In Security Estimation Importance of Design Phase

Sonali P. Kadam¹, Shashank. D. Joshi² and Atul V. Dhaigude³
¹Bharati Vidyapeeth’s college of engineering for Women,  
²Bharati Vidyapeeth’s college of engineering ,  
³Marathwada mitra mandal’s Institute of Tech ,Pune

ABSTRACT
Software development industry is facing a problem in delivering secure software. Software engineers, project managers, and decision makers remain under stress on account of their inability to deliver secure software. It is going to be a cumbersome process and an expensive affair for any software developing unit to implement security features in the previously developed application programs. In this paper we describe the need for implementing security in earlier stage of development and importance of design phase with respect to security.

1.INTRODUCTION
Security is similar to the concept of safety, confidentiality, and reliability. We can not add security once the software is ready or completely build.

The reasons may be following: a) It is not possible to restructure and rewrite an application program in its entirety with security modules  b) Additional investment in making the previously developed software secure is not really advisable and c) It is not suggested to retrain software developers to be security experts. It is prerequisite for all software system to perform properly under the presence of malevolent approaches. Software system should be capable of functioning proactively, meaning thereby, it should be self-defensive. A critical look at literature available on software security estimation reveals unavailability of any standard approach to be followed to estimate security.

2.INTEGRATING SECURITY INTO THE SOFTWARE LIFE CYCLE
Number of security loopholes and vulnerabilities exists due to the defects of security architecture and security mechanism. Hackers and attackers do not create security loopholes; rather they target the weaknesses in the software and exploit them. In order to maintain the software security during the developmental stages, hacking should be made too difficult.

The purpose of making the software secure is to protect the software from all kinds of attacks, errors, bugs, threats, viruses and vulnerabilities.

The key elements of a secure software life cycle process are-
1. security criteria in all software life cycle checkpoints (both at the entry of a life cycle phase and at its exit)
2. adherence to secure software principles and practices
3. adequate requirements, architecture, and design
4. secure coding practices
5. secure software integration/assembly practices
6. security testing practices that focus on verifying the dependability, trustworthiness, and sustainability of the software being tested
7. secure distribution and deployment practices and mechanisms
8. secure sustainment practices
9. supportive tools
10. secure software configuration management systems and processes
11. security-knowledgeable software professionals
12. security-aware project management
13. upper management commitment to production of secure software

Thus there is need to implement security in the early stage of development to ensure that the analysis and design have favorable internal properties that will lead to the development of security end product. This would give developers an opportunity to fix problem, remove irregularities and nonconformance to standards, and eliminate unwanted complexity early in the development cycle. This would significantly help in reducing rework during and after implementation, as well as designing effective test plans and better project and resource planning.

Based on this practice one questionnaire was made for the industry people to get their view and response regarding this issue, the questionnaire was distributed to 35 different IT professionals like system analysist, project managers from different IT companies. And the statistical analysis of these questionnaires is as follows

1. On scale of 10 rate cost of security failure on your last 5 project

Copy Right © INDIACom-2011 ISSN 0973-7529 ISBN 978-93-80544-00-7
2. Which is the following is the major reason of software project security failure
   A inefficient design   
   B inefficient coding   
   C inefficient testing   
   D inefficient requirement

3. What is consequence of software security failure
   A Delayed project   
   B Increase cost   
   C Failed project   
   D All of the above

4. In which phase of SDLC do you practice security
   A Design   
   B Implementation   
   C Testing   
   D Deployment

Survey on security estimation reveals that changes made at final stage is very expensive and difficult. Therefore, security
estimation process should be integrated well in advance, preferably at software development life cycle. In addition design phase is the first step towards problem domain to solution domain. In design phase, software architecture is available. Therefore, it is the most appropriate phase to estimate security of the software. Security estimation of software in this phase will assist to protect software from damages.

3. HOW TO IMPLEMENT SECURITY IN DESIGN PHASE ?

To minimize vulnerabilities and achieve target level security, quantification of security is necessary. Unfortunately, quantities estimation of security in design phase is largely missing. We normally do quality estimation of design once it completed, but we are not doing same thing for security. Design properties are tangible concepts that can be directly assessed by examining the internal and external structure, relationship, and functionality of design component, attributes, methods and classes. And evaluation of a class definition for its external relationship (inheritance type) with other classes and the examination of its internal component, attributes, and method reveals significant information that objectively capture the structural and functional characteristics of a class and its objects.

The design properties of abstraction, encapsulation, coupling, cohesion, complexity and design size are frequently used as being representative of a design security characteristics in both structure as well as object oriented development. Messaging, composition, inheritance, polymorphism, and class hierarchies represent new design concepts which have been introduce by the object oriented paradigm and are thus vital to the security of an object oriented design.

4. CONCLUSION

Software design is backbone of any software design serves well as communication medium between the designer and the user on the one end and act as a basis for the implementation on the other end. Design as an important stage spanning the hole software lifecycle, not only the software development but also for redeveloping legacy system. It is concern with accurately mapping the requirement from the analysis stage to logical model for implementation. The security estimate at software design heavily effect of the final product.

REFERENCES