# Interprocedural Analysis:
## Basic Concepts and Why?

Tomáš Fiedor
xfiedo01

Marek Fešar
xfesar00

16.10.2012

## Abstract

A program is not a static block of code. It consists of many cooperating functions, which can be called from many possible contexts. This shows the need for examining interprocedural dependence during static code analysis. The basic way to perform the interprocedural analysis, as well as its possible difficulties, are shown in this work. Next part focuses on real world usage of this analysis and its importance for code optimization or error detection. Finally, a brief example of application code demonstrates one of the most dangerous vulnerabilities that can be discovered by interprocedural analysis – the buffer overflow – and how it can be exploited, if not properly fixed.

## Keywords

Static Analysis, Compilers, Data Flow Analysis, Interprocedural Analysis, Buffer Overflow.

## Literatura

[1] The MITRE Corporation.: Common Weakness Enumeration – CWE-120: Buffer Copy Without Checking Size of Input ('Classic Buffer Overflow'). [online], Last Updated 14. 5. 2012, [cit. 2012-10-16]. Available at: http://cwe.mitre.org/data/definitions/120.html.

[2] AHO, A.; LAM, M.; SETHI, R.; et al.: *Compilers: Principles, Techniques and Tools.* Pearson Education, 2006, ISBN 978-0321486813.

[3] KHEDKER, U.; SANYAL, A.; KARKARE, B.: *Data Flow Analysis: Theory and Practice.* CRC Press, 2009, ISBN 978-0849328800.