Security Estimation Framework: Design Phase Perspective

Shalini Chandra  
Department of IT,  
BBA University, Lucknow,  
India  
nupur_madhur@yahoo.com

Raees Ahmad Khan  
Department of IT,  
BBA University, Lucknow,  
India.  
khanraees@yahoo.com

Alka Agrawal  
Department of IT,  
BBA University, Lucknow,  
India  
alka_csjmu@yahoo.co.in

Abstract

Generally, security analysis process is carried out through subjective evaluations. Early methods of security attribute analysis emphasize on codes, models and policies. An exhaustive review on software security estimation revealed the fact that there is no standard methodology available to assess software security quantitatively. In absence of any guideline, it is worthwhile developing a prescriptive framework in order to quantify software security. This paper proposes a framework to estimate software security in early stage of software development life cycle. A phase wise sequential approach presented in the paper helps security professionals to estimate security and mitigate vulnerability in design phase.

1. Introduction

Software development industry is facing a problem in delivering secure software. Software engineers, project managers, and decision makers remain under stress on account of their inability to deliver secure software. It is going to be a cumbersome process and an expensive affair for any software developing unit to implement security features in the previously developed application programs. The reasons may be following: a) It is not possible to restructure and rewrite an application program in its entirety with security modules b) Additional investment in making the previously developed software secure is not really advisable and c) It is not suggested to retrain software developers to be security experts [1]. It is prerequisite for all software system to perform properly under the presence of malevolent approaches. Software system should be capable of functioning proactively, meaning thereby, it should be self-defensive.

Security is similar to the concept of safety, confidentiality, and reliability. Number of security loopholes and vulnerabilities exists due to the defects of security architecture and security mechanism. Hackers and attackers do not create security loopholes; rather they target the weaknesses in the software and exploit them. In order to maintain the software security during the developmental stages, hacking should be made too difficult [2]. The purpose of making the software secure is to protect the software from all kinds of attacks, errors, bugs, threats, viruses and vulnerabilities [3]. Security architecture must be designed to cater the needs of product security goals and sensitive information contains therein.

A critical look at literature available on software security estimation reveals unavailability of any standard approach to be followed to estimate security. The fact strengthens the idea of proposing such a framework. As a result, an attempt has been made to propose a framework to quantify security in early stage. The fundamental objective of the framework is to split complexity of security into smaller one to make it easier to handle. The paper is organized as follows: Second section presents an overview on the need for estimating security early in the software development life cycle. Third section does an exhaustive review on the relevant work in the area. Fourth section strengthens the work and forms theoretical basis. Fifth section discusses the framework at length. Sixth section deals with the motivational aspect and future goals. The paper concludes with summarizing the developed framework.

2. Software security estimation: need and challenge

Review of literature and studies in the fields reveals the fact that the security feature cannot be added once software is ready [4]. Rather, it must be integrated within the development life cycle. As a matter of fact,
integration of security features must be a part of software development life cycle [1,3,]. Software industry suffers serious damages due to the lack of security estimation before its implementation. Only penetration testing or penetrate and patch are not sufficient for the purpose. The existence of vulnerabilities in the software reflects that the software can be compromised at any point of time [5]. Recently, Slammer worm infected SQL servers because of software vulnerability [6]. Threats take advantage of hardware and software weaknesses or vulnerabilities. As a result they can affect the violation and breakdown of availability, integrity, confidentiality, and nonrepudiation as well as other aspects of software security [3].

Foremost goal of software development is to deliver high quality secure products that are correct, consistent, and complete. A well established fact, an activity can’t be controlled if it can’t be measured, is widely acceptable. Security falls in this rubric. Therefore, in order to control security, it is mandatory to make an effort to measure it. Security estimation results will facilitate in describing the strengths and weaknesses of software [7]. Design phase is the first step towards problem domain to solution domain. In design phase, software architecture is available. Therefore, it is the most appropriate phase to estimate security of the software. Security estimation of software in this phase will assist to protect software from damages.

Software security estimation is the process of quantitative assessment of product security. Software security may be estimated with combined use of security factors, security metrics, security models and security functions. Software security estimation is a complete structured process. There is need for bringing down error rates at every stage of life cycle. Minimizing error rates reduces probability of security failures and cost [8]. Security modeling techniques including threat modeling and vulnerability modeling, contribute to mitigate security issues [9, 10].

Security measurement of software may help control security of the software and consequently improves level of software security. No methodology or framework exist supporting security measurement of object oriented software systems at the software design level [11]. At the later stages of the software development, security implementation increases the complexity and cost of making changes. A structured approach is required to accomplish the objective of incorporating security early in an efficient and effective manner.

3. Review work

In 1980’s, security accent changed to system analysis and architectures. In 1988, Federic Copigneaux and Sylvain Martin presented a methodology for software security evaluation and certification [12]. Wang and Wulf’s framework is one of the earliest attempts for security measurement. The framework was based on the theory and formal methods of measurements. Unfortunately, framework was not empirically validated. Wang recommended identifying set of guidelines to be used for running experiments and analyzing the results of experiments [13]. Wang’s work strengthens the need for developing set of guidelines to quantify security and analyze the quantification results.

Bharat B. Madan et al. worked on modeling and quantification of security attributes of software systems [14]. Focus of the work is on developing a methodology for analyzing security attributes quantitatively of an intrusion tolerant system rather than model parameterization. They discussed several security related quality attributes and presented a feasibility to quantify security attributes. The work motivated the researcher to identify security attributes and quantify them.

David Gilliam et al. introduced a unified approach able to identify software weaknesses [15]. Unifying the model-based approach with property-based testing using temporal logic properties allows for consistent verification throughout life cycle phases. Anders Bond & Nils Pålhlsson presented a quantitative evaluation framework for component security in distributed information systems [16]. This framework reveals the objective, strength, and weaknesses of the security of components. The work provides a way to combine different aspects affecting the security components, such as the environment in which it operates, and functionality.

Riccardo Scandariato et. al. tried to give shape to the idea of security properties of software that are quantitative in nature with regard to assessment [6]. This allows proactive estimation of software security, especially during the architecture/design phases. They analyzed security principles that are relevant to the purpose of unearthing security properties and proposed suitable metrics to measure them. Dan Taylor and Gary McGraw worked on security improvement program and concluded that generally standard software process approaches focused on sequentially building a level of sufficiency in four areas in a particular order: process, controls, metrics, and improvement [4]. It has been inferred from the work that measurement and analysis
must be included much earlier in the process development model.

Konstantin Beznosov et. al. presented a report on security in industry perspective [1]. The report is based on the state of the practice and recent advances in engineering secure software for the wide range of industrial application domains. Author discussed the needs and importance of security requirements analysis and threat analysis and suggested that developer should properly collect and analyze security requirements [1].

Historically, security experts carry out security analysis process through the subjective evaluations. Early methods of security attribute analysis emphasis on code, models, and policies. Lots of work has been discussed related to the security measurement. But due to the lack of such a framework or guideline estimating security at the early stage, there is heavy demand to develop a security estimation framework in order to quantify security at the early stage of development life cycle.

4. Theoretical background

Literature survey on security estimation reveals that changes made at final stage is very expensive. Therefore, security estimation process should be integrated well in advance, preferably at design phase. Following points forms a strong theoretical basis in support of the methodology followed in the paper to develop a prescriptive framework for security estimation:

- M. Y. Liu’s concerns as none of the existing works have studied measurement properties from a security perspective [11].
- Peterson’s recommendation to analyze and integrate security at design phase, as it is one of the most crucial and flexible stages of SDLC [17].
- Stytz’s remark on software security estimation requirement to assess performance and degree of protection [18].
- R. Scandariato’s suggestion to make an effort to elicit the security factors [6].
- Diego’s recommendation to inspect the relationships between the design artifacts being able to separate them to maximize maintainability and reusability [19].
- Elizabeth’s remarks on the importance of design time metrics, as they stems from their ability to identify and characterize weaknesses early in the application’s life cycle, when such weaknesses cost much less to fix [20].
- Vaughn’s suggestion to develop effective metrics for systems security [11].
- Wang and Wulf’s motivation for theoretical validation of security metrics and possible security attributes [13].
- M. Y. Liu’s attempt to address limitations concerning architectural level security analysis [11].

Given the Liu’s concern, an effort has been made in the paper to address measurement properties from security perspectives. Lack of knowledge about which properties must be considered when it comes to evaluate security has been considered. It has been strongly argued by the researchers that, similar to quality attributes, security properties can be measured at the architectural level [6]. Scandarito’s suggestion strengthens the feasibility of identifying security factors. Diego’s recommendation motivated the authors to establish the relationship between design artifacts and security factors. The facts discussed above forms the strong basis for the proposed work. This paper will focus on security properties at design phase, design characteristics, design time security metrics, correlation between design characteristics and identified security factors using metrics. This appears to have been highly neglected in past work.

5. The framework

Taking into account the need and significance of estimating software security, a prescriptive framework is proposed. The framework may be used in design phase to predict software security quantitatively. A qualitative inference may be drawn on the basis of results obtained using the framework. Software design architecture will be used as an input to the five phased process. Goal of software Security Estimation Framework [SEF] is to provide high level protection to the software and contribute to the mitigation of security failures.

Security estimation process is the second stage of security estimation life cycle. Detailed description of life cycle may be referred at [3]. Security estimation process has been proposed on the basis of integral and basic components of software security. As shown in fig.1, Software security estimation process consists of five phases as follows:

1. Identify Security Factor
2. Identify / Design Metric Suite
3. Validate Metric Suite
4. Quantify Security Factor
5. Estimate Security
Phases of security estimation process are being shown in Figure 1. Overview of each phase is briefly described in the following section.

5.1 Phase I: Identify security factors

This Phase deals with various issues related to security attributes and presents some steps for identifying security factors. Security of software cannot be measured directly. It may be computed using attributes, metrics, and models [21]. Security attributes provide the basis for quantitative security estimation. Goal of this phase is to identify and extract a filtered set of validated security factors. These security factors are encountered in design phase. In this section, a set of activities has been defined which can be performed to identify security factors, as follows:

- Security Requirements & Validation [22]
- Security Factors Perspective
- Security Factor Selection Criteria
- Behavioral Requirements of Security Factors [23]
- Security Factors
- Verify Security Factors [23]
- Hierarchy of Security Factors
- Impact of Security Factors [14]
- Phase Wise Security Factors
- Security Factors In Design Phase.

5.2 Phase II: Identify / Design metric suite

Software runs properly only if it is fully secured against kinds of flaws, vulnerabilities, bugs, and threats. Heterogeneous architecture of software does not allow securitization of security properties at a glance. To meet the exact level of security assurance, quantification of security is required [26]. Security metrics are used to evaluate the robustness of software [27]. Security metrics may facilitate organizations providing decision capabilities if they offer quantitative and objective basis for security assurance [29]. By implementing security metric suite, secured software may be developed. Therefore, it is required to develop security metric suite capable of analyzing security level of software. Design time metrics have ability to identify and categorize weaknesses at early stage of software development life cycle [20]. In this phase, following activities are proposed to identify / design metric suite.

- Security Requirements And Validation [30]
- Classification of Vulnerability [10]
- Identify Software Characteristics
- Analyze Available Security Models [25]
- Analyze Available Security Metrics [25]
- Categorize Security Metrics [29]
- Specify Security Metric Measures
- Finalize Metric Suite
- Design Level Security Metric
- Design Level Security Metric Suite

5.3 Phase III: Validate metric suite

After developing a metric suite, it is desirable to validate the suite. Metric suite validation process integrates security factors, metrics, and functions. Metrics should be validated with respect to validity. Validity criteria provide rationale to validate metrics; they are the specific quantitative relationships that are hypothesized to exist between factors and metrics. Validity criteria support security functions, assessments, controls, and predictions.

In this phase, following metric suite validation activities are introduced using Norman’s concept [28].

- Specify Validation Goals
- Set Validation Standards
- Set Invalidity Criteria
- Identify Validation Framework [28]
- Validate Metric With Security Factors
- Collection of Empirical Data
- Theoretical Validation [26]
- Empirical Validation [26]
- Reliability Analysis
- Analysis and Interpretation
5.4 Phase IV: Quantify security factors

Whenever there is discussion on software security, emphasis is always on the security features such as authentication, encryption and access control. These features are important but they can not make system fully secured alone. It is necessary to quantify their impact on software security [1]. This phase concentrates on quantification of security factors. Quantification of security factors supports for rating of product security. It will not exactly predict the security but data provided with quantification results of security factors give the basis for security estimation. Following activities will be used to quantify security factors:

- Select Model
- Specify Criteria For Software Attributes
- Divide Factors Into Low Level Physical Attribute [13]
- Establish Link
- Set Regression Line
- Quantify Low-Level Security Attribute [13]
- Pre Tryout
- Review & Revision
- Tryout
- Component Sensitivity Analysis [24]

5.5 Phase V: Estimate security

Security of the software can be measured by analyzing the design activities, measurement of security attributes, and its impact on software. Security measurement affects the performance and quality of software [3]. Software security estimation is required to assess performance and the degree of protection [18]. Qualitative evaluation of security may no longer be acceptable [14]. Quantitative assessment of security at early stage of development process will provide the solid base to drive business decision and to enhance the role of security in the software development life cycle [25]. In this phase, following activities are introduced to estimate security.

- Conceptualization
- Contextual Findings
- Quantitative Analysis
- Risk Analysis [22, 24]
- Security Ranking
- Precautionary Measures
- Suggestions for Control & Improvement
- Estimate Cost Reduction [24]
- Estimate Effort Reduction
- Estimate Time Reduction

The first phase of Security estimation process starts with the identification of security factors. Security factors plays major role in security estimation process. Security metric is a high-level measurable attribute, and forms the basis for risk analysis and security management. Security metrics is to be identified or, if not available, will be designed in second phase. This identified or derived security metric suite will be validated. With the help of security metrics, security factors will be computed. Finally, security of the software will be estimated. On the basis of quantitative estimation of the software, qualitative interpretation will be made. Combined use of security factors, security metrics, security models etc. made it possible to estimate software security in design phase.

6. Motivations and future work

Security is an important feature and complex too. Development of security architecture for software is not a one time built-in process; it is based on reuse of existing security specifications. It is suggested by many researchers for the development of secured software security considerations should be integrated at the initial/preliminary stage of the software. Literature survey uncovers the fact that qualitative analysis of security may no longer be acceptable [14]. It is important to estimate security of software to stay in competitive environment. To achieve the purpose, quantitative estimation of security is required in development life cycle.

The paper presents a software security estimation framework for quantitative assessment of software security. All phases of proposed framework are quasi-experimental in nature having three components, each encompassing study, development and validation tasks in one or the other form.

7. Conclusions

A prescriptive framework to estimate software security has been proposed in the paper with theoretical basis. The proposed framework includes five phases. Overview of each phase is discussed in brief. The framework will support the design and analysis of non-functional properties for systems at the software architecture design level effectively. Security estimation framework proposed in this paper will detect and remove defects earlier, which in turn, will reduce development time, effort and budget. Quantitative assessment of security assists on predicting, controlling and improving software security in early stage of software development life cycle.

258
8. References


