VYPe – Compiler Construction
Reconstruction of Data Types for Decompilation
lecture notes abstract

Bc. Peter Matula
xmatul01[at]stud.fit.vutbr.cz

November 7, 2012

Keywords

decompilation, reverse engineering, Lissom, LLVM, ISAC, type analysis, data type reconstruction

Abstract

Decompilation is a process of transforming a machine code into a higher-level programming language. It can be used for source code reconstruction, malware detection, compiler testing, etc. In this presentation I introduce a retargable decompiler, which is being developed by the Lissom project at FIT BUT. The basic principles and infrastructure of such tool are presented and the data type reconstruction problem is discussed in detail. Several existing methods of reverse engineering of types are compared and an algorithm using the data-flow analysis is described. This iterative algorithm reconstructs basic data types of the objects by constructing a set of equations over the program's instructions and solving them using the fixed propagation rules derived from the instruction properties. It uses a concept of lazy rule application which simplifies right-to-left propagation proposed by the autor. The presentation is concluded with a mention of future work on reconstruction of the composite data types.

References

[1] E. N. Dolgova and A. V. Chernov. Automatic reconstruction of data types in the decompilation problem. *Program. Comput. Softw.*, 35(2):105–119, March 2009.

[2] JongHyup Lee, Thanassis Avgerinos, and David Brumley. Tie: Principled reverse engineering of types in binary programs. In *NDSS*. The Internet Society, 2011.

[3] Alan Mycroft. Type-based decompilation (or program reconstruction via type reconstruction). In *Programming Languages and Systems, 8th European Symposium on Programming, Amsterdam, The Netherlands, 22-28 March, 1999, Proceedings*, volume 1576 of *Lecture Notes in Computer Science*, pages 208–223. Springer, 1999.

[4] Katerina Troshina, Yegor Derevenets, and Alexander Chernov. Reconstruction of composite types for decompilation. In *Proceedings of the 2010 10th IEEE Working Conference on Source Code Analysis and Manipulation*, SCAM '10, pages 179–188, Washington, DC, USA, 2010. IEEE Computer Society.

[5] Lukáš Ďurfina, Jakub Křoustek, Petr Zemek, Dušan Kolář, Tomáš Hruška, Karel Masařík, and Alexander Meduna. Design of a retargetable decompiler for a static platform-independent malware analysis. In *The 5th International Conference on Information Security and Assurance*, Communications in Computer and Information Science, Volume 200, pages 72–86. Springer Verlag, 2011.