Lab2-Sample Shellcode Development Lab

In previous years, I installed the Virtual Machine (VM) and downloaded the *SEED-Ubuntu20.04.zip* from the SEED website on my personal computer. Then I had to extract this file before I upload it to the virtual machine. After I completed and ran this virtual machine, it been normal and there was no crash at this time. I had to create a password and username on this Ubuntu virtual machine **[Figure 1: Ubuntu Home Screen].**

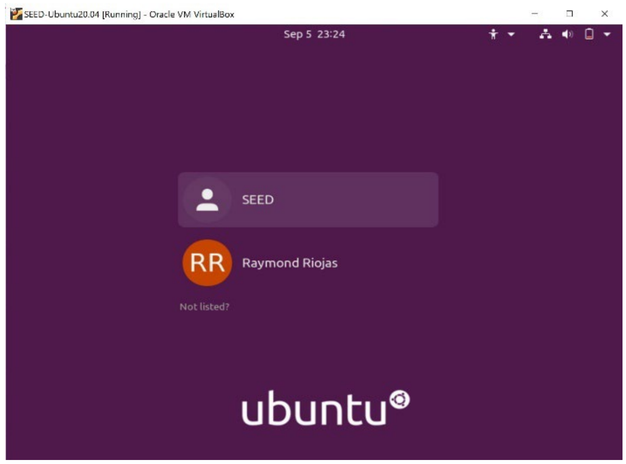


Figure 1: Ubuntu Home Screen

Once I login into this username and password, I had to download the *labsetup.zip* from the Shellcode lab site then I extracted this file into this virtual machine instead personal device **[Figure 2: Labsetup File]**.

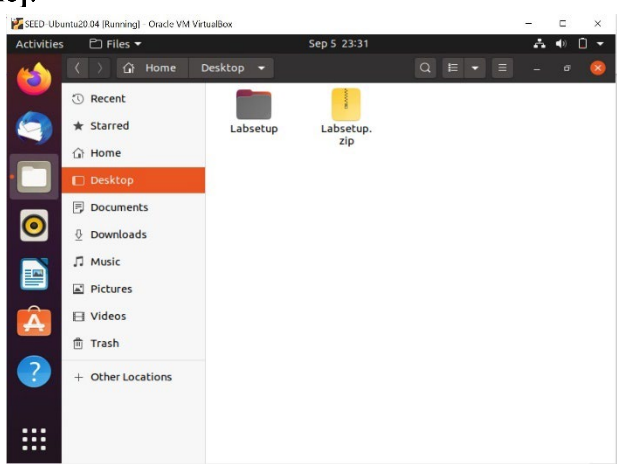


Figure 2: Labsetup File

In this lab, we're going to discuss what shellcode is in the real-world scenario. Shellcode is part of the injection and exploit vulnerabilities in a system. I had to write the code for compiling to object code using *nasm* command. The code is *nasm -f elf32 hello.s -o hello.o* on the terminal before everything is ready. Once I get the object code *hello.o*, that way we can run the linker program which is means that the last step in compilation, so the code is *ld hello.o -o hello*. After linker program, we had to test this file which I typed this code is *./hello* then it say "Hello, world!" **[Figure 3: Hello World Output]**.

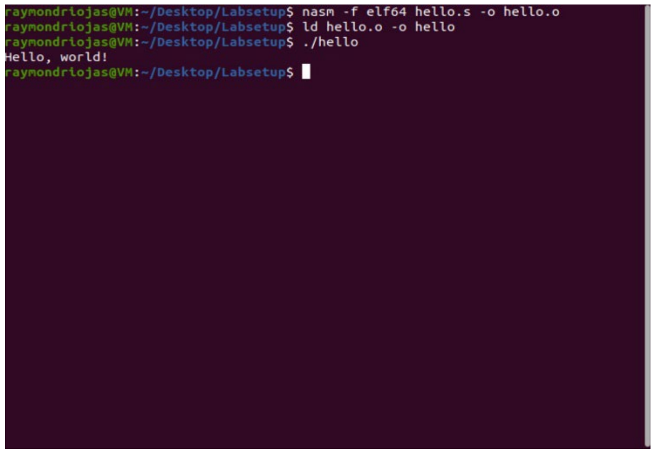


Figure 3: Hello World Output

Once I completed the last step, I had to use the *objdump* command to disassemble the executable or object file. On the VM, I use the code is *objdump -Mintel -d hello.o* then I had to see this hexadecimal numbers on this output **[Figure 4: Hexadecimal]**.

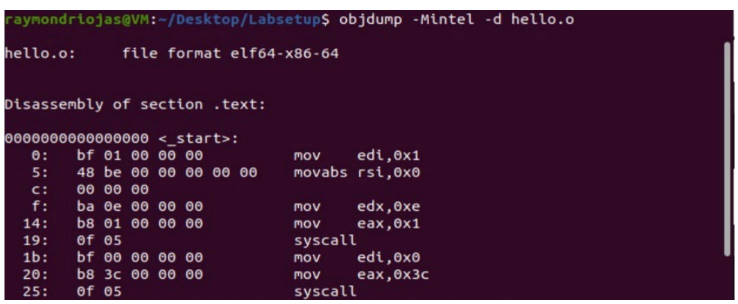


Figure 4: Hexadecimal

Next step, I had to use *xxd* command which means that print out the content of the binary file. Also, I should be able to find any hexadecimal from other output **[Figure 5: xxd Command]**.

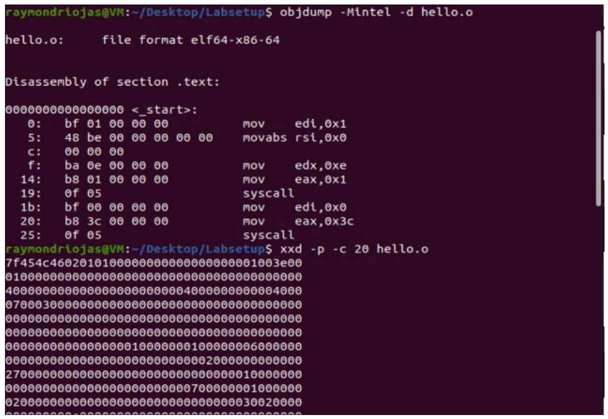


Figure 5: xxd Command

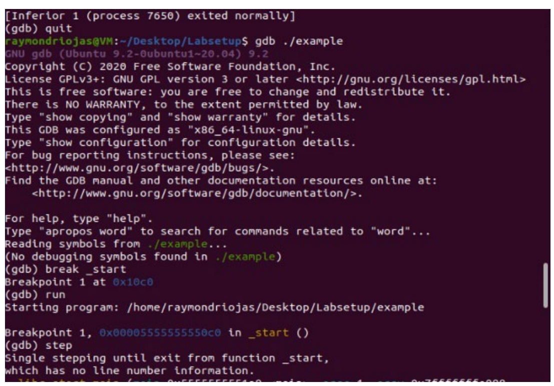
In actual attacks, we had to find the shellcode in our attacking code, which may be written in a language like Python. This python code is going to the terminal, the code is *./converrt.py.* I looked at this output code on terminal, but it said there are 15 from the length of the shellcode **[Figure 6: Attack code]**.

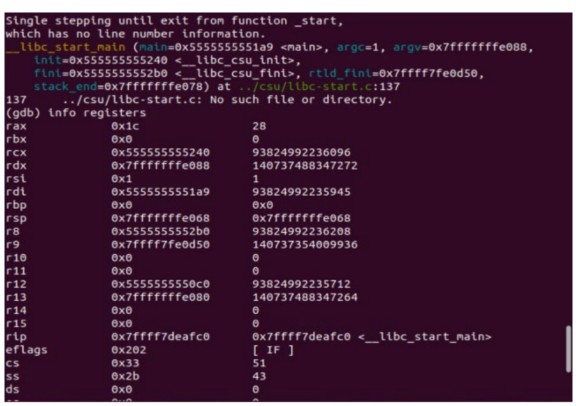
Figure 6: Attack code

Task 2: Writing Shellcode (Approach 1)

I had to create the code for execve() before I uploaded it to the terminal. When I type the command *touch* which means that I can open the program. I have to store three different ways are string, array, address **[Figure 7: Execve()].** I had to compile and run the code with enables the debugging information. There is no error on this code, before I type this command *gdb* which means that to debug the program and show how the program gets the address of the shell string **[Figure 8: gdb commands].**

Figure 7: Execve()





```
rsp            0x7fffffffe068      0x7fffffffe068
r8             0x5555555552b0      93824992236208
r9             0x7ffff7fe0d50      140737354009936
r10            0x0                 0
r11            0x0                 0
r12            0x5555555550c0      93824992235712
r13            0x7fffffffe080      140737488347264
r14            0x0                 0
r15            0x0                 0
rip            0x7ffff7deafc0      0x7ffff7deafc0 <__libc_start_main>
eflags         0x202               [ IF ]
cs             0x33                51
ss             0x2b                43
ds             0x0                 0
es             0x0                 0
fs             0x0                 0
gs             0x0                 0
(gdb) x/16bx 0x5555555551a9
0x5555555551a9 <main>:  0xf3    0x0f    0x1e    0xfa    0x55    0x48    0x89    0
xe5
0x5555555551b1 <main+8>:        0x48    0x83    0xec    0x30    0x64    0x48    0
x8b     0x04
```

Figure 8:gdb commands

I must see different options in *gdb* commands including *break _start*, *run*, *info registers*, and *x/16bx 0x55….1a9.*

Summary

On the first quiz or laboratory, ………………………………