This thesis deepens the research about MACKE, a framework for compositional program analysis based on symbolic execution with the KLEE engine. MACKE decomposes the analyzed program into small units, each containing one function from the program with all its dependencies. These units are analyzed with KLEE in order to expose vulnerabilities inside the function. Then, MACKE matches all discovered problems with the same reason and uses additional KLEE runs aiming to propagate the problem to the entry point of the program. Finally, all results are refined to an error report for the complete program. This thesis is divided into two main parts: The first part revisits the algorithms inside MACKE and discusses improvements covering several weak points in the old implementation. Among others, it refines the unit generation, simplifies the error matching and introduces a new search strategy for KLEE. The second part uses the new implementation to analyze 16 different real-world projects. With the generated data, it shows the efficacy of the proposed changes and compares the performance with pure KLEE analysis. Thereby, several benefits for the uncovered vulnerabilities and the overall program coverage are illustrated.
Quartal: 4. Quartal

Hochschule / Universität:
Technische Universität München; Ludwig-Maximilians Universität; Universität Augsburg

Occurences:
- Einrichtungen > Fakultäten > Fakultät für Informatik > Lehrstühle der Informatik > Informatik 22 - Lehrstuhl für Software Engineering (Prof. Pretschner)