Spring 2015 - CS 4/510 - Topics in Software Validation

Due: June 5 2015

Project 4: Tests Generation Using KLEE Symbolic Executor

This is the exam instruction for project 4. This exam aims to evaluate your understanding on the workflow of using KLEE, its basic usage on self-contained program and its advanced usage on practical programs, all of which are covered in the practice of this project.

Part1: Workflow and Basic Usage of KLEE

In this part, you are asked to use the KLEE you built on a self-contained program, from which you will demonstrate that you managed to setup the whole environment of KLEE and have a good understanding on the workflow and basic usage of KLEE. Please download the source code of kmp.c shipped along with this document, and perform the following tasks. The points for each task and their required submissions are also specified as follows.

1. Tasks:
    a. Introduce symbolic variables to the source of the program by using KLEE's intrinsics. (10%)
    b. Compile the modified source file to llvm bit-code. (10%)
    c. Apply KLEE with appropriate arguments on the bit-code you got from the previous task to generate more than 1000 test cases. (20%)
    d. Use ktest-tool to show one test case in human readable format. (10%)
    e. Use klee-stats to show the coverage of llvm instructions. (10%)
2. Submissions:
    a. A report in plain text, "***report_part1.txt***". It should contain: the command line you used to perform task c ~ d and the corresponding important output. Just copy those information from your terminal to the report. It will be very similar to the terminal output that you have seen in KLEE's tutorial. Also please check the sample of report shipped with this document.
    b. The modified ***kmp.c*** that has symbolic variables introduced.
    c. The compiled bit-code, ***kmp.bc***.
    d. The output folder of klee, "***klee-last/***". Be aware that "klee-last" is only a soft link to the real output folder. Please copy and submit the folder that contains all the outputs you got by using klee on kmp.c.
    e. Please put all the files mentioned above into a folder named "***part1***".

Part2: Advanced Usage of KLEE

In this part, you are asked to use KLEE to test one of the GNU CoreUtils, "**du**". You should already have everything built and be ready to run klee on it directly without extra efforts, if you have done the step 3 of this project practice. You should be able to find the native executable of "du" that was compiled with coverage flag in folder "coreutils-6.11/obj-gcov/src". Also, you should be able to find the bit-code of "du" in folder "coreutils-6.11/obj-llvm/src$". Please perform the following task and prepare the requested submissions as well.

Spring 2015 - CS 4/510 - Topics in Software Validation

Due: June 5 2015

1. Tasks:
    a. Apply KLEE with appropriate arguments on "du.bc" to generate at least 200 test cases. (20%)
    b. Use klee-replay to replay those test cases generated from KLEE on the native executable of "du". (10%)
    c. Use gcov to produce the coverage of native "du" after replaying all the test cases generated by KLEE. (10%)
2. Submissions:
    a. A report in plain text, "***report_part2.txt***". It should contain: the command line you used to perform task a ~ c and the corresponding important output.
    b. The following files: **du, du.gcno, du.gcda, du.bc**
    c. The output folder of klee, "***klee-last/***".
    d. Please put all the files mentioned above into a folder named "***part2***".

For the final submission, please compress those two folders "***part1***" and "***part2***" into a single zip file and upload it to D2L. Thanks for all your efforts.