

**IE 4012**

**OHTS**

**4th Year, 1st Semester**

Lab Report 01

# IO.netgarage Levels

Submitted to

Sri Lanka Institute of Information Technology

In partial fulfillment of the requirements for the

Bachelor of Science Special Honors Degree in Information Technology

03/02/2020

## **Declaration**

I certify that this report does not incorporate without acknowledgement, any material previously submitted for a degree or diploma in any university, and to the best of my knowledge and belief it does not contain any material previously published or written by another person, except where due reference is made in text.

Registration Number: IT17114172

Name: A.U. Sudugala

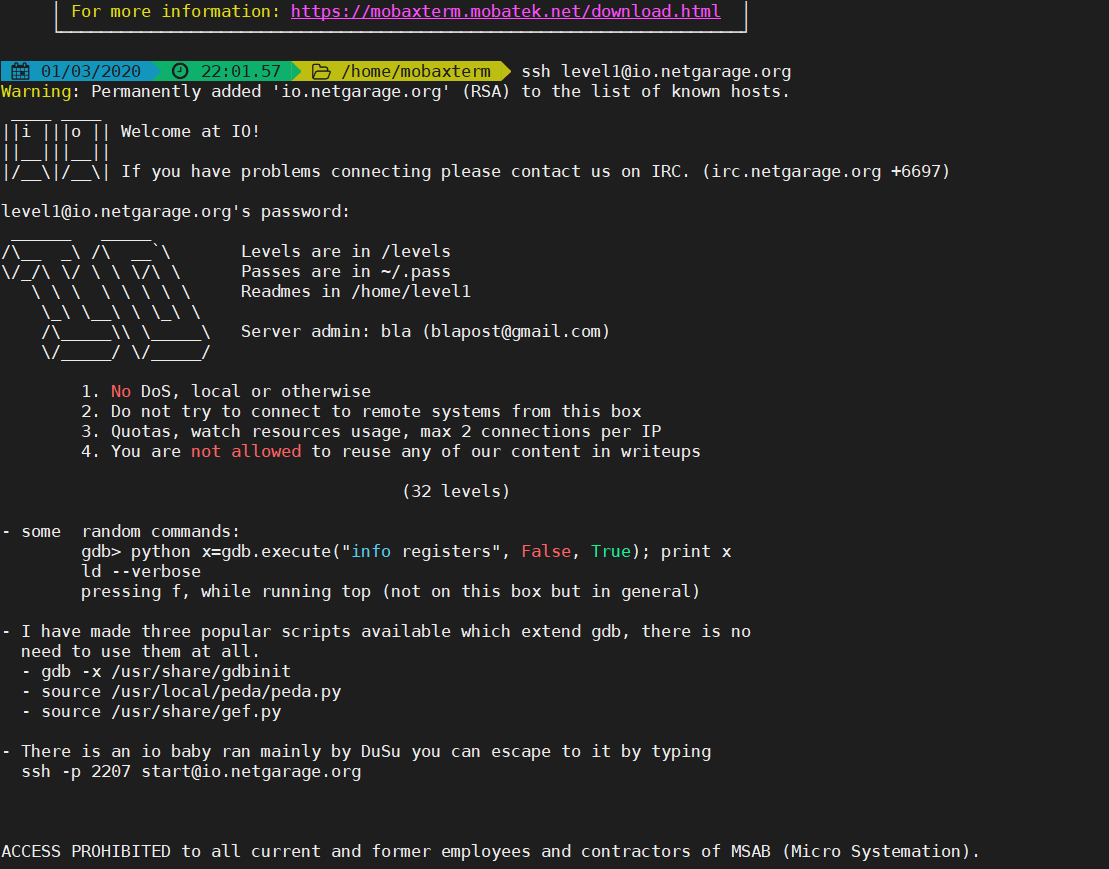
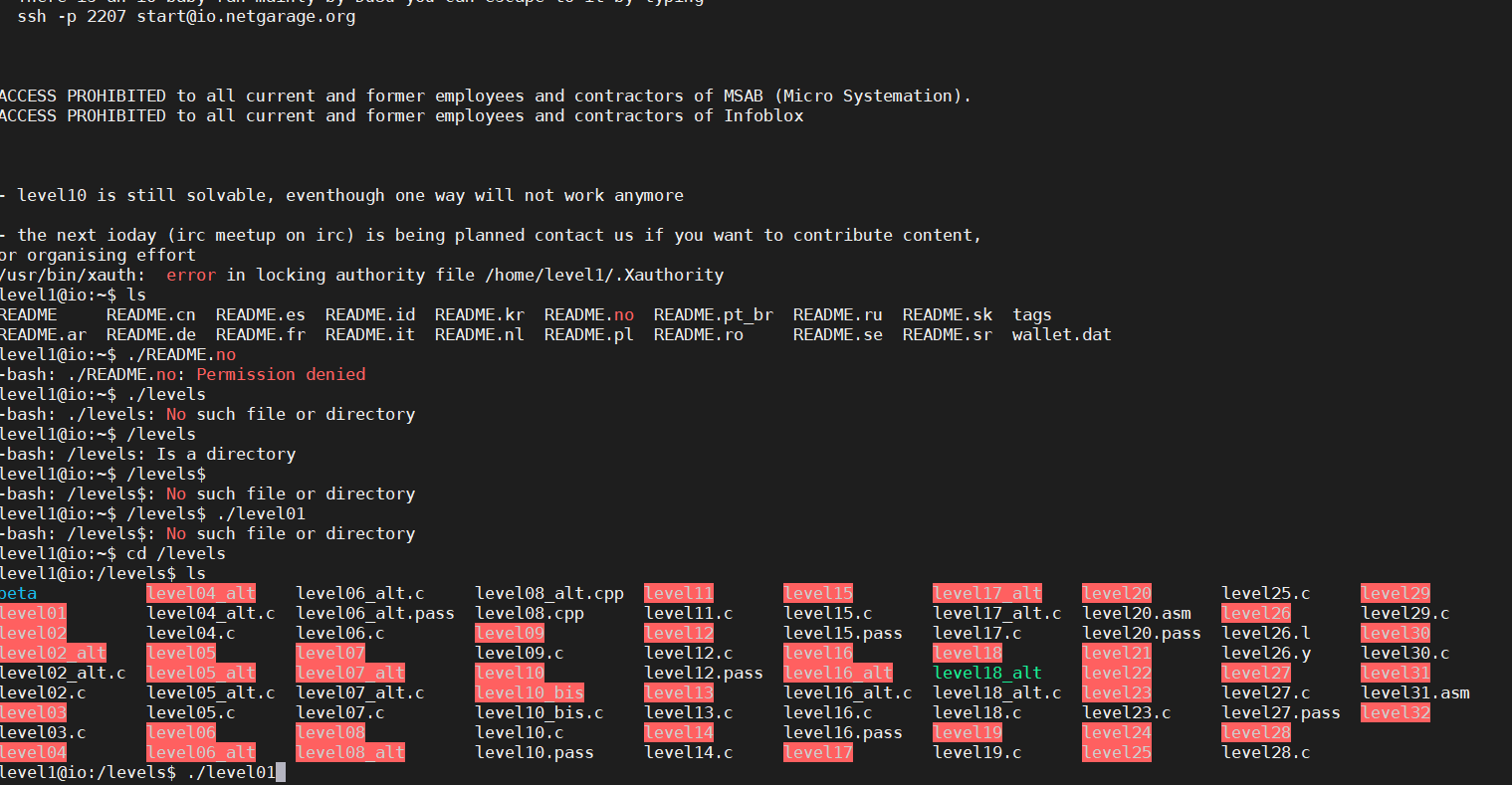
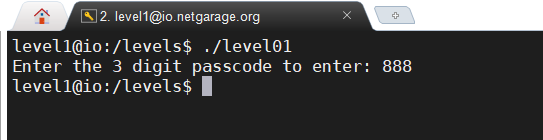
Level 01.

Figure 1: level 1 beginner phase

|  |  |
| --- | --- |
| TITLE | LINK |
| Level 1 | ssh level1@io.netgarage.org |

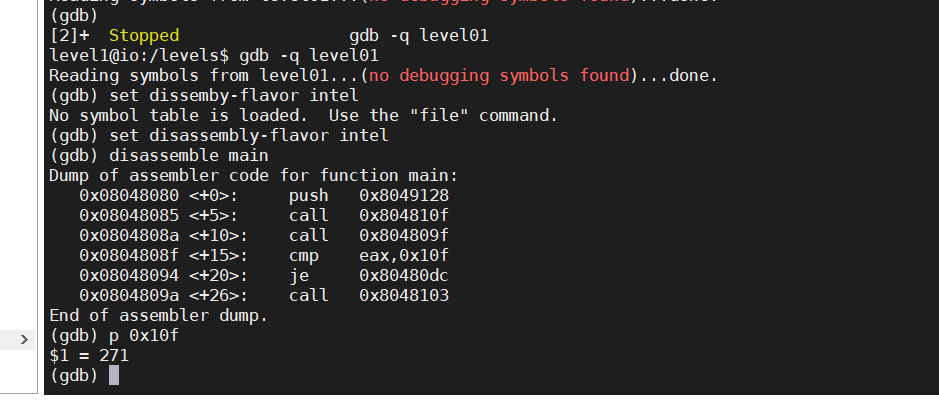
Checking the levels.

In order to proceed further, we must enter 3 digits passcode. Since, we don’t know the exact passcode lets assume that it is 888.



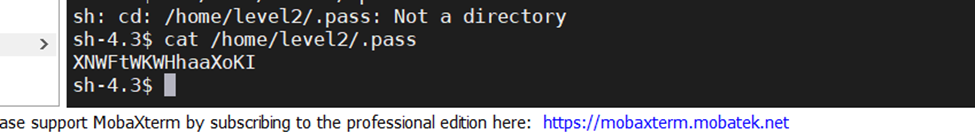
Launching the program under gdb.

Here we can see the disassembly code for the main function.



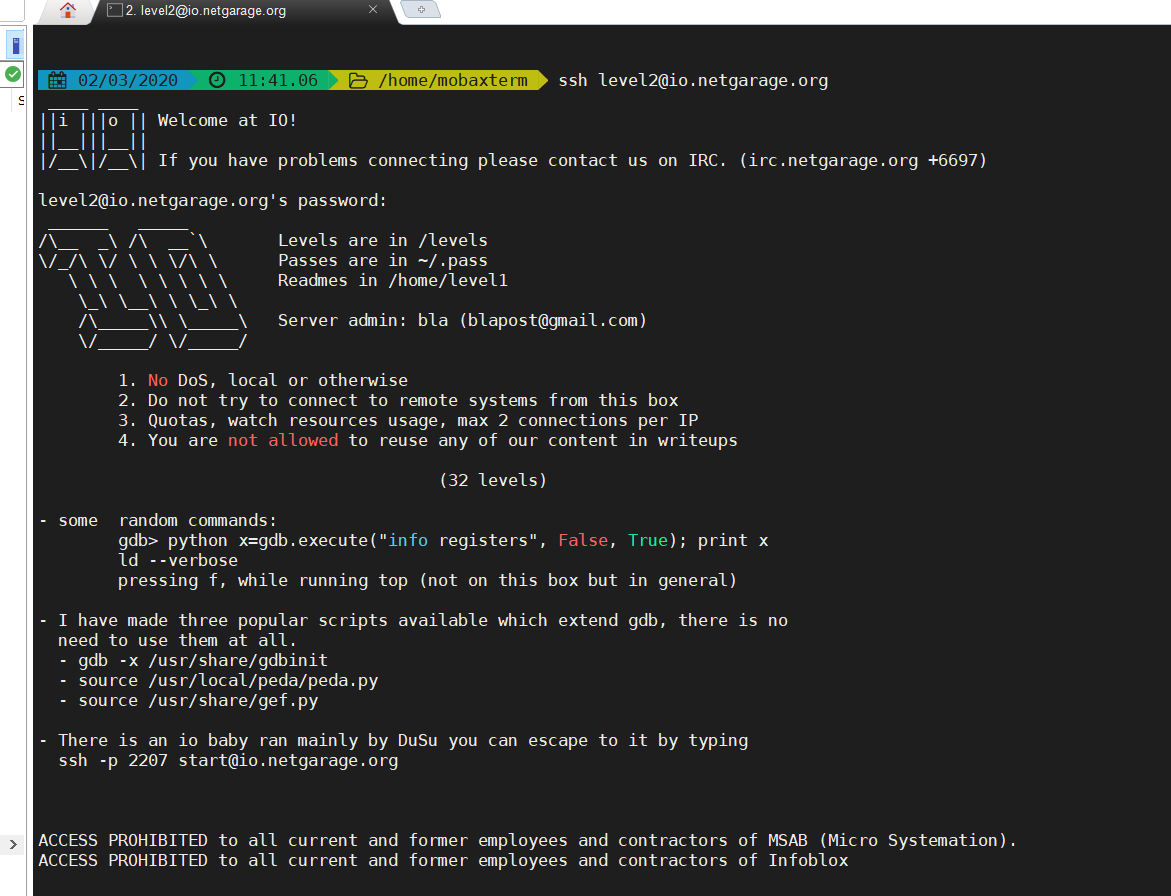
The program compares a fixed value with the value of the register. This value is a hexadecimal value, we can display its decimal value with p in gdb.

So, the passcode is 271

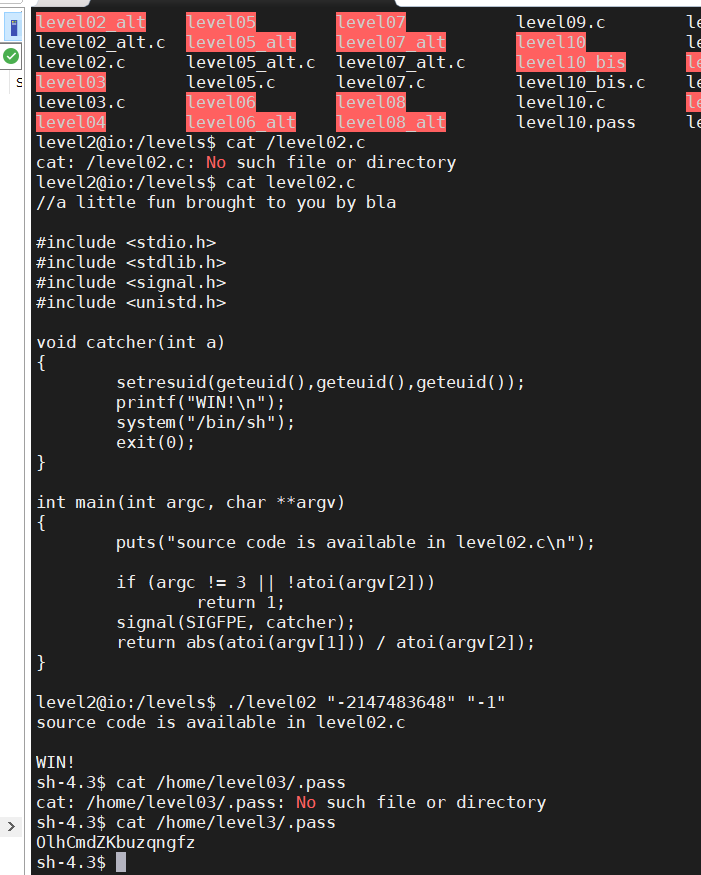


The password for level2 is XNWFtWKWHhaaXoKI

Level 2



|  |  |
| --- | --- |
| TITTLE | LINK |
| Level 2 | ssh level2@io.netgarage.org |



Opening the level02.c file we can see:

* The number of args must be 2, (argv[0] being the caller's name).
* The two arguments should be numbers
* The catcher function will be called on the event SIGFPE (launched for example for a division by 0)
* The return value of the function is argv[1]/argv[2]

If the catcher function is called, it will set the current user identity, and print a win message, before spawning a new shell. This is clearly the indication that we need to raise a SIGFPE exception.

The SIGFPE can be triggered with a 1/0 or a sqrt(-1) for example, in our case, neither can be used.

What we can do instead is try to use an integer value outside of the bound of the integer definition. We can see on the abs reference page that the most-negative value to be out of range is -2147483648, because this will convert to 2147483648, above the MAX\_INT value. So, if we send to abs the value - 2147483648, the result will be also -2147483648 (because of binary max bound and negative values).

But, if we send the value 2147483648 to abs, which is an incorrect value, it should raise an error:

$ ./level02 "-2147483648" "-1"

source code is available in level02.c

Level 3 password: OlhCmdZKbuzqngfz