IE4012
Offensive Hacking Tactical and Strategic
4th Year, 1st Semester

Lab Report 1

Netgarage CTF Walkthrough

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

2020.03.02

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: IT17078306

Name: Jayawardhane H. N

Table of Contents

Table of Figures

1. Introduction

1.1. What is a CTF

CTF stands for the term Capture the Flag. This is a kind of competition where participants will solve different sorts of security problems, defend and capture computer systems. CTFs will further help in carving the skills of professionals and aspiring individuals in the field of cyber security.

1.2. Types of CTFs

1.2.1. Jeopardy CTF

Participants will be facing a number of tasks involving crypto, forensic, web, reverse engineering etc. In this case the player will have to unlock a particular level by capturing the respective flag which could be a fragment of code or a file of that particular level by finding the answer to the clues presented.

1.2.2. Attack – Defense

Each team will be handed over a system with known security vulnerabilities and they will have to patch their system while creating exploits for the opposition's system. In the beginning of the game the teams will be using exploits to steal flags of the opposition and protect themselves.

1.2.3. Mixed

The combination of both jeopardy and attack – defense will be a "Mixed" one

2. Netgarage

2.1. Introduction

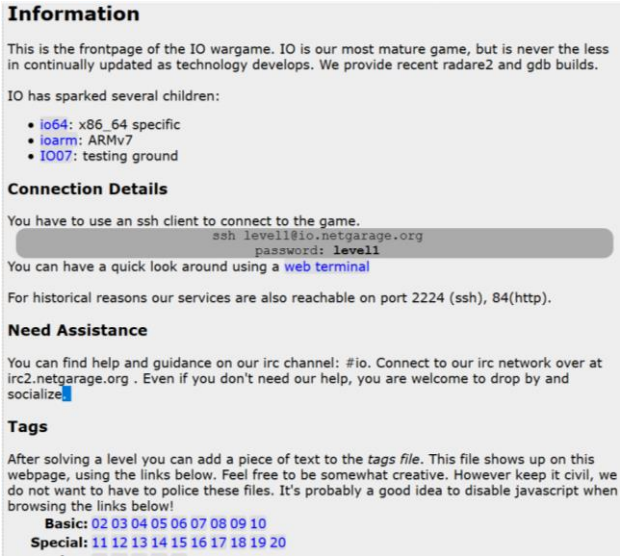
Netgarage is an online wargame where the players can establish a secure shell (ssh) connection to the game server and capture the password or flag for the next level and unlock the levels, one by one.

2.2. Walkthrough

This report will provide a walkthrough from level 1 to 3.

2.2.1. Level 1 to Level 2

When the site http://io.netgarage.org/ is accessed the page will give the password for level1 itself as level1. By using PuTTY the user can create a ssh connection.



Information

This is the frontpage of the IO wargame. IO is our most mature game, but is never the less in continually updated as technology develops. We provide recent radare2 and gdb builds.

IO has sparked several children:

- io64: x86_64 specific
- ioarm: ARMv7
- IO07: testing ground

Connection Details

You have to use an ssh client to connect to the game.

```
ssh level1@io.netgarage.org
password: level1
```

You can have a quick look around using a web terminal

For historical reasons our services are also reachable on port 2224 (ssh), 84(http).

Need Assistance

You can find help and guidance on our irc channel: #io. Connect to our irc network over at irc2.netgarage.org . Even if you don't need our help, you are welcome to drop by and socialize.

Tags

After solving a level you can add a piece of text to the *tags file*. This file shows up on this webpage, using the links below. Feel free to be somewhat creative. However keep it civil, we do not want to have to police these files. It's probably a good idea to disable javascript when browsing the links below!

Basic: 02 03 04 05 06 07 08 09 10

Special: 11 12 13 14 15 16 17 18 19 20

Figure 2.1: ssh connection details and password for level1

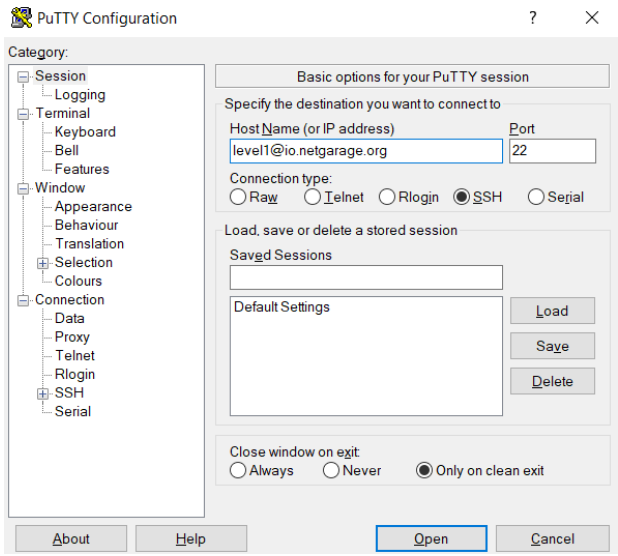


Figure 2.2: Establishing ssh connection using PuTTY

By entering the given credentials, I was able to succesfully access level1.

Once level1 was accessed. By using the cd / command I was able to get into the current directory and then by giving the ls command I was able to get a list of all files in the directory. In there as we can see a .c file doesn't exist for level1 therefore I assumed that level01 was the file that might lead to the password for level2.

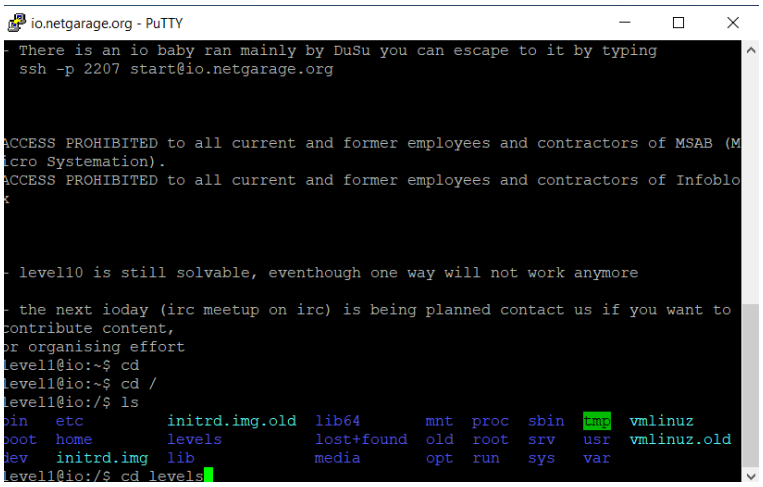


Figure 2.3: List of directories

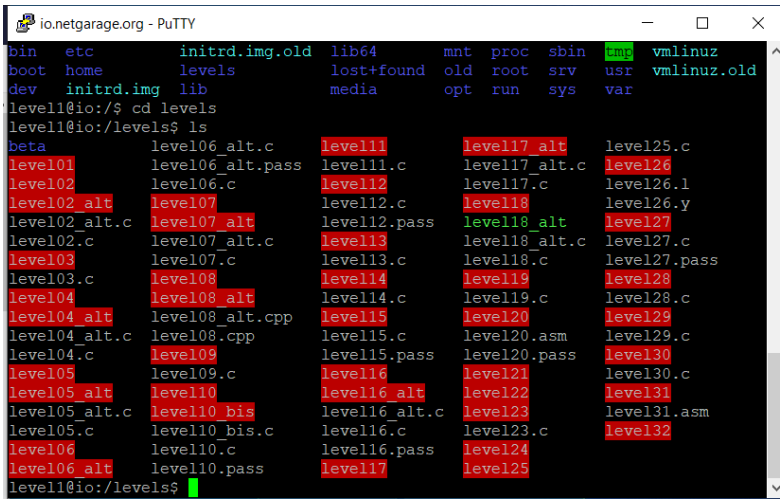


Figure 2.4: List of files

Once I accessed the level01 file, then I used gdb to debug the level01 file in hopes of finding the assembly code which might lead to the password.

For that I first used the *gdb level01* command and then once debugging had started then I used the *set disassembly intel* command to use intel style to debug as I am using Windows for this purpose. I then used the *disas main* command to disassemble the main function.

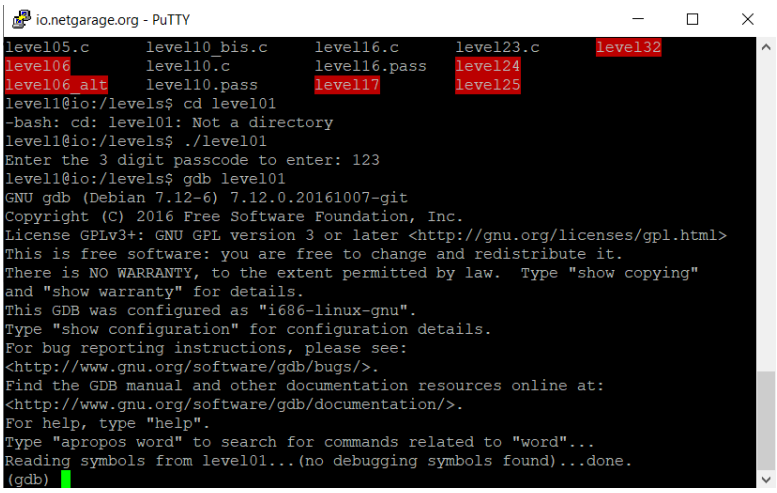


Figure 2.5: Disassembling using gdb

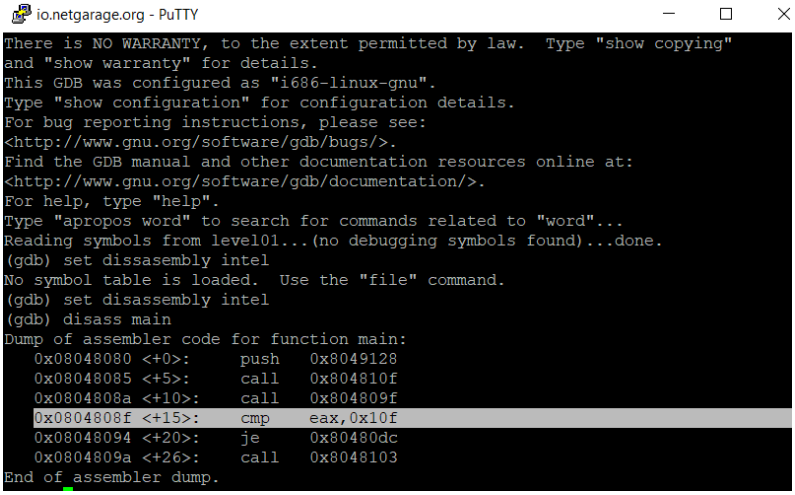


Figure 2.6: Finding memory location

As the disassembled code shows I compared the input of 0x10f with the conditional branch address 0x80480dc, 0x0f was found to be 271 by using the *p 0x10f* command. And then when the level01 file could be accessed by giving the 3-digit password as 271.

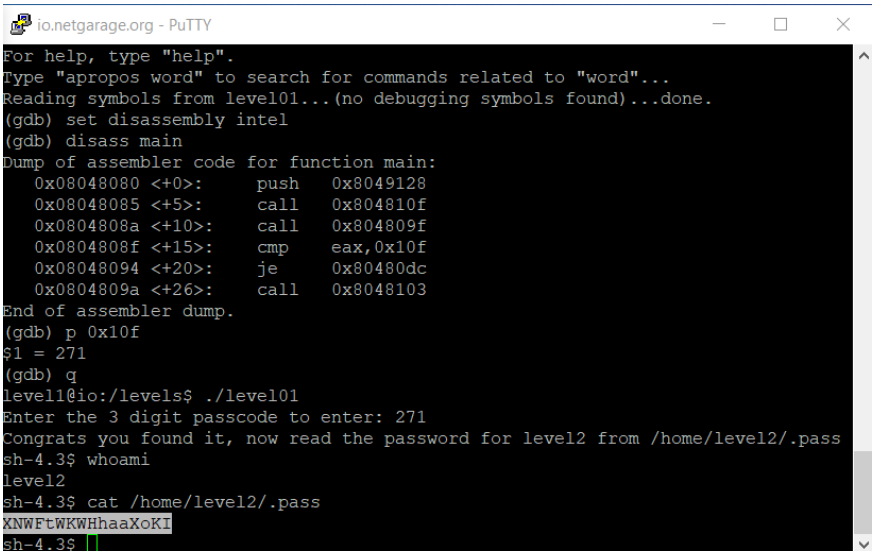


Figure 2.7: Arriving at the password for level 2

For further justification I then typed the *whoami* command to see if I had gained entrance to level 2. Then once it showed level2 I then used the command *cat /home/level2/.pass* to get to the password for level2.

2.2.2. Level 2 to Level 3

Once the password found from the previous section was given when establishing the ssh connection to level 2, access was granted.

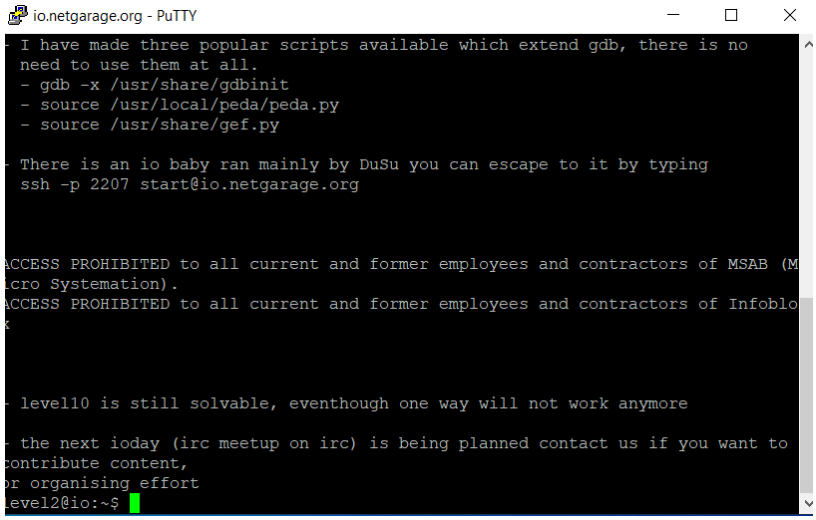


Figure 2.8: Accessing level 2

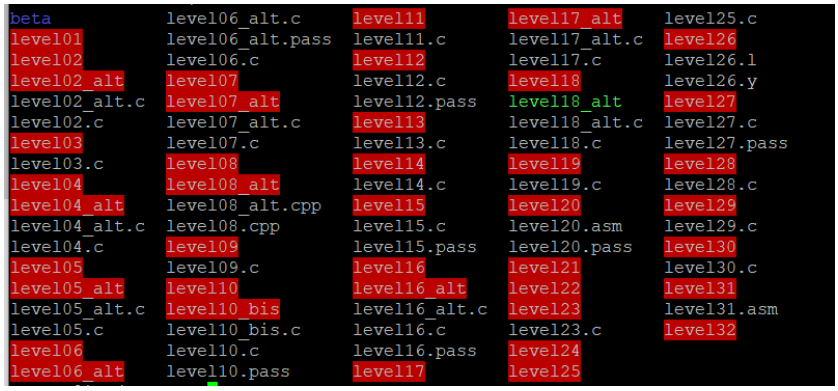


Figure 2.9: All files list in level 2 directory

As figure shows there exists a level02.c file. Once this file was opened the following code was found.

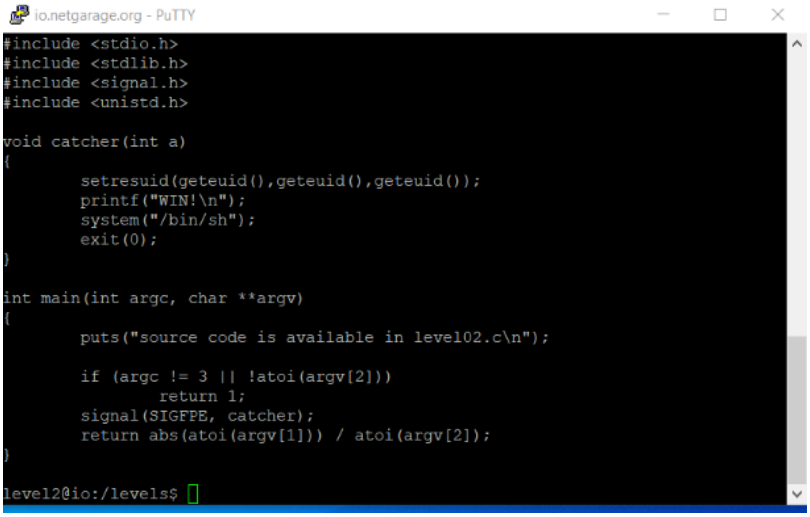


Figure 2.10: level02.c source code

As shown in the figure itself;

1. The number of arguments should be 2
2. The arguments should be numbers
3. Catcher function will be called on the SIGFPE event
4. Return value of the argument will be argv[1] and/or argv[2]

Once the catcher function is called. It will then set the current user id and print win message and the SIGFPE can be then triggered with a 1, 0 or a sqrt (-1). In this case, neither of this can be used.

Therefore, I added a integer value outside of the integer definition bound. And in the abs reference page it shows that -2147483648 is the most negative value out of bounds of int. in this case I then sent “-2147483648” and “-1” as argv[1] and argv[2] respectively.

Figure 2.11: Arriving at password for level 3

By using the *cat/ home/ level3/.pass* command I was able to then gain the password to level 3 as shown in figure.