Spring 5-19-2016

Secure Software Development: A Developer Level Analysis

Nasim Talebi
University of Texas as San Antonio, nasim.talebi@utsa.edu

Emmanuel W. Ayaburi
University of Texas as San Antonio, emmanuel.ayaburi@utsa.edu

Secure Software Development: A Developer Level Analysis

Nasim Talebi
University of Texas as San Antonio
Nasim.Talebi@utsa.edu

Emmanuel W. Ayaburi
University of Texas as San Antonio
Emmanuel.Ayaburi@utsa.edu

Research in Progress

ABSTRACT

Developing secure software is still an important issue in the computing world. Big software firms spend huge sums of money to offer secure software and systems. However, security incidents due to insecure software results in loss of revenue and reputational damages to user firms. Incorporating security requirements early in the development process is the most effective and cheapest method to build secure software. We chose a behavioral lens in order to understand antecedents to secure software development. We explicate the effects of personality, training, education and organizational culture on the development of secure software.

Keywords

Secure software, education, training, personality, organization culture, security awareness

INTRODUCTION

Secured software is the one that meets requirements of confidentiality, integrity, availability and non-repudiation when in operation (Kissel, Stine, Scholl, Rossman, Fahlsing, and Gulick, 2008). The case of Sony's PlayStation Network been hacked due to its susceptibility to SQL injection attacks (Jason, 2011) only highlights the need for secure software. Misalignments of interest of those involved in the software development life cycle (SDLC) is among reasons attributed to this seemingly unsolvable problem (Covarity, 2012). For instance, while developers might be concerned with innovation in their software development, project managers are concerned about the cost of developing the software. In addition, while consumers might be more concerned about the on-time delivery of the software, testing team might advocate for elaborate testing.

Prior research has suggested different solutions such as the inclusion of security requirements in the SDLC (D'aubeterre, Singh, and Iyer, 2008). Most Software development projects start with unclear user requirement, which are fine tune during the development process. This highlights the need to look deeper into the ability and environment of the developer (programmer) in the process with regards to the inclusion of security requirement in the development of software. The dearth of individual and environmental level factors to complement suggested process level factors has inspired this study to take a deeper look at the problem of secure software development. Therefore, we seek to answer the following questions:

RQ1: What software developer (programmer) level factors affect the development of secure software?

RQ2: What developer environmental factors influence the process of secure software development?

To this end, we explore personality and human development literature to identify relevant answers to the above questions.

MODEL DEVELOPMENT

Personality

Personality has been studied in varied context in information systems research and it has been found to influence individual interactions and group performance (Balthazard, Waldman, and Warren 2009). With the close interaction among developers, it is relevant to consider the effect of the personality of those involved in the SDLC. In the software development context, high level of extraversion is found to be related to exceptional performance in software development (Clark, Walz, and Wynekoop, 2003). Despite these finding, the definition of high performance was not elaborate enough to include the security of the developed software. Could personality influence developers differently if the measure of performance were "secure software"? Yang, Kang, and Mason (2008) found that agreeableness of individuals influences the formation of shared mental models among developers which leads to better team performance. Barrick and Mount (1991) found openness to experience influences individual training proficiency and conscientiousness. Extending from Macha, Hallam, and Dietrich (2009), it can be

concluded that personality will affect cognitive empowerment of developers. This study argues that developer's awareness of security in SDLC could differ based on their personalities. Since cognitive abilities influence individual behavior, learning and application of security knowledge, it is hypothesized that:

H1: Personality of developers will have effect on security awareness of software development teams.

Education and Training

Security awareness training and education are essential part of defending Information System (Rezgui and Marks, 2008). While most developers receive good instruction on how to be an efficient developer or coder, these highly technical instructions are broad and do not emphasize adherence to security guidelines (Siponen, 2000). However, a more focused education on security will provide developers with necessary knowledge about secure software development. Although education is important, it has been advocated that it needs to be complement with efficient training. Training enables people to gain needed skills and knowledge for job performance. A more security-focused education along with efficient training provide developers with good reasoning behind the security requirements in software development (Roper, 2006). As training familiarizes developers with who, what, when and where of security requirement in the software development process, education get people to understand the why aspects (Roper, 2006). Thus,

H2: The level of security education a developer has is positively related to software development team security awareness.

H3: The level of security training a developer has is positively related to security awareness of the software development team.

Security Awareness and Compliance

Security awareness is the degree to which developers are knowledgeable about their actions and its effect on software vulnerabilities (Shaw, Chen, Harris, and Huang 2009; Bulgurcu, Cavusoglu, and Benbasat, 2010). Awareness expresses how much the teams know of who has what knowledge about the security requirements. The level of awareness has been demonstrated to influence the team members' performance in distributed software development teams (Espinosa, Slaughter, Kraut, and Herbsleb, 2007). Developers comply with security requirement in either Microsoft's SDL or proprietary guidelines when the level of knowledge about individual security competences is known to every member. High levels of security awareness are expected to lead to high team compliance with security requirements such that likely deficiencies in the process can be quickly arrested. Thus,

H4: The level of team security awareness will have a positive effect on team compliance with security requirements in software development.

Compliance Culture and Secure Software Development

Organizational culture is defined as values, norms, beliefs, and assumptions shared by organizations' members (Hofstede, 1990). The culture of any organization has an effect on the behavior of employees. Iivari and Huisman (2007) found that hierarchical organizations are more security oriented. This will translate into the level of concerns towards the deployment of secure system. The norms, beliefs and values that developers have about security requirements influences how team member comply with security requirements in software development. Hence,

H5: The level of compliance culture in the team will influence development of secure software such that teams with heighten level of culture will produce more secured software.

CONCLUDING REMARKS AND FUTURE RESEARCH DIRECTION

The software developer as the executioner of the software development process and the implementer of the guidelines is expected to be more likely to develop software that is more secured if they have the required training, education, personality mix, awareness and attitude toward compliance with the culture of the organization. The authors hope to empirically validate the model shown in Figure 1 through an experiment with software development students in future.

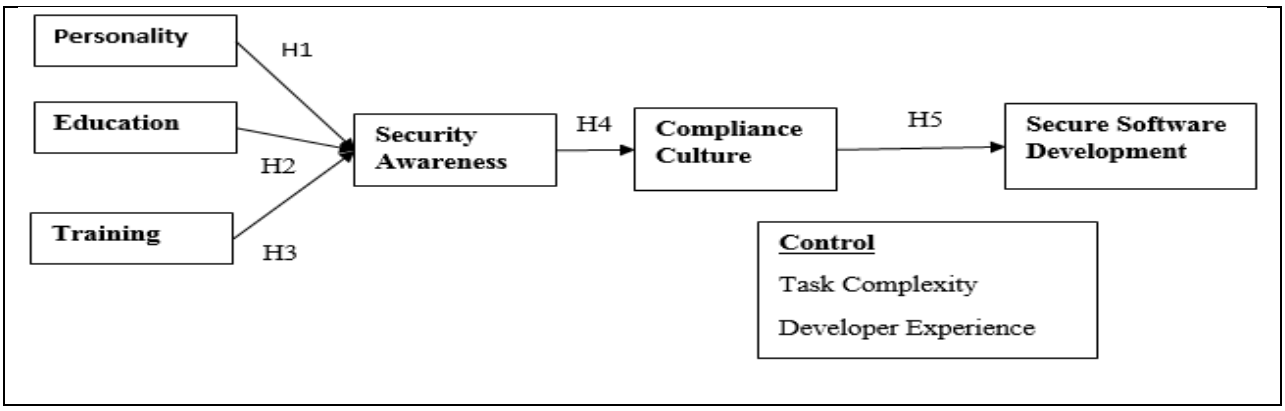

Figure 1. Research model

References

1. Balthazard, P. A., Waldman, D. A., & Warren, J. E. (2009). Predictors of the emergence of transformational leadership in virtual decision teams. The Leadership Quarterly, 20(5), 651-663. doi:10.1016/j.leaqua.2009.06.008
2. Barrick, M.R. & Mount, M.K. (1991). The big five personality dimensions and job performance: A meta-analysis. Personnel Psychology, 44, 1-26.
3. Bulgurcu, B., Cavusoglu, H., & Benbasat, I. (2010). Information security policy compliance: An empirical study of rationality-based beliefs and information security awareness. MIS Quarterly, 34(3), 523-548.
4. Clark, J. G., Walz, D. B., & Wynekoop, J. (2003). Identifying Exceptional Software Developers: A Comparison of Students and Professionals. Communications of the Association of lnformation Systems (CIAS), 11(8), 137-154.
5. Coverity, White Paper, 2012. " Building Security into Your Software Development Lifecycle," (available at http://www.coverity.com/library/pdf/coverity-security-wp.pdf).
6. D'aubeterre, F., Singh, R., & Iyer, L. (2008). Secure activity resource coordination: Empirical evidence of enhanced security awareness in designing secure business processes. European Journal of Information Systems, 17(5), 528-542. doi:10.1057/ejis.2008.42
7. Espinosa, J. A., Slaughter, S. A., Kraut, R. E., & Herbsleb, J. D. (2007). Team knowledge and coordination in geographically distributed software development.Journal of Management Information Systems, 24(1), 135-169. doi:10.2753/MIS0742-1222240104
8. Hofstede, G. (1990). Cultures and organizations: Software of the mind. New York: McGrawHill.
9. Iivari, J., & Huisman, M. (2007). The relationship between organizational culture and the deployment of systems development methodologies. MIS Quarterly, 31(1), 35-58.
10. Jackson, B. 2011. "How to Not Get Hacked Like Sony,"(available at http://www.pcworld.com/article/148007/security.html).
11. Kissel, R., Stine, K.M., Scholl, M.A., Rossman, H., Fahlsing, J., Gulick, J.: Sp 800–64 rev. 2. Security considerations in the system development life cycle (2008). Available at http://csrc.nist.gov/publications/nistpubs/800-64-Rev2/SP800-64-Revision2.pdf
12. Rezgui, Y., & Marks, A. (2008). Information security awareness in higher education: An exploratory study. Computers & Security, 27(7), 241-253. doi:10.1016/j.cose.2008.07.008
13. Roper, C. A., Grau, J. J., Fischer, L. F., & Books24x7, I. (2006;2005;). Security education, awareness, and training: From theory to practice. Burlington, MA: Elsevier Butterworth-Heinemann.
14. Ruben Mancha, Cory Hallam, and Glenn Dietrich, "Self-Efficacy in Software Developers: A Framework for the Study of the Dynamics of Human Cognitive Empowerment", International Journal of Information Technologies and Systems Approach, Vol. 2(2), 2009, pp 34 – 49.
15. Shaw, R. S., Chen, C. C., Harris, A. L., & Huang, H. (2009). The impact of information richness on information security awareness training effectiveness. Computers & Education, 52(1), 92-100. doi:10.1016/j.compedu.2008.06.011

16. Siponen, M. T. (2000). A conceptual foundation for organizational information security awareness. Information Management & Computer Security, 8(1), 31-41. doi:10.1108/09685220010371394
17. Yang, H., Kang, H., & Mason, R. M. (2008). An exploratory study on meta skills in software development teams: Antecedent cooperation skills and personality for shared mental models.European Journal of Information Systems, 17(1), 47-61. doi:10.1057/palgrave.ejis.3000730