* We assume you do the exercises on a raspberry pi (arm 32bit)

This directory contains the first exploitation exercises, focusing on memory corruption with minimal exploit mitigation.

A level is passed when the program behaves as required in the instructions, without being debugged (i.e. outside gdb) .

Unless otherwise instructed, you need to disable ASLR (Address space layout randomization) on the Pi with the following command:

echo 0 | sudo tee /proc/sys/kernel/randomize\_va\_space

Or execute 'aslr.sh' located in this directory with the parameter 0

Note that even if ASLR is enabled, programs debugged under GDB may not have their address space randomized.

Make sure to disable ASLR each time you start or return to an exercise, as it resets on reboot and sometimes on its own after a while.

perform your exercises in the following order:

1. stack

2. shellcode

3. heap

4. shellcode

If you wish to assemble a shellcode you may use the script 'sc.sh' (on the Raspberry Pi) located in the tools directory.

An example assembly file 'sc.as' is also provided.

If you only need to assemble a few instructions you can also use this tool, or an online assembler.

If you wish to test your shellcode, you can use the provided "shellcode tester"

Good luck!