**CSE 410/518 Software Security**

**Instructor: Ziming Zhao  
Homework – 2**

**Reading. Read the following materials.**

[ ] Reading Task 1: Read chapter “Interlude: Process API” in “Operating Systems: Three Easy Pieces” at https://pages.cs.wisc.edu/~remzi/OSTEP/cpu-api.pdf

**Hands-on Tasks.**

**Your username on cse410.cacti.academy: \_\_\_\_\_\_\_\_\_**

[4 points] Task 1: Use **pmap** to show the memory maps of **processmap\_32** and **processmap\_64**. Take screenshots. Briefly explain the outputs.

[5 points] Task 2: Use **strace** on a program you choose, e.g., **firstflag**, **ls**. Take a screenshot. Explain the (1) purpose, (2) parameters, and (3) return value of at least 3 system calls from the output.

[6 points] Task 3: Analyze the program **re\_1\_32**. You don't have access to the source code. You can use any reverse engineering tools, e.g., objdump, GDB, Ghidra, IDA Pro free, or binary ninja cloud, etc. Use ltrace, strace, strings, and any tools you can think of to get a feeling what this program does. Hint: You are supposed to find a secret. Describe what are the expected input for this program. Find the main function, describe what it does.

[6 points] Task 4: Based on what you find on Task 3, get the flag by exploiting **re\_1\_32**. You are supposed to find a secret. Briefly describe how you find the secret of this program and what is the secret? Take screenshots.

[6 points] Task 5: Replicate what the instructor did in class. Exploit **overflowlocal1\_32** and **overflowlocal1\_64**. Take screenshots and explain why your exploit works.

[6 points] Task 6: Replicate what the instructor did in class. Exploit **overflowlocal2\_32** and **overflowlocal2\_64**. Take screenshots and explain why your exploit works.

[6 points] Task 7: Capture the flag by exploiting **overflowlocal3\_32** OR **overflowlocal3\_64**. Take screenshots and explain why your exploit works.

[6 points] Task 8: Capture the flag by exploiting **overflowlocal4\_32** OR **overflowlocal4\_64**. Take screenshots and explain why your exploit works.