1. **PACKET SNIFFING AND SPOOFING LAB/ASSIGNMENT REPORT**

**Name – Subhashini Thirumala**

**Student Number – 700740834**

**email** [**ID-Sxt08341@ucmo.edu**](mailto:ID-Sxt08341@ucmo.edu)

**With reference to the Lab instructions, I have done all the tasks given in the instructions along with lab manual. Detailed report is here under:**

**Screenshots of Brining up a Terminal in Ubuntu 20.04 VM and changing directory to Labsetup folder through linux command, Also dcbuild & dcup commands to bring up the container:**

**Text

Description automatically generated**

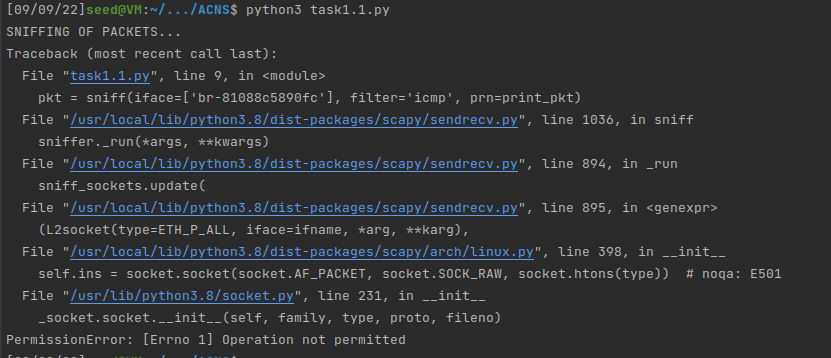
**TASK 1: Python Program and Usage of SCAPY :** This task is to illustrate how to use Scapy using Python. Following is an example. Code for this program is intask1.py file . I have used Pycharm for coding.

Text

Description automatically generated

**Terminal Screenshot of execution part for above example**

**TASK 1.1 and 1.1 A & B: SNIFFING PACKETS USING SCAPY THROUGH FILTERS:** These tasks are to do packet sniffing in Python programs and to do specific types of packets using filters. Always we run the programs with root privileges. **Chmod u+x filename.py** is the command to run program with root privileges in Pycharm.



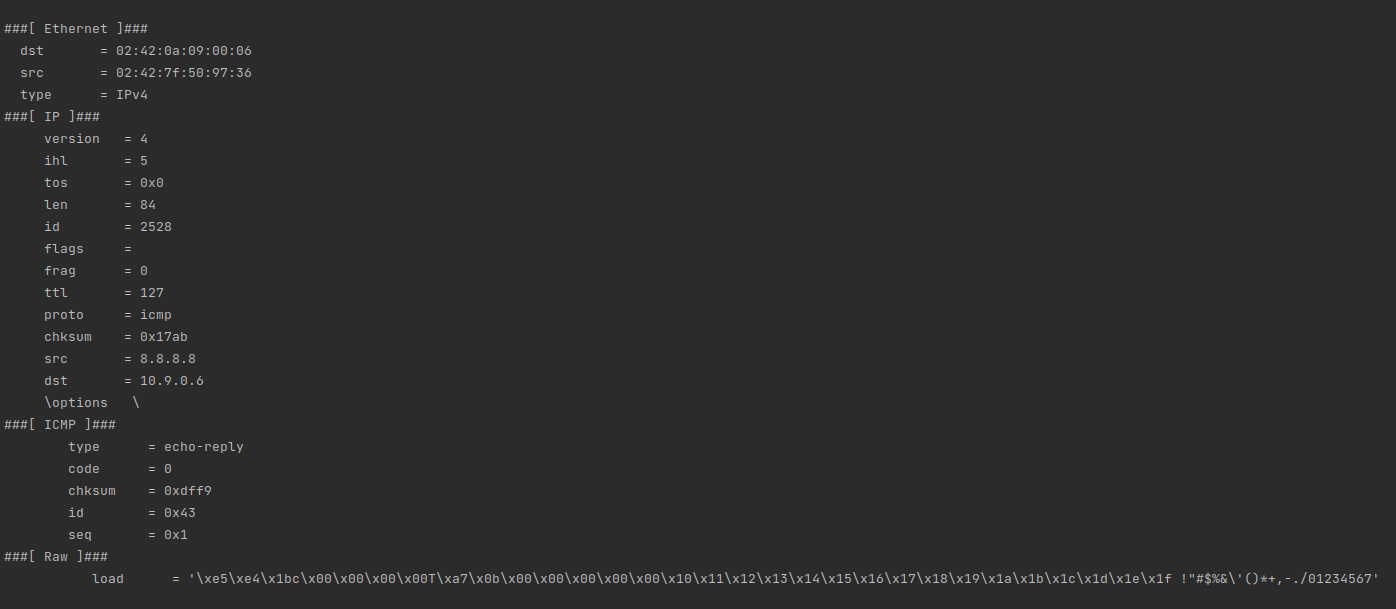
**Terminal Screenshot of execution task 1A**

**1.1B Capture only the ICMP packet**: Code for this program is intask1.1.py file. The program did sniff only ICMP packets on br-81088c5890fc interface in my vmware. Execution in Terminal of Pycharm screenshots are here under.

**Text

Description automatically generated**

**Terminal Screenshot of execution part 1 for above task**

****

**Terminal Screenshot of execution part 2 for above task**

**Text

Description automatically generated**

**Container Screenshot for above task: Ping ICMP 8.8.8.8 from container terminal**

**1.1B Capture any TCP packet that comes from a particular IP and with a destination port number 23:** Capturing TCP packet that comes from 10.9.0.6 IP and with a destination port number 23. Later done telnet from my host VM to the victim at 10.9.0.5. Code for this program is intask1.1tcp.py file. Execution in Terminal of Pycharm screenshots are here under.

Text

Description automatically generated **Terminal Screenshot of execution task 1.1B**

Text

Description automatically generated

**Terminal Screenshot of execution task 1B**

Text

Description automatically generated

**Container Screenshot of execution task 1B**

**1.1B Capture packets comes from or to go to a particular subnet**: you can use 153.91.1.0/24 - the subnet belongs to UCM and ping UCM’s web server at 153.91.1.10 after setting the filter on your sniff program

**Graphical user interface, text

Description automatically generated**

**Screenshot of Execution of subnet task Terminal window**

**A screenshot of a computer

Description automatically generated with medium confidence**

**Text

Description automatically generated**

**Container Screenshot for the commands of task 1B Subnet**

**TASK 1.2: SPOOFING OF ICMP PACKETS:**

Code for this program is intask1.2.py to spoof IP packets with an arbitrary source ip address 8.8.8.8 for the request sent to destination 10.9.0.6 ip. I used ls(IP) to see all the attribute names/values.

Text

Description automatically generated

Text

Description automatically generated

**Screenshot of Execution of task 1.2 Terminal window**

Graphical user interface, text, application

Description automatically generated

**Wireshark Screenshot for the task 1.2**

**TASK 1.3: TRACE ROUTE:** Below is the screen shot of the Traceroute to IP address 1.2.3.4, took TTL 5 to get the echo reply from 1.2.3.4, observed request dropped by 5 different routers. There are 5 different routers in between my VM to IP address 1.2.3.4. Used Task1.3.py file.

**Text

Description automatically generated**

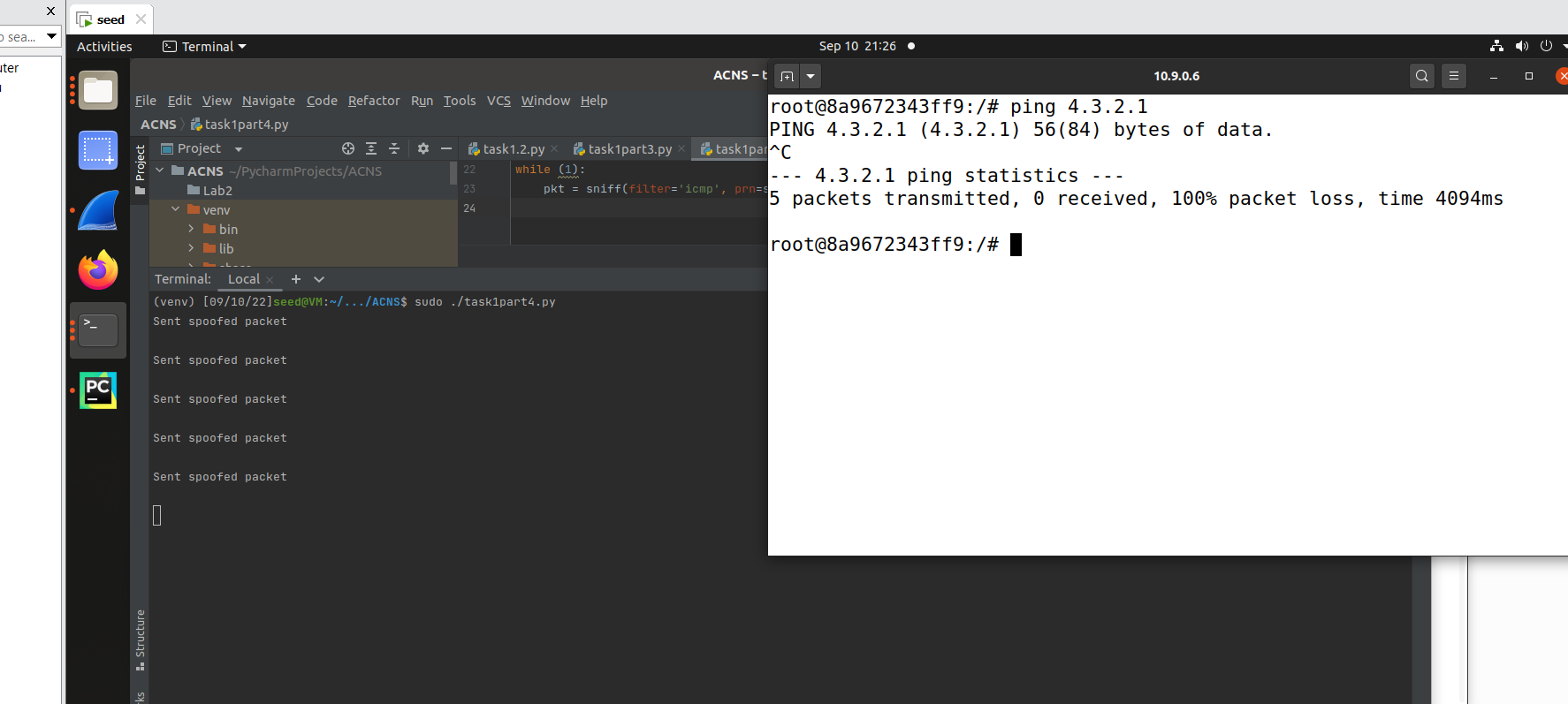
**Screenshot of Execution of Task 1.3 Terminal window**

**Graphical user interface, text, application

Description automatically generated**

**Wireshark screenshot for reply received from the 1.2.3.4**

**TASK 1.4: SNIFFING AND THEN SPOOFING:** I am using my VM and the user Container on the same LAN. From container I did ping 4.3.2.1 then 10.9.0.99 and 8.8.8.8 randomly to observe Sniffing and then Spoofing. And I did observe request and responses on Wireshark tool to analyse their behavior. Screenshots for all the observations were attached hereunder.

****

**Graphical user interface, text, application, table

Description automatically generated**

**Graphical user interface

Description automatically generated**

**Table

Description automatically generated with low confidence**

**Wireshark Screenshot for the observations made for the Task1.4**

**Ip route command to understand ARP protocol work:**

**Text

Description automatically generated**

**Additional Instructions not specified in the original manual:**

**Network Sweeper Program:** The following python program written to ping hosts on a given subnet. I have given two different subnets to observe ie 153.91.152 and also 8.8.8. & 10.9.0 (My VM Ips) .Some of the IP addresses in the given subnet are down and some of them are up since at this particular time of execution the host is down. The screenshots were included hereunder.

**Text

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Graphical user interface, text

Description automatically generated**

**DCDOWN:Text

Description automatically generated with medium confidence**

**Additional Instructions not specified in the original manual:**

1. **Created SNIFF folder under Lab folder inside Ubuntu 20.04 VM in my Laptop, (Seed screenshot):**

**Graphical user interface, text, application, website

Description automatically generated**

1. **Labsetup.zip downloaded through Firefox browser in Ubuntu 20.04 VM in my laptop and extracted plus saved the data into SNIFF folder created earlier, screenshots is hereunder:**

**Graphical user interface, text

Description automatically generated**

**Graphical user interface

Description automatically generated**