CS 361S Midterm Exam #1
Fall 2020
Professor: Seth Nielson

ETHICS STATEMENT:

1. I understand that I am permitted to use any static resource for this exam, such as books, slides, homework assignments, and so forth.
2. However, if I quote or closely paraphrase any source, I am required to cite it.
3. I am NOT allowed to talk to any other human beings including students or non-students. I will keep my email, chat programs, and other forms of communications closed during the exam.
4. I understand that if I am found to have communicated in violation of this ethics statement, or have plagiarized outside sources, I will receive a zero for this exam as well as other potential academic penalties.
5. I understand that Dr. Nielson is not particularly moved by crying, begging, or pleading. I understand that he has failed students for cheating in the past, even if it impacted their ability to continue in a program, get student loans, or other very serious consequences. I understand that if I am caught cheating, the impact of the consequences is not a consideration.
6. I understand that Dr. Nielson may choose to submit samples of my exam answers to various automated analysis systems for comparison in order to establish authorship.
7. I understand that Dr. Nielson may compare my writing samples to other submissions I have made in order to establish authorship.
8. I understand that if I am aware of ANY reason why my work might seem plagiarized or it might otherwise appear that I am violating the rules of this test, it is MY responsibility to speak up as soon as possible and set the record straight.
9. I understand that I am responsible for executing good judgment. If I have any questions about what is or is not ethical for this test, it is MY responsibility to ask.
10. I understand that if I cheat and don't get caught, I will have to be the one looking at myself in the mirror for the rest of my life knowing that any "success" I achieved was through lying and deceit. I understand that these kinds of character decisions can impede my self confidence, reduce my effectiveness in my career, and potentially lead to greater crimes and consequences.

To accept this Statement of Ethics, please type in your submission, "I <fill in your name>, have read Dr. Nielson's Ethics Statement. I understand the statement and fully agree with its principles. I will be bound by these principles while completing this exam."

INSTRUCTIONS:

1. You have the normal class time of 75 minutes to complete this exam.
    a. The Short Answer should take about 15 minutes
    b. The Essay should take about 60 minutes
2. You should create your answers in a text file and upload via Canvas
3. Please keep email, chat, and other communications programs closed

Short Answer Questions (30 points).

Write a short answer to each of the following questions. In most cases, the answer can be a single sentence. In all cases, it should never be more than three sentences.

Use your own words. While this is an open book test, you cannot just copy a quote from somewhere else. Make sure that the answer is based on your understanding, not just a repeat of what someone else said.

1. (5 points). Briefly explain why the attack presented in Ken Thompson's paper, "Reflections on Trusting Trust" cannot be defeated by a code review of the compiler used to compile the login program.

2. (10 points). Briefly explain how to use a nop slide to deal with variations in the exact address of the beginning of attack (shell) code. Specifically, describe the overflow address chosen, the shell code address chosen, and why the nop slide fixes the variations.

3. (15 points). Suppose we have the following ROP attack with gadgets similar to Lab 3:

    0 4 8 12 16 etc

    [G1][G2][0xFFFFFF40][G3][G4][G5][G6][G7][256 bytes of data]

    G1: mov ebx, esp; ret

    G2: pop ecx; ret

    G3: and ecx, 0x0ff; ret

    G4: add ecx, ebx; ret

    G5: xor eax, eax; ret

    G6: inc eax; ret

    G7: mov dword ptr [ecx], eax; ret

    Describe what this attack code does. It should only take 1 sentence. You may assume that for any given gadget, the ESP **has already advanced** before the gadget executes (that is, when G1 is

executing, ESP has already advanced to address 4). Be specific. If it does something with or to memory, you need to identify the address of that memory and what is done to it.

In writing your answer, if you need to describe a memory address on the stack, you may just use the relative offsets shown above the stack diagram. For example, the G1 gadget address is stored on the stack at location 0. The address for G2 is stored on the stack at memory location 4.

It might be easy to have some off-by-one errors (or off by 4 errors) in your answer. So in addition to the 1 sentence answer of what this chain does, you can explain any reasoning for your calculations of memory addresses. You will get most of the credit if your explanation is reasonable, even if you get something wrong in the final calculations.

Also, the number 256 in the data region has no special meaning. I just needed a relatively big number and arbitrarily picked 256. Don't get stuck on it.

ESSAY (70 points).

Choose ONE of the following questions. Write an essay of at least 5 paragraphs in answer to the question. While there is no specific word count, there must be sufficient detail or your answer will be marked down.

1. Do you think fuzzing would be a good way to detect a trojan compiler (as described by Ken Thompson)? If so, provide enough technical detail to explain how the fuzzer would/could catch it. If not, provide enough technical detail to explain why fuzzers aren't good at catching this kind of problem.
2. Does SeL4's formal verification mean that the system cannot benefit from "fuzzing"? Ignore non verified components. For this question, focus exclusively on the formally verified microkernel itself. Is there any benefit to fuzzing? If so, what kind of benefit might it provide?
3. Does SeL4's formal verification mean that it cannot enter a "weird" state as defined by Thomas Dullien in the weird machines paper? Provide sufficient technical details for your explanation. No matter which answer you select, you should explain how SeL4's formal verification does or does not impact the attacker's ability to turn the input to the system into a "program".
4. Return Oriented Programming (ROP) is a very literal example of an attacker converting input into a program as described in the Weird Machines paper. But, recall that the Weird Machines paper is much broader and encompasses worlds in which programs have perfect control flow integrity. Describe how an analogous type of attack programming for these types of programs (with perfect CFI) might be theoretically constructed. In other words, in ROP, we can construct an attack program out of "gadgets." What would we construct a program out of in the case of perfect CFI? What are the analogs?
5. In this semester, we have discussed various solutions to the problem of exploitation. Some focus on getting it right the first time (static checker, formal verification), some focus on detection (fuzzing), and some focus on mitigations (ASLR). We have discussed limitations for each of these. Pick any TWO solutions from ANY category and present a COMBINED approach that you think helps overcome an identified limitation. TWO WARNINGS. FIRST, this is the most open ended question in this section so it will be graded the harshest. If you're going to take it, make sure you

really understand the limitations and how you would overcome it. Don't use this question just because you don't understand the others. SECOND, this is the kind of thing that, perhaps, you might try to search for answers on the web. That is permitted. However, you must cite all of your sources if you use them and must put in enough of your own detail and explanation that it is clear you know what you're talking about.