# Bachelor/Master Thesis Topic

# Grammar-Based Fuzzing for Libre Office

**Motivation and Background**

Software testing is a crucial part of software development. It enables assurance, of correctness, completeness and reliability of software systems. Current state-of-the-art software testing techniques employ symbolic execution-based whitebox fuzzing techniques to generate automatically test inputs that expose software bugs. In practice, many programs require highly structured and complex files. The generation of such valid files is non-trivial and, hence, traditional whitebox fuzzing often lead to bad results spending too much time exploring parser errors, instead of the functional part of the program. Recent approaches like [1] and [2] try to tackle the problem by building a more accurate model of a valid program input, e.g. so-called grammar-based whitebox fuzzing requires the input grammar that is used to generate valid test inputs. The work in [3] shows how input grammars can be mined successfully by leveraging dynamic tainting.

**Goals**

The goal of this project is to guide fuzzing based on a probabilistic grammar for Libre Office documents.

**Description of the Task**

- Getting familiar with the probabilistic grammar for Libre Office
- Implement the probabilistic grammar in a common fuzzing tool, eg. AUTOGRAM [3]
- Perform an experimental evaluation of the mined grammar by using whitebox fuzzing

**Research Type**

| | |
|---|---|
| Theoretical Aspects: | ****✱ |
| Industrial Relevance: | ***✱✱ |
| Implementation | ***✱✱ |

**Prerequisite**

The student should be enrolled in the bachelor/master of computer science program, and has completed the required course modules to start a bachelor/master thesis.

**Skills required**

Programming skills in Java or C++, understanding of, or willingness to learn, the software engineering and software analysis foundations needed for the project.

**References**


[1] Patrice Godefroid, Adam Kiezun, and Michael Y. Levin. 2008. Grammar-based whitebox fuzzing. In Proceedings of the 29th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '08). ACM, New York, NY, USA, 206-215.

[2] Van-Thuan Pham, Marcel Böhme, and Abhik Roychoudhury. 2016. Model-based whitebox fuzzing for program binaries. In Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering (ASE 2016). ACM, New York, NY, USA, 543-553.

[3] Matthias Höschele and Andreas Zeller. 2016. Mining input grammars from dynamic taints. In Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering (ASE 2016). ACM, New York, NY, USA, 720-725.


**Contacts**


Lars Grunske, Humboldt-Universität zu Berlin, Institut für Informatik, Lehrstuhl Software Engineering, Unter den Linden 6, 10099 Berlin, Germany


**Application**

Please contact me during my office hours or send me an email with the title: "[ThesisProject]-GrammarBasedFuzzingofLibreOffice" to se-career@informatik.hu-berlin.de