BRIDGING THE GAP BETWEEN REQUIREMENT AND SECURITY THROUGH SECURE REQUIREMENT SPECIFICATION CHECKLIST

[1]NIKHAT PARVEEN, [2]MD. RIZWAN BEG, [3]M. H. KHAN

Department of Computer Application, Integral University, Lucknow, India
Department of Computer Engineering, I.E.T, Lucknow, India
E-mail: nikhat0891@gmail.com

Abstract- Requirement phase of the software development is the most appropriate and early stage for incorporating security. Unfortunately, no efficient methodology or tool exists to address security at requirement phase. Almost negligible work has been reported to assess the impact of security at requirement phase. Traditionally, security is often an afterthought but it is necessary that security should begin at the requirement level which covers both functional security as well as its emergent characteristics. Hence, it is important to identify security requirements of the system. In this paper, we propose a checklist for security requirement and assess the security with the help of mapping requirement parameters and security attributes. The total weight of security requirement is calculated with the DSR value and placed in security requirement traceability matrix.

Keywords- Software Security, Security Requirement, Confidentiality, Integrity, Availability, Authentication, Non-Repudiation and Access Control

I. INTRODUCTION

Requirement engineering plays a crucial role for any quality software. For any success of software, requirement phase of SDLC is considered as a golden stone through which the quality of the software is judged. Requirement acts as ignition needed by a user to solve a problem or achieve the objective of the software. For any system to be judged, requirements behave as input criteria to check the quality of the software. According to author Boehm, McConnell requirement engineering defects cost 10 to 200 times more to correct once fielded than they would if they were detected during requirements development. It is also proven by the researchers and industry personals that reworking requirements defects on most software development projects costs 40 to 50% of total project effort and the percent age of defects originating during requirements engineering is estimated at more than 50%. The total percent age of project budget due to requirements defects is 25 to 40%. The need to consider security right from the beginning is a fundamental principle of secure software development. Therefore, it is highly desirable to define security requirements.

Requirements define necessary and desired capabilities of the proposed system. Hence, requirements' gathering is the initial step towards the development of software. The requirements are identified and group into functional and non-functional. The functional requirements are used to check the functionality of the product whereas the non-functional requirements define to quality features of the software. Functional requirements define the business rules. The non-functional requirements pay attention on issues like maintainability, portability, usability, security etc. Requirements are concerned with what the system should do whereas the security requirements are concerned with what the system should not do. Security requirements gathering allow gather information about the malicious part of the environment and decides how security breaches can be nullified. Security is treated as the most important non functional characteristics of a system. Therefore, it is highly recommended that the security aspects should be introduced at the initial stage i.e, requirement stage of development instead of adding security features on the existing application. To develop secured software, the core security services like confidentiality, integrity, availability, non-repudiation, authentication and access control should be incorporated in the requirements phase of a software development project. . Many researchers and practitioners in past have treated security of software as qualitative attribute but recently the researchers have suggested that security can be treated as quantifiable attributes.

Software Assurance Technology Centre (SATC) proposed attributes of requirement with regards to ambiguity, completeness, understandability, volatility and traceability to improve the quality of the software. A statement of a requirement is unambiguous if it can only be interpreted one way and the vice versa is termed as ambiguity requirement. A complete requirement document is a combination of correct and consistent requirements which does not left any items to be specified. Understandability relates to the ability of the developers to understand clearly which is meant as verified and validate requirements. Volatility refers to the frequent change that is the modified requirements. Traceability refers to the inherent

provisions for the cross- checking or referencing back to system requirement specifications.

By continuous thinking to identify the factors of secure requirement an effort has been made to enhance the requirement by analyzing the whole software development process. If the requirement is not secure, it can harm the quality of the software in any manner. They are accountable for the loss of secure design which directly relate to financial in terms of over budget or late delivery of products. The observation regarding identification of secure requirement factors is done by regress analysis of secure requirement best practices and security thumb rules with nature of requirements. Literature reveals that no significant solution is available or agreed-upon definitions to identify secure requirement factors at requirement time. The recognized factors of requirement are fully well suited with software requirement specification and having a strong impact to improve security at requirement time.

II. SECURITY REQUIREMENT ANALYSIS

The Security requirement together with the security objectives of a system identifies what attributes should be achieved when. It is basically the process of assuring security that is designed to operate at a level of security which can be consistent with the potential harm that could result from the loss, inaccuracy, alteration, unavailability, or misuse of the data and resources that it uses, controls, and protects. Security requirements are specific to the system that provide for protection of essential services and assets which are often neglected. The security requirement analysis begins by identifying and categorizing the information that is to be used by the software. The information supposed to be categorized according to its sensitivity. Once the information is categorized, the security objective is defined and security requirements can be developed.

III. SECURITY OBJECTIVES

In order to achieve the objective of the security at requirement phase the following goals are set forth for the secure software system.

- Assure that the users and client applications are identified and that their identities are properly verified.
- Assure that the users and client applications can only access data and services for which they have been properly authorized.
- Identify attempted intrusions by unauthorized persons and client applications.
- Assure unauthorized malicious programs which do not infect the application or component such as viruses.
- Assure that the communications and the data are not intentionally corrupted.
- Assure that the interaction between the two parties with the application or component cannot later repudiate those interactions.
- Assure that the confidential communications and data are kept private.
- Allow security personnel to audit the status and usage of the security mechanisms.
- Assure that the centers and their components and personnel are protected against destruction, damage, theft, or surreptitious replacement such as vandalism and terrorism.
- Assure that the system maintenance does not unintentionally disrupt the security mechanisms of the application, component, or center.

To meet the above objectives, we will briefly address the security policy that helps to protect the software system that ensures secure software requirements to the system. According to the author Donald Firesmith and Jan Jurgens security requirements has been suggested, and taken into consideration. They are discussed as below:

1. Confidentiality Requirement: This security requirement specifies that the resources to be used by the legitimate party. It ensures that only authorized users have access to accurate and complete requirement when it is required.
2. Integrity Requirement: This security requirement specifies that an application or component shall ensure that its data and communications are not intentionally corrupted through unauthorized construction, modification, or deletion.
3. Availability Requirement: This security requirement specifies to identify the true end user of the system, the business practices they are performing, and the time period when the end users perform those practices. This requirement helps to analyze the impact of service availability (or unavailability) on the end users' ability to accomplish their business objectives.
4. Authentication Requirement: This security requirement specifies the extent to which an organization, application, component, or hub shall verify the identity of its externals. Authenticity may be of two types: Message authenticity and entity authenticity. Message authenticity means that one can trace back the data from its original source. Entity authenticity means it ensures the party who can identify participants in a protocol, and specifically make sure that the party has actually actively participated in the protocol at the time.
5. Non-Repudiation Requirement: This security requirement specifies the extent to which an organization, application, or component shall prevent a party to one of its interactions (e.g., message,

transaction) from denying having participated in all or part of the interaction. Non-repudiation supports the fair exchange, which means that action that performed cannot be denied.

6. Access Control Requirement: This security requirement specifies mechanism for controlling access of system which protects assets. It keeps permission controllable with a large or frequently changing user-based software system. Sometimes, access control is enforced by guards, in the case of Java Security Architecture; guard objects control access to protected objects.

The majority of requirements engineers are not at all trained in security, and those who have been trained have only knowledge about overview of security architectural mechanisms i.e., passwords and encryption etc. rather than actual security requirements. Therefore, the most common problem with security requirements can be traced only when they are specific. Hence, there should be some systematic approach for software engineers to develop a secure system.

IV. SECURE REQUIREMENT MAPPING

Security should begin at the requirement level and it must cover all the characteristics that secure the process. 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 users are actually not familiar with the security issues and even they don't know what they actually need as security as an end product. To determine the security and reliability of a requirement from users and developers perspectives a secure requirement pertinent questions have been identified and shown in table1.

Table 1 Checklist for Secure Requirements

Security Requirement Sl. No.	Pertinent Question identified to make Requirement Secure
SR 1	Is each requirements is Uniquely Identified?
SR 2	Are the individual requirements uses the same term in different ways?
SR 3	If a requirement makes references to some other facilities; are these described elsewhere in the document?
SR 4	Are related requirements grouped together?
SR 5	Are there any contradictions in the requirement?
SR 6	Do you have to examine other requirements to understand what it means?
SR 7	Is any inspection, security audit review introduced in every phase of the software development life cycle?
SR 8	Is the acceptance criteria are followed by the requirements?
SR 9	Is any requirement clashing with any other requirement?
SR 10	Are the Functional requirements separated from Non- functional requirements
SR 11	Is any requirement contains "will be", "may be" etc
SR 12	Are all the requirements validated and verified?
SR 13	Are the problem requirement and its solution clearly Identified?
SR 14	Are Requirement characteristics defined?
SR 15	Are computer resources requirement such as hardware, software, operating system, etc. Is identified?
SR 16	Is any restriction of function on requirement is identified?
SR 17	Is any requirement is negative
SR 18	Is any requirement conflict with any other requirement made so far?
SR 19	Are all the requirements functionally well
SR 20	Are naming convention and standard followed by the requirements

Based on the pertinent questions identified in table 1, a set of requirement attributes is identified and set of security attribute is identified that met the reliability of the requirements which is shown in table 2 and table 3 respectively.

Table 2: Requirement parameters Identified
AMBIGUOUS
COMPLETENESS
UNDERSTANDABILITY
TRACEABILITY
VOLATILITY

Table 3: Security Attribute Identified
Access Control
Authenticity
Availability
Confidentiality
Integrity
Non-Repudiation

Mapping between Pertinent question with respect to Secure Requirement, Requirement Attributes Identified and Security Attributes Identified.

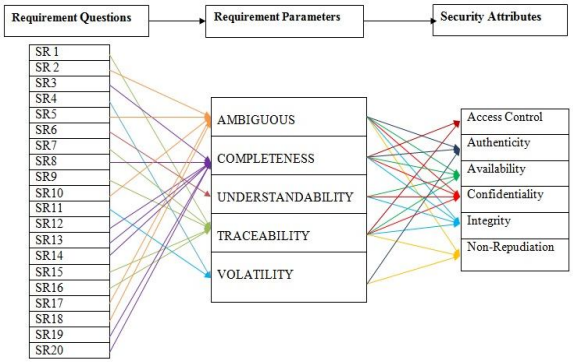

Fig 1: Mapping between requirement questions, requirement parameters and security attributes

The above mapping is prepared by responding the pertinent question on the basis of either true or false in order to provide secure requirements to developer. There is a need to revisit the requirement again and again. The priority of every secure requirement attributes depends upon its sensitivity. In this paper the priority is classified as high, medium and low and its weight is arbitrary assigned as 3 for high, 2 for medium and 1 for low priority as described in Table 4. Confidentiality, integrity and availability are the three main components of security and by mapping receive the maximum relationship; hence the priority is given

as high. Authenticity and non-repudiation is of medium priority and access control is of low priority. The degree of secure requirement can be calculated using the following formula.

$DSR = \Sigma SA_i W$
Where i = 1 to number of occurrence of each security requirement parameters.
DSR = Degree of Secure Requirement
SA = Security Attribute
W = Weight of an attribute

Table 4 Security Attributes and its weight

Security Attributes Priorities	Weight
Confidentiality	3
Integrity	3
vailability	3
Authenticity	2
Non-Repudiation	2
Access Control	1

As the severity of every secure requirements is not same, therefore, the priority of every requirements are assigned as high with weight 3, medium with weight 2 and low with weight 1. With the help of the above table 4, a secure requirement traceability matrix can be formed to calculate the security assessment of each requirement parameters for every questions of checklist as detailed in Table 5.

With the help of the above formula, the degree of security requirements (DSR) can be calculated

$DSR = \Sigma SA_i W$
DSR = 4 x3 + 4 x3+ 4x3+ 3x2+3 x2+2x1
DSR = 12+12+12+6+6+2 = 50
DSR = 50 = Total weight of security requirement.

Table 5 Security Requirement Traceability Matrix

Security Attributes → / ↓ Requirements Parameters	Confidentiality (3)	Integrity (3)	Availability (3)	Authenticity (2)	Non-Repudiation (2)	Access Control (1)	Weight
AMBIGUOUS	√	√	√	√	√		13
COMPLETENESS	√	√	√	√		√	12
UNDERSTANDABILITY	√	√	√				9
TRACEABILITY	√	√	√		√	√	12
VOLATILITY				√	√		4
Total No. of √	4	4	4	3	3	2	50

With the help of security requirement traceability matrix, the impact of security attributes confidentiality, integrity and authenticity is found high with respect to different security attributes which is identified in security requirements. Similarly the impact of traceability, ambiguity and complete requirement parameters is found high with respect to different requirement parameter which is identified in security requirements.

V. SIGNIFICANCE AND FUTURE WORK

With the help of secure requirement checklist, we are able to identified requirement attributes and security attributes simultaneously. By mapping between pertinent requirement questions, requirement parameters and security attributes a security requirement traceability matrix is able to form. With this matrix the impact of security on requirements can be easily identified.

Future work may include the standardization of the results by strong validation of the proposed checklist on a large sample size. In addition, the weights of each security attribute given in the checklist may also be computed to provide more accurate results. This result will help software developers and security experts for building secure software system.

CONCLUSION

The major contribution of this paper is the proposal of a requirement checklist for mapping between requirement parameters and security attributes. The mapping helps to provide secure requirements to developer. The requirement parameters and security attributes are identified and a unique weight is hereby proposed for the secure attributes with the help of mapping. Being prescriptive in nature, the checklist can be easily implemented and it may reassure the integration of the security right from the beginning in the software.

ACKNOWLEDGEMENT

Nikhat Parveen heartily thankful to Prof. (Dr.) R. A. Khan for their valuable support and constant effort to this work.

REFRENCES

[1]. Boehm, B. W. & Papaccio, P. N. “Understanding and Controlling Software Costs.” IEEE Transactions on Software Engineering SE-4, 10 (October 1988): 1462-77.

[2]. McConnell, Steve. “From the Editor - An Ounce of Prevention.”IEEE Software 18, 3 (May 2001): 5-7.

[3]. Nancy R. Mead, 2007. How to compare the security quality requirements engineering (SQUARE) method with other methods, technical note, CMU/SEI-2007-TN-021.

[4]. E. Whitney, “An introduction to gathering requirements, creating Use Cases and the UML,” White Paper, EPS Software Corporation. http://www.eps-cs.com/pdf/whitepaper_the_development_process.pdf.

[5]. H. Schmidt, “Threat- and risk-analysis during early security requirements engineering,” In the Proc. of Availability, Reliability, and Security, ARES’10 International Conference, IEEE Computer Society, 2010, pp. 188 – 195.

[6]. B.B. Madan, K. S. Trivedi, “Security Modeling and Quantification of Intrusion Tolerant Systems, Fast Abstract ISSRE 2002, Chillarge Press.

[7]. NASA Software Assurance Technology Center. Automated Requirements Measurement Tool. http://www.sqa.net/softwarequalitymetrics.html

[8]. IEEE Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998, www.math.uaa.alaska.edu/~afkjm/cs401/IEEE830.pdf

[9]. http://en.wikipedia.org/wiki/Software_security_assurance Jun 23, 2006.

[10]. Firesmith, Donald G., "Engineering Security Requirements." JOURNAL OF OBJECT TECHNOLOGY. 2.1 (2003): 53-68.

[11]. J. Jurjens, “Secure Systems Development with U M L ” , S p r i n g e r -Verlog, 2005.

[12]. Chandra, Shalini, and R.A khan. “Availability State transition model." ACM SIGSOFT Software Engineering Notes. 36.3 (2011): 1-3.

[13]. Salini, P, and S Kanmani. "Survey and analysis on Security Requirements Engineering." Computers and Electrical Engineering. 38. (2012): 1785-1797. <www.elsevier.com/ locate/complacent>.

[14]. Chandra, Shalini, and R.A khan. "Modeling and quantifying security attributes Confidential at Design Stage- An OO Software Perpectives." Pensee Journal. 76.4 (2014): 107-124.

[15]. 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. (2014): 236-240.

[16]. Roger S Pressman, “Software Engineering A Practitioner’s Approach” McGraw. Hill International Edition, MATH

★★★