**CSE 410/510 Software Security**

**Instructor: Ziming Zhao  
Homework – 3**

**Reading. Read the following materials.**

[ ] Reading Task 1: Where the top of the stack is on x86. https://eli.thegreenplace.net/2011/02/04/where-the-top-of-the-stack-is-on-x86/

[ ] Reading Task 2: Stack frame layout on x86-64 https://eli.thegreenplace.net/2011/09/06/stack-frame-layout-on-x86-64/

[ ] Reading Task 3: Using (cat $file; cat) to run a simple BOF exploit <https://security.stackexchange.com/questions/155844/using-cat-file-cat-to-run-a-simple-bof-exploit>

**Hands-on Tasks.**

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

[4 points] Task 1: In a function that is using x86 **cdecl** convention (32-bit), explain what are stored at the following memory locations: 1) [ebp], 2) [ebp+4], 3) [ebp+8], 4) [ebp+0xc], 5) [ebp-8].

[4 points] Task 2: Write down the instructions for **cdecl** function prologue and epilogue. Explain what each instruction does.

[4 points] Task 3: Compare the 32-bit and 64-bit of the fiveparameters challenges. Use objdump to disassemble the binaries. Take screenshots of the instructions of “func” and the parameter passing in “main”. Explain how the argument passing are different for the 32-bit and 64-bit versions.

[6 points] Task 4: Finish challenge overflowret1 32-bit and 64-bit. Take screenshots.

[6 points] Task 5: Finish challenge overflowret2 32-bit. Take screenshots.

[6 points] Task 6: Finish challenge overflowret3 32-bit. Take screenshots.

[6 points] Task 7: Finish challenge overflowretchain 32-bit and 64-bit. Take screenshots.

[9 points] Task 8: Use the information on the slides to login overthewire behemoth1, use buffer overflow techniques to crack the program behemoth1, and get a shell. You do not need to read the password of behemoth2.