Security Integration in SDLC Analysis Phase

Er. Sunil Patel
Department of computer science, CIIT Indore

Abstract
SDLC stack is a layered architecture of application or software development. Effect of any layer is reflected on all the upper layers in other words base layer outcomes are executed in upper layers. Thus if a major changes occurred in lower layer than it is reflected on all the upcoming layers of SDLC stack, in this paper we provide the concept of vulnerabilities check list scanning in analysis phase which scan the current vulnerabilities in system which is required to design. Additionally the system provides the suggestions for correct them to get fast, secure and less time and effort consuming system development.

Keywords
security, performance, SDLC, check list, bug free.

1. Introduction

Software development is a complex task in nature; the same named projects are reflect different behaviours and changed according to the time, users and environments. For example if a client wants an inventory management software for his use according to the need of his inventory the application is developed but at the same time a new customer need an inventory management software then the complete system is changed according to the clients requirements.

To provide ease in development different software process models are provided where according to the project length and environment model is selected. These models are provide a basic view for develop a software projects. These models are a combination of different activities by which quality software is developed and deployed.

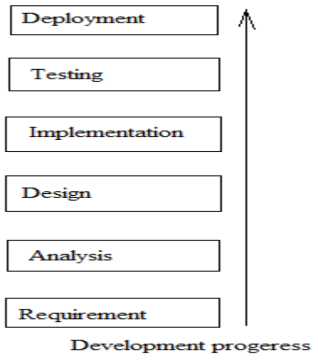

Fig 1 shows the development process

Manuscript received December 5, 2014
Manuscript revised December 20, 2014

Here the stack of SDLC is given in fig which provides the dependency of the stack elements. If requirement is gathered well then analysis and other steps are executed smoothly Else the problems are arises in all phases.

Thus here we select the analysis phase of the development where the gathered information form the end client are analysed and refined in this phase, thus the analysis has to be done appropriately and must be followed by adequate design and implementation. Too often none of the above occurs.

Analysis should be discovering what I call the REAL business requirements deliverable what’s that provide value when satisfied by the product/system how. Very often, though, what is called "analysis" actually is design, which means that the project has skipped to the how without adequately identifying what’s the how must accomplish in order to provide value.

2. BACKGROUND

In this section we provide the basic principal of the secure software development, problem with existing system and our proposed solution.

Success and quality of software depends on the amount of projects features are matched to the end client requirements. Requirement capture and analysis help in identifying stakeholders and their expectations, and capturing these expectations in a form that is responsive to analysis and implementation. Software projects fail to meet their expectations due to problems in articulation of requirements, poor quality of analysis and quite often, a lack of sufficient focus on the business perspective. A framework to remove vulnerabilities: analysis phase in SDLC will find out bugs or errors and missing requirements for analysis.

Developing” A Framework to remove vulnerabilities through analysis stage of SDLC” is very challenging because in many papers it is given theoretical way. it can be demonstrated that changes to the software engineering process can help to reduce the number of defects in new or changed software. Programmers alone do not produce software. Software engineering is a process that has many players and many steps from concept to deployment. The analysts and designers have significant implications on the quality of software being produced. This extension of inclusive security concepts is important if real changes are

to occur in software industry. Empowering on certain levels of the software engineering life cycle chain will permit old and bad habits to remain in place. A Framework to remove vulnerabilities through analysis stage of SDLC having many sub issues are important for development of this software.
Figure shows total vulnerability reported in vulnerability database from 2004 .the remove vulnerability is a very big challenge .applying secure software engineering vulnerability will be remove.

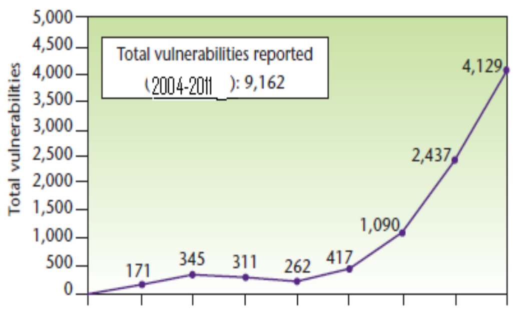

Fig 2 vulnerability report

Normally all the requirements which are useful for development of any software ,but many times requirements given from customer are not satisfactory or not complete .so here secure software engineering concept is very much important . in this project it is possible to find out the lack of requirements and all missing requirements are shown after analysis .hence for the next phase of SDLC it is very easy to take all requirements hence software development will be possible in proper manner.

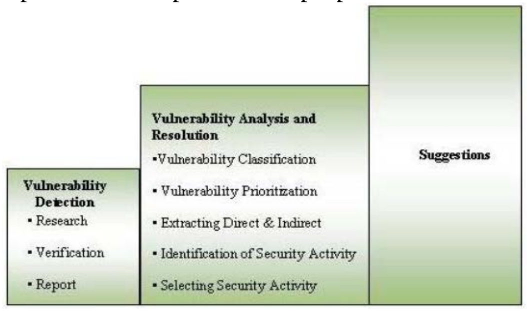
Vulnerability Detection
• Research
• Verification
• Report
Vulnerability Analysis and Resolution
•Vulnerability Classification
• Vulnerability Prioritization
• Extracting Direct & Indirect
• Identification of Security Activity
• Selecting Security Activity
Suggestions

Fig 3 Activities in Vulnerability Life Cycle

In order to prevent vulnerabilities in the software under development, the VLC is designed keeping the following objective in mind:

1. Detection of the vulnerabilities.
2. Classification of the detected vulnerabilities.
3. Identification of recurring vulnerabilities.
4. Determination of direct and indirect causes
5. Listing the activities to resolve the causes of the vulnerability or vulnerability itself.
6. Selection of the optimal set of activities.
7. Suggestions.

To achieve the above objective, the vulnerability life cycle of VLC is divided into three major phases. Each phase of Vulnerability Life Cycle performs activities as given in Figure
The objective of secure software development is to design, implement, configure, and sustain software systems in which security is a necessary property from the beginning of the system's life cycle (i.e., needs and requirements definition) to its end (retirement). Experience has taught that the most effective way to achieve secure software is for its development life cycle processes to rigorously conform to secure development, deployment, and sustainment principles and practices.
Organizations that have adopted a secure software development life cycle (SDLC) process have found almost immediately upon doing so that they have begun finding many more vulnerabilities and weaknesses in their software early enough in the SDLC that they are able to eradicate those problems at an acceptable cost. Moreover, as such secure practices become second nature over time; these same developers start to notice that they seldom introduce such vulnerabilities and weaknesses into their software in the first place.
The goal is to minimize the remaining defects that lead to vulnerabilities.
Scope of this work is vulnerability detection, analysis and resolution can be shown on report. Development of software, all requirements are helpful for next phase. Due to consideration of security from initial phase of software development costing of software will less. Because of this new methodology less time is necessary.
Due to this manual work is reduced because software fed all requirements to the next phase of SDLC, Due to this methodology analyst having less work because of fulfilment of all requirements.

3. Proposed work

In this project requirement objects are entered one by one and checking out either this given requirements are ok or some requirement missing. After putting requirement object there are several link on main page .Detection, requirements, analysis, text report and image report are given on main snapshot.

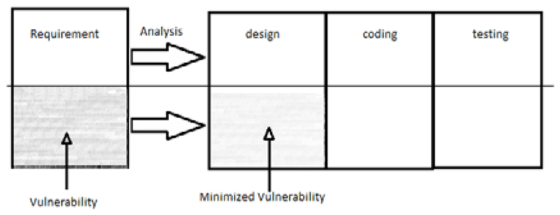

Fig 4 Detect & Resolve Vulnerability by Framework

First go on requirements. Put the requirements with sub requirements if necessary after putting requirement object goes on detection. It will check the checklist and shows Y/N .In analysis option shows detail analysis means which is missing, which is unnecessary .It also shows text and image report. In this way it removes vulnerabilities in analysis phase in SDLC.

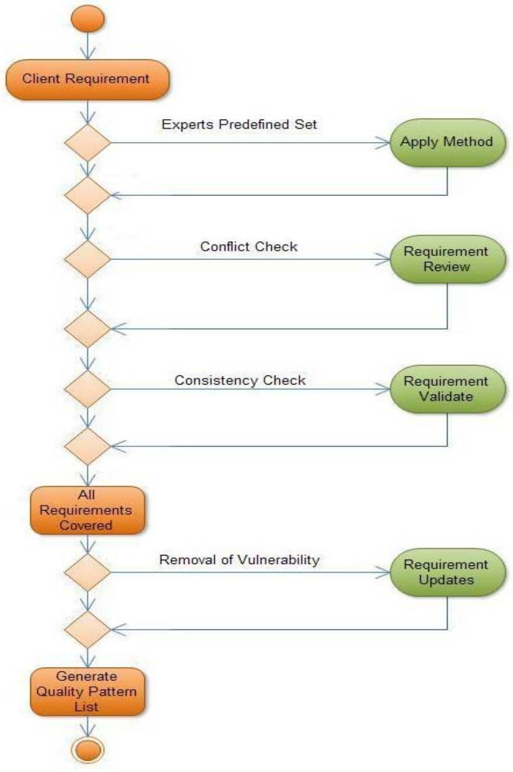

Fig 5 Vulnerability Removal by Framework

This aims to minimize vulnerabilities in the software. Each time output of analysis phase of SDLC is to feed the security checklist where it is verified whether the output fulfils prerequisites for security of the phase. If yes the phase is declared as secure. If not, the output is fed to vulnerability life cycle as an input. In this it is properly analyzed for the detection of the vulnerabilities. Direct and indirect causes of the vulnerability and the actions to resolve the vulnerability are determined. Then documented suggestions in the form of feedback are sending as input to the Next phase of SDLC from where the vulnerable output was received as input.

4. Implementation

To implement such kind of system that is described in the above sections, we use the visual studio IDE (integrated development environment) which provide the rich classes to implement any hard problem in easiest way. For implementation point of view we create a list of client requirement and similarly a check list of all necessary aspects and vulnerabilities.

Using the SQL server database we cross check conflicts, consistency and after that filtered vulnerabilities form the project documents which is passing to the design of the system.

Here we provide some key methods and functions by which we scan and remove the vulnerabilities form any project requirements. The previous version of the projects and methodologies are limited for some particular type of projects here we implement a generalize this concept by provide suppling the additional check list for system, here user can select the check list of any project and remove vulnerabilities from the system required to develop.

Table 1 vulnerability checklist

S. No	method	Description
1	Requirement()	This method is user defined method by which requirements are listed in a user interface
2	Search()	This function is used to search the main requirement and their sub requirements and prepare the list of the search objects in database
3	Vulnerability Analysis()	Using this method system scan Vulnerability and provide the list of gaps and conflicts
4	Suggestion()	This function is executed to prepare a list of suggestion and corrections that are predefined in a database
5	Report()	This function generate the complete report for input and output, with Vulnerability and suggestions in a text file
6	Vulnerability Classification()	This function categorize the different Vulnerabilities found under the project requirement

5. Conclusion and Results

Our proposed work is to provide the security and quality improvement in software development. For that purpose we make efforts and starts with the analysis of SDLC and software processes models. After analysis we found that all the process models having its own limitations and advantages. And additionally we found that the effect of changes are reflects in all the activities and software quality if any problem are remain in any phase of SDLC.
thus we starts working with analysis phase of SDLC where all the requirement of clients are understand and reviewed for making design. so if we improve the initial step then we found the best quality of software and all the remain steps are go smoother. The main advantage of this is reducing efforts and time in application development; additionally it helps on testing and reduces the testing and debugging efforts.
during implementation of the proposed system we found some limitation which is provided by the author in [1]. which is recovered and we implement a generalized software analysis tool for improving QoS and security.

References

[1] A Framework to Detect and Analyze Software Vulnerabilities -Development Phase Perspective-, International Journal of Recent Trends in Engineering, Vol 2, No. 2, November 2009
[2] NEXT GENERATION SOFTWARE SECURITY THROUGH TESTING STAGE OF SDLC, Vidyabhushan A. Upadhye1 and Shashank D. Joshi2, International Journal of Computer Science and Communication Vol. 2, No. 2, July-December 2011, pp. 311-313
[3] New Approach for Predicting Vulnerability at Design Stage for Object Oriented Design, ISSN : 2229-4333(Print) | ISSN : 0976-8491(Online) www.ijcst.com, IJCST VOL. 2, ISSUE 3, SEPTEMBER 2011
[4] Baking in Security During the Systems Development Life Cycle, CROSSTALK The Journal of Defense Software Engineering March 2007
[5] REVIEW ON COMMON CRITERIA AS A SECURE SOFTWARE DEVELOPMENT MODEL, International Journal of Computer Science & Information Technology (IJCSIT) Vol 4, No 2, April 2012, DOI : 10.5121/ijcsit.2012.4207 83
[6] An Integrated Approach to Software Process Improvement at Wipro Technologies: veloci-Q V. Subramanyam Sambuddha Deb Priya Krishnaswamy Rituparna Ghosh March 2004
[7] Secure Software Development Life Cycle Processes: A Technology Scouting Report, Noopur Davis, December 2005 Software Engineering Process Management
[8] Topological Vulnerability Analysis: A Powerful New Approach For Network Attack Prevention, Detection, and Response, Sushil Jajodia and Steven Noel Center for Secure Information Systems, George Mason University 4400 University Drive, S-113 85 Fairfax, Virginia, USA
[9] A Framework for Identifying Software Vulnerabilities within SDLC Phases, (IJCSIS) International Journal of Computer Science and Information Security, Vol. 9, No. 6, 2011
[10] SOURCE CODE ANALYSIS TO REMOVE SECURITY VULNERABILITIES IN JAVA SOCKET PROGRAMS: A CASE STUDY, International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.1, January 2013
[11] Studying Software Vulnerabilities, Dr. Robin A. Gandhi, Dr. Harvey Siy, and Yan Wu, The University of Nebraska at Omaha, The Journal of Defense Software Engineering September/October 2010 http://www.ieice.org/eng/shiori/mokuji.html