Lab #2 : Reverse Engineering Practice

CSE4830 : Reverse Engineering

Reverse Engineering Lab #2

In class, we have discussed calling conventions, dynamic and static analysis of programs, and approaches for understanding and removing obfuscating code. Further, we have introduced several tools including a debugger, disassembler, and decompiler. Finally, we developed scripts to aid in these tasks.

Lab Assignment

For this lab, you will write a *CrackMe* challenge for your teammates to solve. A *CrackMe* is a compiled binary without source code that requests either a user password or key. You must write in the C programming language. Example output is displayed below.

```
$ ./crackme-1
Enter the key to continue >>> abc
<<< Fail
$ ./crackme-1
Enter the key to continue >>> def
<<< Success. Printing flag.txt
flag{The-art-of-simplicity-is-a-puzzle-of-complexity}
```

You may introduce anti-reversing code, complex equations, dead-code, and compile with obfuscation. You may consider using multiple different functions to parse the input into string and math operations. However, your binary must be solved by **at least one other student in the course** or you will be ineligible for the extra credit.

Deliverables

Deliverables include the source code to your problem and your report.

1. (*) Sept 29 by 8:00 PM EST: Your source code and a small textfile including how to solve it.
2. Oct 7 by 11:59 PM EST: Your report and solutions to the other student problems.

Extra Credit

1. (+5) Solve every other teammates' challenge.
2. (+20) Solve every other teammates' challenge first.