SindhUniv. Res. Jour. (Sci. Ser.) Vol. 52 (04) 381-386 (2020)

http://doi.org/10.26692/sujo/2020.12.57

SINDH UNIVERSITY RESEARCH JOURNAL (SCIENCE SERIES)

Measuring Secure Software Development Awareness and Usage among Software Developers

Z. A. MAHER[++]*, M. H. DEPAR*, M. MEMON*, M. Y. KOONDHAR*, P. K. BUTT*, A. SHAH[++]

Faculty of Information & Communication Technology, International Islamic University Malaysia

Received 07th April 2020 and Revised 13thNovember 2020

Abstract: Secure software development practices adoption could be the most influential factor for the future of software industry. Awareness and knowledge about security mechanisms and secure coding techniques plays a vital role in decision to consider security during software development by the software developers. This study aims to explore the security and secure software development awareness among software developers. The results of the study related to the use of secure software development methodologies showed a very low level of adoption level by the developers. It was revealed by the respondents that 82.72% respondents have never used any secure development methodology, about 13.67% respondents were using secure software development methodology in their software development from 1 to 4 years, 2.09% respondents using secure software development methodology in their software development from more than five years and only 1.57% respondents were using secure software development methodologies in their software development from less than a year. All the respondents were well aware of the software security but only 22.51% of the respondents were aware of the latest secure software development methodologies.

Keywords: Use of Secure software development, Software industry, Secure Development, Software Security, Security Awareness.

1. INTRODUCTION

Security awareness is an essential skill which is the need of software developers to develop secure software systems. Developers are required to understand how their developed application will integrate with the other infrastructure of the organization considering the existing security policies. Security engineering is a process which supports secure software development in delivering a secure product that prevents misuse and malicious behavior. The aim of the security engineering is to design, build and test a secure software system. Security engineering process improves the security of a system by adding predefined best practices for the developers. The process enforces developers to focus on security by providing guidelines and concrete steps.

Usually a software developer's performance is historically measured from the quick and feature packed code written, with no or little focus on security requirements. A change of mindset is needed in developers and the organizations to understand the importance of security within their code. The only way to go with is to integrate security throughout Software development life cycle by raising awareness and teaching software developers about secure coding practices and use of security tools.

Secure Software Development

In simple words secure software development can be termed as an assurance process which consists different activities for secure development. There are a number of methodologies present in the literature which incorporate security in each stage of software development process. A number of activities are included to improves software development process, and provide metrics for detecting and minimizing software vulnerabilities in the software development life cycle (SDLC). Risk management has also been integrated into SDLC for the development of secure software. Risk management has not been fully discovered by the engineers and researchers yet due to its difficulty.

Secure Software Development Awareness

Lack of secure software development awareness is one of the biggest risks organization can face. This risk, if exploited, can cost thousands of dollars and have potential to bring down corporations and may result in lawsuits from its clients who face their data loss due to software vulnerabilities occurrences as result of ignorance towards security inclusion during development phase. According to Rogers(Rogers, 1995) various technologies acceptance studies presented that the user should obtain sufficient knowledge and

*Information Technology Centre, Sindh Agriculture University Tandojam, Pakistan

information from various channels to measure the benefits before taking the decision on acceptance of new IT/IS innovation. Therefore, this study attempts to explore the effect of secure software development methodologies awareness on behavioral intention to use secure software development methodologies.

2. STUDY BACKGROUND

There are a number of software development methodologies were proposed which focuses security in different ways during software development at various stages. Some of the most common methodologies include; Cigital Software Security Touchpoints (McGraw, 2006) is a lightweight security engineering process. Cigital integrates core activities of development process for improvement in the security of the product (Steven 2006) and Common Criteria (Keblawi andSullivan 2006) is also a well mature ISO certified security engineering principal. The Microsoft Security Development Lifecycle (SDL) (Howard, and Lipner, 2009) is a software development process presented to reduce the maintenance costs and increase reliability of software related to software security bugs. Other security processes have been included such as Team Software Process (TSP) proposed by (Humphrey, 2000), Security Development Processes (DeWin*et al.*,2009), Correctness by Construction (Hall and Chapman 2002) and Security pattern deployment diagram respectively (Maher *et al.,* 2016)

Secure software development is the collection of different activities combined together in effort to produce a secure software system. It utilizes a number of techniques, starting from the secure design, to develop a secure software development lifecycle and finally developing some secure software coding/programming practices. Inclusion of security concerns into software system was started in 2004, initially it was named as app sec, software assurance, secure coding programs, and then finally termed as Secure Development Life Cycles (SDLs). Microsoft's introduced very first processes in 2004 for developing secure software systems and named it as Security Development Lifecycle. After that many other methodologies and techniques have been introduced by the industry and academia to develop secure applications. Nigerian University of Agriculture presented a process called Secure Software Development Model (SSDM) (Sodiya *et al.*,2006) in which they integrated security activities into software engineering process. The activities suggested were security specification, threat modeling, security specification, Security training review, and penetration testing to include in software engineering process. In this model, security specifications were separated from the functional specification. Another secure development process for developing security requirements of software systems named as Comprehensive, Lightweight Application Security Process (CLASP) was presented by (Gebser *et al.*,2007) It comprises seven best practices for collecting security requirement; which includes: Security awareness, derivation of security requirements, application evaluations, developing vulnerability remediation measures, defining and monitoring metrics, implementation of secure development practices, and providing effective guidelines.

Microsoft presented its Security Development Lifecycle (SDL) which was focused on security integration activities during software development to reduce expected vulnerabilities during software development. The activities presented in SDL are proposed to overcome the security issues. (McGraw, 2006) presented "Seven touchpoints" for software developers to implement security during different stages in software development. Touchpoints provides best practices used extensively over the years in the industry with remarkable results. The 'touchpoints' approach was focused on increasing effectiveness through: security requirements, abuse cases, security operations, architectural risk analysis, code review, risk-based security testing and penetration testing.

3. PROBLEM STATEMENT

It is important for software developers to have knowledge about secure coding techniques and how these techniques applied during software development is essential for the development of secure software systems. Incorporating security in a software system was considered as an afterthought in software development, whereas software developers mainly focus on functional part of the developing software system (Maher *et al.,* 2018). However, increasing security threats have acknowledged the significance of security inclusion in the software development lifecycle (Geer, 2010). From the previous literature a number of studies mentioned security awareness or security related knowledge or skill related to security as a major reflection of information security culture towards secure software development adoption by the developers (Bojmaeh 2015) Security awareness is considered as a major factor towards developer's decision to accept software development methodologies (Geer, 2010). (Maher *et al.*,2020) (Johnson *et al.,* 1999) (Compeau *et al.,* 1999) (Thompson *et al.,* 1991). More precisely, many organizations consider 'security awareness' a key part of security culture. To know the level of security awareness of a software developer is a difficult task as it is highly subjective and diverse between individuals.

There is lack of common method in industry to model and analyze the security for software system. Due to which programmers and developers who are not security experts always face problems to understand security constraints. Software developers face difficult time to deploy security constraints (Bouaziz and Kammoun 2016) because they are not security experts (Vieira and Antunes 2013) Security mechanisms and there implementation is not straightforward. It is a highly complex task that's why it is difficult for the developer to deploy all the security mechanisms and security requirements to secure implementation of the software system. Identifying potential security threats and vulnerabilities at early stages of software development is a difficult task for software developers (Kobashi, *et al.,* 2015) (Maher *et al.,* 2018). Selection of appropriate security mechanism is also proved as a major difficulty in security integrating at software design phase, secondly where and how these selected mechanisms are applied in the system along with its level of abstraction which also increases the complexity of the process(Bouaziz *et al.,* 2011). Concrete guidelines are needed by the developers to develop secure applications (Lodderstedt *et al.,*2002). Lack of awareness about security tools also plays a major role in security consideration during software development.

4. RESEARCH METHODLOGY

This research is the part of a project where quantitative survey methodology was used to reveal the factors affecting secure software development adoption among software developers. The results for the first part of the survey questionnaire is presented here in this paper containing demographic details of the respondents and the software security and secure software development awareness among software developers. The self-administered questionnaire option is used for this study, in order to get a maximum number of responses at all levels. The population for this research is the software engineers and programmers working in software development firms at Malaysia. SPSS software package is used for data analysis in this research.

5. RESULTS AND DISCUSSION

Demographic and assessment of Security and Secure Software development awareness resultswere obtained for this study from total 382 respondents. All the results are discussed in the following sections.

Demographic Results

The type of questions for the demographic information of respondents were related to gender, age, position, Software Development Experience and Education.

There were 382 respondents for this study, the males comprises majority of the participants, as 58.63% of the respondents and the female participants were 41.36 %. Most of the respondents were between the ages of 31 and 40. They were 40.83%, while the second most respondent population fall between the age of 21 to 30 as 30.36% and the remaining respondents fall between 41 and 50 were 19.89%and aged above 60 as 2.61%. According to the job designation, most of the respondents were Junior Developers, with 47.12%, Senior Developers 30.89%, Senior Manager 12.04% and Project Manager 9.94%. According to the working experience of the respondents, most of them having 1 to 3 years of experience, which makes 49.21%, 4 to 10 years of experience population consists of 24.86% respondents, other remaining two categories consists less than 1 year were 2.09% and more than 10 years of experience and above were 23.83%.

Security and Secure Software development awareness assessment results

This section of the questionnaire is about the collection of the preliminary information regarding the awareness related to software security and secure software development practices. It comprises the questions regarding the security awareness, usage of security in software development, awareness about secure software development and usage of secure software development practices.

This part is comprised of five questions, starting from question 6 to 10; the details are depicted in detail in **(Table 1).**

Table-1. General assessment of Security Practices

Variable	Response	Frequency	Percentage
Software Security Awareness	Yes	382	100
	No	00	0.0
Usage Of Security In Software Development (Years)	Never	00	0.0
	< 1	94	24.60
	1-4	228	59.68
	=> 5	60	15.70
Awareness About Latest Secure Software Development Methodologies	Yes	86	22.51
	No	296	77.48
Ever Used Any Secure Software Development Methodology	Yes	66	17.27
	No	316	82.72
Experience Of Using Secure Software Development Methodologies (Years)	Never	316	82.72
	< 1	06	1.57
	1-4	52	13.67
	=> 5	08	2.09

The second part of the questionnaire reveals the awareness and usage of software secure and secure software development practices and methodologies. These facts in return can be used for the initial assessment of security is being treated in software industry. The first question of this part was directly asked about the awareness of software security among the software professionals, which provides two straightforward answers "Yes" and "No". A frequency analysis of the results shows that, all (n=382, 100%) respondents answered with "Yes" option. This portrays that all the respondents are somehow are aware of the software security.

The second question of this part was about the duration (in years) of security inclusion/usage in their software development, which shows that developers are not consistently using security in their software development practice. All of 382 respondents answered that they had used security in there software development in one or other way. Out of 382 respondents, 94 respondents are using security from less than years which is 24.60%. The highest number of the respondents is 228, which is using/including security in their software development form 1 to 4 years which is equal to 59.68% and those respondents (n=60, 15.70%) using/including security in their software development from more than five years.

The third question related to the awareness of latest secure software development methodologies among the respondents, which is also provided with two straight forward answers "Yes" and "No". A frequency analysis of the results shows that the only 86 respondents which is 22.51% answered with "Yes" option. This portrays that the less number of respondents are aware of latest secure software development methodologies and most of the respondents (n=296, 77.48%) are unaware of latest secure software development methodologies.

The fourth question was asked to know that how many of the respondents had ever used any secure software development methodology in their practice, which is also provided with two straight forward answers "Yes" and "No". A frequency analysis of the results shows that the only 66 respondents which is 17.23% of the respondents answered with "Yes" option and 316 respondent with "No" which means (n=316, 82.72%) had never used any secure software development methodology in their software development practice .

The last question of this part was about the duration (in years) of secure software development methodology usage their software development which shows that out of 382 respondents; 316 respondents never used secure development methodology which is 82.72%. Out of 382 respondents, only 02 respondents are using secure software development methodologies in their software development from less than a year, which is 1.57%. The highest number of the respondents is 52, which were using secure software development methodology in their software development from 1 to 4 years, which is equal to 13.67% and also only 8 respondents (n=8, 2.09%) using secure software development methodology in their software development from more than five years.

6. CONCLUSION

This study explored that all the respondents were somehow aware of the software security but very less number of respondents were aware of the latest secure software development methodologies and even less number among those were actually using secure software development methodologies in their practice. It was found that a very low level of secure software development methodologies awareness among software developers. Software firms should provide more facilitating conditions to developers. Such as; initiate security related training and awareness campaigns about secure software development methodologies to improve the adoption level of secure software development practices among software developers.

REFERENCES:

Bojmaeh H. Y. (2015) The Main Factors Influencing Information Security Behavior. International Journal of Science and Engineering Applications. (6):353-6.

Bouaziz R. and S. Kammoun (2016) SCRIStUDIO: A security pattern integration tool. In 2016 International Conference on Information Technology for Organizations Development (IT4OD) 1-6. IEEE.

Bouaziz R., B. Hamid, and N. Desnos (2011) Towards a better integration of patterns in secure component-based systems design. InInternational Conference on Computational Science and Its Applications 607-621. Springer, Berlin, Heidelberg.

Compeau D., C. A. Higgins. and S. Huff (1999) cognitive theory and individual reactions to computing technology: A longitudinal study. MIS quarterly. 1: 145-158.

DeWin B., R. Scandariato K. Buyens J. Grégoire and W. Joosen (2009) On the secure software development process: CLASP, SDL and Touchpoints compared. Information and software technology. 51(7):1152-71.

Geer D. (2010) Are companies actually using secure development life cycles?. Computer. 43(6):12-6.

Howard, M. and S. Lipner, (2009) The security development lifecycle, O'Reilly Media, Incorporated.

Gebser M, B. Kaufmann, A. Neumann, and T. Schaub (2007) clasp: A conflict-driven answer set solver. InInternational Conference on Logic Programming and Nonmonotonic Reasoning 260-265. Springer, Berlin, Heidelberg.

Humphrey, W. S. (2000) Introduction to the team software process (sm), Addison-Wesley Professional.

Hall A. and R. Chapman (2002) Correctness by construction: Developing a commercial secure system. IEEE software. 19(1):18-25.

Johnson R.A., B. C. Hardgrave, and E. R. Doke. (1999) An industry analysis of developer beliefs about object-oriented systems development. ACM SIGMIS Database: the DATABASE for Advances in Information Systems. 30(1): 47-64.

Kobashi T., M. Yoshizawa, H. Washizaki, Y. Fukazawa N. Yoshioka,T. Okubo, and H. Kaiya (2015) TESEM: a tool for verifying security design pattern applications by model testing. In 2015 IEEE 8th International conference on software testing, verification and validation (ICST) 1-8. IEEE.

Keblawi F. and D. Sullivan (2006) Applying the common criteria in systems engineering. IEEE security & privacy. 4(2):50-5.

Lodderstedt T., D. Basin and J. Doser (2002) SecureUML: A UML-based modeling language for model-driven security. InInternational Conference on the Unified Modeling Language 426-441. Springer, Berlin, Heidelberg.

Maher Z. A., H. Shaikh, M.S. Khan, A. Arbaaeen, and A. Shah (2018) Factors Affecting Secure Software Development Practices Among Developers-An Investigation. In 2018 IEEE 5th International Conference on Engineering Technologies and Applied Sciences (ICETAS) 1-6. IEEE.

Maher Z. A., N. F. Sani, J. Din, and M. A. Jabar (2016) Use of Security Patterns for Development of Secure Healthcare Information System. Journal of Medical Imaging and Health Informatics. 6(6):1541-7.

Maher Z. A., A. Shah, H. Shaikh, G. A. Rahu P. K. Butt, S. Chandio, and S. Shaikh (2018) A methodology for modeling and analysis of secure systems using security patterns and mitigation use cases. In2018 7th International Conference on Computer and Communication Engineering (ICCCE) 268-273. IEEE.

Maher Z. A., A. Shah S. Chan-dio H. M. Mohadis, and N.H. Rahim (2020) Challenges and limitations in secure software development adoption-A qualitative analysis in Malaysian software industry prospect. Indian Journal of Science and Technology. 13(26):2601-8.

McGraw, G. (2006) Software security: building security. Vol. 1, Addison-Wesley Professional.

Rogers E. M. (1995) Diffuison of Innovations. Elements of Diffusion.

Steven J. (2006) Adopting an enterprise software security framework. IEEE Security & Privacy. 4(2): 84-7.

Sodiya A. S., S. A. Onashoga, and O. B. Ajayĩ (2006) Towards Building Secure Software Systems. Issues in Informing Science & Information Technology. 1;3Pp.

Thompson R. L., C. A. Higgins, and J. M. Howell (1991) Personal computing: toward a conceptual model of utilization. MIS quarterly. 1:125-43.

Vieira M. and N. Antunes (2013) Introduction to Software Security Concepts. InInnovative Technologies for Dependable OTS-Based Critical Systems 29-38. Springer, Milano.