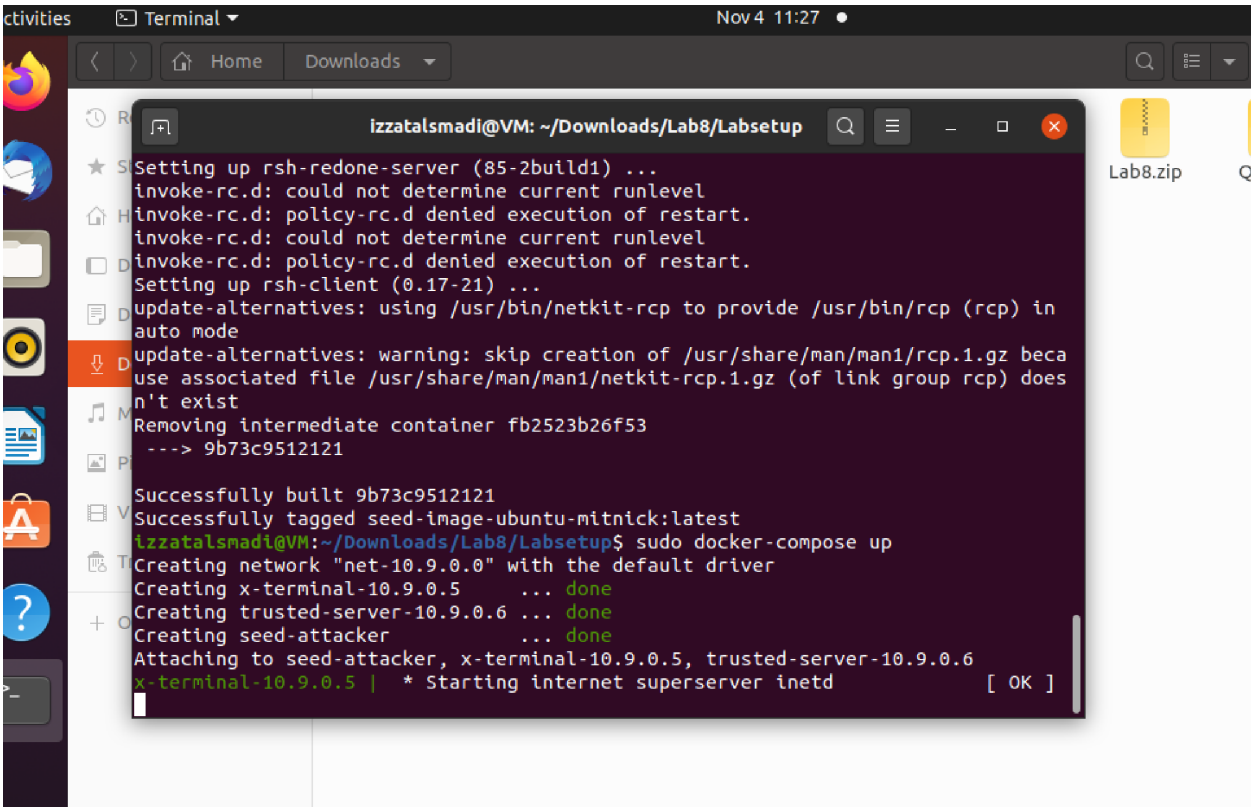
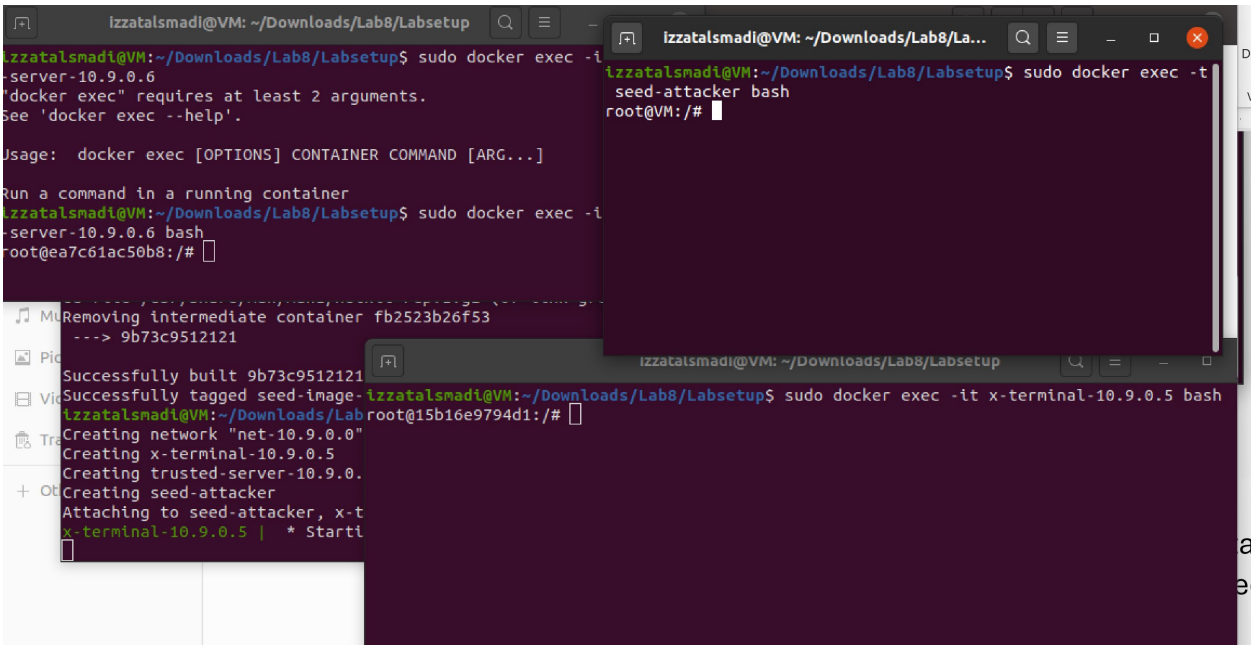
Lab8 Demo 2024

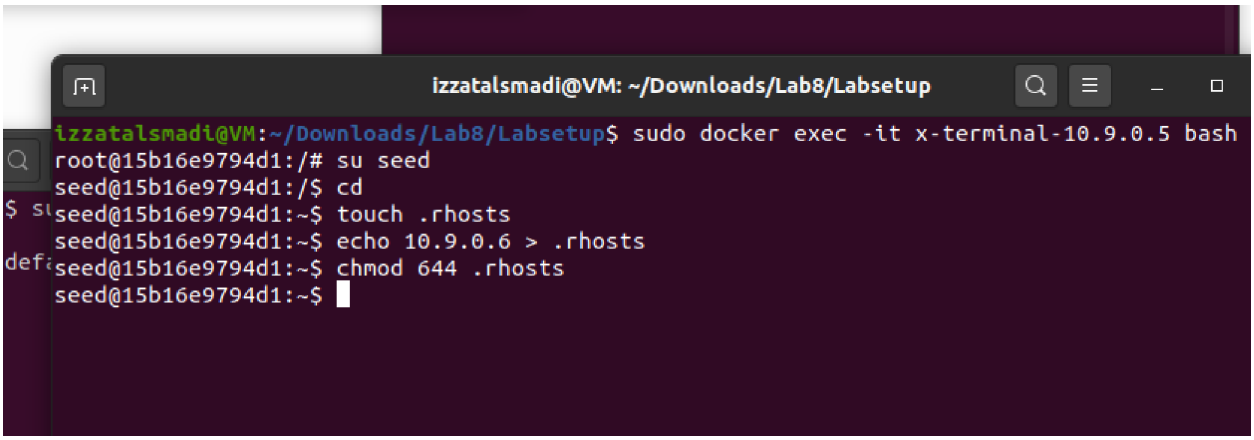
- Build and start the container



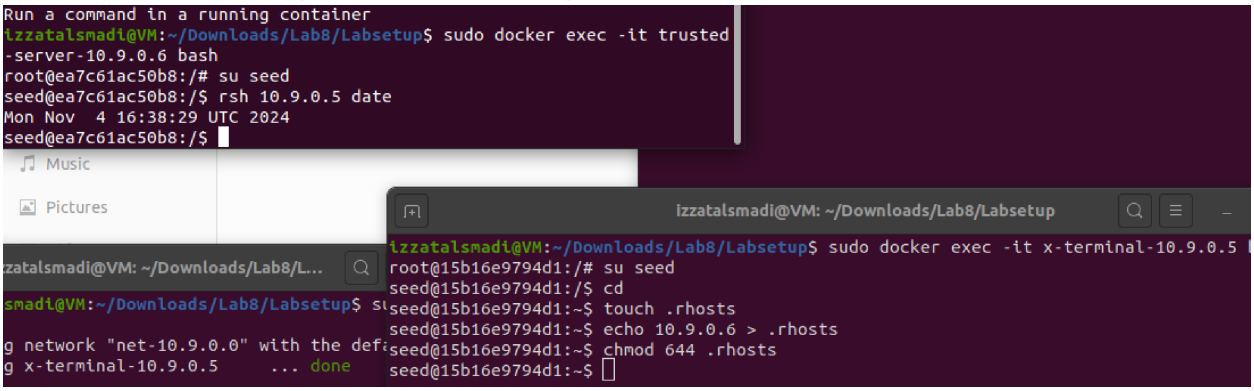
- Notice the network has 3 machines
  1. Attacker-10.9.0.105 2. Trusted-server-10.9.0.6 3. X-terminal-10.9.0.5 Before the attack, we need to set up the trusted relationship between X-Terminal (10.9.0.5) and Trusted Server (10.9.0.6).
  2. Login to x-terminal and trusted server
- Start the VMs



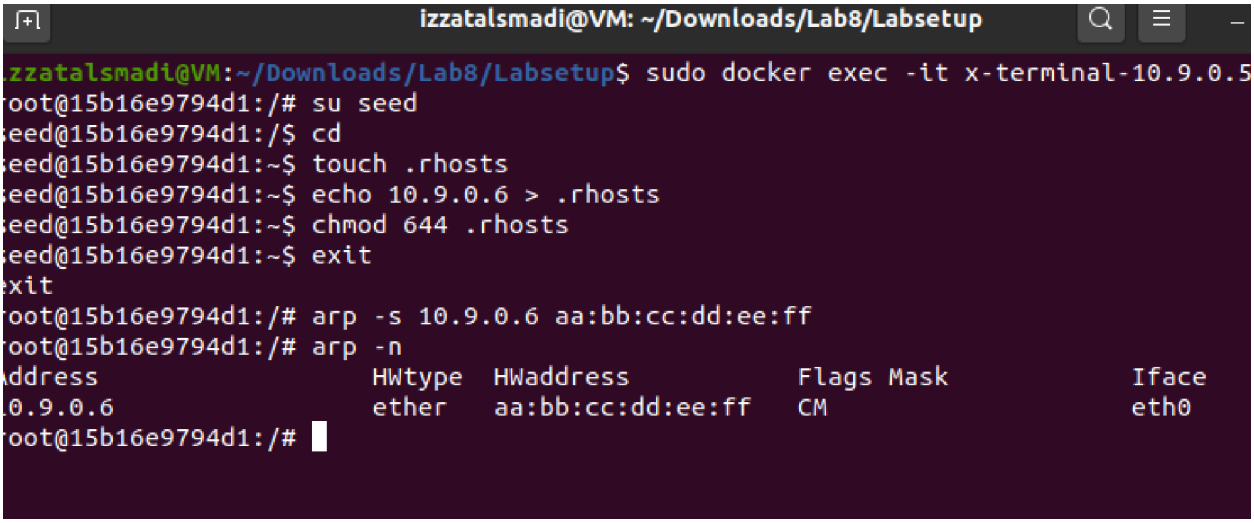
- On X-Terminal: Set up the trust relationship



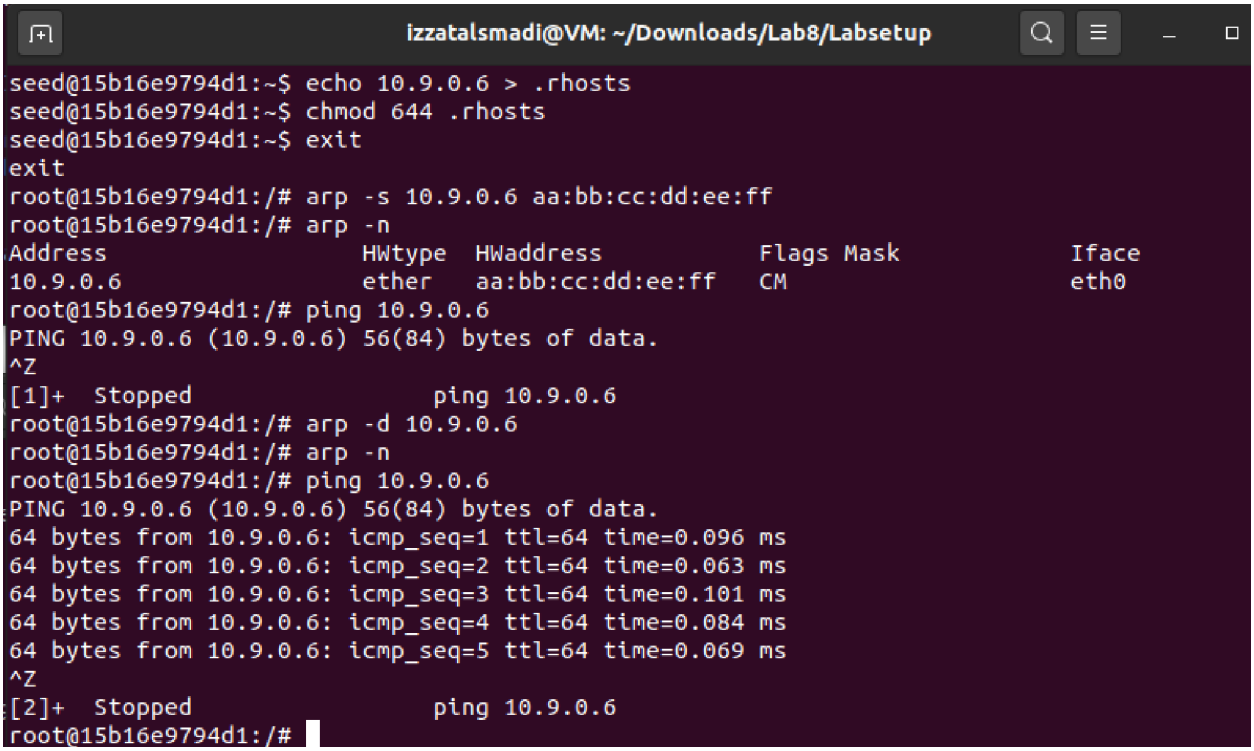
- On Trusted Server: Verify the trust relationship



- Task 1: Simulated SYN flooding



- Ping should work so I deleted ARP record to add ping first



- Now ARP is added properly

```
root@15b16e9794d1:/# arp -n
root@15b16e9794d1:/# 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=0.096 ms
64 bytes from 10.9.0.6: icmp_seq=2 ttl=64 time=0.063 ms
64 bytes from 10.9.0.6: icmp_seq=3 ttl=64 time=0.101 ms
64 bytes from 10.9.0.6: icmp_seq=4 ttl=64 time=0.084 ms
64 bytes from 10.9.0.6: icmp_seq=5 ttl=64 time=0.069 ms
^Z
[2]+  Stopped                 ping 10.9.0.6
root@15b16e9794d1:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:06   C                     eth0
root@15b16e9794d1:/#
```

- Now arp -s

```
10.9.0.6                 ether   02:42:0a:09:00:06   C                     eth0
root@15b16e9794d1:/# sudo arp -s 10.9.0.6 02:42:0a:09:00:06
bash: sudo: command not found
root@15b16e9794d1:/# arp -s 10.9.0.6 02:42:0a:09:00:06
root@15b16e9794d1:/#
superserver inetd
"Lab8" selected (containing 1
```

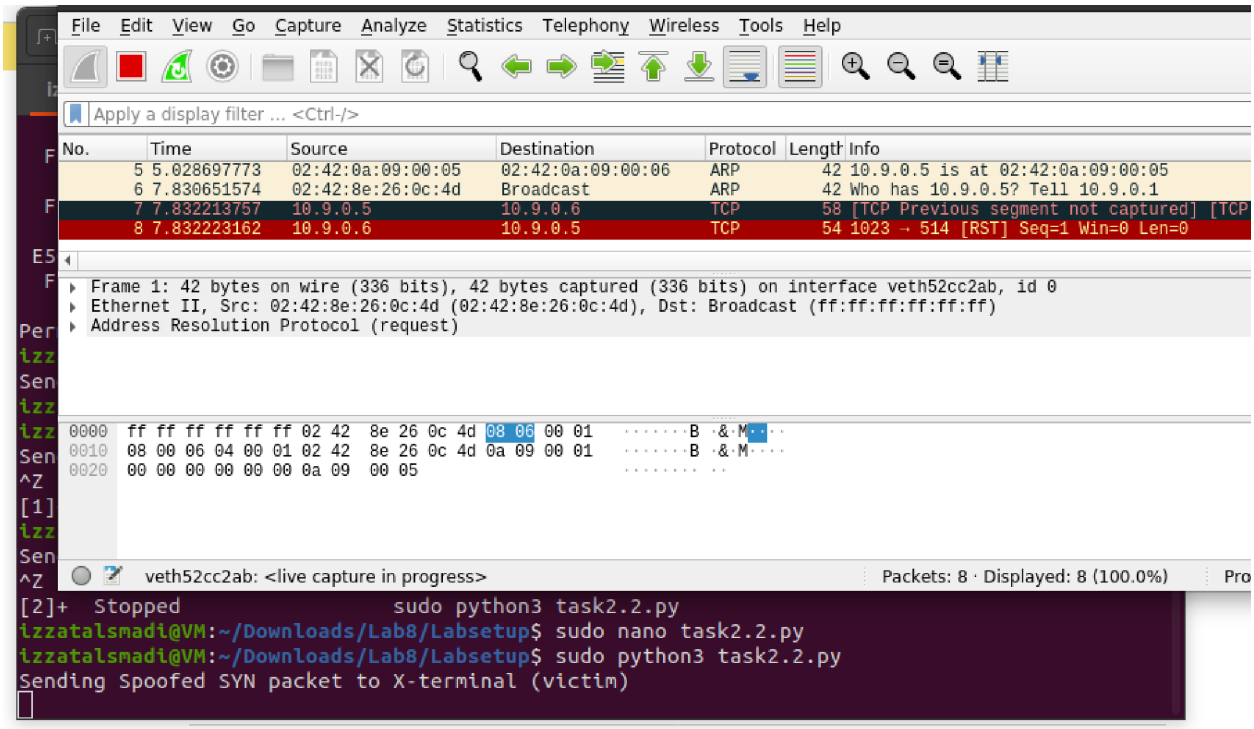
- Notice Flag mask change now

```
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:06   C                     eth0
root@15b16e9794d1:/# sudo arp -s 10.9.0.6 02:42:0a:09:00:06
bash: sudo: command not found
root@15b16e9794d1:/# arp -s 10.9.0.6 02:42:0a:09:00:06
root@15b16e9794d1:/# arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.9.0.6                 ether   02:42:0a:09:00:06   CM                    eth0
root@15b16e9794d1:/#
```

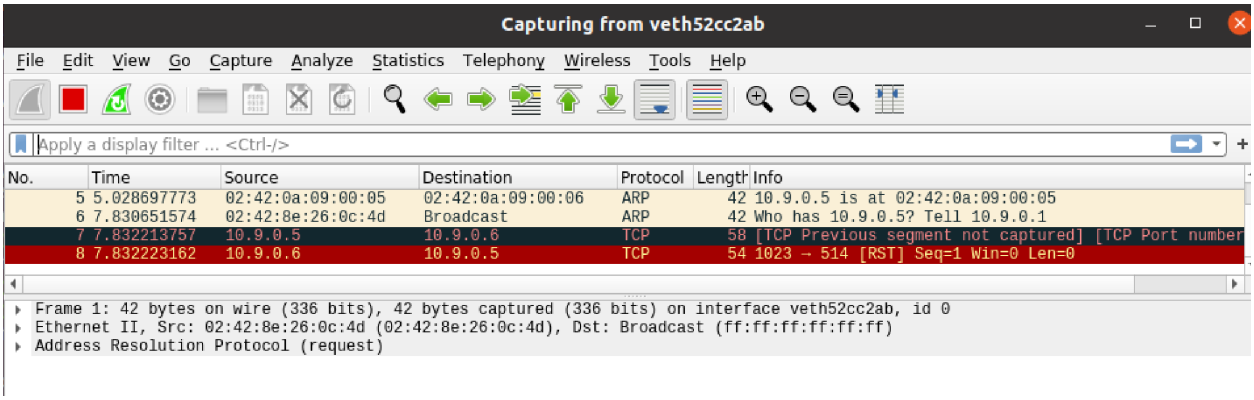
- Task 2: Spoof TCP Connections and rsh Sessions
  To launch the attack, we need to do the following:
- Step 1: Spoof a SYN packet from Trusted server to X-terminal.

```
izzatalsmadi@VM:~/Downloads/Lab8/Labsetup$ python3 task2.1.py
Sending Spoofed SYN packet to X-terminal (victim)
Traceback (most recent call last):
  File "task2.1.py", line 9, in <module>
    send(pkt,verbose=0)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py", line 345, in send
    socket = socket or conf.L3socket(*args, **kargs)
  File "/usr/local/lib/python3.8/dist-packages/scapy/arch/linux.py", line 398, in __init__
    self.ins = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.htons(type))  # noqa
E501
  File "/usr/lib/python3.8/socket.py", line 231, in __init__
    _socket.socket.__init__(self, family, type, proto, fileno)
PermissionError: [Errno 1] Operation not permitted
izzatalsmadi@VM:~/Downloads/Lab8/Labsetup$ sudo python3 task2.1.py
Sending Spoofed SYN packet to X-terminal (victim)
izzatalsmadi@VM:~/Downloads/Lab8/Labsetup$
```

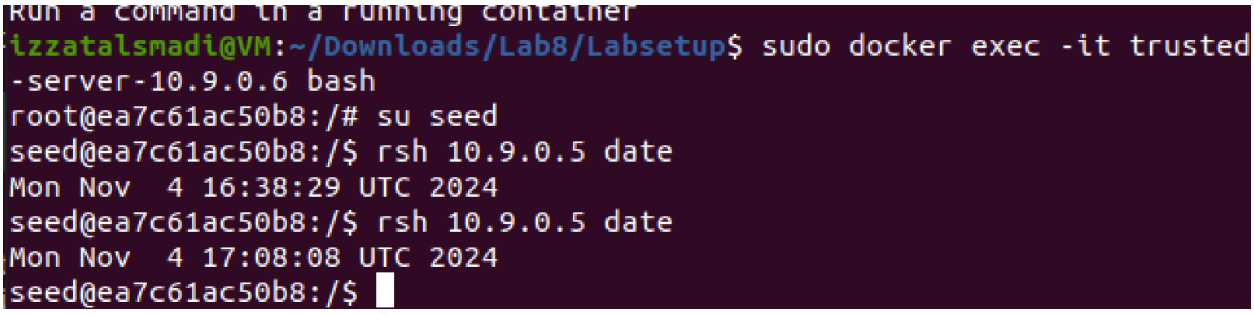
- Step 2: Step 1 will trigger X-Terminal to send out a SYN+ACK. We need to spoof an ACK to finish the handshake protocol.



- Run task2.1 then task2.2 and check wireshark output (successful TCP connection)



- (Create a to the trusted server, then from that connection (while code is running), type: su seed, then rsh 10.9.0.5 date, then monitor output on the other trusted server terminal



- You can see that after getting the RSH data, X-Terminal will initiate the second connection and send it to the Trusted Server. We need to spoof an ACK. If this connection cannot be established, X-Terminal will abort. , then try the code below

- Then complete Task 3 (Task 3: Set Up a Backdoor)