Prospect of static analysis methods for detecting security vulnerabilities in Python

Hristina Gulabovska[a], **Zoltan Porkolab**[b]

[a]Eotvos Lorond University, Faculty of Informatics, Dept of Programming Languages and Compilers
hristina@gulab.me

[b]Eotvos Lorond University, Faculty of Informatics, Dept of Programming Languages and Compilers
gsd@elte.hu

Abstract

Security considerations have primary importance when developing software systems. Security issues affect the code quality and if not detected in development time they may have serious negative impact, e.g. leaking customer data, disruption of service, etc. To detect code vulnerabilities, static analysis is increasingly popular as an extension of traditional validation methods as testing and code review. Static analysis is an automated software verification method analyzing the source code without executing it for detecting software errors. Various static analysis tools have been successfully applied for detect vulnerabilities in languages with a static type system, like C, C++ and Java. There are, however, an increasing demand for similar activities for software implemented in Python. Python is a programming language with a dynamic type system, used in many emerging areas, including data science, machine learning and web applications. In this paper we investigate the possibilities of static analysis in term of detecting security vulnerabilities in Python code. As the dynamic behavior of Python language requires different approaches compared to the existing methods in languages with static type system, we review the existing tool support for detecting security vulnerabilities, and identify those rules which could be successfully covered by new checkers.

Keywords: software technology, Python programming languages, security, static analysis