CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

Security Testing
WS 2023/2024

Prof. Dr. Andreas Zeller
Leon Bettscheider
José Antonio Zamudio Amaya

Exercise 0 (10 Points)

Due: 5 November 2023 23:59

Submit your solutions as a Zip file on your status page in the CMS.

We will provide you a structure to submit your solutions where each task has a dedicated file. You can add new files and scripts if you want, but you may not delete any provided ones. You can verify whether your submission is valid by running:

```
python3 verify.py
```

The output tells you whether a required file, variable, or function is missing and whether a function pattern was altered. If you do not follow this structure or change it, we cannot evaluate your submission. A non evaluable exercise will result in 0 points, so make sure to verify your work before submitting it. Note that the script does not reveal if your solutions are correct.

Exercise 0-1: First Slithering (4 Points)

In this exercise you will install Python and run it.

a. A strange Snake (0 Points)

Install Python 3.12 by following the steps on https://wiki.python.org/moin/BeginnersGuide/Download.

b. Run, Python, Run! (2 Points)

Run the python script **exercise_1b.py** and report the output in **solution_1b.txt**.

Tip: On an Unix system you can directly write the output of a program into a file by piping it.

```
program > output_file.txt
```

c. The Framework (2 Points)

We will provide a simple framework for your submissions. For each assignment we provide a **verify.py** file that validates whether all files, functions, and variables that are required for the assignment exist in your project. We ask you to verify your submissions before handing in, because we can only grade valid submissions. For this exercise, please execute **verify.py** and report its output to **solution_1c.txt**

Exercise 0-2: A large Planet (2 Points)

a. My Binder (0 Points)

You can directly edit, experiment, and try the code from The Fuzzing Book on a mybinder server. To do so, please navigate to the chapter about Introduction to Software Testing, hover over Resources, and click on Edit as Notebook. Try out the entire code in the notebook. You can follow these steps with any other chapter of The Fuzzing Book. Do not despair, the server requires some time to start (≈30 seconds).

b. Jupyter (0 Points)

Install Jupyter Notebooks by following the steps on https://jupyter.org/install . We recommend to install JupyterLab via pip, i.e. `pip3 install jupyterlab`

c. My own Notebook (2 Point)

Start a Jupyter notebook server on your machine by running jupyter notebook from the directory of this exercise sheet. Your browser should automatically open the index page of the server, if not open the link provided by jupyter notebook. Open the notebook **exercise_2b.ipynb** in the shown environment and follow the next steps.

1. Create a new markdown cell with an arbitrary content.
2. Execute the two code cells.
3. Delete the markdown cell that has the content `Delete this cell`.

Do not forget to save your changes.

Exercise 0-3: The Fuzzing Book (4 Points)

a. Package (0 Points)

Install the Python package from The Fuzzing Book by running

```
pip3 install fuzzingbook
```

This package allows you to access each function and class from The Fuzzing Book's code from within your Python programs by importing it.

Make sure that you are always on the latest version of `fuzzingbook` throughout the course.

The latest version number can be found at `https://pypi.org/project/fuzzingbook/`.

To find our which version of `fuzzingbook` you have installed, run `pip3 show fuzzingbook`.

b. Try me (4 Points)

Edit the python script **exercise_3b.py**, such that it imports the Fuzzing: Breaking Things with Random Inputs chapter and produces data with a simple random fuzzer. Use the fuzzer function to do this. The script will then write the produced data to a file **solution_3b.txt**. Include this file with random data in your submission. You can take a look at the file and see how different executions produce independent random data.