Introduction to reverse engineering

This activity consists in « cracking » a program where a password is required to get access. In this little crash course you will learn:

- The basic functionalities of Ghidra
- Familiarise yourself a little bit more with memory concepts
- Read and even write a little bit of assembly code
- Learn to make connections between the decompiler code and assembly code
- A few cracking techniques

Due to the nature of this activity, no source code will be provided. You will need to be creative, rigorous, and make use of your problem-solving skills to get to the bottom of these exercices (and of course, use google)

In the src folder, there is a program called « AnswerTo42 ». There are three exercices to complete, although you are not forced to do them all.
I would also like to introduce the concept of evals in this activity. You will not get any eval points, but maybe we can negotiate a cookie. Here is how it will go:

- Create a new folder « atelier_eval ». In this folder, you will create your ex00/ex01(and so on) folders.
- Evals will be done on the evaluated student computer
- Evaluated student should be able to explain to his evaluator his process to complete each exercice, Evaluators should receive clear and concise explanations.
- If a program is there, the evaluated needs to launch his program and show that it has been properly patched, and open that same program in Ghidra and show where modifications has been made.
- Only 1 eval is necessary but feel free to do more if you want. If you want a moulinette, me(llaplant)will play the role.

Ex00:

You will need to find the correct password embedded in the program, and put the correct answer in the following file.

- File to push: password.txt

Ex01:

You will need to change the password to your login, followed by 42. (Ex: llaplant42).

- File to push: AnswerTo42_patched1

Ex02:

You will need to patch the program so it lets you enter it, no matter the password.

- Fichier a remettre: AnswerTo42_cracked

Partie 2

This part is a « bonus » part. In the src folder, there is a program called « FallingWhale ». Compared to the previous program, this one is rather difficult. If you decide to give it a shot, you need to provide these files at once:

- A file containing a clear and concise documentation on how the program works.
- A « keygen » written in C(has to be written to the norme, except for the 25 line cap and 5 functions cap) that can generate a valid key for the original program.(Be creative on this one !)

Note: It is possible that Ghidra will export your patched programs with the extension .bin. Don’t panic, you can safely remove the extension and the program will still be usable inside the terminal(MacOS will also launch it even if the extension stays)

Happy cracking !