Code Vulnerability Identification and Code Improvement Methods using Advanced Machine Learning

L.O. Ruggahakotuwa[1], S.R. Liyanage[2]

Dealing with cyber-attacks has become a routine task of modern information systems. The misconfigurations of source code can result in security vulnerabilities that potentially encourage attackers to exploit them and compromise the system. The security of a software critically depends on its underlying source code because hackers always hunt for the loopholes of the software which reflects the vulnerabilities of the source code. To mitigate the above-identified threats the researches have produced several commercially used tools such as Vera code, OWASP, source clear, etc. But still, the frequency of threats and data breaches is very high. 'Veracode' is capable of doing both static and dynamic analysis but it is very expensive software. 'OWASP' uses only the static analysis to automate the detection of the vulnerabilities. 'Source clear' is capable of scanning the repositories either manually or automatically. As all the above-mentioned tools are in their testing phase, number of false positives of the results can be high. In this research, we investigated how to automatically identify the software vulnerabilities by conducting a live scan to detect the error fragments and how to correct the detected source code vulnerabilities automatically at the development stage. This system consists of mainly two phases, Error Detection, and Error Correction. Error Detection is done through a live scan. In the live scan, both Static analysis and Dynamic Analysis will run in parallel. In the dynamic analysis the source code was run in the background and checked with random input data. In Static analysis, the source code was checked line by line and verified by another Rule-Based Engine. The source code is highlighted with markers based on the two outputs of static and dynamic analysis. Research analyzed several machining learning models for better accuracy and performance. After the most suitable machine learning model was identified, the model was trained with enough training samples to develop a generalized model. The final system was implemented to identify vulnerable code segments in Java source codes and suggest corrected code fragment to the developer.

Keywords: Vulnerability; Machine learning; CVE; Static Analysis; Dynamic Analysis

[1] Department of Computer Systems Engineering, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka, *laneesha.r@sliit.lk*
[2] Department of Software Engineering, University of Kelaniya, Kelaniya, Sri Lanka