**CSE 410/510 Software Security**

**Midterm CTF Part 1**

**March 14, 2022**

**Total 80 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.**

**Multiple choice questions. Highlight the correct answer. 2.5 points for each.**

1. Which one of the following descriptions about the Intel architecture RET instruction is correct?
	1. The RET instruction pops whatever EBP/RBP points to to EIP/RIP
	2. The RET instruction pops whatever ESP/RSP points to to EIP/RIP
	3. The RET instruction checks if EBP/RBP points to is a valid code address, if yes it pops the value to EIP/RIP
	4. The RET instruction checks if ESP/RSP points to is a valid code address, if yes it pops the value to EIP/RIP
2. Which one of the following descriptions of the Intel architecture CALL instruction is correct?
	1. The CALL instruction pushes the destination address onto the stack
	2. The CALL instruction only moves the destination address into EIP/RIP
	3. The CALL pushes the address of the instruction after call onto the stack, then moves the destination address to EBP/RBP
	4. The CALL pushes the address of the instruction after call onto the stack, then moves the destination address to EIP/RIP

**CTF Challenges.**

**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.**

1. [35] 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?

 b. [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?

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