Security Testing in Requirements Phase of SDLC

S. K. Pandey

Department of Information Technology, Board of Studies,
The Institute of Chartered Accountants of India
(Set up by an Act of Parliament)
NOIDA - 201309, INDIA

Mona Batra

Asst. Professor, Dept. of CSE
International Institute of Management, Engineering & Technology
JAIPUR-302022, INDIA

ABSTRACT
The importance and real potential of security in Requirements Engineering (RE) is now being well recognized. The inclusion of security controls and measures during the requirements phase helps to design, implement, develop and maintain secure and cost effective software. Security testing is one of the prominent techniques to reveal defects in the requirements specification. The requirement phase is the foremost phase to integrate security into software development process. In this paper, we review current scenario of security testing in requirements phase and try to identify the major research directions, based on the related published work. Researcher/s can select any of the area and start the investigation in the area. In this way, this work may be useful for entry level researchers in the concerned area/s.

Keywords
Security Testing, Security Requirements, Requirements Engineering, Secure Requirements Engineering etc.

1. INTRODUCTION
Requirements phase is the basic building block of any software. This is the phase, which involves information gathering about the customer's needs; in easiest terms, identify the problem that the product is expected to solve. In earlier days, many big software problems occur due to poorly taken requirement phase. In the SDLC, requirements inspection is recommended early to reveal defects in the requirements specification. One of the National Aeronautics and Space Agency (NASA) reports describes that the phase of requirement is backbone of whole SDLC [1]. Requirements phase activities if performed correctly then it will lead to high performance and cost effective software system. One of the recent research studies indicates that success in 68% of technology projects is "improbable" due to poor requirements analysis [2].

According to NASA findings, "problems that are not found until testing are at least 14 times more costly to fix than if the problem was found in the requirements phase"[3]. Moreover, another study revealed that the root cause of 56% of all the defects is errors introduced in the requirements phase. Approximately half of these defects are a result of poorly written, ambiguous, and incorrect requirements [4]. According to Strategies for Project Recovery report, 37 percent of the projects are at risk due to imprecise requirements [5]. Some other experts found that accurately capturing system requirements is the major factor in the failure of 90% of large software projects. The reason for integrating testing earlier into the software development life cycle is a simple economics. Studies have shown two findings: First that the majority of defects have their root causes in poorly defined requirements and second that the cost of fixing an error is cheaper if it is found earlier. The issue is scrap and rework. If a defect was introduced while coding, just fix the code and re-compile. However, if a defect has its roots in poor requirements and is not discovered until integration testing then re-do the requirements, re-do the design, re-do the code, re-do the tests and re-do the user documentation. It is all this *"re-do" work that sends projects over budget and over schedule*. Reduction of the cost and efforts is done by implementing the security *aspect right from the beginning* i.e. from the requirement phase itself.

Beyond this introduction on the background details, rest of the paper is organized as follows: In Section II, a survey of security testing research is briefly reported, whereas in Section III, we present the future research directions. Finally conclusion is drawn in Section IV.

2. A SURVEY OF SECUIRTY TESTING RESEARCH
Various researchers are underway on the different aspects of security testing in requirements phase. However, a rapid growth has been visualized recently. Some significant contributions bear weight and appear valuable among all. A selection from the trend setting research contributions are briefly described one by one for analysis on the advances, as follows:

A Charan Kumari and K Srinivas presented a Multi-objective Quantum-inspired Hybrid Differential Evolution (MQHDE) and its application on a real-world project. For the solution to the problem of changeable requirements, MONRP has been studied by researchers using different meta-heuristic search techniques. The efficiency of the proposed MQHDE is tested on a real-world application and the results are compared against the state-of-the-art multi-objective evolutionary algorithm NSGA-II and found that the functioning of MQHDE is promising and therefore can be used with confidence for the solution of real-world instances of MONRP [6].

The Open web application security project (OWASP) developed a cheat sheet. This cheat sheet provides an "at a glance" quick reference on the most important initiatives to build security into multiple parts of software development processes including requirement phase. It describes various policies and rule for security requirements in SDLC [7].

Saima Amber, Narmeen Shawoo and Saira Begum suggested a framework in which risk management is executed within Requirements Engineering (RE) process; for this purpose they considered three models of risk management. These models

are compared on the basis of risk identification methodologies. Finally, they derived a framework, which is based on UML oriented approach for modeling and reasoning about risk during the requirements analysis process [8].

Souhaib Besrour and Imran Ghani presented a paper to measure the security and related verification method in requirements engineering (RE). In order to address these issues, they proposed a new set of tools. First is the Effective Security Check List (ESCL), which is a check list with security questions that should be considered for measuring security. Secondly, the Traceability Matrix(TM), which is a two dimensional matrix to measure security during RE. Thirdly, Requirement Engineering Assessment Document (READ), which is a tool containing all statistical information about security performance during RE. The inclusion of these models in existing models helps to increase the security level of existing models [9].

Juan M. Carrillo De Gea, Joaquin NicolsS, Jose L. Ferna Ndez AlemaN, Ambrosio Toval, Christof Ebert and Aurora VizcaíNop performed a survey on requirements engineering tools. The paper presents the results of a survey answered by RE tool vendors. The objective of the paper is to gain an insight into current RE tools [10].

P. Salini and S. Kanmani published a survey paper on Security Requirements Engineering. They reviewed various issues, types and methods on Security Requirements Engineering (SRE). They analyzed and compared different methods of RE [11].

S. K. Pandey proposed a Security Maturity Model (SMM), in which five maturity levels have been proposed. These levels are based on the security vigilance occurring at the various stages of SDLC for any software starting from requirement itself [12].

Seda Gurrses, Magali Seguran and Nicola Zannone presented a paper that describes experience gained in the elicitation and analysis of requirements in a large-scale security-oriented European research project, which was originally conceived as an architecture-driven project. They illustrated the challenges that can be faced in large-scale research projects along with the discussion of how those practices and methods can be integrated into the requirements engineering process and possibly improved to address the identified challenges. In the last, they briefly explained the benefits that a proper requirements analysis [13].

Sultan Aljahdali, Jameela Bano and Nisar Hundewale published a review paper on goal oriented requirement engineering. This paper helps in identifying various concepts, terminology, significance and techniques of Goal oriented requirements engineering [14].

Smriti Jain and Maya Ingle developed a model namely Software Requirement Gathering Instrument that helps to gather security requirements from the various stakeholders. The proposed instrument helps the developers to gather security functional requirements as well as security requirements and incorporate security during all the phases of software development. They proposed a case study that describe the integration of the SSRGI with Software Requirements Specification (SRS) document as specified in standard IEEE 830-1998. Proposed SSRGI will support the software developers in gathering security requirements in detail during requirements gathering phase. It combines security requirement gathering and requirement analysis in one phase [15].

Manoj Ashok Wakchaure and Shashank D. Joshi described a new way to remove vulnerability through analysis stage of SDLC. They identified different conditions, which helps in identifying vulnerabilities in requirements phase of software development. They performed an analysis of single domain i.e. geographical database [16].

The most comprehensive framework on IT security requirements is currently the SQUARE method presented by the SEI of Carnegie Mellon University [17].

Robin A. Gandhi, Harvey Siy and Yan Wu performed a research work on Software Vulnerabilities. Common Weakness Enumeration (CWE) is a community-developed dictionary of software weakness types and their relationships. They describe the use of Common Weakness Enumeration (CWE) to study and prevent vulnerabilities in specific software projects [18].

Benjamin Fabian, Seda Gurses, Maritta Heisel, Thomas Santen, Holger Schmidt proposed a conceptual framework, which establishs the interrelations between the different concepts and notions used in security engineering. They performed a comparison of a framework with current security requirements engineering approaches, such as the Common Criteria, Secure Tropos, SREP, MSRA, as well as methods based on UML and problem frames. and at last they discussed similarities between various security methods [19].

Shamal Faily and Ivan Flechais proposed the IRIS (Integrating Requirements and Information Security) meta-model. It is a conceptual model for usable secure requirements engineering. They described a practical application of the meta-model through a case study in the Critical Infrastructure domain [20].

Malik Imran Daud developed a modern approach for software development called Extreme programming (XP). In this modern approach, developer do not need to wait for complete requirements. Priority of this approach is to incorporate security in each phase of software development life cycle. For this purpose, XP provides a guidance to recheck our security requirements. A new model has been designed that uses the principle of extreme programming. This innovative model focuses on the development of secure and error free software [21].

Bender RBT Inc. proposed a methodology that security testing at requirement phase overcome two major problem areas-first validating the various properties of requirement i.e. correctness, completeness, unambiguousness and logical consistency. Second, designing a necessary and sufficient test cases from those requirements that give full confirmation that design and code fully meet those requirements [22].

C. Banerjee and S. K. Pandey published a paper on secure requirement engineering, in which they focused on various aspects to integrate security in SDLC. They proposed twenty one security rules to be followed in the entire SDLC along with the validation results [23].

M. A. Hadavi, V. S. Hamishagi and H. M. Sangchi presented a paper that states peculiarities and deficiencies in security requirements engineering. This paper focuses on the current

research situation by reviewing and classifying the efforts into four main categories: security requirements in the standard software development processes, security requirements engineering consist of eliciting and modeling security requirements and threat modeling as a basis for security requirements engineering. It also presents various challenges and open problems for each category [24].

Charles B. Haley, Robin Laney, Jonathan D. Moffett and Bashar Nuseibeh proposed a framework for security requirements elicitation and analysis. The framework is based on building a context for the system. It represents security requirements as constraints, and developing satisfaction arguments for the security requirements. The system context is described using a problem-oriented notation, then it is validated against the security requirements through construction of a satisfaction argument. The satisfaction argument comprises of two parts: a formal argument that the system can meet its security requirements and a structured informal argument supporting the assumptions expressed in the formal argument. They evaluated the framework by applying it to a security requirements analysis within an air traffic control technology evaluation project [25].

Reijo Savola introduced a preliminary framework for security evaluation based on security requirement definition, behavior modeling and evidence collection [26].

T. Y. Chen, Pak-Lok Poon, Sau-Fun Tang, T. H. Tse and Yuen Tak Yu proposed a methodology for applying testing to requirements inspection for software quality assurance. The proposed methodology incorporates two approaches: requirements inspection and software testing. These two are the most important quality assurance activities. perspective-based reading(PBR), approach used in requirements inspection, operates under the premise that different information in a specification has different levels of importance for different uses of the document and Classification-Tree Method is used in software testing [27].

Paolo Giorgini, Fabio Massacci, and Nicola Zannone proposed a method for secure requirement engineering. This Secure Tropos methodology is a formal framework for modelling and examining security, that enhances the agent-oriented software development methodology i*/Tropos. They illustrated the Secure Tropos approach and some later refinements of the Secure Tropos methodology to address some of its shortcomings they also introduce the ST-Tool, a CASE tool that supports Secure Tropos methodology [28].

Orlena C. Z. Gotel and Anthony C. W. Finkelstein performed an analysis of the requirements traceability problem. In this analysis, they identified the underlying nature of the requirements traceability problem. They gave a distinction between pre-requirements specification (pre-RS) traceability and Post-requirement specification(post-RS) traceability. It help in improving the problem that the developers generally identify after the development of a software due to poor requirement traceability [29].

Thitima Srivatanakul, John A. Clark, and Fiona Polack proposed a technique, Hazard & Operability Analysis (HAZOP), and applies it to one widely used functional requirement elicitation component i.e. UML use cases, to provide systematic analysis of potential security issues at the start of system development [30].

3. RELATED RESEARCH DIRECTIONS

RE is a very active research area with a wide variety of methods and mathematical models. In current scenario, there is not any single approach that fulfills all the security related needs of secure requirements engineering. Researchers are continuously working in this area to have some useful findings. Moreover, future work may be done on any of the following research area/s (A pictorial representation is given in Figure 3.1) in the best interest of academics as well as industry.

- Software security is an emergent property of the system and has to be incorporated throughout the lifecycle. It would be beneficial to integrate it from the beginning of software development i.e. from the requirements phase. To confirm this mitigation, security testing is one of the prominent ways, which requires further investigation.
- There appears a need for the development of a framework that would assist in combining multiple security requirement methods.
- Work may be initiated on the comparison and evaluation of the current security requirements engineering approaches, such as the Common Criteria, Secure Tropos, SREP, MSRA.
- Work may be done on combining the strong points of the two well known approaches, Security Checklist and Traceability Matrix with a strong focus on security requirements elicitation and analysis in addition with security testing.
- A well-defined software security policy, risk rating for software and a checklist that supports the policy and requirements is essentially needable along with a risk assessment tool to address risk associated with a software project at requirements phase.
- Further research work is needed on how to integrate RE technologies in identifying various concepts, terminology, significance and techniques of goal oriented requirements engineering.
- Work may be done on identifying and addressing software security vulnerabilities prior to product deployment, i.e. at requirement phase.
- Work may be done on identifying and addressing software security vulnerabilities prior to product deployment, i.e. at requirement phase.
- Future work may also be done on developing secure requirements evaluation approaches for requirements phase of software development life cycle.
- Work may also be commenced on developing a framework for qualitatively and quantitatively analyzing the results of risk and task analysis to visualize the security of respective software development phase.
- One of the future work may be to develop a Testing Metrics for requirements quality and check the testing metrics for goal oriented requirement phase.
- Further, a threat detection technique can also be developed, which combines advantages of various threats detection techniques.
- Future task may be done to develop a security testing tool that have to be more efficient to preserve confidentiality, complexity, integrity and availability for the requirements phase.

- A new mathematical model may also be developed for checking the completeness and logical consistency of requirements specification

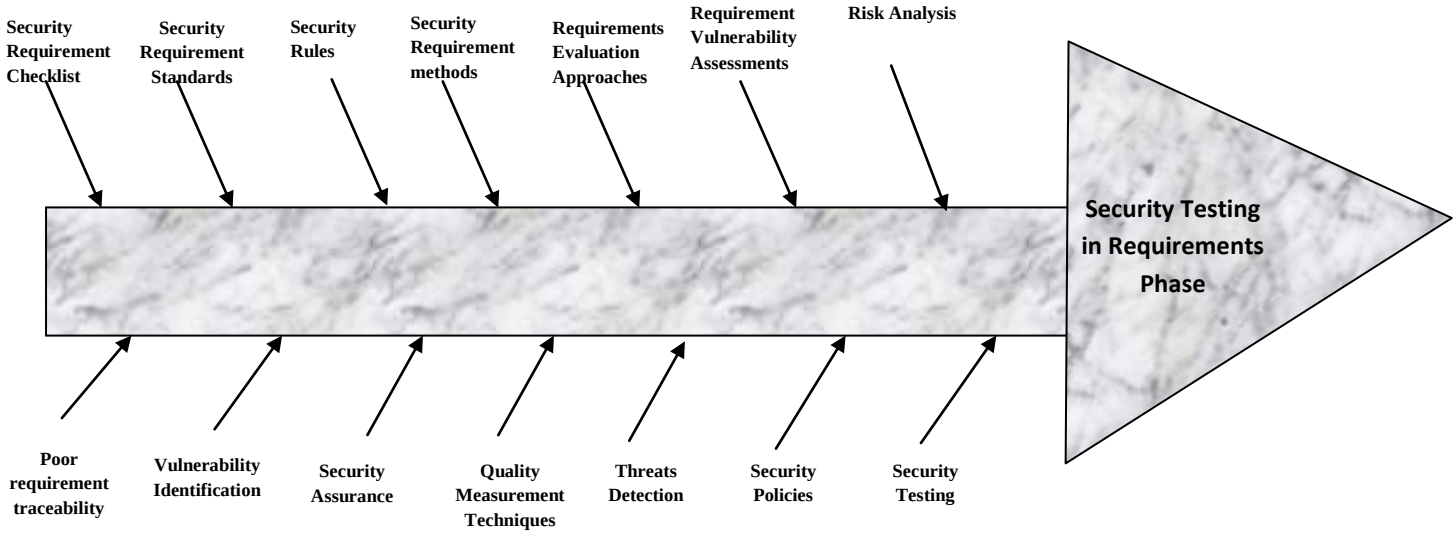

Figure 3: Research Directions

4. CONCLUSION

A review on security testing in requirements phase is discussed in detail in the paper. Though, it is not an easy task but researchers have made a significance progress in this area. This research work provides the knowledge of various security aspects that should be fulfilled during requirements phase that will improve the quality of the software. It is realized that efficient ways to incorporate security since the inception of software product in the development life cycle is highly desired. Due to the growing demand of IT, it raised many new critical research questions. Keeping all these aspects in mind, we presented number of research areas that will provide a new direction in the area of research. The research done on any of the given direction/s, may be very fruitful for academia and industry both and may enhance the security of a software level at initial stages only.

5. REFERENCE

[1] A report from NPR i.e. NASA Procedural Requirement , NPR 7150.2A. Retrieved on 6, March,2013. http://www.academia.edu/2557748/SoftwareStrategy_for_Reuse_Final_Study_Report

[2] The report, The Impact of Business Requirements on the Success of Technology Projects. Retrieved on March, 7, 2013. http://www.batimes.com/articles/the-impact-of-business-requirements-on-the-success-of-technology-projects.html

[3] Rosenberg, L., Hyatt, L., Hammer, T., Huffman, L., Wilson, W: Testing Metrics for Requirement Quality, Eleventh International Software Quality Week, San Francisco, *CA.*

[4] Mogyorodi, G. What is Requirement Based Testing? The Journal of Defense Software Engineering, 2003, pp 12-15.

[5] The report, called Strategies for Project Recovery. Retrieved on March, 5, 2013. http://www.zdnet.com/blog/projectfailures/cio-analysis-why-37-percent-of-projects-fail/12565

[6] Kumari A Charan and Srinivas K 2013. Search-based Software Requirements Selection: A Case Study, International Journal of Computer Applications 64(21):28-34.

[7] The Open Web Application Security Project (OWASP) cheat sheet in 2012 . Retrived on March ,7 ,2013 . https://www.owasp.org/index.php/Secure_SDLC_Cheat_Sheet#Purpose

[8] Amber Saima, Shawoo Narmeen, Begum Saira 2012 Determination of Risk During Requirement Engineering Process, Determination of Risk During Requirement Engineering Process, VOL. 3, NO. 3, pp 358-364.

[9] Besrour Souhaib and Ghani Imran 2012 Measuring Security in Requirement engineering, International Journal of Informatics and Communication Technology (IJ-ICT) Vol.1, No.2, pp 72-81.

[10] Carrillo De Gea Juan M. , Nicolss Joaquin , Fernandez Aleman Jose L. , Toval Ambrosio , Ebert Christof and Vizcaíno Aurora 2012 Requirements engineering tools: Capabilities, survey and assessment, Journal Information

and Software Technology, Volume 54, Issue 10, pp 1142-1157.

[11] Salini P. and Kanmani S. 2012 Survey and analysis on Security Requirements Engineering, Journal Computers and Electrical Engineering, Volume 3, Issue 6, pp 1785-1797.

[12] Pandey S. K. 2012 Security Vigilance System through level driven Security maturity model, International Journal of Computer Science, Engineering and Information Technology (IJCSEIT), Volume 2, No. 2, pp 10-17.

[13] Gurses Seda, Seguran Magali and Zannone Nicola 2011. Requirements engineering within a large-scale security-oriented research project, Journal of Requirements Engineering, 18:43–66.

[14] Aljahdali Sultan, Bano Jameela and Hundewale Nisar 2011 Goal Oriented Requirements Engineering - A Review, -1-880843-83-3/ISCA CAINE.

[15] Jain Smriti, Ingle Maya 2011 Software Security Requirements Gathering Instrument, International Journal of Advanced Computer Science and Application Vol. 2, No. 7, pp 116-121.

[16] Wakchaure Manoj Ashok and Joshi Shashank D. 2012 A Framework to remove by vulnerability through Analysis Stage of SDLC, International Journal of Science, Technology & Management (IJSTM) vol2, issue-2.

[17] Christian T. and Mead N. 2010. Security Requirements Reusability and the SQUARE Methodology, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, Technical Note CMU/SEI-2010-TN-027. Retrieved on March, 7, 2013 http://www.sei.cmu.edu/library/abstracts/reports/10tn027.cfm

[18] Gandhi Robin A., Siy Harvey, Wu Yan Studying Software Vulnerabilities Retrieved on March, 6, 2013. https://buildsecurityin.us-cert.gov/bsi/1209-BSI/version/1/part/4/data/1009GandhiSiyWu.pdf?branch=main&language=default

[19] Fabian Bennjamin, Gurses Seda, Heisel Maritta, Santen Thomas, Schmidt Holger. 2010 A comparison of security requirements engineering methods, Springer, Requirements engineering Volume 15, Issue- 1, pp 7-40.

[20] Faily Shamal and Flechais Ivan 2010 A Meta-Model for Usable Secure Requirements Engineering, SESS, pp-29-35.

[21] Daud Malik Imran 2010 Secure Software Development Model, A Guide for Secure Software Life cycle. International MultiConference of Engineers and Computer Scientists, Vol I, March 17 - 19, pp 1500-2246.

[22] The report on Requirement based Testing Process. Bender RBT Inc. , NY 12804 518-743-8755, 2009. Retrieved on March,8, 2013. http://benderrbt.com/Bender-Requirements%20Based%20Testing%20Process%20Overview.pdf

[23] Banerjee, C. and Pandey, S. K.. 2009. Software Security Rules: SDLC Perspective. (IJCSIS) International Journal of computer science and information security Vol. 6, No.1, pp. 123-128.

[24] Hadavi M. A., Hamishagi V. S. and Sangchi H. M. 2008. Security requirements Engineering; State of the Art and Research Challenges, International MultiConference of Engineers and Computer Scientists Vol I, pp 19-21.

[25] Haley Charles B., Laney Robin, Moffett Jonathan D. and Nuseibeh Bashar 2008 Security Requirements Engineering: A Framework for Representation and Analysis, IEEE Transaction on software engineering, Vol 34, No.1 , pp. 133-153.

[26] Savola Reijo 2007. Requirement Centric Security Evaluation of Software Intensive Systems, in the proceedings of the IEEE 2nd International Conference on Dependability of Computer Systems, IEEE-0-7695-2850-3/07, pp 135-144.

[27] Chen, T.Y., Poon, P., Tang, S., Tse T., Yu, Y 2006. Applying Testing to Requirements Inspection for Software Quality Assurance, Information Systems Control Journal 6,. Retrieved on March, 6. http://www.google.co.in/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&ved=0CDgQFjAB&url=http%3A%2F%2Fciteseerx.ist.psu.edu%2Fviewdoc%2Fdownload%3Fdoi%3D10.1.1.63.9296%26rep%3Drep1%26type%3Dpdf&ei=WPo_UbL8BIbIrQeWioAo&usg=AFQjCNG4hgX_A54Bedz57lcP4kTzmsh73Q&sig2=iJ9SmeaLk2XyEtRaVYlkQ&bvm=bv.43287494,d.bmk

[28] Giorgini Paolo, Massacci Fabio and Zannone Nicola 2004 Security and Trust Requirements Engineering, Department of Information and Communication Technology University of Trento – Italy. Retrived on: March,7, 2013 http://eprints.biblio.unitn.it/534/1/016.pdf

[29] Gotel Orlena C. Z. and Anthony Finkelstein C.W. An Analysis of the Requirements Tracability problem, Imperial college of Science, Technology & Medicine. Department of Computing, 180 Queen's Gate London SW7 2BZ. Retrieved on March, 8, 2013. http://csis.pace.edu/~ogotel/research/GOTEL93%20An%20Analysis%20of%20the%20Requirements%20Traceability%20Problem.pdf

[30] Srivatanakul Thitima, Clark John A., Polack Fiona 2004 Effective Security Requirements Analysis: HAZOP and Use Cases, 7th International Conference, volume 3225 of LNCS(Springer).