A Framework to Quantify Security: Complexity Perspective

Suhel Ahmad Khan, *Member, IACSIT*, and Raees Ahmad Khan

***Abstract*—This paper discus the detailed research review on security quantification of object oriented software and put forth an effort to characterize design time software security. An effort through complexity perspective is to identify the involved factors of complexity and its probable impact with object oriented design parameters to quantify security. Complexity is taken as a key factor to software security quantification at early stage of development life cycle. A Security Assessment through Complexity (SAC) framework is proposed and relationship from design parameters to complexity and security is defined in accordance with its anticipated influence and importance.**

***Index Terms*—Security, complexity attributes, OO design constructs, security quantification.**

I. Introduction

A series of tragedies and chaos caused by the insecure software proves that the software security may be simply a matter of death and life at time. Security is now becoming foremost concern for software industries. Controlling and improving software design security have been an important issue in software security engineering. Security estimation of software may heavily affect security of the final product. The analysis of security parameters and their impact on security will ease up to uncover the strengths and weakness of the software and provide the basis for carrying out cost and benefit analysis [1]. Generally, security problem arises because of the lack of inherent security measures. Effort in respect of early and accurate security estimation needs to be undertaken for worthwhile software development. It is apparent that fixing bugs, mitigating vulnerabilities, removing irregularities and nonconformance to standards and eliminating unwanted complexity early in the development life cycle leads to the development of secure software.

II. Design Complexity and Security

Most of the software contains security flaws because of their complex design nature. Design complexity is most important factor of software security [2]. Design reviews validate the requirements and implement the design description. To implement security at the end of software development increases the cost and complexity of making changes [3]. The only way to develop systems with required functionality and performance that can also withstand malicious attacks is to design and implement them to be secure. Using the concept of software security estimation during development of software, security can be measured by analyzing design activities, measurement of security attributes and its impact on software, security team may improve/control software security. There is need to develop a scientific structured approach to deal in a word of complex software design to ensure that application software are secure.

Manuscript received April 15, 2012; revised May 30, 2012.

The authors are with the Department of Information Technology, Babasaheb Bhimrao Ambedkar University (A Central University), (e-mail:ahmadsuhel28@gmail.com, khanraees@yahoo.com).

III. The Framework

A reliable quantitative estimate of software security is highly desirable at an early stage of software development life cycle. Literature survey reveals that nothing significant, precise and clear exists in this regard that can be used to quantify software security in early stage of development. The regress analysis shows that not enough work has been done in this regard to find out any framework which deals to quantify security through complexity perspective.

A. Premises

The following premises have been considered when the proposed framework is being used to quantify security using complexity attributes which having impacts on security.

- The identified factors of software security e.g. Integrity, Availability, Confidentiality covers all aspects of security quantification approach.
- The identified complexity factors having impact on security attributes and its behavior are best suited in Object Oriented Design Perspective.
- An integrated approach to measurement of software security through complexity perspective is feasible.
- This approach is applicable at early stage of development e.g. at design phase to uncover errors as for as possible.
- A common set of features for desired metrics may be used to form the basis for its development.

B. Generic Guidelines

The guidelines before following the process to assessment of security may be listed as follows:

- Assure compliance/observance to collect the common set of essential and desirable features of proposed methodology.
- Identify and persist all security and complexity factors to be measured in object oriented design perspective.
- Proper correlation of identified attributes with their related metrics.
- The values of attributes must be persisting for quantification and designing process of quality product.

These guidelines will affect the quality and performance of the software. It discusses the need to develop a scientific structured approach to deal in a word of complex software design to ensure its security.

IV. The Framework Development Process

The framework produces an approach to identify security and complexity factors at design time and correlate these attributes in order to quantify security through complexity perspective. There is always required a process which control all the activities related to software design to mitigate complex structure because after a certain level things are more difficult to understand and manage. Due to this inconvenience more vulnerabilities and wider attack space for security breaches arise. The term complexity is one of the most important factors for software design and development which directly or indirectly involve with software security, quality, development time, cost, reliability, maintenance and all possible achievements of software users. A framework, Security Assessment through Complexity (SAC) in Fig. 1. is an approach to finalize design by quantified assessment using security best practices correlating them with security attributes and complexity factors. Each phase is discussed as follows.

The conceptualization is the primary phase of any problem solving activity. This phase will elicit the design parameters and related metrics in the mirror of regress review by best design practices and consolidated rule set for developing secure design to realize the problem related facts [4]. The following steps will assess needs and significance to understand the requirements of primary phase for secure design issues with respect to design complexity. It will check the possibility, scope availability of necessary tools etc for undertaking such a development. Identify Theoretical Basis through Best Practices deals with the specification of the theoretical viability for framework implementation. 'Select Metrics Attributes' emphasizes the selection of relevant attributes or inherent characteristics for the software development paradigm as complexity perspective. An effort has been made to identify the related factors of security and complexity which are helpful for security quantification using design complexity as a key attribute in object oriented design perspective. 'Establish Measures and Strategy' refers to a conclusive statement of measurements to be made, behavioral prediction of used metrics and strategy for incorporating the features in metrics computation. Quantification of security is possible. Most of approaches are either theoretical basis or can be used as best practices [5]. A multiple regression technique has been established for formulation of security models. Theoretical and empirical validation proves the validity of used measures. After experts review & revision the 'Finalization' refers to acceptance of valid design and its valid quantifiable values for design complexity and security.

V. Framework Implementation

The framework proposed is implemented in the following manner:

A. Security Design and Conceptualization

The basic idea is to conceptualize an estimation technique to measure the severity of software. The identified term complexity is combination of its component types and its interconnectedness. It is obvious that complexity of whole structure is increased at design time, but hierarchical decomposition breaks whole structure into smaller modules with proper functionality and less interdependency. It distributes design complexity form more to fewer for better understanding and appropriate functioning. The common secure design principle which was developed by Saltzer and Schroeder are built upon the idea of simplicity, separation and restriction [6] is helpful to conceptualize the term complexity with security.

More inherited classes constitute deeper hierarchy. In other hand simple design and less inherited classes having less fault and requires low testing and maintenance. The higher number of inheritance metrics like Depth of Inheritance Tree, Number of Children add up deeper design tree and complex design structure. High coupling shows more interconnectedness of components which constitute complex design that can go against the concept of simplicity. Encapsulation wraps data structure and behavior in single unit. It provides freedom for implementing abstraction with interfaces. This implementation changed without affecting any other object. This is most feasible technique to preserve the integrity. Design cohesion of class reduces the complexity of design.

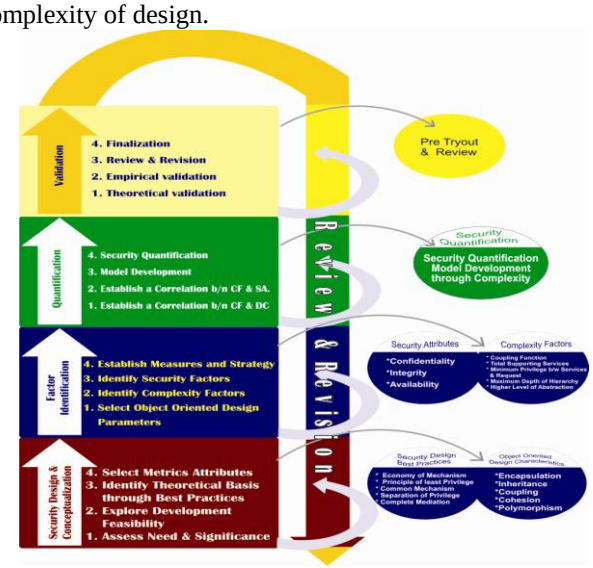

Fig. 1. Security assessment through complexity (SAC)

B. Factor Identification

Security is multidimensional attributes. It comprises three attributes: CIA-Confidentiality, Integrity and Availability. Security enhancement is possible when its quantified assessment data is available. Total supporting services is union of behavior of class elements and efforts to provide protection to the basic components of object oriented design. To gain maximum strength of protection it is mandatory to keep design complexity low by preventing unnecessary privilege grant to services. Privileges should be minimal according to interaction between services and requests. Most of the services are holding the dynamic behavior. The behavior of components is analyzed by counting services at run time environment when they demonstrate polymorphic behavior.

Total Supporting Services=Behavior of Components **U** Maximum Strength of Protection

Decomposition is the process of defining the generalizations and classifications that compose an abstraction [7], [8]. Keeping in mind this assumption,

decomposition is merged with higher level of abstraction to maintain the theoretical basis that larger the number of methods invoked from an object, increases the design complexity. The motivation of hierarchical decomposition of design is to provide free space and allows the designers to take design decisions independently to distribute complexity across multiple components with less interdependence. The identified complexity factors are combined according of their physical and psychological behavior for better assessment. The identified factors of complexity to quantify security are Coupling Function, Total Supporting Services, Minimum Privilege between Services and Requests, Maximum Depth of Hierarchy and Higher Level of Abstraction.

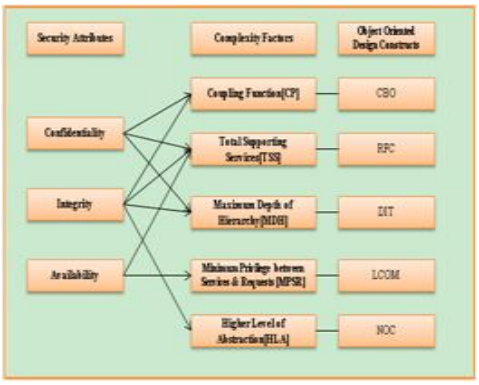

Fig. 2. Model development

C. Quantification

Quantification analysis of software security at early stage enables the evaluation and assessment of security and provides the basis for assessment security technologies. The proposed model development is mentioned in Fig. 2. The generic quality models [9], [10] have been considered as a basis to develop the Security Quantification Model for Object Oriented Design shown which involves the following steps to identify the factors of object oriented software that influences complexity at Design phase, Identification of Object Oriented Design Characteristics, A means of linking of them .A case study has been taken with six different class diagram to measure the complexity values in term of object oriented design metric. This technique establishes a relationship between dependent variable and multiple independent variables. The Multiple Regression equation takes the following form:

D. Confidentiality Quantification Model

Confidentiality refers unauthorized disclosure of information. It also limits the access of information in right direction and prevention of disclosure of information to unauthorized users.

$$\text{Confidentiality} = .599 - .623 \times CP + 341 \times TSS - 1.25 \times MDH \quad (1)$$

E. Integrity Quantification Model

Integrity is the concept of credibility of information resources. It allows the possible authorized changes to insure that during alteration of information that appropriateness of design or information must be insured at origin level or source level.

$$\text{Integrity} = 0.578 + 0.071 \times CP - 0.119 \times TSS + 1.074 \times HLA \quad (2)$$

F. Availability Quantification Model

Availability is the readiness for correct services, which can also be defined as the degree to which a system or component is operational when required for use.

$$\text{Availability} = .654 + .048 \times TSS - .180 \times MPSR \quad (3)$$

VI. Conclusion

A strong theoretical basis for security design & conceptualization has been proposed in this framework development. This development is concentrated over security quantification in object oriented environment targeted design complexity with negative impact on security. The proposed security models are very much helpful to quantify security as complexity perspective. The Viable experiments are helpful to validate the framework and informal reviews and revisions are carried out throughout the entire phases of development process to finalize accurate quantifiable values.

Acknowledgment

This work is sponsored by UGC, New Delhi, India under F. No. 34-107\2008 (SR).

References

[1] M. S. Bharat and K. S. Trivedi, "Modeling and Quantification of Security Attributes of Soware Systems," in *Proc. the International Conference on Dependablfte Systems and Networks*, pp. 504-514. IEEE, 2002,

[2] R. A. Khan and S. A. Khan, "A Roadmap for Security," *International Journal of Computer Science & Emerging Technologies*, vol. 1, pp. 5-8, issue 1, June 2010.

[3] G. Peterson, "Collaborating in a Secure Develop. Process," *Info. Security Bulletin*, vol. 9, pp. 165-172, June 2004.

[4] R. S. Pressman, *Software Engg. A Practitioner's Approach* Mcgraw. Hill International Edit., 2001

[5] S. Chandra and R. A. Khan, "Software Security Metric Identification Framework (SSM)," *International Conference on Advances in Computing, Communication and Control, ICAC3'09*, ACM.

[6] P. Meland and J. Jenesen, ARES, "Secure Software Design in Practices," in *Proc.of IEEE*, 2008.

[7] L. Rogenberg and D. Brennan, "Principle Components of Orthogonal Object Oriented Metrics (323-08-14)," *White Paper Analyzing Results of NASA Object oriented Data*, Oct. 2001.

[8] S. Lawrence and R. K. Cunningham, "Why Measuring Security is Hard," *IEEE Computer and Reliability Societies*, pp. 46-54, October 2010.

[9] R. G. Dromey, "A Model for Software Product Quality," *IEEE Trans. on Soft*. vol. 21, no.2,, pp. 146-162, Feb. 1995.

[10] C. Wang and Wulf, "A Framweork for Security Measurement," in *Proc. of National Information Systems Security Conference*, vol. 7, no. 10, pp. 522-533, October 1997.

S. A. Khan is pursing PhD in Information Technology form Babasaheb Bhimrao Ambedkar University (A Central University), Vidya Vihar,Raibareli Road, Lucknow. He has completed his MCA Degree from Uttar Pradesh Technical University, Lucknow. This Young, energetic Research Fellow, who has completed a Full Time Major Research Project funded by University Grants Commission, New Delhi. The funded research project entitled "Quantifying Security in Early Stage of Development Life Cycle: An Object oriented Software Perspective" has been successfully submitted to UGC. The author has more than five years of teaching & research experience. He is currently working in the area of Software Security and Security Testing. He has also published & presented papers in refereed journals and conferences. He is also a member of IACSIT and Internet Society.