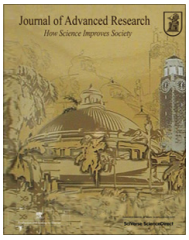
Journal of Advanced Research (2014) 5, 463–472

Cairo University

Journal of Advanced Research



ORIGINAL ARTICLE

Capturing security requirements for software systems

Hassan El-Hadary, Sherif El-Kassas *

Department of Computer Science & Engineering, The American University in Cairo, Egypt

ARTICLE INFO

Article history:
Received 6 October 2013
Received in revised form 1 March 2014
Accepted 3 March 2014
Available online 12 March 2014

Keywords:
Application security
Security requirements engineering
Security threat modeling
Problem frames

ABSTRACT

Security is often an afterthought during software development. Realizing security early, especially in the requirement phase, is important so that security problems can be tackled early enough before going further in the process and avoid rework. A more effective approach for security requirement engineering is needed to provide a more systematic way for eliciting adequate security requirements. This paper proposes a methodology for security requirement elicitation based on problem frames. The methodology aims at early integration of security with software development. The main goal of the methodology is to assist developers elicit adequate security requirements in a more systematic way during the requirement engineering process. A security catalog, based on the problem frames, is constructed in order to help identifying security requirements with the aid of previous security knowledge. Abuse frames are used to model threats while security problem frames are used to model security requirements. We have made use of evaluation criteria to evaluate the resulting security requirements concentrating on conflicts identification among requirements. We have shown that more complete security requirements can be elicited by such methodology in addition to the assistance offered to developers to elicit security requirements in a more systematic way.

Introduction

During the last decade, software systems security has become an increasingly growing concern due to the large number of incidents and attacks targeting software systems [1]. Attackers exploit software vulnerabilities and cause threats to the systems such as stealing sensitive information, manipulating data and causing denial of service. One of the grand challenges in information security is to develop tools and principles that allow construction of large-scale systems for important security critical applications such as e-banking systems and electronic voting systems [2].

Secure software development includes integrating security in different phases of the software development lifecycle (SDLC) such as requirements, design, implementation and testing. Early consideration for security in requirement phase helps in tackling security problems before further proceeding in the process and in turn avoid rework [3]. Several approaches have been proposed that upgrade previous requirement engineering approaches to let it support security such as goal oriented [4], agent oriented [5] and UML use case based [6].

Abbreviations: AF, abuse frame; SPF, security problem frame; PF, problem frame; SR, security requirement; AR, anti-requirement.

* Corresponding author. Tel.: +202 2615 2974.
E-mail addresses: sherif@aucegypt.edu (S. El-Kassas).

Peer review under responsibility of Cairo University.

2090-1232 © 2014 Production and hosting by Elsevier B.V. on behalf of Cairo University.
http://dx.doi.org/10.1016/j.jare.2014.03.001

In order to integrate security with requirement engineering, we have to consider security requirements. The basic task of security requirement engineering is to identify and document requirements needed for developing secure software system. Satisfying such *security requirements* should lead to more secure software system [7]. We adopted the definition that considers security requirements as constraints on the functionality of the system focusing on what should be achieved. We agree that the security requirements should be expressed as positive statements and not negative statements. Expressing requirements in such way can help in verifying its satisfaction [7]. Security requirements can be elicited by analyzing the assets to be protected and the threats from which these assets should be protected [8].

Security requirements need to be adequate as possible. They need to be explicit, precise, complete and non-conflicting with other requirements [4]. However, knowledge of security is a basic necessity prior to practicing security requirement engineering. The analyst should have background on how to identify and analyze the system assets, threats, vulnerabilities and requirements. One of the challenges for secure software systems development is to assist developers in performing security requirements engineering [9]. A more effective approach for security requirement engineering is needed to provide a more systematic way for eliciting adequate security requirements.

Problem frames [10] are means that can be used to reuse previous knowledge in modeling software problems in the requirement engineering process. Several approaches provide solutions to adapt security while following a problem frames based requirement engineering process such as abuse frames [11] and security problem frames [12]. Problem frames are used in different frameworks for identifying security requirements such as Haley's approaches [7,13]. However, we have a gap between such approaches. No integration is presented in the literature that bridges them together although they complement each other. This paper proposes a methodology for security requirement elicitation that provides a more systematic way for software developers in order to elicit adequate security requirements while following a problem frames based requirement engineering process. The methodology considers security while eliciting the requirements of software systems using problem frames. The main goal of the methodology is to assist developers to elicit adequate security requirements during the requirement engineering process with the aid of previous security knowledge. A security catalog, based on problem frames, is constructed for this purpose. The scope of the methodology is limited to the requirements phase in the SDLC, and is not intended to cover security through the entire SDLC.

This paper is organized as follows. Section "Related work" discusses related approaches for security requirement elicitation. Section "Methodology" presents our proposed methodology for security requirement elicitation. Section "Results and discussion" compares results of applying our methodology with two related methodologies. Section "Conclusion" summarizes our work and suggests areas for future work.

Related work

Different requirement engineering approaches are updated in order to consider security such as UML use cases [6,14,8], agent oriented [5,15], goal based [4,16] and problem frames based requirement engineering [7,12,17]. New models are introduced to represent threats that can be exploited by the attackers such as attack trees [18], misuse cases [6], anti-models [4] and abuse frames [11]. Moreover, threats classification and analysis techniques are introduced such as STRIDE and DREAD [19]. Thus, the approaches are updated to consider threats and elicit security requirements that mitigate such threats.

Moreover, reusing security knowledge is tackled in different approaches in order to assist software developers in eliciting security requirements in a systematic way. For example, security problem frames [12], misuse cases templates [3], and anti-models patterns [20] are used to form *generic* model based catalogs which are not specified for a particular application. Thus, the developer can make reuse of such generic models and templates during elaborating threats and security requirements.

Our methodology is mainly based on problem frames. In Section "Problem frames," we will cover problems frames and approaches that integrated security with problem frames.

Problem frames

Problem frames [10] are means that can be used in the requirement engineering process to describe software development problems. They can help in analyzing problems to be solved where interaction between the software and domains in the system context is described. Problem frames are useful in requirements engineering because they help in decomposing the system context into simpler subproblems which are mapped to well-known problem classes [21]. Thus, problem frames provide helpful means to reuse previous knowledge in modeling software problems including security related problems.

Different approaches provide solutions to integrate security while using problem frames based requirement engineering process. Abuse frames [11] and security problem frames [12] are means for modeling security problems. Moreover, problem frames are utilized in different frameworks for eliciting security requirements. For example, Haley's approaches [7,13] made use of problem frames in order to identify vulnerabilities and elicit security requirements.

Methodology

The proposed methodology aims at early integration of security with software development. It considers security while eliciting the requirements of software systems using problem frames. The methodology aims at identifying security requirements with the aid of previous security knowledge through constructing a security catalog for this purpose. The security catalog consists of problem frame models for *threats* and the corresponding *security requirements*. Threats are modeled using abuse frames while security requirements are modeled using security problem frames.

Section "The methodology steps" describes the methodology steps while giving examples for applying the methodology on a software banking system. Section "Methodology iterations and outputs" elaborates how the methodology iterates through its steps. Section "Security catalog" illustrates the structure and the contents of the security catalog used throughout the methodology.