Software Security Issues: Requirement Perspectives

Nikhat Parveen, Md. Rizwan Beg, M. H. Khan

Abstract— With the fast growing of software development life cycle, software engineering under a huge pressure to deliver the business requirements without paying too much attention to the security issues that the software might encountered. Many applications are outsourced too where the application development lacks strong integration of software security. For this reason security issues becomes a problem for the growth of a business and availability of requirement by its customers. Therefore, security should be tightly bound during the software development life cycle, right from the beginning. This paper discusses 21 issues related to security with respective to requirement that can eliminate vulnerabilities at the early development of life cycle. By keeping these issues at back of mind, working people in the area of software security can able to build more secure software and also the goals of security can easily be identified..

Index Terms— Software security, software development life cycle, security requirement issues, confidentiality, integrity and availability.

1 INTRODUCTION

THE most important question arises related to security that "What makes secure software different?" Many researchers realize that security is an innate property of the software that has to be built in. Most of application lacks security today. The traditional practices are even no more effective for developing secure applications. The IT services companies are also improving their SDLC with the security at the very early stages. The current scenario is that the application demand tight software security embedded inside to prevent hackers getting in, due to this reason addressing security earlier in the lifecycle will help to mitigate the risks of application security attacks.

Maintaining a high level of security is not so simple, to endorse it the security issues has to be taken under higher priority. Unlike functional requirement, non-functional requirement of application, security and its performance are always given lower priority. If the importance of these non functional factors fails it will directly loss the customer value and their faith. If the researchers and developers tries to identify and analyze the reason behind the cause of security breach, they generally put blame entirely on virus attack, denial of service, spam mail etc. If we analyze from the depth of our knowledge we will found that the facts of the most important factors in software security breach, is, bad software which is actually behind every security problem and malicious attack. By identifying and targeting each individual security threats and providing solution for those attacks and if we also put focus on the security aspect of software, we surely can build a more robust and reliable system in totality.

Since the researchers and developers have done significant work in the field of integrating security throughout the software development life cycle, 'right from the beginning', still a major portion of work needs to be carried out in order to made software more secure and reliable. In this paper we intend to propose twenty one security issues which if practically applied from the very beginning of software development life cycle i.e., from requirement analysis phase, it will definitely contribute in secure and reliable development of software. The paper is organized as follows: in section II 'Software Security' is defined, and, 'Security Issues' is listed, section III describes 'From Security Issues to Security Requirements and section IV focus on' Impact of Security Issues in Design, Implementation and Deployment of software, with conclusion.

2 SOFTWARE SECURITY

Software security is an approach which protects from risk and maintains security failures in order to increase system reliability [1]. The main objective of software security is to have knowledge about the attacker and to foresee attacker's motive and perception. Developing secured software is not an advantage but it has become a necessity for software organization. Traditionally, software development is thought of as building software that works under normal conditions. But when the security aspect is clubbed with building software, the designer and developer focal point becomes attacker's perspective and 'how they can become a threat to the software'.

Software security can be thought of as building software which performs under intentional and unintentional malicious attack [2]. The software security should exhibit ability to defend itself and the system from the attacker's exploitation and misuse of software security loop holes. Software with build-in security should reflect features like predictable execution, trustworthiness, conformance, attack resistant, attack tolerant and attack resilient.

Security should begin at the requirement level and it must cover all the characteristics that secure the process [3]. Security is the degree of resistance to, or protection from attack. In order to elicit security requirements one's should have the knowledge regarding security issues. The most common issues are CIA (Confidentiality, Integrity and Availability) [4,5].

- C : Confidentiality is prevention of unauthorized disclosure of information.
- I : Integrity is prevention of unauthorized modification of information.

• A : Availability is prevention of unauthorized withholding of information.

Confidentiality ensures that only authorized user can access regardless of where the information is kept and how it is accessed. This can be maintained by mechanism like access control, password, biometrics, encryption, privacy and ethics whereas integrity is to safeguard the accuracy and completeness of information and processing methods from being changed intentionally, unintentionally, or accidentally. Integrity needs to be maintained for ensuring privacy, security and reliability of data and information. The mechanism to maintain integrity is configuration management and auditing. Similarly availability is to ensure access of information and related assets for authorized users whenever needed. Availability can also be maintained by mechanisms like data backup plan, disaster management or business continuity.

Still the researchers and developers found that there are various issues regarding security. The best way to spread software security knowledge is to trained software development staff on serious security issues. Researchers have done tremendous job in this direction but there are various issues that can be addresses. On the basis of literature review, the issues are proposed and listed below.

1. Access Control: selective restriction of access to a place or other resource
2. Accountability: responsibility to someone or for some activity.
3. Accuracy: the quality of being near to the true value
4. Assessment /Evaluation: The classification of someone or something with respect to its worth.
5. Auditability: The ability to achieve accurate results.
6. Authorization: Permission to access a resource.
7. Availability: The quality of being at hand whenneeded.
8. Awareness: A state of elementary or undifferentiated consciousness.
9. Confidentiality: Discretion in keeping secret information.
10. Consistency: A harmonious uniformity or agreement among things or parts.
11. Error Classification: A wrong action attributable to bad judgment or ignorance or inattention. The act of distributing errors into classes or categories of the same type.
12. Excellence: The possessing good qualities in high degree.
13. Flexibility: The quality of being adaptable or variable.
14. Fortification: A defensive wall or other reinforcement built to strengthen a place against attack.
15. Identification /Authentication: The act of identify fact or reliable information.
16. Integrity: An undivided or unbroken completeness or totality with nothing wanting.
17. Interoperability: The ability to exchange and use information.
18. Non-Repudiation: Non refusal to acknowledge.
19. Prevention: The act of preventing.
20. Privacy: The state of being concealed or hidden.
21. Unambiguity: Clarity achieved by the avoidance of ambiguity.

By analyzing the above listed security issues, at the early stage of software development, it is found that this security issues will elicit the requirement of the software in better way and helps to improve the production of software security.

3 From Software Security Issues to Security Requirements

Security, like beauty, is in the eye of the beholder. The specific security requirement particularly is different for business purpose, selfie, user preferences and/or defense perspective. The TCSEC [6] Glossary defines security requirement as “the set of laws, rules, and practices that regulate how an organization manages, protects, and distributes sensitive information".

Security requirement can be classified into three groups: They are

- ***Functional Security Requirement***: It is a property of a system required to check condition or capability needed to control or limit the fulfillment of requirements.
- ***Non-functional Security Requirement***: It is a property of a system required to ensure fulfillment of requirements with respect to abuse or misuse conditions.
- ***Derived Security Requirement:*** It is an implicit from Functional/non-functional state of requirements.

A security requirement is a manifesto of a high- level organizational policy into the detailed requirement of specific system.

Fig 1 shows the relationship between classifications of Security Requirement and each group has its own security issues.

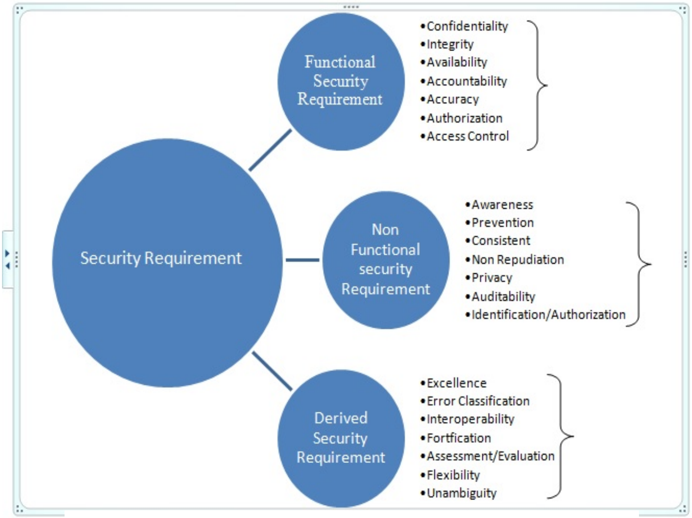

Fig 1 Software Security Requirement Issues

During development of software, developers must review the above listed issues in order to reduce vulnerabilities in the system. If the software engineers have these issues at the back of their minds during eliciting requirement for software development, it will certainly reduce the re-work for the developer and improve the quality of the software product. The issues are described in detail from requirement perspectives.

1. ***Issue related to Access Control***: It provides a form of authenticated user accesses only what they are authorized for [7]. This issue suggests that accesses to resources and services should be permission based. If the authenticated user is permitted/allowed to access resources and services then these permitted users should not be denied access to that services that they are legitimately expect to received. To produce secure software, it is mandatory to implement access control at requirement perspective.
2. ***Issue related to Accountability:*** Accountability is a functional requirement that plays a vital role for maintaining number of logs for any task/ activities or action performed to accomplish a work in order to maintain security of prevention [8]. It involves tracking of activities of processes and maintains log details. Security can be maintained through audit transactions which help to determine the attacker or source of attack if occurred in software [9, 10].
3. ***Issue related to Accuracy:*** For any secured and reliable software system, it is necessary to produce accurate requirement for the system [3]. The issues relate that the software development team should perform various activities such as scenarios, interviews, etc. in order to produce correct and accurate requirement for the software. Accuracy can also be considered in terms quality factors such as speed, time.
4. ***Issue related to Assessment/Evaluation:*** Assessment is an inherent classification that can be applied on process to get a quality software product. These issues suggest that each and every process should be classified in terms of requirement which ensures that necessary security controls are integrated during requirement, design and implementation of a software product [23].if a assessment of a product is done properly it means consistency is maintained which means quality a subset of security. Assessment or evaluation if done considering the current security environment can help the software developer to analyze and measure the level of security implementation in their software product versus industry standards and best practices [12].
5. ***Issue related to Auditability***: Auditing in security is a feature to achieve accurate result. This issue suggests that Auditability must be implemented to judge the accountability feature of software security and assist in redesign a full proof security policy and procedures for implementing a secure software system [3]. It helps the security auditor to thoroughly understand the flow of information and develop a plan for properly securing the system.
6. ***Issue related to Authorization:*** It is a process for assessing the security of a system by verifying the specified operation performed by authenticated person. Authorization can be implemented through access control where access is authorized. [11].
7. ***Issue related to Availability:*** *I*t is a property for ready-to-service in corrective manner [13]. It needs to be a security goal as the loss of availability is referred to as "denial-of-services". This issue states that a balanced approach needs to be maintained between security and availability of requirement that should be highly secure and available all time. By maintaining availability the system is responsible for delivering; storing and processing information are accessible when needed.
8. ***Issue related to Awareness:*** The issue of awareness in requirements of software security requires more adequate research treatment. Requirements as expressions of interests, goals and needs in software development are rarely physically located [14]. This issue relate with constant acquisition of new information and updation of existing knowledge related to security aspect for development of software. This can be resolved by providing security awareness program for training software development team on critical software security issues [5, 6].
9. ***Issue related to Confidentiality:*** Prevention or protection against access of information by unauthorized person can be defined confidentiality. This issue suggests that confidentiality should be maintained by ensuring that information is not accessed by any unauthorized person [10]. It can be resolved by providing assurance that the information ***is*** shared only among authorized users [13].
10. ***Issue related to Consistency:*** Consistency is key feature when evaluating security effectiveness and performance for eliciting requirement of software. It is advisable for requirement engineer to maintain consistent requirement protocol or standards for securing the software [15, 16].
11. ***Issue related to Error Classification:*** Due to bad error handling and lack of proper knowledge regarding errors, security could not be maintained. Software developers and software security practitioners should be concerned about the various errors which create problems leading to soft-

ware vulnerability. This issue suggests that the error should be categorized and classified according to the security rules. If the error perceived, it should be removed at the early stage of software development.

12. ***Issue related to Excellence:*** The possessing of good qualities in high degree by any software refers to excellence. This issue suggests that the security is a subset of quality. Hence in order to achieve security in totality, the quality of the software should be of highest standards [17].
13. ***Issue related to Flexibility:*** Flexibility in software security can be defined in terms of quality of being adaptable or variable when external changes occur. This issue suggests that the various requirements concerning security should not be rigid and must be flexible as required.
14. ***Issue related to Fortification:*** Integrity is an important ingredient of software and it should be maintained throughout the software engineering process [13]. The issue suggests that the various process used in security engineering process should be secured in individuality and totality. Only the concerned individual should have access to the technicalities of ***software security and for the rest it should be kept as secret.***
15. ***Issue related to Identification/Authentication:*** The act of identifying fact or reliable information is termed as authentication. This issue suggests that the process of identification and authentication must be implemented to find legitimate user who log on to a system with respect to the authorized access rights [18]. Authentication can be judge by various techniques which include use of passwords***, biometric techniques, smart cards, certificates, etc.***
16. ***Issue related to Integrity:*** In terms of software security, integrity is concerned with securities that prevent unauthorized modification of information. Integrity should be maintained by ensuring information that cannot be altered by unauthorized person [10, 13]. Integrity of information provides an assurance that the data is authentic and complete.
17. ***Issue related to Interoperability:*** The ability to exchange and use information is the era of globalization. Most software is platform independent which provides interoperability. It means that software that able to exchange information directly infect the security of the software [19]. Hence this issue suggests that if one or more software are interacting together then all the software involved in the communication must be secured.
18. ***Issue related to Non-Repudiation***: In general the non refusal of any transaction is termed as non-repudiation. Any parties involved in a transaction cannot repudiate the validity of transaction [20]. The objective of non-repudiation is to validate the transaction that is committed between two parties. Non-repudiation can be obtained through digital signatures.
19. ***Issue related to Prevention:*** "Prevention is better than cure", a well known quotation, the developer must keep this in mind and build the software with security in such a way that when threat attack on software internally or externally can be safeguard and protect from infection. Security in the software should be synchronized to prevent any kind of threat from internal or external source rather than cure it.
20. ***Issue related to Privacy:*** Privacy can be seen as an art of being concealed or isolated from the presence or view of others [8]. Security and privacy issues now become major concerns in the requirement of software systems. The issue suggests that privacy can ensures individual's right to control what information is collected, how it is used, who has used it, who maintains it, and what purpose it is used for [21]. Privacy protection as a tool of security can be implemented by designing and enforcing sound privacy and data protection laws and technologies [22].
21. ***Issue related to Unambiguity:*** Unambiguity always been issues for software security, this issue suggests that the implementation of software security will achieve clarity, by avoidance of ambiguity.

The above listed 21 issues related to requirement perspective suggest that if these issues will be improved properly at the early stage of software development, it will certainly reduce the re-work of developer and also helps to meet the security goals of the software.

4 Impact of Security Issues in Design, Implementation and Deployment of software

Security should be consider from the requirement phase, all the personnel from requirement engineers to software developers must be aware of most recent software security issues, especially functional requirement security issues and non-functional security issues. The awareness of security issues should be more technical and in-depth for SDLC team members.

In order to achieve the security goals for the software, the security will be tightly bound from requirement phase to deployment phase of software development life cycle. If the security issues are properly followed and removed by requirement engineer, it will help the system software designer to design more secure software and the programmer will be able to produce secure code. Following the above mentioned security issues, the implementation engineer will able to configure and run software more securely. The deployment engineer will able to deploy more secured software in open environment.

5 Conclusion

Security is not something that is addressed at the end of the product lifecycle nor is it a specific objective that occurs during project execution. Security must be everywhere. Traditionally, security issues are first considered during the Design phase of the software development life cycle once the software requirement specification has been frozen. The paper tried to produce issues related to securities and forced requirement engineer to address security at requirement phase so that the vulnerabilities can be mitigate at the early development of software. Each security issues must be resolved so that the security must be tightly bound from the beginning to the deployment of secure software.

REFERENCES

[1] Chandra, S, and R A Khan. "Software Security: A Quantitative Approach." CSI Communication: 19-23. August 2010.

[2] Julia H. Allen, Sean Barnum, Robert J. Ellison, Gary McGraw, Nancy R. Mead: Software Security Engineering: A Guide for Project Managers, Addison Wesley Professional, 2008, pp 6-8.

[3] Parveen, Nikhat, Rizwan Beg, and M. H. Khan. "Integrating Security and Usability at Requirement Specification Process." International Journal of Computer Trends and Technology (IJCTT) 10: 236-240.

[4] G.McGraw, "Software Assurance for Security",IEEE Computer 32(4), pp. 103-105(April,1999).

[5] http://searchwarp.com/swa268042.html

[6] Sterne, Daniel F. "On the Buzzword "Security Policy"." IEEE. (1991)

[7] Keromytis, Angelos D, and Jonathan M. Smith. "Requirements for Scalable Access Control and Security Management Architectures." ACM Transactions on Internet Technologies, Vol. 7.No.4 (2007).

[8] Breaux, Travis D., Annie I. Anton, and Eugene H Spafford. "A distributed requirements management framework for legal compliance and accountability."ScienceDirect Computers &Society. I.I0 (2008):

[9] "SECURITY REQUIREMENTS." National Information Assurance Training and Education Center. IRI, Pocatello, Idaho. Web. 28 May 2014.

[10] "Security Requirements Engineering: A Framework for Representation and Analysis." IEEE Computer Society. 34.1 (2008): 133-153.

[11] "Authorization." From Wikipedia to encyclopedia. 2014.

[12] "Common Criteria for Information Technology Security Evaluation."http://www.commoncriteriaportal.org/. Common Criteria, Web. 28 May 2014.

[13] Avizienis, Algirdas, Jean-Claude Laprie,, et al. "Basic Concepts and Taxonomy of Dependable and Secure Computing." IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING,. 1.1 (2004)

[14] Damian,, Daniela, and Brian Corrie. "Awareness meets requirements management: awareness needs in global software development."

[15] HEITMEYER, CONSTANCE L., RALPH D JEFFORDS, and BRUCE G LABAW. "Automated Consistency Checking of Requirements Specifications." ACM Transactions on Software Engineering and Methodology,. 5.3 (1996): 231-261.

[16] Manzoor,, Umar, Kiran Ijaz, Wajiha Shamim,, and Arshad Ali Shahid. "Ensuring Data Security & Consistency in FTIMA - A Fault Tolerant Infrastructure for Mobile Agents." World Academy of Science, Engineering and Technology International Journal of Electrical, Electronic Science and Engineering. 1.3 (2007): 465-470.

[17] http://en.wikipedia.org/wiki/Excellence.

[18] "Web Authentication Security." SANS Institute InfoSec Reading Room. 06 june 2003. Reading.

[19] Headayetullah, Md, and G.K. Pradhan. "INTEROPERABILITY, TRUST BASED INFORMATION SHARING PROTOCOL AND SECURITY: DIGITAL GOVERNMENT KEY ISSUES." International Journal of Computer Science and Information Technology. 2.3 (2010):

[20] Tsai, Chii-Ren. "Non-Repudiation In Practice."

[21] Breaux, Travis D., and Annie I Anton. "Analyzing Regulatory Rules for Privacy and Security Requirements." IEEE TRANSACTIONS ON SOFTWARE ENGINEERING,. 34.1 (2008): 5-20.

[22] Liu, Lin, Eric Yu , and John Mylopoulos. "Security and Privacy Requirements Analysis within a Social Setting."

[23] "Information Technology Security Assessment." Assessment. 2013.