

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].

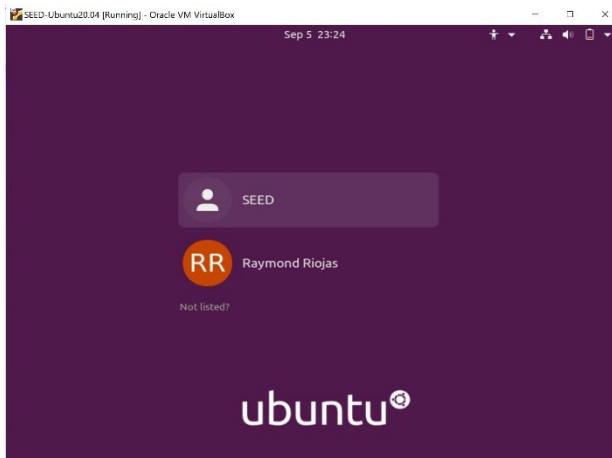


Figure 1: Ubuntu Home Screen

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].

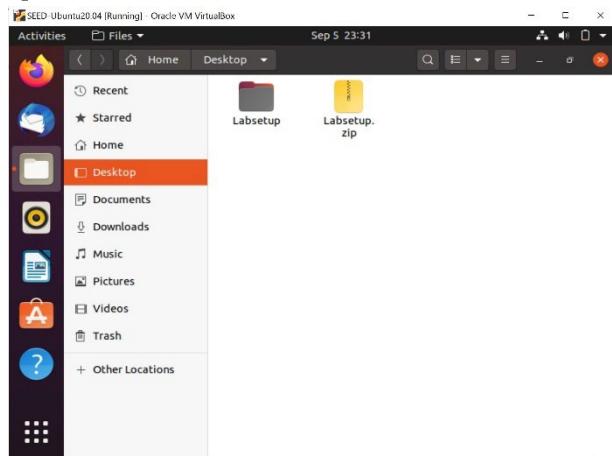


Figure 2: Labsetup File

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 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].



```
raymondriojas@VM:~/Desktop/Labsetup$ nasm -f elf64 hello.s -o hello.o
raymondriojas@VM:~/Desktop/Labsetup$ ld hello.o -o hello
raymondriojas@VM:~/Desktop/Labsetup$ ./hello
Hello, world!
raymondriojas@VM:~/Desktop/Labsetup$
```

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].



```
raymondriojas@VM:~/Desktop/Labsetup$ objdump -Mintel -d hello.o
hello.o:      file format elf64-x86-64

Disassembly of section .text:
0000000000000000 <_start>:
 0: bf 01 00 00 00          mov    edi,0x1
 5: 48 be 00 00 00 00       movabs rsi,0x0
 c: 00 00 00
 f: ba 0e 00 00 00          mov    edx,0xe
14: b8 01 00 00 00          mov    eax,0x1
19: 0f 05                  syscall
1b: bf 00 00 00 00          mov    edi,0x0
20: b8 3c 00 00 00          mov    eax,0x3c
25: 0f 05                  syscall
```

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

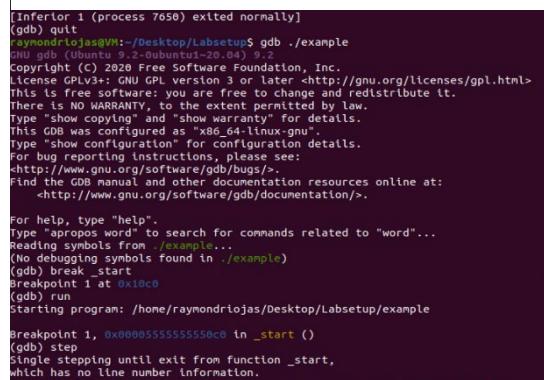
Task 2: Writing Shellcode (Approach 1)

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].



```
mysha64.c    main.c    example.c    shellExec.c
1 #include <stdio.h>
2 #include <unistd.h>
3 #include <stdlib.h>
4
5 int main()
6 {
7     char *bin_sh = "/bin/sh";
8     char *argv[] = { "bin/sh", NULL };
9     char *envp[] = { NULL };
10
11     if(execve(bin_sh, argv, envp) == -1)
12     {
13         perror("execve");
14         exit(EXIT_FAILURE);
15     }
16
17     return 0;
18 }
```

Figure 7: Execve()



```
[Inferior 1 (process 7650) exited normally]
(gdb) quit
$ raymondriojas@Vim:~/Desktop/Labsetup$ gdb ./example
GNU gdb (Ubuntu 9.2-0ubuntu1~20.04)
Copyright (C) 2020 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 "x86_64-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 ./example...
(No debugging symbols found in ./example)
(gdb) break _start
Breakpoint 1 at 0x10c0
(gdb) run
Starting program: /home/raymondriojas/Desktop/Labsetup/example

Breakpoint 1, 0x00005555555550c0 in _start ()
(gdb) step
Single stepping until exit from function _start,
which has no line number information.
```

<img alt="Screenshot of a terminal window showing the registers of the program during debugging. It lists various registers with their addresses and values, such as rax, rbx, rcx, rdx, rsi, rdi, rbp, rsp, r8, r9, r10, r11, r12, r13, r14, r15, r16, r17, r18, r19, r20, r21, r22, r23, r24, r25, r26, r27, r28, r29, r30, r31, r32, r33, r34, r35, r36, r37, r38, r39, r40, r41, r42, r43, r44, r45, r46, r47, r48, r49, r50, r51, r52, r53, r54, r55, r56, r57, r58, r59, r60, r61, r62, r63, r64, r65, r66, r67, r68, r69, r70, r71, r72, r73, r74, r75, r76, r77, r78, r79, r80, r81, r82, r83, r84, r85, r86, r87, r88, r89, r90, r91, r92, r93, r94, r95, r96, r97, r98, r99, r100, r101, r102, r103, r104, r105, r106, r107, r108, r109, r110, r111, r112, r113, r114, r115, r116, r117, r118, r119, r120, r121, r122, r123, r124, r125, r126, r127, r128, r129, r130, r131, r132, r133, r134, r135, r136, r137, r138, r139, r140, r141, r142, r143, r144, r145, r146, r147, r148, r149, r150, r151, r152, r153, r154, r155, r156, r157, r158, r159, r160, r161, r162, r163, r164, r165, r166, r167, r168, r169, r170, r171, r172, r173, r174, r175, r176, r177, r178, r179, r180, r181, r182, r183, r184, r185, r186, r187, r188, r189, r190, r191, r192, r193, r194, r195, r196, r197, r198, r199, r200, r201, r202, r203, r204, r205, r206, r207, r208, r209, r210, r211, r212, r213, r214, r215, r216, r217, r218, r219, r220, r221, r222, r223, r224, r225, r226, r227, r228, r229, r230, r231, r232, r233, r234, r235, r236, r237, r238, r239, r240, r241, r242, r243, r244, r245, r246, r247, r248, r249, r250, r251, r252, r253, r254, r255, r256, r257, r258, r259, r260, r261, r262, r263, r264, r265, r266, r267, r268, r269, r270, r271, r272, r273, r274, r275, r276, r277, r278, r279, r280, r281, r282, r283, r284, r285, r286, r287, r288, r289, r290, r291, r292, r293, r294, r295, r296, r297, r298, r299, r290, r291, r292, r293, r294, r295, r296, r297, r298, r299, r300, r301, r302, r303, r304, r305, r306, r307, r308, r309, r3010, r3011, r3012, r3013, r3014, r3015, r3016, r3017, r3018, r3019, r3020, r3021, r3022, r3023, r3024, r3025, r3026, r3027, r3028, r3029, r3030, r3031, r3032, r3033, r3034, r3035, r3036, r3037, r3038, r3039, r30310, r30311, r30312, r30313, r30314, r30315, r30316, r30317, r30318, r30319, r30320, r30321, r30322, r30323, r30324, r30325, r30326, r30327, r30328, r30329, r30330, r30331, r30332, r30333, r30334, r30335, r30336, r30337, r30338, r30339, r30340, r30341, r30342, r30343, r30344, r30345, r30346, r30347, r30348, r30349, r30350, r30351, r30352, r30353, r30354, r30355, r30356, r30357, r30358, r30359, r30360, r30361, r30362, r30363, r30364, r30365, r30366, r30367, r30368, r30369, r30370, r30371, r30372, r30373, r30374, r30375, r30376, r30377, r30378, r30379, r30380, r30381, r30382, r30383, r30384, r30385, r30386, r30387, r30388, r30389, r30390, r30391, r30392, r30393, r30394, r30395, r30396, r30397, r30398, r30399, r303100, r303110, r303120, r303130, r303140, r303150, r303160, r303170, r303180, r303190, r303200, r303210, r303220, r303230, r303240, r303250, r303260, r303270, r303280, r303290, r303200, r303210, r303220, r303230, r303240, r303250, r303260, r303270, r303280, r303290, r303300, r303310, r303320, r303330, r303340, r303350, r303360, r303370, r303380, r303390, r303300, r303310, r303320, r303330, r303340, r303350, r303360, r303370, r303380, r303390, r303400, r303410, r303420, r303430, r303440, r303450, r303460, r303470, r303480, r303490, r303400, r303410, r303420, r303430, r303440, r303450, r303460, r303470, r303480, r303490, r303500, r303510, r303520, r303530, r303540, r303550, r303560, r303570, r303580, r303590, r303500, r303510, r303520, r303530, r303540, r303550, r303560, r303570, r303580, r303590, r303600, r303610, r303620, r303630, r303640, r303650, r303660, r303670, r303680, r303690, r303600, r303610, r303620, r303630, r303640, r303650, r303660, r303670, r303680, r303690, r303700, r303710, r303720, r303730, r303740, r303750, r303760, r303770, r303780, r303790, r303700, r303710, r303720, r303730, r303740, r303750, r303760, r303770, r303780, r303790, r303800, r303810, r303820, r303830, r303840, r303850, r303860, r303870, r303880, r303890, r303800, r303810, r303820, r303830, r303840, r303850, r303860, r303870, r303880, r303890, r303900, r303910, r303920, r303930, r303940, r303950, r303960, r303970, r303980, r303990, r303900, r303910, r303920, r303930, r303940, r303950, r303960, r303970, r303980, r303990, r303100, r303110, r303120, r303130, r303140, r303150, r303160, r303170, r303180, r303190, r303200, r303210, r303220, r303230, r303240, r303250, r303260, r303270, r303280, r303290, r303300, r303310, r303320, r303330, r303340, r303350, r303360, r303370, r303380, r303390, r303400, r303410, r303420, r303430, r303440, r303450, r303460, r303470, r303480, r303490, r303500, r303510, r303520, r303530, r303540, r303550, r303560, r303570, r303580, r303590, r303600, r303610, r303620, r303630, r303640, r303650, r303660, r303670, r303680, r303690, r303700, r303710, r303720, r303730, r303740, r303750, r303760, r303770, r303780, r303790, r303800, r303810, r303820, r303830, r303840, r303850, r303860, r303870, r303880, r303890, r303900, r303910, r303920, r303930, r303940, r303950, r303960, r303970, r303980, r303990, r303100, r303110, r303120, r303130, r303140, r303150, r303160, r303170, r303180, r303190, r303200, r303210, r303220, r303230, r303240, r303250, r303260, r303270, r303280, r303290, r303300, r303310, r303320, r303330, r303340, r303350, r303360, r303370, r303380, r303390, r303400, r303410, r303420, r303430, r303440, r303450, r303460, r303470, r303480, r303490, r303500, r303510, r303520, r303530, r303540, r303550, r303560, r303570, r303580, r303590, r303600, r303610, r303620, r303630, r303640, r303650, r303660, r303670, r303680, r303690, r303700, r303710, r303720, r303730, r303740, r303750, r303760, r303770, r303780, r303790, r303800, r303810, r303820, r303830, r303840, r303850, r303860, r303870, r303880, r303890, r303900, r303910, r303920, r303930, r303940, r303950, r303960, r303970, r303980, r303990, r303100, r303110, r303120, r303130, r303140, r303150, r303160, r303170, r303180, r303190, r303200, r303210, r303220, r303230, r303240, r303250, r303260, r303270, r303280, r303290, r303300, r303310, r303320, r303330, r303340, r303350, r303360, r303370, r303380, r303390, r303400, r303410, r303420, r303430, r303440, r303450, r303460, r303470, r303480, r303490, r303500, r303510, r303520, r303530, r303540, r303550, r303560, r303570, r303580, r303590, r303600, r303610, r303620, r303630, r303640, r303650, r303660, r303670, r303680, r303690, r303700, r303710, r303720, r303730, r303740, r303750, r303760, r303770, r303780, r303790, r303800, r303810, r303820, r303830, r303840, r303850, r303860, r303870, r303880, r303890, r303900, r303910, r303920, r303930, r303940, r303950, r303960, r303970, r303980, r303990, r303100, r303110, r303120, r303130, r303140, r303150, r303160, r303170, r303180, r303190, r303200, r303210, r303220, r303230, r303240, r303250, r303260, r303270, r303280, r303290, r303300, r303310, r303320, r303330, r303340, r303350, r303360, r303370, r303380, r303390, r303400, r303410, r303420, r303430, r303440, r303450, r303460, r303470, r303480, r303490, r303500, r303510, r303520, r303530, r303540, r303550, r303560, r303570, r303580, r303590, r303600, r303610, r303620, r303630, r303640, r303650, r303660, r303670, r303680, r303690, r303700, r303710, r303720, r303730, r303740, r303750, r303760, r303770, r303780, r303790, r303800, r303810, r303820, r303830, r303840, r303850, r303860, r303870, r303880, r303890, r303900, r303910, r303920, r303930, r303940, r303950, r303960, r303970, r303980, r303990, r303100, r303110, r303120, r303130, r303140, r303150, r303160, r303170, r303180, r303190, r303200, r303210, r303220, r303230, r303240, r303250, r303260, r303270, r303280, r303290, r303300, r303310, r303320, r303330, r303340, r303350, r303360, r303370, r303380, r303390, r303400, r303410, r303420, r303430, r303440, r303450, r303460, r303470, r303480, r303490, r303500, r303510, r303520, r303530, r303540, r303550, r303560, r303570, r303580, r303590, r303600, r303610, r303620, r303630, r303640, r303650, r303660, r303670, r303680, r303690, r303700, r303710, r303720, r303730, r303740, r303750, r303760, r303770, r303780, r303790, r303800, r303810, r303820, r303830, r303840, r303850, r303860, r303870, r303880, r303890, r303800, r303810, r303820, r303830, r303840, r303850, r303860, r303870, r303880, r303890, r303900, r303910, r303920, r303930, r303940, r303950, r303960, r303970, r303980, r303990, r303100, r303110, r303120, r303130, r303140, r303150, r303160, r303170, r303180, r303190, r303200, r303210, r303220, r303230, r303240, r303250, r303260, r303270, r303280, r303290, r303300, r303310, r303320, r303330, r303340, r303350, r303360, r303370, r303380, r303390, r303400, r303410, r303420, r303430, r303440, r303450, r303460, r303470, r303480, r303490, r303500, r303510, r303520, r303530, r303540, r303550, r303560, r303570, r303580, r303590, r303600, r303610, r303620, r303630, r303640, r303650, r303660, r303670, r303680, r303690, r303700, r303710, r303720, r303730, r303740, r303750, r303760, r303770, r303780, r303790, r303800, r303810, r303820, r303830, r303840, r303850, r303860, r303870, r303880, r303890, r303800, r303810, r303820, r303830, r303840, r303850, r303860, r303870, r303880, r303890, r303900, r303910, r303920, r303930, r303940, r303950, r303960, r303970, r303980, r303990, r303100, r303110, r303120, r303130, r303140, r303150, r303160, r303170, r303180, r303190, r303200, r303210, r303220, r303230, r303240, r303250, r303260, r303270, r303280, r303290, r303300, r303310, r303320, r303330, r303340, r303350, r303360, r303370, r303380, r303390, r303400, r303410, r303420, r303430, r303440, r303450, r303460, r303470, r303480, r303490, r303500, r303510, r303520, r303530, r303540, r303550, r303560, r303570, r303580, r303590, r303600, r303610, r303620, r303630, r303640, r303650, r303660, r303670, r303680, r303690, r303700, r303710, r303720, r303730, r303740, r303750, r303760, r303770, r303780, r303790, r303800, r303810, r303820, r303830, r303840, r303850, r303860, r303870, r303880, r303890, r303800, r303810, r303820, r303830, r303840, r303850, r303860, r303870, r303880, r303890, r303900, r303910, r303920, r303930, r303940, r303950, r303960, r303970, r303980, r303990, r303100, r303110, r303120, r303130, r303140, r303150, r303160, r303170, r303180, r303190, r303200, r303210, r303220, r303230, r303240, r303250, r303260, r303270, r303280, r303290, r303300, r303310, r303320, r303330, r303340, r303350, r303360, r303370, r303380, r303390, r303400, r303410, r303420, r303430, r303440, r303450, r303460, r303470, r303480, r303490, r303500, r303510, r303520, r303530, r303540, r303550, r303560, r303570, r303580, r303590, r303600, r303610, r303620, r303630, r303640, r303650, r303660, r303670, r303680, r303690, r303700, r303710, r303720, r303730, r303740, r303750, r303760, r303770, r303780, r303790, r303800, r303810, r303820, r303830, r303840, r303850, r303860, r303870, r303880, r303890, r303800, r303810, r303820, r303830, r303840, r303850, r303860, r303870, r303880, r303890, r303900, r303910, r303920, r303930, r303940, r303950, r303960, r303970, r303980, r303990, r303100, r303110, r303120, r303130, r303140, r303150, r303160, r303170, r303180, r303190, r303200, r303210, r303220, r303230, r303240, r303250, r303260, r303270, r303280, r303290, r303300, r303310, r303320, r303330, r303340, r303350, r303360, r303370, r303380, r303390, r303400, r303410, r303420, r303430, r303440, r303450, r303460, r303470, r303480, r303490, r303500, r303510, r303520, r303530, r303540, r303550, r303560, r303570, r303580, r303590, r303600, r303610, r303620, r303630, r303640, r303650, r303660, r303670, r303680, r303690, r303700, r303710, r303720, r303730, r303740, r303750, r303760, r303770, r303780, r303790, r303800, r303810, r303820, r303830, r303840, r303850, r303860, r303870, r303880, r303890, r303800, r3038

```

rsp      0x7fffffff068   0x7fffffff068
r8      0x555555552b0   93824992236288
r9      0x7fffff7fe0d50   140737354069936
r10     0x0      0
r11     0x0      0
r12     0x555555550c0   93824992235712
r13     0x7ffffffe080   140737488347264
r14     0x0      0
r15     0x0      0
rip     0x7ffff7deafc0   0x7ffff7deafc0 <__libc_start_main>
eflags   0x202      [ IF ]
cs      0x33      51
ss      0xb       43
ds      0x0      0
es      0x0      0
fs      0x0      0
gs      0x0      0
(gdb) x/16bx 0x555555551a9
0x555555551a9 <Main>: 0xf3    0x0f    0x1e    0xfa    0x55    0x48    0x89    0
xe5
0x555555551b1 <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,