UGA **G-INP**
UFR IM2AG **Ensimag**

Master Cybersecurity

Code (de-)Obfuscation

Before Starting , download from the Moodle all the necessary files...

Note that ***a short report*** is expected for Part 2 …

Part 1 - Self-modifying code

Exercise 1

- Read, compile, and execute the file foo-add.c
- Modify it such that the 2nd call to function foo prints i-- (instead of i++)

Exercise 2

complete the file foo-shell.c such that the 2nd call to function foo executes the shell code (given in the file).

Indication: you just need to copy the shell code at foo() address !

Part 2 - Code Obfuscation with Tigress

Main resources: https://tigress.wtf/ and https://ligerlabs.org/lectures.html (part on "software protection").

Start first by installing tigress and running a demo example as explained here.

Exercise 3

1. Write a first (very simple) C code from your own, e.g., :

```
int main (){
        int x=42;
        return x++;
}
```

2. Using tigress, obfuscate it using the **Virtualize** transformation

Both by reading the source-level result produced by tigress (result.c) and running (possibly under gdb) the corresponding executable (a.out), explain **precisely** the content of the obfuscated code (bytecode, dispatcher, etc.).

Try some variants/extensions of this transformation to see how the obfuscated code is modified.

3. Write a second example with some non trivial control-flow graph (i.e., with one loop and a few conditional statements inside and outside the loop). Proceed as for the previous question with this new example using the transformations **AddOpaque** and **Flatten**

4. Finally, try some other kinds of transformations examples like **Data Transformations** (https://tigress.wtf/data-transforms.html) or **Self Modify** transformation or etc.

Part 3 - A guided CTF solving

1. Solve the proposed CTF (located in the `CTF/ctf1` directory) following this complete tutorial.
The goal is **clearly not** not to copy/paste each proposed command, but to understand their purpose and the corresponding lightweight (yet practical) obfuscation technique they address. On the way you will also see how to reverse a code dynamically simply using gdb ...

2. Once this flag has been (fairly :-) obtained, you can try the next challenge (`CTF/ctf2`) …
For this second challenge, on the Ensimag machine, you may have to update your LD_LIBRARY_PATH environment variable ...