Option 1

You are tasked with creating C or C++ code that keeps track of some parameters. These parameters could be used for a game [gold, power, population, armour, abilities,etc] or perhaps a training tool [levels completed, accuracy over time, time spent, required courses, etc] - but the end use of the parameters is not our concern, we just want them to be protected from tampering.

Develop Phase [100]:

Part one of the develop phase [50]: Create a mechanism to securely store the initial defaults of at least 10 parameters within your program. These parameters must be of multiple data types (e.g ints, doubles, char* etc) and at least one parameter must be a complex data structure (array, linked list, etc). For this part you should consider encoding strategies of your data. You may also want to embed validation in case someone is able to figure out your encoding (e.g. redundancy checks to detect tampering?).

Part two of the develop phase [50]: Develop a mechanism in which if values are changed during the execution of your program, you store a supplemental file containing the values of your parameters. You do not need to develop a fully functional program (e.g. game/trainer/etc), but your program should launch and either prompt the user to change values, or after prompting automatically change some values. Your program should securely store values to this file and load from this file and be able to 1) detect tampering of the file and 2) if tampering is detected, overwrite it with the internally stored default values.

Attack Phase [50]:

Using any and all of the tools you have seen, dissect your project. Identify and attempt to change values, attempt to spoof your own software.

WriteUp/Submission [50]:

Document your development and design choices, include the dissection and reverse engineering of your program. Address the strengths and weaknesses of your program and how you would iterate and make your program more resilient to reverse engineering.

- You must use version control, include me,and:

  5 pts describe program:requirements, installation, usage [readme.md]

  10 pts documentation of design choices [design.md]

  10 pts documentation of your dissection and analysis [analysis.md]

  25 pts reflection of the strengths, weaknesses, and future changes [nop.md]

BONUS [25]: Using lessons from the attack phase create a second version of your program that further hardens it from reverse engineering.

Option 2

You have been asked to write a C or C++ code obfuscator - you may use any language you want. Your job is to take existing C or C++ code and obfuscate all variable values and data structures.

Develop Phase [100]:

Part one of the develop phase [50]: For this part you should consider encoding strategies of the data. You may also want to embed validation in case someone is able to figure out your encoding (e.g. redundancy checks to detect tampering?).

Part two of the develop phase [50]: Develop a mechanism that would obfuscate more complex data structures and possibly even logic structures. E.g. arrays, switch statements, etc.

Attack Phase [50]:

Using any and all of the tools you have seen in this class, dissect code that you have obfuscated using your software. Identify everything you can.

WriteUp/Submission [50]:

Document your development and design choices, include the dissection and reverse engineering of a sample program. Address the strengths and weaknesses of your software and how you would iterate and make your program more effective at obfuscation.

- You must use version control, include me,and:

  5 pts describe program:requirements, installation, usage [readme.md]

  10 pts documentation of design choices [design.md]

  10 pts documentation of your dissection and analysis [analysis.md]

  25 pts reflection of the strengths, weaknesses, and future changes [nop.md]

BONUS [25]: Using lessons from the attack phase create a second version of your software that better obfuscates code from reverse engineering.