Ensimag **UGA**
Master CybserSecurity & Master MoSiG **Year 2016-17**

Software Security & Secure Programming

Lab session 1 : security weaknesses in C

Before you start :

Copy the content of directory `~mounlaur/SoftwareSecu/LabSession_1` into a directory of your choice . . .

You should write a **report** on the work you did during this lab session and give it back during the class on **tuesday 25/10**.

Exercise 1

Compile the program `exercise_1.c` :

```
gcc -o exercise_1 exercise_1.c
```

Execute this program (without arguments) :

```
./exercise_1
```

You should get a crash, explain why.

How can you correct this problem (in order to get an error message instead of a crash) ?

Exercise 2

Look at the source code of the C program `exercise_1.c`. This program takes as input two integer arguments on the command line (`argv[1]` and `argv[2]`).

Compile this programm with gcc :

```
gcc -o exercise_2 exercise_2.c
```

Execute it with some random arguments :

```
./exercise_2 5 10
./exercise_2 2 17
   etc.
```

This program may leed to several possible results :

- print "You loose"
- infinite loop
- crash
- etc.

Explain each different result you get, drawing the execution stack.

Find the program input allowing to print "You win"!

Disassemble this program using the objdump command :

```
objdump -S exercise_2
```

Look at the assembly code of functions <main>. Try to understand this code, and to retrieve the offsets in the stack of the local variables.

Indication : in this 64-bits architecture registers ebp (frame pointer) and esp (stack pointer) are called rbp end rsp ...

Exercise 3

Look at the source code of the C program exercise_3.c. This program takes as input one directory name and prints its content (like the ls command).

Compile this program with gcc :

```
gcc -o exercise_3 exercise_3.c
```

Run it :

```
./exercise_3 /tmp
```

If the argument string is too long, then an error message is printed and the user is requested to enter a character string.

Explain why this program is *vulnerable*.

Find how you can use this program to execute any shell command of your choice (e.g, /bin/sh, xcalc, etc.)