SOEN 7481 Software Verification and Testing

Assignment, Winter 2025

1 Assignment Description.

In this assignment, you will need to implement a simple coverage-guided fuzzer. Fuzzing is commonly used to uncover various types of bugs (e.g., security) and has received much attention from both the industry and academia. You will need to use Python and Jupyter notebook for this assignment. You can extend the Jupyter notebook file that we covered in the class.

2 Coverage-guided Fuzzing.

You will need to implement a *smarter* way to create fuzzed input for your program. From *Introduction To Fuzzing*, we only covered how to get coverage information when running fuzzed inputs. However, we are still randomly generating fuzzed inputs. In the first part of the assignment, you need to implement an algorithm to be smarter when generating the fuzzed inputs.

- You need to maintain a population (i.e., a list of seeds) that you can use for mutation. Each seed in the population will be an input that provided additional coverage information.
- You will take a seed from the population and fuzz the seed to test the program. If the population is empty, you can mutate an initial seed that you provide manually.
- You need to take the seed from the population semi-randomly, but the seeds that provided more additional coverage have a higher chance of being selected.
- If the new input increased the coverage, add this new input to the population as a new seed. Assign a score to the seed (i.e., how much coverage it improved).
- Repeat and go back to Step 2 until you execute your fuzzer X times (i.e., your pre-defined number of trials).

3 Testing your fuzzer.

You need to run your fuzzer on 3 Python programs. You can choose any reasonable size program that you find (e.g., the program cannot be a simple `print 'Hello World'`). You can find existing programs or you can write the programs yourself.

For example program, you need to try different fuzzing approaches that we covered in the class, in addition to the fuzzer you wrote. You need to discuss and compare each fuzzing approach (e.g., pure randomly vs coverage-guided) in terms of coverage. You also need to plot how the coverage increases for the inputs that you tried (i.e., the same as the plot that we showed in class).

In addition to coverage, if your program found any bug (e.g., unexpected failure or crash), please record it.

4 Final report.

You need to use GitHub to version-control your Jupyter notebook. You can see how to put Jupyter on Github. Your final report will be the Jupyter file that you worked on. Please include descriptions for different steps that you take when working on the assignment.

5 Hosting code and report on GitHub.

Everything needs to be stored on GitHub, including all of your code, team information etc.

Please make your repository private. Please share your repository with the TA and the instructor (petertsehsun@gmail.com).

6 Submitting your assignment.

We will use Moodle for the submission. More details to come.

7 Evaluation

The final evaluation will be based on all factors in the deliverables. We will look at the quality of your code and report.