Estimating Software Security of a College ERP System: A Case Study Using Software Product Security Framework Approach

Smriti Jain[1*], Nitin Kulkarni[2] and Maya Ingle[3]

[1] CS Department, Acropolis Institute of Management Studies and Research, Indore, M.P., India
[2] FCA, Acropolis Institute of Technology and Research, Indore, M.P., India
[3] School of Computer Science, DAVV, Indore, M.P., India
*smritijain2791@rediffmail.com

Abstract. Security is a major concern for business applications. With the intertwined dependencies of various security measures to secure an enterprise application, security must be included in the development process since its inception, i.e. from requirements gathering till the operation of the software. Various security controls are available to secure process as well as software product. Security consideration during the development process can lead to secured software. It can be supported by the help of security framework that considers security at managerial level, design and development. The capability to measure the security efforts throughout the development process may help to judge the overall security of the software. In this paper, we provide an emergent tool to analyze security efforts, and controls are integrated while developing secured software system. The Security Factor shall facilitate the development team to identify security efforts during the development of software using Software Product Security (SPS) framework. It shall also support in determining the improvement areas for secured software product development process.

Keywords: Software Security, Software Product Security Framework, Security Factor.

1 Introduction

The complexity of software give rise to improperly designed software leading to insecure software.The complexity is due to multiple and complex dependencies and integration that may lead to inadequate or overlooked secure design and implementation. Secure design needs to consider various security aspects before the inception of implementation process. The insecure software is subject to variety of attacks that may include gaining unauthorized access, data leaks, etc. It also leads to exploitation of software weaknesses. Thus, the insecure software increases the likelihood of attacks, and enhances maintenance cost and lost opportunity cost. Increasing security breaches are the result of the security vulnerabilities present in the software systems. These vulnerabilities are the consequence of insure design and coding [1]. Thus, it is crucial to consider security at every stage of the Software Development Life Cycle (SDLC) [2]. This shall lead to fewer security issues and

S. Bhalerao et al. (eds.), *Proceedings of the International Conference on Recent Advancement and Modernization in Sustainable Intelligent Technologies & Applications (RAMSITA-2025),*Advances in Intelligent Systems Research 192,
https://doi.org/10.2991/978-94-6463-716-8_74

speedier defect handling. Incorporating security during software development shall also reduce the overall cost of software [3].

Security breaches lead to financial burden on the business that include legal liabilities and recovery of data. Hence it is important for the businesses to consider secured software as one of the objectives. Businesses that consider security as one of the organizational policies, assists in supporting the development of secured software system. A security framework further encourages development of secured software system since its inception. It outlines the activities that are responsible for the creating secured software. It may act as a blueprint for the secured development process of software. It shall support an organization to decide key security functionalities required to safeguard the software system. Such a framework designed to minimize the number of vulnerabilities and lessen the potential impact of the unaddressed vulnerabilities [4]. The following section of the literature review details certain frameworks.

2 Literature Review

A number of frameworks are available to alleviate the business risks due to security. Secured software can be the result of the framework that uses twelve practices divided in four domains - Governance, Intelligence, SDL TouchPoints and Deployment [5]. Another framework describes security in terms of business resources and divides security in seven domains such as people and industry, data and information etc. [6]. Enterprise Software Security Framework (ESSF) is a structure that focuses on security in an enterprise at executive and application portfolio level [7]. Security IT framework merges elements of security framework and security policy domain. Security framework includes industry standards, policies and procedures, and security services; while security policy domain elements cover data protection, data classification, risk management, and so on [8]. An integrated three-stage framework based on security regulation analysis discusses eight security activities for secure development requirements and system security requirements [9]. The frameworks are mostly concerned with the overall security of the system. Another framework Secure Software Development Framework (SSDF) organizes security practices into four groups viz. Prepare the Organization, Protect the Software, Produce Well-Secured Software, and Respond to Vulnerabilities. It discusses what all security to be added to and integrated with each SDLC phase [4]. In our previous work, we developed Software Product Security (SPS) framework that may systematically incorporate security throughout the software product development process [10]. These frameworks do not provide a method that can help to judge the impact of the efforts of the development team in incorporating security in the software developed using a framework.

Security attacks are related to the vulnerabilities being identified. Hence, security strength of software system must be recognized as it is considered equivalent to the market price of such vulnerability [11]. In this paper, we suggest a mathematical model that estimates the security concerns of the system using SPS framework. In Section 3, we provide a brief review of SPS framework. The Security Factor of SPS framework is proposed in Section 4. A case study on ERP system of an engineering institute to elaborate on Security Factor is presented in Section 5. Finally, in Section 6, we present results and conclusion

3 Software Product Security Framework

The generalized Software Product Security framework is designed to develop secured software. It consists of three layers viz. control at top, security aspects in middle and development at the bottom. The layered SPS framework helps to consider security as an overall organizational control. It may also facilitate in providing guidelines for incorporating security systematically from the pre-development phase within the product itself [10]. The SPS framework can be summarized in Table 1:

Table 1. Summary of SPS framework

Layer	Purpose	Key Activities
Control Layer	Governance and Managerial Control	Policy formulation and implementation, minimum permission using principle of least privilege, separation of duties, documenting security requirements, security planning, training the system designers as well as developers for security implementation
Security Aspects Layer	Identification of security features such as attributes of security, security standards, technology and checklist	Identification of functional and non-functional security requirements, compliance with standards, use of security standards, using tools to design security, use of security checklist
Development Layer	Implementation of security measures	Follow Secured Software Development process, software process management models like CMM and ISO 9001 and the software metrics

4 Security Factor of the SPS Framework

In this section, we put forward a proposal to evaluate security of the software product taking into account the features of SPS framework. Thus, we try to develop a mathematical model to estimate the security involved in a software using the framework. The security concerned can be represented through Security Factor (F_s).The security factor is estimated on the basis of our previous work on SPS framework which is discussed in Section 3. Let k, l and m be the total measures for each layer of the framework and let W_A1, W_A2 and W_A3 be the weights assigned to the measures of control layer, security aspects layer and development layer respectively. The weights can be represented by the binary values 0 and 1 for the absence or presence of the characteristic of the layer. Figure 1 mentions the steps to calculate the Security Factor.

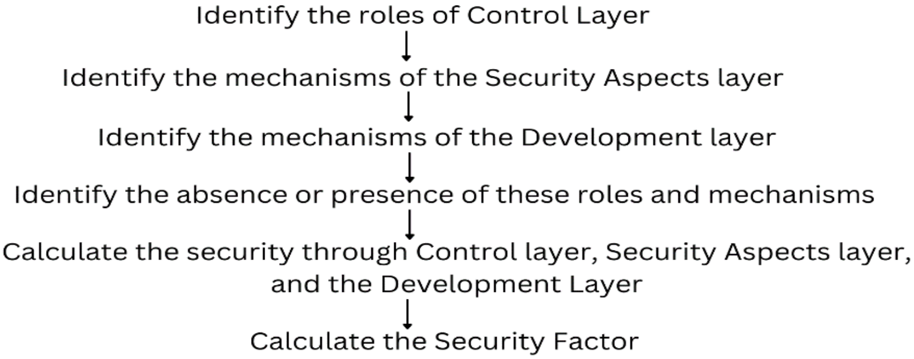
Identify the roles of Control Layer
Identify the mechanisms of the Security Aspects layer
Identify the mechanisms of the Development layer
Identify the absence or presence of these roles and mechanisms
Calculate the security through Control layer, Security Aspects layer, and the Development Layer
Calculate the Security Factor

Fig. 1. Steps to find Security Factor

The security factor can be calculated as follows:

Let the function W_A (i, j) specifies the mechanisms and roles of the three layers of the SPS framework.

W_A (i, j) = 1, if j^{th} assessable security feature at i^{th} layer exists
= 0, otherwise

Then, Security via Control Layer, $S_{CL} = \sum_{j=1}^{k} WA(1, j) / k$

Security due to Aspects Layer, $S_{AL} = \sum_{j=1}^{l} WA(2, j) / l$

and, Security through Development Layer, $S_{DL} = \sum_{j=1}^{m} WA(3, j) / m$

where k, l, and m > 0. Then, Security Factor (F_s)can be expressed as:

$$F_s = \frac{1}{3}\left[S_{CL} + S_{AL} + S_{DL}\right]$$

or,

$$F_s = \frac{1}{3}\left[\sum_{j=1}^{k} WA(1, j) / k + \sum_{j=1}^{l} WA(2, j) / l + \sum_{j=1}^{m} WA(3, j) / m\right] \quad \ldots \quad (1)$$

The value of F_s can range between 0 and 1 i.e. $0 \leq F_s \leq 1$

5 Case Study

The Engineering College ERP system is designed to help manage various aspects of the college such as admissions, staff details, student details, course details, transport facility, examination scheduling and results, details of fee and library resources etc. and manage portal for accessing such details. It also supports in academic planning, activity calendar, student attendance, leave management, lesson plans, and many other such aspects. Given the sensitive nature of the data it handles, the SPS framework is utilized to bolster its security.

The software development process of the ERP system is controlled by the governance i.e. the management of the college, the director, and Head of the Departments (HoDs) to achieve security objectives such as confidentiality, availability and integrity of information to its users. The managerial level consists of IT manager who helps to decide security policies and guide the development team by educating through training sessions. It also decides roles and responsibilities of the various stakeholders in consultation with governance. For example, the admission team is allowed to enroll the students and edit the details, if required, while the faculties are only allowed to view these details. Similarly, the CCE marks are input by faculties, while admission team is not allowed to view these data. The control should also illustrate that all the agencies required for software development shall be working properly. The management did not consider on risk assessments and security considerations to achieve secured software while achieving the security attributes were considered through control. Security can also be incorporated by specifying minimum security permission for the various types of users based on their roles. Hence, minimum permissions were defined for students, faculties and other roles. The separation of duties has not been considered by the managerial team. Based on the security objectives, security policies, the development team is guided by the governance on documenting security requirements, development, and testing. The efforts have not been put forth to train the system architects and the programmers.

The security features of ERP system have been identified through the security aspects layer. The mechanisms to be considered are Confidentiality, Integrity, Accountability, etc. In this regard, security requirements relate to authentication, authorization, confidentiality, privacy etc. These security attributes have not been explicitly considered during the requirements gathering stage. Further, security standards from National Institute of Standards and Technology (NIST) have been used to achieve a secured product [12][13]. The development process overlooked any of the methodologies for gathering secured security requirements with the tools like Software Security Requirements Gathering Instrument (SSRGI) and SQUARE [14][15]. Static code checkers have been used to identify common flaws in coding while traceability matrix, penetration testing and fuzz testing have been used for identification of security loopholes [16]. A security checklist for requirements, design, coding issues facilitated to achieve more secured product.

The security aspects of ERP are then implemented with the help of control mechanisms being identified. The development process followed includes

Touchpoints from Cigital which focuses on best security practices. Further, the development process also considers the prevalent security flaws as listed in CWE, CVE, OWASP etc. The quality of the software is enhanced by employing Team Software Process (TSP) and Personal Software Process (PSP) [17]. The organization did not use CMM/ ISO for quality improvement. Moreover, security metrics have not been used for quality assurance.

Based on the roles and mechanisms, ERP development process have k=12, l=6, and m=5. The weights assigned to the variable codes at the three layers are illustrated in Table 3, Table 4 and Table 5. The weights assigned are based on absence or presence of security consideration of the various identified factors of the three layers of the SPS framework as described in Section 4.Equation 1 can be used to compute the Security Factor F_s.

6 Results and Conclusion

The Section 3 establishes a mathematical model to identify the security level that can be achieved using the SPS framework. Based on the case study in Section 4, the results are summarized in Table 2.

Table 2. Security Factor from Three Layers.

Layer	Security Through Layer	Measurable Security
Control Layer	S_{CL}	0.667
Security Aspects Layer	S_{AL}	0.667
Development Layer	S_{DL}	0.600

On combining the results of the three layers, $F_s = 0.645$. It indicates that the software evaluated using the SPS framework is estimated to be 64.5% secured. The security gap mainly owes to the non-consideration of security risk assessment, absence or low training to the programmers and system architects to focus on secured software development and implementation of separation of duties. Further, the end software product shall feature the security attributes. These attributes must be considered since the inception of the development process. They can be gathered using secured requirements gathering tools. Lack of specifically gathering security requirements also owe to security gap.There is also a need to study the current security flaws so that best security and development process can be followed by the system designers and developers.CMM/ ISO and similar standards may possibly enhance the overall quality of the software. The security metrics shall support security considerations throughout the development process, thereby enhancing the product's security.

Security is an emergent property of the software that shall help to safeguard the business system against loss and theft of important data. In this paper, we described SPS framework that provide the holistic approach to security. Further, we formulated Security Factor, F_s to evaluate software product being developed considering the features of SPS framework. It can be concluded that F_s shall help determine the improvement areas with respect to the three layers of SPS framework to ensure development of secured product.

Table 3. Roles and Sample Weights of Control Layer

Variable Codes	Roles	Weights $W_A(1, j)$
C1	Manage achieve security objectives	1
C2	Deciding the policies to implement security	1
C3	Guiding the development team	1
C4	Defining the roles and responsibilities	1
C5	Ensuring that entire development team is working in accordance with the norms of secured development	1
C6	Incorporating security considerations and performing risk assessments	0
C7	Attaining security attributes	1
C8	Define minimum access permission	1
C9	Instituting separation of duties	0
C10	Provide guidance on documenting security requirements, secured development and testing	1
C11	Provide training to focus on secured development process to the system architects	0
C12	Provide training to consider security during implementation to the programmers or backend developers	0

Table 4. Mechanisms and Sample Weights for Security Aspects Layer

Variable Codes	Mechanisms	Weights $W_A(2, j)$
A1	Contribution of security attributes (Confidentiality, Integrity, Accountability etc.)	1
A2	Consideration of security attributes in the security requirements stage	0
A3	Security standards (e.g. CC, NIST etc.) to achieve a secured product	1
A4	Use of tools in security requirements gathering	0
A5	Use of tools in coding andtesting help in developing more secured product	1
A6	Security checklists for requirements, design, code issues etc. for more secured product	1

Table 5. Mechanisms and Sample Weights for Development Layer

Variable Codes	Mechanisms	Weights $W_A(3, j)$

D1	Secure Software development methodology followed	1
D2	Consideration of security flaws during development process	1
D3	Follow Team Software Process and Personal Software Process	1
D4	Use of CMM/ ISO etc.	0
D5	Use of security metrics	0

References

1. Gilliam, D., Kelly, J., Powell, J., and Bishop, M.: Development of a Software Security Asessment Instrument to Reduce Software Security Risk. In:Tenth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Washington, DC, USA, IEEE Computer Society, 144-149 (2001).
2. Khan, R., Khan, S. U., Khan, H., and Ilyas, M.:Systematic Literature Review on Security Risks and its Practices in Secure Software Development. IEEE Access, (2022).
3. M. D., Hanif, T. A., Iman, C.M., Ravie, and S., Abdolhossein: Effects of Software Security on Software Development Life Cycle and Related Security Issues. International Journal of Computational Intelligence and Information Security, 6(8), 4-12 (2015).
4. Secure Software Development Framework (SSDF), NIST, https://csrc.nist.gov/Projects/ssdf,last accessed 2021/10/01.
5. McGraw, G. and Chess, B.: Software [In]security: A Software Security Framework: Working towards a Realistic Maturity Model, InformIT. http://www.informit.com/articles/ article.aspx?p=1271382, last accessed 2008/10/15.
6. Buecker, A. et. al.: Introducing the IBM Security Framework and IBM Security Blueprint to Realize Business-Driven Security, Red books, (2009).
7. Steven, J.: Adopting an Enterprise Software Security Framework.IEEE Security & Privacy, The IEEE Computer Society, 64-67, (2006).
8. Integrated Security Architectural Framework: https://www.mtsolutions.net/wp-cotent/uploads/2020/06/IntegratedSecurityArchitecuralFrameworkWhitepaper.pdf, last accessed 2021/11/23.
9. Jaekwan, P and Yongsuk,S.:A Development Framework for Software Security in Nuclear Safety Systems: Integrating Secure Development and System Security Activities. Nuclear Engineering and Technology, 46(1),47-54 (2014).
10. Jain, S. and Ingle, M.: Generalized Software Security Framework. In:International Conference on Advance Science, Engineering and Information Technology.1(4), 413-417 (2011).
11. Rehman, S. and Mustafa, K.: Research on Software Design Level Security Vulnerability.ACM SIGSOFT, Software Engineering Notes, 34(6), 1-5 (2009).
12. Grance, T., Hash, J. and Stevens, M.:Security Considerations in the Information System Development Life Cycle, NIST SPECIAL PUBLICATION 800-64 REV. 1.
13. Guttman, B. and Roback, E.A.:An Introduction to Computer Security: The NIST Handbook, NIST Special Publication 800-12, (1995).
14. Jain, S. and Ingle, M.: Software Security Requirements Gathering Instrument.International Journal of Advanced Computer Science and Applications. 2(7), 108-115 (2011).
15. 15. Mead, N.R., Hough, E.D., and Stehney II, T.R.:Security Quality Requirements Engineering (SQUARE) Methodology, CMU/SEI, Technical Report CMU/SEI-2005-TR-009, ESC-TR-2005-009, (2005).

16. 16. Jain, S.: Involving Security in Software Development Process – A Suggestive View. In:Proc. of National Conference on Electrical, Electronics and Computer Science held at IIST, Indore, (2010).
17. 17. Davis, N., and Mullaney, J.:The Team Software Process^{SM} (TSP^{SM}) in Practice: A Summary of Recent Results, Carnegie Mellon Software Engineering Institute, Pittsburgh, (2003).