

Assignments for Introduction to SW Security, 2020.

Robert Iakobashvili

Version 1.2, dd. 30.04.2020

1. Submission

Submit the assignments only via Moodle box until the deadline.

There is no submission "after the deadline".

For exceptional cases, contact your Department. The Department is the body that is contacting lecturers with requests for exceptional treatment confirming that the requests are complying with the university rules and regulations.

Submission of the exercises is individual.

2. Submission material

Include source files, jar files for Java tests, making, building files, projects, etc. and a README.txt file stating in English how to build the files and how to run the tool.

State in each README and source file your name and ID.

Arrange a folder/directory and within it two subfolders: one JAVA, another C each with relevant assignments.

Compress your folder/directory with all files to a single submission file using zip, gzip, tar.gz or tar.

3. Java Assignment

From the CERT guidelines below, demonstrate the two issues:

[ERR08-J. Do not catch NullPointerException or any of its ancestors](#)

and

[EXP01-J. Do not use a null in a case where an object is required](#)

- 3.1 Develop your own examples, two test cases, reproducing and demonstrating the issues in your program when working **with files**. Use Java-8 for your examples.
- 3.2 The program to be written in line with standard Java coding conventions and well commented by spelled English.
- 3.3 Be well prepared for the frontal exam including in-depth understanding of every line of your code and testing of its running capabilities.

4. C-Assignment. GCC Sanitize Options.

Modern gcc compiler includes `-fsanitize=<option>` options for catching potential issues.

<https://gcc.gnu.org/onlinedocs/gcc/Instrumentation-Options.html>

Write a small C-language program including four functions each demonstrating a different issue of your choice flagged by GCC-sanitizer.

Each of the four functions to be:

- 4.1 Compiling correctly and not flagged by gcc compiler without `-fsanitize=<option>`;
- 4.2 Relevant `-fsanitize=<option>` shall flag the issue.
- 4.3 The submission to include:
 - 4.3.1 C-program with four functions clearly named.
 - 4.3.2 Makefile or building instructions with and without `-fsanitize` options.
 - 4.3.3 Readme file with the version of gcc compiler (output of `gcc -v`) that you used.
- 4.3 Be well prepared for the frontal exam including in-depth understanding of every line of your code and testing of its running capabilities.

Note: Make sure that the version of gcc compiler you are using supports the options you are demonstrating.