Can YOU crack the code?

…then participate in the third NSA Codebreaker Challenge! Visit https://codebreaker.ltsnet.net for more details.

Codebreaker Challenge 3.0

DISCLAIMER: The following is a FICTITIOUS story meant for providing a realistic context for the Codebreaker Challenge and is not tied in any way to actual events.

Background

NSA has discovered that the leadership of a terrorist organization is using a new method of communicating secret messages to their operatives in the field. A copy of the program being used by one of the terrorists has been recovered along with an archive of text files that are believed to contain hidden messages. At first glance, the program appears to simply check stock information, but this is likely a ruse to make it appear innocuous. Your mission is to reverse engineer this software and develop capabilities to exploit the secret messaging component. There are 4 different tasks for you to complete, with each increasing in difficulty and building off the previous task(s).

Getting Started

To get started, visit https://codebreaker.ltsnet.net and click ‘Register’ to create an account. Be sure to use your university *.edu address during account creation so that your progress on the challenge can be tied back to your university. Once your account has been created, log-in to access the challenge problem materials on the ‘Challenge’ page.

To solve these challenges, you will need to analyze the executable file with low-level tools such as a disassembler, debugger, hex editor, Linux binutils, etc. We recommend downloading the IDA disassembler from https://www.hex-rays.com/products/ida/support/download_demo.shtml. IDA is a very powerful reverse-engineering tool and used by professional security researchers around the world. But it can be intimidating to novice users, so here are a few tips to get you started:

1. The strings present in executable files can provide substantial insight into the program functionality. To view the strings in IDA (after disassembling a file), select View -> Open Subviews -> Strings.
2. IDA creates useful code and data cross-references that make it easier to navigate the executable. For example, by double-clicking on a string, its location in the binary will be revealed. Clicking on the xref link next to the string will take you to the code that references the string. Using strings and xrefs will help you focus your reverse-engineering efforts on the relevant areas of code.
3. The "Hex View" in IDA, which can be accessed from either View -> Open Subviews -> Hex Dump or by clicking the Hex View tab when IDA first loads, is useful for identifying byte sequences that you may want to modify in order to change the behavior of the program. If you select an instruction in the main disassembly window and then switch to the Hex View, the bytes associated with that instruction will be highlighted.
4. To test a patch that you wish to apply, try running the program in debugger and making the modification in memory to see whether your patch works as expected. When you are ready to

make the patch permanent, a hex editor can be used to save your modifications to the program binary.

5. Remember that any value xor'd with itself is equal to zero. Examples:
   a. A xor A = 0
   b. if A xor B = C, then C xor A = B and C xor B = A
6. Base64 is a binary-to-text encoding scheme that represents binary data in an ASCII string format by translating it into a radix-64 representation. There are many commandline tools and free open-source libraries available that will base64 encode/decode data for you.
7. Steganography is the art of hidden writing. The goal is to be able to hide the mere existence of a secret message from third parties. This challenge problem incorporates the use of one steganographic technique to encode hidden message in otherwise innocuous text. For a nice introduction to steganography see http://www.garykessler.net/library/fsc_stego.html.

For more information about reverse-engineering, check out Lectures 1-5 on the following page: https://plus.google.com/113611061190453317839. Also, the Intel reference manuals may be downloaded from here: http://www.intel.com/content/www/us/en/processors/architectures-software-developer-manuals.html. These documents will help you understand the x86 instruction set and architectural details. Good luck!