Tainted object propagation analysis for PHP 5 based on Pixy

Diploma thesis

Oliver Klee
Bonner Str. 63, 53173 Bonn
pixy@oliverklee.de

Bonn, January 2, 2013

Rheinische Friedrich-Wilhelms-Universität Bonn
Institut für Informatik III
Professor Dr. Armin B. Cremers

universitätbonn

Contents

1 Introduction

1.1 Motivation: Why a current static PHP security scanners is important

Currently, there is no free and high-quality static code analysis tool available (and still maintained) that can find vulnerabilities in PHP 5.4.x code. This is a problem because new vulnerabilities in web applications are found almost daily [osv11], and PHP is used for more than 75 % of the top-million sites [W3T12a], including Facebook (using the HipHop PHP compiler [Zha10], Wikipedia and WordPress.com [W3T12b].

1.2 Research problems and approach

This thesis tackles the following research goals:

- Create an alias analysis that takes PHP 5's pass-by-reference for objects by default into account.
- Enhance the lexer and parser (both part of PhpParser) with most of the new keywords and concepts introduced in PHP 5.0 through 5.4.
- Analyze the security ramifications of the new keywords and concepts introduced in PHP 5.0 through 5..

1.2.1 Technical goals

In addition to the research goals, there are a few technical goals that needed to be achieved in order to achieve the research goals mentioned above:

- Adapt Pixy to work with Java 7 without any warnings. (Pixy was created using at most Java 6, but probably only using Java 1.5.)

- Get Pixy to parse PHP 5 code in the first place. (Pixy currently could handle PHP code only up to PHP version 4.2.)

- Enhance Pixy to also load PHP class files that are not directly included, but are supposed to be loaded via a PHP autoloader.

The technical goals are mostly necessary due to the fact that the Pixy code base had not been maintained (or even touched) since 2006, and both PHP (i.e., the scanned language) as well as Java (i.e., the scanner's language) had evolved in the meantime. In addition, the product code should be maintainable and well-structured so that it will be of real future use instead of a throw-away prototype.

2 PHP

2.1 Challenges in static analysis for PHP

3 Vulnerabilities in PHP web applications

4 Static analysis

5 Review of existing static PHP vulnerability scanners

6 Pixy

7 PHP 5.4

8 Alias analysis for the new default pass-by-reference in PHP 5

9 Implementation details and problems encountered

10 Experimental evaluation of the modified version of Pixy

11 Discussion

11.1 Related work

11.2 Conclusions

11.3 Further work

Bibliography

[osv11] OSVDB: The Open Source Vulnerability Database. http://osvdb.org/ (retrieved 2011-01-10), 2011.

[W3T12a] W3Techs. Usage of server-side programming languages for websites. http://w3techs.com/technologies/overview/programming_language/all (retrieved 2012-11-16), 2012.

[W3T12b] W3Techs. Usage statistics and market share of PHP for websites. http://w3techs.com/technologies/details/pl-php/all/all (retrieved 2012-11-16), 2012.

[Zha10] Haiping Zhao. HipHop for PHP: Move Fast. https://developers.facebook.com/blog/post/2010/02/02/hiphop-for-php--move-fast/ (retrieved 2012-11-16), 2010.

List of Figures

List of Tables