Hw 3. Symbolic Execution

Deadline: 2021/5/7

A. Use Angr to find the exploit for stack buffer overflow

A stack buffer overflow occurs when a program writes to a memory address on the program's call stack outside of the intended data structure.
In this question, we have prepared a vulnerable C program and the script (simple_exploit.py, full_exploit.py) that invokes the Angr[1] symbolic execution engine to find the input (exploit) for triggering the stack overflow vulnerability in the C program.

```
while sm.active:
    sm.step()
if sm.unconstrained:
    for un in sm.unconstrained: # 1-1
        print  ( "stdout:\n" + bytes(un.posix.dumps(1)) )
        print  ("stdin:\n" + un.posix.dumps(0) + "\n")
```

Have a look at the script, in particular the lines marked with Python comment symbol "**#**". You may want to have a look at Angr documentation to help you understand the script.

1. Describe the overall code structure of the script.
2. Explain the purpose of the code on the lines marked with comment symbol.
3. Test the exploit(input) on the C program and show your results.

B. Enhanced example

In this question, you need to reduce the time for running the Angr script.

1. Rewrite the script (exploit.py) to speed up its execution. Describe your improvements briefly.
2. Run the new script and show your results.
3. Measure and compare the execution time of the new script versus the original script. ($ time python exploit.py)

[1] https://angr.io/

C. Submission

- For each question, you should submit a report with a screenshot of the results and a short description.
- Put your files in directory{Student_ID} and compress as {Student_ID} .zip
- Only submit the zip file to e3.
- Directory example:

```
0866017
| -- Question1/
    | -- report.pdf
    | -- exploit.py
    | -- ...
| -- Question2/
    | -- …
```

- 5% score penalty for homework submitted in wrong formats.
- Write down your thoughts as a basis for the bonus point.
- You can answer in Chinese or English.