**CY 5770 Software Vulnerabilities and Security**

**Midterm CTF**

**Total 160 Points + Bonus Points**

**Your full name:**

**Your CTF platform username:**

**You are allowed to use google, or refer to lecture slides, homework, code during the exam. You can use online disassemblers, such as Binary Nanji.**

**But, you cannot communicate with anyone in or outside of the class. You cannot use LLM, e.g., ChatGPT, either.**

**For each challenge, you should clearly show your exploit, screenshot of successful exploitation, and a very brief description of how you did it. Even if you fail to capture the flag, you may get some points for documenting your steps.**

**Some hints: Use “file” to check if the executable is 32-bit or 64-bit. Use “./checksec --file executable-name” to see if the stack is executable or if stack cookie is inserted. Use “strace” to see what system calls a program makes.**

**All four challenges are under the Midterm25Spring category.**

**The first person to capture each flag receives a 5-point ‘first blood’ bonus.**

1. [40] (Medium to Hard) Capture the flag of ***re\_6\_64***.

a. [4] Is stack executable? Can you overwrite RET address on stack? Is there a canary to protect the stack? (You can use checksec.sh, objdump, etc., to find out)

b. [4] Where does this program take input?

c. [4] Describe your high-level idea on how to exploit this challenge.

1. [40] (Medium) Capture the flag of ***overflow10\_64***. Source code is provided.

a. [4] Is stack executable? Can you overwrite RET address on stack? Is there a canary to protect the stack? (You can use checksec.sh, objdump, etc., to find out)

b. [4] Where does this program take input?

c. [4] Describe your high-level idea on how to exploit this challenge.

1. [40] (Medium) Capture the flag of ***overflowret12\_32***.

a. [4] Is stack executable? Can you overwrite RET address on stack? Is there a canary to protect the stack? (You can use checksec.sh, objdump, etc., to find out)

b. [4] Where does this program take input?

c. [4] Describe your high-level idea on how to exploit this challenge.

1. [40] (Easy) Capture the flag of ***re\_7\_64***.

a. [4] Is stack executable? Can you overwrite RET address on stack? Is there a canary to protect the stack? (You can use checksec.sh, objdump, etc., to find out)

b. [4] Where does this program take input?

c. [4] Describe your high-level idea on how to exploit this challenge.