Total points: 100

CMPT403 (2025 Fall)
Assignment 2

Programming assignment

1 Stack Overflow Exploit [40 points]

You have been given a binary file `sof`, which is compiled from `sof.c`, under the `sof` folder. Your goal is to provide proper input so that when `sof` executes, the `targe_function()` will be called, and eventually "Yes! You did it!" will be printed out.

Store the proper input in `sof.txt`, and the exact command I will use to mark your input is:

```
cat sof.txt |./sof
```

If the attack is successful, "Yes! You did it!" will be printed. Otherwise, "Sorry, you failed." will be printed.

2 Heap Overflow Exploit [30 points]

You have been given a binary file `hof`, which is compiled from `hof.c`, under the `hof` folder. Your goal is to provide proper input so that when `hof` executes, a general user will obtain the admin's permissions, and eventually "Congrats! You are now admin!" will be printed out.

Store the proper input in `hof.txt`, and the exact command I will use to mark your input is:

```
cat hof.txt |./hof
```

If the attack is successful, "Congrats! You are now admin!" will be printed. Otherwise, "Nah, regular user." will be printed.

3 Use After Free Exploit [30 points]

You have been given a binary file `uaf`, which is compiled from `uaf.c`, under the `uaf` folder. Your goal is to provide proper input so that when `uaf` executes, the `defaultUser` will obtain the admin's permissions, and eventually "Privilege escalation! defaultUser is now admin!" will be printed out.

Store the proper input in `uaf.txt`, and the exact command I will use to mark your input is:

```
cat uaf.txt |./uaf
```

If the attack is successful, “Privilege escalation! defaultUser is now admin!” will be printed. Otherwise, “Still a normal user.” will be printed.

Tips

You may need to store non-printable characters in the file to exploit the programs. An easy way to store non-printable characters in a file is to use Python. For example, the following command can save the non-printable characters `0x00 0x01 0x02 0x03` into `p.txt`:

```
python -c "print('\x00\x01\x02\x03')" > p.txt
```

All the binary files are executable on the CSIL computers, and these computers will be used when grading. So, it'd be better to test your answers on these computers before submission.

Please use the provided binary files to test your answers, as they might not work if you compile your own binary files using the source code.

Submission Instruction

Write a document, named `writeup.pdf`, to briefly explain how you exploit the programs provided, one paragraph for each program.

Put `writeup.pdf` together with `sof.txt`, `hof.txt`, and `uaf.txt` in a folder with name your @sfu.ca user name appended by ‘-hm2’. For example, my sfu username is `wujl`, and my folder name would be `wujl-hm2`. Compress the file and submit. Use the tar command to compress the folder. For example, `tar czvf wujl-hm2.tgz wujl-hm2`. Upload the `tgz` file in Canvas.

You can submit the `tgz` files any number of times and the system will accept the last submission for each file, which overwrites previous submissions. You are encouraged to make submissions as early as possible. Please remember to name your files correctly.

Keep in mind that plagiarism is a serious academic offense; you may discuss the assignment, but write your assignment alone and do not show anyone your answers and code.

(Important) LLM Usage. If you have used LLM (Large Language Model, such as ChatGPT) during the assignment, write a section titled with `LLM Usage` to describe how LLM is used in your assignment in `writeup.pdf`. If LLM is used while not stated in the pdf file, it will be considered as plagiarism.