	[3]
Altinkemer, K. et al.	2008	Vulnerabilities and Patches of Open Source Software: An Empirical Study	[4]
Anbalagan, P. & Mladen V.	2010	Towards a Bayesian approach in modeling the disclosure of unique security faults in open source projects	[6]
Anbalagan, P. and Mladen V.	2008	Towards a Unifying Approach in Understanding Security Problems	[5]
Banday, M. T.	2011	Ensuring Authentication and Integrity of Open Source Software using Digital Signature	[7]
Bosu, A.	2014	Characteristics of the vulnerable code changes identified through peer code review	[9]

Bosu, A. & Jeffrey C. C.	2014	Impact of Developer Reputation on Code Review Outcomes in OSS Projects: An Empirical Investigation	[10]
Bosu, A. et al.	2014	Identifying the characteristics of vulnerable code changes: An empirical study	[11]
Bosu, A. et al.	2014	When are OSS developers more likely to introduce vulnerable code changes? A case study	[12]
Chehrazi G. et al.	2016	The impact of security by design on the success of open source software	[14]
Colomina, I. et al.	2013	A study on practices against malware in free software projects	[15]
Cowan, C.	2003	Software Security for Open-Source Systems	[16]
Crowston, K. & Barbara S.	2008	Bug fixing practices within free/libre open source software development teams	[17]
Damiani, E. et al.	2009	OSS security certification	[22]
Edwards, N. & Liqun C.	2012	An Historical Examination of Open Source Releases and Their Vulnerabilities	[24]
Erturk, E.	2012	A Case Study in Open Source Software Security and Privacy	[25]
Feng, Q. et al.	2016	Towards an architecture-centric approach to security analysis	[27]
HP Fortify's Security Research Group	2008	How Are Open Source Development Communities Embracing Security Best Practices	[28]
Groven, A. K. et al	2010	Security measurements within the framework of quality assessment models for free/libre open source software	[30]
Jordan, T. B. et al.	2014	Designing Interventions to Persuade Software Developers to Adopt Security Tools	[33]
Kim, B. et al	2015	Design of exploitable automatic verification system for secure open source software	[34]
Krishnamurthy, S. & Arvind K. T.	2006	Bounty Programs in Free/Libre/Open Source Software	[38]
Li, Z. et al.	2006	Have things changed now?: An empirical study of bug characteristics in modern open source software	[40]
Meneely, A. et al.	2014	An Empirical Investigation of Socio-technical Code Review Metrics and Security Vulnerabilities	[41]
Meneely, A. & Laurie W.	2009	Secure open source collaboration: An empirical study of Linus' law	[42]
Meneely, A. and Laurie W.	2010	Strengthening the empirical analysis of the relationship between Linus' Law and software security	[43]
Martin, M. etl al.	2005	Quality practices and problems in free software projects	[45]
Mockus, A. et al.	2002	Two case studies of open source software development: Apache and Mozilla	[46]
Mourad, A. et al.	2006	Security Hardening of Open Source Software	[47]
Nagy, C. & Spiros M.	2009	Static security analysis based on input-related software faults	[48]
Pham, R. et al.	2013	Creating a Shared Understanding of Testing Culture on a Social Coding Site	[50]
Ransbotham, S.	2010	An Empirical Analysis of Exploitation Attempts based on Vulnerabilities in Open Source Software	[53]
Ripoche, G. & Les G.	2003	Scalable automatic extraction of process models for understanding FOSS bug repair	[54]
Ryoo, J. et al.	2016	The Use of Security Tactics in Open Source Software Projects	[55]
Tan, L. et al.	2014	Bug characteristics in open source software	[58]
Tawileh, A. et al.	2006	Modelling the economics of free and open source software security	[59]
Vangaveeti, A.	2015	An Assessment of Security Problems in Open Source Software	[61]
Vouk, M. & Laurie W.	2013	Using software reliability models for security assessment - Verification of assumptions	[63]
Walden, J. et al	2009	Security of open source web applications	[64]
Xiong, M. etl al.	2004	Perspectives on the Security of Open Source Software	[67]