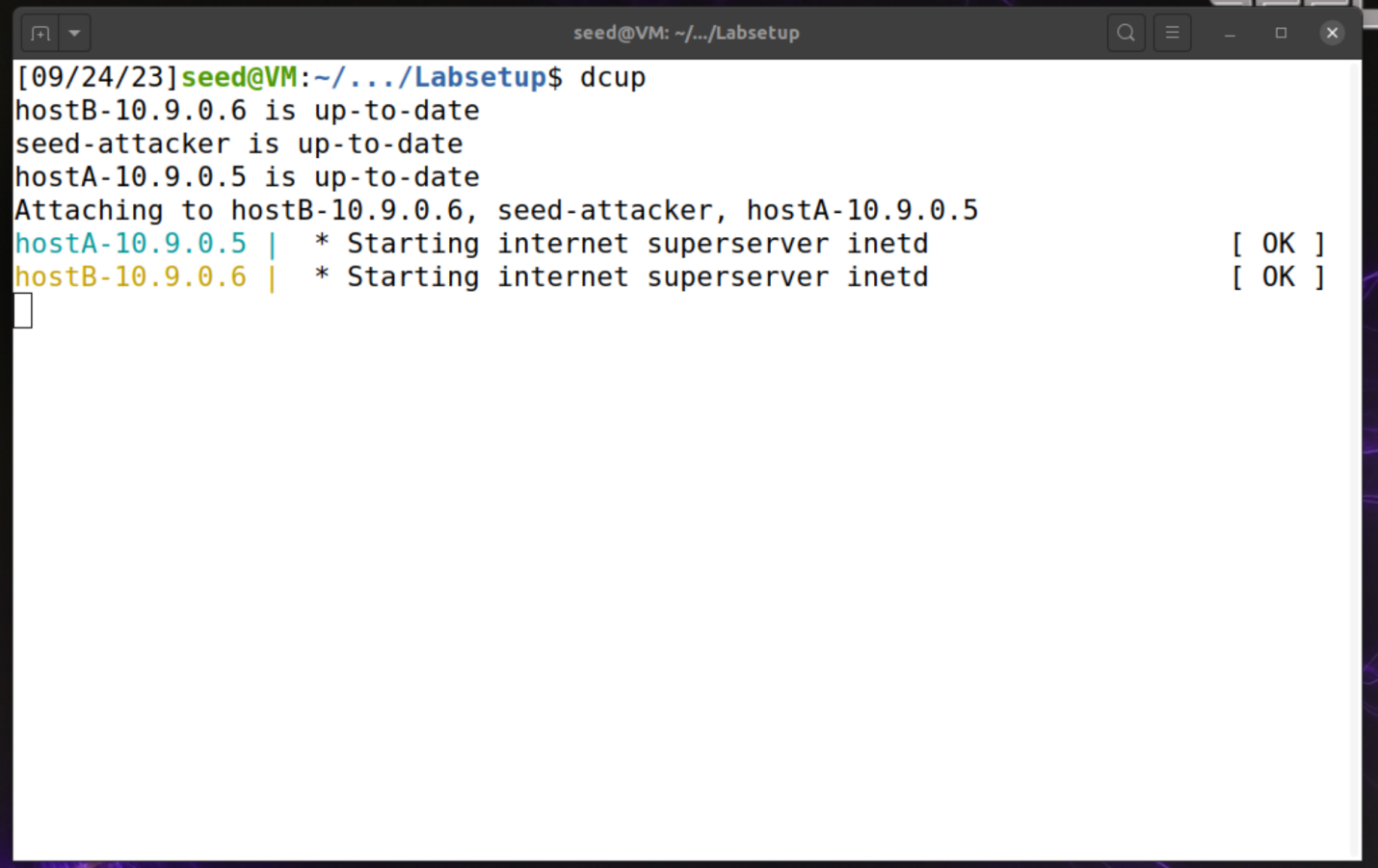
Lab 1: Packet Sniffing and Spoofing

Name: Guo Yuchen

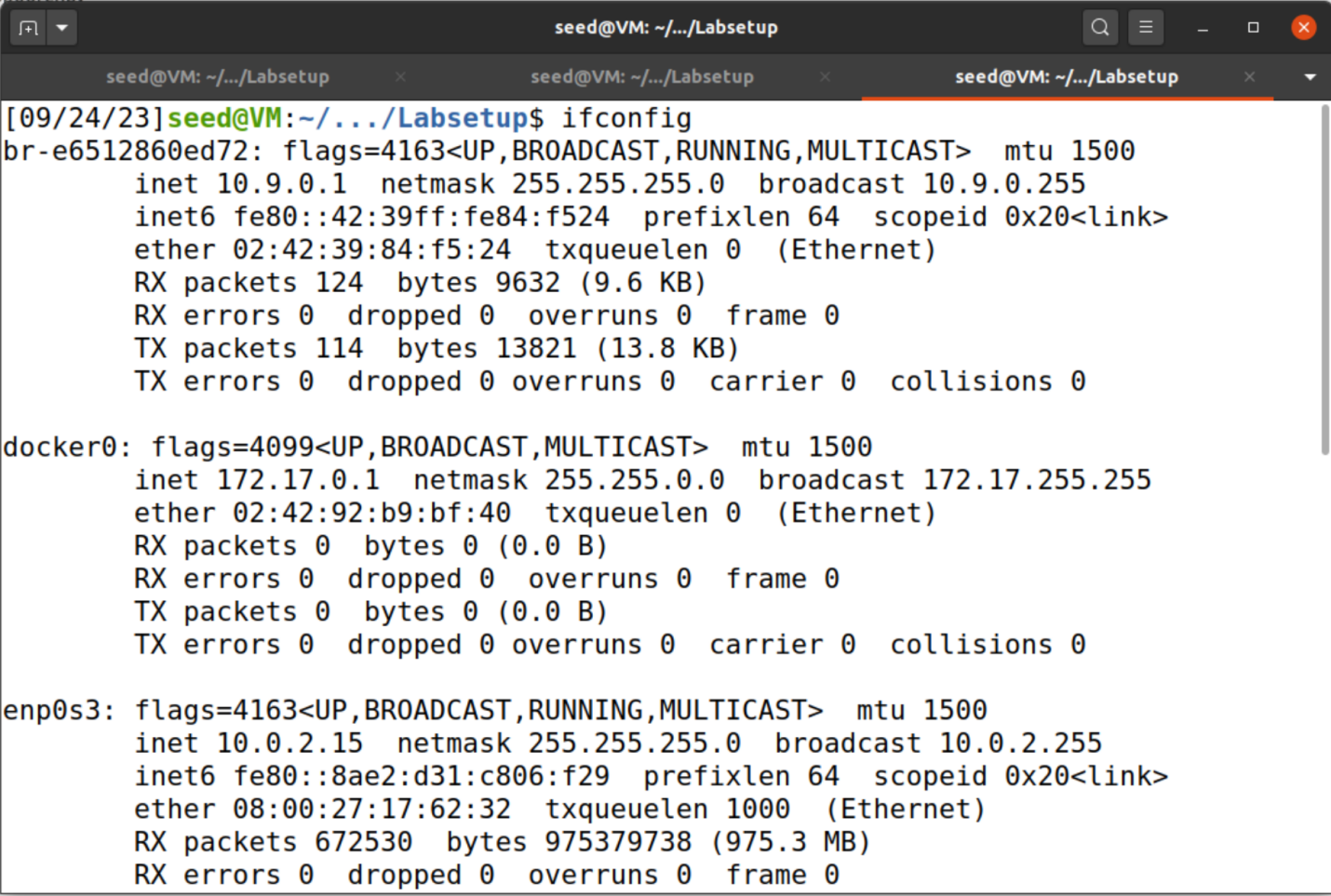
Student ID: 1004885

# Setup and preparation

Starting the containers using `dcup`



Using ifconfig to get network interface ID br-e6512860ed72



# Task 1.1A

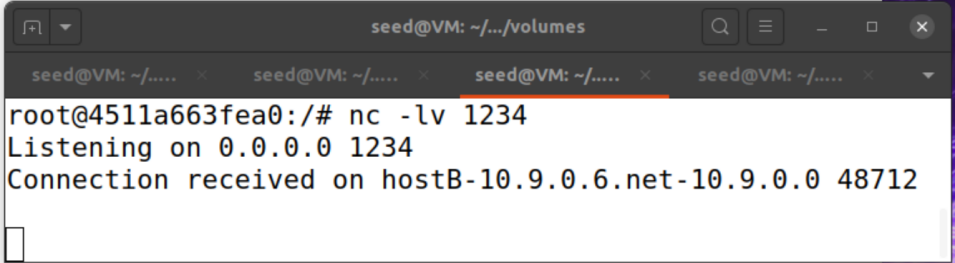
Create a Python file `sniffer.py` and make it executable using `chmod a+x sniffer.py`.

文本

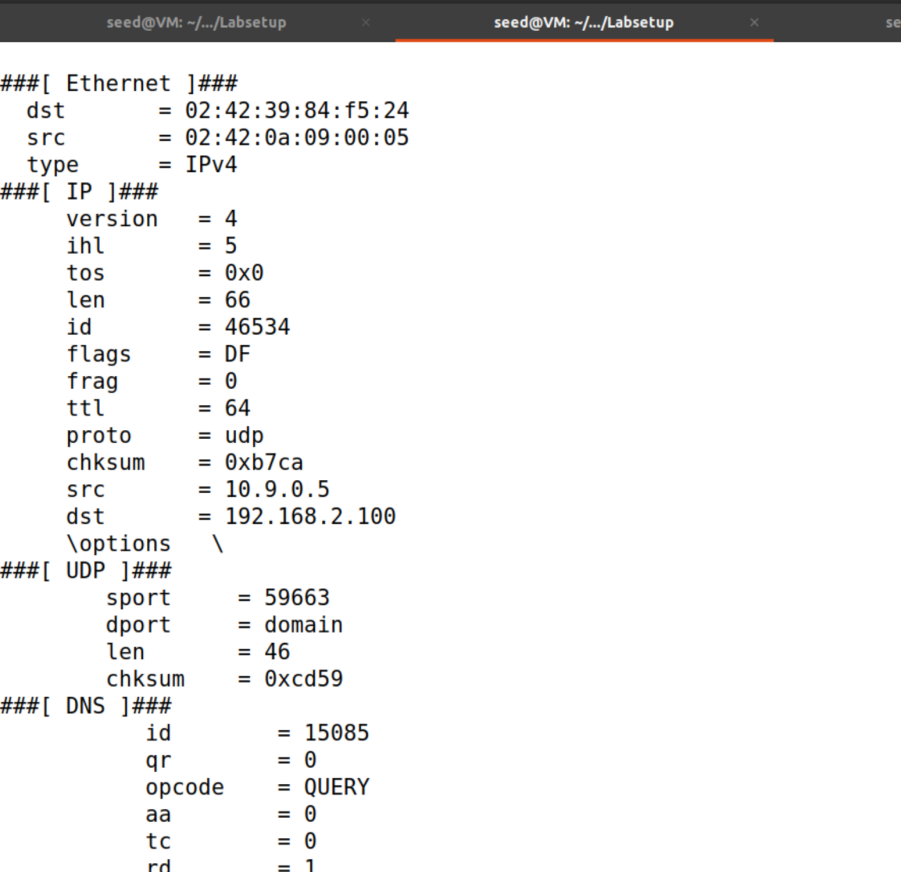
描述已自动生成

## Running sniffer.py with root privilege

To create some traffic, use `nc` command on host 10.9.0.5 to listen to the connection from host 10.9.0.6



Traffic captured by Scapy:



## Running sniffer.py without root privilege

文本

描述已自动生成

## Observation

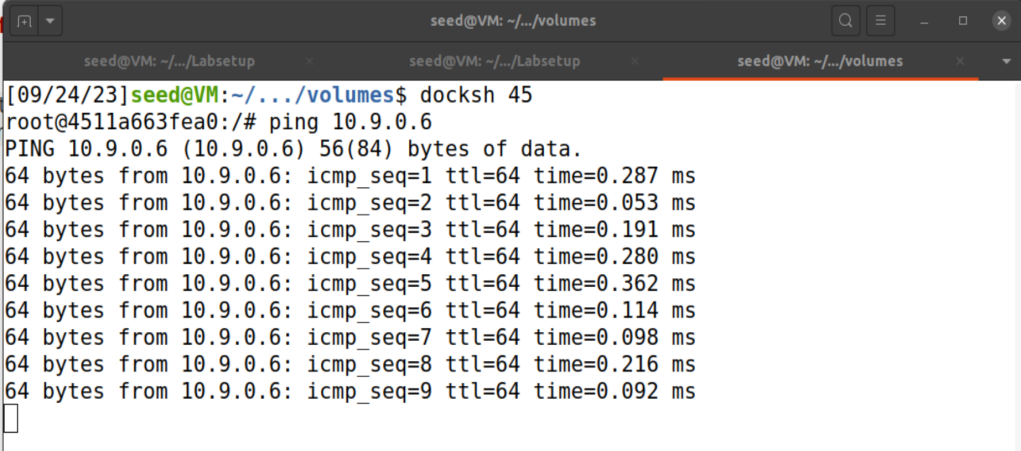
1. Only root can sniff packages.
2. Reasons: only root is allowed to open raw sockets, which is used by sniff to receive packets such that the packet won’t be intercepted by the normal socket.

# Task 1.1B

## Capture icmp packet

Modify the filter parameter to be “icmp” to capture only the ICMP packet. Ping from 10.9.0.5 to 10.9.0.6 since ping sends icmp package.





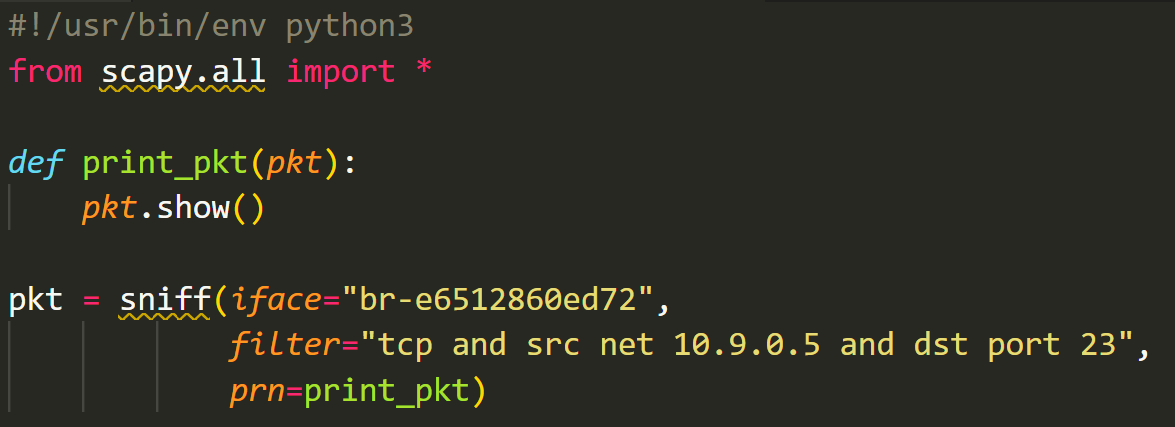
The packets are captured by the sniffer as shown:

图形用户界面, 文本, 应用程序

描述已自动生成

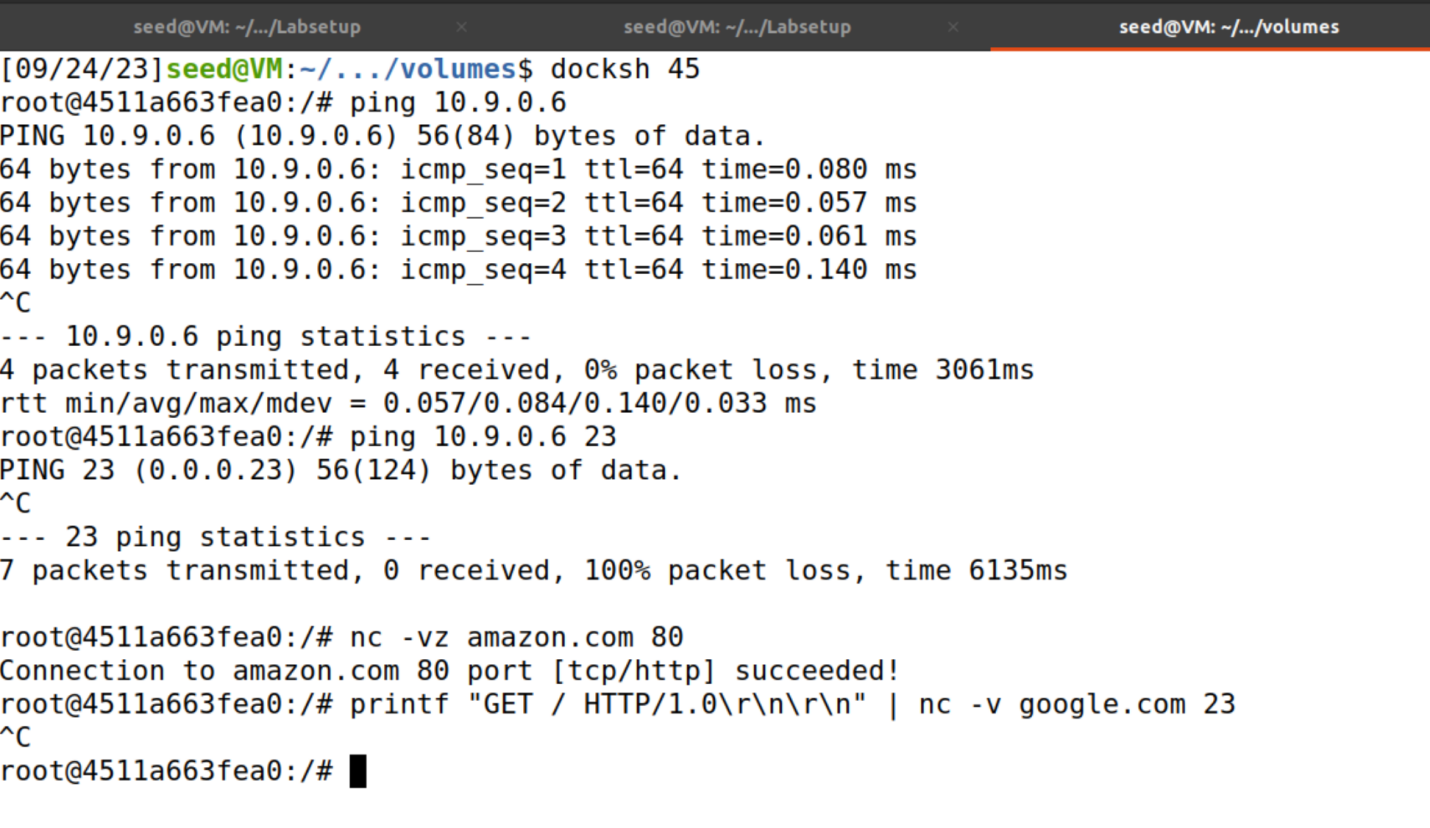
## Capture any TCP packet that comes from a particular IP and with a destination port number 23.

Change the filter parameter accordingly.

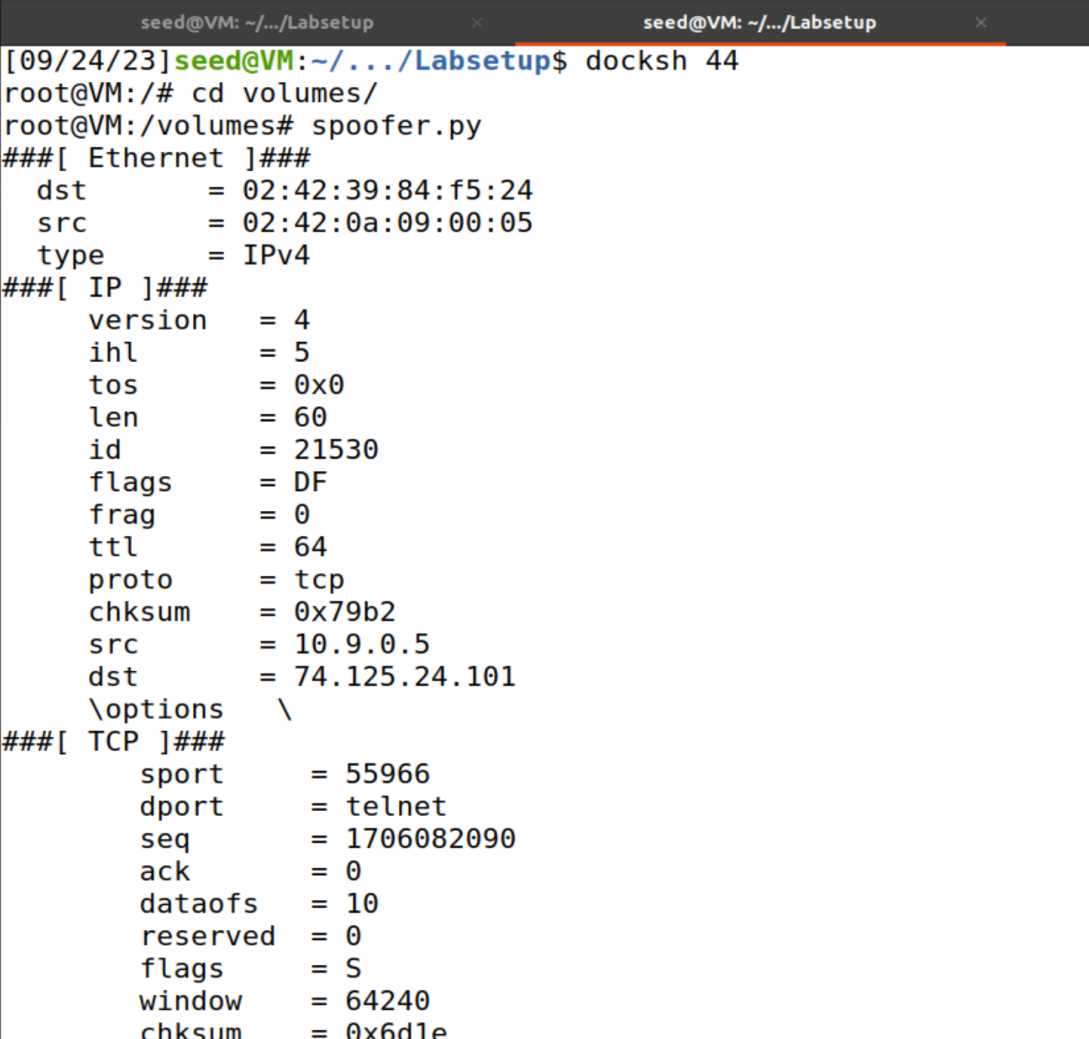


Then try several different command:

* ping 10.9.0.6: wrong type, wrong dst port
* ping 10.9.0.6/23: wrong type
* nc -vz amazon.com 80: wrong dst port
* nc -v google.com 23



Sniffer captures:



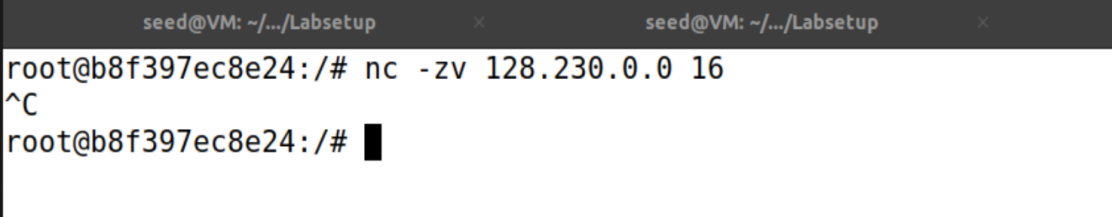
It only responds for the last case, when the port number, src ip, and packet type fit the filter.

## Capture packets comes from or to go to a particular subnet

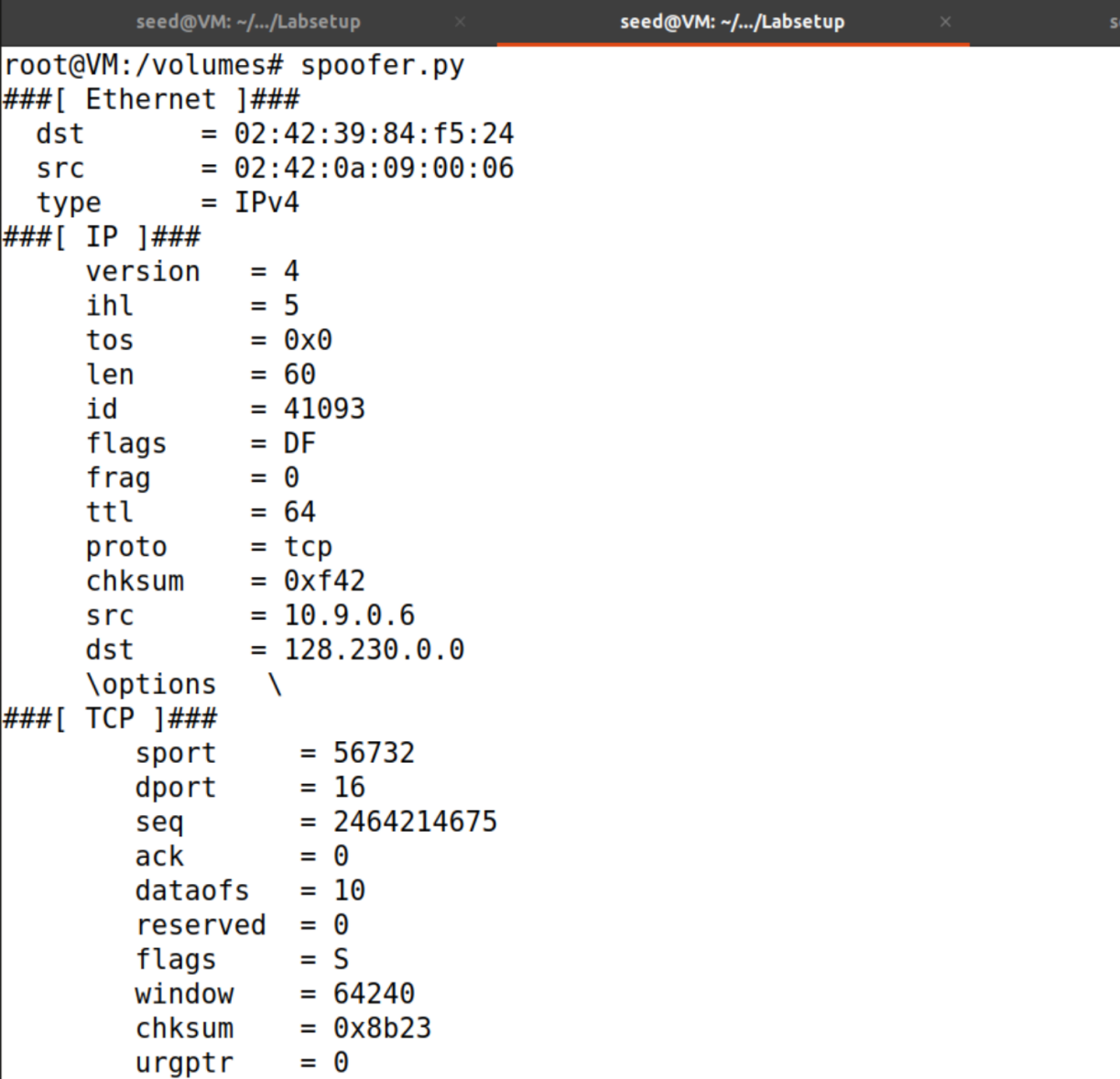
Choose the subnet to be 128. 230.0.0/16 and change to filter parameter accordingly.



Let one of the host sends a tcp packet to this ip address:



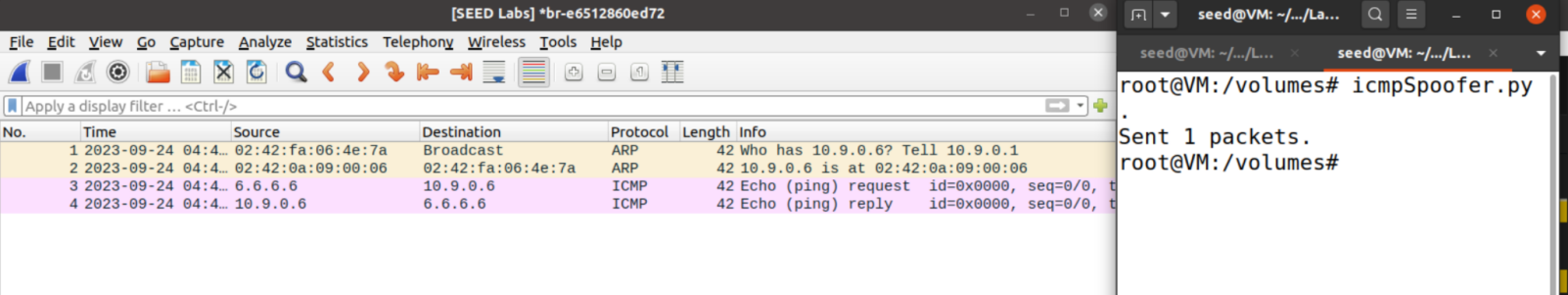
Then the sniffer program captures the packet which has dst and port as specified:



# Task 1.2

icmpSpoofer.py:



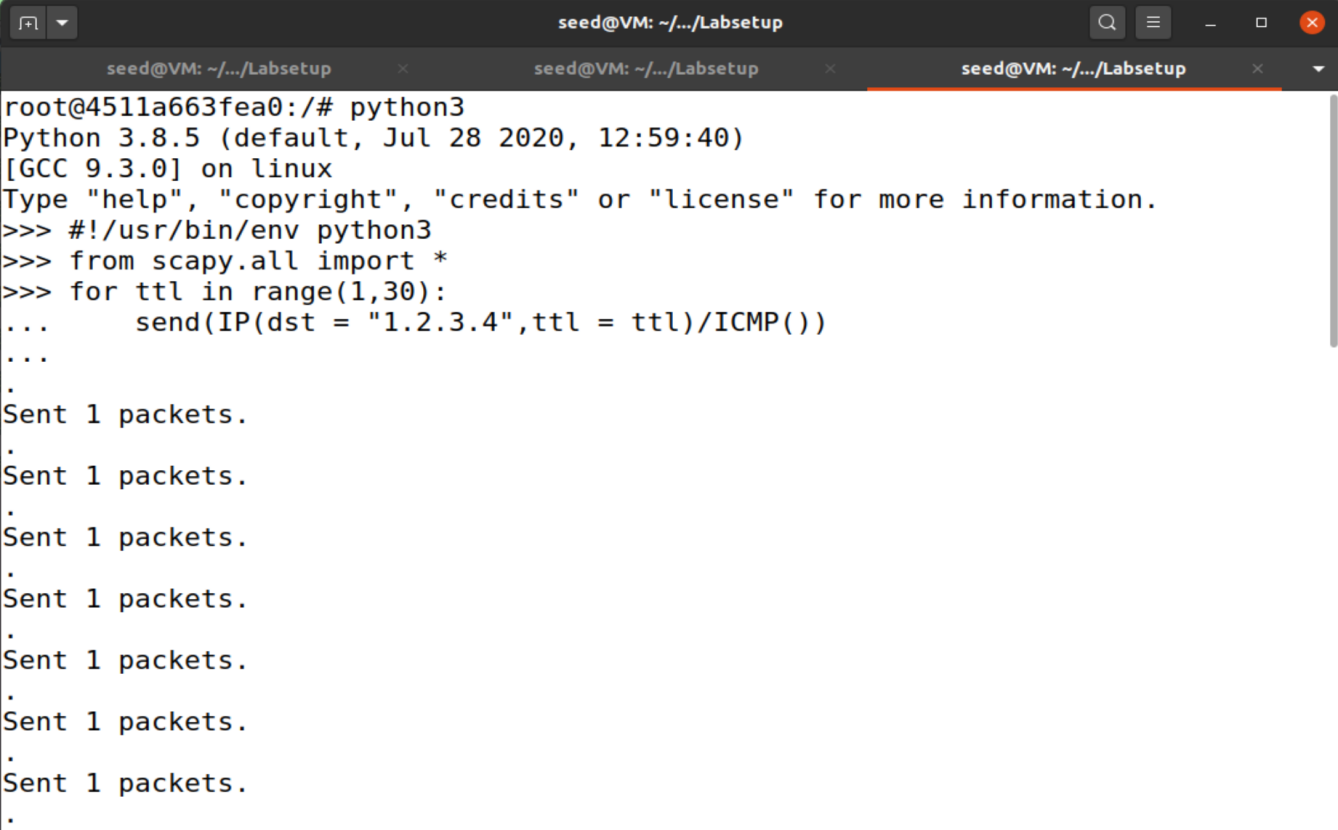


In the icmpSpoofer.py, the program sends a spoofed icmp echo request with source ip 6.6.6.6 to the host 10.9.0.6. And as observed in the Wireshark (the last line), the host 10.9.0.6 receives an icmp echo reply from the spoofed source 6.6.6.6, which is specified in our spoofer program.

# Task 1.3

To estimate the distance between the VM and a selected destination in terms of number of routers, let one of the hosts to run the program as shown in the terminal.

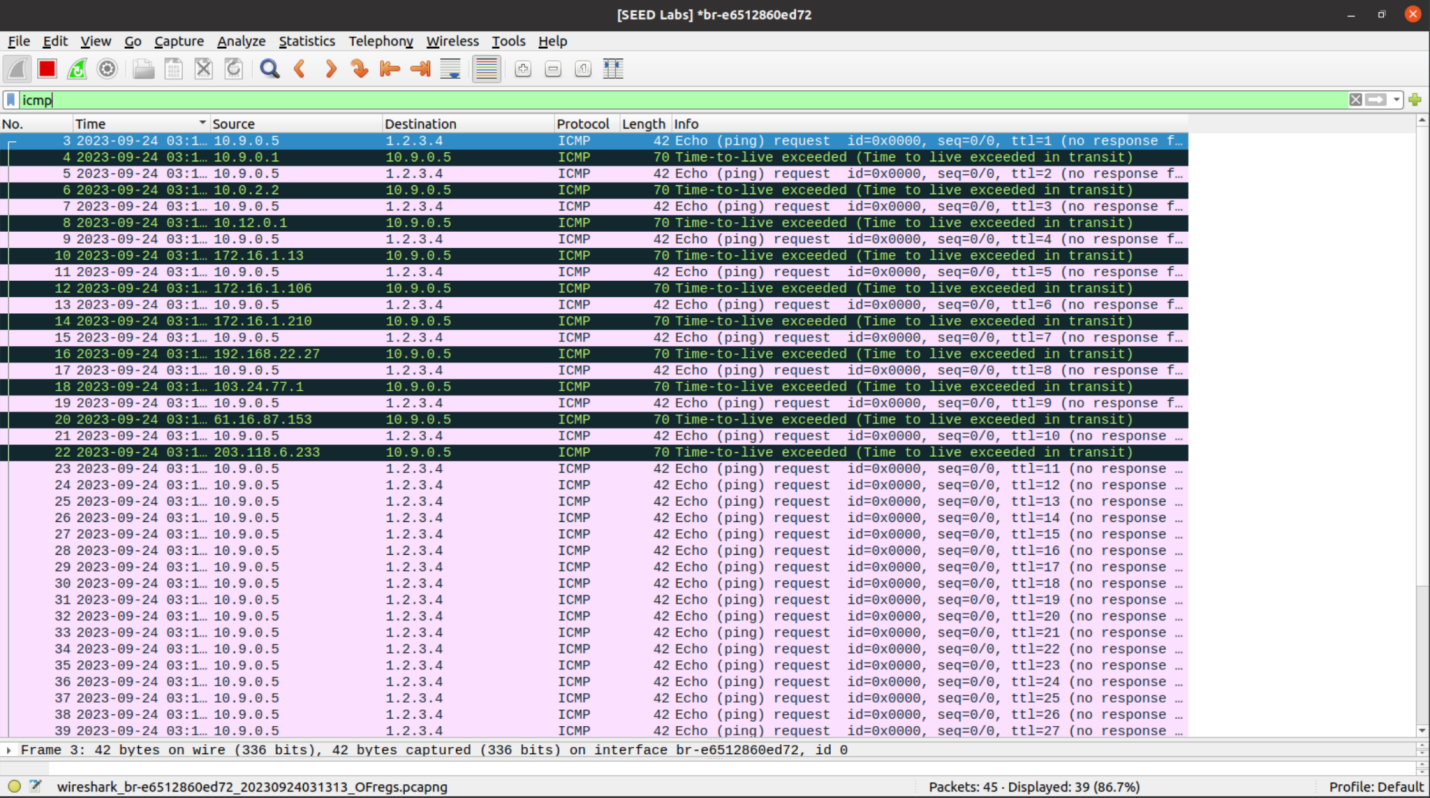
Basically, we are creating a sequence of icmp packets with a destination ip address to be 1.2.3.4, and repeatedly increased ttl from 1 to 30.



Then as observed and recorded in the Wireshark, the routes are shown in the black lines source column, which I summarized as follows:

1. 10.9.0.1
2. 10.0.2.2
3. 10.12.0.1
4. 172.16.1.13
5. 172.16.1.106
6. 172.16.1.210
7. 192.168.22.27
8. 103.24.77.1
9. 61.16.87.153
10. 203.118.6.233

And since 1.2.3.4 does not exist, the host could not get an echo reply from there.



# Task 1.4

The sniff and spoof program ss.py is as follows:

文本

描述已自动生成

After a packet is sniffed, the callback function spoof\_pkt is called to spoof the packet. The spoof\_pkt will reverse the destination and source ip address, take the icmp id and seq number, then construct an icmp echo reply message to send it back.

## ping 1.2.3.4

# a non-existing host on the Internet

文本

描述已自动生成

Observation: the ping program receives 3 echo replies, which are spoofed by the spoofer program, although 1.2.3.4 does not exist.

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

## ping 10.9.0.99

# a non-existing host on the LAN

文本

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

Observation: no packets is received, means that the spoofer could not spoof any message. Because the 10.9.0.99 is on the same LAN with the host 10.9.0.5, thus before they can exchange message, it needs the MAC address of 10.9.0.99. Since it’s not stored in the ARP cache, it sends out a broadcast message to ask for reply, and the real communication only takes place after it acquires the destination device’s MAC address. In our case, there is neither ARP reply or spoofed ARP reply, our spoofed icmp packet will not be received.

## ping 8.8.8.8

# an existing host on the Internet

文本

描述已自动生成

图形用户界面, 文本, 应用程序

描述已自动生成

Observation: both ping requests (the real reply and the spoofed reply) are received by the ping program. But the ones that are spoofed by me(ttl is set to 40 to distinguish) are marked as duplicated since they come later than the real one.