COMPSCI 561: CTF Directions

Will Lillis

April 24, 2023

Basics:

This CTF can be found at https://github.com/WillLillis/COMPSCI-561-CTF.

Begin by SSH-ing into the target machine as user `labuser`. The password is `labuser`. Upon landing in the target machine, you will notice there are four 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 its contents have been encrypted using the `flag` binary! There is also `decrypt.c`, which includes code to read in a user's text. The remaining portion of the code used to decrypt the flag has been intentionally left out by the owner of the machine! Finally, there is a C header file, `base64.h`. This includes some helpful functions that will perform base64 encoding/decoding.

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:

- `strings <binary-file>`: This command will display all strings contained within `<binary-file>` over a given length.
- `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.
- `nano`, `vim`, and `emacs`: 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 could be done by writing a simple program to do the decryption for you.