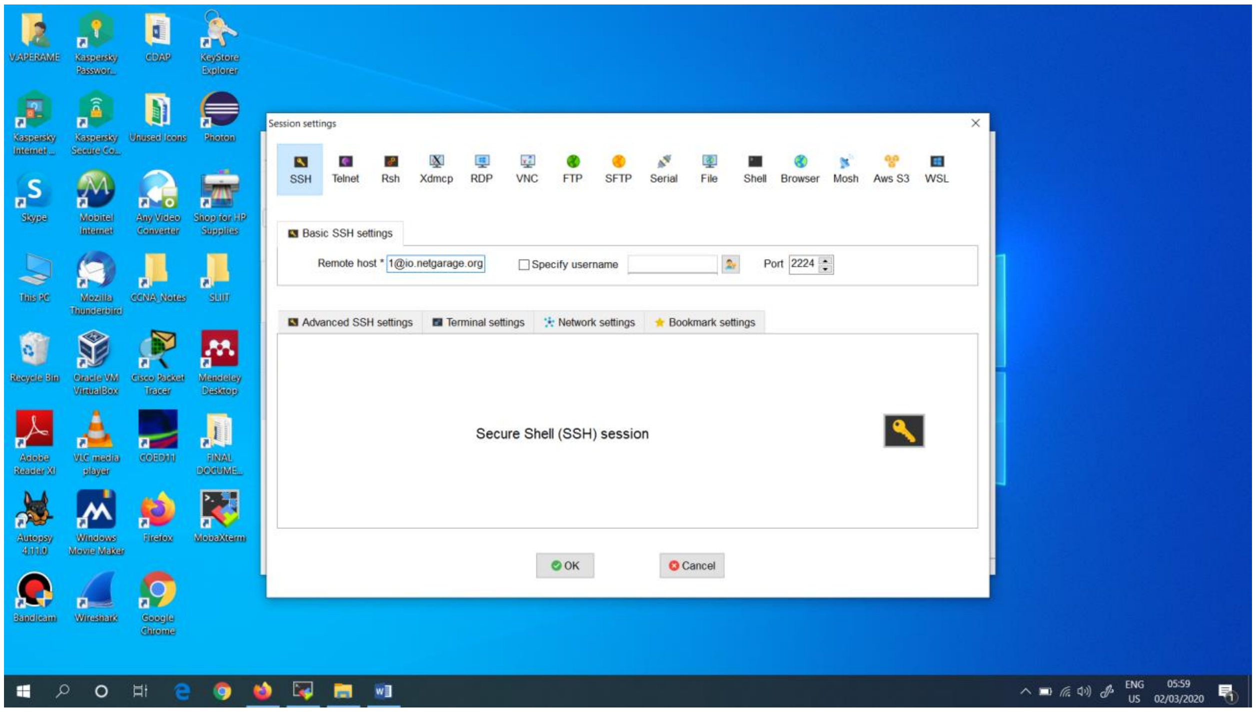
**IT17157674 - V.Aperame**

# OHTS Lab 1, Level 1

At first most, I opened the "MobaXterm" terminal and typed the Remote host as level1@io.netgarage.org and typed the port as 2224. This is shown in the following screen shot (Figure 1).



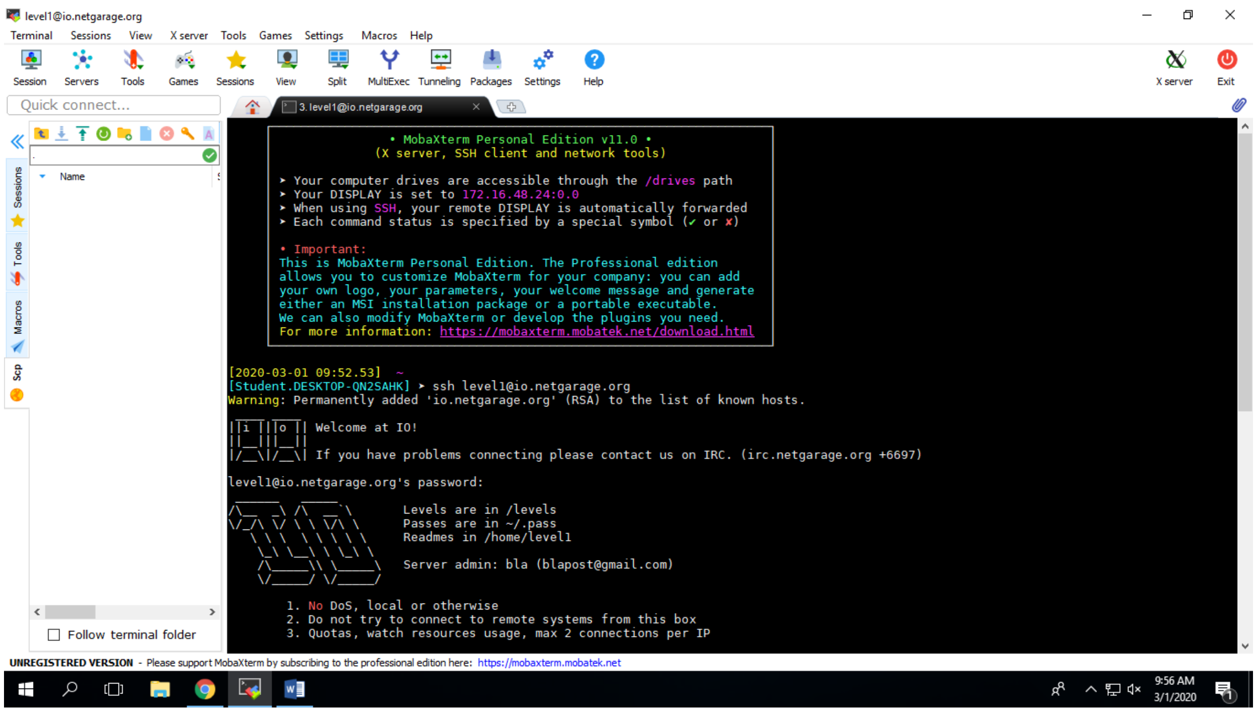
Remote host * 1@io.netgarage.org
☐ Specify username
Port 2224

*Figure 1: Typed the Remote host and Port*

Then, I opened the "MobaXterm" and got a new terminal. Then, typed the username and password. This is shown in the following screen shot (Figure 2).

Username – level1@io.netgarage.org

Password – **level1**

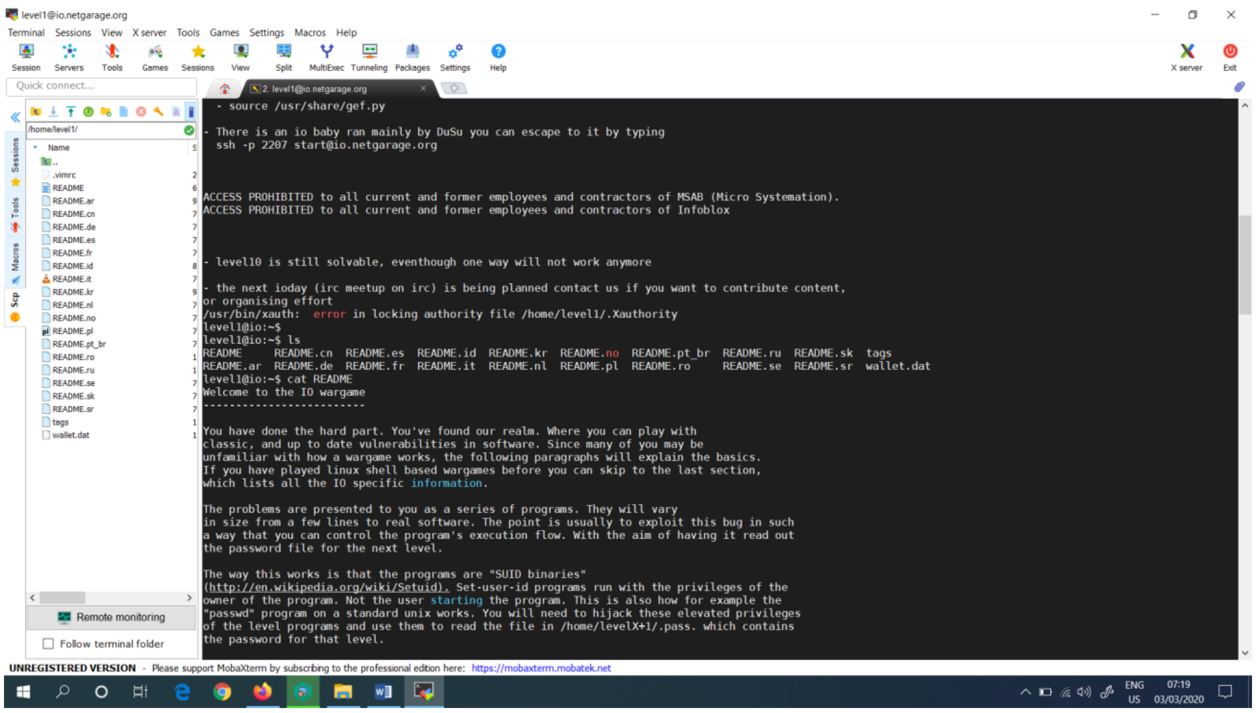


```
  • MobaXterm Personal Edition v11.0 •
(X server, SSH client and network tools)

➤ Your computer drives are accessible through the /drives path
➤ Your DISPLAY is set to 172.16.48.24:0.0
➤ When using SSH, your remote DISPLAY is automatically forwarded
➤ Each command status is specified by a special symbol (✔ or ✘)

• Important:
This is MobaXterm Personal Edition. The Professional edition
allows you to customize MobaXterm for your company: you can add
your own logo, your parameters, your welcome message and generate
either an MSI installation package or a portable executable.
We can also modify MobaXterm or develop the plugins you need.
For more information: https://mobaxterm.mobatek.net/download.html

[2020-03-01 09:52.53]  ~
[Student.DESKTOP-QN2SAHK] ➤ ssh level1@io.netgarage.org
Warning: Permanently added 'io.netgarage.org' (RSA) to the list of known hosts.

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

level1@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
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

*Figure 2: Typed Username and Password in order to login*

In order to go to the level 2, need to do the following shell commands. They are as follows. First, in order to get what are the list of directory contents the command "**ls**" is used. The Figure 3 shows what are the lists of files found.

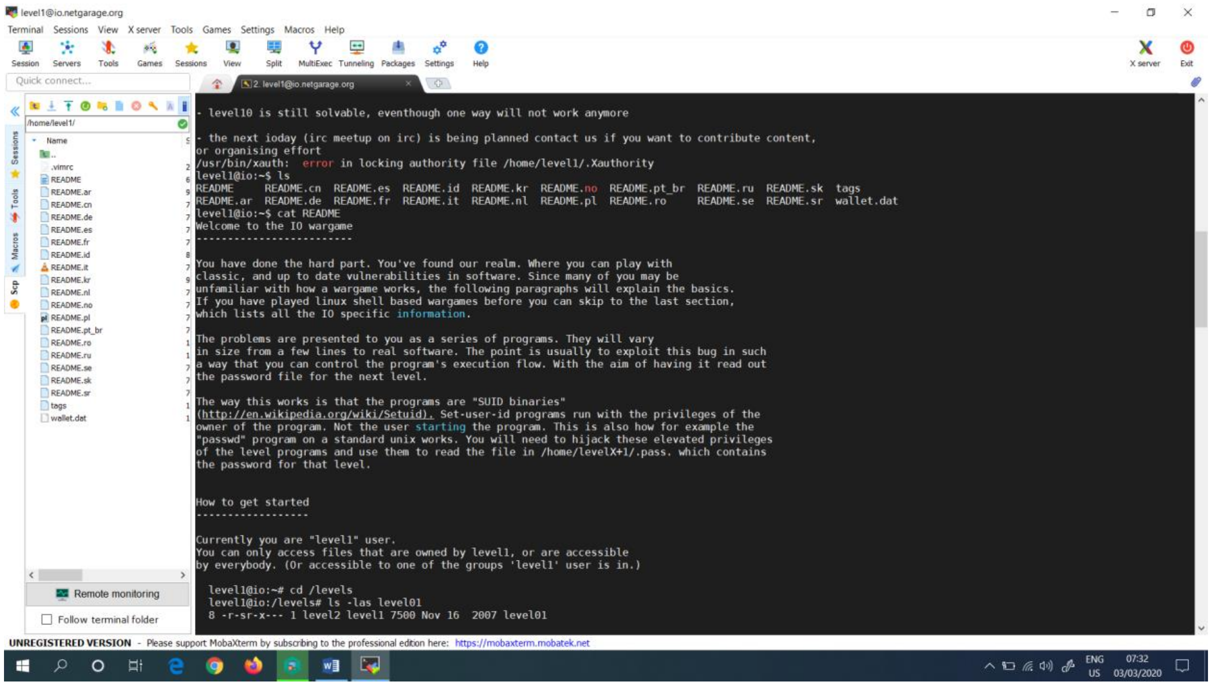


```
  - source /usr/share/gef.py

- There is an io baby ran mainly by DuSu you can escape to it by typing
  ssh -p 2207 start@io.netgarage.org



ACCESS PROHIBITED to all current and former employees and contractors of MSAB (Micro Systemation).
ACCESS PROHIBITED to all current and former employees and contractors of Infoblox



- level10 is still solvable, eventhough one way will not work anymore

- the next ioday (irc meetup on irc) is being planned contact us if you want to contribute content,
or organising effort
/usr/bin/xauth:  error in locking authority file /home/level1/.Xauthority
level1@io:~$
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"
(http://en.wikipedia.org/wiki/Setuid). Set-user-id programs run with the privileges of the
owner of the program. Not the user starting the program. This is also how for example the
"passwd" program on a standard unix works. You will need to hijack these elevated privileges
of the level programs and use them to read the file in /home/levelX+1/.pass. which contains
the password for that level.
```

UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

*Figure 3: Typed "ls" command*

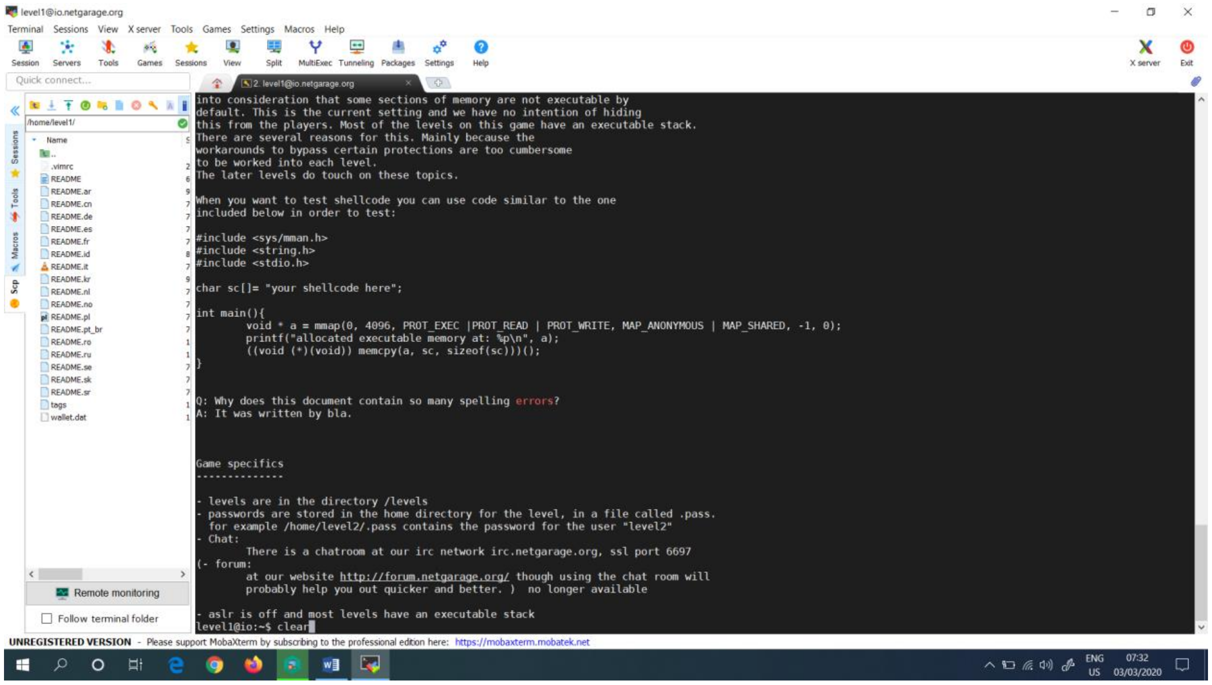
Then, for the purpose to create single or multiple files, to view the contain of file, and to concatenate files and to redirect the output in the terminal the "**cat**" command is used. So, then I used the "**cat**" command. The Figure 4 shows the typed "**cat"** command.



```
- level10 is still solvable, eventhough one way will not work anymore

- the next ioday (irc meetup on irc) is being planned contact us if you want to contribute content,
or organising effort
/usr/bin/xauth:  error in locking authority file /home/level1/.Xauthority
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"
(http://en.wikipedia.org/wiki/Setuid). Set-user-id programs run with the privileges of the
owner of the program. Not the user starting the program. This is also how for example the
"passwd" program on a standard unix works. You will need to hijack these elevated privileges
of the level programs and use them to read the file in /home/levelX+1/.pass. which contains
the password for that level.


How to get started
------------------

Currently you are "level1" user.
You can only access files that are owned by level1, or are accessible
by everybody. (Or accessible to one of the groups 'level1' user is in.)

  level1@io:~# cd /levels
  level1@io:/levels# ls -las level01
  8 -r-sr-x--- 1 level2 level1 7500 Nov 16  2007 level01
```

*Figure 4: Typed "cat" command*

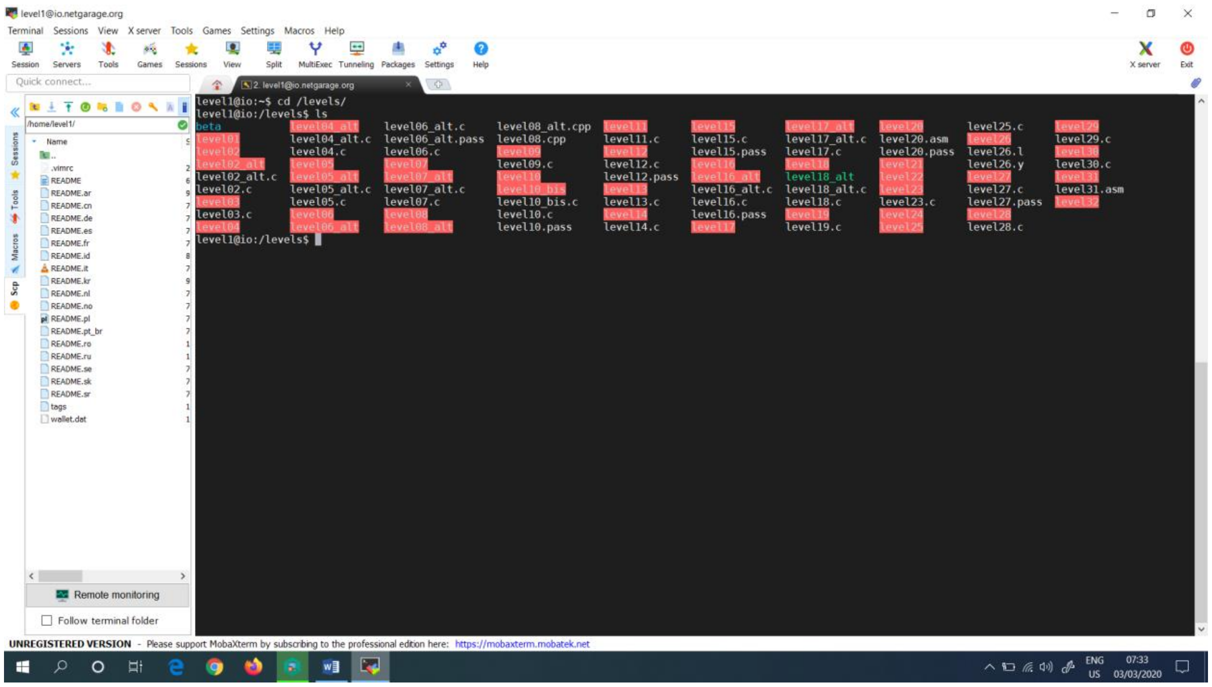
After all the commands came, then at the end of the shell terminal, I typed the **"clear"** command in order to clear the screen. With this command, it helps to view only the needed commands which need to appear in the shell. The following Figure 5 shows the typed **"clear"** command.



```
into consideration that some sections of memory are not executable by
default. This is the current setting and we have no intention of hiding
this from the players. Most of the levels on this game have an executable stack.
There are several reasons for this. Mainly because the
workarounds to bypass certain protections are too cumbersome
to be worked into each level.
The later levels do touch on these topics.

When you want to test shellcode you can use code similar to the one
included below in order to test:

#include <sys/mman.h>
#include <string.h>
#include <stdio.h>

char sc[]= "your shellcode here";

int main(){
        void * a = mmap(0, 4096, PROT_EXEC |PROT_READ | PROT_WRITE, MAP_ANONYMOUS | MAP_SHARED, -1, 0);
        printf("allocated executable memory at: %p\n", a);
        ((void (*)(void)) memcpy(a, sc, sizeof(sc)))();
}


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/level2/.pass contains the password for the user "level2"
- 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:~$ clear
```

*Figure 5: Typed "clear" command*

Then, as the next step, in order to change the directory, the command **"cd"** is used. After changing the directory, when I typed the **"ls"** command it shows all the presented lists under the directory named as "levels". The Figure 6 shows what are the lists found under the directory named "levels".

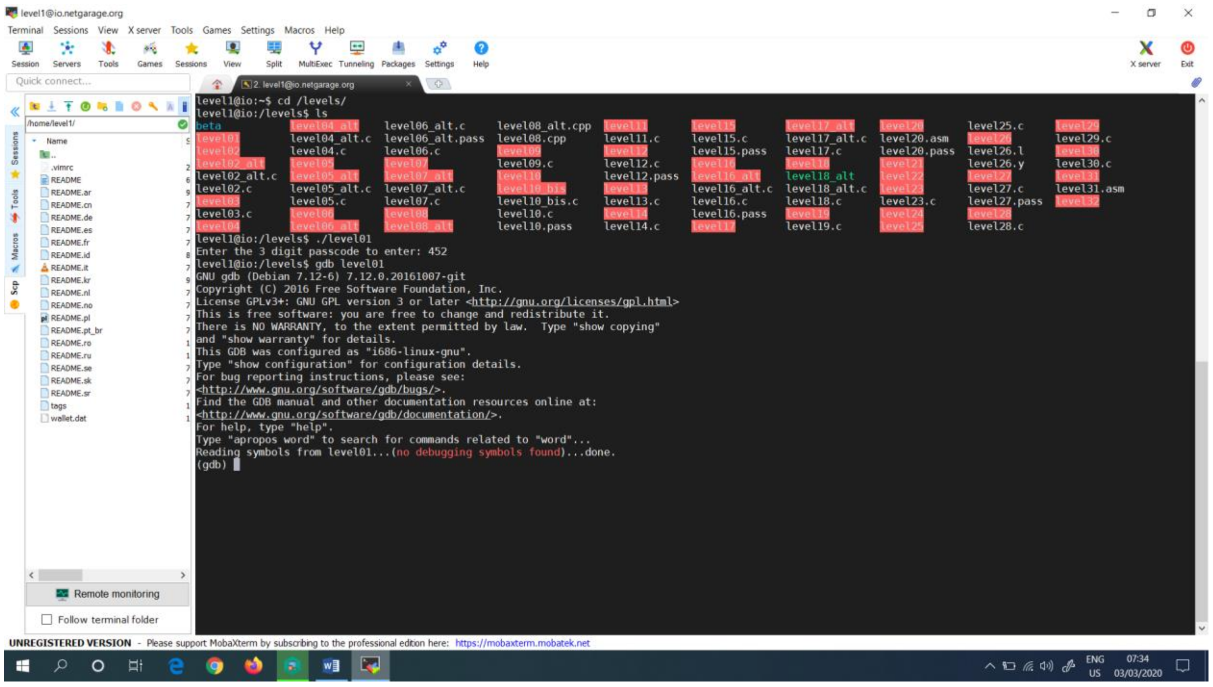


```
level1@io:~$ cd /levels/
level1@io:/levels$ ls
beta           level04_alt    level06_alt.c     level08_alt.cpp  level11       level15         level17_alt    level20       level25.c     level29
level01        level04_alt.c  level06_alt.pass  level08.cpp      level11.c     level15.c       level17_alt.c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

level1@io.netgarage.org
Terminal Sessions View X server Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
X server Exit
2. level1@io.netgarage.org
/home/level1/
Remote monitoring
Follow terminal folder
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

*Figure 6: Typed "ls" command, to view the files under the directory named as "levels"*

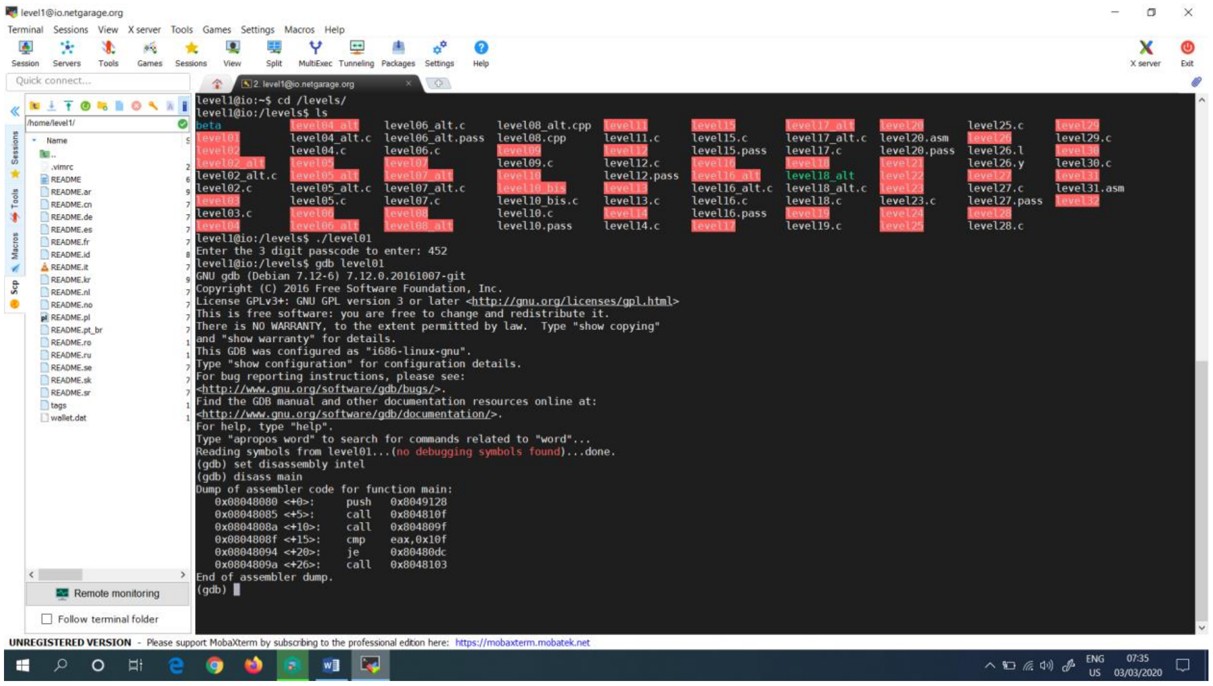
Then, in order to enter inside the level01, I typed the command as **". /level01"**. After that, it helps to enter inside the level01 terminal. Then the terminal asks to type the 3-digit passcode, for the security reasons. Then I typed the passcode as 452. Next, I typed as **"gdb level01"**. So, the terminal changed as **"gdb"** in order to get the upcoming commands. After this, the commands will also be written under the terminal named as **"gdb"**. The Figure 7, shows the above mentioned scenario.



```
level1@io:~$ cd /levels/
level1@io:/levels$ ls
beta          level04_alt    level06_alt.c     level08_alt.cpp  level11      level15         level17_alt    level20        level25.c     level29
level01       level04_alt.c  level06_alt.pass  level08.cpp      level11.c    level15.c       level17_alt.c  level20.asm    level26       level29.c
level02       level04.c      level06.c         level09          level12      level15.pass    level17.c      level20.pass   level26.l     level30
level02_alt   level05        level07           level09.c        level12.c    level16         level18        level21        level26.y     level30.c
level02_alt.c level05_alt    level07_alt       level10          level12.pass level16_alt     level18_alt    level22        level27       level31
level02.c     level05_alt.c  level07_alt.c     level10_bis      level13      level16_alt.c   level18_alt.c  level23        level27.c     level31.asm
level03       level05.c      level07.c         level10_bis.c    level13.c    level16.c       level18.c      level23.c      level27.pass  level32
level03.c     level06        level08           level10.c        level14      level16.pass    level19        level24        level28
level04       level06_alt    level08_alt       level10.pass     level14.c    level17         level19.c      level25        level28.c
level1@io:/levels$ ./level01
Enter the 3 digit passcode to enter: 452
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) 
```

*Figure 7: The terminal changed as "gdb"*

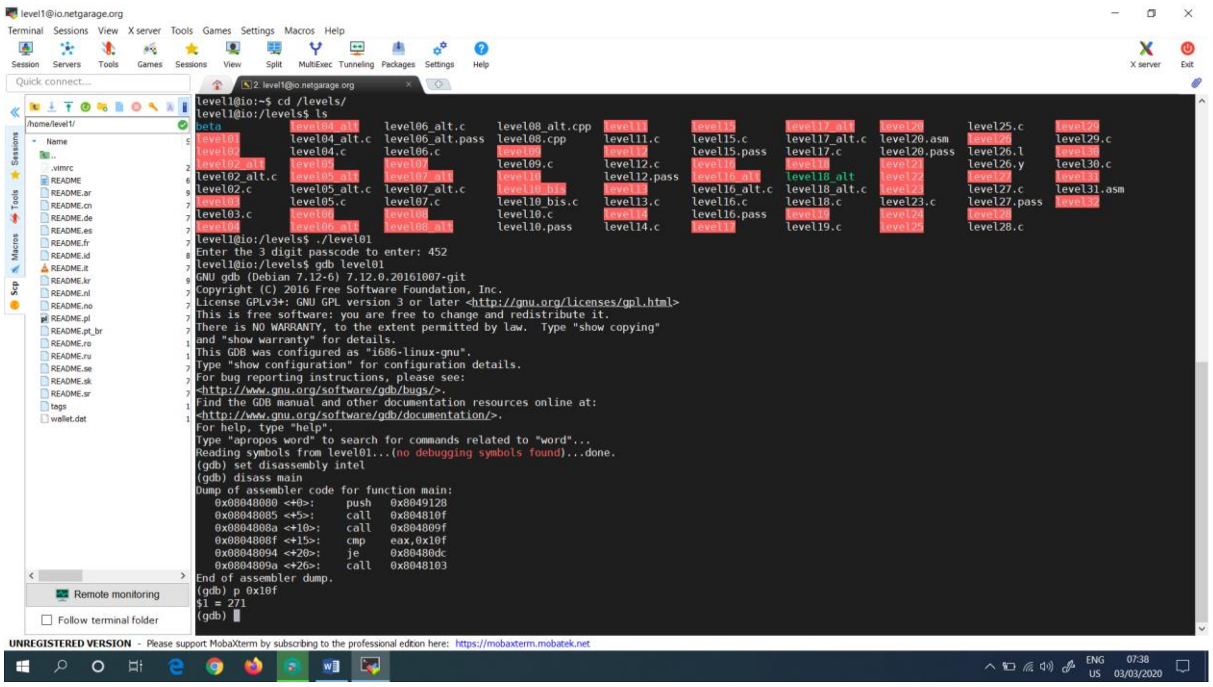
Next, I typed the command as **"set disassembly intel"**. This command helps to gain, gdb which uses the Intel disassembly style, which is popular among Windows users. Next, under the **"gdb"** terminal, I again typed as **"diass main"**. It helps to specifies the function to disassemble. This helps to see what are the assembler code available for the main function. And also it helps to produce the disassembly output of the entire function. The Figure 8 shows what are the available assembler functions under the main function.



```
level1@io:~$ cd /levels/
level1@io:/levels$ ls
beta          level04_alt    level06_alt.c     level08_alt.cpp  level11      level15         level17_alt    level20        level25.c     level29
level01       level04_alt.c  level06_alt.pass  level08.cpp      level11.c    level15.c       level17_alt.c  level20.asm    level26       level29.c
level02       level04.c      level06.c         level09          level12      level15.pass    level17.c      level20.pass   level26.l     level30
level02_alt   level05        level07           level09.c        level12.c    level16         level18        level21        level26.y     level30.c
level02_alt.c level05_alt    level07_alt       level10          level12.pass level16_alt     level18_alt    level22        level27       level31
level02.c     level05_alt.c  level07_alt.c     level10_bis      level13      level16_alt.c   level18_alt.c  level23        level27.c     level31.asm
level03       level05.c      level07.c         level10_bis.c    level13.c    level16.c       level18.c      level23.c      level27.pass  level32
level03.c     level06        level08           level10.c        level14      level16.pass    level19        level24        level28
level04       level06_alt    level08_alt       level10.pass     level14.c    level17         level19.c      level25        level28.c
level1@io:/levels$ ./level01
Enter the 3 digit passcode to enter: 452
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) set disassembly intel
(gdb) disass 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 8: Available assembler code for the main function*

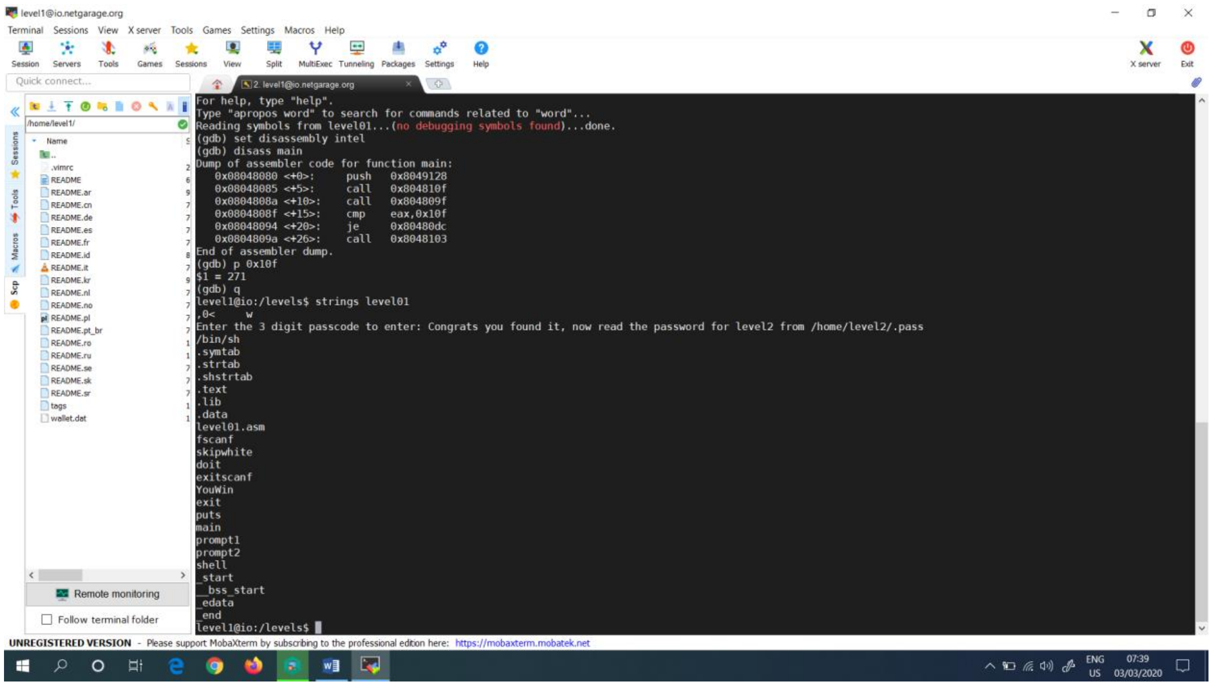
To compare the two files byte by byte the **“cmp”** command is used. Then, it helps to find out whether the two files are identical or not. The value is a hexadecimal value, so it can be display with its decimal value **“p”** in gdb. Then, I compared the entered value, which is stored in the **“eax”** register. Figure 9 helps to understand the above mentioned procedures.



```
level1@io:~$ cd /levels/
level1@io:/levels$ ls
beta          level04_alt    level06_alt.c     level08_alt.cpp  level11        level15        level17_alt    level20        level25.c      level29
level01       level04_alt.c  level06_alt.pass  level08.cpp      level11.c      level15.c      level17_alt.c  level20.asm    level26        level29.c
level02       level04.c      level06.c         level09          level12        level15.pass   level17.c      level20.pass   level26.l      level30
level02_alt   level05        level07           level09.c        level12.c      level16        level18        level21        level26.y      level30.c
level02_alt.c level05_alt    level07_alt       level10          level12.pass   level16_alt    level18_alt    level22        level27        level31
level02.c     level05_alt.c  level07_alt.c     level10_bis      level13        level16_alt.c  level18_alt.c  level23        level27.c      level31.asm
level03       level05.c      level07.c         level10_bis.c    level13.c      level16.c      level18.c      level23.c      level27.pass   level32
level03.c     level06        level08           level10.c        level14        level16.pass   level19        level24        level28
level04       level06_alt    level08_alt       level10.pass     level14.c      level17        level19.c      level25        level28.c
level1@io:/levels$ ./level01
Enter the 3 digit passcode to enter: 452
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) set disassembly intel
(gdb) disass 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) p 0x10f
$1 = 271
(gdb)
```

*Figure 9: Compare the value with the “eax” register*

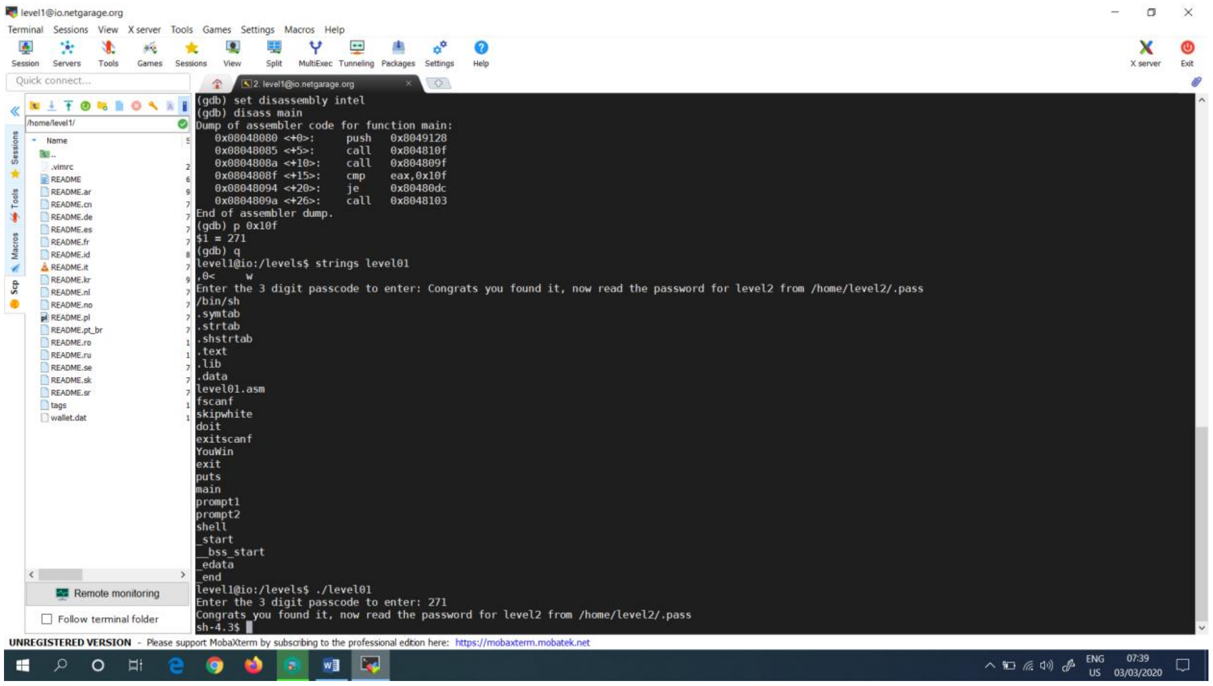
Next, I typed as **“strings level01”**. Because, to see what are the strings available in level01. With the Figure 10, I can identify the word **“YouWin”**.



```
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from level01...(no debugging symbols found)...done.
(gdb) set disassembly intel
(gdb) disass 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) p 0x10f
$1 = 271
(gdb) q
level1@io:/levels$ strings level01
,0<     w
Enter the 3 digit passcode to enter: Congrats you found it, now read the password for level2 from /home/level2/.pass
/bin/sh
.symtab
.strtab
.shstrtab
.text
.lib
.data
level01.asm
fscanf
skipwhite
doit
exitscanf
YouWin
exit
puts
main
prompt1
prompt2
shell
_start
__bss_start
_edata
_end
level1@io:/levels$
```

*Figure 10: Identified the string "YouWin"*

Again, it asks to enter for the 3-digit passcode, and typed as 271. This is shown in the Figure 11.



```
(gdb) set disassembly intel
(gdb) disass 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) p 0x10f
$1 = 271
(gdb) q
level1@io:/levels$ strings level01
,0<     w
Enter the 3 digit passcode to enter: Congrats you found it, now read the password for level2 from /home/level2/.pass
/bin/sh
.symtab
.strtab
.shstrtab
.text
.lib
.data
level01.asm
fscanf
skipwhite
doit
exitscanf
YouWin
exit
puts
main
prompt1
prompt2
shell
_start
__bss_start
_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 11: Entered the 3-digit passcode*

Again, typed the **"ls"** command to get the lists. This is shown in Figure 12.

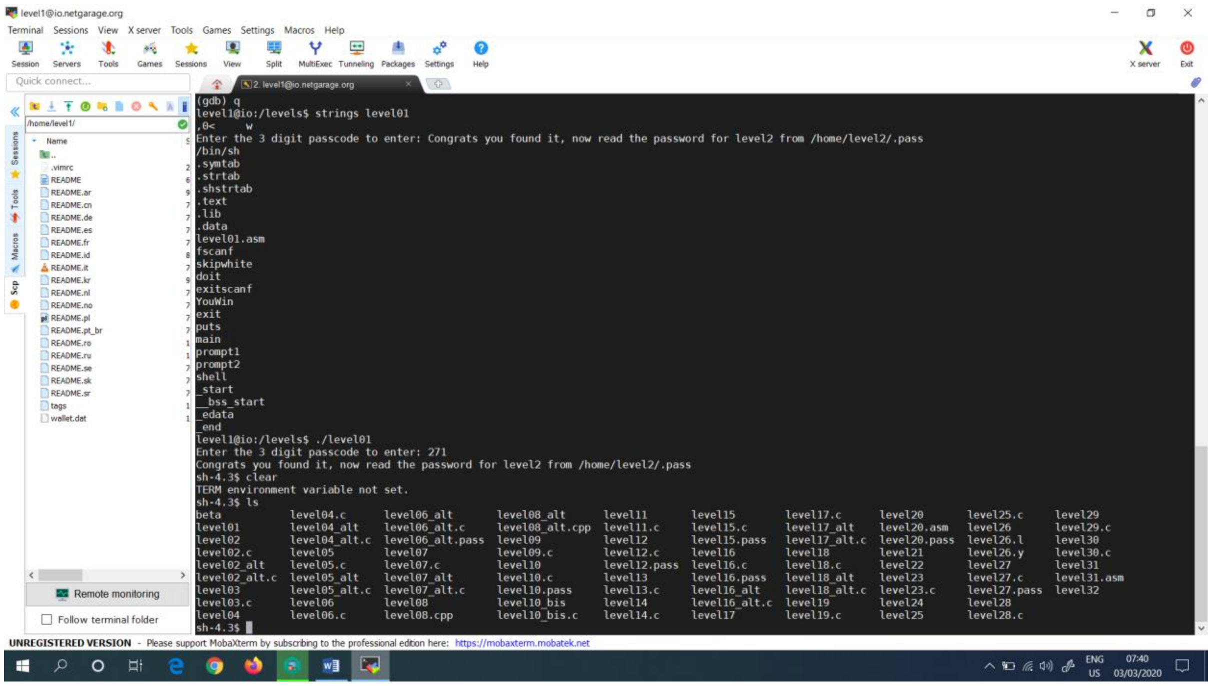

```
(gdb) q
level1@io:/levels$ strings level01
,0<    w
Enter the 3 digit passcode to enter: Congrats you found it, now read the password for level2 from /home/level2/.pass
/bin/sh
.symtab
.strtab
.shstrtab
.text
.lib
.data
level01.asm
fscanf
skipwhite
doit
exitscanf
YouWin
exit
puts
main
prompt1
prompt2
shell
_start
__bss_start
_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$ clear
TERM environment variable not set.
sh-4.3$ ls
beta          level04.c     level06_alt       level08_alt      level11       level15        level17.c      level20        level25.c      level29
level01       level04_alt   level06_alt.c     level08_alt.cpp  level11.c     level15.c      level17_alt    level20.asm    level26        level29.c
level02       level04_alt.c level06_alt.pass  level09          level12       level15.pass   level17_alt.c  level20.pass   level26.l      level30
level02.c     level05       level07           level09.c        level12.c     level16        level18        level21        level26.y      level30.c
level02_alt   level05.c     level07.c         level10          level12.pass  level16.c      level18.c      level22        level27        level31
level02_alt.c level05_alt   level07_alt       level10.c        level13       level16.pass   level18_alt    level23        level27.c      level31.asm
level03       level05_alt.c level07_alt.c     level10.pass     level13.c     level16_alt    level18_alt.c  level23.c      level27.pass   level32
level03.c     level06       level08           level10_bis      level14       level16_alt.c  level19        level24        level28
level04       level06.c     level08.cpp       level10_bis.c    level14.c     level17        level19.c      level25        level28.c
sh-4.3$
```

*Figure 12: Entered the "ls" command*

Next, typed as **"id"**, in order to get the number of the id. The following Figure 13 shows the following.

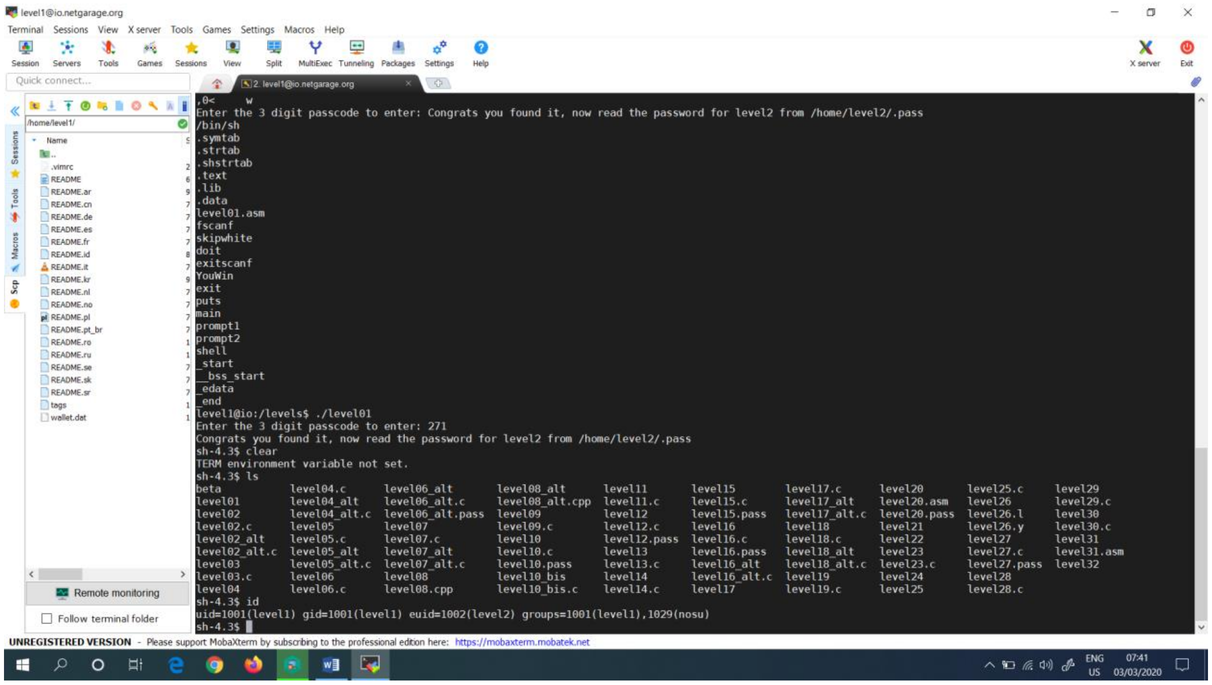
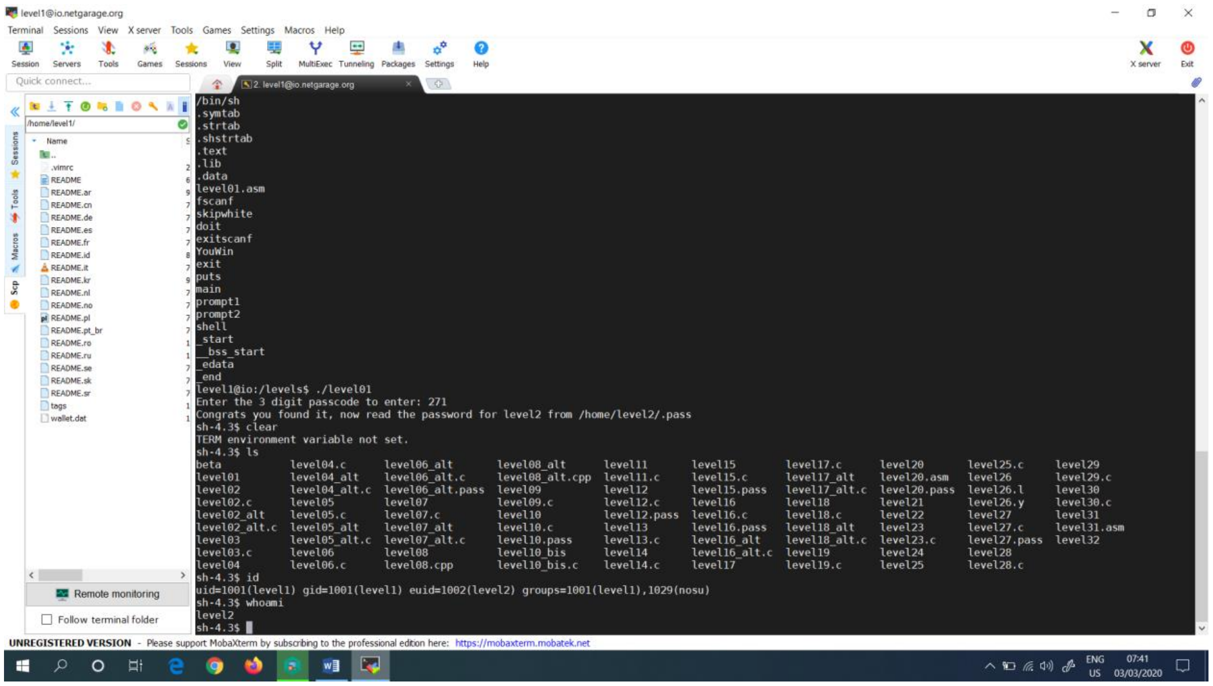

```
,0<    w
Enter the 3 digit passcode to enter: Congrats you found it, now read the password for level2 from /home/level2/.pass
/bin/sh
.symtab
.strtab
.shstrtab
.text
.lib
.data
level01.asm
fscanf
skipwhite
doit
exitscanf
YouWin
exit
puts
main
prompt1
prompt2
shell
_start
__bss_start
_edata
_end
level1@io:/levels$ ./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$ clear
TERM environment variable not set.
sh-4.3$ ls
beta          level04.c     level06_alt       level08_alt      level11       level15        level17.c      level20        level25.c      level29
level01       level04_alt   level06_alt.c     level08_alt.cpp  level11.c     level15.c      level17_alt    level20.asm    level26        level29.c
level02       level04_alt.c level06_alt.pass  level09          level12       level15.pass   level17_alt.c  level20.pass   level26.l      level30
level02.c     level05       level07           level09.c        level12.c     level16        level18        level21        level26.y      level30.c
level02_alt   level05.c     level07.c         level10          level12.pass  level16.c      level18.c      level22        level27        level31
level02_alt.c level05_alt   level07_alt       level10.c        level13       level16.pass   level18_alt    level23        level27.c      level31.asm
level03       level05_alt.c level07_alt.c     level10.pass     level13.c     level16_alt    level18_alt.c  level23.c      level27.pass   level32
level03.c     level06       level08           level10_bis      level14       level16_alt.c  level19        level24        level28
level04       level06.c     level08.cpp       level10_bis.c    level14.c     level17        level19.c      level25        level28.c
sh-4.3$ id
uid=1001(level1) gid=1001(level1) euid=1002(level2) groups=1001(level1),1029(nosu)
sh-4.3$
```

*Figure 13: Entered the "id" command*

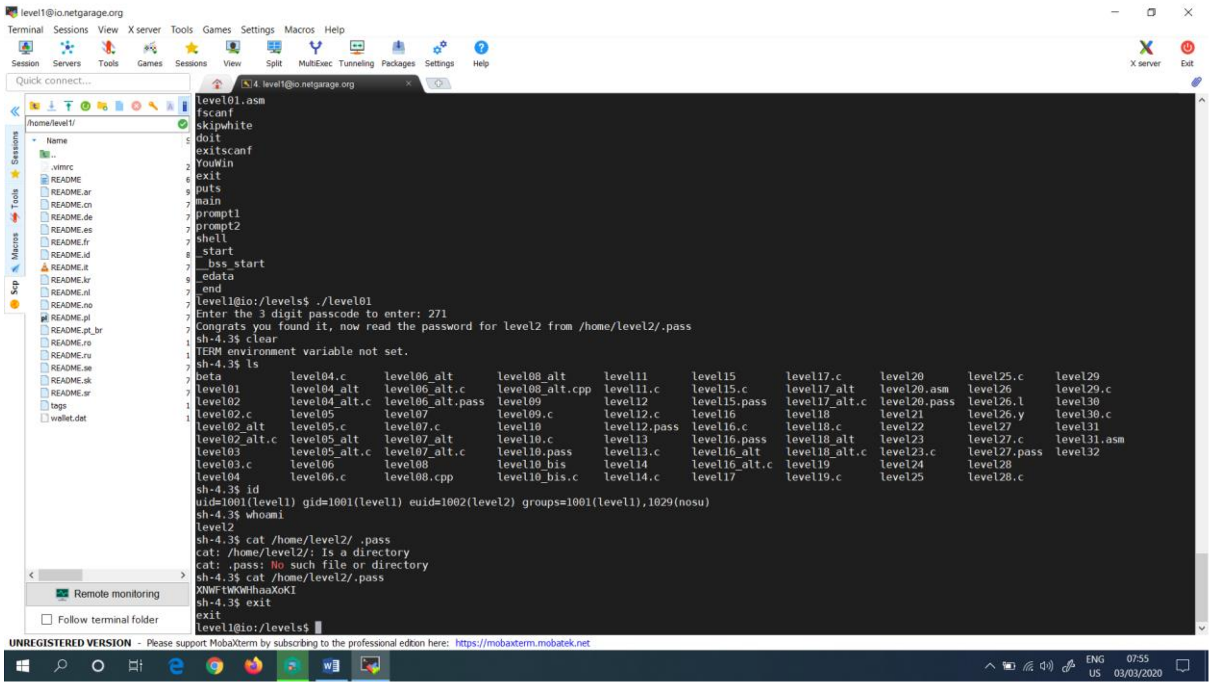
Next, entered the **"whoami"** command, to check in which level I am in. This is shown in the Figure 14.



```
/bin/sh
.symtab
.strtab
.shstrtab
.text
.lib
.data
level01.asm
fscanf
skipwhite
doit
exitscanf
YouWin
exit
puts
main
prompt1
prompt2
shell
_start
__bss_start
_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$ clear
TERM environment variable not set.
sh-4.3$ ls
beta          level04.c     level06_alt       level08_alt     level11       level15         level17.c      level20        level25.c     level29
level01       level04_alt   level06_alt.c     level08_alt.cpp level11.c     level15.c       level17_alt    level20.asm    level26       level29.c
level02       level04_alt.c level06_alt.pass  level09         level12       level15.pass    level17_alt.c  level20.pass   level26.l     level30
level02.c     level05       level07           level09.c       level12.c     level16         level18        level21        level26.y     level30.c
level02_alt   level05.c     level07.c         level10         level12.pass  level16.c       level18.c      level22        level27       level31
level02_alt.c level05_alt   level07_alt       level10.c       level13       level16.pass    level18_alt    level23        level27.c     level31.asm
level03       level05_alt.c level07_alt.c     level10.pass    level13.c     level16_alt     level18_alt.c  level23.c      level27.pass  level32
level03.c     level06       level08           level10_bis     level14       level16_alt.c   level19        level24        level28
level04       level06.c     level08.cpp       level10_bis.c   level14.c     level17         level19.c      level25        level28.c
sh-4.3$ id
uid=1001(level1) gid=1001(level1) euid=1002(level2) groups=1001(level1),1029(nosu)
sh-4.3$ whoami
level2
sh-4.3$
```

*Figure 14: Entered the "whoami" command*

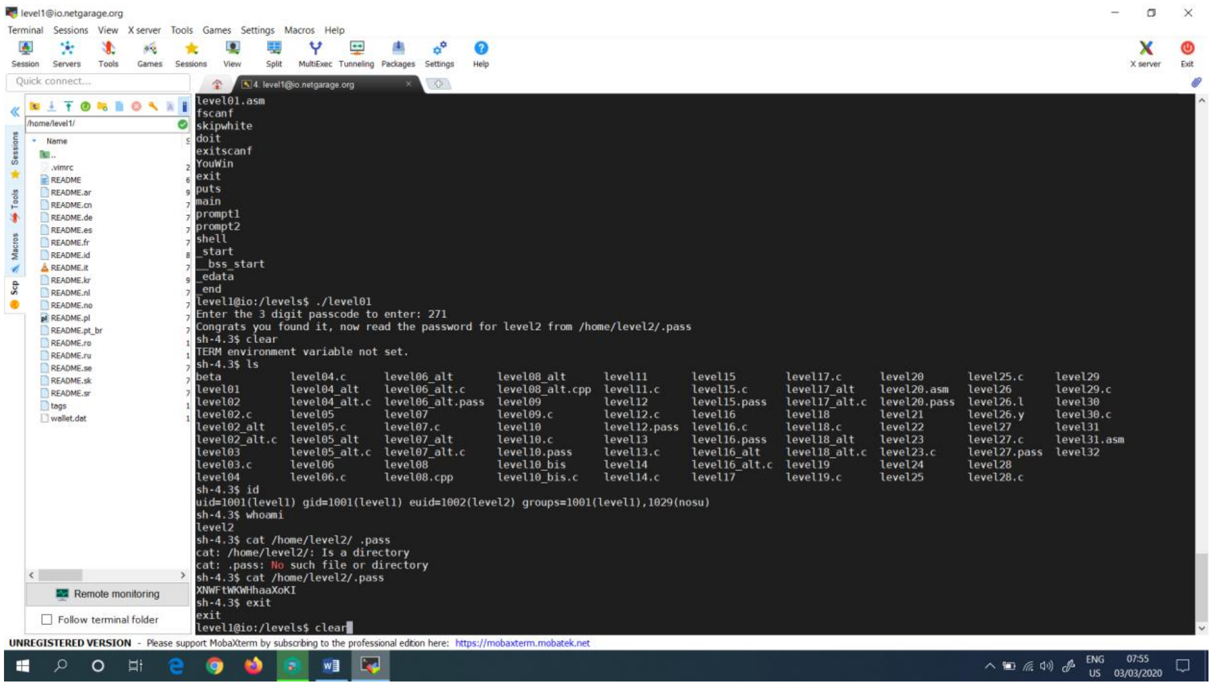
To find the ssh password for level 2, I typed the command as **" cat /home/level2/.pass"**. Then I got the password as **"XNWFtWKWHhaaXoKI"**. **"je"** will jump to the label if the values are equal. Since this jump command which is presented with the String **"YouWin"**, I assumed the password is 271. This is shown in Figure 15.



```
level01.asm
fscanf
skipwhite
doit
exitscanf
YouWin
exit
puts
main
prompt1
prompt2
shell
_start
__bss_start
_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$ clear
TERM environment variable not set.
sh-4.3$ ls
beta          level04.c     level06_alt       level08_alt     level11       level15         level17.c      level20        level25.c     level29
level01       level04_alt   level06_alt.c     level08_alt.cpp level11.c     level15.c       level17_alt    level20.asm    level26       level29.c
level02       level04_alt.c level06_alt.pass  level09         level12       level15.pass    level17_alt.c  level20.pass   level26.l     level30
level02.c     level05       level07           level09.c       level12.c     level16         level18        level21        level26.y     level30.c
level02_alt   level05.c     level07.c         level10         level12.pass  level16.c       level18.c      level22        level27       level31
level02_alt.c level05_alt   level07_alt       level10.c       level13       level16.pass    level18_alt    level23        level27.c     level31.asm
level03       level05_alt.c level07_alt.c     level10.pass    level13.c     level16_alt     level18_alt.c  level23.c      level27.pass  level32
level03.c     level06       level08           level10_bis     level14       level16_alt.c   level19        level24        level28
level04       level06.c     level08.cpp       level10_bis.c   level14.c     level17         level19.c      level25        level28.c
sh-4.3$ id
uid=1001(level1) gid=1001(level1) euid=1002(level2) groups=1001(level1),1029(nosu)
sh-4.3$ whoami
level2
sh-4.3$ cat /home/level2/ .pass
cat: /home/level2/: Is a directory
cat: .pass: No such file or directory
sh-4.3$ cat /home/level2/.pass
XNWFtWKWHhaaXoKI
sh-4.3$ exit
exit
level1@io:/levels$
```

*Figure 15: Got the password for level 2*

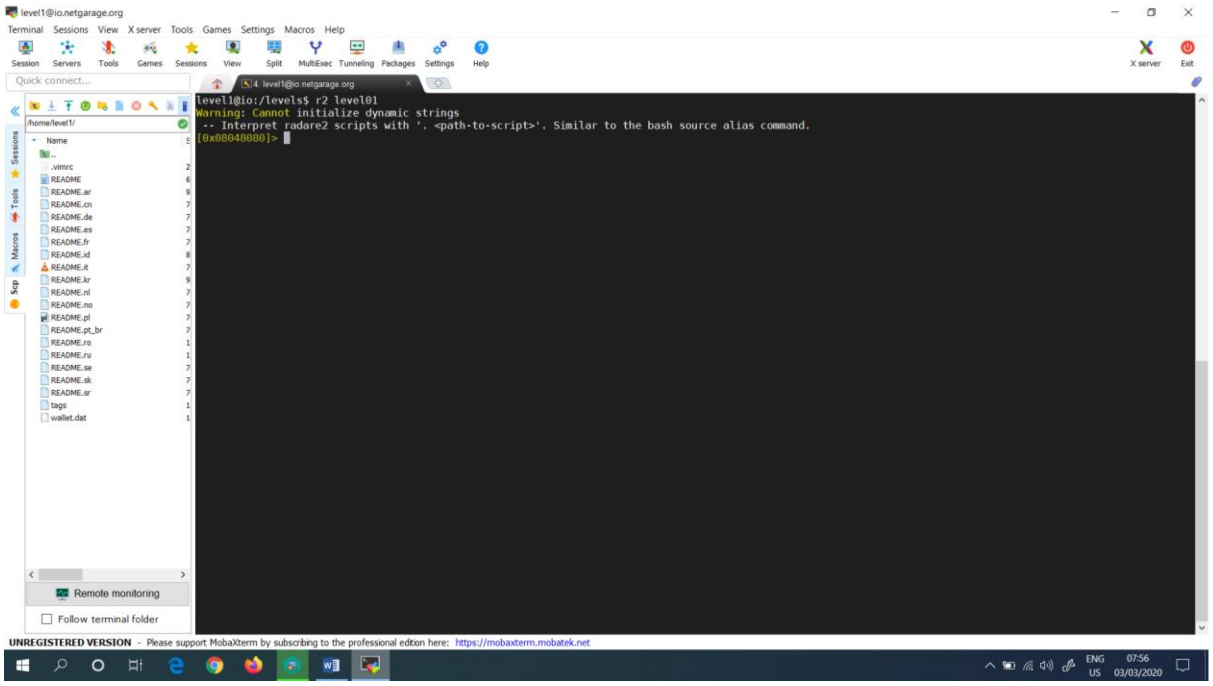
Then, cleared the terminal. This is shown in Figure 16.



```
level01.asm
fscanf
skipwhite
doit
exitscanf
YouWin
exit
puts
main
prompt1
prompt2
shell
_start
__bss_start
_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$ clear
TERM environment variable not set.
sh-4.3$ ls
beta           level04.c      level06_alt       level08_alt      level11       level15        level17.c      level20       level25.c     level29
level01        level04_alt    level06_alt.c     level08_alt.cpp  level11.c     level15.c      level17_alt    level20.asm   level26       level29.c
level02        level04_alt.c  level06_alt.pass  level09          level12       level15.pass   level17_alt.c  level20.pass  level26.l     level30
level02.c      level05        level07           level09.c        level12.c     level16        level18        level21       level26.y     level30.c
level02_alt    level05.c      level07.c         level10          level12.pass  level16.c      level18.c      level22       level27       level31
level02_alt.c  level05_alt    level07_alt       level10.c        level13       level16.pass   level18_alt    level23       level27.c     level31.asm
level03        level05_alt.c  level07_alt.c     level10.pass     level13.c     level16_alt    level18_alt.c  level23.c     level27.pass  level32
level03.c      level06        level08           level10_bis      level14       level16_alt.c  level19        level24       level28
level04        level06.c      level08.cpp       level10_bis.c    level14.c     level17        level19.c      level25       level28.c
sh-4.3$ id
uid=1001(level1) gid=1001(level1) euid=1002(level2) groups=1001(level1),1029(nosu)
sh-4.3$ whoami
level2
sh-4.3$ cat /home/level2/ .pass
cat: /home/level2/: Is a directory
cat: .pass: No such file or directory
sh-4.3$ cat /home/level2/.pass
XNWFtWKWHhaaXoKI
sh-4.3$ exit
exit
level1@io:/levels$ clear
```

*Figure 16: Cleared the terminal*

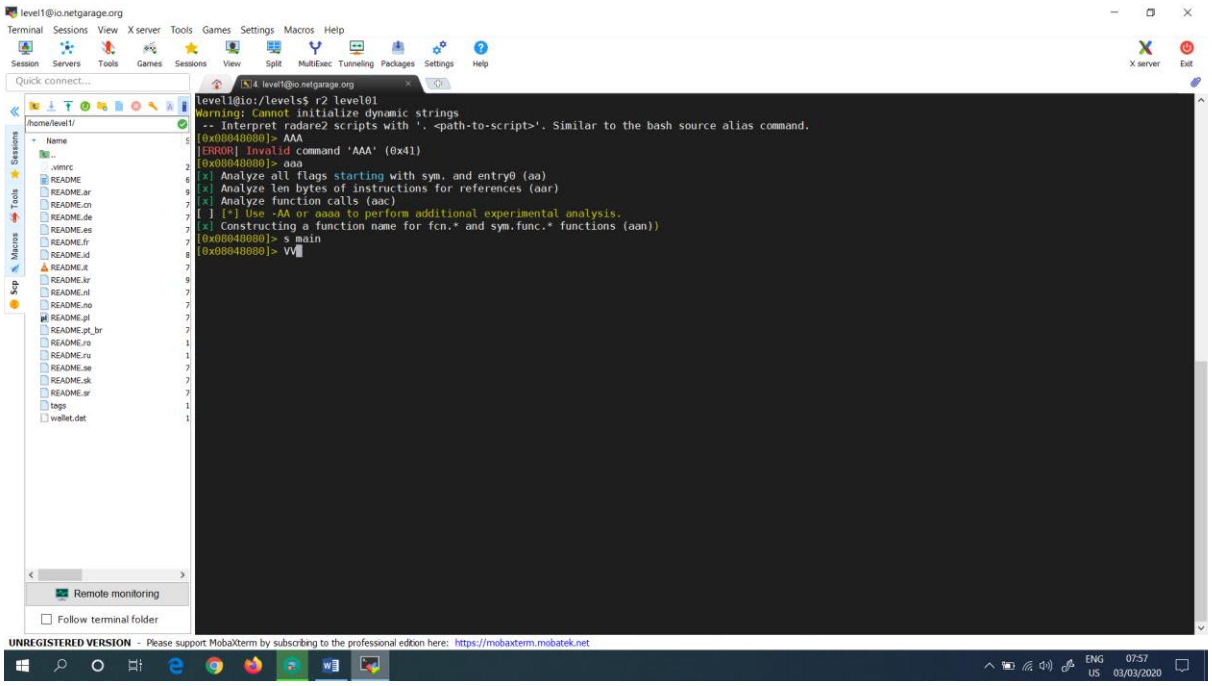
Then, entered inside the memory address which is in level01. Shown in Figure 17.



```
level1@io:/levels$ r2 level01
Warning: Cannot initialize dynamic strings
 -- Interpret radare2 scripts with '. <path-to-script>'. Similar to the bash source alias command.
[0x08048080]>
```

*Figure 17: Entered inside the memory address*

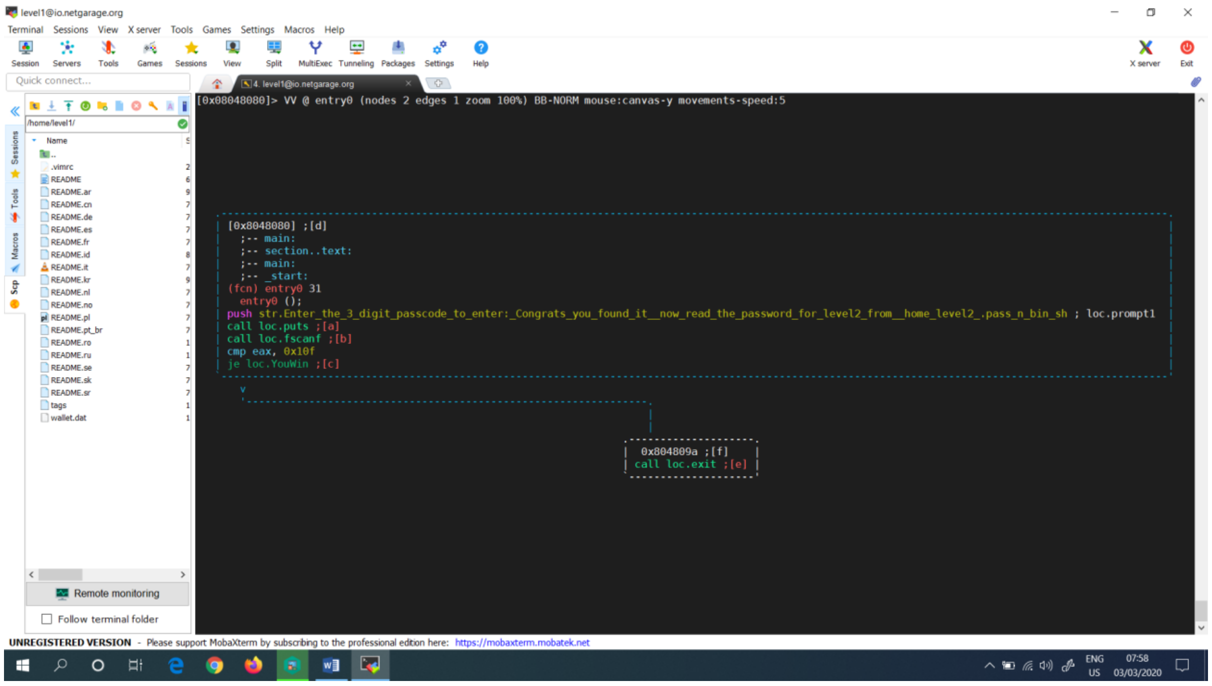
Then, went inside the main function of that particular memory address, which is shown in Figure 18.



```
level1@io:/levels$ r2 level01
Warning: Cannot initialize dynamic strings
 -- Interpret radare2 scripts with '. <path-to-script>'. Similar to the bash source alias command.
[0x08048080]> AAA
|ERROR| Invalid command 'AAA' (0x41)
[0x08048080]> aaa
[x] Analyze all flags starting with sym. and entry0 (aa)
[x] Analyze len bytes of instructions for references (aar)
[x] Analyze function calls (aac)
[ ] [*] Use -AA or aaaa to perform additional experimental analysis.
[x] Constructing a function name for fcn.* and sym.func.* functions (aan))
[0x08048080]> s main
[0x08048080]> VV
```

*Figure 18: Entered inside the main function of the particular memory address*

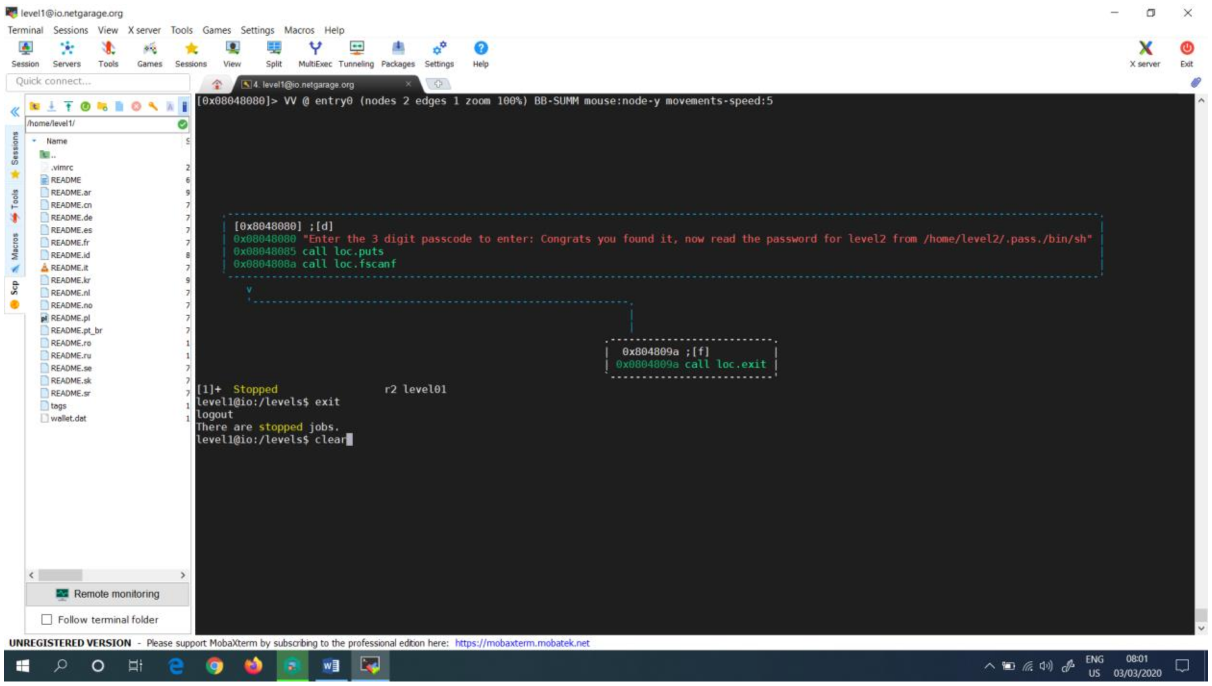
Then, it showed the following output (Figure 19).



```
[0x08048080]> VV @ entry0 (nodes 2 edges 1 zoom 100%) BB-NORM mouse:canvas-y movements-speed:5

 .-----------------------------------------------------------------------------------------------------------------------------.
 | [0x8048080] ;[d]                                                                                                            |
 |   ;-- main:                                                                                                                 |
 |   ;-- section..text:                                                                                                        |
 |   ;-- main:                                                                                                                 |
 |   ;-- _start:                                                                                                               |
 | (fcn) entry0 31                                                                                                             |
 |   entry0 ();                                                                                                                |
 | push str.Enter_the_3_digit_passcode_to_enter:_Congrats_you_found_it__now_read_the_password_for_level2_from__home_level2_.pass_n_bin_sh ; loc.prompt1 |
 | call loc.puts ;[a]                                                                                                          |
 | call loc.fscanf ;[b]                                                                                                        |
 | cmp eax, 0x10f                                                                                                              |
 | je loc.YouWin ;[c]                                                                                                          |
 `-----------------------------------------------------------------------------------------------------------------------------'
     v
     '-----------------------------------------------------.
                                                           |
                                                           |
                                                  .-----------------.
                                                  |  0x804809a ;[f]  |
                                                  | call loc.exit ;[e] |
                                                  `-----------------'
```

*Figure 19: Output*

Finally, I cleared the level01 terminal. This is shown in (Figure 20).



```
[0x08048080]> VV @ entry0 (nodes 2 edges 1 zoom 100%) BB-SUMM mouse:node-y movements-speed:5

     .------------------------------------------------------------------------------------------------------------.
     | [0x8048080] ;[d]                                                                                           |
     | 0x08048080 "Enter the 3 digit passcode to enter: Congrats you found it, now read the password for level2 from /home/level2/.pass./bin/sh" |
     | 0x08048085 call loc.puts                                                                                   |
     | 0x0804808a call loc.fscanf                                                                                 |
     `------------------------------------------------------------------------------------------------------------'
        v
        '--------------------------------------------------.
                                                           |
                                                           |
                                                   .-------------------------.
                                                   |  0x804809a ;[f]         |
                                                   | 0x0804809a call loc.exit |
                                                   `-------------------------'
[1]+  Stopped                 r2 level01
level1@io:/levels$ exit
logout
There are stopped jobs.
level1@io:/levels$ clear
```

*Figure 20: Cleared the level01 terminal*