Qualitative and Quantitative Evaluation of Static Code Analysis Tools

Lakshmi Manohar Rao Velicheti[1], Dennis C. Feiock[1], Rajeev R. Raje[1], James H. Hill[1]
[1]Department of Computer and Information Science, Purdue School of Science

Static code analysis (SCA) is a methodology of detecting errors in programs without actually compiling the source code to binary format and executing it on a machine. The main goal of a SCA tool is to aid developers in quickly identifying errors that can jeopardize the security and integrity of the program. With the vast array of SCA tools available, each specializing in particular languages, error types, and detection methodologies, choosing the optimal tool(s) can be a *daunting* task for any software developer, or organization. This, however, is not a problem associated only with SCA tools, but applies to any application domain where many tools exist and a selection of a subset of these tools is needed for effectively tackling a given problem.

To address this fundamental challenge with selecting the most appropriate SCA tool for a particular problem, this research is performing a comprehensive study of different available SCA tool, both commercial and open-source. The end goal of this study is to not only evaluate how different SCA tools perform with respect to locating specific errors in source code (*i.e.*, the quality of the tool), but to model the behavior of each SCA tool using quantitative metrics gathered from the source code, such as source lines of code (SLOC), cyclometic complexity, and function points. The behavioral model can then be used to prescreen existing (and new) source code, and select the most appropriate SCA tool, or set of SCA tools, that can identify the most errors in the source code undergoing analysis.