# Static Taint Analysis for PHP

November 2016

| | |
|---|---|
| candidate | Pavel Jurásek |
| supervisors | Martin Steffen and Daniel Fava Schnetzer |
| group | PMA |
| type | 60 ECTS |
| recommended background | programming languages, program analysis and compilers, PHP |
| study program | computer science |

## Short description

The task is to design and implement a static analysis for PHP, improving security aspects of the language.

## Background and motivation

PHP [1] ("PHP: Hypertext Prepocessor") is a general-purpose scripting language, widely used for webpages and server-side applications. As a scripting language, it is interpreted and relies heavily on *dynamic,* i.e., run-time checks, as opposed to static type-checking or other static analysis methods. There is a trend, through, that even for dynamic scripting languages such as PHP, the roubustness and security of applications are increased by adding *static* analyses and checks to the language. One example in this setting are the attempts to equip PHP with a so-called *gradual* type system.

## Problem setting

In this project, the task is to design and implement a *static* data flow analysis, in particular a taint analysis, with the goal of preventing standard security flows like SQL-injections. The aim, specifically and in contrast to existing such analyses, is to target the *object-oriented* features of the language

1. Get a overview over PHP (especially its oo features), its semantics, and available PHP tools (including existing parsers).

2. Define the *scope* of the covered features and

3. Design the taint analysis for the features chosen to be covered. A planned focus will be on object-oriented aspects of the language.

4. Implement the analysis in PHP.

5. Evaluate the implementation wrt. performance and scalability as well as precision (i.e., how many false alarms it may report).

**Keywords:** program analysis, PHP, static analysis, security analysis

## References

[1] M. Achour, F. Betz, A. Dovgal, N. Lopes, H. Magnusson, G. Richter, D. Seguy, J. Vrana, and et.al. PHP manual. available at `http://php.net/manual/en/index.php`, Nov. 2016.