Title: An Effective Computational Technique for Taxonomic Position of Security Vulnerability in Software Development

Author: Amit Kumar Srivastava Shishir Kumar

PII: S1877-7503(17)30891-8
DOI: http://dx.doi.org/doi:10.1016/j.jocs.2017.08.003
Reference: JOCS 736

To appear in:

Received date: 14-12-2016
Revised date: 18-7-2017
Accepted date: 8-8-2017

Please cite this article as: Amit Kumar Srivastava, Shishir Kumar, An Effective Computational Technique for Taxonomic Position of Security Vulnerability in Software Development, *<![CDATA[Journal of Computational Science]]>* (2017), http://dx.doi.org/10.1016/j.jocs.2017.08.003

An Effective Computational Technique for Taxonomic Position of Security Vulnerability in Software Development

Amit Kumar Srivastava[a,*], Shishir Kumar[b]

[a,b] *Jaypee University of Engineering & Technology, Guna 473226, India*

Abstract

An increasing demand of security standards in open networks and distributed computing environment have become critical issues for automation of the business process workflow.At automation level, it is a challenging task to methodically analyze the security constraint during the composition of business process component.For the complete automation of business process must scuritinez the flow of security patterns which consist of the bit value of the respective parameter which is the key entity for identifying the security vulnerability. There is various phase wise security patterns have been used to identify the security vulnerabilities during the black/white box testing phase of the service development. In respect of automation in business logic, this article introduces a machine learning computational technique that classifies the possible types of phase wise class categories of security vulnerability. The performance matrix along with comparative analysis suggests that the proposed approach proficiently matches the attack pattern to respective security pattern, which can classify phase wise class categories of security vulnerability in software component development.

Keywords: Attack Pattern, Principal Component Analysis, Data Pre-Processing, Normalization, Feed forward Back Propagation, Multilayer Perceptron

1. Introduction

In 2002, the National Institute of Standards and Technology published a survey about the knowledge of software engineers, stating that they spend an average of the 70-80% of their time in the information security risk management process(ISRM) during testing phase. The study reports that the testing and debugging process cost over $50 billion annually in the US only[1].The ISRM process presented as sequences of activities comprise planning, implementation, control and monitoring of application measurements and imposition of the security policies. The identified existing deficiencies [2, 3] in ISRM are estimated with the little reference to the organization's actual situation and performed on an intermittent, non-historical basis.

Nowadays, software developers are facing challenges in minimizing the number of defects during the software development. Using defect density parameter developers can identify the possibilities of improvements in the product. Since the total number of defects depends on module size, so there is need to calculate the optimal size of the module to minimize the defect density.This relationship could be used for optimization of overall defect density using an effective distribution of modules sizes[4, 5, 6].

Many software developers can work in parallel with the open source project using the web as a shared resource. The defect density of such projects is often required to be predicted for the purpose to ensure quality standards. Static matrix for defect density prediction require extraction of abstract information from the code. Repository matrix, on

Email address: amitsri1983@gmail.com (Amit Kumar Srivastava)
*Corresponding author