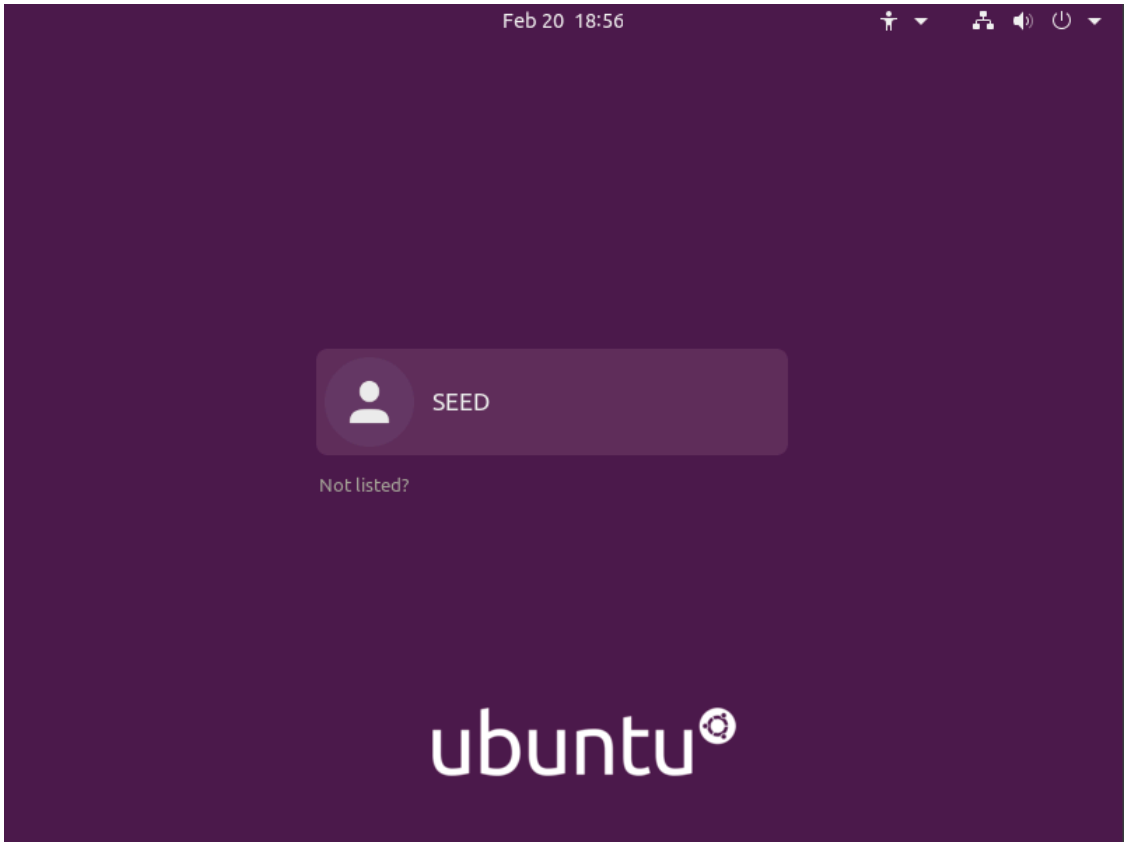
Lab4 Submission: Morris Worm

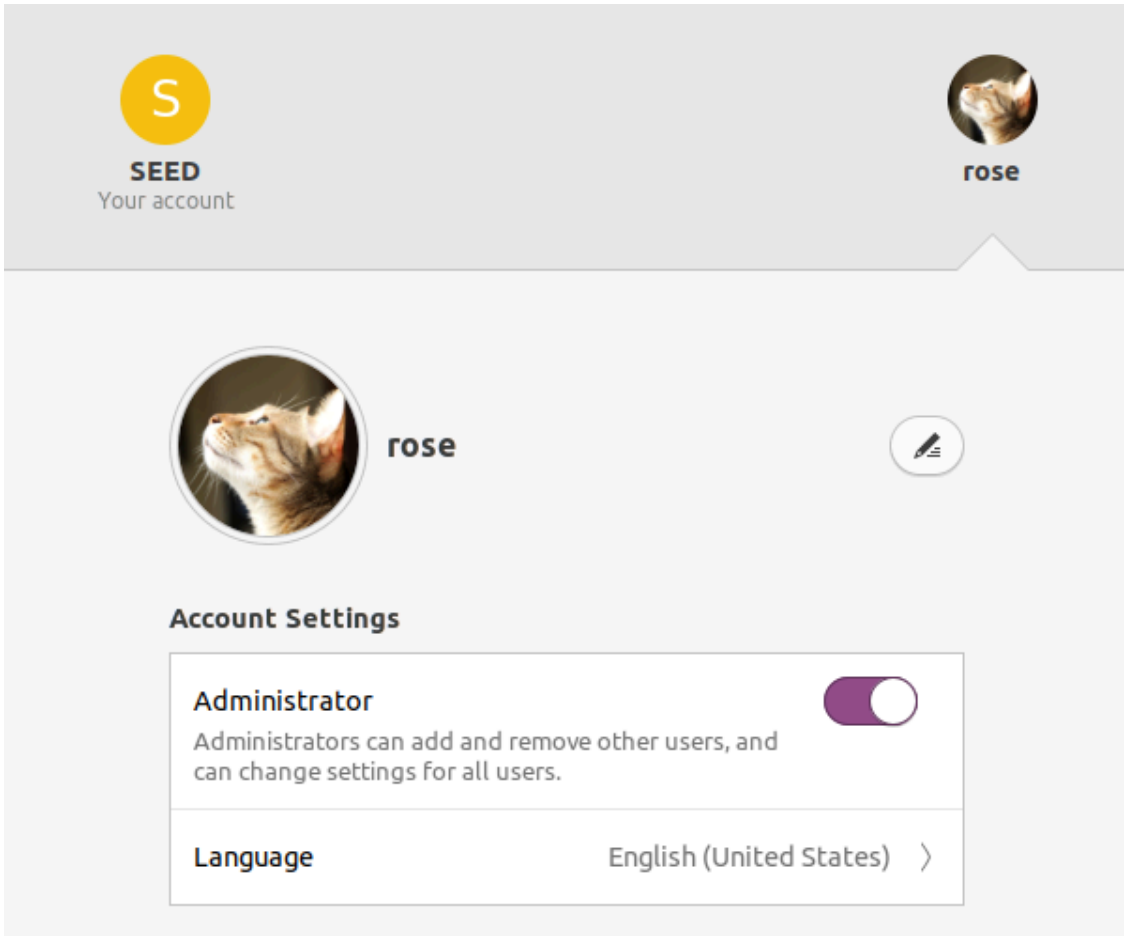
Samantha Jackson
CSCI 4321
Computer Security
February 20, 2025

SEED 20.04 Installation

Luckily going into this lab I just got a new computer last week so I'm hopeful that this time the VBox configurations will work. First of all, I had to enable CPU Virtualization in UEFI Firmware settings as it was not enabled by default. You can find that option under SVM Mode on an AMD CPU. I believe my problem with lab two may have simply been a problem with the CPU, but I had Intel Virtualization enabled on that laptop. This time SEED boots to the lock screen successfully, so I can proceed to the lab.

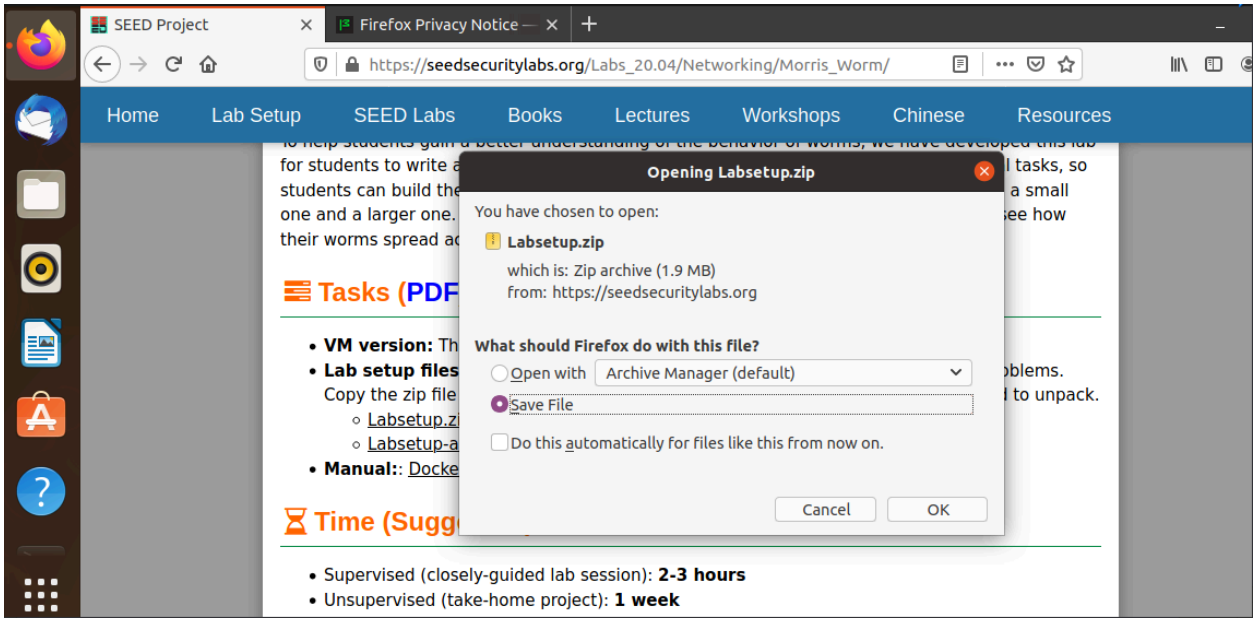


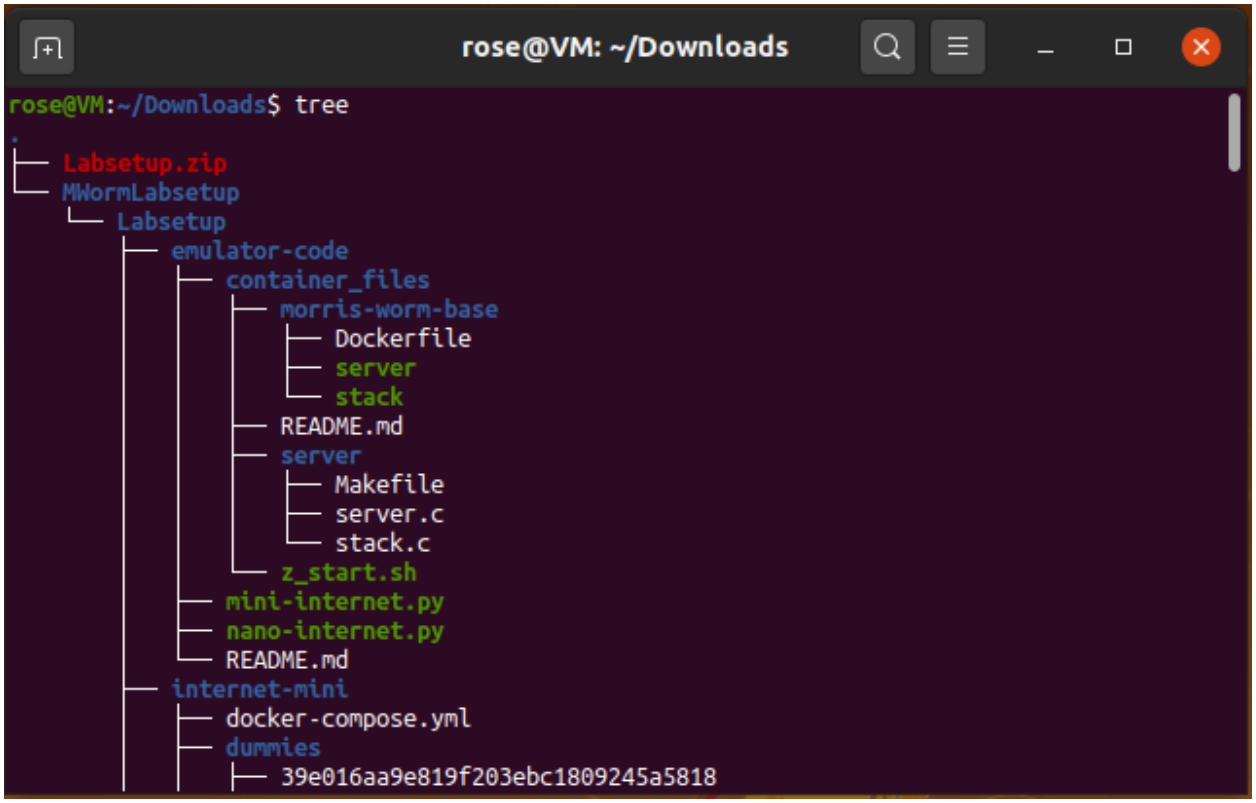
Before going into the lab I proceed to make a user: rose password: #5b47cf!. I selected a profile picture as well for personalization and turned on administrative permissions.



internet-nano Dockers

I boot into my new user and start by starting a docker in internet-nano. Firstly, I download the lab files from the seedsecuritylabs website and mkdir ‘MWormLabsetup’ then I use the unzip command to unzip the Labsetup.zip file in MWormLabsetup.





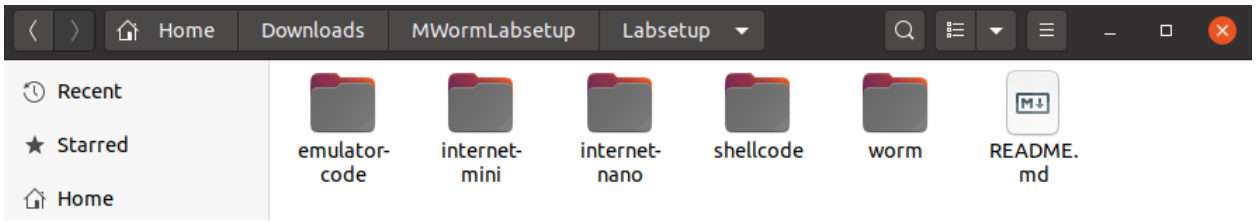
Then I move to the internet-nano dir and use ‘sudo docker-compose build’. Then I utilize ‘sudo docker-compose up’ to start the docker. For whatever reason after this step I don’t know where the map folder is, so I’m tinkering. Eventually I decide to open localhost:8080/map.html to see if it opens.

```
rose@VM:~/Downloads$ cd MWormLabsetup
rose@VM:~/Downloads/MWormLabsetup$ cd Labsetup
rose@VM:~/Downloads/MWormLabsetup/Labsetup$ cd internet-nano
rose@VM:~/Downloads/MWormLabsetup/Labsetup/internet-nano$ sudo docker-compose build
seedemu-internet-client uses an image, skipping
Building morris-worm-base
Step 1/7 : FROM handsonsecurity/seedemu-multiarch-base:buildx-latest
buildx-latest: Pulling from handsonsecurity/seedemu-multiarch-base
96d54c3075c9: Pulling fs layer
96d54c3075c9: Downloading [>                                                  ]  278.5kB
/27.51MB78df: Download complete
96d54c3075c9: Downloading [==>                                                ]  1.405MB
/27.51MB5bdd: Downloading [>                                                  ]  540.7kB
96d54c3075c9: Downloading [=======>                                           ]  3.932MB
/27.51MB5bdd: Downloading [=>                                                 ]  3.224MB
96d54c3075c9: Downloading [==========>                                        ]  5.636MB
/27.51MB5bdd: Downloading [==>                                                ]  6.992MB
/136.7MB
96d54c3075c9: Downloading [==============>                                    ]  8.176MB
96d54c3075c9: Pull complete
b71b0f5178df: Pull complete
d61d24945bdd: Pull complete
19b77ae2a0e3: Pull complete
Digest: sha256:9d65ed5afa7ba3e29607e517a217564958ec4765c9ec813b63d092c1b2bad0fc
```

```
Successfully built f673ddf71686
Successfully tagged internet-nano_rs_ix_ix100:latest
rose@VM:~/Downloads/MWormLabsetup/Labsetup/internet-nano$
```

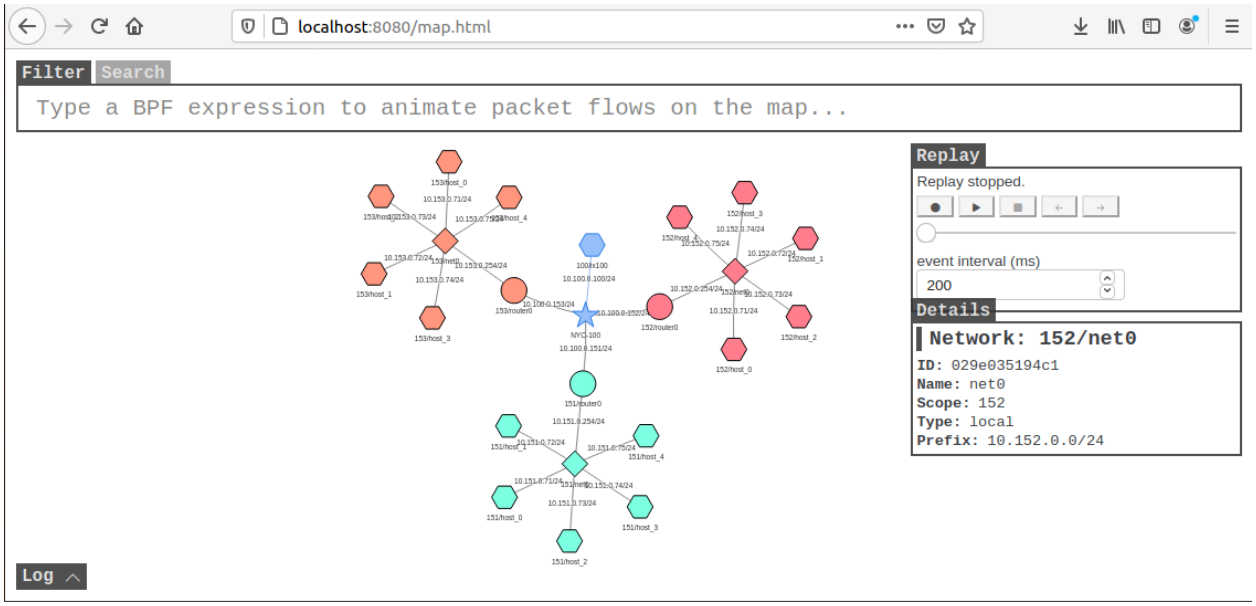
```
rose@VM:~/Downloads/MWormLabsetup/Labsetup/internet-nano$ sudo docker-compose up
Creating network "internet-nano_default" with the default driver
Creating network "internet-nano_net_151_net0" with the default driver
Creating network "internet-nano_net_ix_ix100" with the default driver
Creating network "internet-nano_net_152_net0" with the default driver
Creating network "internet-nano_net_153_net0" with the default driver
Pulling seedemu-internet-client (handsonsecurity/seedemu-multiarch-map:buildx-latest)...
buildx-latest: Pulling from handsonsecurity/seedemu-multiarch-map
2ff1d7c41c74: Pull complete
b253aeafeaa7: Pull complete
3d2201bd995c: Pull complete
1de76e268b10: Pull complete
d9a8df589451: Pull complete
6f51ee005dea: Pull complete
5f32ed3c3f27: Pull complete
0c8cc2f24a4d: Pull complete
0d27a8e86132: Pull complete
abe80f076312: Pull complete
6908e862c0a6: Pull complete
1de918b5ac72: Pull complete
4f4fb700ef54: Pull complete
e8636bf23cfa: Pull complete
0a728f919fbb: Pull complete
```

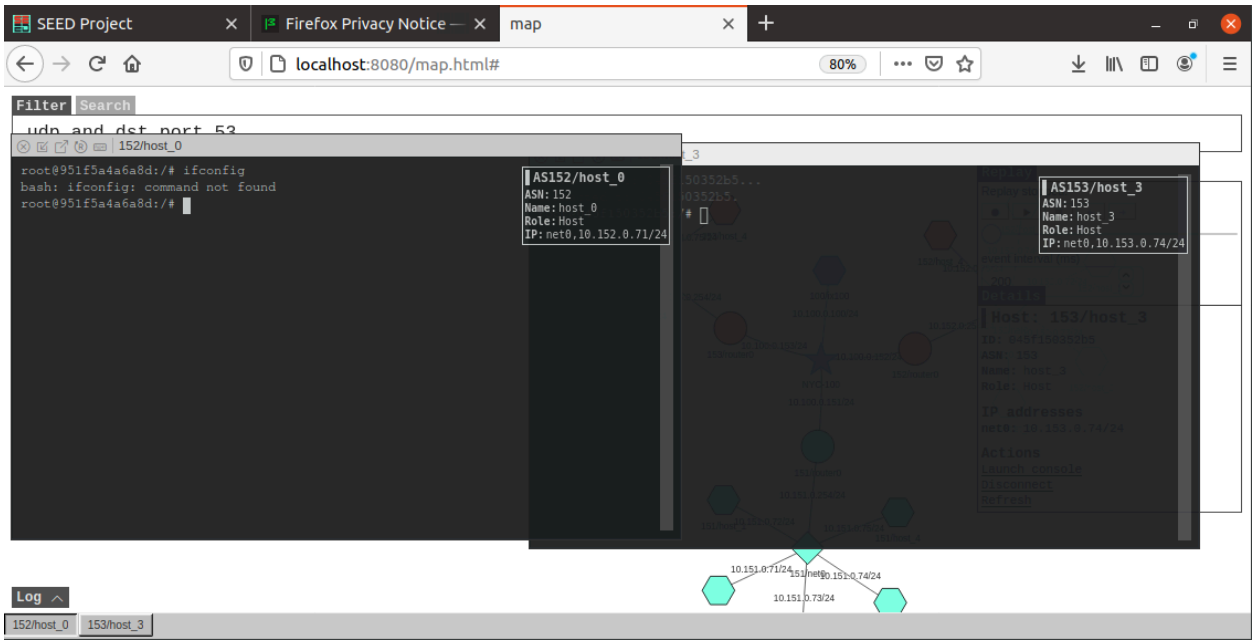
```
1, as100rs-ix100-10.100.0.100, internet-nano_ee6b6326cce7e5be4913cbfc86f3c820_1, as153r-router0-10.153.0.254, as152r
-router0-10.152.0.254, as151r-router0-10.151.0.254, as153h-host_3-10.153.0.74, as152h-host_1-10.152.0.72, as153h-hos
t_0-10.153.0.71, as151h-host_1-10.151.0.72, as152h-host_4-10.152.0.75, as153h-host_1-10.153.0.72, as152h-host_2-10.1
52.0.73, as153h-host_2-10.153.0.73, as151h-host_2-10.151.0.73, as151h-host_3-10.151.0.74, as152h-host_3-10.152.0.74,
 as152h-host_0-10.152.0.71, as153h-host_4-10.153.0.75, as151h-host_0-10.151.0.71, as151h-host_4-10.151.0.75
as100rs-ix100-10.100.0.100          | ready! run 'docker exec -it 48513f48394a /bin/zsh' to attach to this node
as100rs-ix100-10.100.0.100          | bird: Started
as151h-host_1-10.151.0.72           | ready! run 'docker exec -it 0636802a8313 /bin/zsh' to attach to this node
as151r-router0-10.151.0.254         | ready! run 'docker exec -it d9dc3efac098 /bin/zsh' to attach to this node
as151r-router0-10.151.0.254         | bird: Started
as152h-host_1-10.152.0.72           | ready! run 'docker exec -it fcd9466e925b /bin/zsh' to attach to this node
as152h-host_2-10.152.0.73           | ready! run 'docker exec -it 743a97ea7b07 /bin/zsh' to attach to this node
as152h-host_4-10.152.0.75           | ready! run 'docker exec -it 3e3b22a92f51 /bin/zsh' to attach to this node
as152r-router0-10.152.0.254         | ready! run 'docker exec -it f37636476f18 /bin/zsh' to attach to this node
as152r-router0-10.152.0.254         | bird: Started
as153h-host_0-10.153.0.71           | ready! run 'docker exec -it ce0fc8c8f635 /bin/zsh' to attach to this node
as153h-host_1-10.153.0.72           | ready! run 'docker exec -it 3072e46af604 /bin/zsh' to attach to this node
as153h-host_3-10.153.0.74           | ready! run 'docker exec -it 045f150352b5 /bin/zsh' to attach to this node
as153r-router0-10.153.0.254         | ready! run 'docker exec -it 4ab6e2bb6505 /bin/zsh' to attach to this node
as153r-router0-10.153.0.254         | bird: Started
internet-nano_ee6b6326cce7e5be4913cbfc86f3c820_1 exited with code 0
internet-nano_morris-worm-base_1 exited with code 0
internet-nano_39e016aa9e819f203ebc1809245a5818_1 exited with code 0
as153h-host_2-10.153.0.73           | ready! run 'docker exec -it 27dbd8c14b37 /bin/zsh' to attach to this node
as151h-host_3-10.151.0.74           | ready! run 'docker exec -it 5da380449021 /bin/zsh' to attach to this node
as151h-host_2-10.151.0.73           | ready! run 'docker exec -it b3096e29b422 /bin/zsh' to attach to this node
as152h-host_0-10.152.0.71           | ready! run 'docker exec -it 951f5a4a6a8d /bin/zsh' to attach to this node
as152h-host_3-10.152.0.74           | ready! run 'docker exec -it 962dae6c1548 /bin/zsh' to attach to this node
as153h-host_4-10.153.0.75           | ready! run 'docker exec -it b25d51b50b8c /bin/zsh' to attach to this node
as151h-host_0-10.151.0.71           | ready! run 'docker exec -it 5c5d0da820c7 /bin/zsh' to attach to this node
as151h-host_4-10.151.0.75           | ready! run 'docker exec -it c01f3e4467c5 /bin/zsh' to attach to this node
```

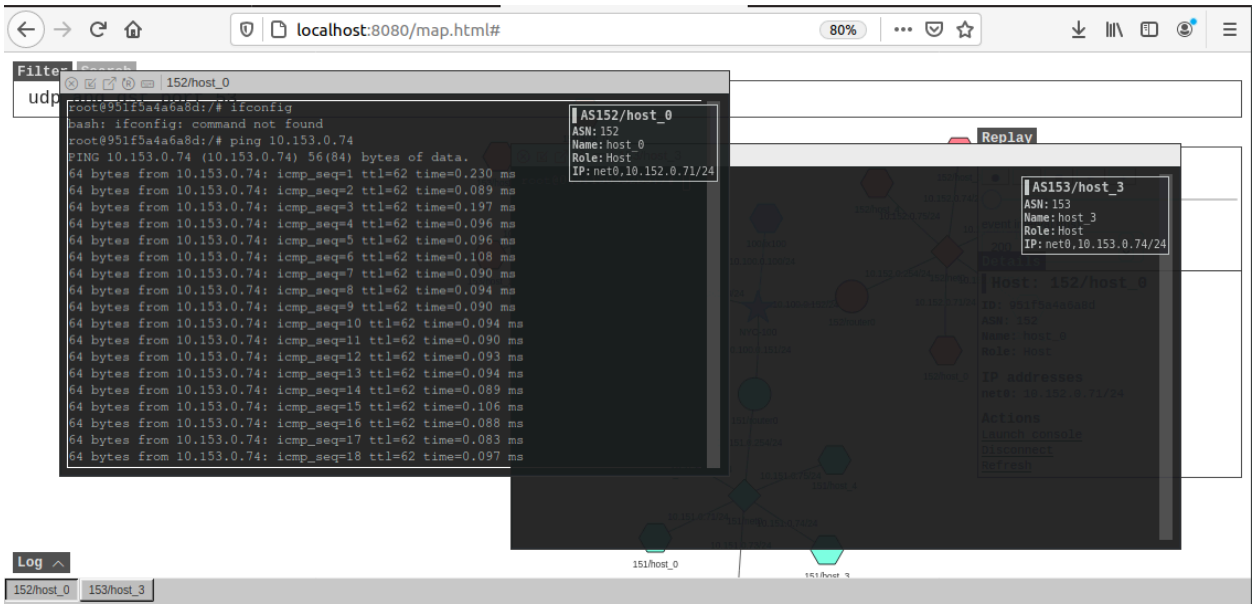


map.html

I have now opened the localhost:8080/map.html page. For whatever reason ifconfig doesn't work on these nodes that isn't really relevant however. I use AS152/host_0 to ping AS153/host_3 at IP 10.153.0.74. This shows that the network topology is working as intended.


localhost:8080/map.html
Filter
Search
Type a BPF expression to animate packet flows on the map...
Replay
Replay stopped.
event interval (ms)
200
Details
Network: 152/net0
ID: 029e035194c1
Name: net0
Scope: 152
Type: local
Prefix: 10.152.0.0/24
Log


SEED Project
Firefox Privacy Notice
map
localhost:8080/map.html#
80%
Filter
Search
152/host_0
root@951f5a4a6a8d:/# ifconfig
bash: ifconfig: command not found
root@951f5a4a6a8d:/#
AS152/host_0
ASN: 152
Name: host_0
Role: Host
IP: net0,10.152.0.71/24
AS153/host_3
ASN: 153
Name: host_3
Role: Host
IP: net0,10.153.0.74/24
Log
152/host_0
153/host_3



worm.py Edits

In order to make the worm.py code relevant to the topology we are working with I made the following edits to worm.py. When running this python script we get this output:

```
# Find the next victim (return an IP address).
# Check to make sure that the target is alive.
def getNextTarget():
    while True:
        a = randint(151, 153) # All network IDs
        b = randint(71, 80) # All host IDs
        ipaddr = f"10.{a}.0.{b}"
        print("Now attacking...")
        print(ipaddr)

        # Get the output of the ping command
        try:
                output = subprocess.check_output(f"ping -q -c1 -W1 {ipaddr}", shell=True)

                result = output.find(b'1 received')

                if result == 1:
                        print(f"{ipaddr} is not alive")
                else:
                        print(f"*** {ipaddr} is alive, launch the attack")
                return ipaddr
        except Exception as e:
                print(e)

```

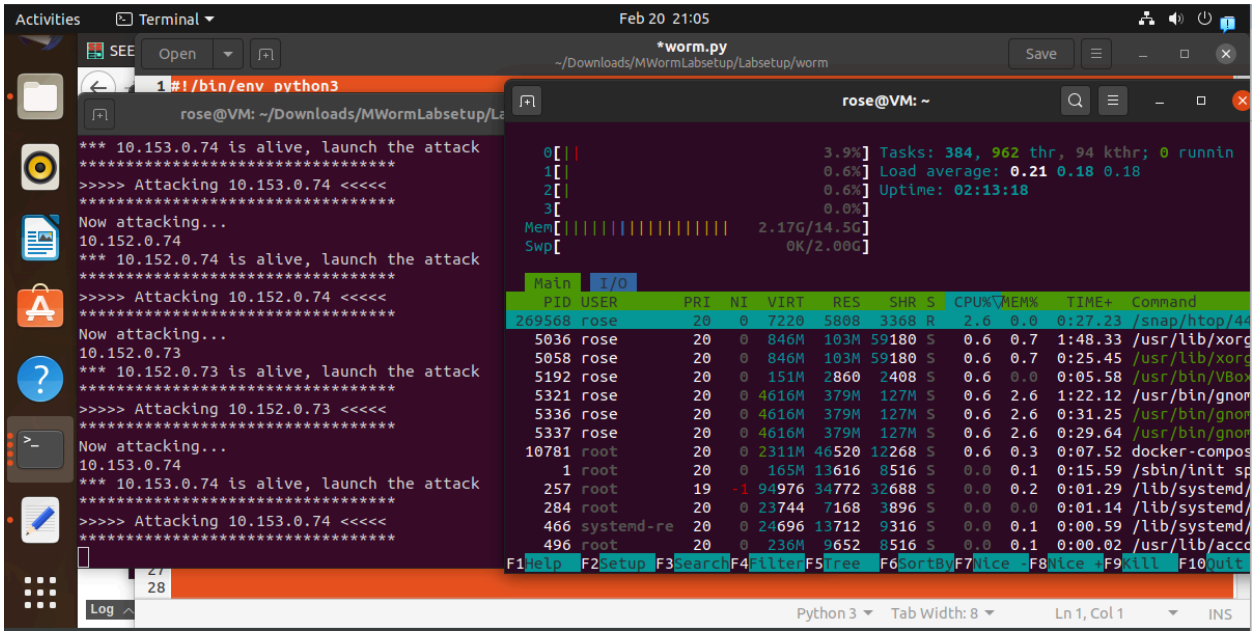
```python
# Check whether the current host is infected with the worm
def isInfectedAlready():
        exists = os.path.exists('badfile')
        if exists:
                return True
        else:
                return False

```

```
rose@VM:~/Downloads/MWormLabsetup/Labsetup/worm$ sudo python3 worm.py
The worm has arrived on this host ^_^
Now attacking...
10.153.0.71
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
*** 10.153.0.71 is alive, launch the attack
**********************************
>>>>> Attacking 10.153.0.71 <<<<<
**********************************
rose@VM:~/Downloads/MWormLabsetup/Labsetup/worm$
```

To make this worm continue to run after this initial attack I remove the exit status at the end of the python script. It should now continue to run repeatedly and attack multiple hosts. I made some additional edits for some bug fixes on this program, it was trying to access host IDs that didn't exist because they were out of range. I limited it to 71-75.

Additionally I made changes to the if/else statements because the way it was coded originally didn't allow the statement "ipaddr is not alive" to be accessible so it would just print junk. That is now fixed so the if statement is alive and else is not alive.

As we can see this produces a clean output, and you can see the memory consumption in htop.



In this lab I learned how to use Morris worm to attack different IP addresses and how to edit it to target specific IPs based on the network topology I’m working with. I also learned how to debug this python script. I'll probably continue to tinker with it. Additionally I was able to get SEED labs virtual images to work on this new computer so I should no longer have issues with lab setups in the future. I also gained some understanding on how docker works.