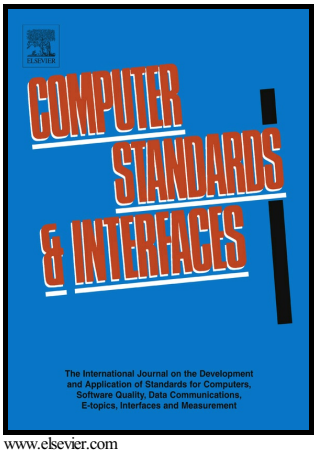
Author's Accepted Manuscript

Exploring Software Security Approaches in Software Development Lifecycle: A Systematic Mapping Study

Nabil M. Mohammed, Mahmood Niazi, Mohammad Alshayeb, Sajjad Mahmood



www.elsevier.com

PII: S0920-5489(16)30115-5
DOI: http://dx.doi.org/10.1016/j.csi.2016.10.001
Reference: CSI3147

To appear in: *Computer Standards & Interfaces*

Received date: 5 May 2016
Revised date: 3 October 2016
Accepted date: 3 October 2016

Cite this article as: Nabil M. Mohammed, Mahmood Niazi, Mohamma Alshayeb and Sajjad Mahmood, Exploring Software Security Approaches i Software Development Lifecycle: A Systematic Mapping Study, *Compute Standards & Interfaces,* http://dx.doi.org/10.1016/j.csi.2016.10.001

This is a PDF file of an unedited manuscript that has been accepted fo publication. As a service to our customers we are providing this early version o the manuscript. The manuscript will undergo copyediting, typesetting, an review of the resulting galley proof before it is published in its final citable form Please note that during the production process errors may be discovered whic could affect the content, and all legal disclaimers that apply to the journal pertain

ACCEPTED MANUSCRIPT

Exploring Software Security Approaches in Software Development Lifecycle: A Systematic Mapping Study

Nabil M.Mohammed[1], Mahmood Niazi[1,2], Mohammad Alshayeb[1] and Sajjad Mahmood[1]

[1]Information of Computer Science Department, King Fahd University of Petroleum and Minerals

Saudi Arabia

[2] Faculty of Computing, Riphah International University, Islamabad, Pakistan

alqadhi82@gmail.com.{ mkniazi, alshayeb, smahmood}@kfupm.edu.sa

ABSTRACT

There is an increase use of security driven approaches to support software development activities, such as requirements, design and implementation. The objective of this paper is to identify the existing software security approaches used in the software development lifecycle (SDLC). In order to meet our goal, we conducted a systematic mapping study to identify the primary studies on the use of software security techniques in SDLC. In total, we selected and categorized 118 primary studies. After analyzing the selected studies, we identified 52 security approaches and we categorized them in to five main categories, namely, 'secure requirements modeling', 'vulnerability identification, adaption and mitigation', 'software security focused process', 'extended UML-based secure modeling profiles', 'non UML-based secure modeling notations'. The results show that the most frequently used approaches are static analysis and dynamic analysis that provide security checks in the coding phase. In addition, our results show that many studies in this review considered security checks around the coding stage of software development. This work will assist software development organizations in better understanding the existing software security approaches used in the software development lifecycle. It can also provide researchers with a firm basis on which to develop new software security approaches.

Keywords: Systematic mapping study, Empirical Study, Software Development Life Cycle, Software Security

1. INTRODUCTION

In this modern age, software-intensive systems have become an important part of our lives. We highly depend on software systems in several areas of our daily activities, such as financial services, telecommunications, electronics, transportation, home appliances, and more. As the software system is involved in various aspects of society, security becomes an important issue and a vital requirement for the software system[1][2]. Many security issues such as confidentiality, availability and integrity need to be preserved in order to consider software as secure [3].

Traditionally, software security is considered only in the later stages of software development, by incorporating security concerns as an afterthought [4]. As a consequence, the risk of introducing new security vulnerabilities into various stages of software development lifecycles is increased. Following the traditional method of securing software has led to the 'Penetrate and Patch' approach, in which the security specialist tries to assess the software by breaking it from its environment via exploiting common security vulnerabilities. Successful penetration leads to patch development and deployment of the identified vulnerabilities. Security is mostly treated as an add-on feature in the software development lifecycle, and is addressed by security professionals using antivirus, platform security, proxies, firewalls, and intrusion prevention systems, [5][6].

The defensive mechanisms which are supplemented to a software system towards the end of the development cycle, such as intrusion detection systems and firewalls, are not enough and can lead to costly reworks [2]. Research has also shown that that such supplementary approaches to address security-related concerns are not sufficient and cal lead to significant number of changes in addition to any intangible consequences caused by a security breach [6]. To address these reworks and changes, security challenges need to be addressed from the beginning of software development lifecycles (i.e. from the software requirements gathering until software maintenance) [6]. To this end, secure software engineering has recently become a very active area of research. In software engineering different

Accepted manuscript

Download English Version:

https://daneshyari.com/en/article/4955063

Download Persian Version:

https://daneshyari.com/article/4955063

Daneshyari.com