**CSE 410/510 Software Security**

**Midterm CTF**

**Oct 24, 2022**

**Total 160 Points**

**Your full name:**

**Your CSE410/510 CTF platform username:**

**You are allowed to use google, or refer to lecture slides, homework, code during the exam. But, you cannot communicate with anyone in or outside of the class.**

**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.**

1. [40] Exploit challenge-1 to capture the flag. The source code of this challenge is in /code

 a. [4] Is this a 32bit or 64-bit program?

 b. [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)

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

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

1. [40] Exploit challenge-2 to capture the flag.

 a. [4] Is this a 32bit or 64-bit program?

 b. [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)

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

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

1. [40] Exploit challenge-1 to capture the flag.

 a. [4] Is this a 32bit or 64-bit program?

 b. [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)

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

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

1. [40] Exploit challenge-2 to capture the flag. The source code of this challenge is in /code

 a. [4] Is this a 32bit or 64-bit program?

 b. [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)

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

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