Estimation of OOPS Software Security in Design Phase Perspective

Rajat Sharma[1], Shobhit Sinha[#2]

1Department of Computer Application, Shri Ramswaroop Memorial University, Lucknow – Deva Road 225003, Uttar Pradesh, India

#2 Goel Group of Institutions, Lucknow -Faizabad Road 226028, Uttar Pradesh, India

[1]rajatcivilsharma@gmail.com
[2]*sinha.shobhit@gmail.com

***Abstract*— Enhancing the features of object oriented design and by improving the security constraints with quality of the software is a great challenge for the researchers today. We aim to identify security factors and its impact on security. It has become clear that the most challenging task of security development is to provide required functionality and to fulfill specific properties of security such as trustworthy design which improves the security of software. The design stage takes as its initial inputs the requirement identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and /or prototype efforts. The design elements are intended to describe the software in sufficient detail, such that skilled developers and engineers may develop and deliver the software with minimal additional input design. Design attributes effects on security as well as on quality thus we want it to validate as a security factor in design phase.**

***Keywords*— Software, Security. Object, Oriented, Design, Attributes**

I. Introduction

Modern software security is continuously increasing in dimension and complexity. Security issues have an ever greater effect on quality of software [1].During the last decades; it has become clear that the most challenging task of security development is to provide required functionality and to fulfill specific properties of security such as trustworthy design which improves the security of software. Software is very vulnerable which can be attacked to cause damage to previously healthy software. This infected software can replicate itself and be carried across networks to cause damage in other systems. Security needs mainly include data confidentiality, integrity, availability, authentication, authorization and access control. These are the important security factor contribute in best of the software security. Software security and its factor identification is the broad area which may reveal many facts that will enhance security. Here a proposal for research in the same area that is identification of security factor, its impact on other security factor. Design features which will include security factors will improve the security as well as quality of software [2]. The Design Phase result in one of the two key elements to the project: the design. Without a details design, the second key element, the software, cannot be constructed, implemented, trained upon, or operated. The approval of the Design Phase deliverables, the completion of the design project status review, and the approval to proceed to the next phase, signifies the end of the Design Phase for secure software with user friendly

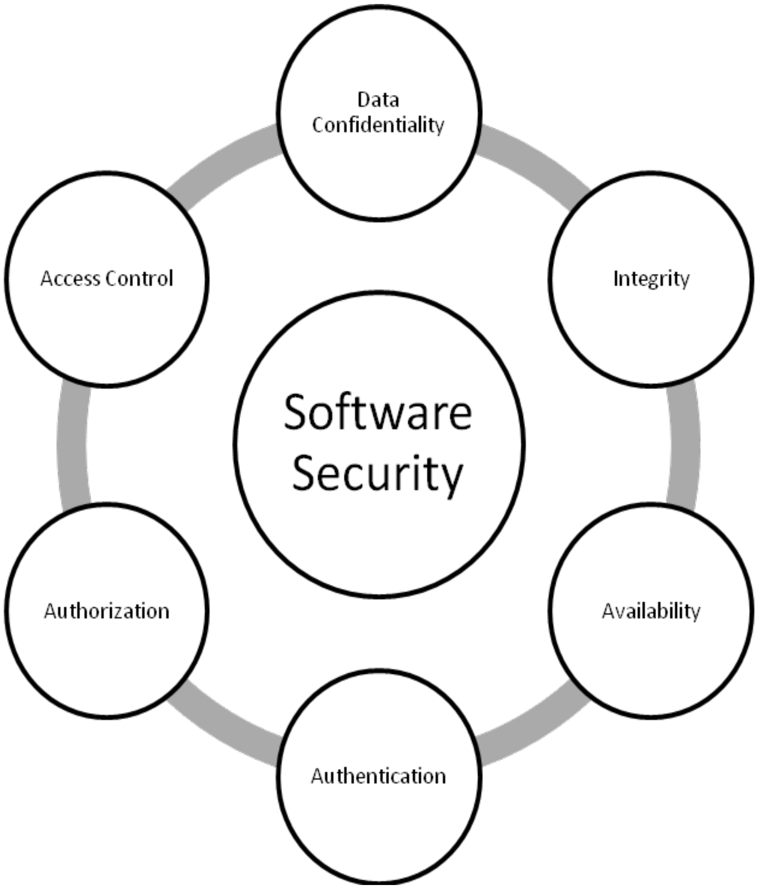

Fig. 1 Security factors contributing in best of software security

II. Origin of the research area

The evolving security environment in the early twenty-first century creates new challenges for all, including for software. Design properties will implicate in stonework the way for security across the Software region towards the twentieth first century, and will contributed to important autonomous conversion in the Software Security area. Methods of software coding are being used that are inherently insecure. Logic modules are written that create security subjects when they are combined, the implementations are rolled out that create vulnerabilities rather than eliminate them. There is no question the development is a complex undertaking. When 100 programmers collaborate on a software project, they work on individual software factions or architectural components [3]. These components are often developed separately and periodically stabled to ensure that they work together. But when it comes to tie it all together it takes a conductor to orchestrate the many different components and developer groups into a cohesive whole for its constancy purposes. This complexity of development makes integration testing difficult to accomplish in a comprehensive manner. These concerns and complexities in today's software world lead to problems, with security vulnerability being one of the more significant by product. With development forces working furiously just to make an application work within all of its components within the sometime unspoken pressure to be done on time and under budget, security for success appears to be path of least resistance. Security for points of failure and looking for ways to "break" the software are often ignored. Even when they aren't ignored, it is the sophisticated process that needs to be managed to accomplish the breadth and depth of software necessary to ensure security.

III. Security Attributes vs. Object oriented features

Ways of improvement to control properly complexity factor, which is related to security attribute [4]. The main objective of this proposal is to lessen the complexity of security with the enhancement of security using security factor and using it in designation of securities. We shall describe this occurrence as Security Attributes vs. Object oriented features. We will find object oriented constrains and its mapping in security factors. Many of the abilities that are claimed for security are exactly those that are wanted by those who have to use or deploy software, yet run almost directly counter to the commercial needs of typical software development businesses, where a continuous revenue stream that means usually through the mechanism of upgrades or high-priced support is needed. Design attributes effects on security as well as on quality thus we want it to validate as a security factor in design phase.

IV. Conclusions

Objective of the proposed project is to enhance the features of object oriented design and by this to improve security as well as quality of the software. We have already identified security factors and its impact on security. The design stage takes as its initial inputs the requirement identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and /or prototype efforts. The design elements are intended to describe the software in sufficient detail, such that skilled developers and engineers may develop and deliver the software with minimal additional input design.

Acknowledgment

We would like to thank Dr Bineet Kumar Gupta, Head, Department of Computer Application, Faculty of Computer Science and Engineering, Shri Ramswaroop Memorial University

We would also like to thank Dr Promila Bahadur, Faculty of Computer Science and Engineering, Shri Ramswaroop Memorial University, for their guidance and support throughout this research work.

References

[1] A Forrester Consulting Coverity , "The Software Security Risk Report the Road to Application Security

[2] Begins in Development September 2012", Available at: http//www.coverity.com/liberary/pdf/thesoftwaresecurity-risk-report.pdf Accessed on Jan 10 2015.

[3] Ansar Y., Giorgini P., Fabio M.,Nicola Z., "From Trust to Dependability through Risk Analysis" Proceeding to Second International Conference on Availability, Reliability and Security,IEEE Xplore,2007,pp. 19-26

[4] Bulgurcu B., Cavusoglu H., Benbasat I., "Information Security Policy Compliance: An Empirical Study of Rationionality-Based Beliefs and Information Security Awareness", Imformation Security compliance ,volume 34, Issue 3,2010,pp. 523-548.

[5] E.V.Bartlett, S Simpson, Durability and Reliability,Alternative Approaches to Assessment of Component Performance overtime, Available at: http//www.irbnet.de/daten/iconda/CIB8616.pdfAccessed on 20 Sep 2016

[6] R. Pandey, V. Akella, and P. Devanbu. Support for system evolution through software composition. In ICSE '98 International Workshop on the Principles of Software Evolution, 1998.

[7] R. Pandey, R. Olsson, and K. Levitt. Policy-driven runtime support for secure execution of user code in extensible kernels. (http://seclab. cs.ucdavis- • edu/~pandey/ariel, html).

[8] D. E. Perry and A. L. Wolf. Foundations for the study of software architecture. ACM SIGSOFT Software Engineering Notes, October 1992.

[9] A. Pickholz. Software protection method and apparatus. United States Patent 4,593,353, 1986.

[10] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous connections and onion routing. IEEE Journal on Selected Areas in Communication Special Issue on Copyright and Privacy Protection, 1998.

[11] J. O. Ruanaidh, H. Petersen, A. Herrigel, S. Pereira, and T. Pun. Cryptographic copyright protection for digital images based on watermarking techniques. Theoretical Computer Science, 226(1-2):117- 142, Sept. 1999.

[12] T. Sander and C. F. Tschudin. On software protection via function hiding. In Information Hiding, pages 111-123. Springer-Verlag, 1998.

[13] T. S. Souder and S. Mancoridis. A tool for securely integrating legacy systems into a distributed environment. In Working Conference on Reverse Engineering (WCRE), Atlanta, GA, October 1999.