ACS 545 Cryptography and Network Security

Lab 3: ARP Cache Poisoning Attack Lab

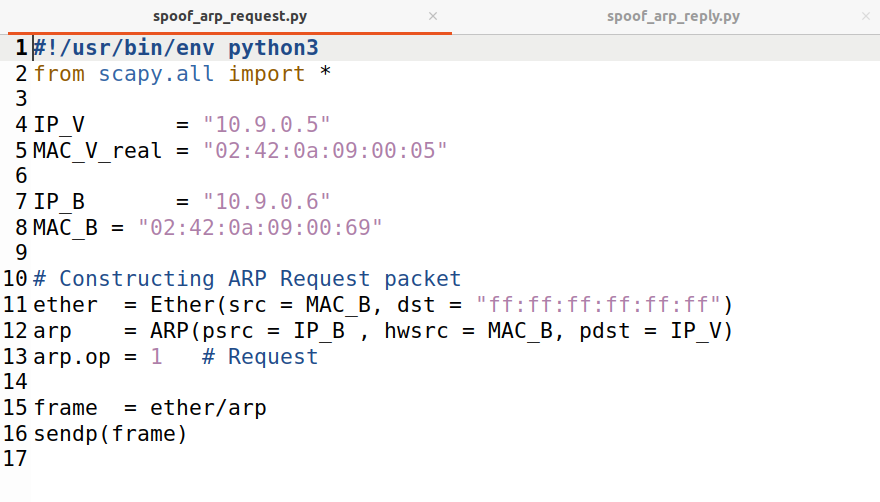
Name: Vijayagiridharan Subramanian

**Task 1: ARP Cache Poisoning**

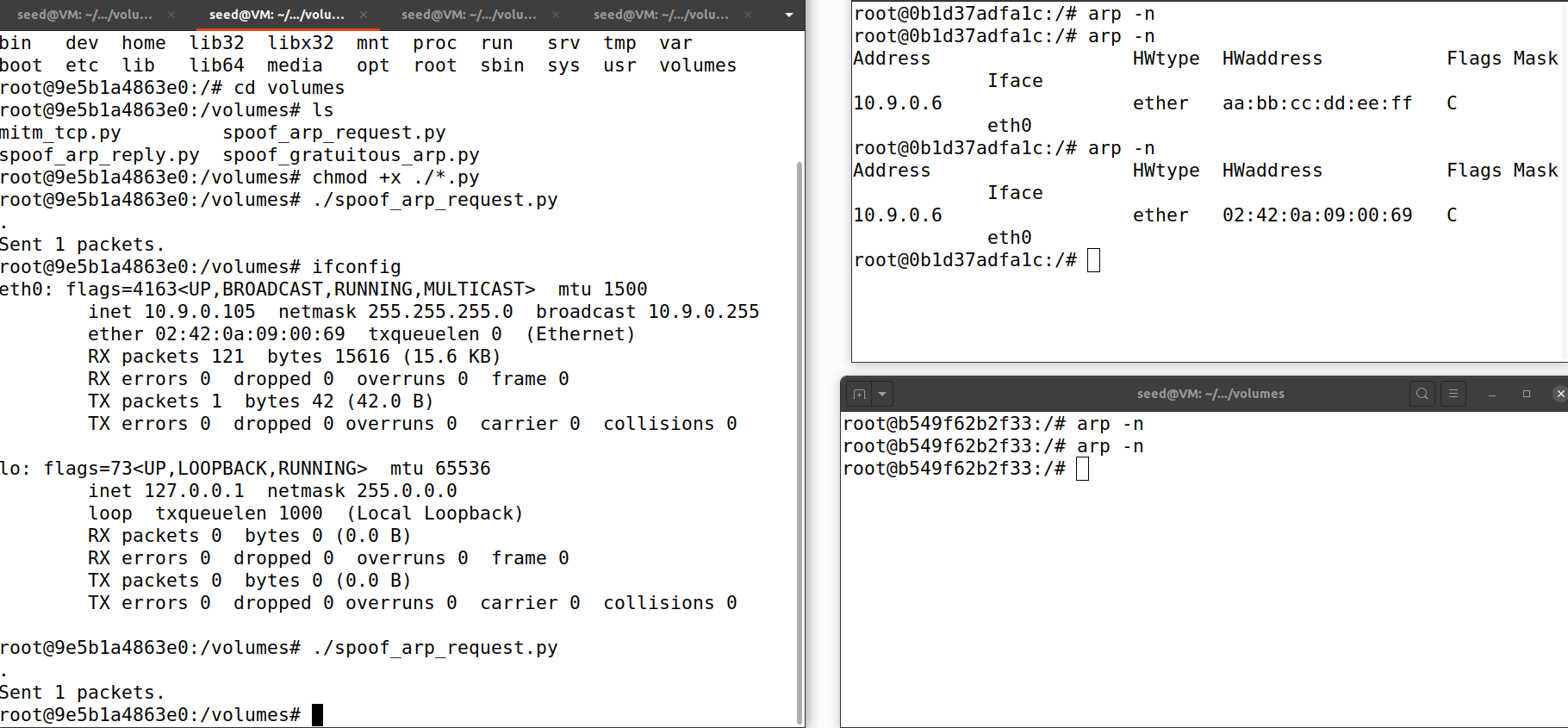
**Task 1.A ( using ARP request)**

The task was to construct an ARP request packet on host M to map B’s IP address to M’s MAC address, send the packet to host A, and check whether the attack is successful or not.

**CODE:**



**Implementation:**



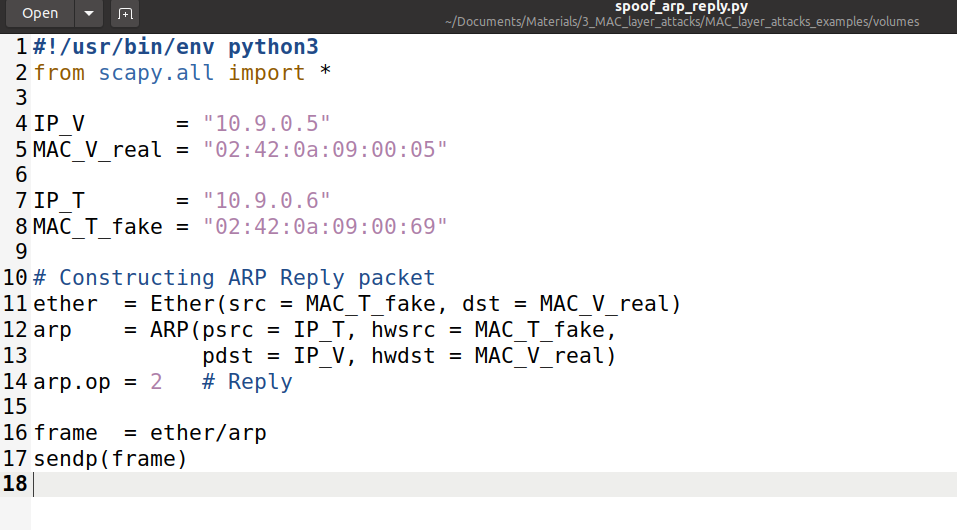
The left side tab is attacker M, right side top table is host A and down one is host B.

When we run spoof\_arp\_request.py on the left tab. And while checking arp -n cache is displayed. We can find Hardware address change before and after the attack.

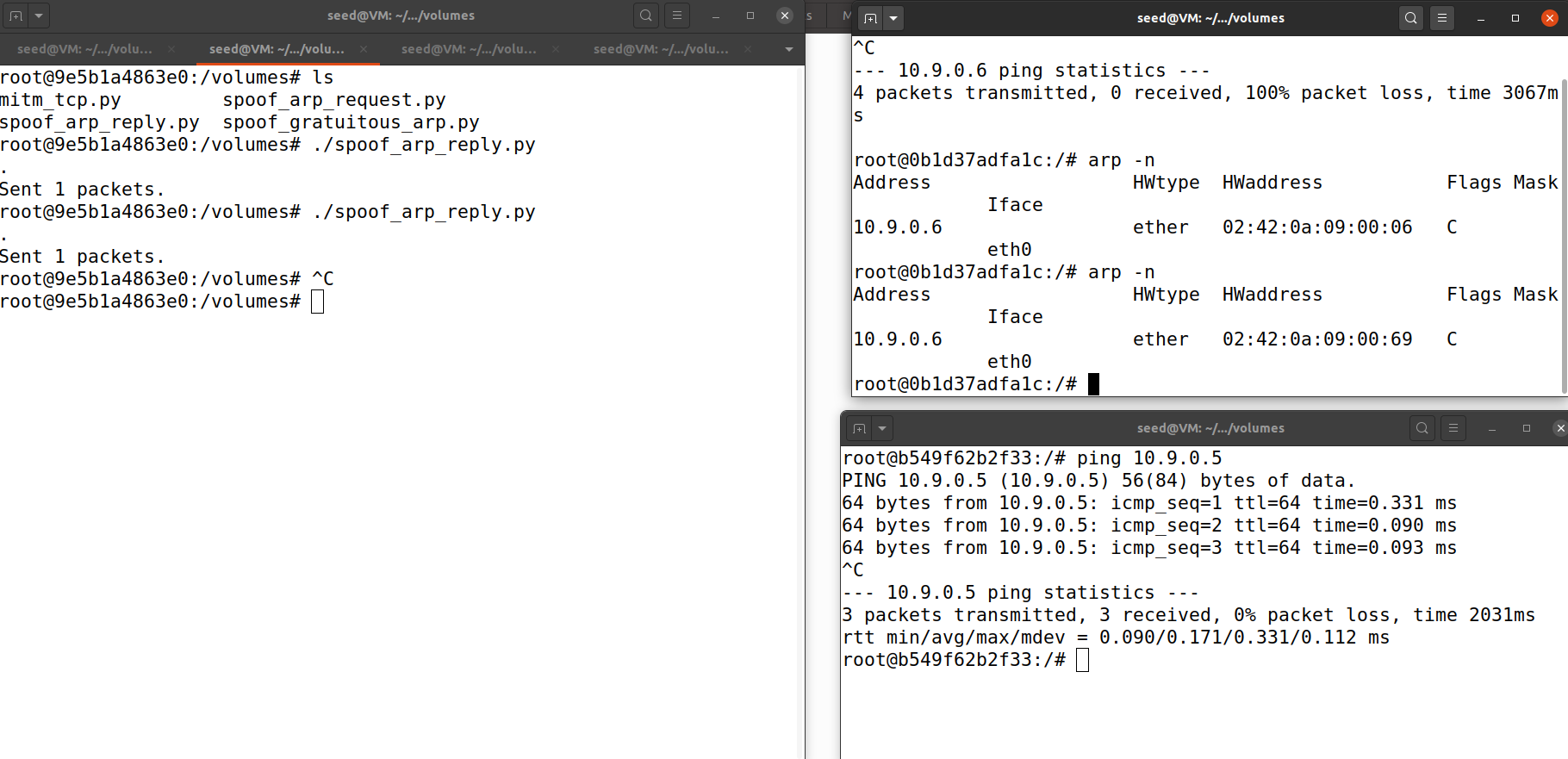
We construct an ARP request packet on host M to map B’s IP address to M’s MAC address, send the packet to host A, and the attack is successful.

**Task 1.B (using ARP reply)**

**CODE:**

****

**Scenario 1: B’s IP is already in A’s cache**

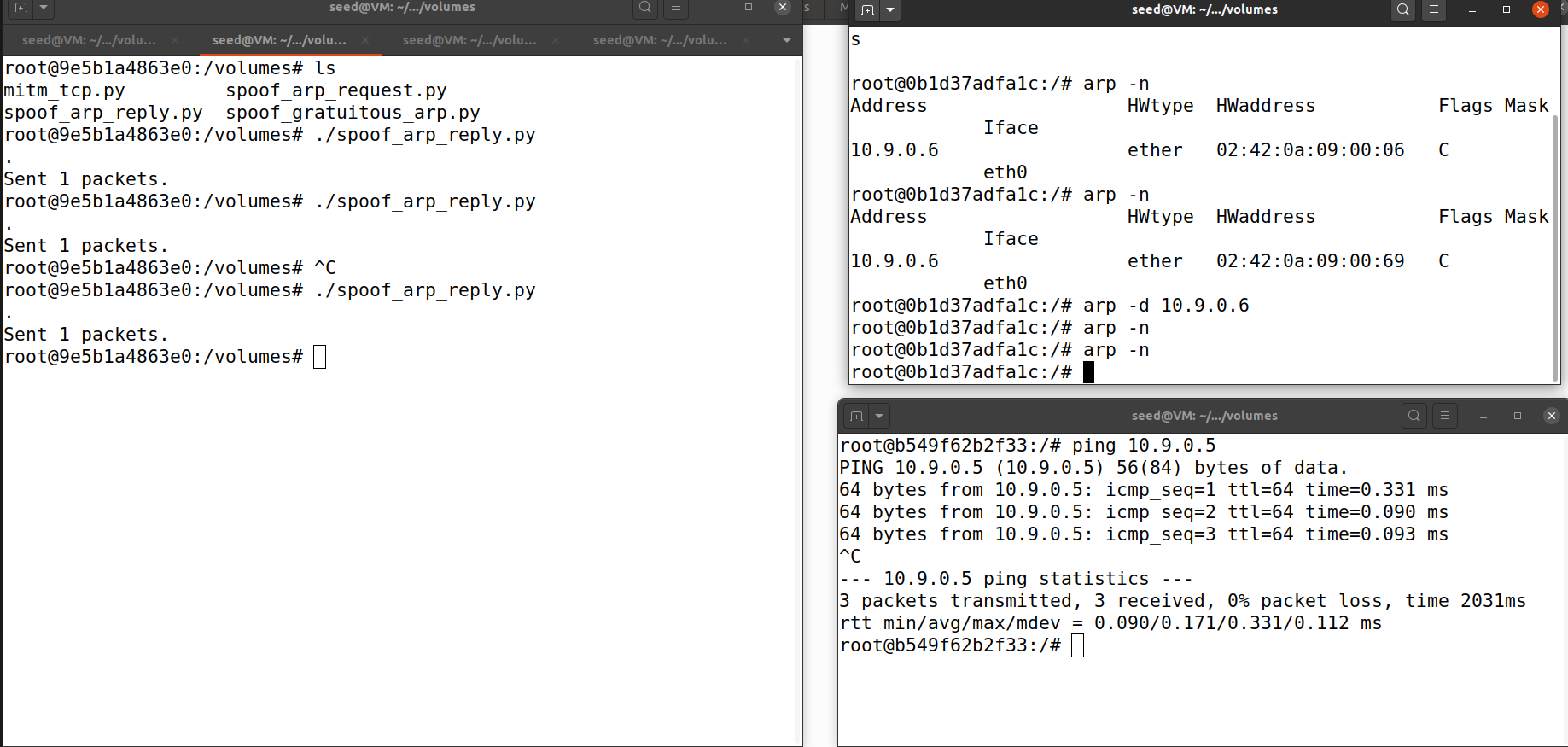
****

The left side tab is attacker M, right side top table is host A and down one is host B.

We ping 10.9.0.5 in Host B and run the program in M, Above host A shows ARP cache poisoning occurred there.

Here ARP cache poisoning occurs because of caches already present

**Scenario 2: B’s IP is not in A’s cache.**



Here there is no poison because of ARP cache because we removed it with arp -d 10.9.0.6

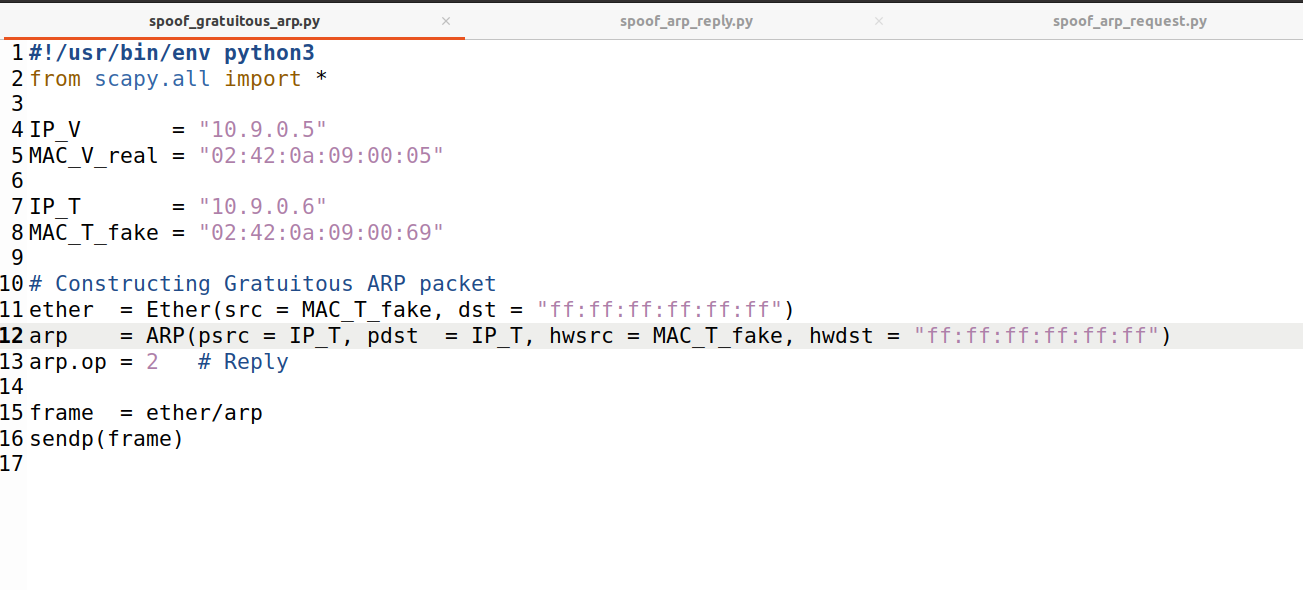
The left side tab is attacker M, right side top table is host A and down one is host B.

On host M, constructed an ARP reply packet to map B’s IP address to

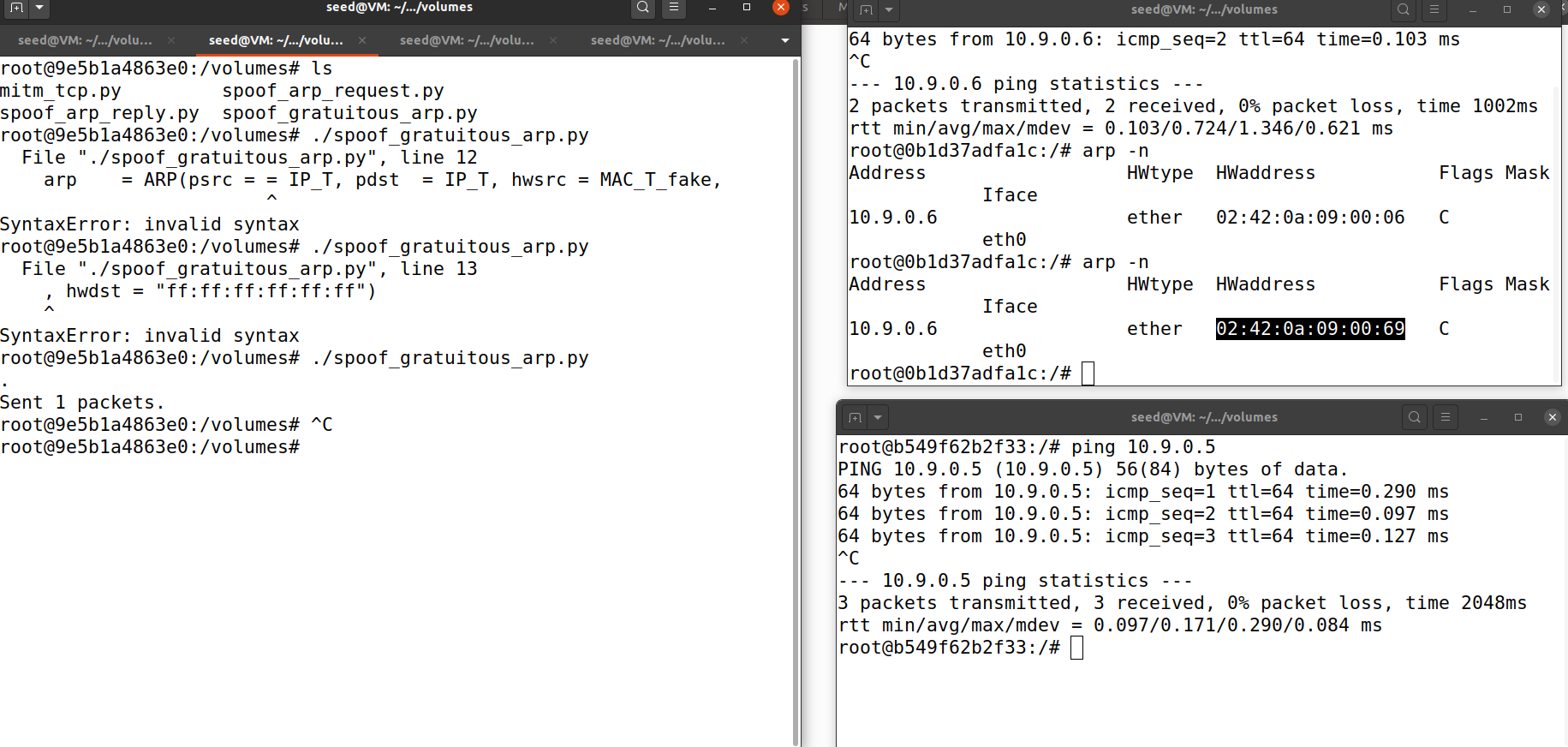
M’s MAC address and Sended the packet to A and showed the attack was successful .

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

**CODE:**

****

**Scenario 1: B’s IP is already in A’s cache.**

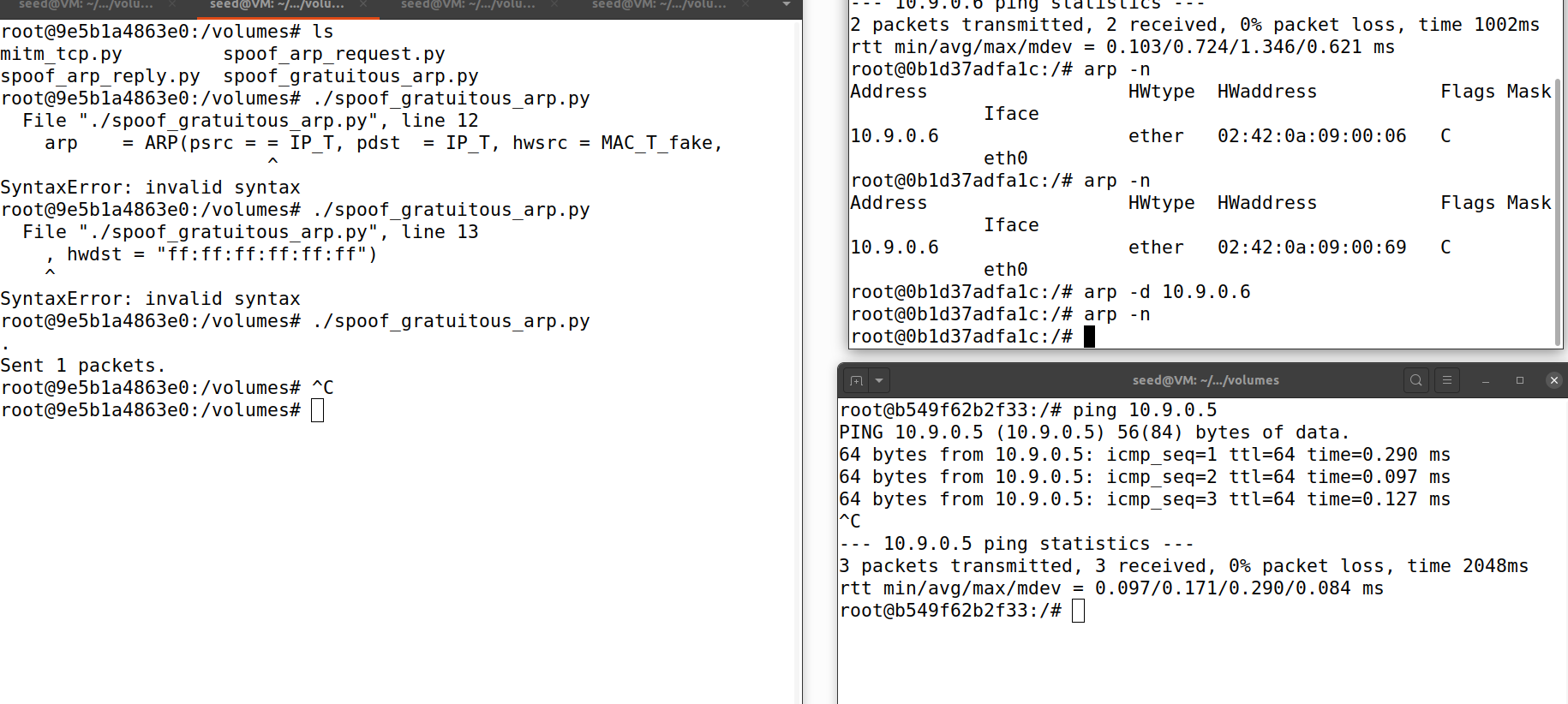
****

The left side tab is attacker M, right side top table is host A and down one is host B.

We ping 10.9.0.5 in Host B and run the program in M, Above host A shows ARP cache poisoning occurred there. The attacker hardware address is found in host A.

Here ARP cache poisoning occurs because of caches already present

**Scenario 2: B’s IP is not in A’s cache.**

****

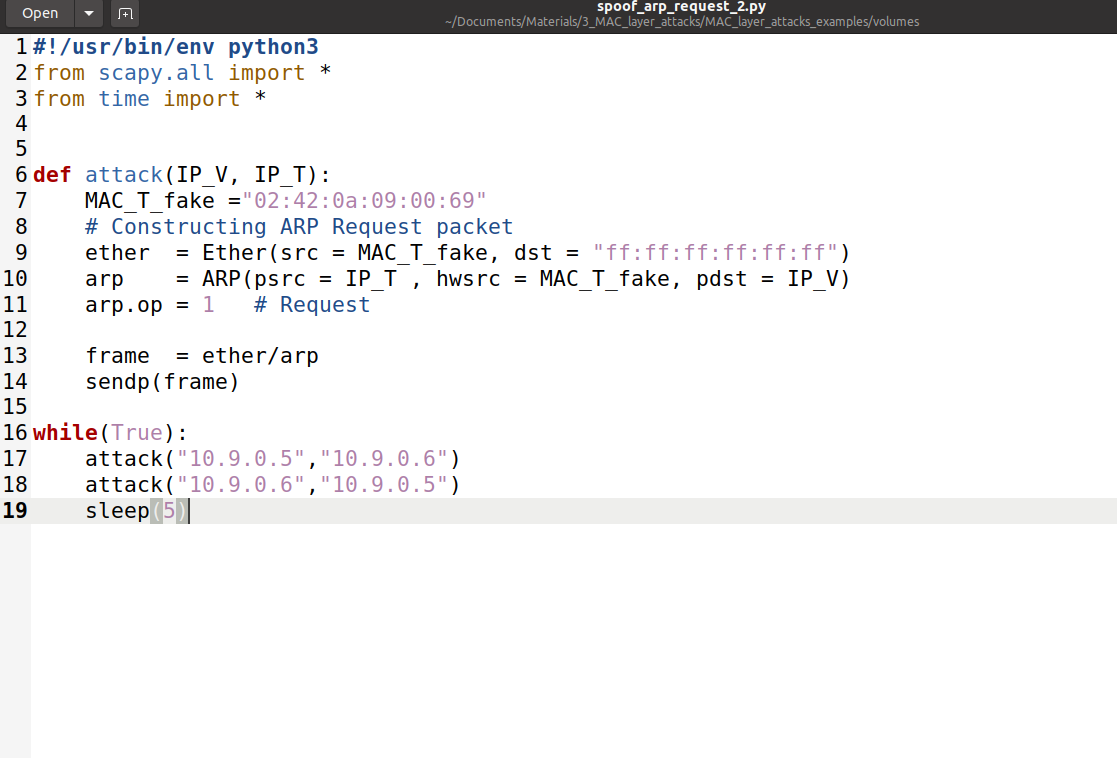
The left side tab is attacker M, right side top table is host A and down one is host B.

Here there is no poison because of the ARP cache because we removed it with arp -d 10.9.0.6. Then while running arp -n, no output is received and no poisoning occured.

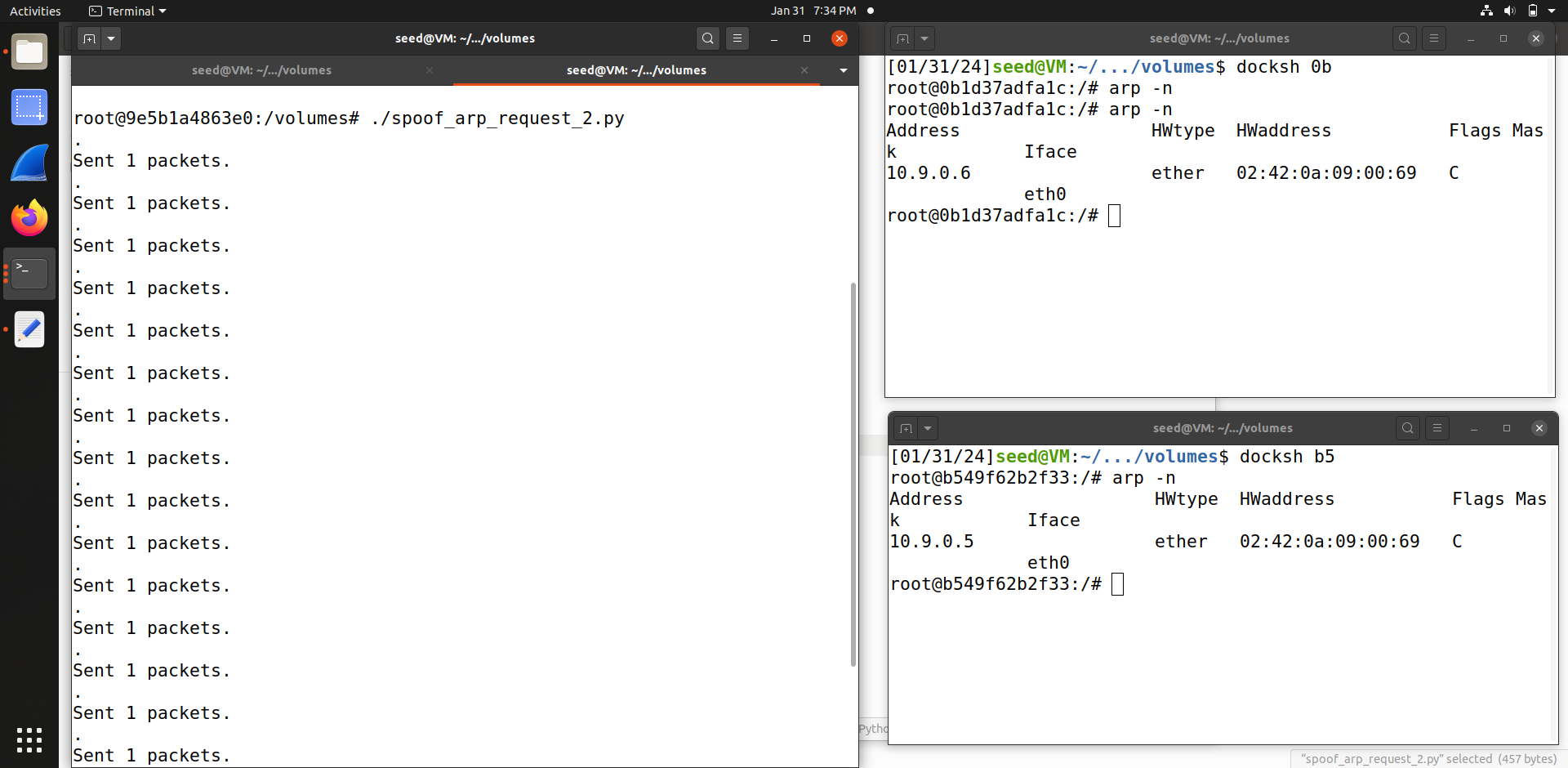
**Task 2: MITM Attack on Telnet using ARP Cache Poisoning**

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

**Code:**

****

**Implementation**

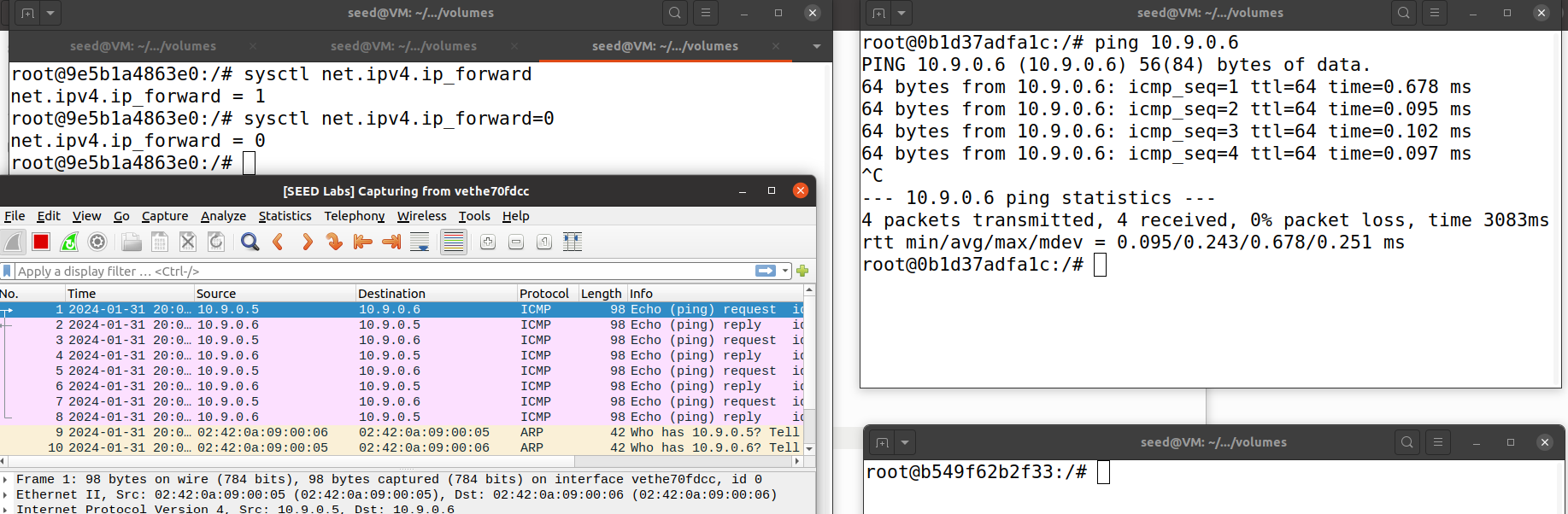
****

The left side tab is attacker M, right side top table is host A and down one is host B. In above code ARP request attack is done by the attacker, in host A, arp -n is used to check cache poisoning because of ARP. similarly In host B, we are checking for cache poisoning and find that the hardware address is changed.

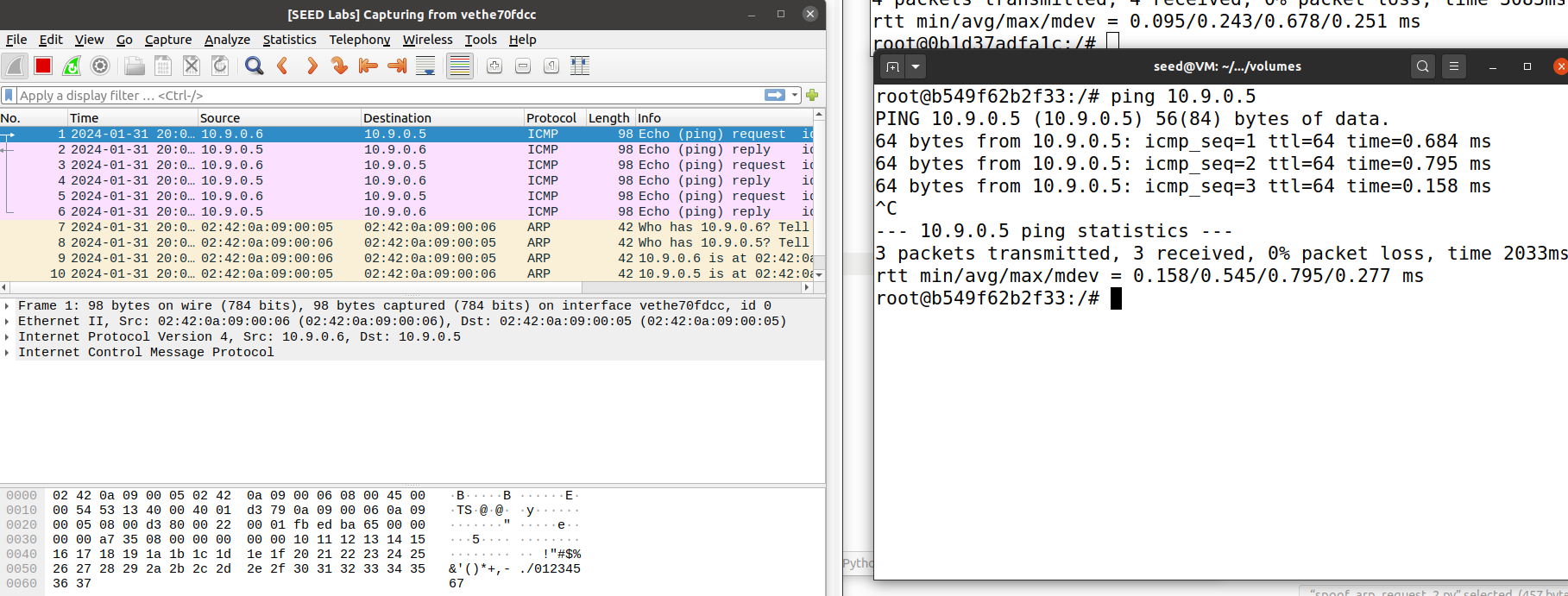
Once we run the code in the attacker (left), we can find the sending packet for every 5 seconds.

**Step 2 (Testing)**

**We are pinging in Host A :**

****

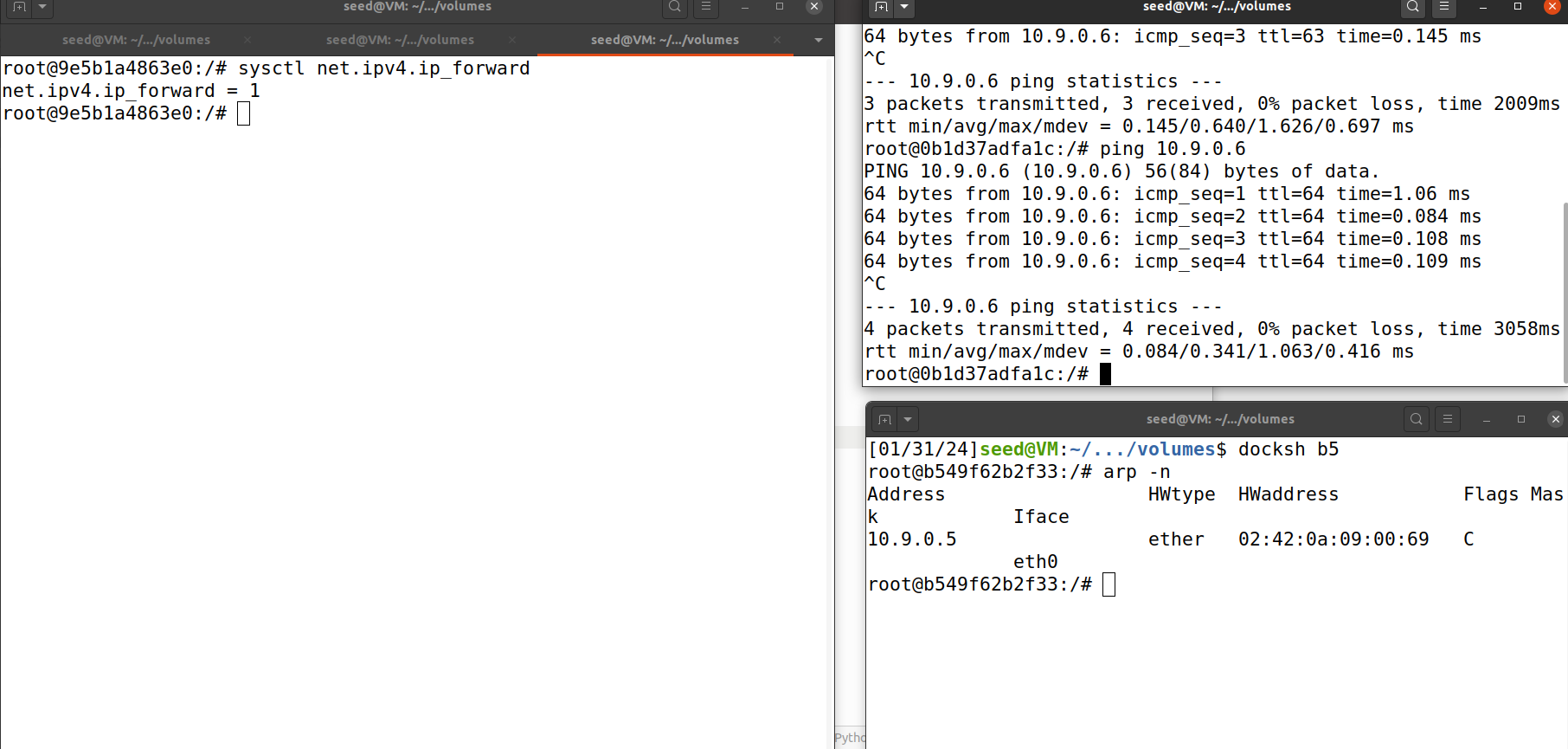
**Now we are pinging in host B:**

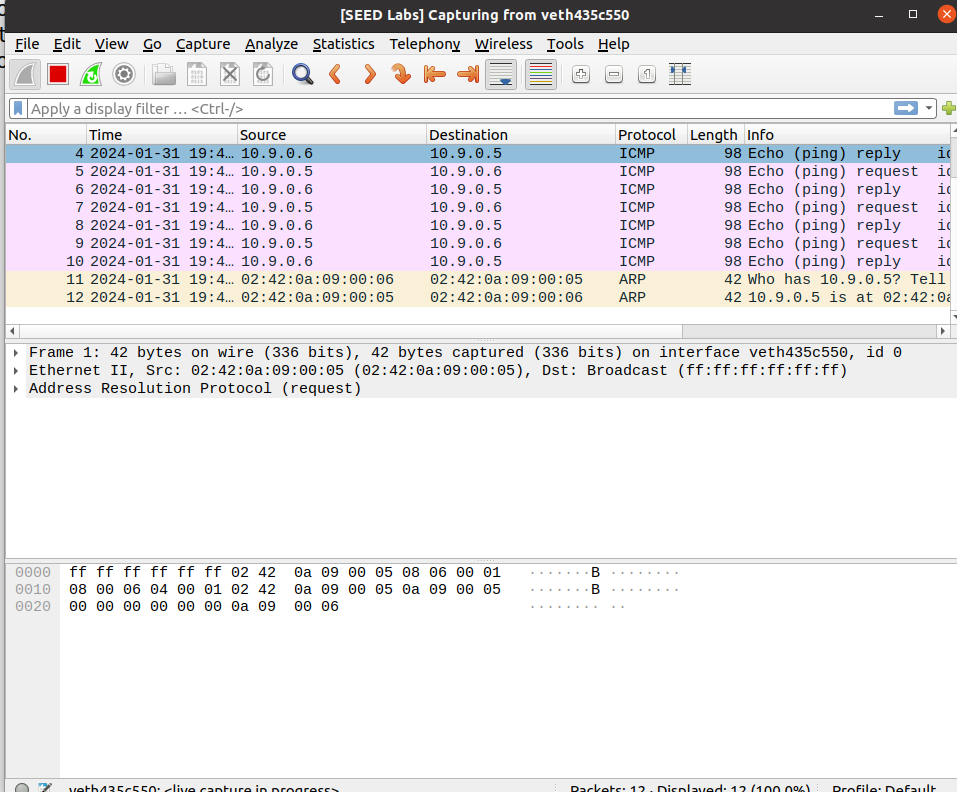
****

After the attack is successful, we try to ping each other between Hosts A and B, and corresponding wireshark results are mentioned on the right side. Now IP forwarding is OFF with command **sysctl net.ipv4.ip\_forward= 0** in the attacker

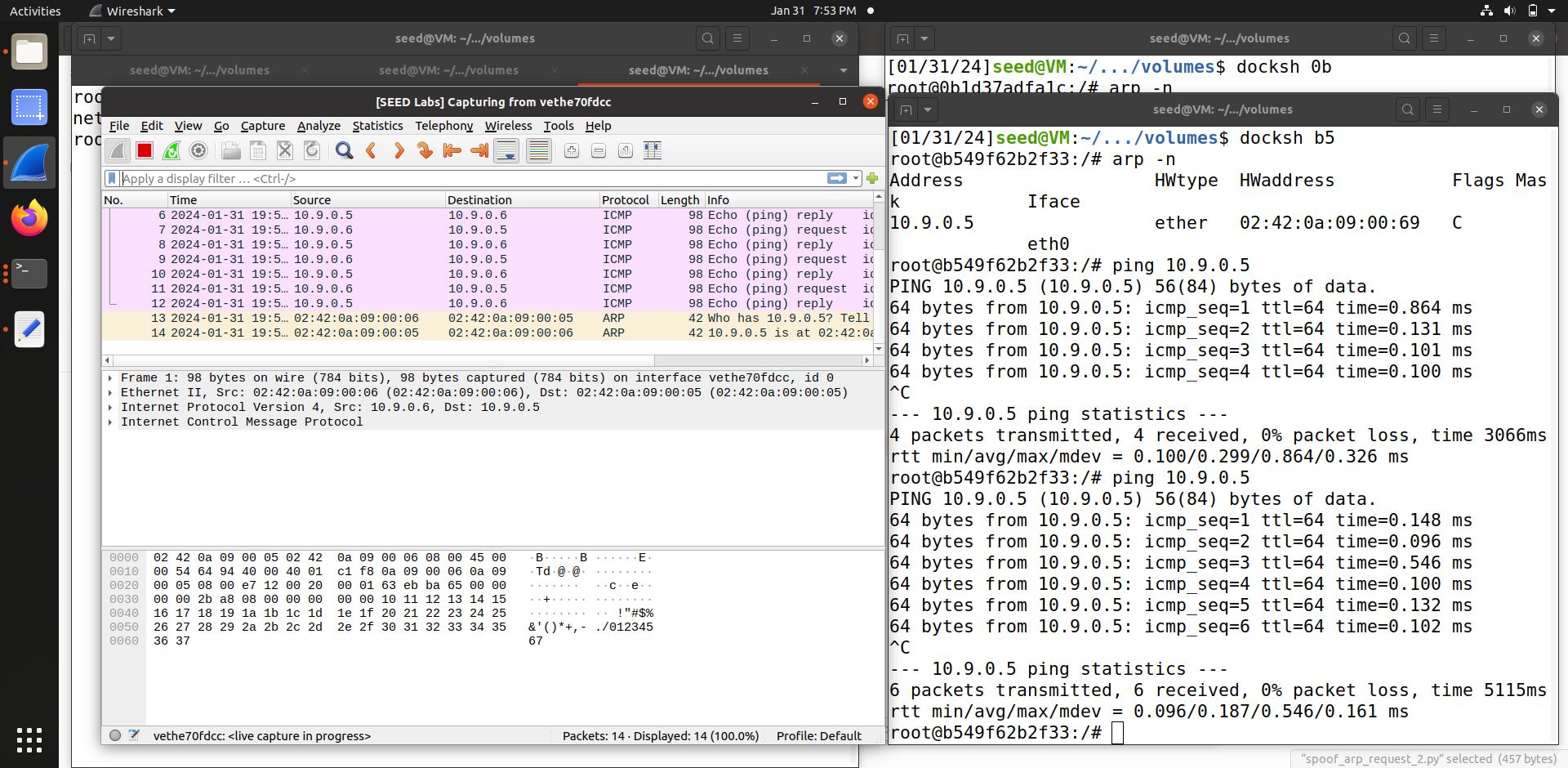
**Step 3 (Turn on IP forwarding).**

**Similarly Turned IP forwarding on and started to ping host A.**

****

****

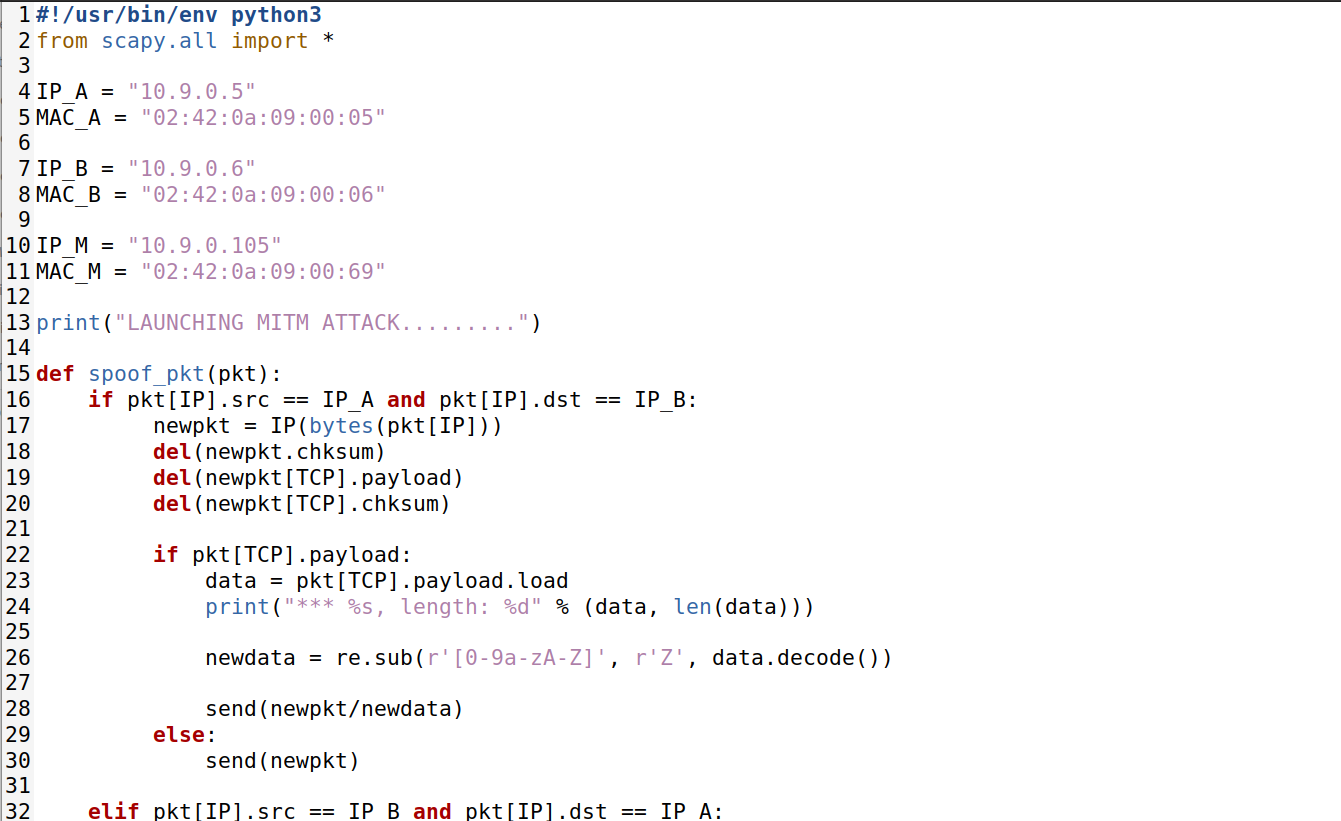
**Similarly Now We ping Host B.**

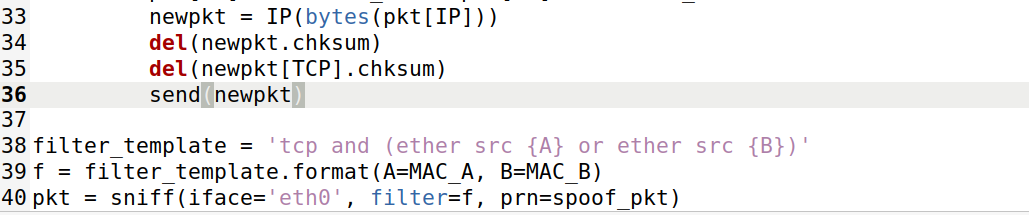
****

After the attack is successful, we try to ping each other between Hosts A and B, and corresponding wireshark results are mentioned on the right side. Now IP forwarding is ON with command **sysctl net.ipv4.ip\_forward= 1** in the attacker. We can find results on wireshark once we started pinging the address.

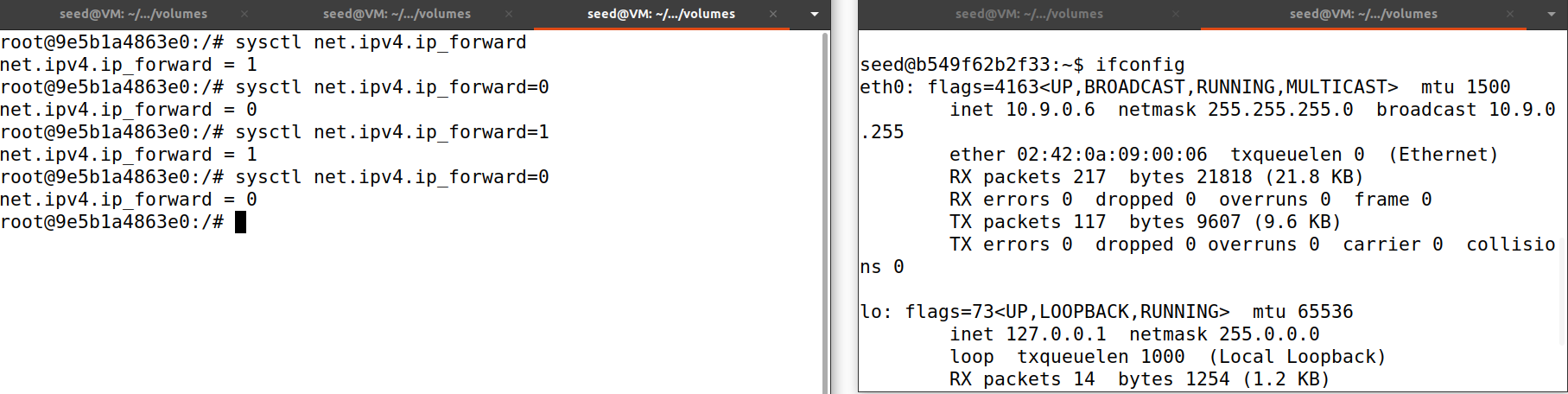
**Step 4 (Launch the MITM attack).**

**Code:**

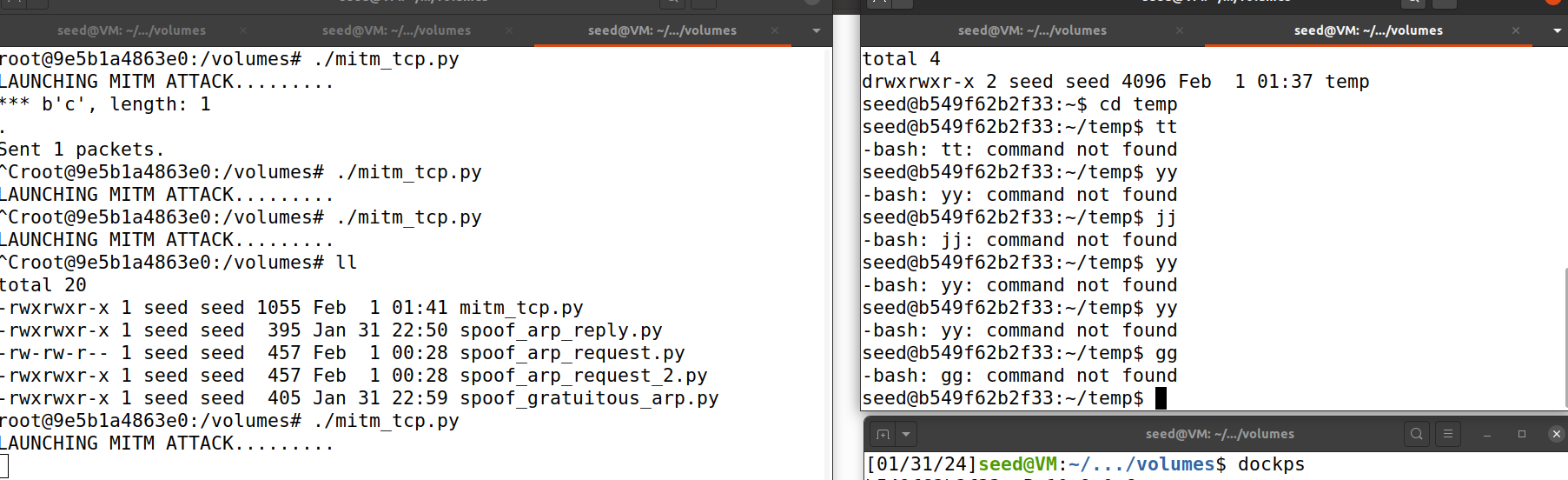
****

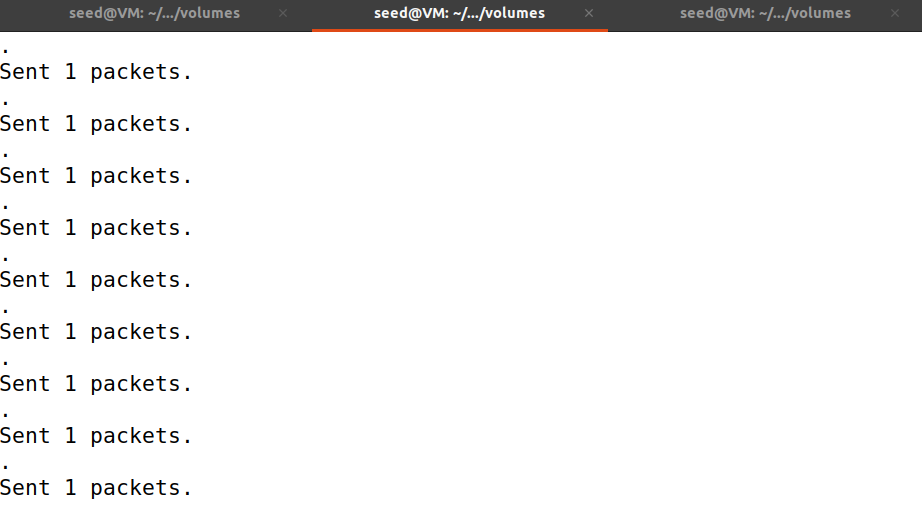
****

**We made changes in telnet . Here host A and host B are communicating with Telnet, Left side is the attacker and right side is host.**

****

Now we launch MITM attack (MAn in the middle attack), we started to put random values in rightside host and sent 1 packet message in the attacker.

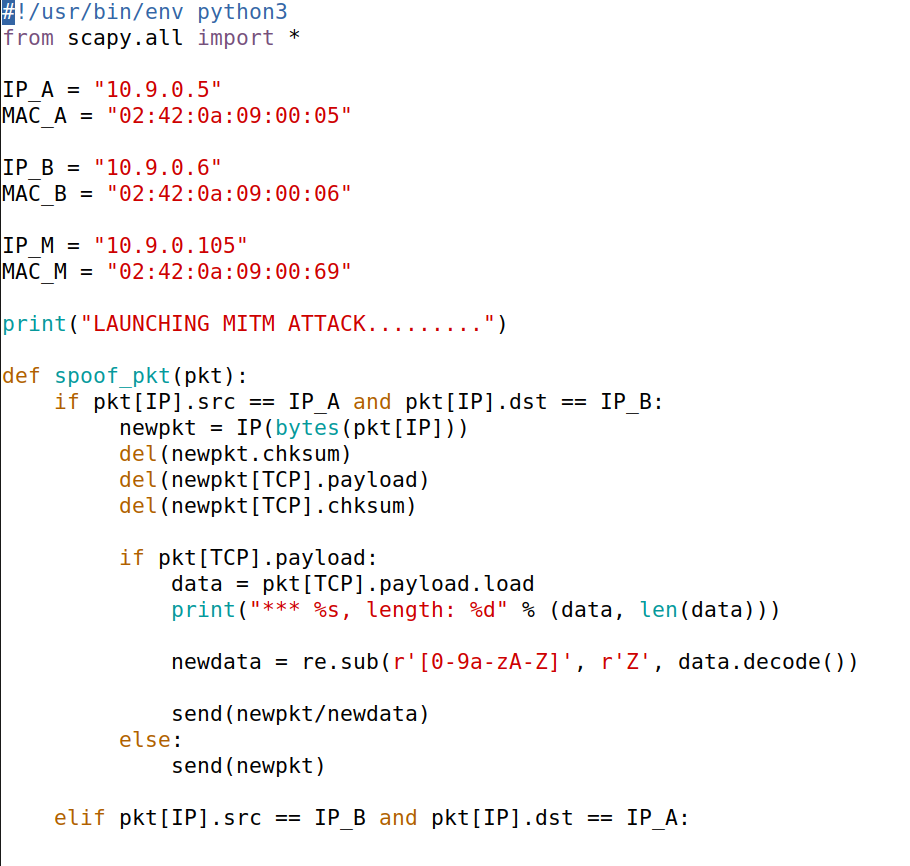
****

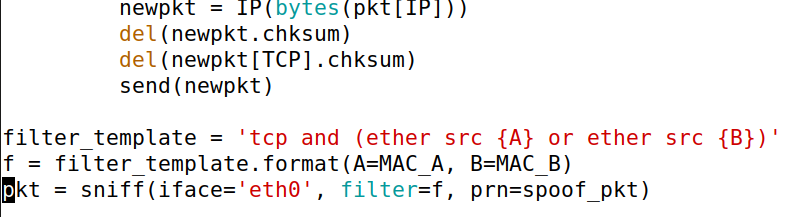
****

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

**CODE:**

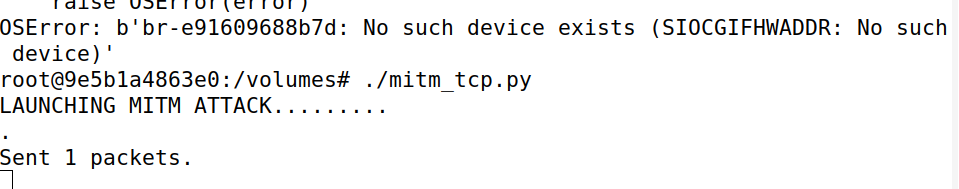
**Below is the code**

****

****

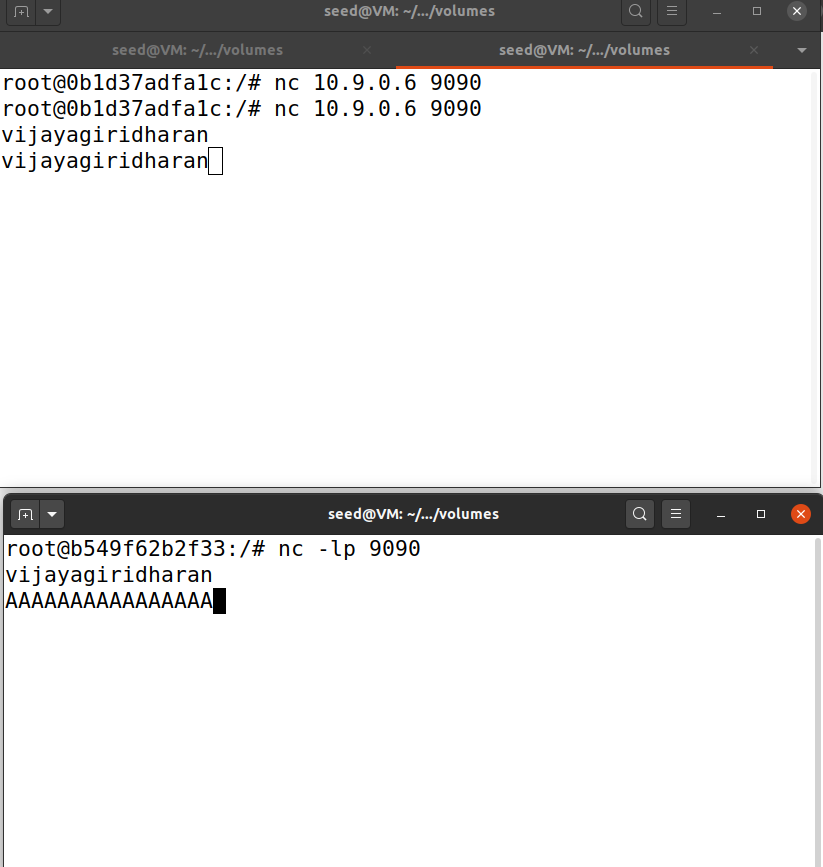
Instead of using telnet for communication, hosts A and B are use netcat. In order to alter the data exchanged between A and B, host M wishes to eavesdrop on their conversation. WE link A and B using a netcat TCP connection using the following commands:

**This is Attacker Container M: we started the MITM attack by running the command.**

****

Below are Host A on top and Host on bottom. Now we run command nc -lp 9090 in Host B and then we run nc 10.9.0.6.9090 in Host A. we enter some values in Host A which is reflected on the other host . But after the MITM attack , the values are replaced with the same number of A’s in other host.

**Below are before and after MITM attacks:**

****