Integrating Security and Usability at Requirement Specification Process

Author: Nikhat Parveen[1], Rizwan Beg[2], M. H. Khan[3]

[1,2]*Department of Computer Application, Integral University, Lucknow, India.*

[3]*Department of Computer Engineering, I.E.T, Lucknow, India.*

ABSTRACT : *To construct any software, Requirement process is the common knowledge in most development organization. For any development of process security requirement is rarely supported. Over the years, researcher and developers have undergone many methodologies and techniques to secure software development life-cycle. A critical review for the development of secure software, Security and its usability is integrated at requirement specification process.*

Keywords *- Software security, security requirement, usability requirement, risk analysis.*

I. INTRODUCTION

Security for a software system has always been invitro and addressed only in the production environment through perimeter security like firewall, proxy, intrusion prevention system, antivirus, and platform security. This was the reason of security being considered as nonfunctional requirement [1]. An article by Gary McGraw points out that there is a fundamental tension between functionality and security, and this is why integrating security into the software lifecycle is so crucial and still lagging behind [2]. Integrating security in software development requires trained experts and dedicated resources.

Security and Usability seem to be found odds. It is found that improving one affects the other. Techniques to incorporate security issues have already been developed [3], but there is an important aspect for securing system is its usability. If user tries to change password periodically in order to improve security this impact a greater burden on users. Poor usability may also reduce security by driving users to workarounds, when users hard-to-remember passwords to their workstations. Sometimes, a password may be replaced by hardware token; this relieves the user of having to remember a password but this also imposes a new burden on the user to carry the token wherever that access is required. Poor usability is also an impediment to privacy protection. Hence, usability and security are thus attributes that can substitute against each other.

II. RELATED WORK OF SECURE AND USABLE SYSTEMS

Jerome H. Saltzer and Michael D. Schroder's "The Protection of Information in Computer Systems" addressed security and usability and also defined the basic principle of information protection. The authors have mentioned about the human interface be designed for ease of use, which helps the user to routinely check the protection mechanism correctly and automatically [4].

In 2002 Ka-Ping illustrates ten guidelines for usable security that are recommended and In 2003 Flechais et al, presented a novel method for building secure and usable software. They present the secure software development process AEGIS, which provides important tools for developing secure and usable systems.

Several important workshops programs have discussed the importance of security, usability and privacy of software system.

i) Human-computer interaction (HCI) community had held meeting which was focused on security. The HCI and Security System (HCI-Sec) workshop was held in conjunction with the ACM Computer-Human Interaction conference in 2003.[5]

ii) A Usenix Security Conference was held in 2003 which has

discussed the birds-of-a-feather that described about the security and its usability.[6]

iii) In 2004 the first workshop was held on Usable Privacy and Security Software that was held at Rutgers University.

iv) In 2005 an annual Symposium on Usable Privacy and Security (SOUPS) was formed which has discussed the research areas of: passwords and accounts; authentication of mobile devices; security models and decisions making; integrating usability with security education; privacy, security, and public policy.

Anne Adams have mentioned two problems regarding secure and usable password as users lack of security awareness and lack of knowledge which produces security mechanisms and systems which are not Usable[9].

III. OVERVIEW OF REQUIREMENT SPECIFICATION PROCESS:

Developing secure software is a tedious task; this process seeks to accommodate frequently competing factors that include functionality, scalability, simplicity, time-to-market, usability etc. the software engineer researchers has keenly focused on improving the modeling abilities in terms of non-functional requirement such as stability[10], performance[11], fault tolerance[12], security and usability[8] .

It has been found that the most ignored part of secured software development lifecycle is the security requirement engineering process and its usability requirement engineering process. Security should begin at the requirements level and it must cover all the characteristics that secure the process. Security is the degree of resistance to, or protection from harm. It comprises of threats, vulnerabilities, and risk and abuse cases.

Security and usability must go together; hence both should begin at requirement level. It has been found that systems that are secure but not usable will not be used, while systems that are usable but not secure will get hacked, compromised, and otherwise rendered useless [13].Usability is the term derived from user-friendly, it is the degree through which it can measure in terms of easy learning, task efficiency, recallability understandability and user satisfaction.

Fig 1 describe the flow diagram of secure and usable requirement specification process which describes the core steps that consists of identifying functional and non functional requirements, identify security requirements which comprises of threats, vulnerabilities, and risk and abuse cases. From functional and non functional requirement identify usability requirement which compromises of easy learning, task efficiency, recall ability, understandability and user satisfaction. Finally conduct security analysis and usability test that address the risk of security and its usability.

Identify functional and non functional requirement: the foremost task of any process is to gather requirement and then identify the functional and non functional requirements. It is basically used to elicit the quality goals of the process. In this process the assets will be identified with respect to security and its usability. The participants must have the domain knowledge; the eliciting techniques will be based on different needs and specifically the need for usability.

Identify security requirement: In order to elicit security requirements it is necessary to have the knowledge regarding security properties. The most common properties CI5A are defined as follows [1, 2].

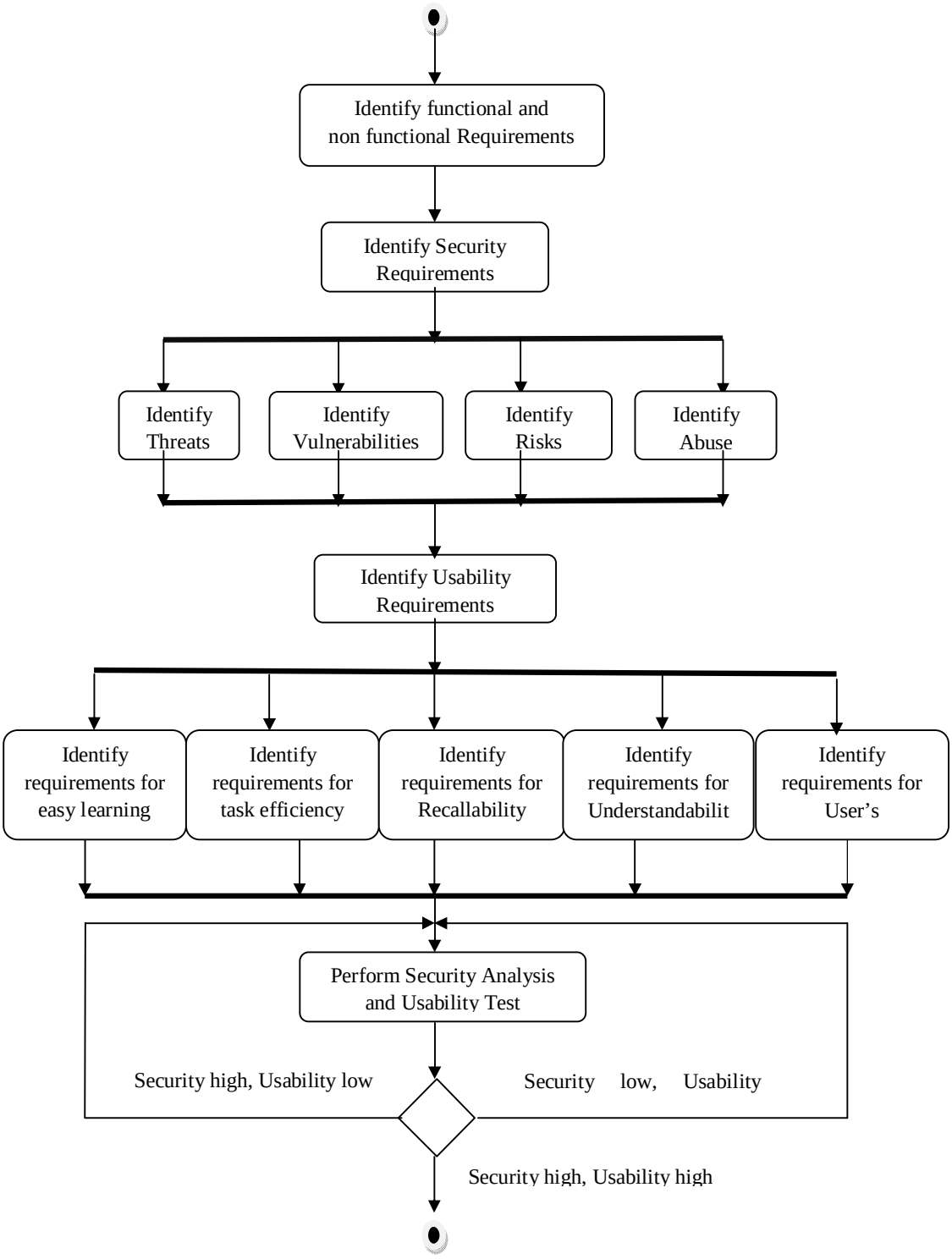

Fig.1: Flow Diagram for secure and usable requirements specification process.

- Confidentiality: this property is concerned with securities that prevent unauthorized disclosure of information.
- Integrity: this property is concerned with securities that prevent unauthorized modification of information.
- Availability: this property is concerned with securities that prevent unauthorized accessibility of users for information and resources.
- Authentication: this property is concerned with securities that prevent unauthorized users.
- Authorization: this property is concerned with securities that prevent unauthorized access control.
- Accounting: this property is concerned with securities that prevent unauthorized actions that affect the information and resources.
- Anonymity: this property is concerned with securities that prevent unauthorized identity.

Authentication, authorization and anonymity are generally considered in terms of access control.

Security always concerned with risk analysis and in order to design security countermeasures the risk analysis process must identify four general security issues: threats, vulnerabilities, risks and abuse cases.

- Identify Threats: A threats contain potential sources of attacks which include attacker, their motive, target and the resources.
- Identify Vulnerabilities: those areas that can be easily exploited. The use of security advisories, security scanners and good knowledge of security being used. It is also important to have the past knowledge of attacks in the system.
- Identify Risk: A risk may give rise to one or more risks. This can be resolved by security experts who have the knowledge and experience which is necessary to assess those risks.
- Identify Abuse Cases: An abuse case is a misuse case which produces harmful result to the system. This harmful result decreases the security (CI5A) of the system.

Identify usability requirement: Usability is about how the user perceives and uses the system process. In order to elicit usability requirement the five common usability factors must take in considerations [14].

- Easy Learning: The system should be easy to learn for both nonusers and users with experience from similar systems.
- Task Efficieny: The system should be efficient in daily use.
- Recallability: The system should be easy to remember for the casual user.
- Understandability: The user should understand what the system does.
- User's Satisfaction: The user should feel satisfied with the system.

Perform security analysis and usability test: for any software process, requirement engineering has always been critical for its success. To perform security analysis, one must understand the requirement of the process. This task should involve reviewing all existing high- level system documentation. It is advisable to review the requirement on the basis of security and its usability. This phase compromises of identification of security requirement, ensuring developers security awareness and conducting risk- analysis. It also performs sub activities as identification of usability requirement and usability test which measures effectiveness, efficiency and satisfaction of the software process. If the requirement is found highly secured and highly usable the process will stop and moved to design phase..

Security is basically used to control the undesirable actions while usability tries to make the desirable action easier for users. It may also be true that improving one also improves the other. A usable system will minimize unintentional errors, while a secure system will aim at ensuring that undesirable actions in a system are prevented or mitigated. This process can be introduced at requirement phase which helps to minimize the re-work later.

IV. CONCLUSION

Application software designed with security is much safer than those where security is afterthought. Traditionally, Security issues are considered during Design phase and its usability has been found after the implementation phase of software development life cycle process. This paper has presented a flow diagram for secure and usable requirements specification process which describes the core steps that consists of identifying functional and non functional requirements identify security requirements which comprises of threats, vulnerabilities, and risk and abuse cases. From functional and non functional requirement identify usability requirement which compromises of easy learning, task efficiency, recallability, and understandability and user satisfaction. Finally security analysis and usability test is performed to ensure the requirement specification process is highly secured and highly usable.

V. Acknowledgement

Nikhat Parveen thanks research assistant Nilu Singh, Department of IT, BBAU, Lucknow for their constant effort and heartly thankful to Prof. (Dr.) R.A.Khan for their valuable support to this work.

REFERENCES

[1] Asoke K Talukder, "Security-aware Software Developemnt Life Cycle(SaSDLC)- Processes and Tools", IWOCON 2009, Cairo,Egypt, 28-30 April 2009.

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

[3] Ivan Flechais, Cecilia Mascolo and M. Angela Sasse, 2006. Integrating Security and Usability into the Requirements and Design Process, Proceedings of the Second International Conference on Global E-Security, London, UK, http://www.softeng.ox.ac.uk/personal/Ivan.Flechais/downloads/icges.pdf

[4] J.H. Saltzer and M.D. Schroeder, "The Protection of Information in Computer Systems," Proc. IEEE, vol. 63, no. 9, 1975, pp. 1278–1308.

[5] A. Adams and M.A. Sasse, "Users Are Not the Enemy," Comm. ACM, vol. 42, no. 12, 1999, pp. 41–46.

[6] A. Whitten and J.D. Tygar, "Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0," Usenix Assoc., 1999, pp. 169–184.

[7] Ka-Ping, Y. *User Interaction Design for Secure Systems.* 2002. http://zesty.ca/sid

[8] Flechais, I., Sasse, M. A., & Hailes, S. M. Bringing Security Home: A process for developing secure and usable systems. New Security Paradigms Workshop 2003.

[9] Adams A, Sasse M A and Lunt P: 'Making passwords secure and usable', Thomas, editors, 'People and Computers XII', Proceedings of HCI'97, Bristol, Springer (August 1997).

[10] Jazayeri, M., 2002. On Architectural Stability and Evolution. Reliable Software Technologies-Ada-Europe, Vienna, Austria, pp: 17-21. http://www.infosys.tuwien.ac.at/Staff/mj/papers/ar chstab.pdf

[11] Denaro, G., A. Polini and W. Emmerich, 2004. Performance testing of distributed component architectures. Beydeda, S. and V. Gruhn (Eds.). Building Quality into COTS Components-Testing and Debugging. Springer. http://www.cs.ucl.ac.uk/staff/w.emmerich/publications/BeyadaGruhn/PerformanceTesting.pdf

[12] Guerra, P.A.D.C., C. Rubira and R. de Lemos, 2003. A Fault-Tolerant Software Architecture for Component-Based Systems. Lecture Notes in Computer Science. 2677: 129-149. Springer.

[13] Lorrie F.C. and Simson G., Guest Editors' Introduction: Secure or Usable?, Published by the IEEE Computer Society, SEPTEMBER/OCTOBER 2004 (Vol. 2, No. 5) pp. 16-18, 1540-7993/04/$31.00 © 2004 IEEE

[14] S.Lausen, Usability Requirements in a Tender Process, Published in: Proceedings of OZCHI'98, IEEE Computer Society, 1998.