A New Approach to Locate Software Vulnerabilities Using Code Metrics

Mohammed Zagane, Université Oran 1, Algeria
https://orcid.org/0000-0003-2508-0265

Mustapha Kamel Abdi, Université Oran 1, Algeria

Mamdouh Alenezi, Prince Sultan University, Riyadh, Saudi Arabia
https://orcid.org/0000-0001-6852-1206

ABSTRACT

Automatic vulnerabilities prediction assists developers and minimizes resources allocated to fix software security issues. These costs can be minimized even more if the exact location of vulnerability is correctly indicated. In this study, the authors propose a new approach to using code metrics in vulnerability detection. The strength part of the proposed approach lies in using code metrics not to simply quantify characteristics of software components at a coarse granularity (package, file, class, function) such as complexity, coupling, etc., which is the approach commonly used in previous studies, but to quantify extracted pieces of code that hint presence of vulnerabilities at a fine granularity (few lines of code). Obtained results show that code metrics can be used with a machine learning technique not only to indicate vulnerable components wish was the aim of previous approaches but also to detect and locate vulnerabilities with very good accuracy.

KEYWORDS

Code Metrics, Machine Learning, Vulnerability Detection, Vulnerability Prediction Models

1. INTRODUCTION

Exploiting software vulnerabilities cause most of the information security issues. Manuel detection of these vulnerabilities is a time-consuming tedious task and very costly in terms of time and budget. To assist developers and minimize these costs, tools that can automatically predict vulnerable components must be used to let developers focus their efforts on most likely vulnerable components. These costs can be minimized even more if the used tools have the capacity to identify the exact location of vulnerabilities.

Researchers have proposed many approaches to automatically predict vulnerabilities. One of these approaches consists of using software metrics as indicators of vulnerabilities. In this field of vulnerability prediction, software metrics are used to quantify software characteristics such as complexity, coupling, etc. and to build vulnerability prediction models (VPMs) based on machine learning techniques. Researcher’ aim was to evaluate hypotheses that are a correlation exist between...
software characteristics and vulnerabilities such as the one that code complexity is the enemy of software security. The main limit of such approaches is that the proposed VPMs do not have the capacity of detecting the exact location of vulnerabilities in the source entity at much smaller granularity. Because metrics are calculated at a coarse level of granularity (package, file, class, function). Instead, they only predict if the source entity is vulnerable or not. Another limit which is the reason behind not widely adopting VPMs is they give predictions of low accuracy or with high false positives or high false negatives.

In a recent study (Li, Zou, Xu, Jin, et al., 2018; Li, Zou, Xu, Ou, et al., 2018), researchers proposed an interesting way to deal with vulnerabilities. The authors proposed the concepts of Code Gadgets (CGs), Syntax-based Vulnerability Candidate (SyVC) and Semantics-based Vulnerability Candidate (SeVC) which are pieces of code composed of semantically related lines that can hint the existence of vulnerabilities. Instead of analyzing the whole code of software entity, they proposed to focus on CG1. They used this technique to represent the program in a fine granularity of CG which lead to pin down the locations of vulnerabilities and to produce a text-mining-based vector representation of programs to train a deep-learning-based system of vulnerability detection.

In this study, the authors propose a new approach to using code metrics with a machine learning technique in vulnerabilities prediction. The proposed approach is based on the recently proposed concept of CG and aims to improve VPMs that use code metrics as input data. The contribution of this study is threefold:

- **Proposing a new approach to detect and locate vulnerability using code metrics and machine learning**: The strong point of the proposed approach lies in using code metrics not to simply quantify characteristics of software components such as complexity, coupling, etc. which is the approach in previous studies, but to quantify extracted pieces of code that hint presence of vulnerabilities at a fine granularity (CG);
- **Proposing a dataset of code metrics generated from labelled CGs**: As part of the study, authors propose and make publicly available a dataset of code metrics generated from labelled CGs, which can be used by others researchers to evaluate and develop VPMs based on machine learning and deep learning;
- **Validating the concept of CG in other contexts**: The concept of CG was initially proposed and used with text-mining-based vector representation to develop a deep learning system of vulnerability detection. The present study also makes validation of this concept using machine learning and code metrics.

The remainder of this paper is organized as follows: Section 2 presents background and the most relevant related work, Section 3 describes the proposed approach and the methodology followed to carry out the study, Section 4 presents the experimentations, Section 5 presents the obtained results and discussion, Section 6 presents the limitations of the study, Section 7 summarizes the work done in this study and indicates some perspectives.

2. BACKGROUND

2.1. Vulnerability Prediction Model (VPM)

A software vulnerability is a specific type of software defect that affects information security and privacy in a software system after successful exploitation by attackers. Therefore, techniques used for defect prediction are also used for vulnerability prediction. One of these techniques is DPM (Defect Prediction Models) (Basili, Briand, & Melo, 1996; Gao, Khoshgoftaar, Wang, & Seliya, 2011; Menzies, Greenwald, & Frank, 2007; Syeed, Hammouda, & Systä, 2014; Turhan, 2007). DPMs are
A Software Engineering Framework for Context-Aware Service-Based Processes in Pervasive Environments
www.igi-global.com/chapter/software-engineering-framework-context-aware/77700?camid=4v1a

On the Design of a Knowledge Management System for Incremental Process Improvement for Software Product Management
www.igi-global.com/article/design-knowledge-management-system-incremental/70925?camid=4v1a