Plan for a Master's Project

Title	Taint analysis to detect injection vulnerabilities using Rascal
Document version	0.3.0 (third draft version)
Student	Ruud van der Weijde | 10453857
Host organization	Werkspot | www.werkspot.nl | Technology | Amsterdam
Contact person	Winfred Peereboom | Technology Director | winfred.peereboom@werkspot.nl | 06-46446888
Project summary (short version)	I will do a static taint analysis to find injection vulnerabilities in the codebase of Werkspot. The implementation will be done in Rascal. First I will try to analyze the old part of the system, using ZendFramework and Symfony (both first versions of the frameworks) , and doctrine12 (or something). If possible, I will expand the investigation to the entire codebase, containing over 2.5 million lines of code.
Research questions (main ones)	- Can injection vulnerabilities be found in the software of Werkspot using taint analysis in Rascal. - How much of PHP can be analyzed. Which parts cannot be resolved. Can application specific settings provide better results?
Research method	Implement in Rascal in the following steps: - Global analysis (no restrictions) - Find out how the compiler works - Define and add taint restrictions - Define and add untain restrictions - Apply to inter-procedural flow - Apply to intra-procedural flow - Add application based settings to optimize results - Include optimization. - Object construction optimization. - Other optimizations to improve analysis results. - Deal with false pos/neg results. - Produce final results
Expected results	Thesis, analysis tool, human readable results of analysis.
Required expertise	PHP, Rascal, taint + static code analysis, PHP compiler
Global timeline	- Jan: - Finish the project plan. - Feb: - Thesis: background, problem description, motivation. - Rascal: basic proof of concept in Rascal - Mrt: - Thesis: describe analysis method. - Define taint and untaint definitions. - Rascal: Focus on inter-procedural analysis. - Apr: - Thesis: update analysis method, write about result analysis. - Rascal: Also apply intra-procedural analysis. - May:

	- Thesis: update thesis, add first results to thesis. - Rascal: apply application based settings. - Jun: - Thesis: add results to thesis. - Rascal: Handle false (pos/neg) results. - Jul: - Thesis: Finish, write conclusion, future work, limitations - Aug: - Room to fix delays in the progress.
Main risks	- Too ambitious: provide better scope on the project. - Unclear project goals: define exactly what is expected. - Too many false positives: better define what is a possible threats. Add more constraints during the processing

Project summary:

According to the TIOBE index, PHP the 6th most popular programming language. PHP is used in about 4.6% of all programs.
PHP is difficult to fully analyze because PHP is a dynamic programming language. Due to variable includes, variable variables, variable function calls, variable callback functions and eval function, it is hard to make call graphs by static analysis.
In PHP it is possible to write code that will have injection vulnerabilities. This injection can happen from the outside (through user input) or from the inside (by programming structures).
In this thesis I will try to find methods to analyze PHP code to identify injection vulnerabilities. I hope to find a way to measure accurately and keep the false positive and false negatives to the minimum.
To be able to provide Werkspot results of this research, I want to display the analysis on the software in a readable format. This month (January) I want to investigate if it is possible to create a plugin for an existing analysis tool (for instance in Sonarqube) to provide extra information on the software analysis done by the analysis tool.
Research questions:

- Is static (taint) analysis precise enough to find vulnerabilities. How much information can dynamic analysis add to find more precise results, resulting in less false positives.
- How many (dynamic) includes can be resolved? Can application specific analysis upgrade the number of resolved includes?
- Is it possible to use Rascal for taint analysis, to find vulnerabilities? (Note: I first want to find out more about dynamic analysis.)
- Is the method of choice (is to be defined) capable of dealing with programs having multi million lines of code (like Werkspot has).

Give a short (at most 1 page) description of the project. Make sure that the following issues are covered:

- *The global context of the project,*
- *The relevancy of your research,*
- *The specific research questions to be answered in the project.*

Explain the project.

Problem analysis:

The website of Werkspot is created in 2005 and evolved afterwards. New features and components are added over time. The frameworks (Symfony and Zend

Framework) on which Werkspot is build are no longer supported. Because Werkspot strives for external (product quality, user experience) and internal (code quality/complexity) quality, the existing code should be upgraded to a newer framework. The framework of choice isSymfony 2.
Symfony 2 is a thin, light weight framework which should lead to better maintainable code, because the code will be easier to read and have less coupling. This should raise the internal quality of the software reduce the maintenance costs.
In the legacy code of the old framework, Werkspot is interested in the software parts which contain the most complex code, security-, and/or performance issues. When Werkspot is provided with indication numbers on complexity, security and performance, Werkspot can decide what components should be replaced before other components.
Because PHP is a dynamic language it is not easy to analyze it. The hardest parts are dynamic includes, aliases, variable variables, function callbacks, and type inference. More detailed information can be found in the literature study.
Here you present your analysis of the problem situation that your research will address. How does this problem manifest itself at your host organization? Also summarizes existing scientific insight into the problem (result of your literature survey, see below).

Research method:

The difficulty in this research is that PHP is a dynamic language and therefor had to fully statically analyze PHP code.
I expect from the literature survey to get answers on how to analyze PHP code and how injection detecting/locator algorithms are applied in other programming languages. Algorithm of other programming languages might be applicable to PHP. The source of my literature will be academic literature.
The research I will do is to see how I can analyze PHP code to find injection vulnerabilities.

The proof of concept I will create is a small program to detect injection problems in PHP by a few simple test cases. This will for instance be injection problems in variable function calls, variable variables and eval usage.
I think the search method will be Action Research, which is defined as: "the researchers attempt to solve a real-world problem while simultaneously studying the experience of solving the problem". (TODO: this needs to be confirmed. Might be an experiment or case study. Not sure about this.)
Hypothesis are to be defined. (TODO: define hypothesis)
Timeline:

- Jan: literature study, project plan, proof of concept.
- Feb-Mei: write thesis, find injection vulnerabilities using method(s) x(,y) (TODO: find out how I will do this.).
- Jun-Jul: finish thesis, present the data to Werkspot in usable form (as information, not data).

The validation of the research will be done by creating a number of test cases with different injection problems and test cases without injection problems. With these test I will be able to verify the used method and will be able to identify false positive and false negative results.
Another option to validate the results if to compare the results with existing tools to find injection. For this validation I will use the tools: x, y, z (TODO: add tools to validate, these are to be defined.)
Present how you are going to find the answers to your research question. This chapter should cover:

- *What will make the research difficult?*
- *What is the input you expect from the literature survey*

- *What sources will you use and how will you use / document them?*
- *What experiments / research will you do?*

What proof of concept will you make?

- *What method will you use?*
- *Which hypothesis do you have?*
- *Present a time line*
- *How will you validate your research?*

Expected results of the project:

- Successfully being able to correctly find injection vulnerabilities.
- Present the results in a readable way to Werkspot, for instance as a Sonarqube plugin, so Werkspot can use it to analyze their software.
- High quality thesis.

Give an explicit list of all the results that are expected from the project.

Required expertise for this project:

- PHP language knowledge
- Taint analysis
- Statical code analysis
- Research validation (using statistics?)
- …
- (TODO: More to be defined.)

Give a list of the expertise that is needed for the successful completion of this project. Examples are:

- *Java programming*
- *Make and Makefiles*
- *Configuration management*
- *Testing*
- *Etc., etc.*

Timeline:

(TODO: when more detailed knowledge of the plan is known, update the timeline with more details.)
Jan: write project plan, finish: 31 jan.
Feb: write thesis, detect injection in PHP.
Mrt: write thesis, detect injection in PHP.
Apr: write thesis, detect injection in PHP.
May: write thesis, detect injection in PHP, provide results.
Jun: write thesis, provide results.
Jul: write thesis, provide results.
Aug: room for possible delay.
An overview of what activity will take place when, and what milestones/deadlines the project has.

Risks:

(TODO: update these risks when more information is available.)

- Project goals are too ambitious.
- Project goals are not concise enough.
- Used method provides insufficient results (too many false positive) to detect injection.
- (TODO: Make the items above measurable, like SMART. Will do this when more project information is available.)

Give an assessment of the risks for this project and present counter measures. These may include:

- *Project goals are too ambitious.*
- *Project goals are too vague.*
- *The information about the project is unavailable/incomplete/too difficult.*
- *The gap between your expertise and the required expertise is too large*

Literature survey (can also be a separate document):

See attached file.

Describe the following for related scientific literature on your topic:

- *Brief summary of contents,*
- *Relation to your topic: how does the work described in the reference differ from your approach, what results have they obtained, what open questions do they still have?*

A guideline is to include between 10 and 20 papers on your topic in the survey. The exact number depends on the topic and available literature.

Bibliography:

See attached file.