Static Slicing of Binary Executables

Ákos Kiss, Judit Jász, and Gábor Lehotai

Program slicing is a technique for determining the set of statements of a program that potentially affect the value of a variable at some point in the program. Static slicing computes slices using static analysis, without making any assumptions regarding the sliced program´s input, while dynamic slicing computes slices for specific test inputs. Both static and dynamic techniques of intra and interprocedural slicing of high-level languages has been greatly studied in the literature.

Analysis of machine code is somehow different from the analysis of high-level languages since central notions as control structures and variables are missing. At machine code level on the architectures of our days the analysis has to work with fully unstructured control flow, registers and memory locations.

In this paper we explain how to apply intraprocedural static analysis to binary executables and we introduce a method to extend the usual analysis of registers to the local stack image of procedures. This way the local variables usually residing on stack can be taken into account during analysis. The extension of the conventional technique enables us to create more precise but still safe slices of binary programs.

The result of the analysis can be useful for reverse engineering tools of binary programs.