COMPSCI 561: CTF Directions

Will Lillis

April 23, 2023

Basics:

Upon landing in the target machine, you will notice there are three files of interest in the `/home/labuser` directory. The first is a program binary, `flag`. The second is `flag.enc`. The file `flag.enc` contains the flag for this CTF, but it has been encrypted using the `flag` binary! There is also `decrypt.c`, which includes code to read in a user's text, perform some operation on it, and outputs it back to the screen. This mystery operation to decrypt the flag has been intentionally left out by the owner of the machine!

Goal:

Your goal is to decrypt the `flag.enc` file in order to capture the flag.

Available tools/commands:

There are a few commands/ tools at your disposal:

- `gdb`: A command line debugger. It works well for programs written in C!
- `objdump -d <binary-file>`: This command will disassemble `<binary-file>`, showing the assembly code that makes up the program. You may want to pay particular attention to `do_encrypt`. Additionally, adding the `-Mintel` flag may help with readability.
- `strings <binary-file>`: This command will display all strings contained within `<binary-file>` over a given length.
- `nano` and `vim`: There are a few text editors available on the system. Maybe they could be used to edit `decrypt.c` in order to decrypt the `flag.enc` file?
- `gcc <src-file.c> -o <binary-name>`: This command will take a file `src-file.c` containing code written in C and produce a runnable program binary `binary-name`.

Your Tasks:

1. The `flag` program prompts the user for an input, takes in said input, encrypts it, and prints it to the console. Figure out how this encryption works. This can be accomplished by running the program, or perhaps via other means of inspecting the binary.
2. Decrypt the `flag.enc` file. This can be done by hand or by writing a simple program to do the decryption for you. There is a `decrypt.c` file available on the machine that can be readily modified to help with this.