14-741/18631: Homework 4

Due: Tuesday, March 30, 2021 (by 10:39am Eastern)

Name:

Andrew ID:

Total (100 pts max):

Guidelines (Please read before starting!)

- Be neat and concise in your explanations.
- You must use at most one page for your explanation for each Problem (code you wrote may be on additional pages). Start each problem on a new page. You will need to map the sections of your PDF to problems in Gradescope.
- To access the CTF problems, create the SSH proxy as per the guide on Canvas.
- For CTF problems, **you must use the following format in your explanation**:
  - CTF Username
  - Flag
  - Explain the vulnerability in the program, and explain conceptually how that vulnerability can be exploited to get the flag.
  - How did you exploit the vulnerability? List the steps taken and the reasoning behind each step. The TA grading should be able to replicate the exploit following the steps. Feel free to make references to your code! **Note that *"use XYZ online solver"* is not sufficient - you must explain how the online solver derived the answer for full credit.**
  - Append your source code in the same writeup. Your source code should be readable from the writeup PDF itself. Note that this does not count towards the page count above.

  Omitting any of the above sections would result in points being deducted.
- Some questions are marked as **Team work**. For those, you will be asked to come up with a joint write up/solution. Only one needs to put the solution in the write up, and the other simply write "*see [teammate andrewid]*". However, individual CTF username and flag still need to be put in the write up for CTF questions. This only applies to questions marked as "Team work".
- It is highly recommended that you use Python for your assignment. You may use other languages that you are familiar with, but the teaching team will not be able to support or debug language specific errors.
- Please check your English. You won't be penalized for using incorrect grammar, but you will get penalized if we can't understand what you are writing.
- Proofs (including mathematical proofs) get full credit. Statements without proof or argumentation get no credit.
- There is an old saying from one of my math teachers in college: "In math, anything partially right is totally wrong." While we are not as loathe to give partial credit, please check your derivations.

- Write a report using your favorite editor. Note that **only PDF submissions will be graded.**
- Submit to Gradescope a PDF file containing your explanations and your code files before 10:39am Eastern Standard Time on the due date. You can find the timing for EST here: `https://time.is/EST`. Late submissions incur penalties as described on the syllabus (first you use up grace credits, then you lose points).
- If you choose to use a late day, you do not have to inform the instructors. We will calculate the number of late days used at the end of the semester based on the time of submission on Gradescope.
- Post any clarifications or questions regarding this homework to Piazza.
- **Team work** As a team, beyond designated team questions, you are encouraged to shared resources (e.g., TA's help, online resources you found helpful); you are encourages to set up virtual study sessions with your teammate(s) to check each other's progress and discuss homework assignment solutions.
- **This is not a group assignment. Beyond your teammate, feel free to discuss the assignment in general terms with other people, but the answers must be your own.** Our academic integrity policy strictly follows the current INI Student Handbook `http://www.ini.cmu.edu/current_students/handbook/`, section IV-C.
- Good luck!

Exploiting a Buffer Overflow

You will use the CTF server from HW1 for this assignment!

Reverse Engineering Each of these program has a vulnerability similar to that described in AlephOne paper assigned in class. Please use the AlephOne paper as guideline, and go through the following steps:

1. Use `objdump` to dump the assembly of this program. You only need to focus on two functions (main and vuln) in the dump result.
2. Analyze the program using `gdb` and identify the assembly code that is/are vulnerable.
3. Draw a map of the stack with addresses and contents (such as return address, variables, ...) stored in the stack.
4. the stack address when the program is executed without `gdb` is 0x30 or 0x40 higher than the stack address when the program is executed with `gdb` because of environment variables used by `gdb`
5. use `Python` to compose the non-ASCII input of your program.

1 Team work: My first buffer overflow (20 points)

Overflow the stack and replace the return address with the address of the win method. Solve the Buffer Overflow problem on the 14741 CTF Server. Submit a writeup containing your CTF teamname and what steps you used to solve the problem, following the format given in the guidelines. *Make sure to cd to the folder containing the problem, or the program will seg fault when you win and it attempts to read the flag.*

You are expected to work with you teammate on this problem. In the submitted write up, include your own CTF username, but come up with a joint answer explaining how you derived your answer. One of the team member submit the answer in the write up, the other writes "*see [teammate's andrewid]*". Note that for all other questions in this homework assignment, you have to write your own answers.

2 Buffer Overflow 2 (20 points)

Now you will need to get shellcode to run. However, the stack is randomized! We will teach you a neat trick to jump straight to your buffer. Solve the Buffer Overflow 2 problem on the 14741 CTF Server. Submit a writeup containing your CTF teamname and what steps you used to solve the problem, following the format given in the guidelines. *Pay careful attention to the hint about "cat" or else you might get a shell but not be able to do anything because stdin is closed*

3 Buffer Overflow 3 (20 points)

We've added something like a stack canary. Can you figure out what the stack canary is? Solve the Buffer Overflow 3 problem on the 14741 CTF Server. Submit a writeup containing your CTF teamname and what steps you used to solve the problem, following the format given in the guidelines.

4 Buffer Overflow 4 (20 points)

The stack is no longer executable! You need to do a return to libc attack. Solve the Buffer Overflow 4 problem on the 14741 CTF Server. Submit a writeup containing your CTF teamname and what steps you used to solve the problem, following the format given in the guidelines. *Here's some python code that lets you interact with an executable. Put it in your home folder as buffer4.py, cd to the folder with the problem and do python ~/buffer4.py*

```
from pwn import *
p = process("./vuln")
print p.readline()
p.sendline("A"*190)
```

5 Buffer Overflow 5 (20 points)

The stack has been randomized again! This time, the author removed that one neat trick. But on a 32 bit machine, there aren't a lot of bits of randomness. Do **not** use the `jmp esp` gadget from buffer overflow 2. Submit a writeup containing your CTF teamname and what steps you used to solve the problem, following the format given in the guidelines.