Proceeding of the International Conference on Advanced Science, Engineering and Information Technology 2011

Hotel Equatorial Bangi-Putrajaya, Malaysia, 14 - 15 January 2011

ISBN 978-983-42366-4-9

Generalized Software Security Framework

Smriti Jain[#], Maya Ingle[*]

[#] *SCSIT, Devi Ahilya Vishwa Vidhyalaya*

Khandwa Road, Indore, 452001, India

Tel.:+91 9993125858, E-mail:smritijain2791@rediffmail.com

**Indore Institute of Computer Applications*

Rau - Pithampur Road, Indore, 453331, India

Tel.:+91 9893278823, E-mail:maya_ingle@rediffmail.com

***Abstract*— Security of information has become a major concern in today's digitized world. As a result, effective techniques to secure information are required. The most effective way is to incorporate security in the development process itself thereby resulting into secured product. In this paper, we propose a framework that enables security to be included in the software development process. The framework consists of three layers namely; control layer, aspect layer and development layer. The control layer illustrates the managerial control of the entire software development process with the help of governance whereas aspect layer recognizes the security mechanisms that can be incorporated during the software development to identify the various security features. The development layer helps to integrate the various security aspects as well as the controls identified in the above layers during the development process. The layers are further verified by a survey amongst the IT professionals. The professionals concluded that the developed framework is easy to use due to its layered architecture and, can be customized for various types of softwares.**

***Keywords*: security aspects, control, governance, development**

I. INTRODUCTION

In order to mitigate the risks through business goals, developers and managers must understand the need to build secured systems. Understanding the need for incorporating security during software development process is not difficult, but developing such a software can be a challenge. Secured development normally implies the process of producing reliable, stable, bug free and vulnerability free software [1]. The objective of software security is to deliver a vulnerable and defect free software. On the other hand it may be possible that the software may be attacked by unauthentic manners. Therefore, the software should be able to limit the damages caused by these attacks. Also, it should be able to recover the damaged part of a software as fast as possible at the same time. The security concerns must be global in nature and must be applied at all possible locations. Thus, it is evident that the security is concerned more about process than the product. It is apparent to integrate the organization's security efforts during the software development process. Security when implemented during the process will result in a secured product. Software security is also the result of many activities carried out by stakeholders and the processes followed by them. Thus, the security process is a fusion of three elements viz. People, Process and Technology (PPT) thereby producing a secured software system [2].

It has been observed that a framework developed to produce a secured software uses twelve practices. These practices are mainly divided into four domains: such as Governance, Intelligence, SDL Touchpoints and Deployment. This framework has been used to develop a maturity model theoretically [3]. Security framework also focuses on governance; business process automation and integration; federation; and containment and defense. It has been focused on the integration of these terms into strategies and processes of business organizations to achieve a sound security system. When the framework is used with business-driven approach, it helps tie all the strategic elements like data, lines of business, corporate and regulatory guidelines, and perimeter defense [4]. In literature, it is also found that a framework (extension of PPT framework), includes governance additionally as all the security solutions become ineffective without the support of senior management. Inclusion of governance also helps to align software security with the business practices [5]. Enterprise Software Security Framework is a two layered framework focusing on who,

what and when structure. At executive level, framework consists of governance, planning security strategy and defining roadmap whereas at the application portfolio level, it defines analysis, training, security, policy, infrastructure development, application development and quality assurance [6]. Integrated security IT framework is a result of integrating security framework and security policy domain elements with risks. The security framework consists of mainly industry standards, policies, procedures and security services. They help in defining and implementing requirements and measure success. Security policy domain elements like data protection, risk management, data classification, system development lifecycle security etc. are derived from industry best practices. The integrated framework is developed on the basis of security policy domain elements that helps decision makers to prioritize projects while addressing security [7]. It is evident from the literature that there exist many frameworks for secured software products but these frameworks either are domain specific or has some limitations. Thus, there is a need of generalized framework to achieve a secured product.

In this paper, we propose a generalized framework that enables security to be included in the software development process. Section II elaborates the proposed software product security framework (SPS Framework). This framework incorporates the process management including various security aspects so as to produce more secured product. Section III provides the methodology for data collection and analysis to validate the proposed framework by the help of case studies. Finally, we discuss the results and conclude in Section IV.

II. PROPOSED SECURITY FRAMEWORK

The generalized framework for the development of secured software is organized in a three layered structure as depicted in the Figure 1. The three layers are control, security aspects and development. These layers are explained in the following subsections.

A. Control

The control layer is the top layer of the software security framework as shown in Figure 1. Controlling not only keeps unwanted people out of the system, but also gives right access to right kind of person. It illustrates the managerial control of the entire software development process with the help of governance. Governance defines the roadmap for developing secured software through business objectives. It regulates the software development process by stating the practices that help organize, manage, and measure a software security initiative. Effective governance also requires incorporating risk assessments and security considerations into each phase of SDLC. It has key role in deciding the policies and guidance for implementing security in software. This will help the system designers and the developers to implement security during the process of software development.

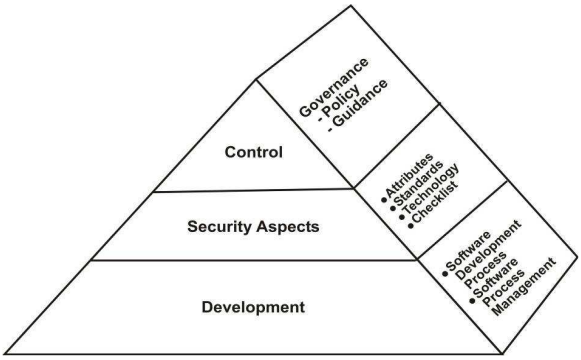

Figure 1: Software Product Security Framework (SPS Framework)

Policy - One of the most important features of governance is to define policy for roles of different users of the system with their responsibilities. When an application is deployed, the deployer will map the roles to security identities in the operational environment. The minimum permission required for a particular role name should be specified by principle of least privilege which is very important in meeting integrity objectives. Another security policy to be implemented is separation of duties for internal control in prevention of fraud and errors.

Guidance - The governance also guides the way to effectively implement specific security policies, roles and responsibilities and, security controls, documenting security requirements, documenting and validating security capabilities. The governance can control the development process of software by the various measurement and metrics. The guidance can also be provided for promoting international cooperation in the area of IT security. This can be achieved by the Common Criteria for IT security evaluation. The process of guidance also includes training the system designers and developers to help them understand the need and the implementation of security.

B. Security Aspects

The security aspect layer is the middle layer of SPS framework which helps in identification of various security features thereby recognizing the security mechanisms. The security aspects help in gathering security requirements and focus on security objectives of the software as well as organization. While designing, various aspects like security attributes, standards, technology and checklist must be considered as shown in Figure 1. These aspects are discussed below.

Security attributes – Security attributes are the security properties of secured software. Confidentiality, integrity, and availability are the security attributes important to information. Authentication, authorization, non-repudiation, and privacy relate to the people who use that information. Authentication, authorization and audit are external system functional attributes. Some other security attributes are survivability and attackability. Other attributes which are security related are reliability, availability, performability, and safety. These attributes help to achieve the security goals of software.

Security standards – The standards facilitate the development of more secured system. The commonly used security standards are given in Common Criteria (CC), National Institute of Standards and Technology (NIST), Open Web Application Security Project (OWASP) and the Web Application Security Consortium (WASC), System Security Engineering Capability Maturity Model etc. Another commonly used standard is ISO 17799 which is an Information Security Management Standard.

Security Technology - A number of tools like Software Quality Requirements Engineering (SQUARE), Threat Analysis, Attack Trees, STRIDE etc. help design security in software during the development cycle. These tools focus on requirements gathering, designing, coding and testing. The other tools to handle vulnerabilities during coding are static and dynamic code analyzers. To ensure that the resultant software is secured, black box security testing (including fuzz testing), white-box testing, penetration testing etc. must be conducted.

Security Checklist - Although brainstorming, experience and knowledge are the best way to gather security requirements, a formal approach to incorporate security is important. The checklist can cover the areas of requirements and specification, design and code issues, and maintenance and decommissioning of software and systems.

C. Development

The lower layer, i.e. the development layer, helps to implement the security aspects with the help of control layer during software development. This layer mainly focuses on firstly, the software development process, and secondly, on the software process management to achieve secured software as shown in Figure 1.

Software Development Process - The attributes, standards, tools and checklists can be applied to the software development process which is being followed by the organization. The software development methodology followed by the organizations can be either linear or iterative in nature. The iterative development methods have feedback property that helps to uncover issues early before the problem can lead to disaster. Team Software Process (TSP) while working in coordination with Personal Software Process (PSP), can improve the levels of quality and productivity of the software project development team. Thus, the software development method followed can also be helpful in achieving secured product.

Software Process Management – The software development process can be improved by the software process management models and the software metrics. Capability Maturity Model (CMM) and ISO 9001 focus on how well the organizations follow the well defined software development processes. The other management processes to improve software development include ISO 15504, ISO 9000, ISO 15504, also known as Software Process Improvement Capability Determination (SPICE), V-model etc. System Security Engineering Capability Maturity Model (SSE-CMM) describes the characteristics of an organization's security engineering process that must exist to ensure good security engineering throughout SDLC. A variety of metrics are available which can help to improve the software development process like number of lines, program size, cyclomatic complexity, function point analysis, bugs per line of line of code etc. These metrics help in measuring size/ complexity, schedule, quality and cost. These metrics help to control the software development process. Software process management methods when combined with appropriate development process that can incorporate security can lead to more secured product.

III. DATA COLLECTION AND ANALYSIS

A. Data Collection

A survey has been conducted using a self-designed questionnaire to examine the research questions. This questionnaire consists of attributes under control, mechanisms describing security aspects and development; and general information. We have gathered the information from various levels IT professionals viz. directors, senior managers, project leaders etc. A sample of 34 thus comprised of 20 IT professionals with moderate and 14 IT professionals with high experience. Based upon the data, percentage analysis was conducted to validate the questionnaire.

The need of control layer, security aspect layer and the development layer in the security framework had been accepted by all the respondents. Professionals agreed that the various functions of control layer as discussed in Section II-A, are essential to achieve a secured product as shown in Figure 2. Some of the roles like proper working of all the agencies, specifying minimum permission, separation of duties, and providing training to the system designers and developers, is accepted by less than 60% of the IT professionals. The reason for low acceptance is likely due to rejection of the fact that the training can help the system designers and developers, avoid the security flaws. The security aspects like security attributes, standards, technology and checklist, as discussed in Section II-B, are very basic requirements and are important for a secured software. Mechanisms describing the security aspects layer are also considered by more than 75% the respondents as shown in the Figure 3. The development mechanisms like type of software development process followed and use of Team Software Process (TSP), Personal Software Process (PSP), CMM/ ISO, metrics etc. (as discussed in Section II-C) were considered helpful for the development of a secured process by the respondents as shown in Figure 4.

B. Case Studies

On the basis of the survey, the data is analyzed and discussed in the form of case studies for moderate and high experienced IT professionals.

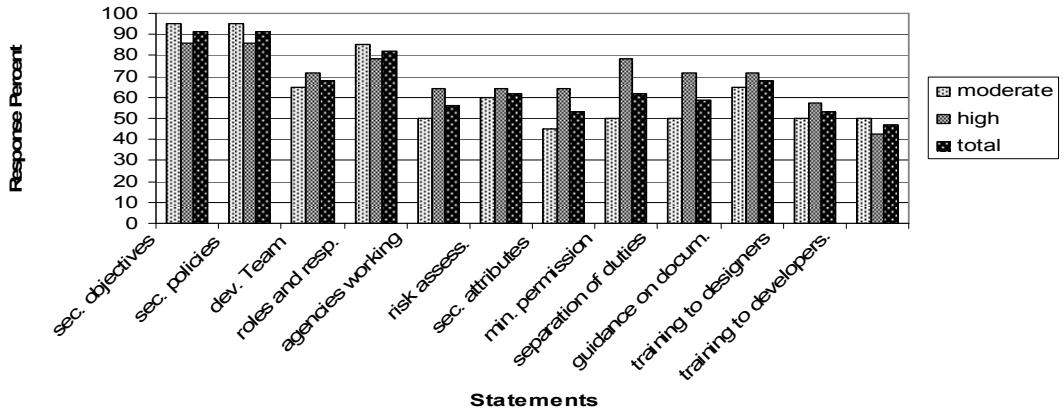

Figure 2 : Perception About Attributes Under Control

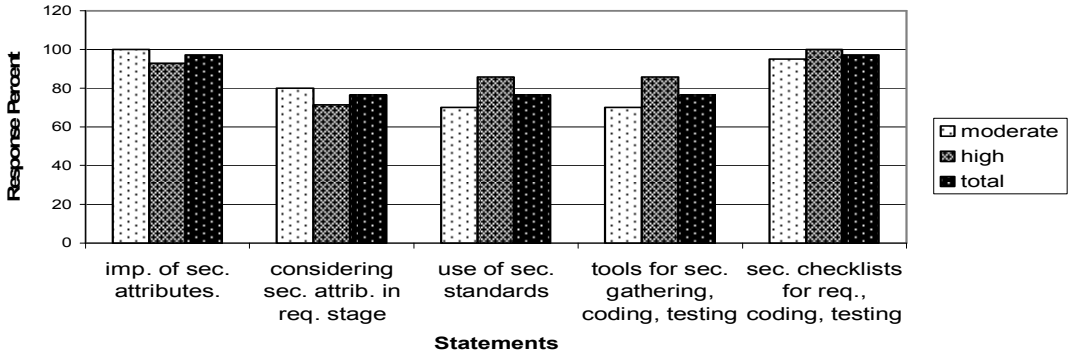

Figure 3 : Perception About Mechanisms Describing Security Aspects

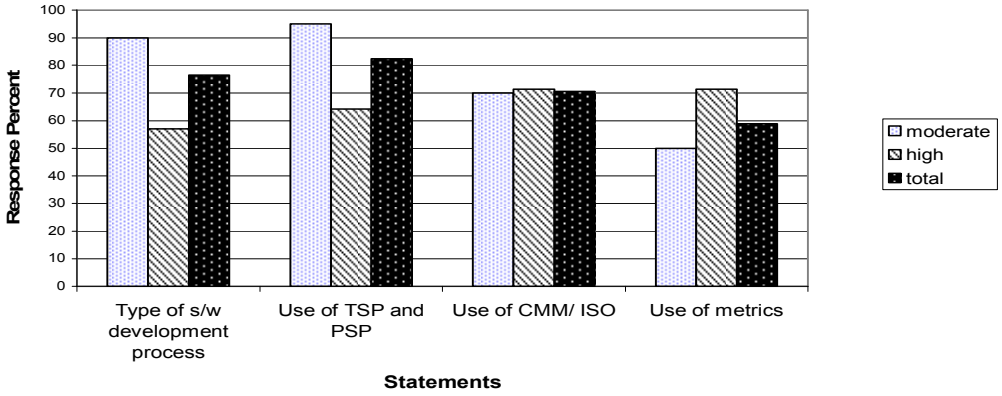

Figure 4: Perception About the Mechanisms for Development to Include Security

Case I – Moderately experienced IT professionals – The control layer is accepted as it is able to assign roles and responsibilities, provide access control and identify security requirements thus helping to incorporate security during the development. The roles like ensuring that all agencies are working properly, achieving security attributes, specifying minimum permission, implementing separation of duties, providing training to the system designers and developers are not validated while others are validated by the professionals. The reason for negative response could be the lack of knowledge of the IT professionals in the field of software security. Hence providing proper training to the developers and designers is vital. All the mechanisms describing security aspect and the development layer except the use of metrics had been accepted by the IT professionals and the same are described in the Section II-B and II-C. The use of metrics during development can help the governance to achieve the security attributes as described in security aspects layer.

Case II – Highly experienced IT professionals – The highly experienced group of IT professionals accepted all the roles of the control layer except providing training to the developers and designers to produce more secured software. The reason could be the lack of awareness of implications of the need of security training. Only software

testing methods are being emphasized as tools to identify the vulnerabilities. Security testing does not remove the flaws from the software being developed but it is an attempt to fill the gap between system developed and actual operation of these systems. According to the highly experienced IT professionals, the type of development method followed does not affect the security. But, the type of development process followed, which provides feedback mechanisms like iterative methods, could help accomplish security in a software product. Also, the security mechanisms when incorporated during the development method followed can also help attain security.

IV. RESULTS AND CONCLUSION

There are a number of differences in the views of moderately and highly experienced IT professionals while considering the aspects of the three layers of SPS framework. This has reduced the overall acceptance of the number of features like ensuring all the agencies are working properly, achieving security objectives, guidance on documentation (refer Figure 2), and use of metrics as shown in the Figure 4. These aspects have been given high importance by more than 60% of the highly experienced IT professionals. This may be due to more exposure in the field of security aspects. The need of security training to system designers and developers is given importance by approximately 50% of the respondents (refer Figure 2). The proposed framework is practical and can be easily customized to suit the development of different kinds of software.

In this paper, we have proposed a generalized framework that includes software security within the software development process to produce a secured product. The framework is divided in three layers namely; control, aspect and development layers. Security is included with the help of governance, security attributes, technology, standards, and checklist in the software development process. The need of the control, security aspects, and development layer had been accepted by all the IT professionals. The control layer can help assign the roles and responsibilities, security requirements, secured architecture of the software. The aspect layer can help achieve various security aspects viz. confidentiality, integrity and authenticity by attributes, standards, technology and checklist. This layer could help accomplish the control as defined in previous layer and the security objectives. The development layer with the help of appropriate process selected can incorporate the security aspects to produce more secured software. We have further verified the framework with the help of a survey.

This layered framework will help the development team to incorporate security in the product systematically, to analyze any non-conformity, if exists. It will be also helpful to implement security policies into secured software so as to produce the highly secured quality products. Development team who use this framework can reuse its design and implementation. The implementation of the various layers can be done as per the aspects discussed. By reusing the framework developers can customize the development methodology to include security in the development process.

REFERENCES

[1] Glyn Geoghegan, A Corsaire (2004). "Secure Development Framework", [Online] Available: http://research.corsaire.com/whitepapers/040220-secure-development.pdf, 05 April 2004.

[2] (2002) "Defining an enterprise-wide Security Framework", *Network Magazine – Technology Decisions for the Enterprise,* [Online] Available: http://www.networkmagazineindia.com/200211/guest.shtml.

[3] Gary McGraw and Brian Chess. "Software [In]security : A Software Security Framework: Working towards a Realistic Maturity Model". [Online] Available: http://www.informit.com/articles/article.aspx?p=1271382, Oct 15, 2008.

[4] Gary J. Evans. "IT Security: Understanding the Issues, Creating the Solutions." *IT Security*, IBM security ebook. pp 3-5.

[5] Abdelwahab Hamou-Lhadj and AbdelKrim Hamou-Lhadj. "A Governance Framework for Building Secure IT Systems". *International Journal of Security and Its Applications*. Vol. 3, No. 2, April, 2009.

[6] John Steven. "Adopting an enterprise Software Security Framework". *IEEE Security & Privacy*. The IEEE Computer Society, pp. 64-67, 2006.

[7] (2010) Integrated Security Architectural Framework, White Paper [Online] Available: http://www.cisco.com/web/about/security/cspo/docs/IntegratedSecurityArchitecturalFrameworkWhitepaper.pdf.

[8] Anthony Gerkis and Jack Danahy. "Software Security Governance in the Development Lifecycle: A Practical Guide", from Accenture and Ounce Labs, 2007.

[9] David P. Gilliam, Thomas L. Wolfe, Josef S. Sherif. "Software Security Checklist for the Software Life Cycle", In *Proc*. of WETICE'03, 2003.

[10] David Ferraiolo and Richard Kuhn . "Role-Based Access Controls", [Online] Available: http://hissa.nist.gov/rbac/paper/node5.html. Retrieved on Jan 2010.