Project proposal - Discovering Software Vulnerability Prediction Models

Gilberto Recupito
Università degli Studi di Salerno
Email: g.recupito@studenti.unisa.it

Dario di Dario
Università degli Studi di Salerno
Email: d.didario@studenti.unisa.it

Abstract

Software vulnerability is a particular type of software defect that defines a weakness in the system that can be exploited by an attacker to change its functioning. An attack on the system can have considerable consequences for the system in terms of damage and costs. Vulnerabilities also arise during the software development phase, as certain types of systems are really complex. Although the vulnerabilities are different from defects, they can be defined as a subset of the latter and it's not possible to affirm that a software system doesn't have vulnerabilities. For this reason software solutions for recognize them are born, obtaining a general view of the project state, so than it is possible to act and try to resolve the problem. These tools are generally called Vulnerability Prediction Model (VPM). So, we present in this paper an approach based on collecting the information of existing VPMs, rebuilding them and creating a performance benchmark. The next step is to illustrate how they work and compare the difference approaches to highlight the features of each VPMs and analyze the results provided in terms of precision and recall.

Keywords

Vulnerability, Prediction, Metrics, Security, Software, Open Source, VPM, Precision, Recall, Accuracy, Benchmark.

1 Introduction

A software vulnerability can be defined as a weakness in the software system that can be exploited by an attacker in order to change the behaviour of the system. Because the number of software systems increases everyday also the number of vulnerabilities increases. An attacker can use vulnerabilities to access into the system and he might take control to damage it,- as launching new attacks or obtaining some privileged information that he can use for his own benefit. Considering this, it is important to know the different types of vulnerabilities, their prevention and detection in order to try to avoid their presence in the final software version of the system and then reduce the possibility of attacks and costly damages. For instance there are many type of vulnerabilities, like:

- Buffer overflow
- XSS
- SQL Injection

Vulnerability prediction models (VPM) are believed to hold promise for providing software engineers guidance on where to prioritize precious verification resources to search for vulnerabilities. VPM is a relatively recent field of study which aims at automatically classifying software entities as vulnerable or not.
Neuhaus et al. [1] was among the first vulnerability prediction approaches. The authors discovered a correlation between import/function calls and vulnerabilities and used it to form a prediction approach, using the includes and function calls of the files under analysis. In other words, the includes and function calls were the features used to train a classifier. Their results on the Mozilla Firefox project showed that a recall of 45 % and a precision of 70% can be achieved. Zimmerman et al. [2] presented an empirical study on Windows Vista to evaluate the efficacy of code churn, code complexity, dependencies and organizational measures to build a vulnerability prediction model. They obtained good precision but low recall. Gegick et al. [3] used the warnings of security tools along with complexity metrics to build their prediction models. Morrison et al [4] presented a vulnerability prediction model based on 5 categories of metrics: churn, complexity, dependency ,legacy and size metrics. Morrison used machine learning techniques to build VPMs.
For instance he uses:

- Logistic regression (LR)
- Naive Bayes (NB)
- Recursive Partitioning (RP)
- Support Vector Machine (SVM)
- Tree Bagging (TB)
- Random Forest (RF)

These techniques are applied on different versions of Windows 7 and windows 8 and the obtained results are compared, showed and collected in terms of precision and recall. Precision is the probability that a file classified as vulnerable is indeed vulnerable. Recall is the probability that a vulnerable file is classified as such.

Basically, the goal of our study focus on collecting the information of existing VPMs, rebuilding them and creating a performance benchmark. The next step is to illustrate how the VPMs work and compare the difference approaches to highlight the features of each VPMs and analyze the results provided in terms of precision, recall and accuracy.

2 Motivation

During the development phase of a software, given the continuous growth and evolution of the software, it's easy that there are vulnerabilities. The prediction of these vulnerabilities, even before the software can enter the market, therefore allows the software to avoid leaving exploitable weaknesses by the attackers.

3 Aims

The aim of this project is to analyze and re-implement the VPM methodologies, to compare and study the performances in terms of precision, accuracy and recall, thus creating a benchmark. The steps that we will follow in order to obtain the aims of this project are: Finding a set of open-source existing projects in order to tests the several VPMs, extracting vulnerabilities for the selected projects, describing how the actual situation is about vulnerability prediction and defining the effectiveness of the studied models.

References

[1] S. Neuhaus, T. Zimmermann, C. Holler, and A. Zeller, “Predicting vulnerable software components”.In: Proceeding of the 14th ACM Conference on Computer and Communication Security (October 2007)

[2] T. Zimmermann, N. Nagappan, and L. Williams, “Searching for a needle in a haystack: Predicting security vulnerabilities for windows vista,” in ICST'10

[3] M. Gegick, P. Rotella, and L. Williams, “Predicting Attack-prone Components,” in ICST'09.

[4] P. Morrison, K. Herzig, B. Murphy, L. Williams, ”Challenges with Applying Vulnerability Prediction Models”.