EXP #8

ICMP REDIRECT ATTACK LAB

SLIDES BY: MOHAMAD BALAWI

OUTLINE

Introduction

Task 1: Launching ICMP Redirect Attack

- Question 1
- Question 2
- Question 3

Task 2: Launching the MITM Attack

- Question 4
- Question 5

Internet Control Message Protocol (ICMP)

- The Internet Control Message Protocol (ICMP) is a network layer protocol used by network devices to diagnose network communication issues. ICMP is mainly used to determine whether or not data is reaching its intended destination in a timely manner. Commonly, the ICMP protocol is used on network devices, such as routers. ICMP is crucial for error reporting and testing, but it can also be used in distributed denial-of-service (DDoS) attacks.
- Common uses of ICMP:
  - **Ping**: sends ICMP echo request messages to a target device and waits for an echo reply.
  - **Traceroute**: traces the path packets take from one networked device to another. It sends packets with increasingly higher TTL (Time to Live) values and listens for ICMP Time Exceeded messages from routers along the path.

ICMP Redirect Message

- ICMP redirect is an error message sent by a router to the sender of an IP packet.
- Redirects are used when a router believes a packet is being routed incorrectly, and it would like to inform the sender that it should use a different router for the subsequent packets sent to that same destination. ICMP redirect can be used by attackers to change a victim’s routing.

ICMP Countermeasure in Ubuntu

- In the Ubuntu operating system, there is a countermeasure against the ICMP redirect attack. In the Compose file, we have already turned off the countermeasure by configuring the victim container to accept ICMP redirect messages.

```
// In docker-compose.yml
sysctls:
        - net.ipv4.conf.all.accept_redirects=1
// To turn the protection on, set its value to 0
# sysctl net.ipv4.conf.all.accept_redirects=0
```

MTR command

The `mtr` command stands for "My Traceroute." It's a network diagnostic tool that combines the functionality of the `traceroute` and `ping` commands.

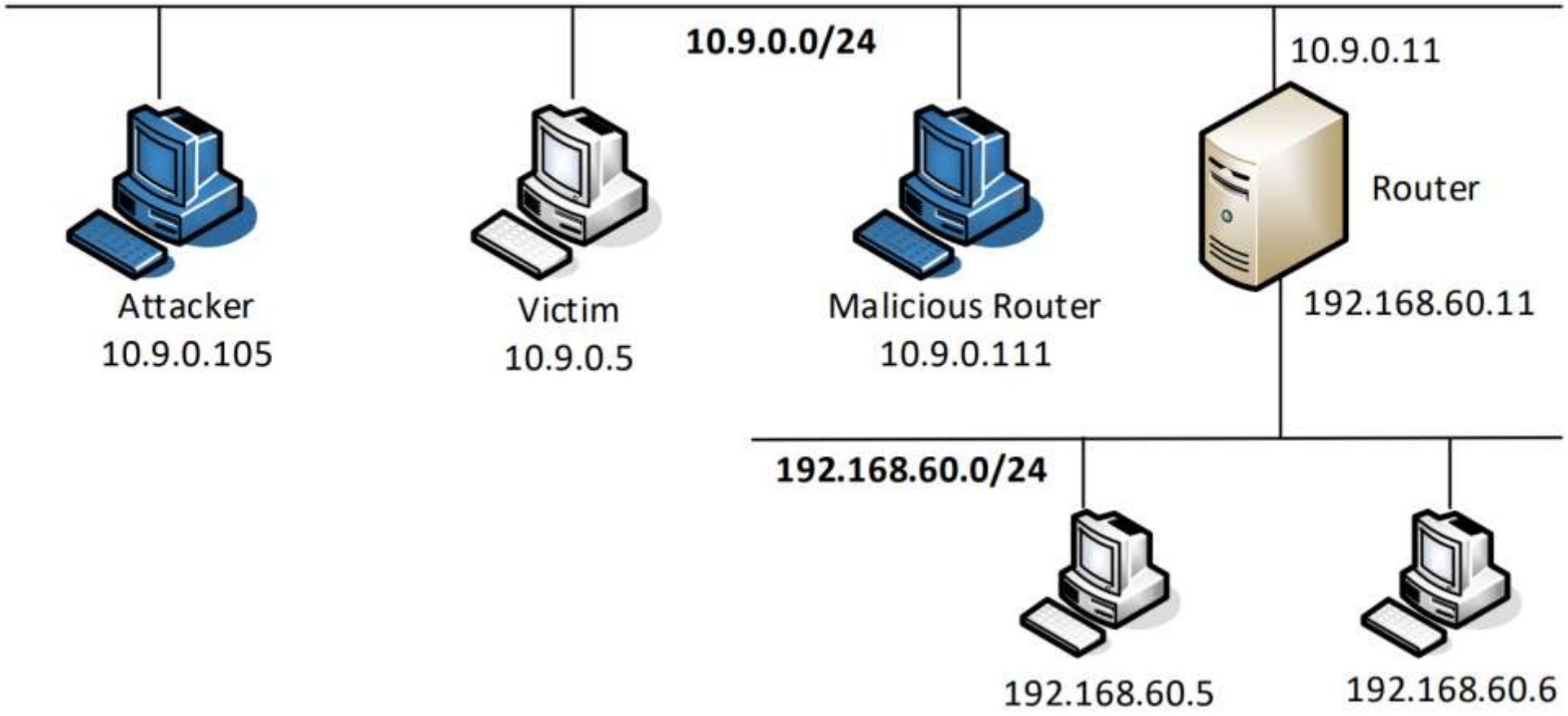
TRACEROUTE

Sends a series of packets towards the destination with increasing Time-to-Live (TTL) values. Each router along the path decrements the TTL of the packet until it reaches 0, at which point the router sends back an ICMP Time Exceeded message. By analyzing these messages, traceroute can determine the path taken by the packets

MTR

combines the functionality of traceroute and ping. Like traceroute, it sends packets towards the destination with increasing TTL values, but unlike traceroute, it continuously pings each hop along the route, providing ongoing statistics about packet loss and latency to each hop.

Lab Setup



TASK 1

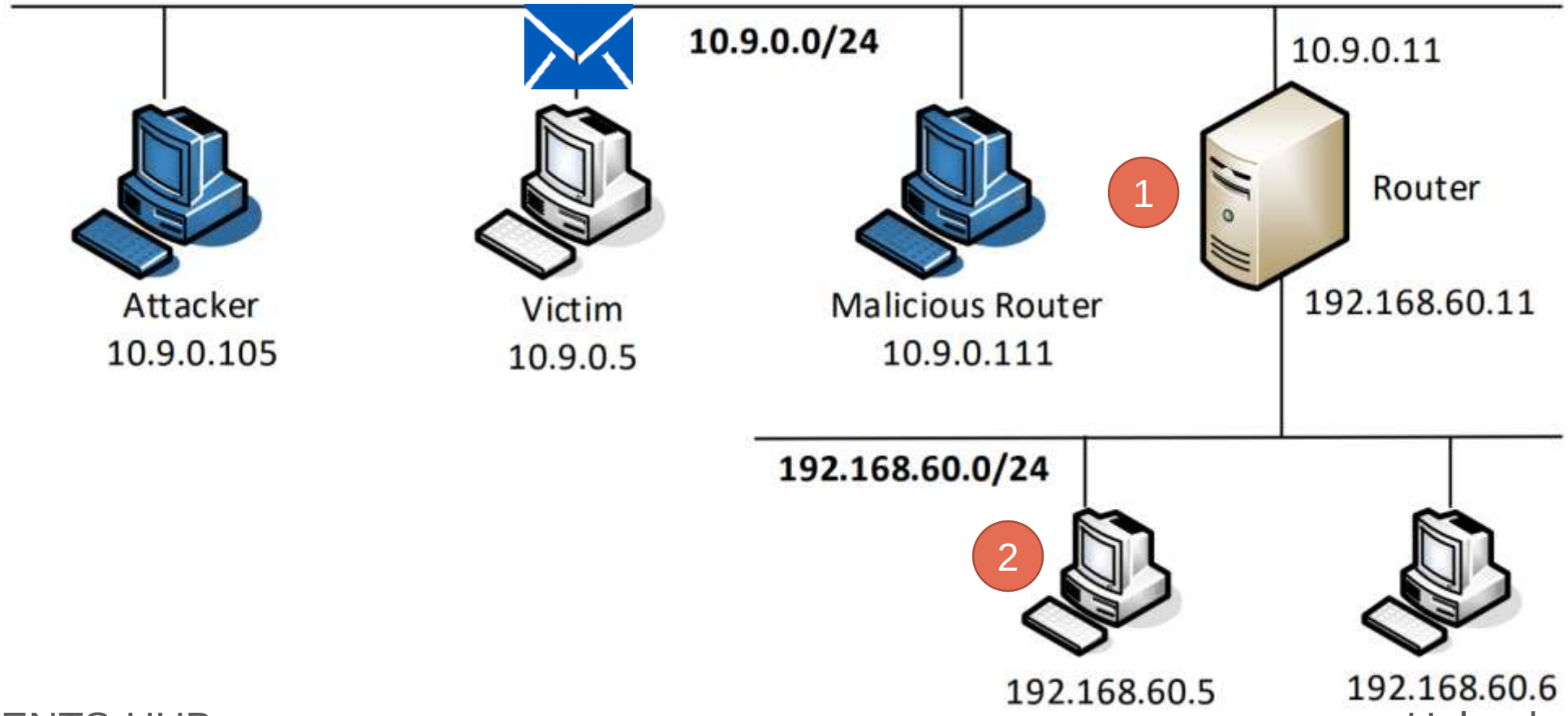
Launching ICMP Redirect Attack

Task 1: Launching ICMP Redirect Attack

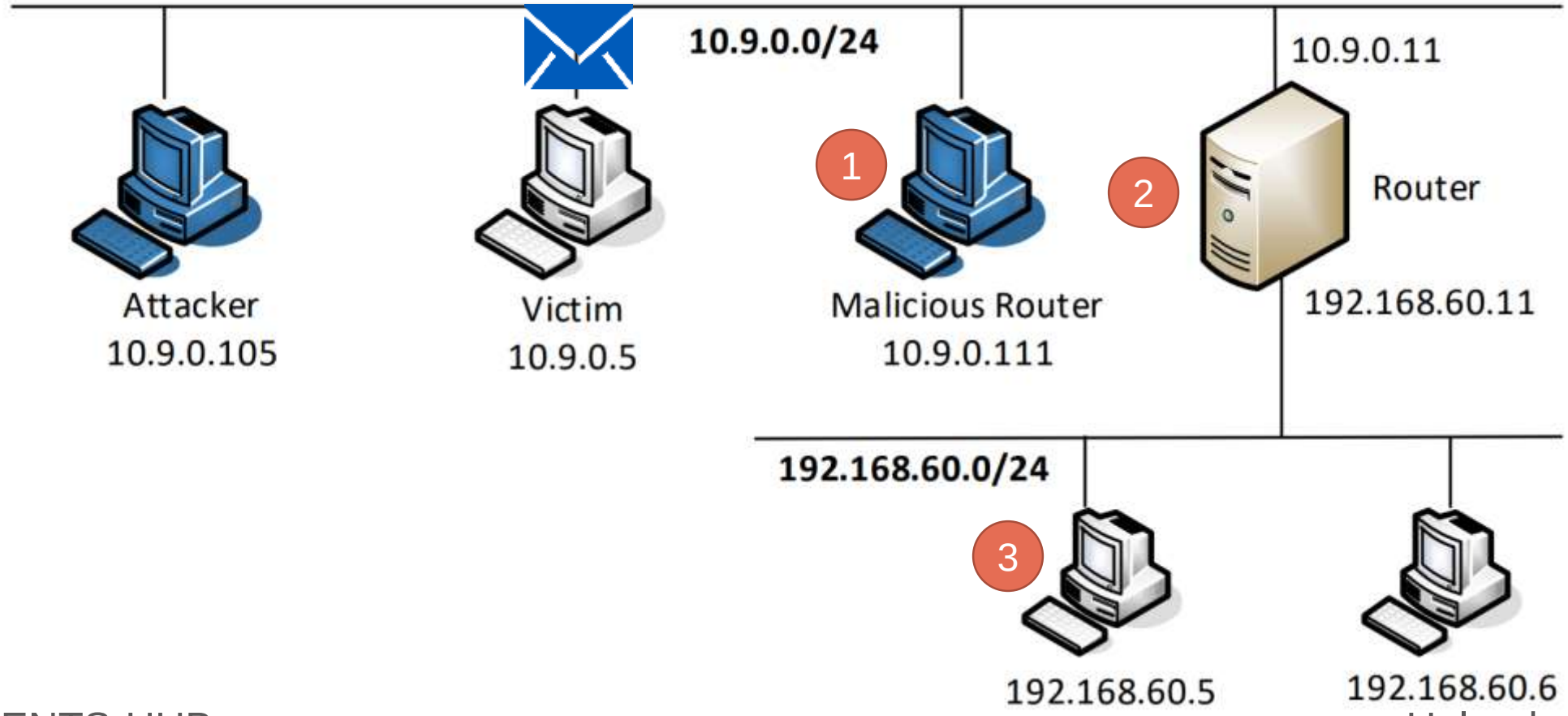
- For this task, we will attack the victim container from the attacker container.
- In the current setup, the victim will use the router container (192.168.60.11) as the router to get to the 192.168.60.0/24 network. If we run ip route on the victim container, we will see the following:

```
# 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
```

Expected Packet Path



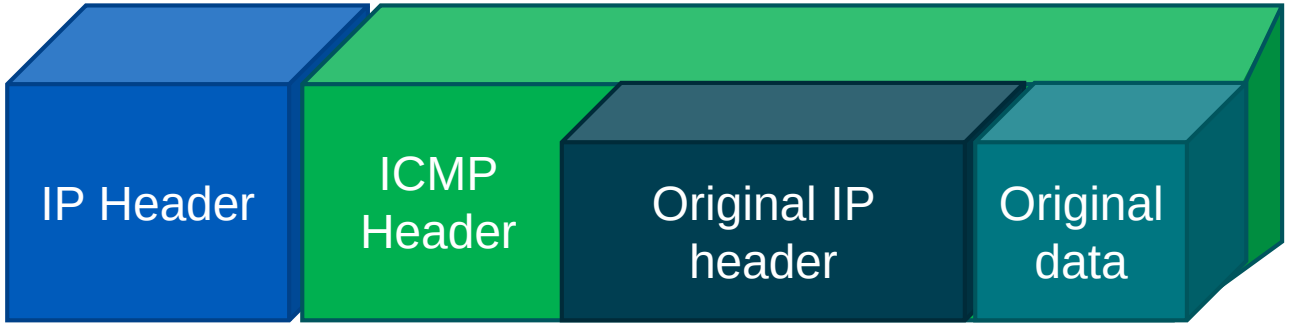
The Actual Packet Path After the Attack



ICMP Redirect Message Structure

The ICMP redirect message consists of 4 main components:

1. IP Header: contains the target and source IPs.
2. ICMP Header: contains the information needed for the redirection such as the correct gateway and the ICMP type and code.
3. Original IP Header: represents the original IP header that was sent.
4. Original Data: represents the original request by the sender.



Scapy Code Snippet

- A code skeleton is provided in the following, with some of the essential parameters left out. Students should fill in the proper values in the places marked by @@@@.

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

ip = IP(src=@@@@, dst=@@@@)
icmp = ICMP(type=@@@@, code=@@@@)
icmp.gw = @@@@
# The enclosed IP packet should be the one that triggers the redirect message.
ip2 = IP(src=@@@@, dst=@@@@)
send(ip/icmp/ip2/ICMP())
```

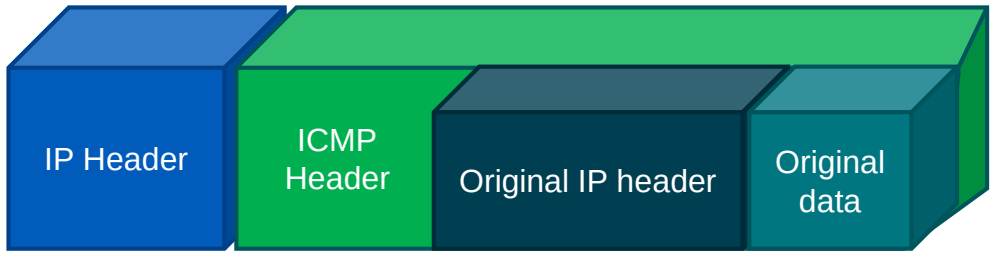
More Information About the Code Snippet

ICMP Type	Message
0	Echo reply
3	Destination unreachable
4	Source quench
5	Redirect
8	Echo
9	Router advertisement
10	Router selection
11	Time exceeded
12	Parameter problem
13	Timestamp
...	...

ICMP Code	Message
0	Net is unreachable
1	Host is unreachable
2	Protocol is unreachable
3	Port is unreachable
4	Fragmentation is needed
5	Source route failed
6	Destination network is unknown
7	Destination host is unknown
8	Source host is isolated
9	Communication is prohibited
...	...

More Information About the Code Snippet

- `ICMP.gw`
  - gw Stands for "gateway".
  - It specifies the IP address of the new gateway (router) that the sender should use for the specified destination network.
- `send( ip/icmp/ip2/ICMP() )`
  - Corresponds to the following ICMP message structure:



Our Goal

- First of all we need to execute the following command before our attack and take a screenshot:

```
mtr -n 192.168.60.5
```

- After that we need to constantly ping the target host form our victim container:

```
ping 192.168.60.5 > /dev/null &
```

- Then we need to modify the code snippet to help the attacker impersonate the router's identity, and tell the victim that its last attempt should be redirected to the malicious router.
- Then execute the code inside the attacker container.
- Execute mtr command again, observe the difference, and take a screenshot:

```
mtr -n 192.168.60.5
```

Question 1

- Before proceeding we need to flush the routing cache:

```
ip route flush cache
```

- 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.

Question 2

- Before proceeding we need to flush the routing cache:

```
ip route flush cache
```

- 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.

Question 3

- Before proceeding we need to flush the routing cache:

```
ip route flush cache
```

- If you look at the `docker-compose.yml` file, you will find the following for the malicious router container:

```
    sysctls:
        - net.ipv4.conf.all.send_redirects=0
        - net.ipv4.conf.default.send_redirects=0
        - net.ipv4.conf.eth0.send_redirects=0
```

- What are the purposes of these entries? Please change their value to 1, and launch the attack again. Please describe and explain your observation.
- Make sure to change their value back to 0 before moving to the next task.

TASK 2

Launching the MITM Attack

BIRZEIT UNIVERSITY

Task 2: Launching the MITM Attack

- Using the ICMP redirect attack, we can get the victim to use our malicious router (`10.9.0.111`) as the router for the destination `192.168.60.5`. Therefore, all packets from the victim machine to this destination will be routed through the malicious router. We would like to modify the victim's packets. Before launching the MITM attack, we start a TCP client and server program using netcat. See the following commands.
- On the destination container 192.168.60.5, start the netcat server:

```
nc -lp 9090
```

- On the victim container, connect to the server:

```
nc 192.168.60.5 9090
```

IP forwarding on Hosts

- IP forwarding on hosts refers to the capability of a computer to forward network packets between its network interfaces. In typical networking setups, hosts (computers) have multiple network interfaces, such as Ethernet, Wi-Fi, or virtual interfaces. When IP forwarding is enabled on a host, it allows the host to act as a simple router, forwarding packets between these interfaces.
- We need to disable IP forwarding on the malicious router:

```
sysctl net.ipv4.ip_forward=0
```

Man in the Middle (MITM) Code

- Once the IP forwarding is disabled, our program needs to take over the role of packet forwarding from the victim to the target, of course after making changes to the packets.
- Since the packet's destination is not for us, the kernel will not give the packet to us; it will simply drops the packet. However, if our program is a sniffer program, we will get the packet from the kernel. Therefore, we will use the sniff and-spoof technique to implement this MITM attack. In the following, we provide a sample sniff-and-spoof program, which captures TCP packets, makes some changes, before sending them out.
- You can find the code from the lab setup files.

Question 4

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

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.