Buffer-Overflow Attack

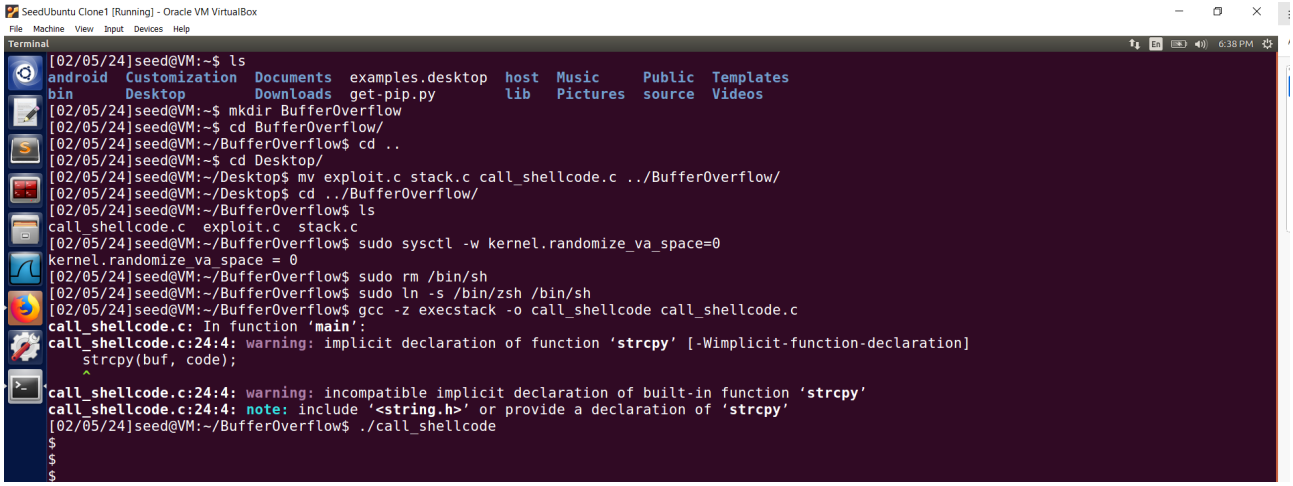
Programming Assignment 1

EDMUND LAWERH AMANOR System & Software Security, Spring 2024

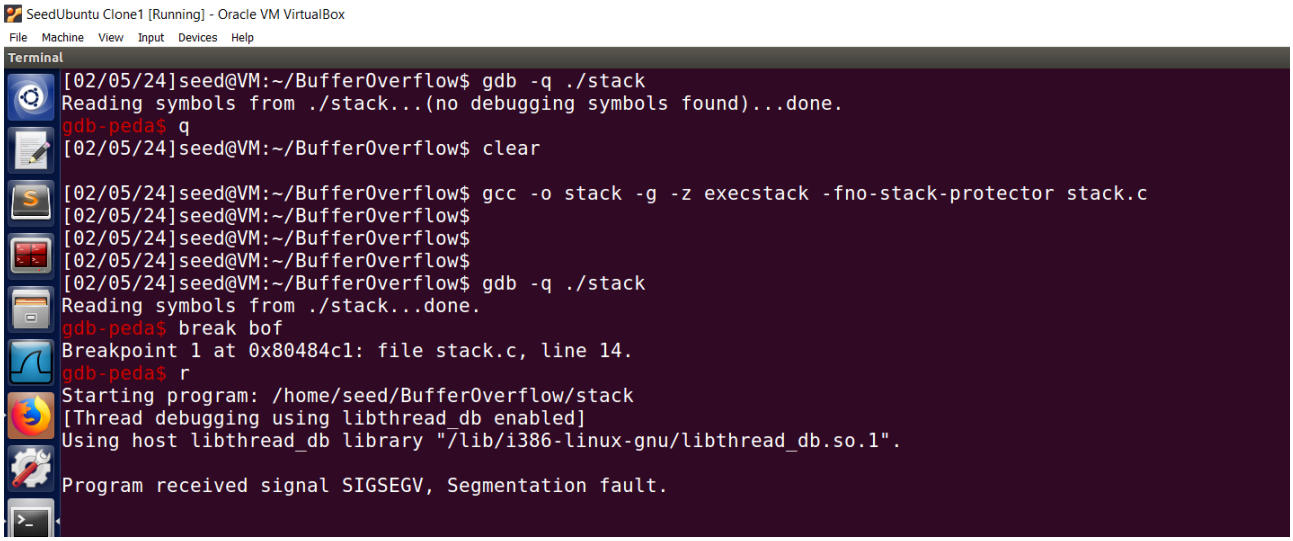
This report is a documentation of my observation of the process while I simulated the buffer-overflow attack in the SEED virtual machine. I provided screenshots, from the various stages.

I commented out the steps I took and observations made directly in the terminal window, both before and after executing commands to make is easy to follow what was happening.

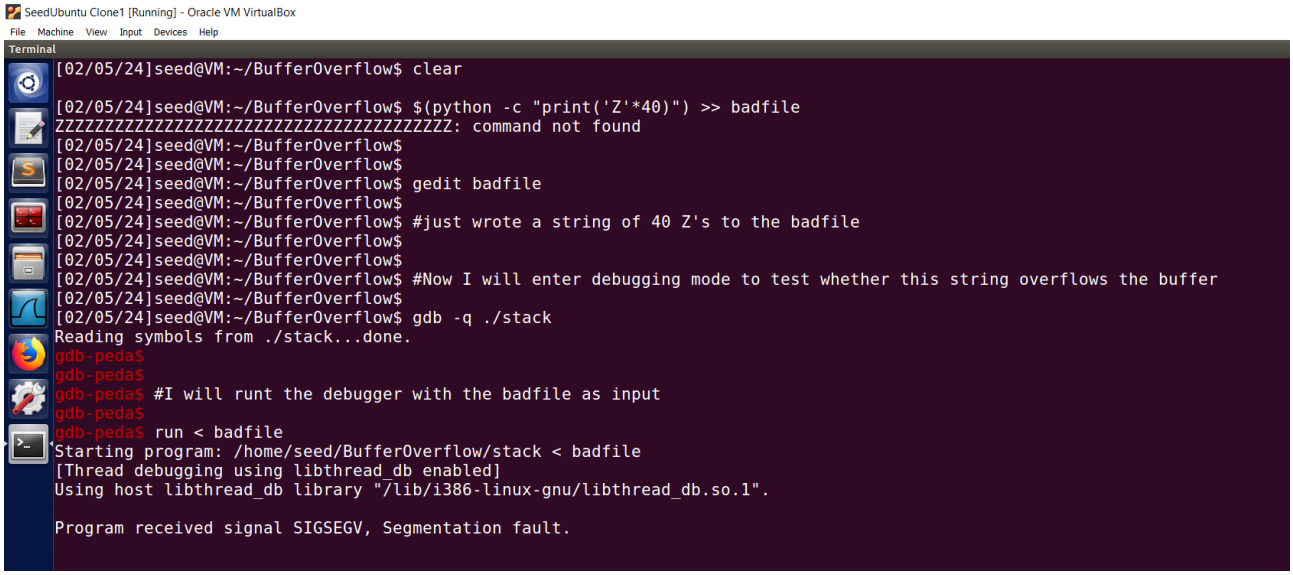
- The image below shows the initial setup procedure to test if the shell program we compiled really works before incorporating it into our exploit code.



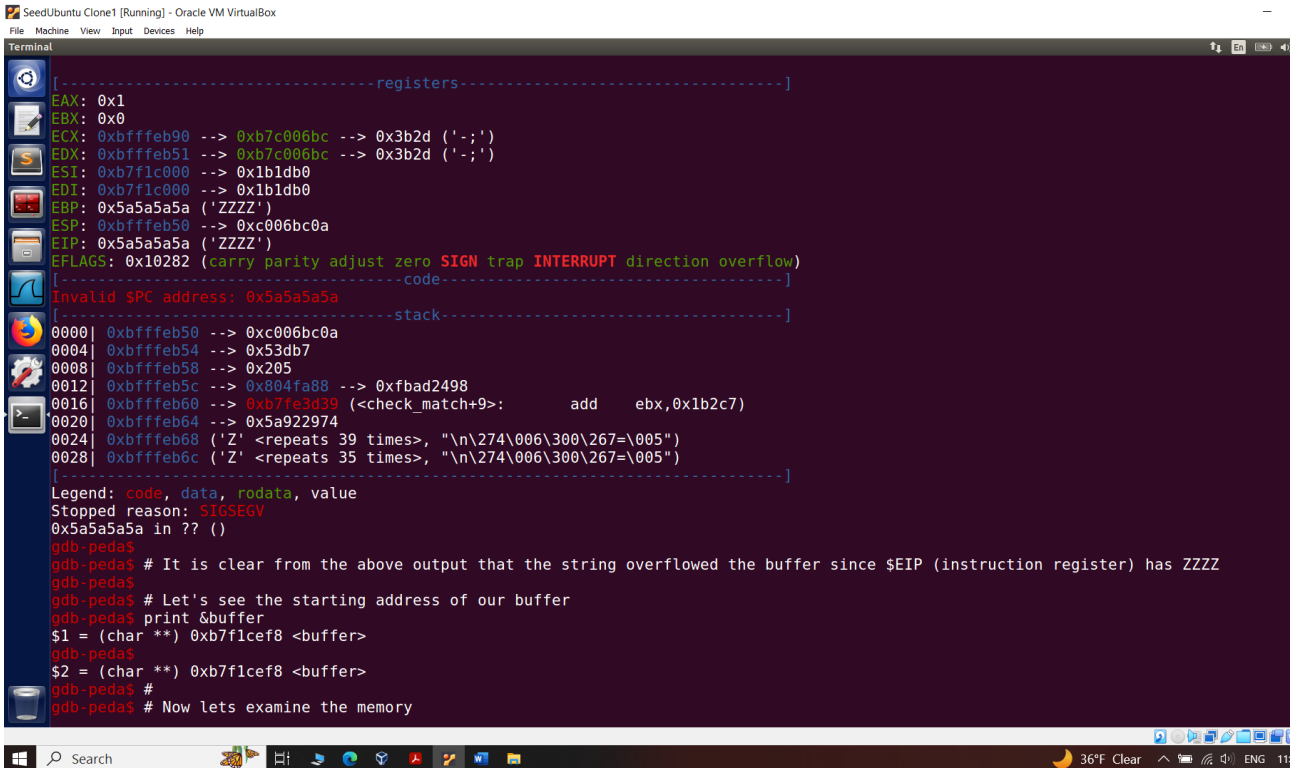
- Next, we show the compilation and debugging of the stack.c program to determine what our offset value should be.



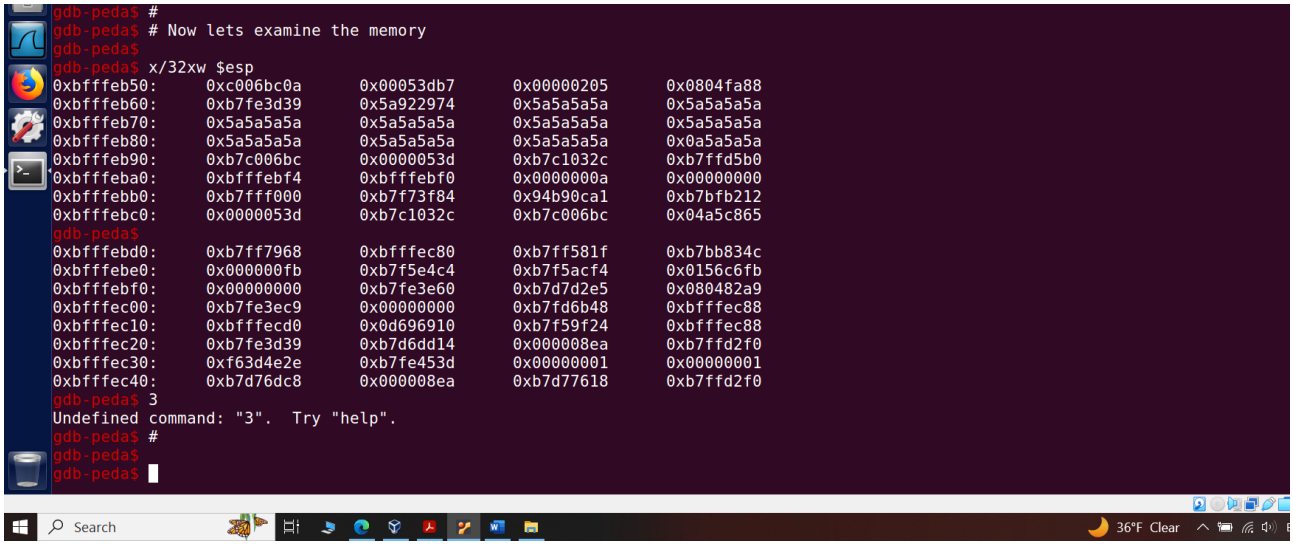
- At this point, I created a badfile manually, to test the vulnerable program [stack.c] to know exactly where the buffer begins to overflow into the $EBP register and then subsequently into the $EIP register.



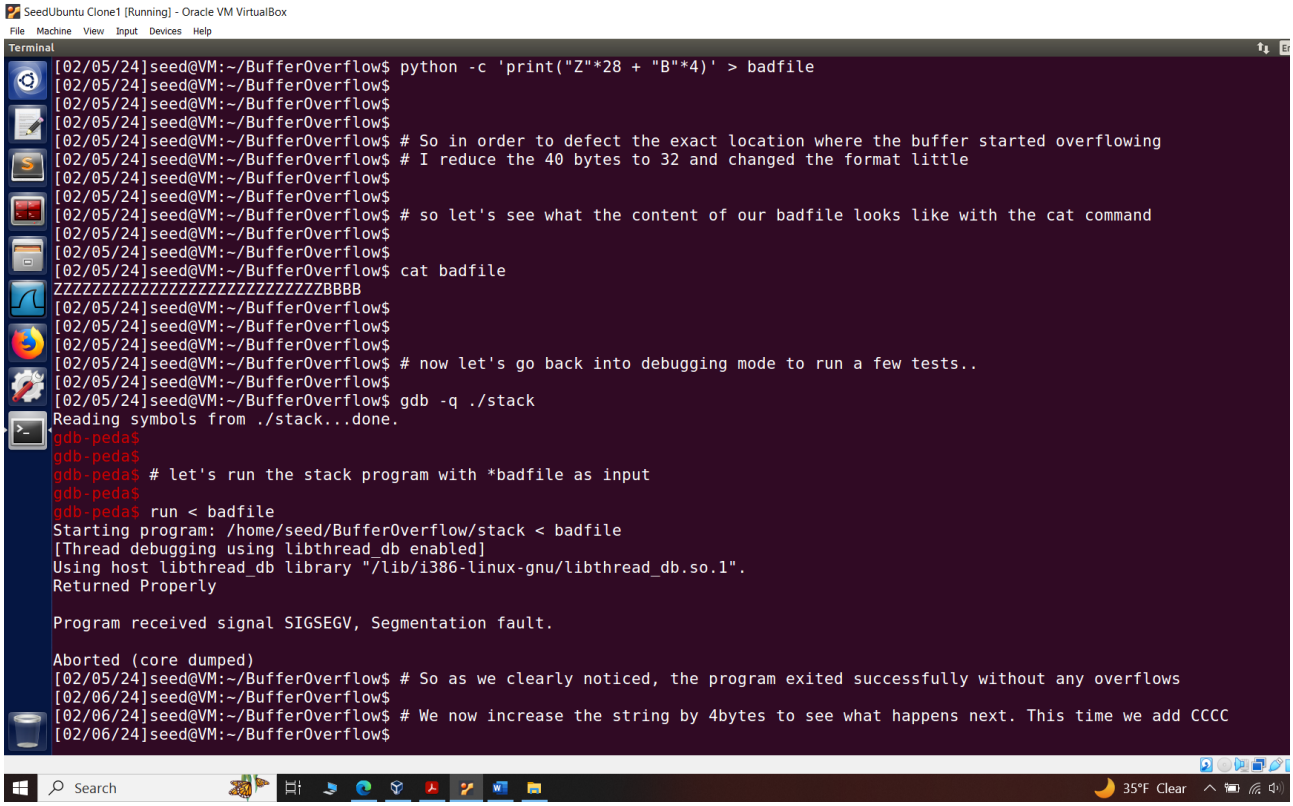
- The image below shows the content of the registers after trying to overflow the buffer with 40-string of Z (0x5a).



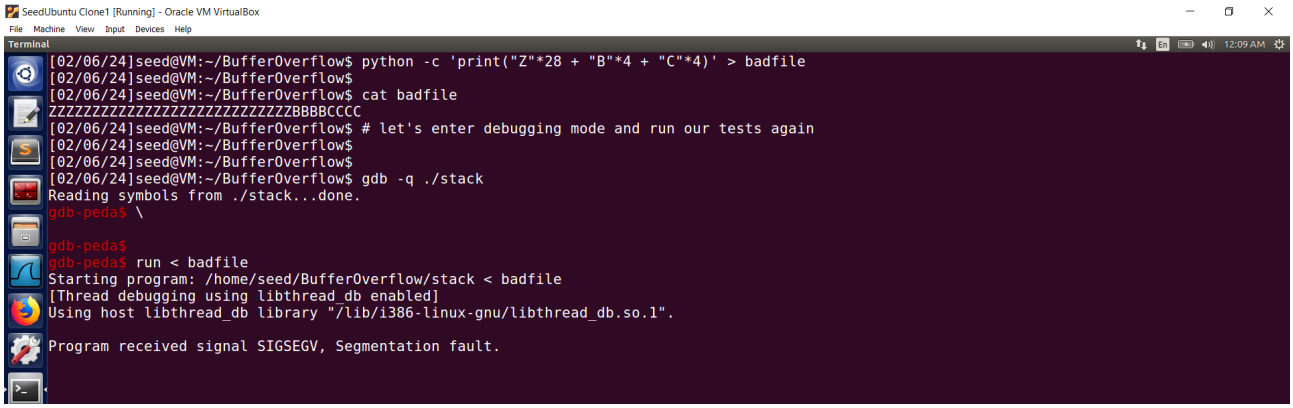
- Below is the memory dump when I examined it, following the overflow above.

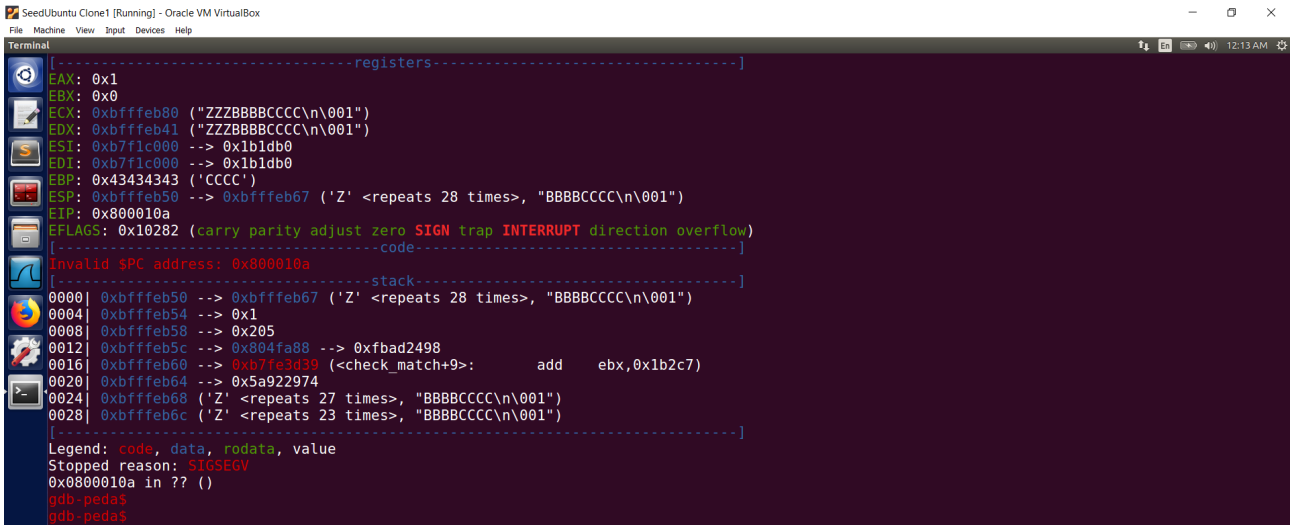


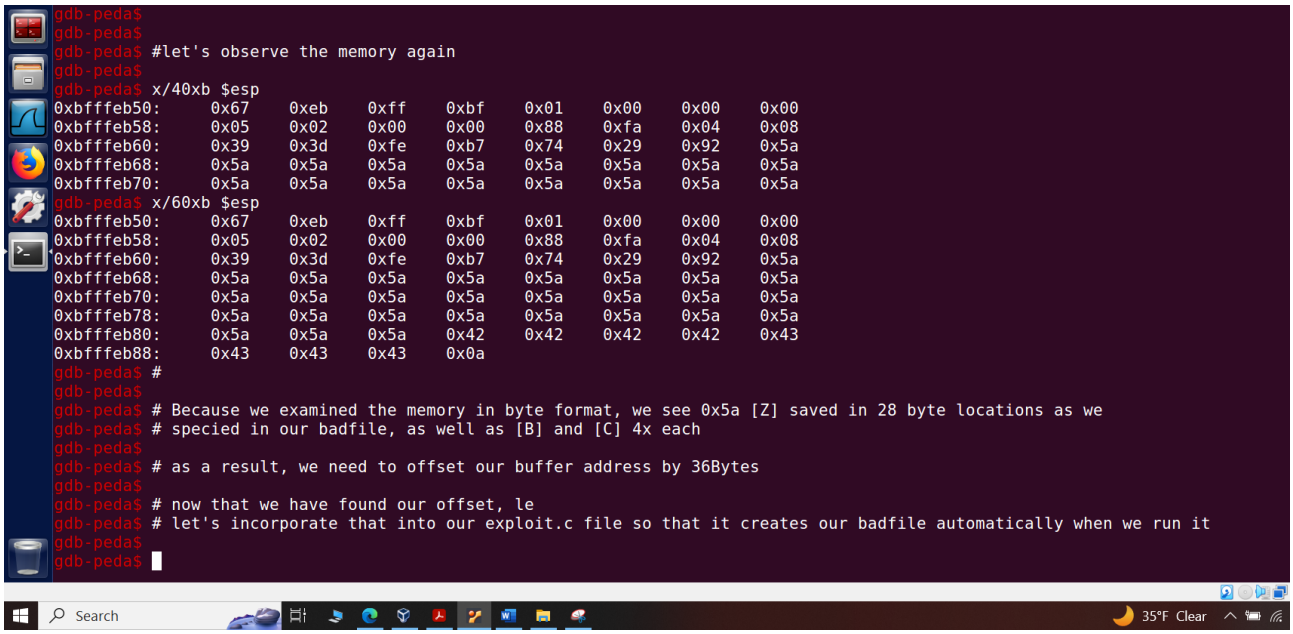
- The next snapshot give details on what I did to determine the exact offset value to use in my exploit code.



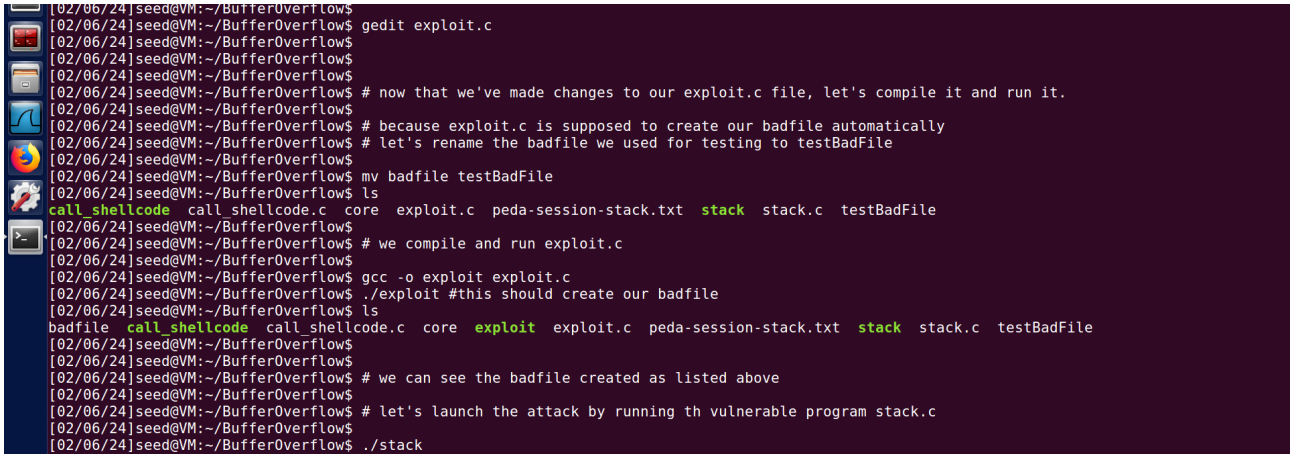
- The next three(3) snapshots gives details on the debugging process we just started above...

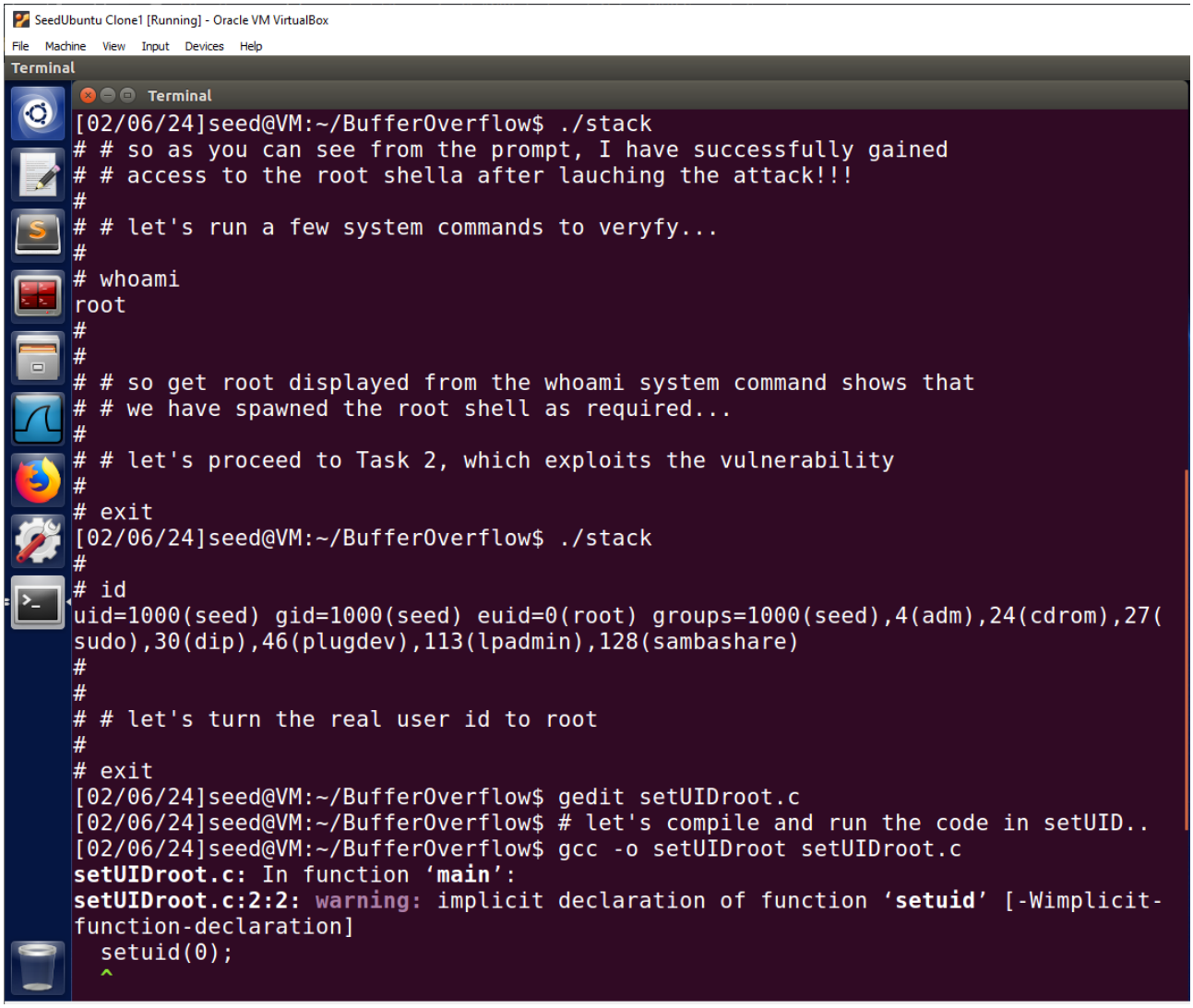


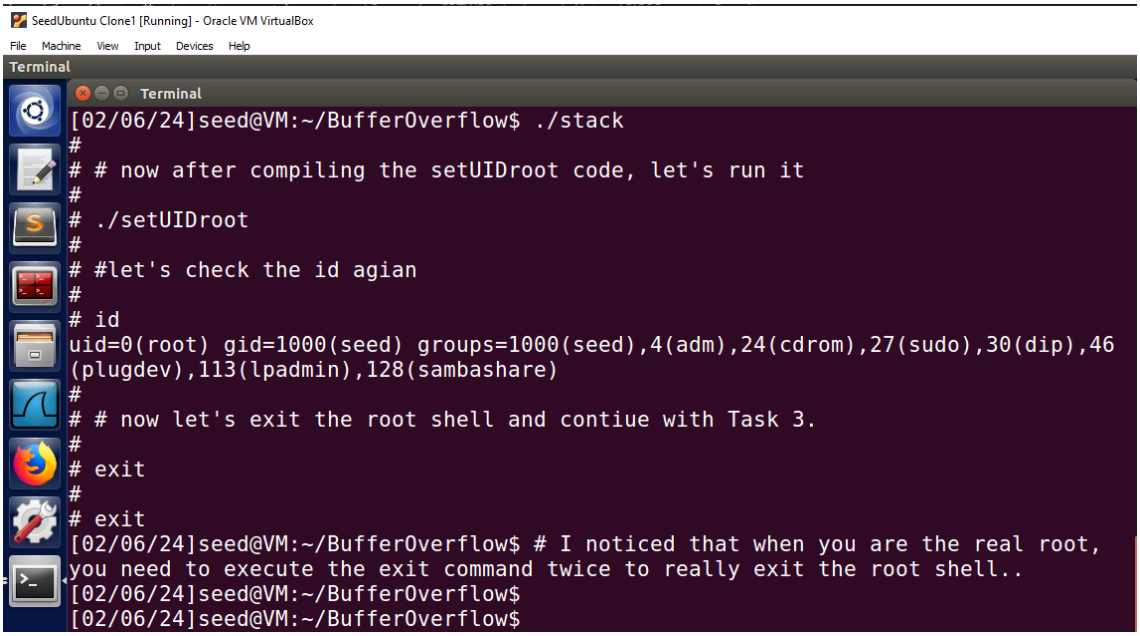




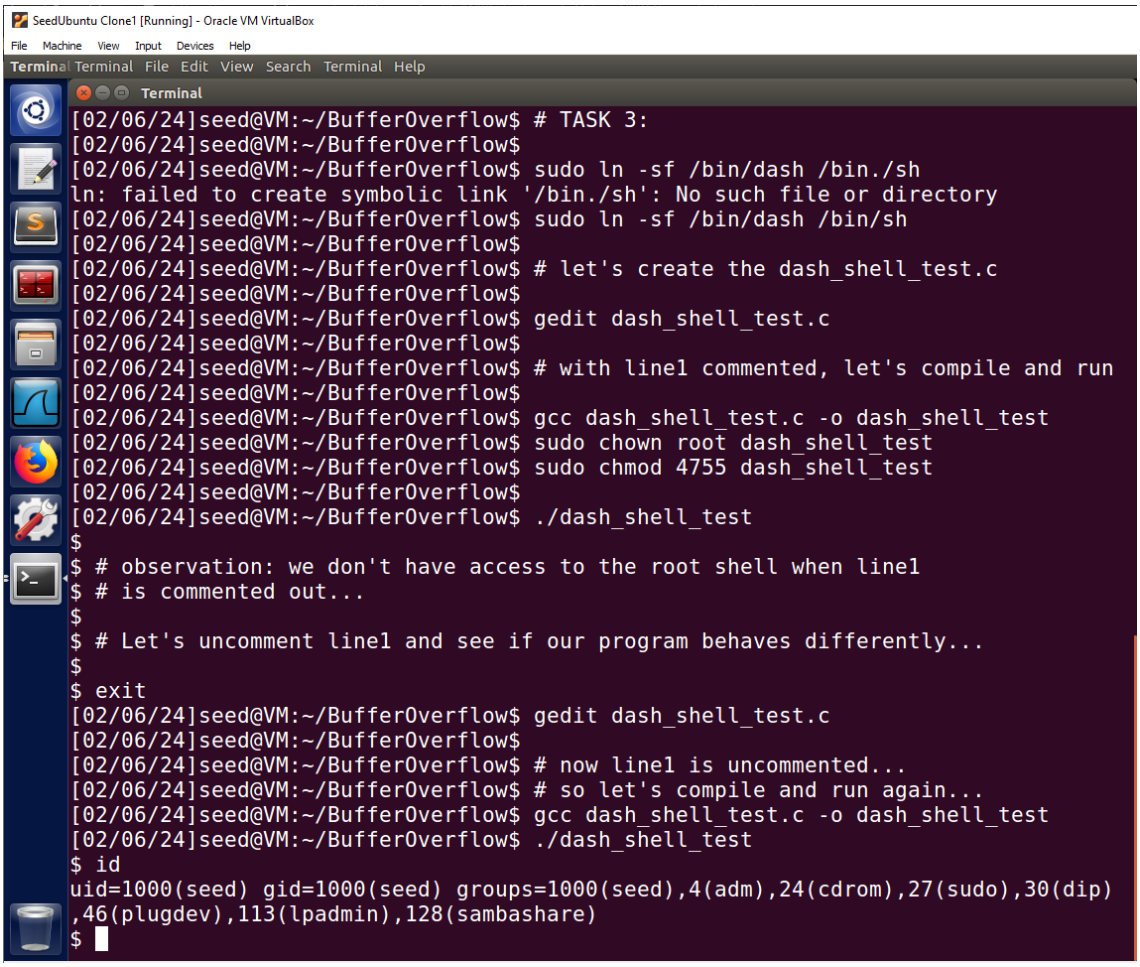
- The next three (3) images below gives a detailed description and execution of the exploit after editing [exploit.c] to work with our offset of 36 bytes.

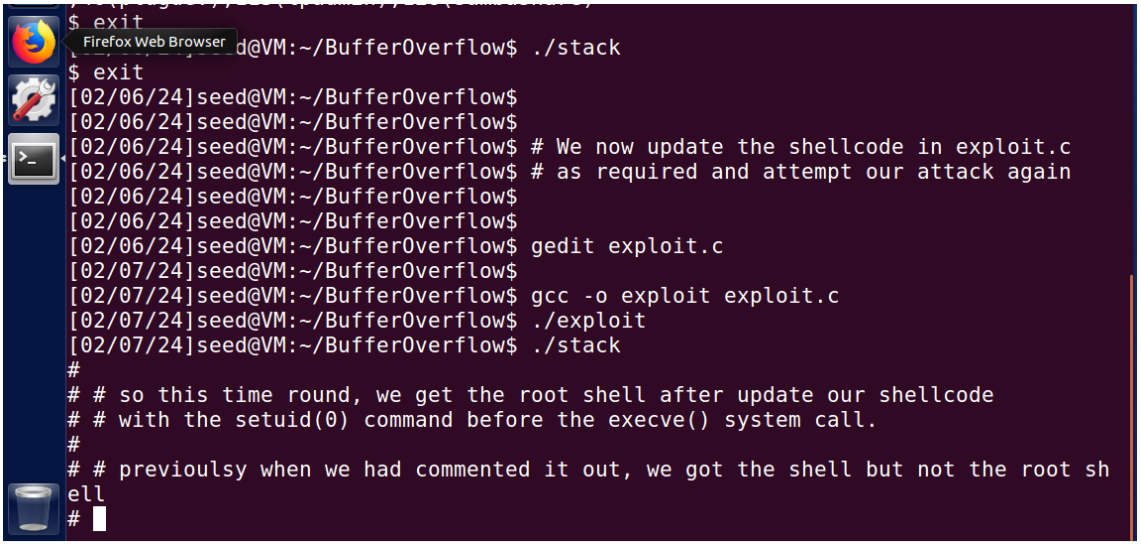




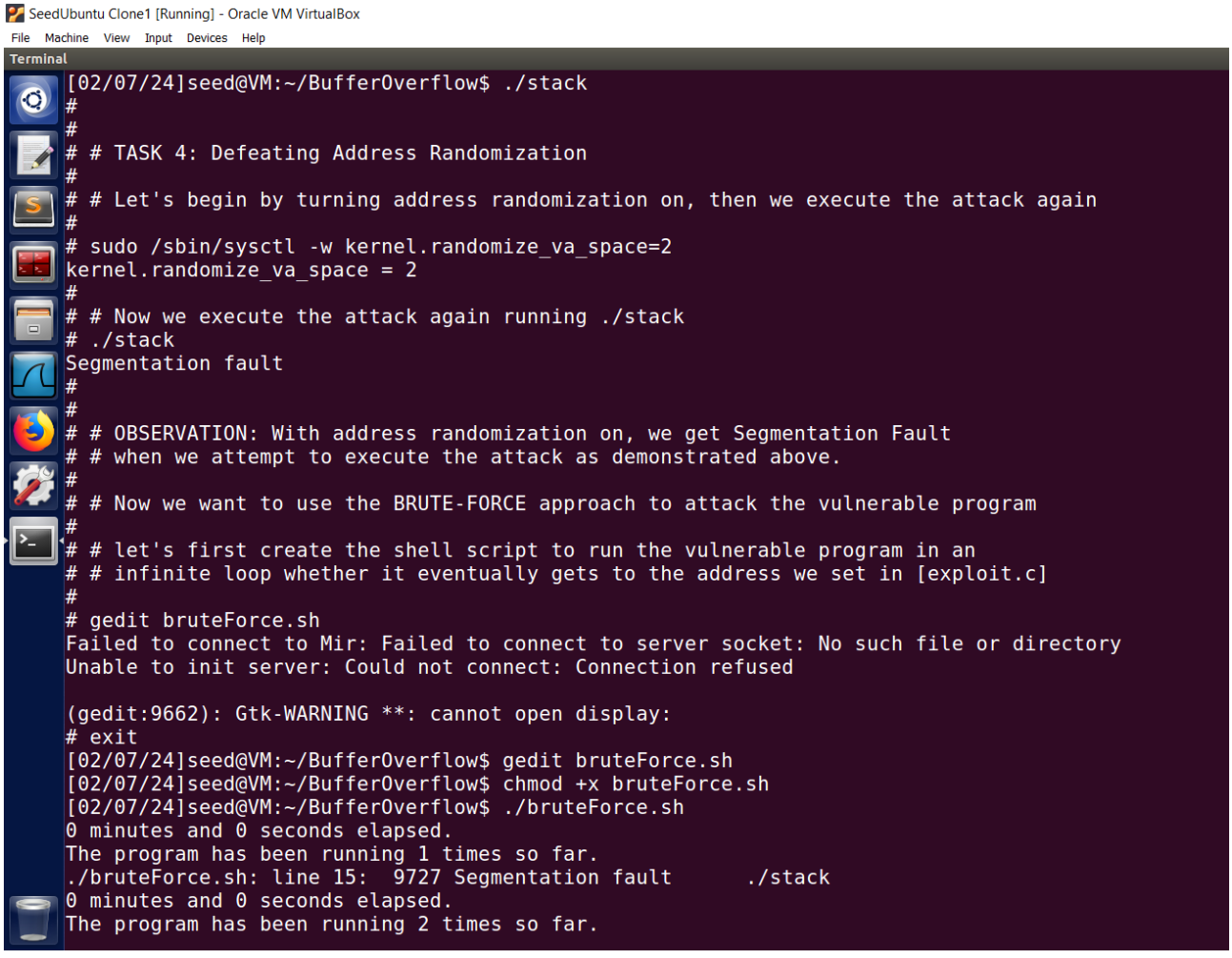


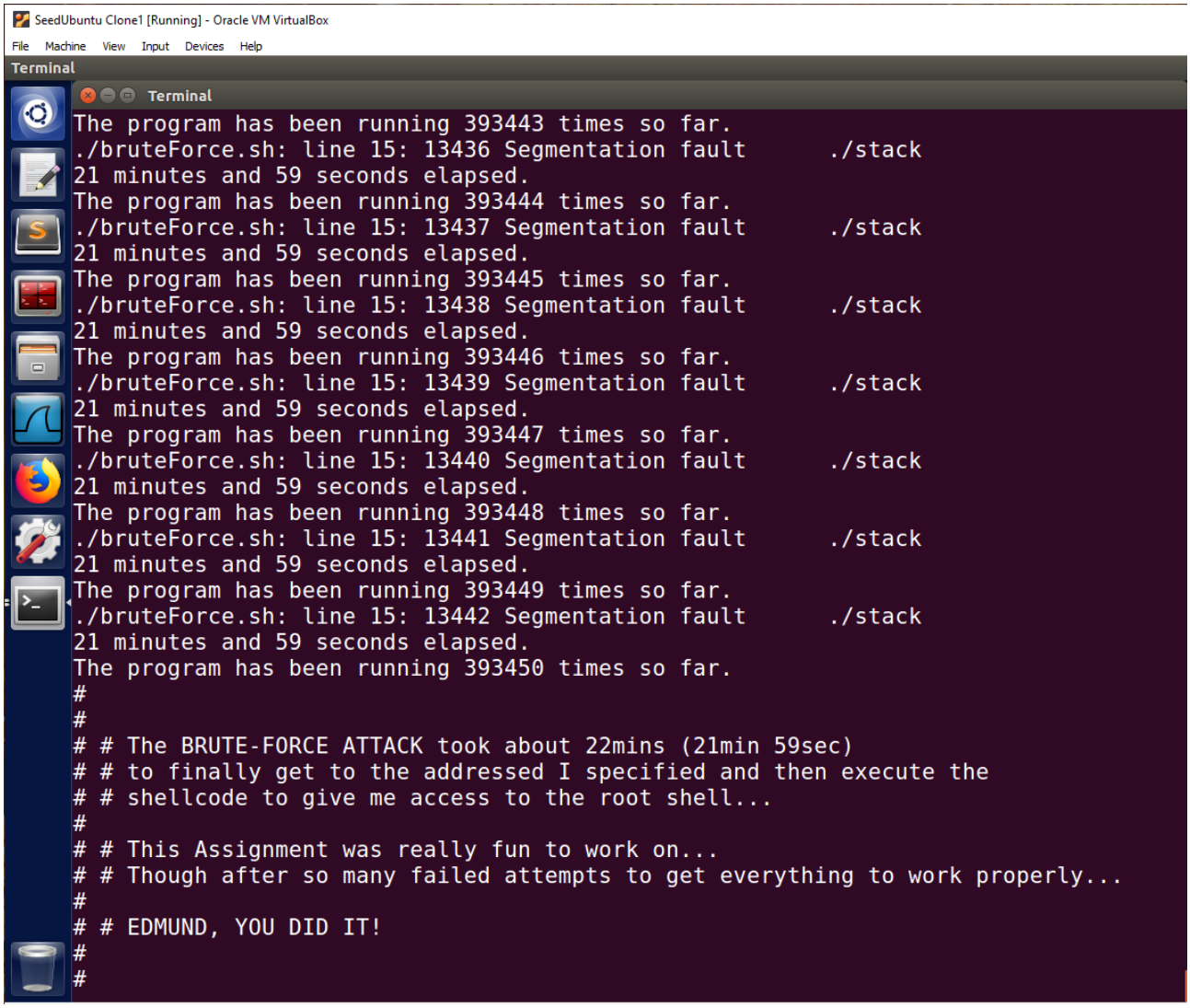
- Now we proceed to TASK 3, where we defeat dash's countermeasure. We do this by invoking setuid(0) before executing execve() system call. My observations are provided as comments in the terminal window (see the next 2 images) as shown below.





- Finally, we work on task 4. Here, the address randomization is turned on and we rather use a BRUTE-FORCE approach to execute the attack. Observations and comments are well documented in the snapshots that follow.




SeedUbuntu Clone1 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
Terminal
The program has been running 393443 times so far.
./bruteForce.sh: line 15: 13436 Segmentation fault ./stack
21 minutes and 59 seconds elapsed.
The program has been running 393444 times so far.
./bruteForce.sh: line 15: 13437 Segmentation fault ./stack
21 minutes and 59 seconds elapsed.
The program has been running 393445 times so far.
./bruteForce.sh: line 15: 13438 Segmentation fault ./stack
21 minutes and 59 seconds elapsed.
The program has been running 393446 times so far.
./bruteForce.sh: line 15: 13439 Segmentation fault ./stack
21 minutes and 59 seconds elapsed.
The program has been running 393447 times so far.
./bruteForce.sh: line 15: 13440 Segmentation fault ./stack
21 minutes and 59 seconds elapsed.
The program has been running 393448 times so far.
./bruteForce.sh: line 15: 13441 Segmentation fault ./stack
21 minutes and 59 seconds elapsed.
The program has been running 393449 times so far.
./bruteForce.sh: line 15: 13442 Segmentation fault ./stack
21 minutes and 59 seconds elapsed.
The program has been running 393450 times so far.
#
#
The BRUTE-FORCE ATTACK took about 22mins (21min 59sec)
to finally get to the addressed I specified and then execute the
shellcode to give me access to the root shell...
#
This Assignment was really fun to work on...
Though after so many failed attempts to get everything to work properly...
#
EDMUND, YOU DID IT!
#
#