ICMP LAB – OPTIONAL LAB

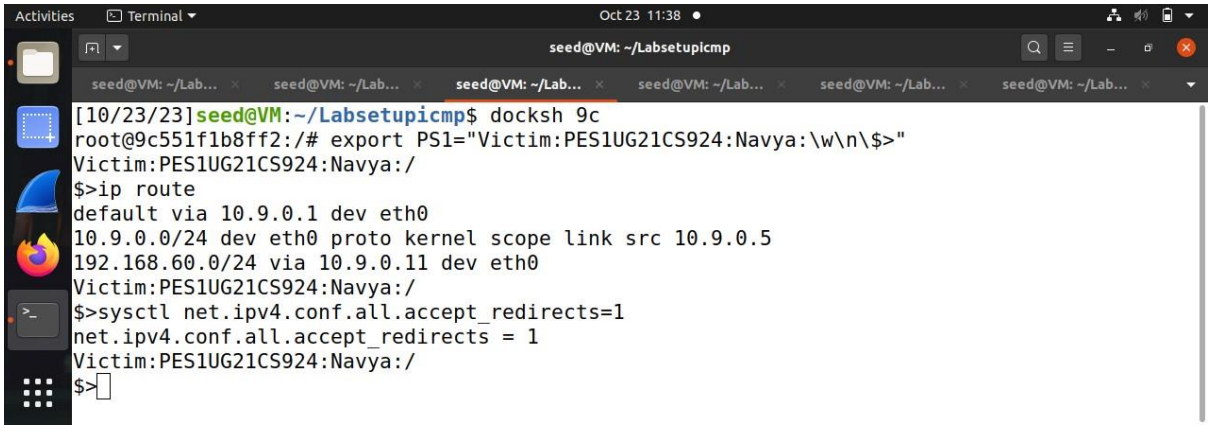
NAME: NAVYA PERAM

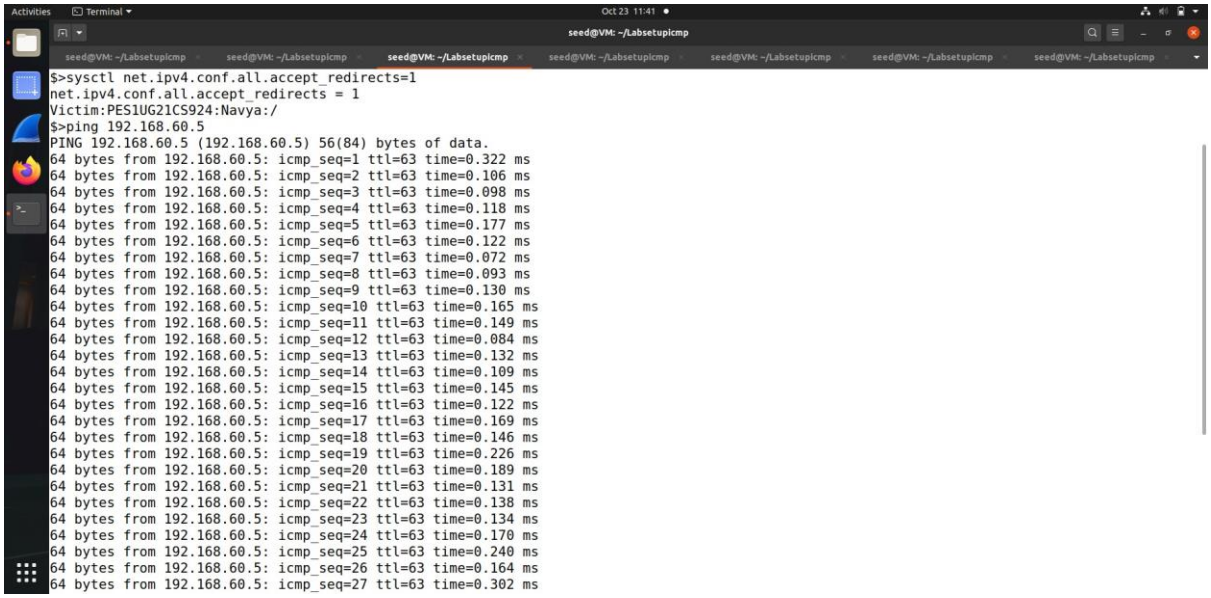
SRN: PES1UG21CS924

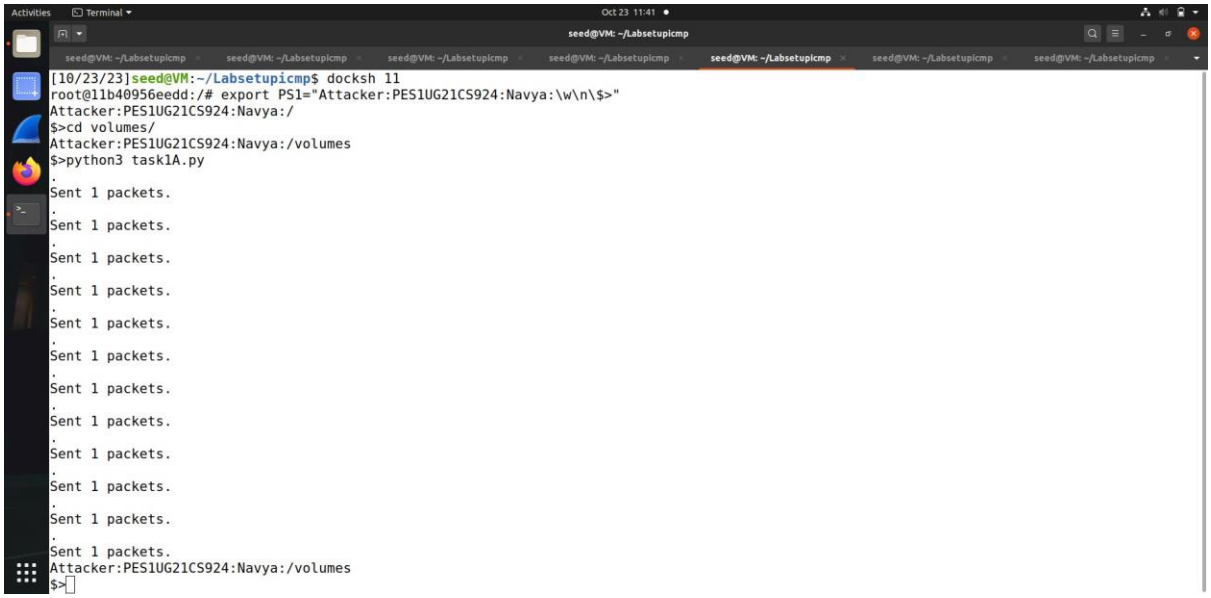
Task 1

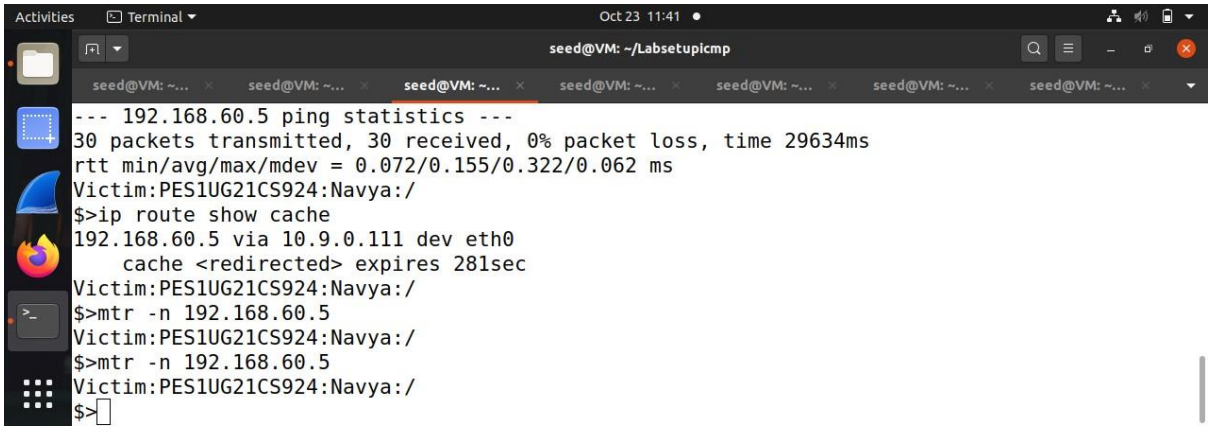
In the given code, we use ip route command to make sure that the victim is routed properly through 10.9.0.11. We then change the value of redirects to 1, which then allows the system to process and take action on redirected ICMP packets upon receiving them, allowing it to optimize the routing performance.

In the given code, the script creates an IPv4 packet (ip) with the source IP address as the real gateway and the destination IP address as the victim. Within this IPv4 packet, it embeds an ICMP redirect message (icmp) with type 5 (Redirect).The gateway (icmp.gw) in the ICMP message is set to the fake gateway's IP address, 10.9.0.111 .



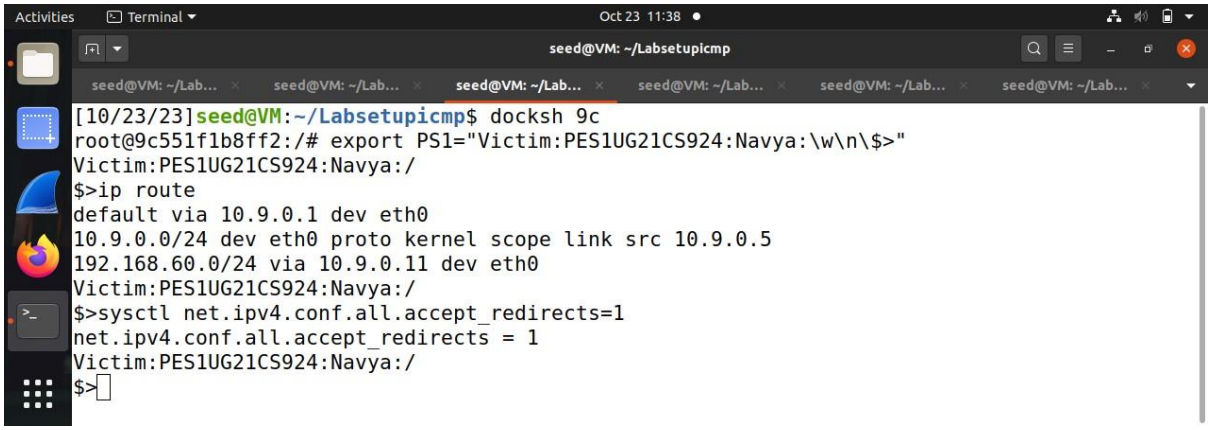


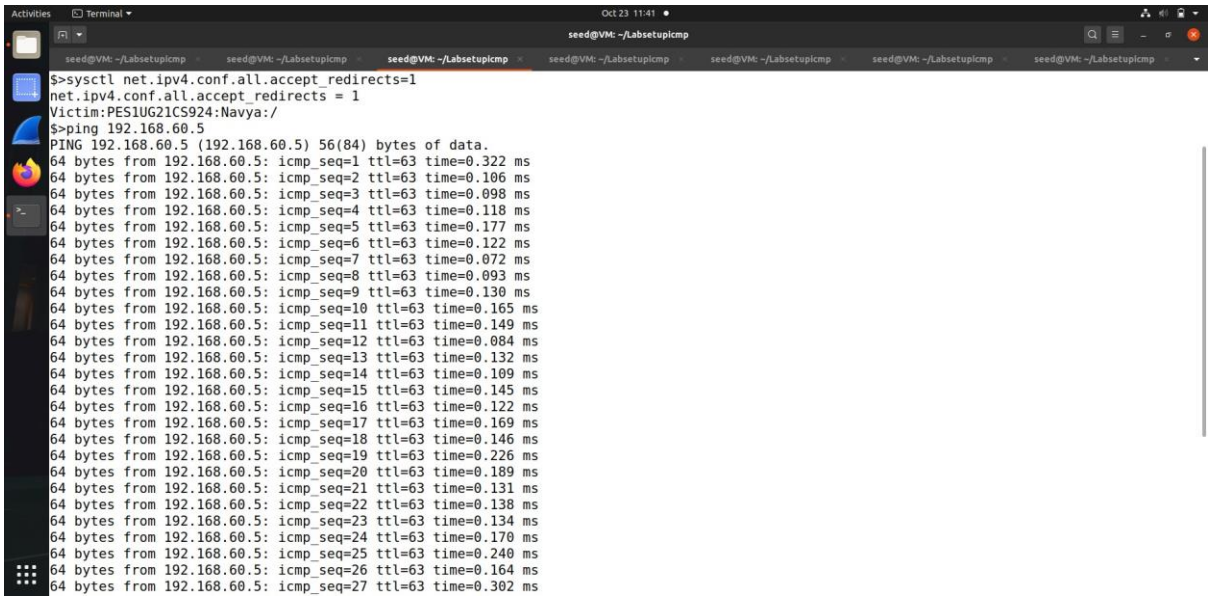


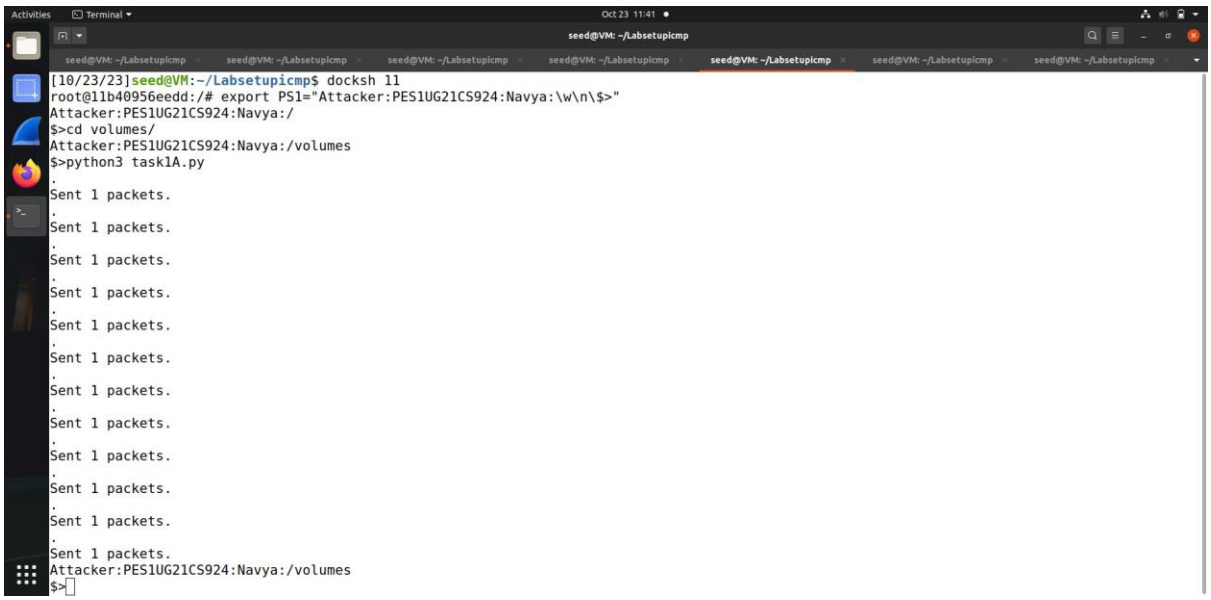


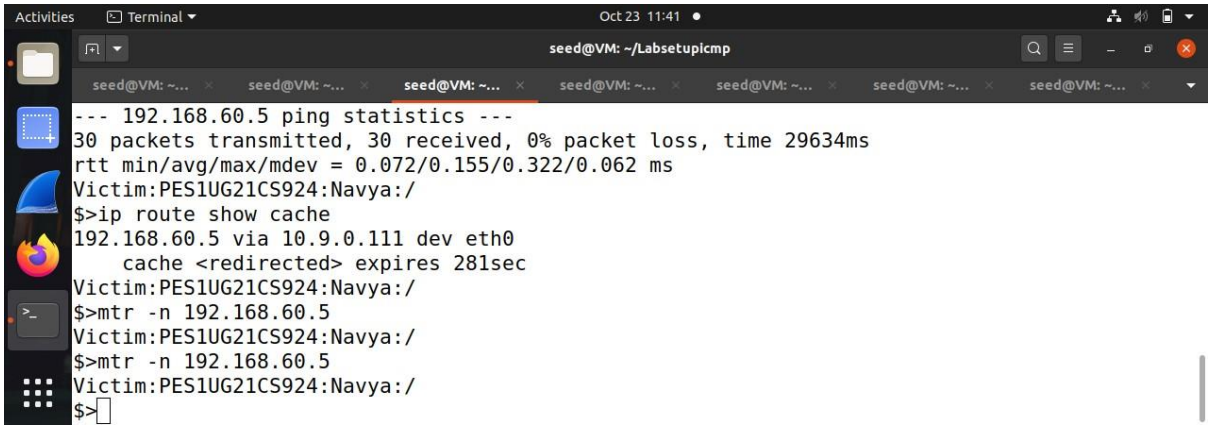
On observing the above code, we find that the entries in the routing cache have been overwritten by the redirected ICMP packets. The routing cache also gives the expiration time of the redirected entries, which is 281 sec. In the above traceroute program run, we can see that the packets have indeed been rerouted properly using the fake getaway ip address, 10.9.0.111, the malicious router.

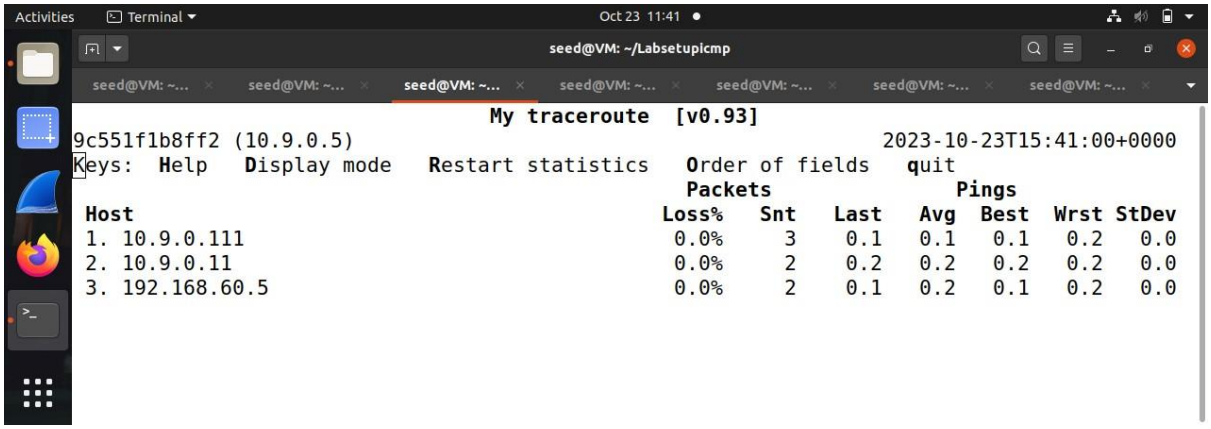
Question 1


Activities
Terminal
Oct 23 11:38
seed@VM: ~/Labsetupicmp
[10/23/23]seed@VM:~/Labsetupicmp$ docksh 9c
root@9c551f1b8ff2:/# export PS1="Victim:PES1UG21CS924:Navya:\w\n\$>"
Victim:PES1UG21CS924:Navya:/
$>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
Victim:PES1UG21CS924:Navya:/
$>sysctl net.ipv4.conf.all.accept_redirects=1
net.ipv4.conf.all.accept_redirects = 1
Victim:PES1UG21CS924:Navya:/
$>


$>sysctl net.ipv4.conf.all.accept_redirects=1
net.ipv4.conf.all.accept_redirects = 1
Victim:PES1UG21CS924:Navya:/
$>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=0.322 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.106 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.098 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.118 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.177 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.122 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.072 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.093 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=0.130 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=0.165 ms
64 bytes from 192.168.60.5: icmp_seq=11 ttl=63 time=0.149 ms
64 bytes from 192.168.60.5: icmp_seq=12 ttl=63 time=0.084 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=0.132 ms
64 bytes from 192.168.60.5: icmp_seq=14 ttl=63 time=0.109 ms
64 bytes from 192.168.60.5: icmp_seq=15 ttl=63 time=0.145 ms
64 bytes from 192.168.60.5: icmp_seq=16 ttl=63 time=0.122 ms
64 bytes from 192.168.60.5: icmp_seq=17 ttl=63 time=0.169 ms
64 bytes from 192.168.60.5: icmp_seq=18 ttl=63 time=0.146 ms
64 bytes from 192.168.60.5: icmp_seq=19 ttl=63 time=0.226 ms
64 bytes from 192.168.60.5: icmp_seq=20 ttl=63 time=0.189 ms
64 bytes from 192.168.60.5: icmp_seq=21 ttl=63 time=0.131 ms
64 bytes from 192.168.60.5: icmp_seq=22 ttl=63 time=0.138 ms
64 bytes from 192.168.60.5: icmp_seq=23 ttl=63 time=0.134 ms
64 bytes from 192.168.60.5: icmp_seq=24 ttl=63 time=0.170 ms
64 bytes from 192.168.60.5: icmp_seq=25 ttl=63 time=0.240 ms
64 bytes from 192.168.60.5: icmp_seq=26 ttl=63 time=0.164 ms
64 bytes from 192.168.60.5: icmp_seq=27 ttl=63 time=0.302 ms

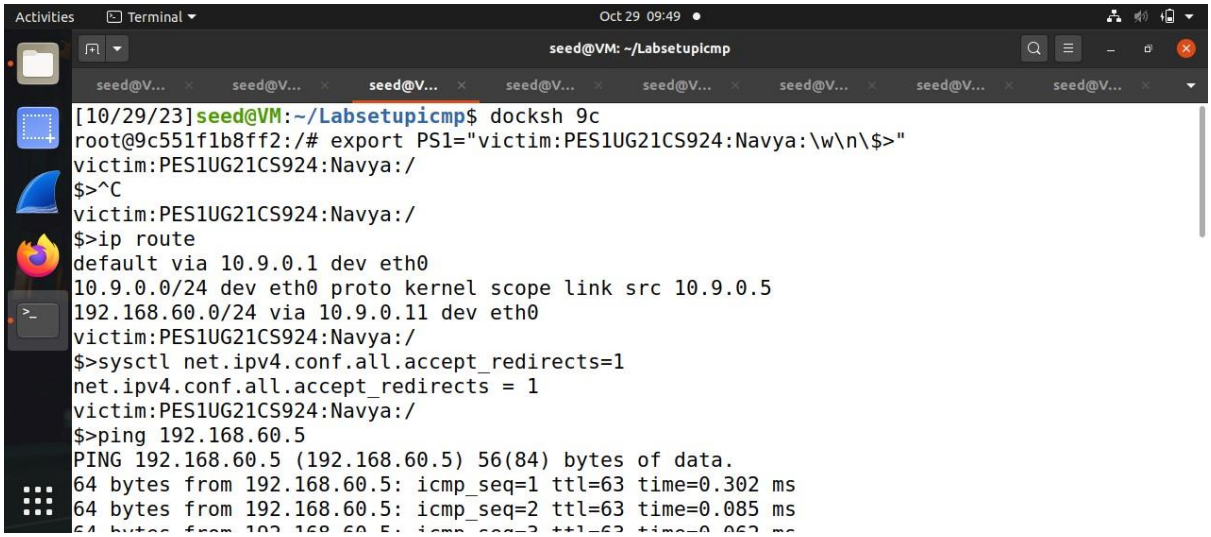

[10/23/23]seed@VM:~/Labsetupicmp$ docksh 11
root@11b40956eedd:/# export PS1="Attacker:PES1UG21CS924:Navya:\w\n\$>"
Attacker:PES1UG21CS924:Navya:/
$>cd volumes/
Attacker:PES1UG21CS924:Navya:/volumes
$>python3 task1A.py
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
Attacker:PES1UG21CS924:Navya:/volumes
$>

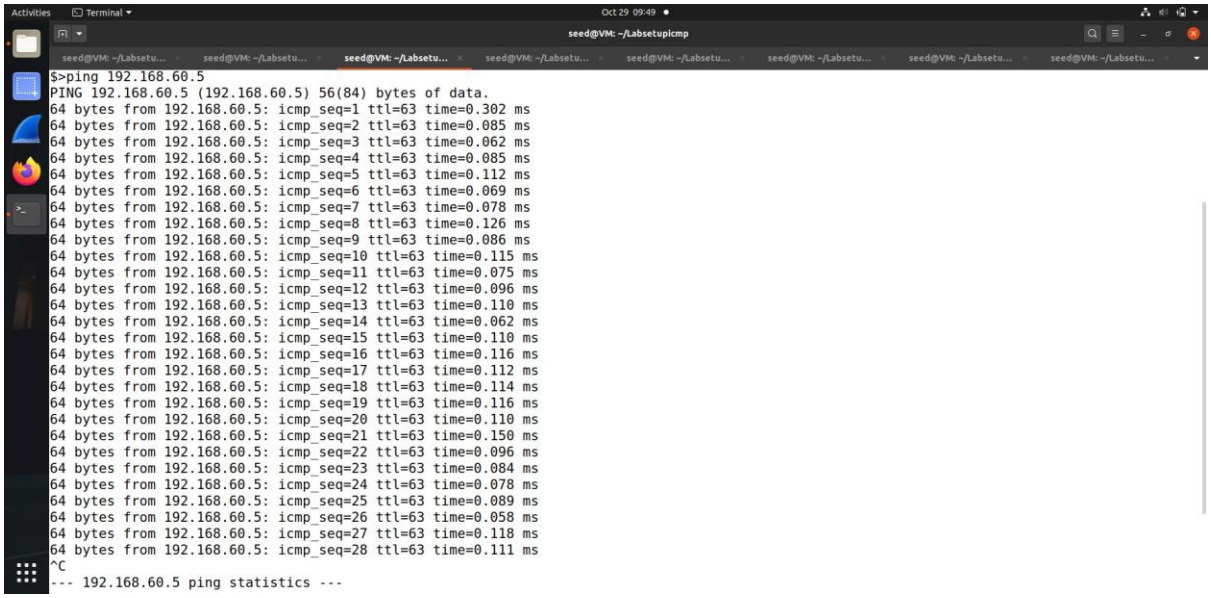


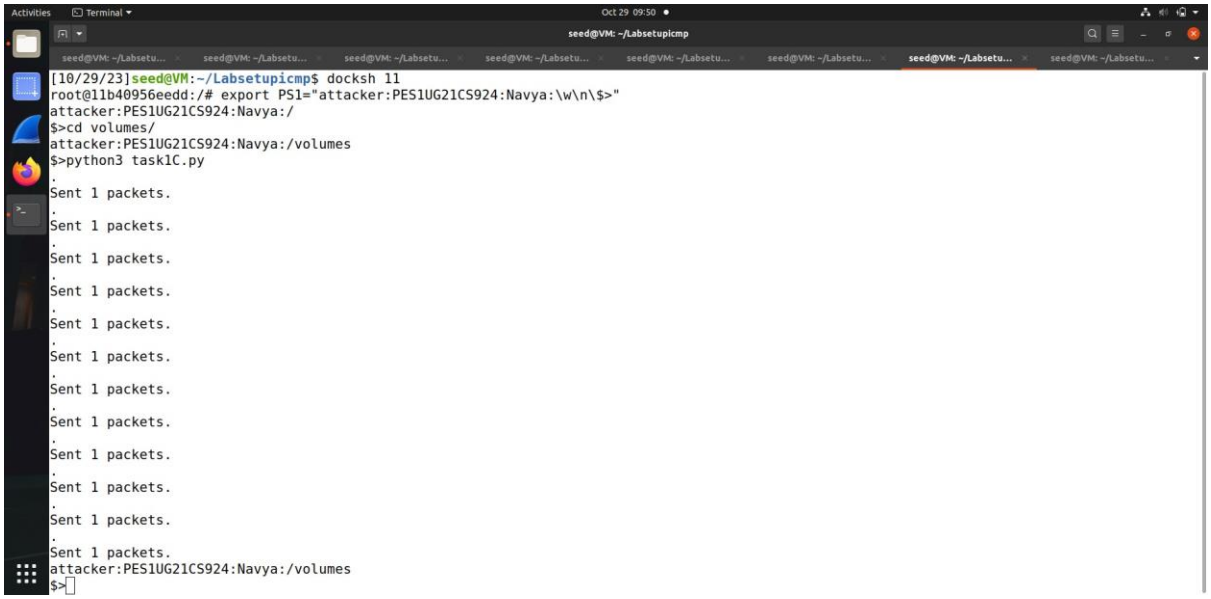


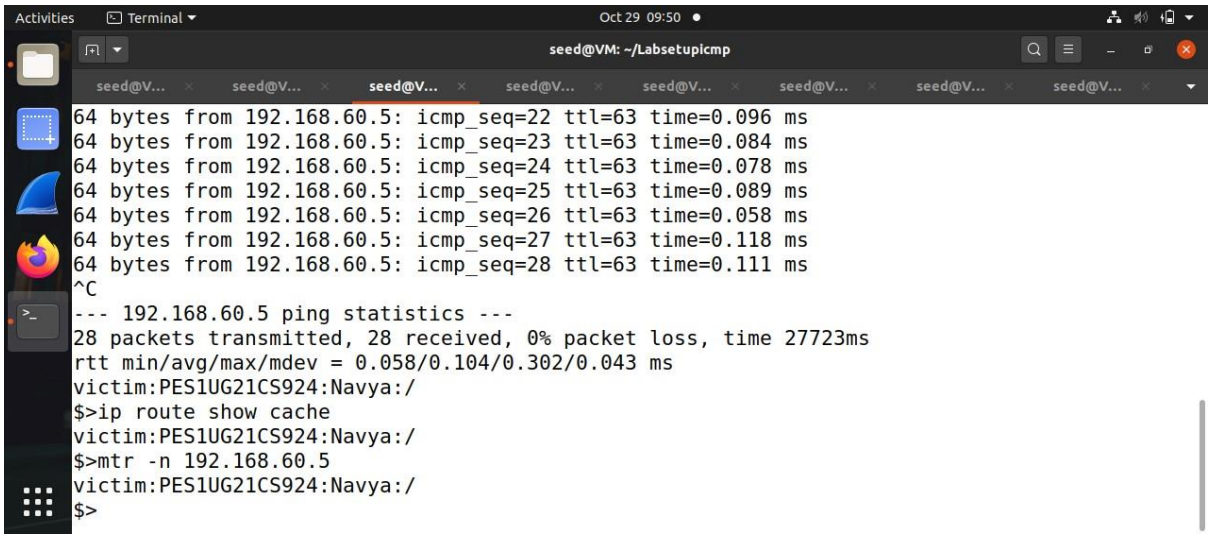
Redirecting traffic to a remote machine located on a different network is generally not feasible using ICMP redirects. The purpose of ICMP redirects is to optimize routing within the local network by guiding hosts to more efficient routes based on the local routing infrastructure. Attempting to redirect traffic to a remote machine would require control over routers and gateways in the remote network, which is typically outside the scope and control of the local network.

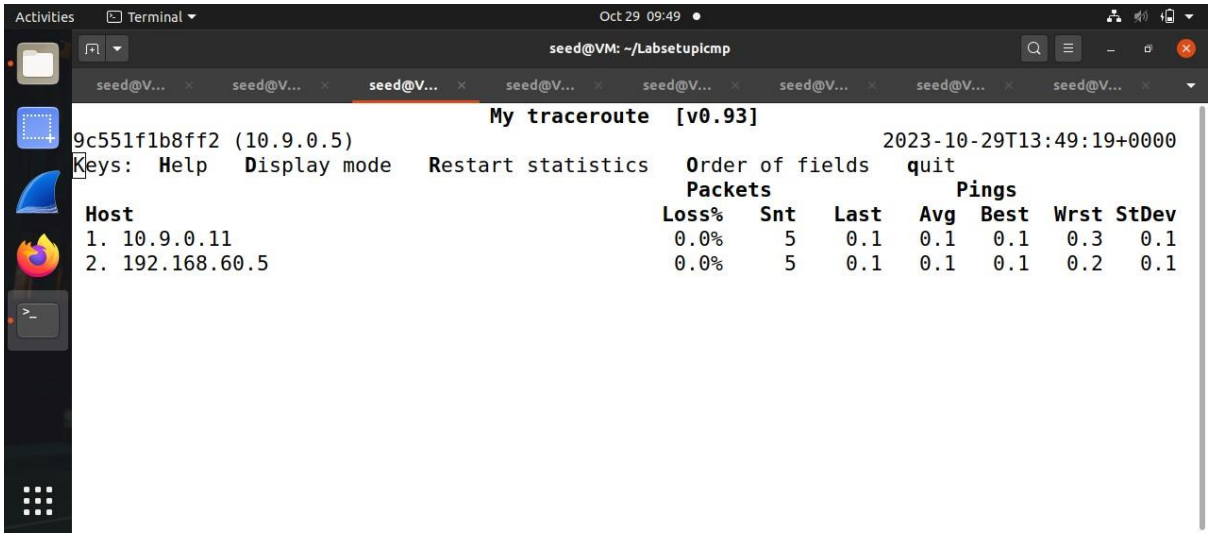
Question 2




Activities
Terminal
Oct 29 09:49
seed@VM: ~/Labsetupicmp
$>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=0.302 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.085 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.062 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=0.112 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.069 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=0.126 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=0.086 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=0.115 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=0.096 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=0.110 ms
64 bytes from 192.168.60.5: icmp_seq=14 ttl=63 time=0.062 ms
64 bytes from 192.168.60.5: icmp_seq=15 ttl=63 time=0.110 ms
64 bytes from 192.168.60.5: icmp_seq=16 ttl=63 time=0.116 ms
64 bytes from 192.168.60.5: icmp_seq=17 ttl=63 time=0.112 ms
64 bytes from 192.168.60.5: icmp_seq=18 ttl=63 time=0.114 ms
64 bytes from 192.168.60.5: icmp_seq=19 ttl=63 time=0.116 ms
64 bytes from 192.168.60.5: icmp_seq=20 ttl=63 time=0.110 ms
64 bytes from 192.168.60.5: icmp_seq=21 ttl=63 time=0.150 ms
64 bytes from 192.168.60.5: icmp_seq=22 ttl=63 time=0.096 ms
64 bytes from 192.168.60.5: icmp_seq=23 ttl=63 time=0.084 ms
64 bytes from 192.168.60.5: icmp_seq=24 ttl=63 time=0.078 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=0.058 ms
64 bytes from 192.168.60.5: icmp_seq=27 ttl=63 time=0.118 ms
64 bytes from 192.168.60.5: icmp_seq=28 ttl=63 time=0.111 ms
^C
--- 192.168.60.5 ping statistics ---


Activities
Terminal
Oct 29 09:50
seed@VM: ~/Labsetupicmp
[10/29/23]seed@VM:~/Labsetupicmp$ docksh 11
root@11b40956eedd:/# export PS1="attacker:PES1UG21CS924:Navya:\w\n\$>"
attacker:PES1UG21CS924:Navya:/
$>cd volumes/
attacker:PES1UG21CS924:Navya:/volumes
$>python3 task1C.py
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
attacker:PES1UG21CS924:Navya:/volumes
$>


Activities
Terminal
Oct 29 09:50
seed@VM: ~/Labsetupicmp
64 bytes from 192.168.60.5: icmp_seq=22 ttl=63 time=0.096 ms
64 bytes from 192.168.60.5: icmp_seq=23 ttl=63 time=0.084 ms
64 bytes from 192.168.60.5: icmp_seq=24 ttl=63 time=0.078 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=0.058 ms
64 bytes from 192.168.60.5: icmp_seq=27 ttl=63 time=0.118 ms
64 bytes from 192.168.60.5: icmp_seq=28 ttl=63 time=0.111 ms
^C
--- 192.168.60.5 ping statistics ---
28 packets transmitted, 28 received, 0% packet loss, time 27723ms
rtt min/avg/max/mdev = 0.058/0.104/0.302/0.043 ms
victim:PES1UG21CS924:Navya:/
$>ip route show cache
victim:PES1UG21CS924:Navya:/
$>mtr -n 192.168.60.5
victim:PES1UG21CS924:Navya:/
$>



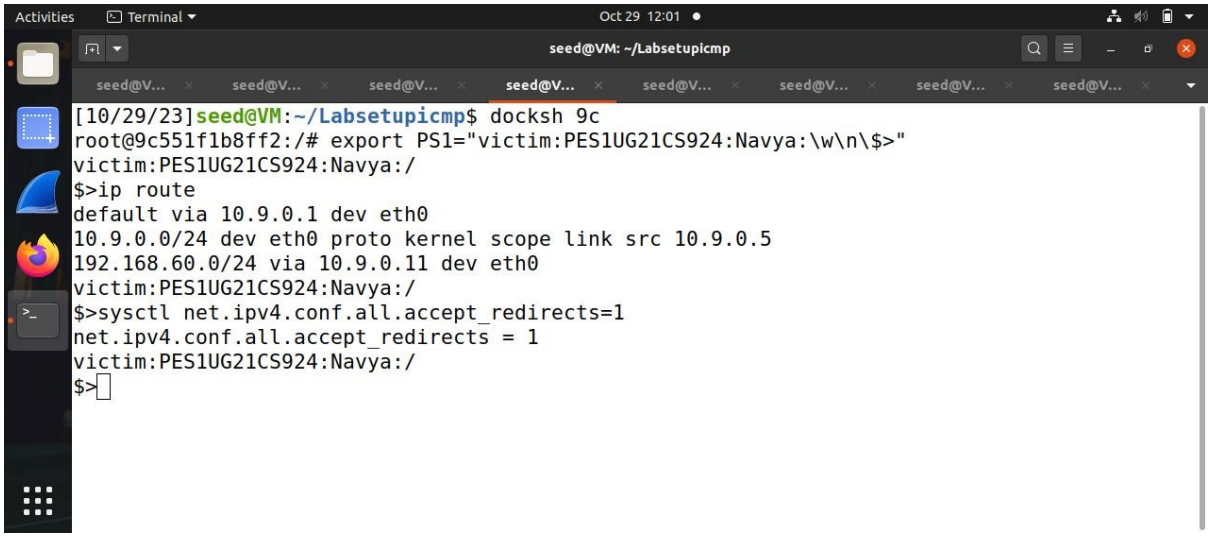
We can see that the above process has not been done in an appropriate manner, as it does not use the malicious router to reroute the packets. ICMP redirects are created to inform hosts about better routes and for it to be done in an effective manner, the specified destination should be a responsive gateway or router. In the case of a non-existing machine, there is no entity to acknowledge the redirect, making the redirection meaningless and inefficient. We are also unable to see any redirected packets in the cache, as seen by the show cache command.

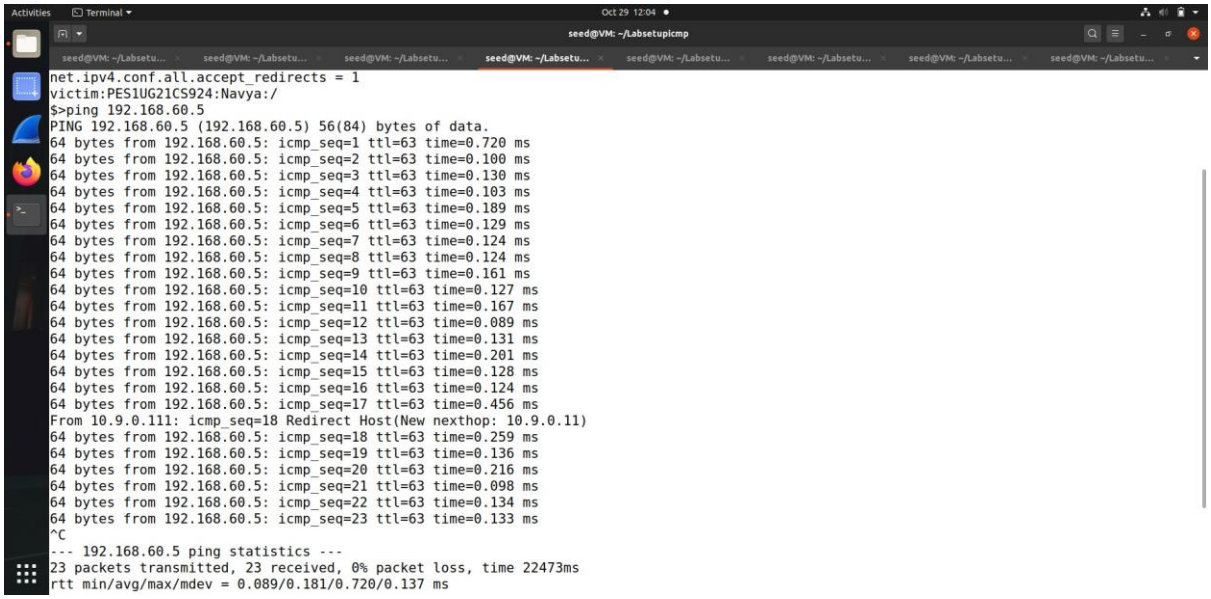
Question 3

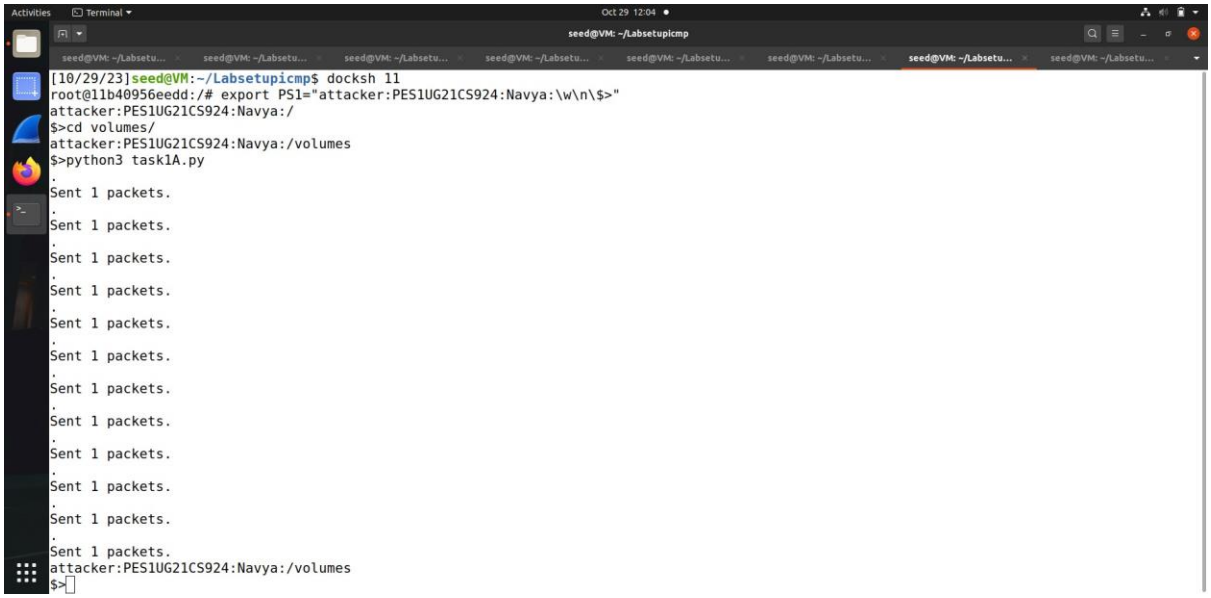
net.ipv4.conf.all.send_redirects=1: This entry sets the send_redirects parameter for all network interfaces on the container to 1, allowing the container to send ICMP redirect packets.

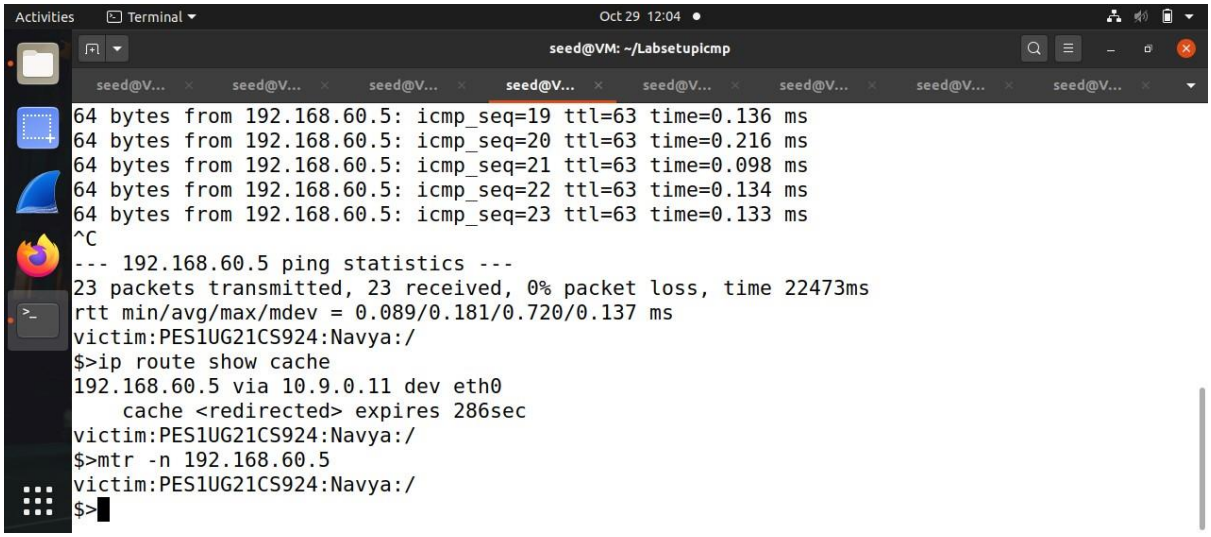
net.ipv4.conf.default.send_redirects=1: This entry sets the send_redirects parameter for the default network interface to 1. The default network interface is often the primary interface used for outgoing traffic.

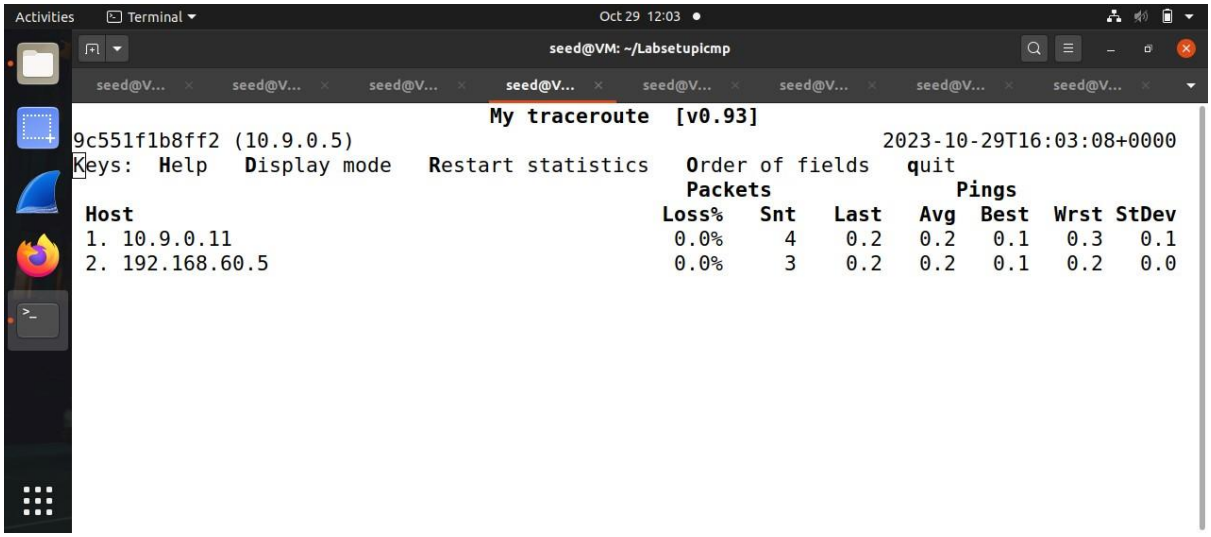
net.ipv4.conf.eth0.send_redirects=1: This entry sets the send_redirects parameter for a specific network interface, eth0, to 1. These commands allows the container to send ICMP redirect packets.


Activities
Terminal
Oct 29 12:01
seed@VM: ~/Labsetupicmp
[10/29/23]seed@VM:~/Labsetupicmp$ docksh 9c
root@9c551f1b8ff2:/# export PS1="victim:PES1UG21CS924:Navya:\w\n\$>"
victim:PES1UG21CS924:Navya:/
$>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
victim:PES1UG21CS924:Navya:/
$>sysctl net.ipv4.conf.all.accept_redirects=1
net.ipv4.conf.all.accept_redirects = 1
victim:PES1UG21CS924:Navya:/
$>


net.ipv4.conf.all.accept_redirects = 1
victim:PES1UG21CS924:Navya:/
$>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=0.720 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=0.130 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.103 ms
64 bytes from 192.168.60.5: icmp_seq=5 ttl=63 time=0.189 ms
64 bytes from 192.168.60.5: icmp_seq=6 ttl=63 time=0.129 ms
64 bytes from 192.168.60.5: icmp_seq=7 ttl=63 time=0.124 ms
64 bytes from 192.168.60.5: icmp_seq=8 ttl=63 time=0.124 ms
64 bytes from 192.168.60.5: icmp_seq=9 ttl=63 time=0.161 ms
64 bytes from 192.168.60.5: icmp_seq=10 ttl=63 time=0.127 ms
64 bytes from 192.168.60.5: icmp_seq=11 ttl=63 time=0.167 ms
64 bytes from 192.168.60.5: icmp_seq=12 ttl=63 time=0.089 ms
64 bytes from 192.168.60.5: icmp_seq=13 ttl=63 time=0.131 ms
64 bytes from 192.168.60.5: icmp_seq=14 ttl=63 time=0.201 ms
64 bytes from 192.168.60.5: icmp_seq=15 ttl=63 time=0.128 ms
64 bytes from 192.168.60.5: icmp_seq=16 ttl=63 time=0.124 ms
64 bytes from 192.168.60.5: icmp_seq=17 ttl=63 time=0.456 ms
From 10.9.0.111: icmp_seq=18 Redirect Host(New nexthop: 10.9.0.11)
64 bytes from 192.168.60.5: icmp_seq=18 ttl=63 time=0.259 ms
64 bytes from 192.168.60.5: icmp_seq=19 ttl=63 time=0.136 ms
64 bytes from 192.168.60.5: icmp_seq=20 ttl=63 time=0.216 ms
64 bytes from 192.168.60.5: icmp_seq=21 ttl=63 time=0.098 ms
64 bytes from 192.168.60.5: icmp_seq=22 ttl=63 time=0.134 ms
64 bytes from 192.168.60.5: icmp_seq=23 ttl=63 time=0.133 ms
^C
--- 192.168.60.5 ping statistics ---
23 packets transmitted, 23 received, 0% packet loss, time 22473ms
rtt min/avg/max/mdev = 0.089/0.181/0.720/0.137 ms


[10/29/23]seed@VM:~/Labsetupicmp$ docksh 11
root@11b40956eedd:/# export PS1="attacker:PES1UG21CS924:Navya:\w\n\$>"
attacker:PES1UG21CS924:Navya:/
$>cd volumes/
attacker:PES1UG21CS924:Navya:/volumes
$>python3 task1A.py
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
attacker:PES1UG21CS924:Navya:/volumes
$>



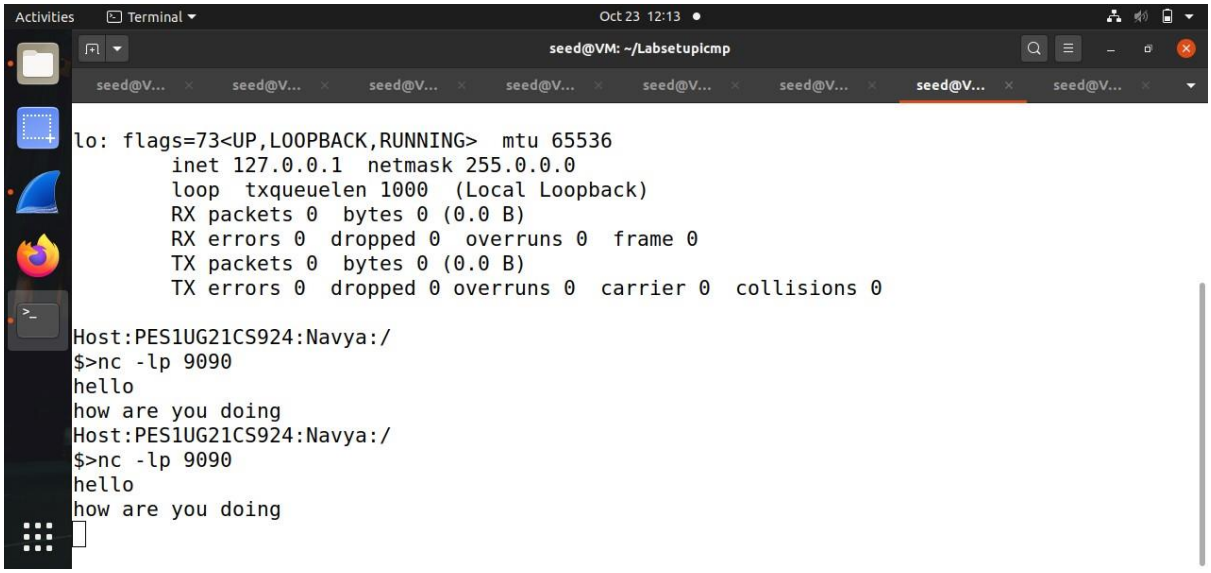


We find out that the process has not been done in the appropriate manner, since only the victim and the host IP address are visible in the rerouting table. It does not make use of the malicious router to redirect the packets in the appropriate manner. However, we are able to see the redirected packets in the cache, showing that it has been done by the router.

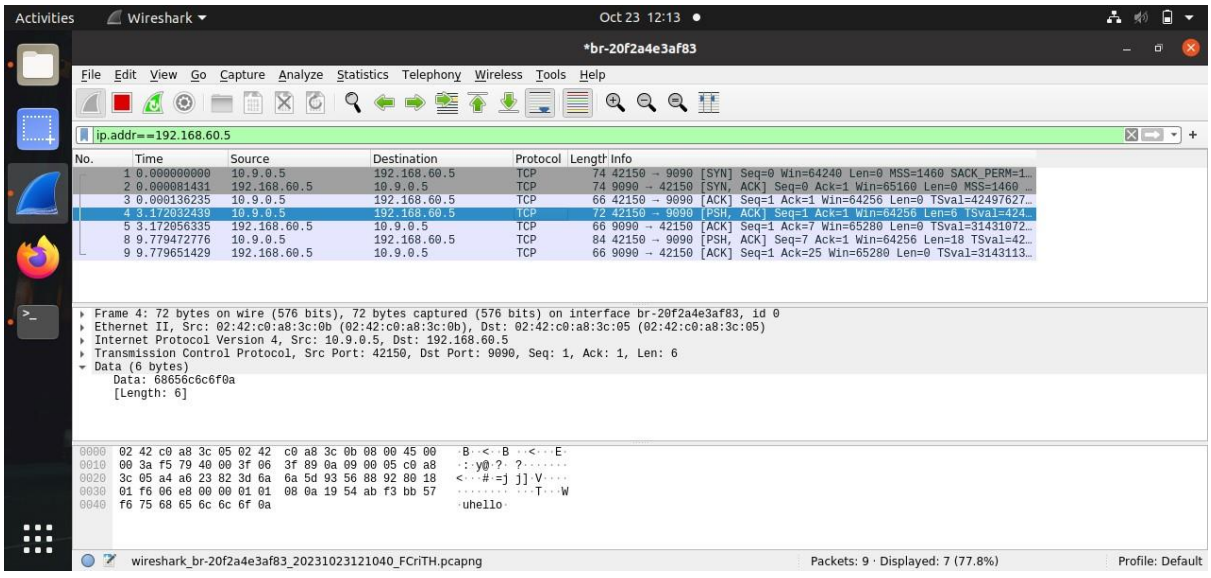
Task 2

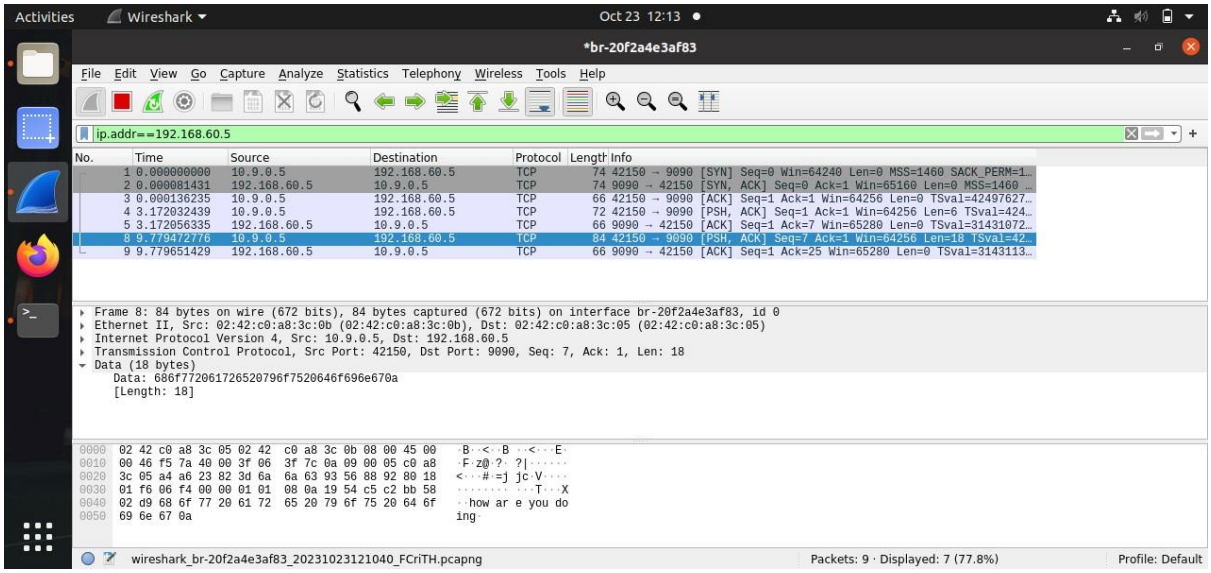
Task 2A

We create a netcat connection between the destination server and the victim. We can then conform this connection by sending and receiving the given messages on both sides. We create the netcat connection on the port 9090, where it actively listens for connections.


Activities
Terminal
Oct 23 12:13
seed@VM: ~/Labsetupicmp
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
Host:PES1UG21CS924:Navya:/
$>nc -lp 9090
hello
how are you doing
Host:PES1UG21CS924:Navya:/
$>nc -lp 9090
hello
how are you doing


Activities
Terminal
Oct 23 12:13
seed@VM: ~/Labsetupicmp
192.168.60.5 via 10.9.0.111 dev eth0
cache <redirected> expires 281sec
Victim:PES1UG21CS924:Navya:/
$>mtr -n 192.168.60.5
Victim:PES1UG21CS924:Navya:/
$>mtr -n 192.168.60.5
Victim:PES1UG21CS924:Navya:/
$>ip route flush cache
Victim:PES1UG21CS924:Navya:/
$>nc 192.168.60.5 9090
hello
how are you doing
^C
Victim:PES1UG21CS924:Navya:/
$>nc 192.168.60.5 9090
hello
how are you doing

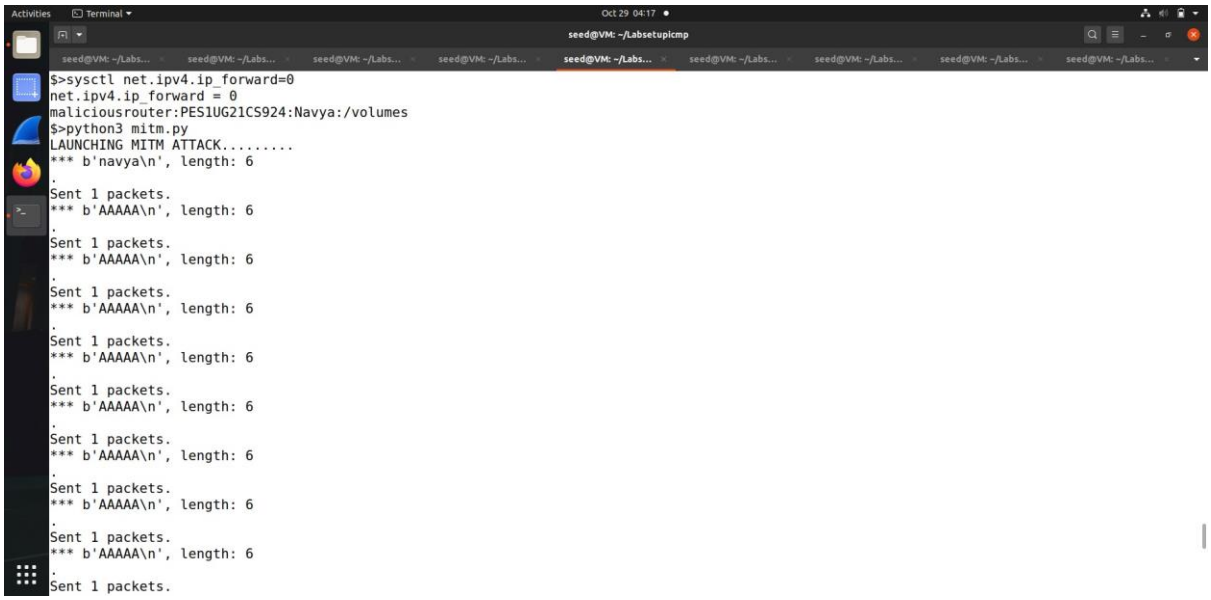

Activities
Wireshark
Oct 23 12:13
*br-20f2a4e3af83
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr==192.168.60.5
No. Time Source Destination Protocol Length Info
1 0.000000000 10.9.0.5 192.168.60.5 TCP 74 42150 → 9090 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1…
2 0.000081431 192.168.60.5 10.9.0.5 TCP 74 9090 → 42150 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 …
3 0.000136235 10.9.0.5 192.168.60.5 TCP 66 42150 → 9090 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=42497627…
4 3.172032439 10.9.0.5 192.168.60.5 TCP 72 42150 → 9090 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=6 TSval=424…
5 3.172056335 192.168.60.5 10.9.0.5 TCP 66 9090 → 42150 [ACK] Seq=1 Ack=7 Win=65280 Len=0 TSval=31431072…
8 9.779472776 10.9.0.5 192.168.60.5 TCP 84 42150 → 9090 [PSH, ACK] Seq=7 Ack=1 Win=64256 Len=18 TSval=42…
9 9.779651429 192.168.60.5 10.9.0.5 TCP 66 9090 → 42150 [ACK] Seq=1 Ack=25 Win=65280 Len=0 TSval=3143113…
Frame 4: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface br-20f2a4e3af83, id 0
Ethernet II, Src: 02:42:c0:a8:3c:0b (02:42:c0:a8:3c:0b), Dst: 02:42:c0:a8:3c:05 (02:42:c0:a8:3c:05)
Internet Protocol Version 4, Src: 10.9.0.5, Dst: 192.168.60.5
Transmission Control Protocol, Src Port: 42150, Dst Port: 9090, Seq: 1, Ack: 1, Len: 6
Data (6 bytes)
Data: 68656c6c6f0a
[Length: 6]
0000 02 42 c0 a8 3c 05 02 42 c0 a8 3c 0b 08 00 45 00
0010 00 3a f5 79 40 00 3f 06 3f 89 0a 09 00 05 c0 a8
0020 3c 05 a4 a6 23 82 3d 6a 6a 5d 93 56 88 92 80 18
0030 01 f6 06 e8 00 00 01 01 08 0a 19 54 ab f3 bb 57
0040 f6 75 68 65 6c 6c 6f 0a
wireshark_br-20f2a4e3af83_20231023121040_FCriTH.pcapng
Packets: 9 · Displayed: 7 (77.8%)
Profile: Default

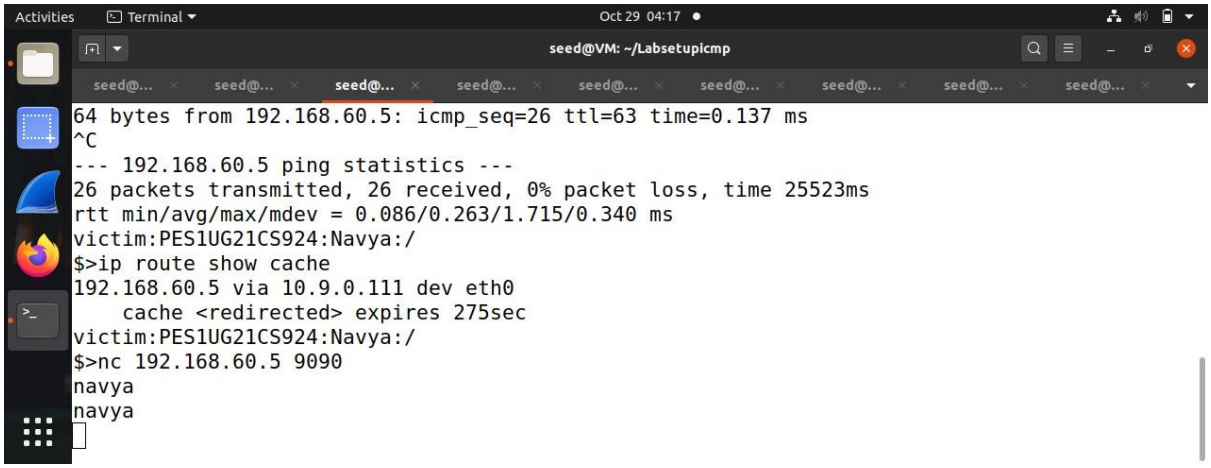


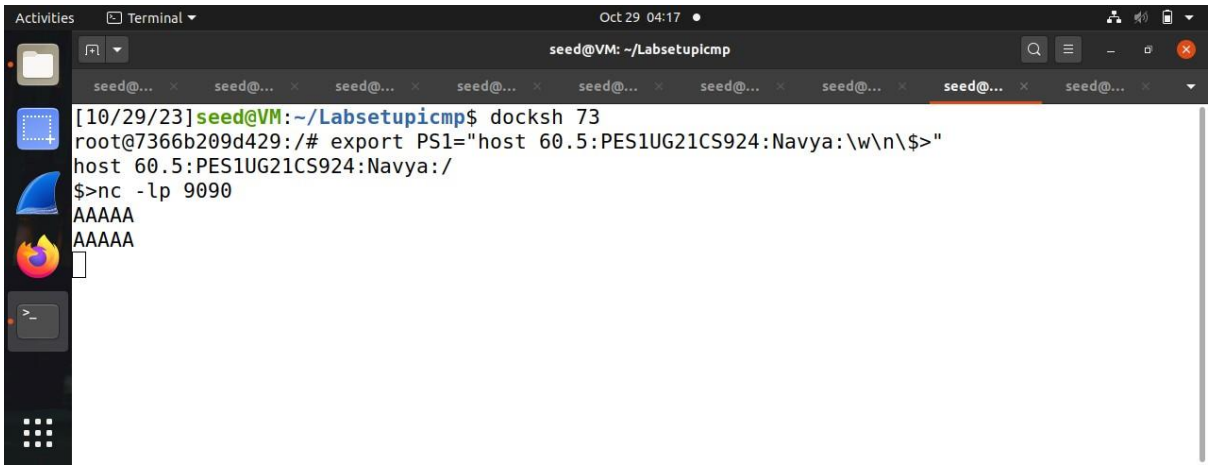
We can observe the messages sent and receive between the victim and the host conforming the connection in the above wireshark images.

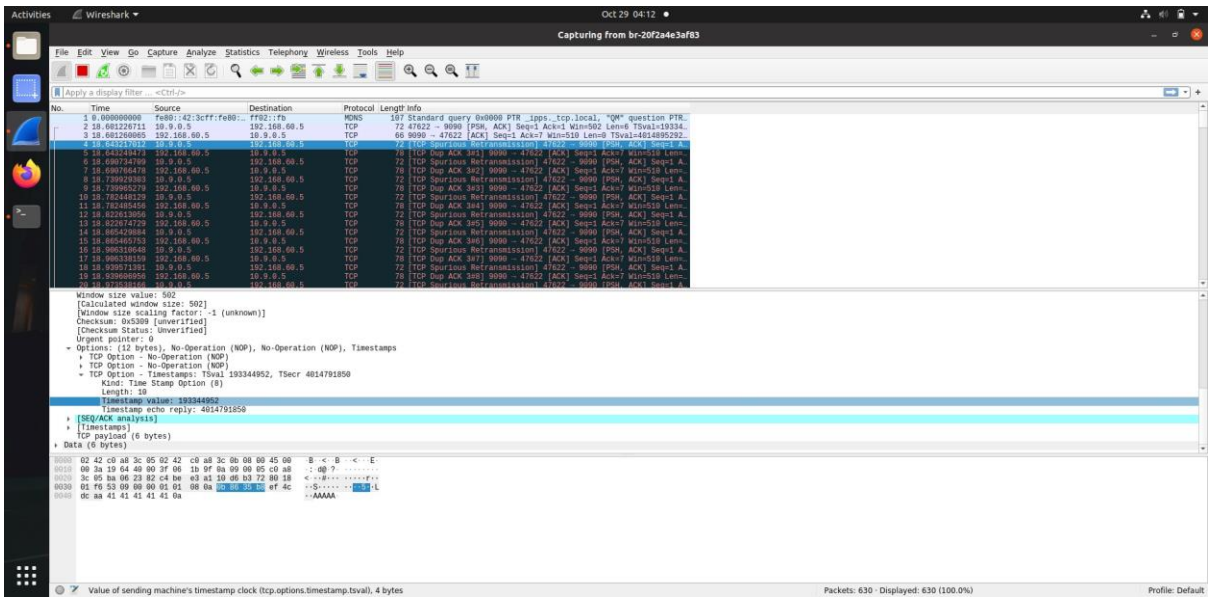
Task 2B

In this program we set the ip forwarding to 0, it disables IP packet forwarding. In this state, the system will not act as a router and will not forward packets between network interfaces. Which is the default setting for many Linux systems that are not configured as routers or gateways. We then run the mitm attack on the malicious router, and since the IP forwarding is disabled the program needs to forward the packet from the victim to the target, after making the appropriate changes. Which it does as a sniff and spoof process, changing all the input characters to As.









We observe that the output is a sequence of A's of the input proving that the program has been successful.

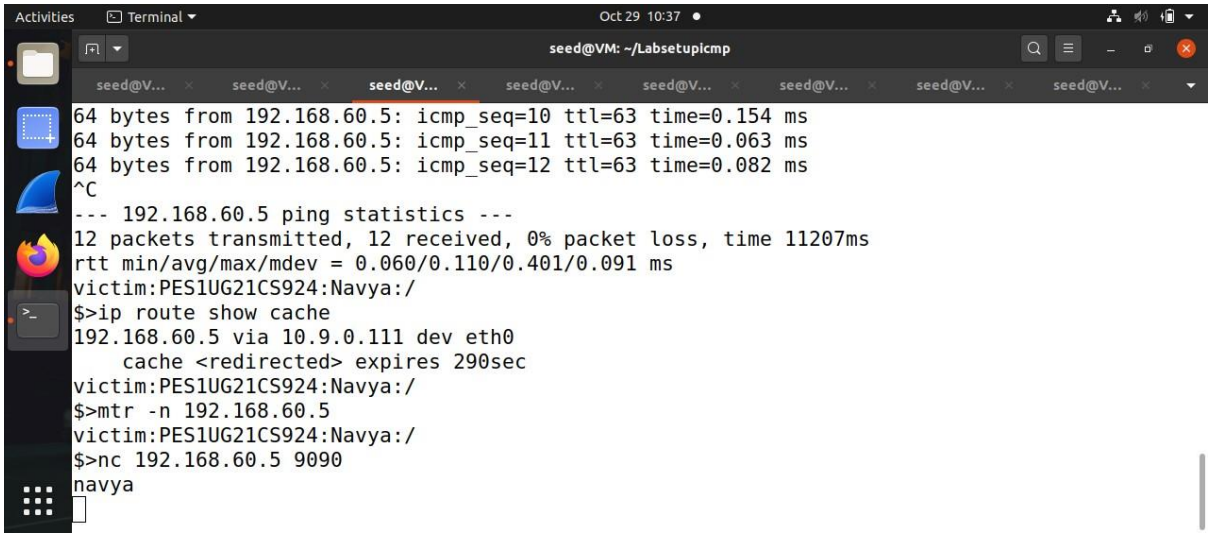
Question 4

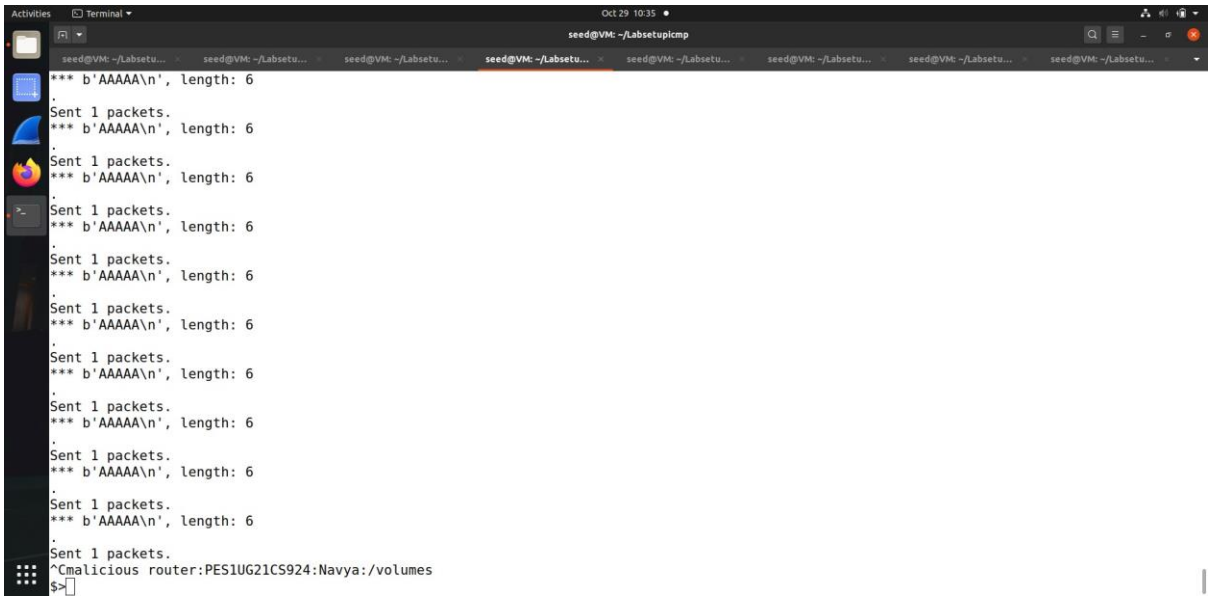
In our MITM code, we capture the traffic coming from the victim, from 10.9.0.5 . the MITM program needs to capture traffic in the outgoing direction from the victim to the target

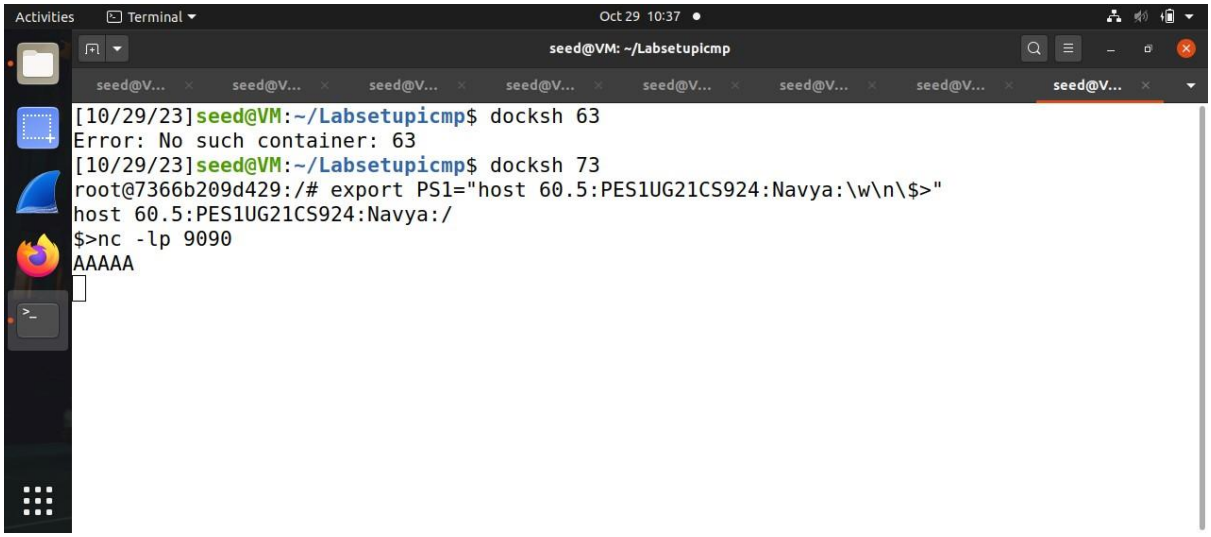
destination. Since the attacker's goal is to manipulate the victim's routing decisions. By capturing the outgoing traffic from the victim, the attacker can intercept and manipulate the packets before they reach the target, 192.168.60.5. Once intercepted, the attacker can then inject malicious routing information in the form of ICMP redirect packets to make the victim believe that the malicious router (10.9.0.111) is the then required route for reaching the target. Capturing outgoing traffic allows the attacker to control and manipulate the victim's routing behavior for packets sent to 192.168.60.5, enabling the MITM attack as a result.

Question 5

A IP address change

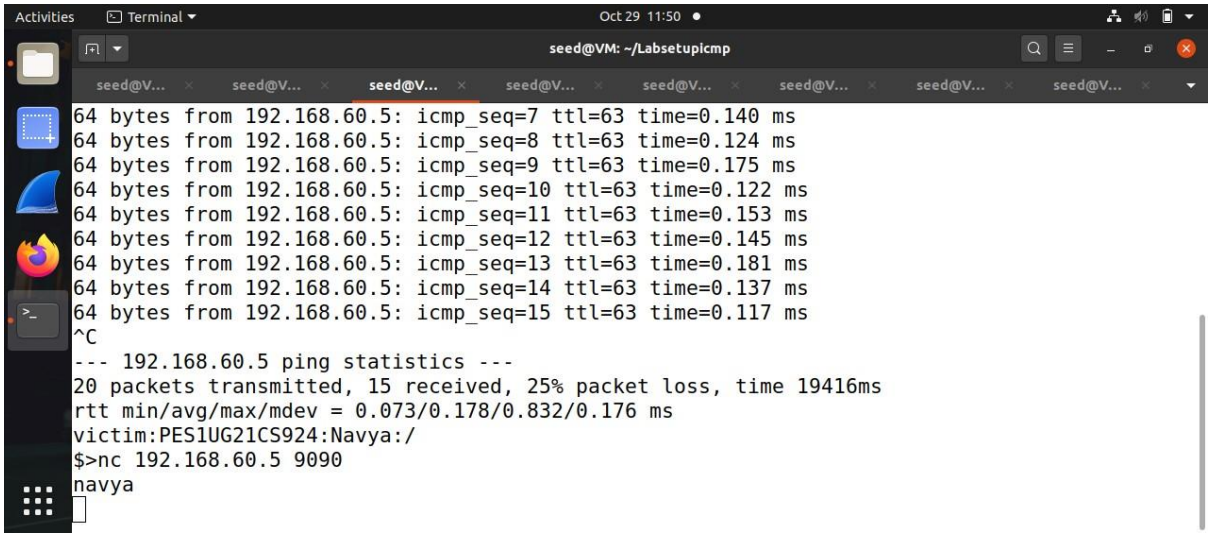


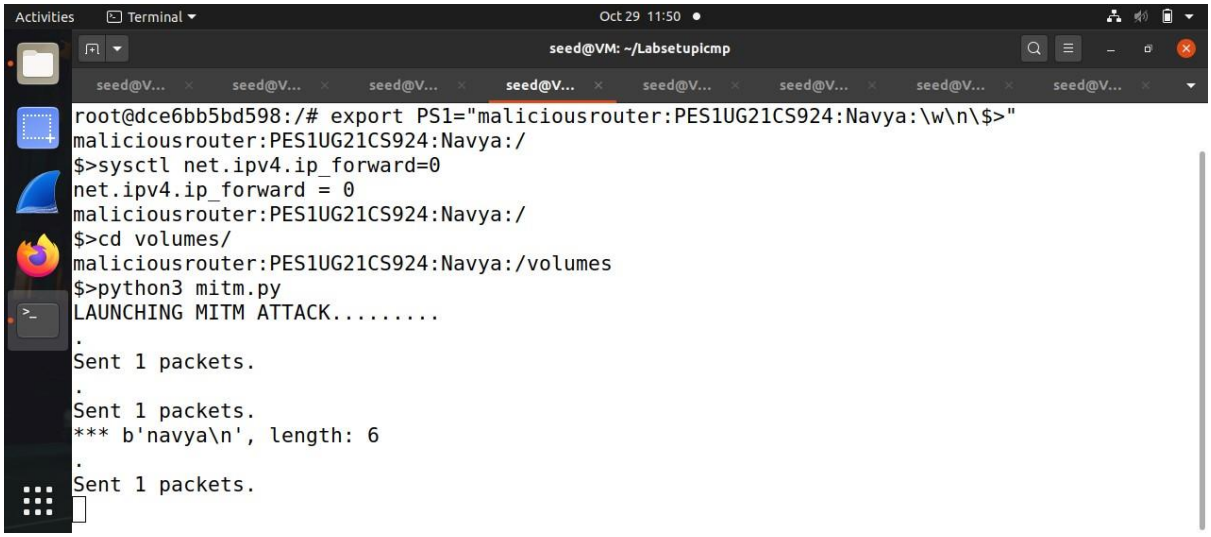


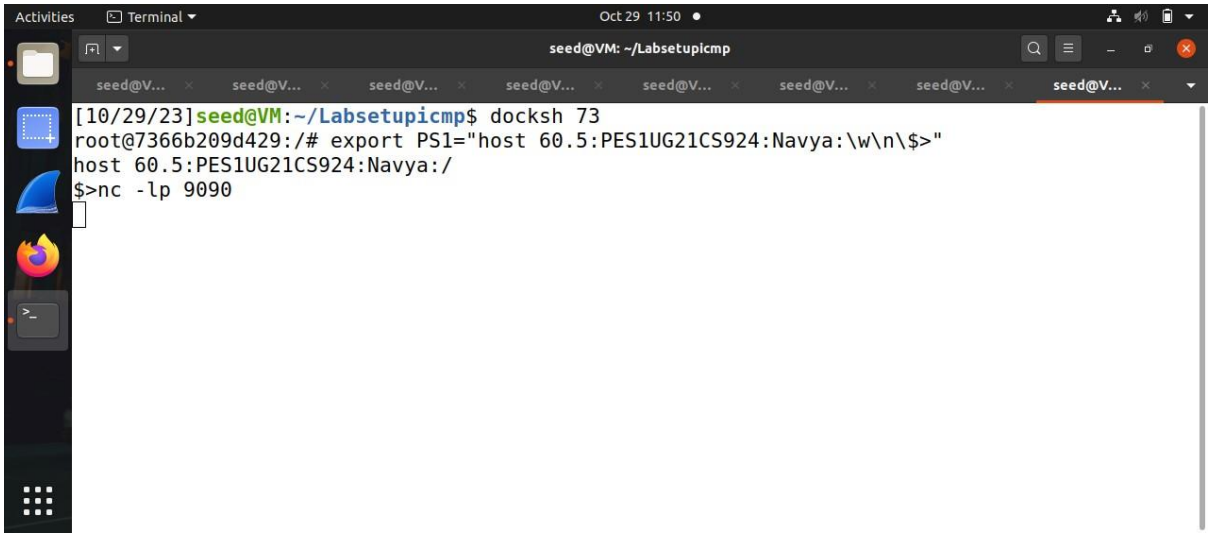


In the above program, setting the filter to capture traffic from host A using its IP address ('tcp and src host 10.9.0.5') is the correct choice. This filter will capture all TCP traffic with a source IP address of 10.9.0.5, which is the IP address of host A. This approach accurately targets and captures traffic from host A, allowing the attacker to intercept and manipulate packets effectively. We observe that the Tcp input packets have been changed with respect to the output which replaced all the characters of the input with capital A.

A MAC address change







We haven’t been able to receive the spoofed output in the above context.

However in general, changing the IP address is always preferred over the mac address as capturing traffic by MAC address would require in-depth knowledge of the MAC address of host A, and it would only work within the local network segment.

Additionally, MAC addresses are typically not used in higher-level network filtering because they are specific to the local network segment and are not visible when the traffic crosses routers. Therefore, filtering by MAC address would be less practical for capturing traffic in this MITM attack scenario.