FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Review report of a final thesis

Reviewer:	Pierre Donat-Bouillud, Ph.D.
Student:	Bc. Matúš Ferech
Thesis title:	Efficient fuzz testing of web services
Branch / specialization:	Computer Systems and Networks
Created on:	1 June 2023

Evaluation criteria

1. Fulfillment of the assignment

[1] assignment fulfilled
▸ **[2] assignment fulfilled with minor objections**
[3] assignment fulfilled with major objections
[4] assignment not fulfilled

The final thesis has fulfilled the assignment: it surveys the state of the art of fuzz testing of web services, presents methods for minimization of the outputs of fuzzing.
The state of the art on minimization is less explored, mainly focusing on internal reduction inspired by Hypothesis (within two rust libraries, proptest, and minithesis), but external reduction, including (possibly hierarchical) delta debugging, should have been briefly explained. The resulting open-source fuzzer is efficient, performs payload minimization and measure the roundtrip time of endpoints, thus making it possible to find potential endpoints susceptible to DoS attacks. The fuzzer is thoroughly tested on 3 different real-world web services.

2. Main written part

92 / 100 (A)

The thesis is easy to read, well structured, understandable.
Spelling mistakes are rare; the most notable is in the title of 5.1. I also found an orphan Table (Table 4.1 in section 4.9) that was not referred to anywhere in the text.

In the section about the architecture of openapi-fuzzer, it is claimed it is stateless. Stateful fuzzers encover bugs that openapi-fuzzer could not but the author of the thesis says that he "believe[s] these bugs are relatively rare". I was not totally convinced here; I would expect more qualitative and quantitative arguments.

What a bug is for openapi-fuzzer is not totally clear to me:
- in the implementation section, it says an interesting payload is 5XX status code, or status code not among expected status codes defined in the OAS

- Table 4.1 shows 5XX and semantic for error reporting
- in the testing and evaluation section, it says only 5XX are counted as bugs, but then that 5XX and 4XX codes will be counted separately

Table 6 shows n/a for the running times of openapi-fuzzer on Vault and Vault authentificated but the text states the fuzzing times were long, so I expected to see actual running times (in minutes?).

The master thesis builds upon the bachelor thesis of the student, which can be challenging with respect to citation rules. However, when the student reuses actual text from the bachelor thesis, which rarely happens, it is properly cited. More generally, sources are properly cited.
The code uses open-source libraries (MIT/BDSD or Apache licenses) and is itself licensed under AGPL: those licenses are compatible.

3. Non-written part, attachments 95 /100 (A)

The code of the fuzzer is stored on github. It is coded in Rust, using several established rust libraries, such as the relevant proptest and arbitrary libraries to help with the payload generation and minimization.

The payloads generated for the testing and evaluation section are stored in a gitlab repository, along with the commands used to run the fuzzer on the 3 web services. To make it fully reproducible, there should also be a script that would automatically set up the fuzzed web services with the right environment. The seeds should also be stored: I can see that the commands included in the gitlab repository save the seeds but the actual seed do not seem to be actually in the repository.

4. Evaluation of results, publication outputs and awards 100 /100 (A)

The thesis could probably result in a publication, as it outperforms the state of the art in terms of number of bugs found.

The first version of openapi-fuzzer is popular, with more than 400 stars on github, so I expect this new version to be as much used.

The overall evaluation 95 /100 (A)

As a whole, this thesis is an excellent work, well written, and interesting.

Questions for the defense

- Schemathesis exhibited a lot of internal errors: what are they?
- How easy would it be to make openapi-fuzzer stateful, to fuzz sequences of requests?

Instructions

Fulfillment of the assignment

Assess whether the submitted FT defines the objectives sufficiently and in line with the assignment; whether the objectives are formulated correctly and fulfilled sufficiently. In the comment, specify the points of the assignment that have not been met, assess the severity, impact, and, if appropriate, also the cause of the deficiencies. If the assignment differs substantially from the standards for the FT or if the student has developed the FT beyond the assignment, describe the way it got reflected on the quality of the assignment's fulfilment and the way it affected your final evaluation.

Main written part

Evaluate whether the extent of the FT is adequate to its content and scope: are all the parts of the FT contentful and necessary? Next, consider whether the submitted FT is actually correct – are there factual errors or inaccuracies?

Evaluate the logical structure of the FT, the thematic flow between chapters and whether the text is comprehensible to the reader. Assess whether the formal notations in the FT are used correctly. Assess the typographic and language aspects of the FT, follow the Dean's Directive No. 52/2021, Art. 3.

Evaluate whether the relevant sources are properly used, quoted and cited. Verify that all quotes are properly distinguished from the results achieved in the FT, thus, that the citation ethics has not been violated and that the citations are complete and in accordance with citation practices and standards. Finally, evaluate whether the software and other copyrighted works have been used in accordance with their license terms.

Non-written part, attachments

Depending on the nature of the FT, comment on the non-written part of the thesis. For example: SW work – the overall quality of the program. Is the technology used (from the development to deployment) suitable and adequate? HW – functional sample. Evaluate the technology and tools used. Research and experimental work – repeatability of the experiment.

Evaluation of results, publication outputs and awards

Depending on the nature of the thesis, estimate whether the thesis results could be deployed in practice; alternatively, evaluate whether the results of the FT extend the already published/known results or whether they bring in completely new findings.

The overall evaluation

Summarize which of the aspects of the FT affected your grading process the most. The overall grade does not need to be an arithmetic mean (or other value) calculated from the evaluation in the previous criteria. Generally, a well-fulfilled assignment is assessed by grade A.