CNS LAB 2

PACKET SNIFFING AND SPOOFING USING PCAP

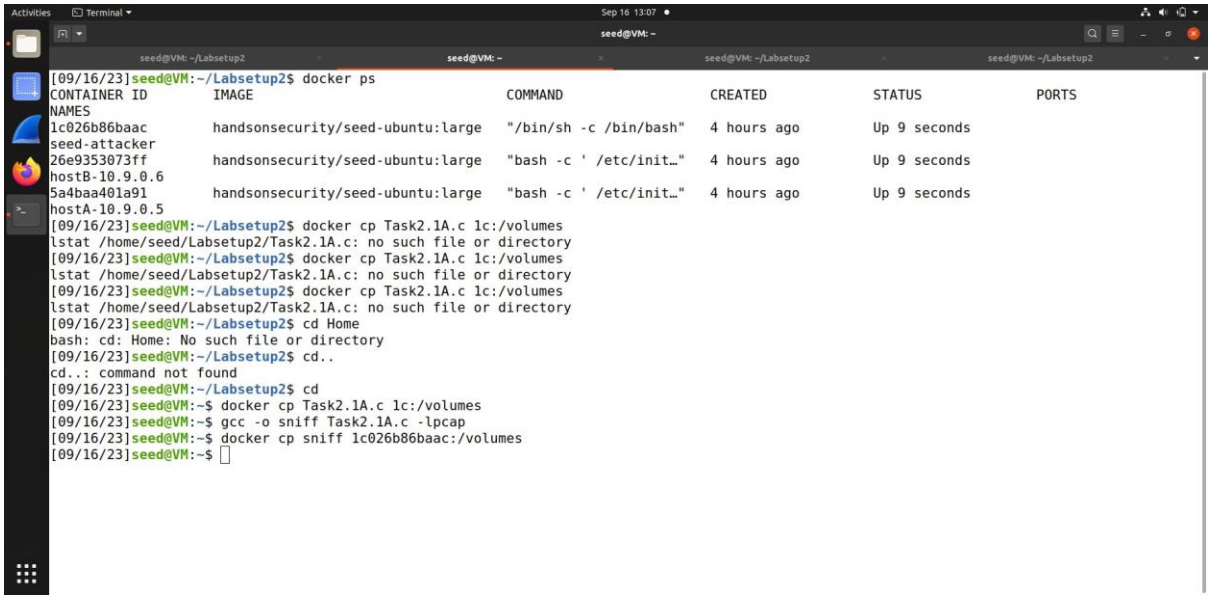
NAME: NAVYA PERAM

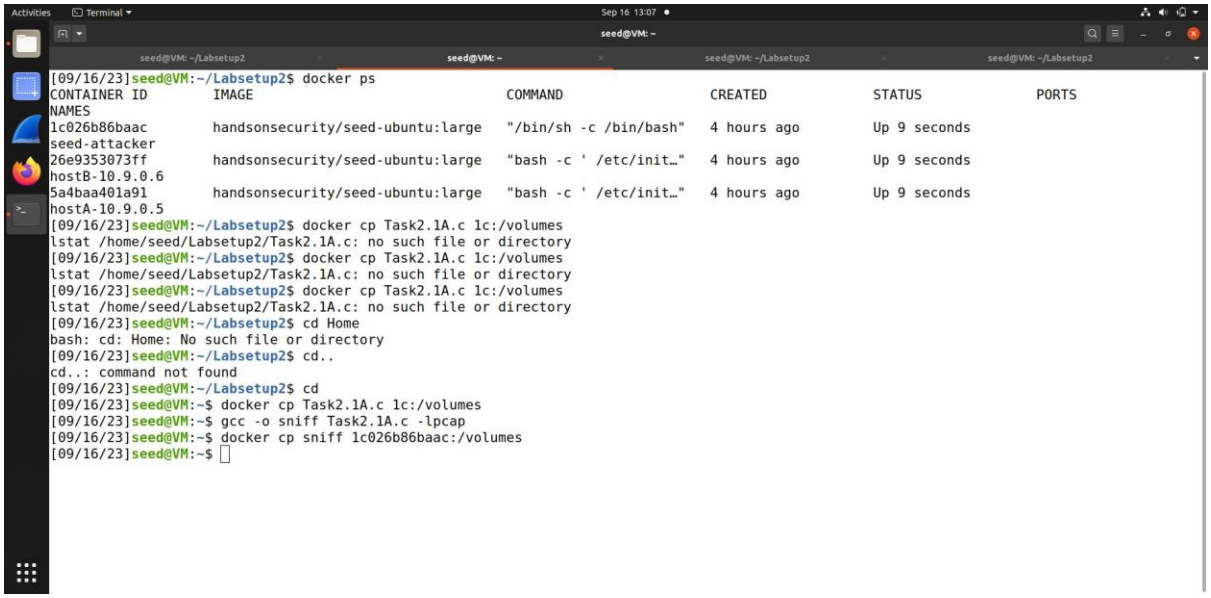
SRN: PES1UG21CS924

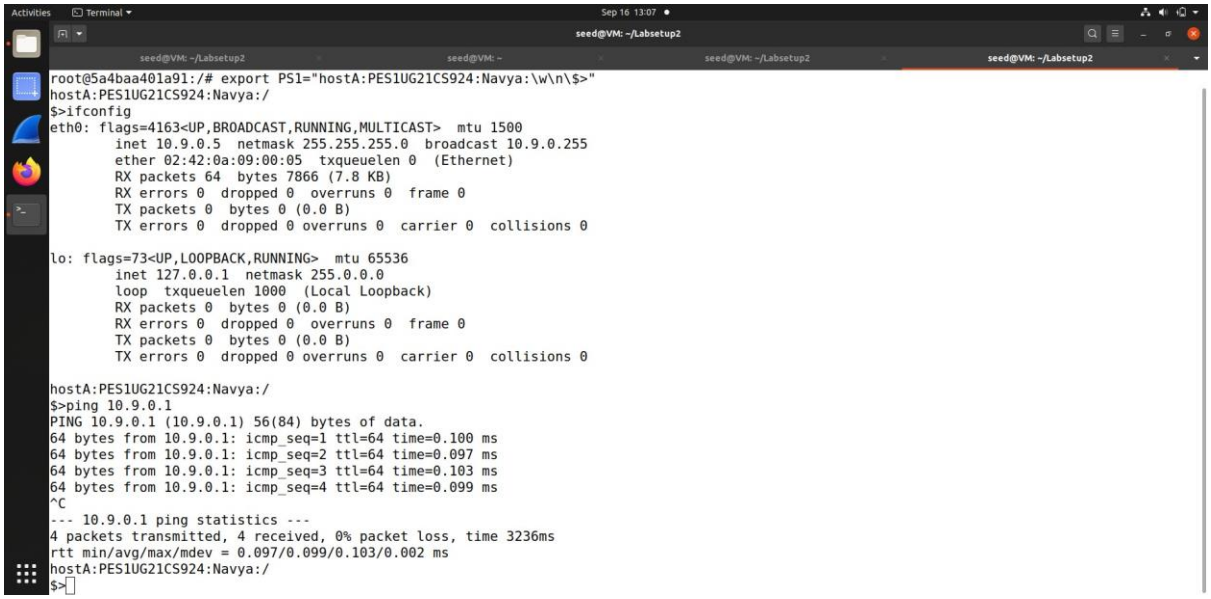
SEC: F

Task 2.1 A

In this task, we are using a sniffer program with a pcap library to analyze and print the source and destination of each packet. The icmp packets sent and received are captured and various information, such as their header and IP protocol info are stored. Here, the ipheader and the ethheader are used to store information about the ip protocols and the ethernet header. The got_packet() function, then extracts information about the packet into the two structures and then displays various info about the packet such as the protocol, source and destination ip addresses.

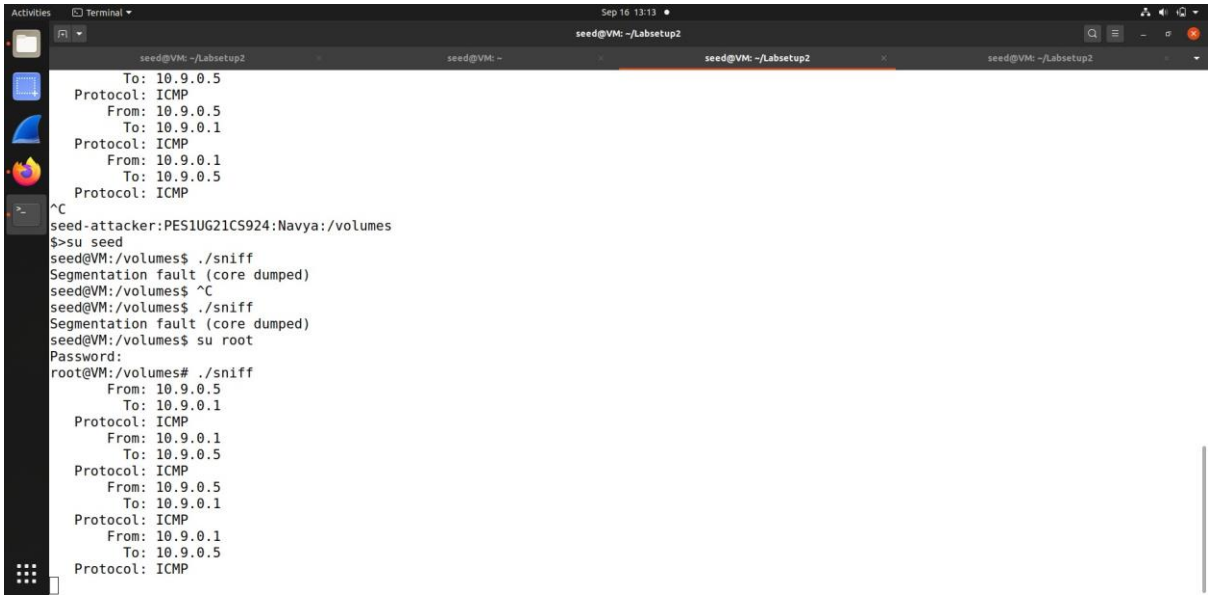






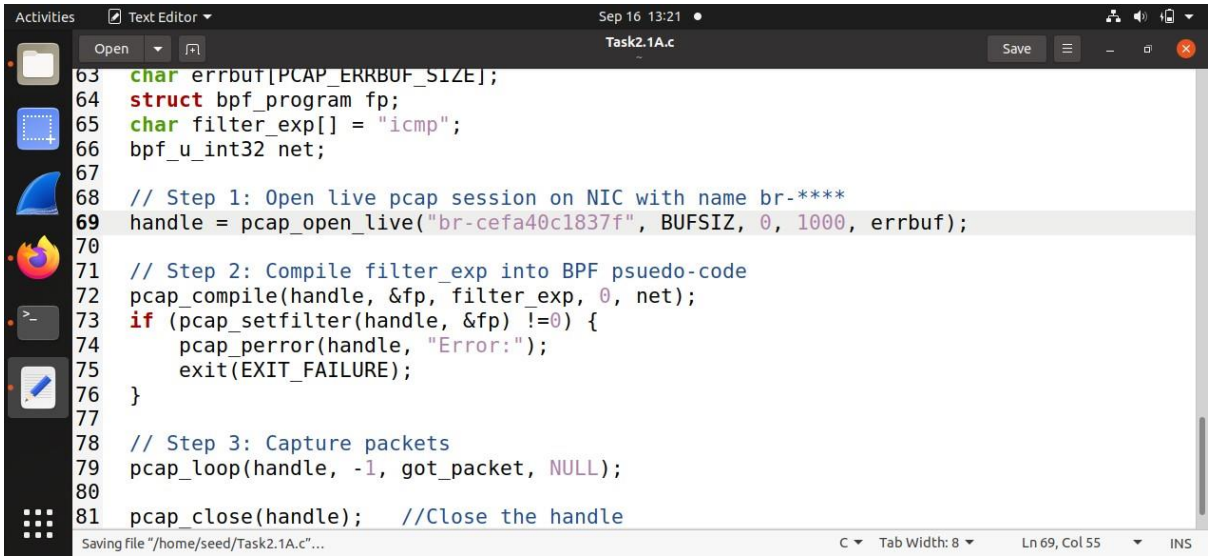
1. The various functions are:
   Pcap_open_live() - this is used to initiate the packet capture on the given interfaces.
   Pcap_perror() – it is used to print error messages while debugging
   Pcap_compile - this reads the filter expression and then applies it to the packet capture
   Pcap_loop() – this is used repeatedly throughout the code whenever we need to capture packets and process them
   Pcap_close() – this function is used at the end, to stop packet capturing

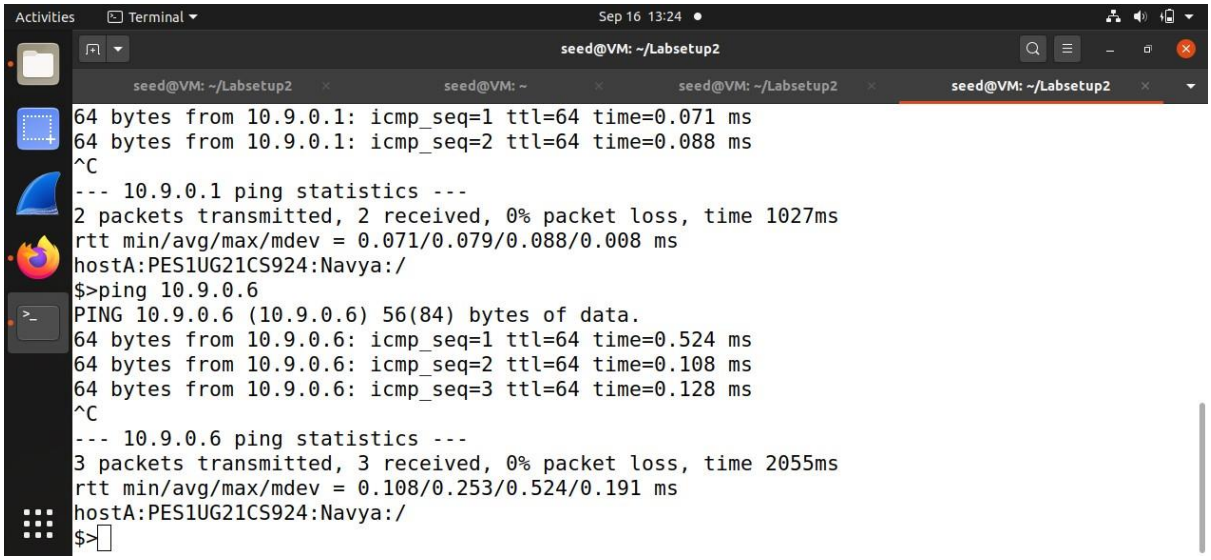
2. Without root



We cannot capture the packets in the absence of root privileges due to privacy and security concerns. There are no checks in this mode, which may allow unauthorized users from flooding the network with malicious or unnecessary packets. The program will not be able to access a raw socket.

PROMISCUOUS MODE



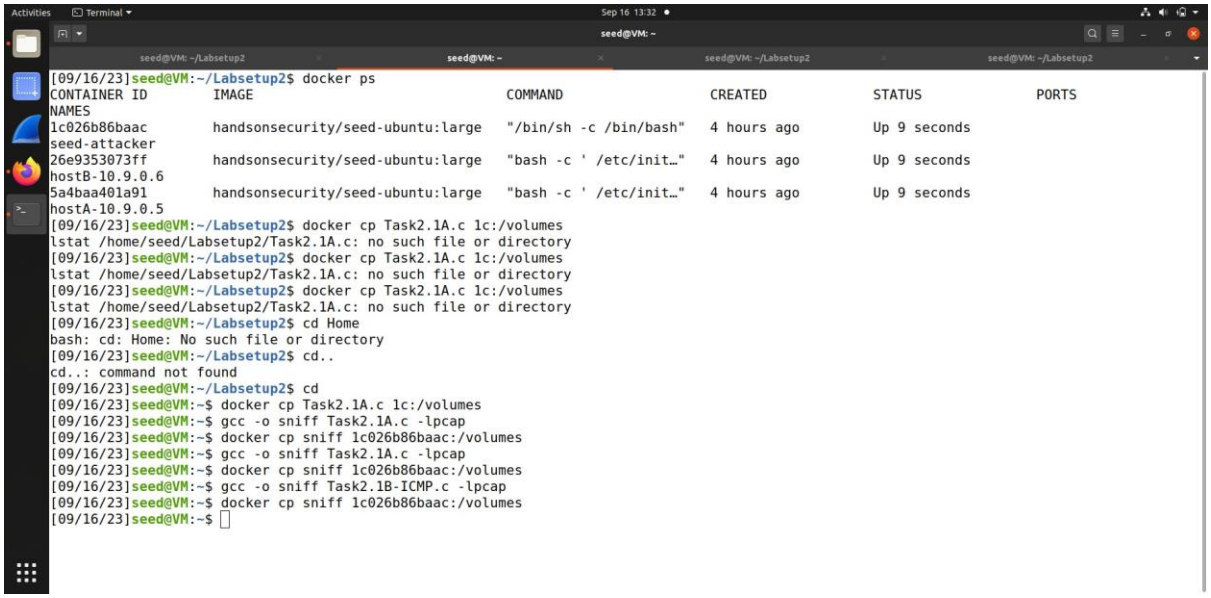


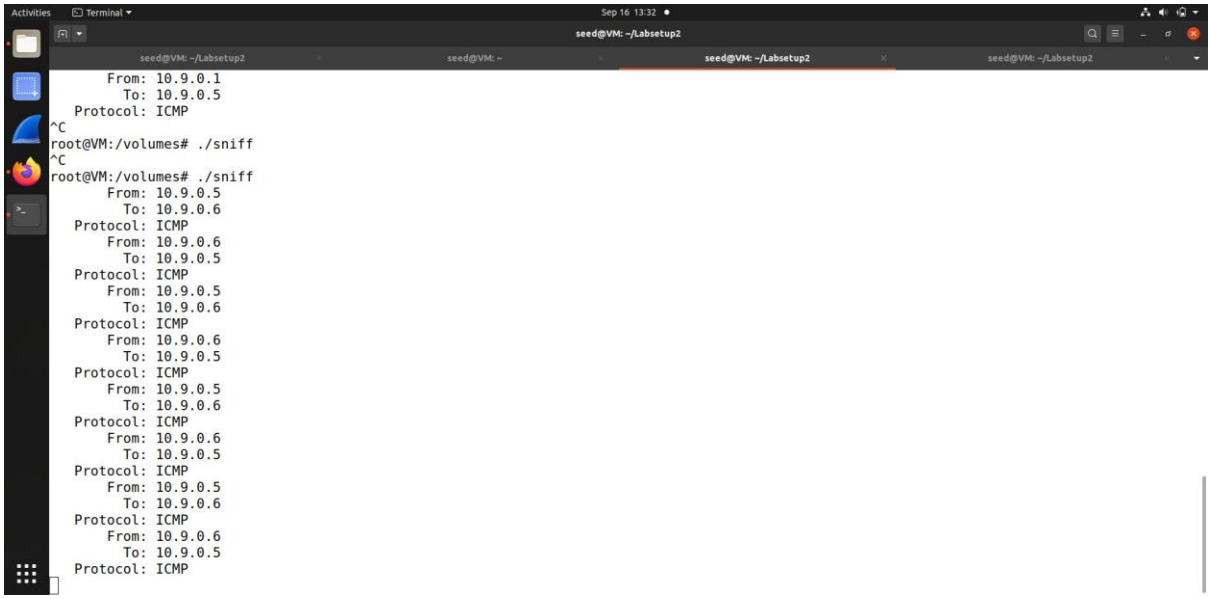
3. In the case of promiscuous mode, when it is turned on, all the packets sent in that particular network are captured. However, on turning it off , the packets which are sent between the host machine and the other machine are only captured.

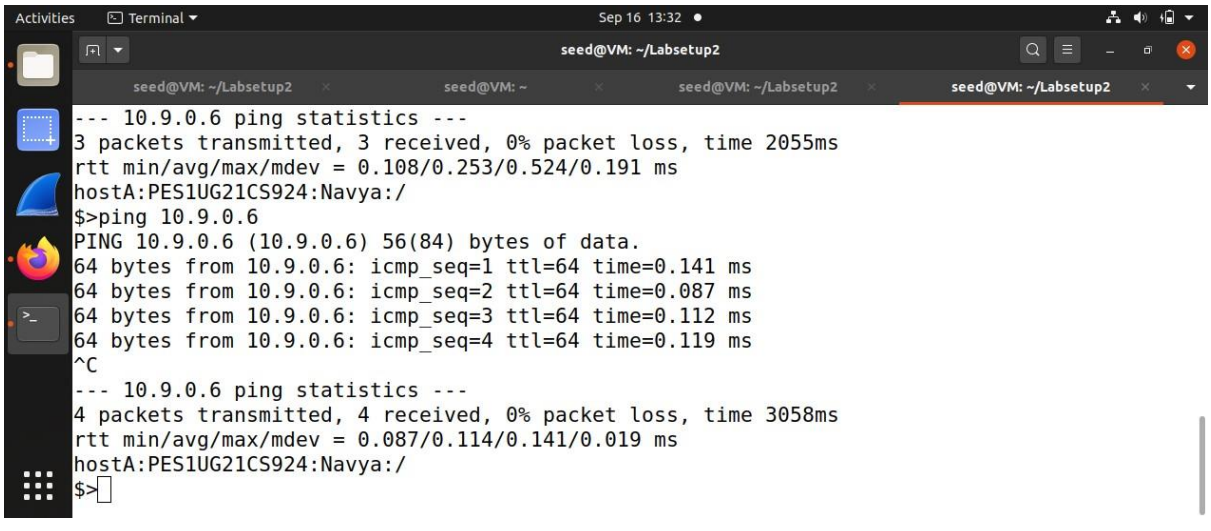
Task 2.1 B

ICMP

This process is similar to the previous process, where the packets sent and received are captured. However, here only the packets sent and received between the hosts A and B are captured and the rest are ignored.

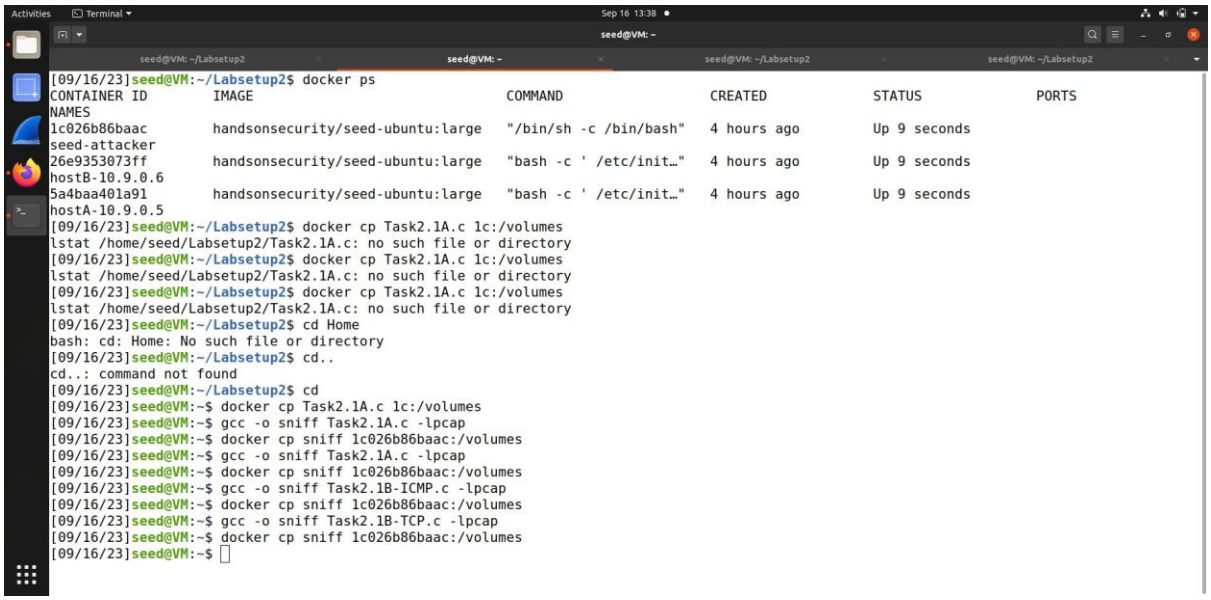

Activities
Terminal
Sep 16 13:32
seed@VM: ~
seed@VM: ~/Labsetup2
seed@VM: ~
seed@VM: ~/Labsetup2
seed@VM: ~/Labsetup2
[09/16/23]seed@VM:~/Labsetup2$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
1c026b86baac handsonsecurity/seed-ubuntu:large "/bin/sh -c /bin/bash" 4 hours ago Up 9 seconds
seed-attacker
26e9353073ff handsonsecurity/seed-ubuntu:large "bash -c ' /etc/init…" 4 hours ago Up 9 seconds
hostB-10.9.0.6
5a4baa401a91 handsonsecurity/seed-ubuntu:large "bash -c ' /etc/init…" 4 hours ago Up 9 seconds
hostA-10.9.0.5
[09/16/23]seed@VM:~/Labsetup2$ docker cp Task2.1A.c 1c:/volumes
lstat /home/seed/Labsetup2/Task2.1A.c: no such file or directory
[09/16/23]seed@VM:~/Labsetup2$ docker cp Task2.1A.c 1c:/volumes
lstat /home/seed/Labsetup2/Task2.1A.c: no such file or directory
[09/16/23]seed@VM:~/Labsetup2$ docker cp Task2.1A.c 1c:/volumes
lstat /home/seed/Labsetup2/Task2.1A.c: no such file or directory
[09/16/23]seed@VM:~/Labsetup2$ cd Home
bash: cd: Home: No such file or directory
[09/16/23]seed@VM:~/Labsetup2$ cd..
cd..: command not found
[09/16/23]seed@VM:~/Labsetup2$ cd
[09/16/23]seed@VM:~$ docker cp Task2.1A.c 1c:/volumes
[09/16/23]seed@VM:~$ gcc -o sniff Task2.1A.c -lpcap
[09/16/23]seed@VM:~$ docker cp sniff 1c026b86baac:/volumes
[09/16/23]seed@VM:~$ gcc -o sniff Task2.1A.c -lpcap
[09/16/23]seed@VM:~$ docker cp sniff 1c026b86baac:/volumes
[09/16/23]seed@VM:~$ gcc -o sniff Task2.1B-ICMP.c -lpcap
[09/16/23]seed@VM:~$ docker cp sniff 1c026b86baac:/volumes
[09/16/23]seed@VM:~$

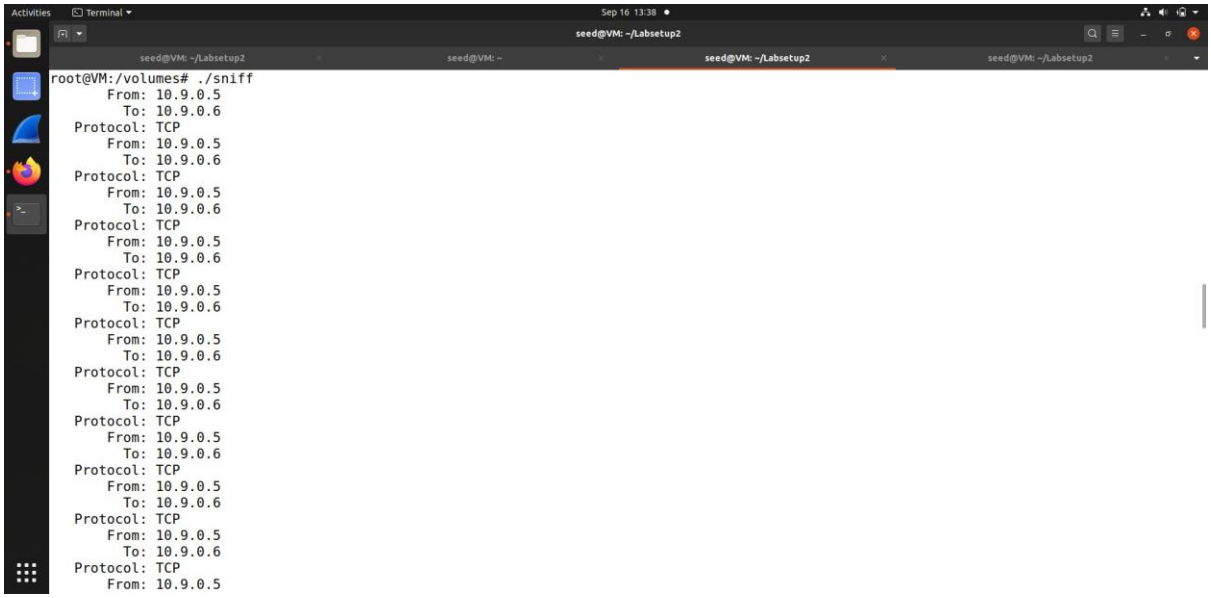

Activities
Terminal
Sep 16 13:32
seed@VM: ~/Labsetup2
seed@VM: ~/Labsetup2
seed@VM: ~
seed@VM: ~/Labsetup2
seed@VM: ~/Labsetup2
From: 10.9.0.1
To: 10.9.0.5
Protocol: ICMP
^C
root@VM:/volumes# ./sniff
^C
root@VM:/volumes# ./sniff
From: 10.9.0.5
To: 10.9.0.6
Protocol: ICMP
From: 10.9.0.6
To: 10.9.0.5
Protocol: ICMP
From: 10.9.0.5
To: 10.9.0.6
Protocol: ICMP
From: 10.9.0.6
To: 10.9.0.5
Protocol: ICMP
From: 10.9.0.5
To: 10.9.0.6
Protocol: ICMP
From: 10.9.0.6
To: 10.9.0.5
Protocol: ICMP
From: 10.9.0.5
To: 10.9.0.6
Protocol: ICMP
From: 10.9.0.6
To: 10.9.0.5
Protocol: ICMP


Activities
Terminal
Sep 16 13:32
seed@VM: ~/Labsetup2
seed@VM: ~/Labsetup2
seed@VM: ~
seed@VM: ~/Labsetup2
seed@VM: ~/Labsetup2
--- 10.9.0.6 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2055ms
rtt min/avg/max/mdev = 0.108/0.253/0.524/0.191 ms
hostA:PES1UG21CS924:Navya:/
$>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=0.141 ms
64 bytes from 10.9.0.6: icmp_seq=2 ttl=64 time=0.087 ms
64 bytes from 10.9.0.6: icmp_seq=3 ttl=64 time=0.112 ms
64 bytes from 10.9.0.6: icmp_seq=4 ttl=64 time=0.119 ms
^C
--- 10.9.0.6 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3058ms
rtt min/avg/max/mdev = 0.087/0.114/0.141/0.019 ms
hostA:PES1UG21CS924:Navya:/
$>

TCP

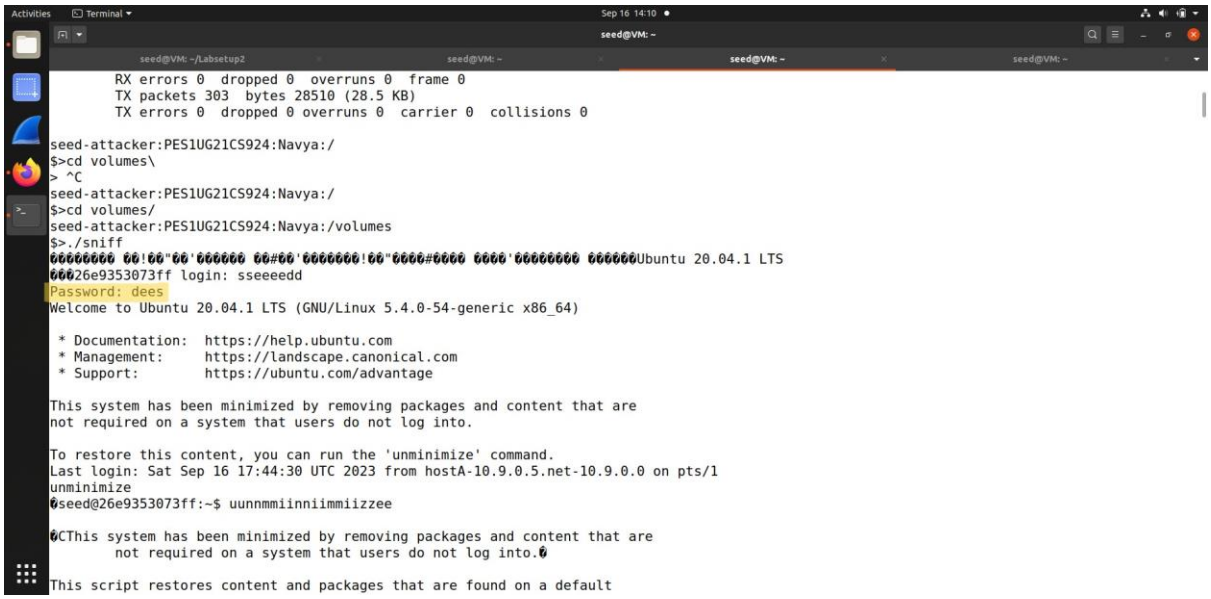
This process is also similar to the previous process, however, only the TCP packets sent between ports 10 and 100, between he given two machines are captured.

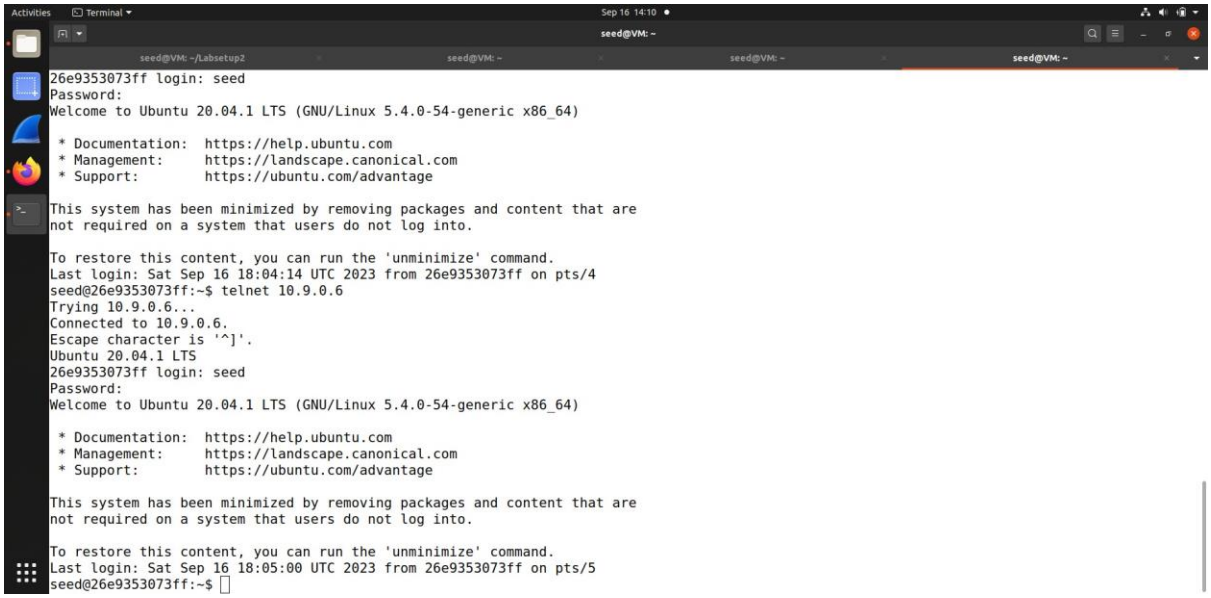




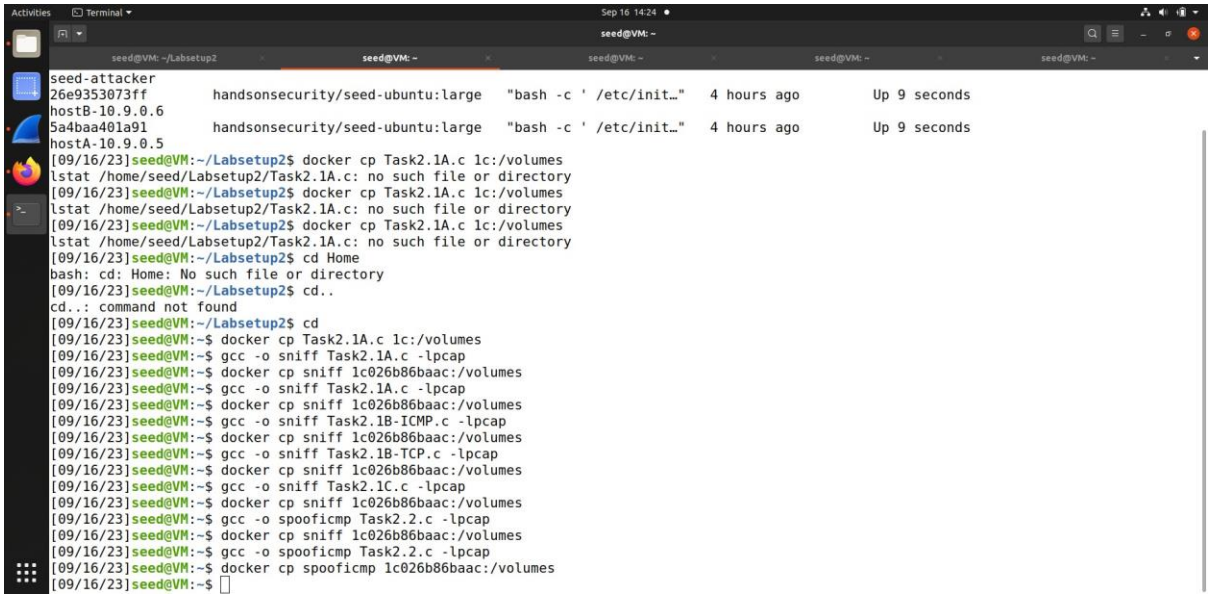
Task 2.1C :SNIFFING PASSWORDS

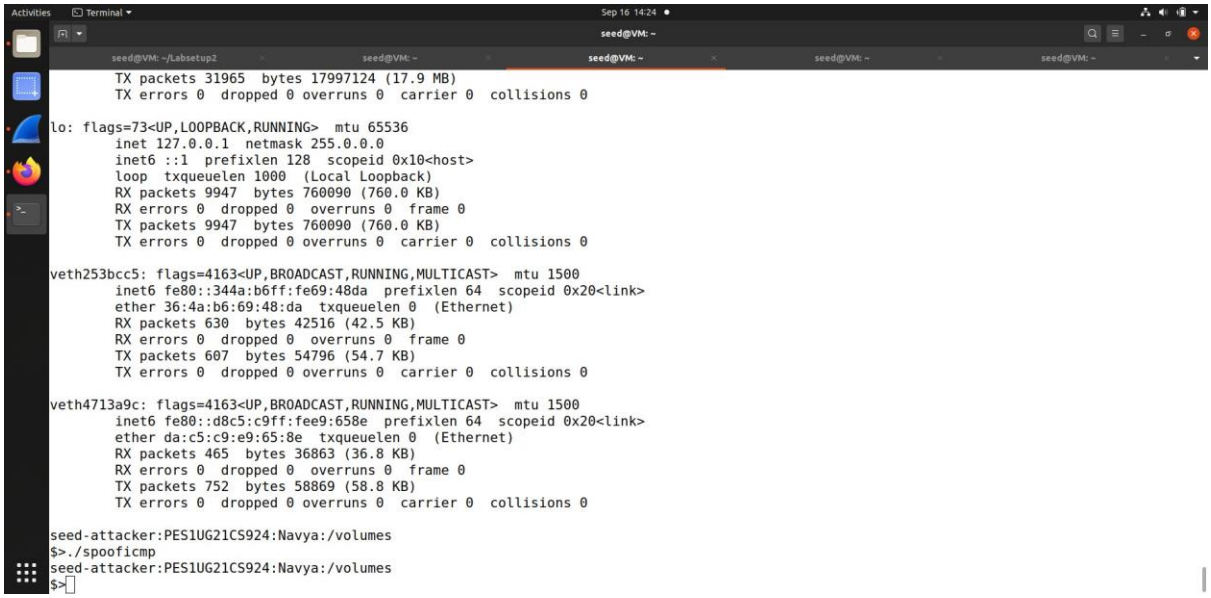
In the given process, the BPF filter is set to the telnet port, which is 23. Hence, only the telnet packets are captured and displayed. Since, telnet also has the ability of remote monitoring, we can view all the information on the Host A's screen along with the password given there, as shown in the photos.

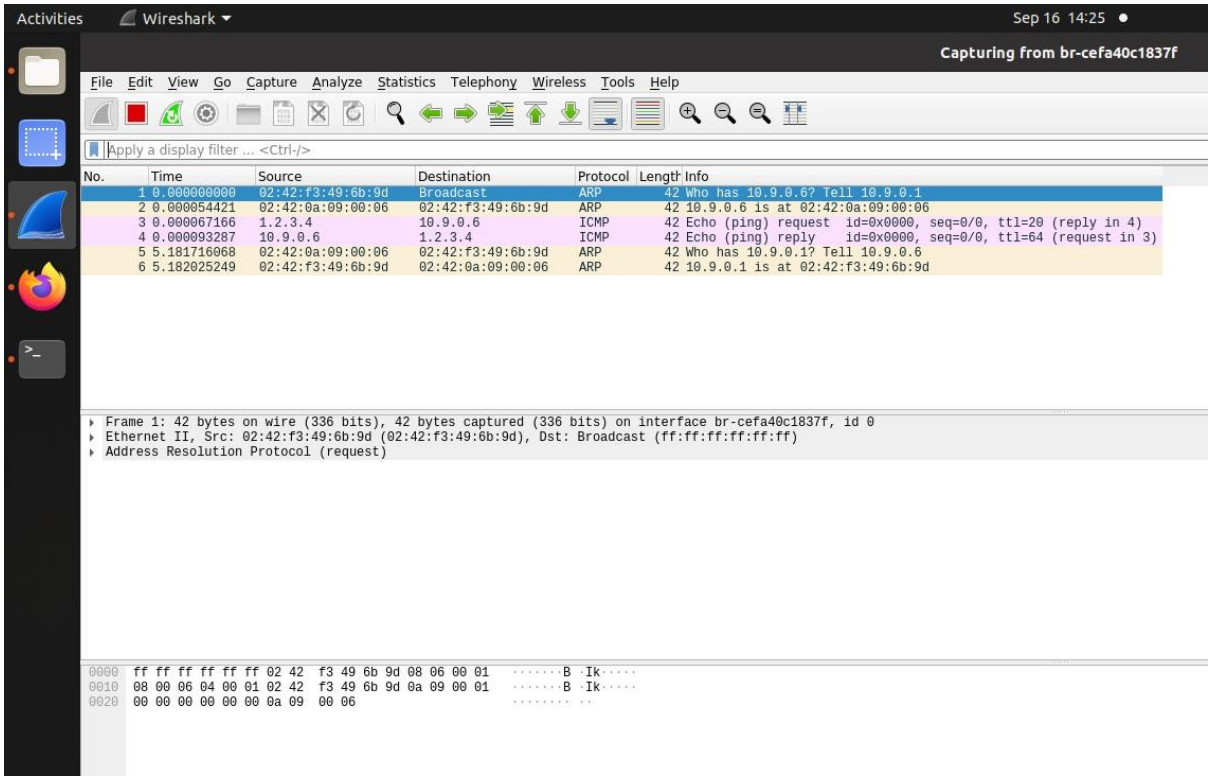




Task 2.2: SPOOFING







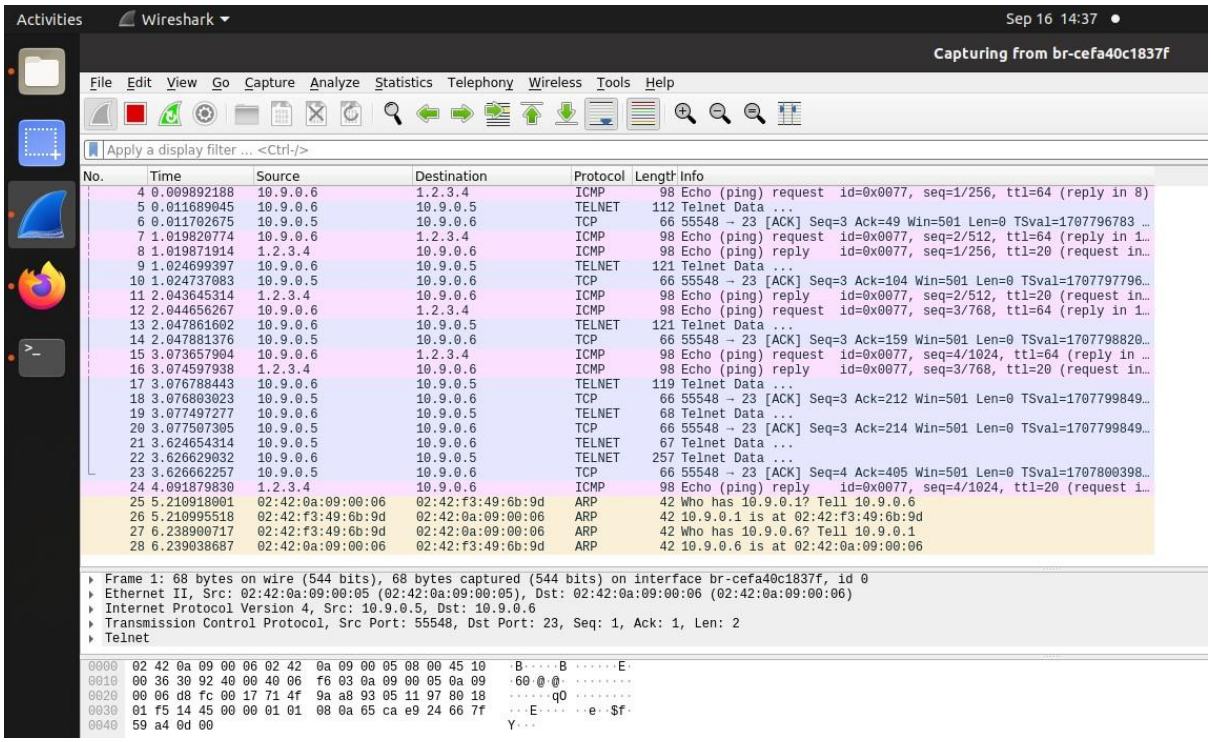
In the above process, icmp packets are spoofed and are sent. We use the in_inchksum() function in the code, to calculate the checksum values of the icmp protocol. The packet we spoofed is ahown to have a source IP address of 1.2.3.4 and is shown to be sent to the destination 10.9.0.6, which is our host B.

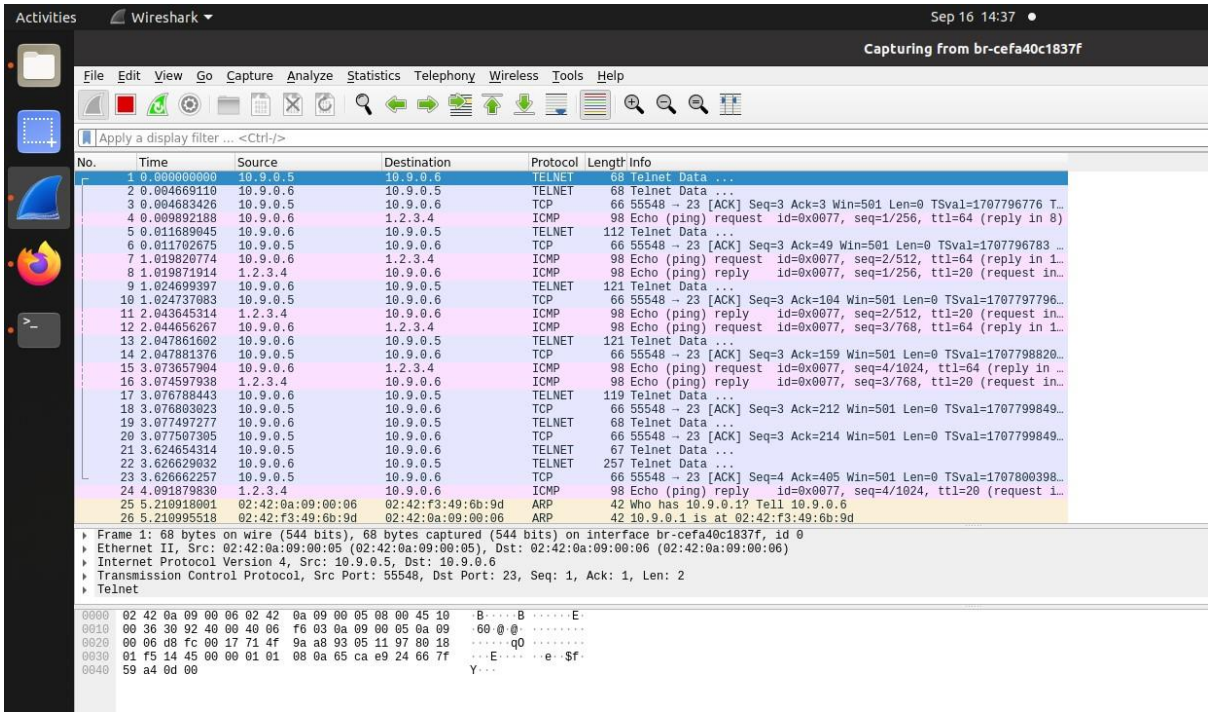
4. No, we don't need to calculate the checksum for the IP headers as this is already done by the system. In case, there is an occurrence of an incorrect value, then the system would calculate and correct it and then replace it with the right value.

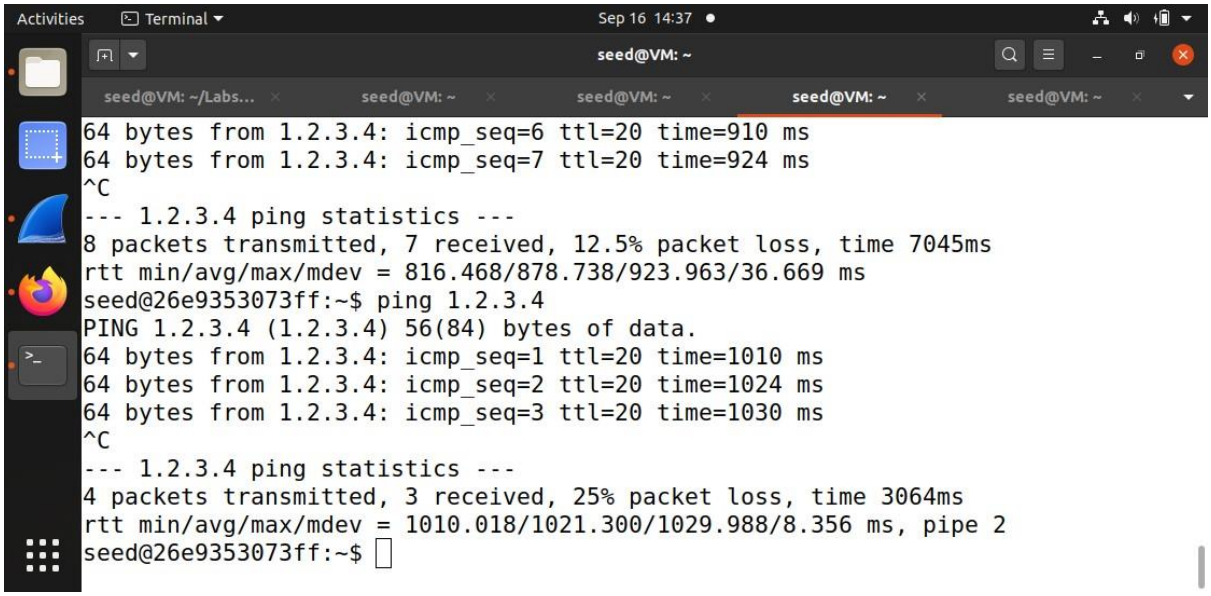
5. We need the root privilege to run the programs in promiscuous mode and to implement raw sockets. When we run a program in the absence of the root privilege, then it will fail at the socket setup stage .

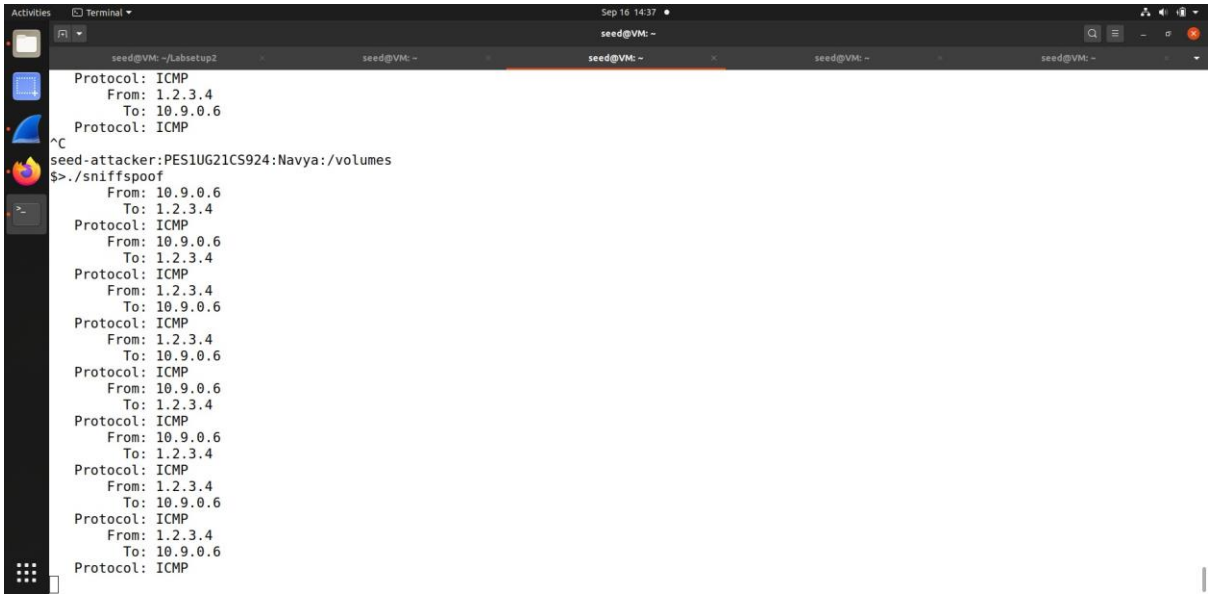
Task 2.3: SNIFF AND SPOOF

In the given process, all the icmp packets which are sent and received are captured. We use various structures such as ipheader, ethheader and the icmpheader to store iformation about their respective protocols – ip, ethernet and icmp. Similar to the codes in the above processes, the in_chcksum() calculates the checksum for both the ip and icmp protocols and the got_packet() function once on receiving the packet, it extracts the information and displays the source and destination ip addresses along with the packet protocol. We also use the send_raw_ip_packet(), to spoof a packet by creating a raw socket and then sending the spoofed packet through it. Since, we have both the attacker and the hosts on the same network, an attacker can easily sniff any packet and spoof a reply immediately, which can be confused by the host as of an actual reply sent by the ip address it has pinged.




Activities
Wireshark
Sep 16 14:37
Capturing from br-cefa40c1837f
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1 0.000000000 10.9.0.5 10.9.0.6 TELNET 68 Telnet Data ...
2 0.004669110 10.9.0.6 10.9.0.5 TELNET 68 Telnet Data ...
3 0.004683426 10.9.0.5 10.9.0.6 TCP 66 55548 → 23 [ACK] Seq=3 Ack=3 Win=501 Len=0 TSval=1707796776 T…
4 0.009892188 10.9.0.6 1.2.3.4 ICMP 98 Echo (ping) request id=0x0077, seq=1/256, ttl=64 (reply in 8)
5 0.011689045 10.9.0.6 10.9.0.5 TELNET 112 Telnet Data ...
6 0.011702675 10.9.0.5 10.9.0.6 TCP 66 55548 → 23 [ACK] Seq=3 Ack=49 Win=501 Len=0 TSval=1707796783 …
7 1.019820774 10.9.0.6 1.2.3.4 ICMP 98 Echo (ping) request id=0x0077, seq=2/512, ttl=64 (reply in 1…
8 1.019871914 1.2.3.4 10.9.0.6 ICMP 98 Echo (ping) reply id=0x0077, seq=1/256, ttl=20 (request in…
9 1.024699397 10.9.0.6 10.9.0.5 TELNET 121 Telnet Data ...
10 1.024737083 10.9.0.5 10.9.0.6 TCP 66 55548 → 23 [ACK] Seq=3 Ack=104 Win=501 Len=0 TSval=1707797796…
11 2.043645314 1.2.3.4 10.9.0.6 ICMP 98 Echo (ping) reply id=0x0077, seq=2/512, ttl=20 (request in…
12 2.044656267 10.9.0.6 1.2.3.4 ICMP 98 Echo (ping) request id=0x0077, seq=3/768, ttl=64 (reply in 1…
13 2.047861602 10.9.0.6 10.9.0.5 TELNET 121 Telnet Data ...
14 2.047881376 10.9.0.5 10.9.0.6 TCP 66 55548 → 23 [ACK] Seq=3 Ack=159 Win=501 Len=0 TSval=1707798820…
15 3.073657904 10.9.0.6 1.2.3.4 ICMP 98 Echo (ping) request id=0x0077, seq=4/1024, ttl=64 (reply in …
16 3.074597938 1.2.3.4 10.9.0.6 ICMP 98 Echo (ping) reply id=0x0077, seq=3/768, ttl=20 (request in…
17 3.076788443 10.9.0.6 10.9.0.5 TELNET 119 Telnet Data ...
18 3.076803023 10.9.0.5 10.9.0.6 TCP 66 55548 → 23 [ACK] Seq=3 Ack=212 Win=501 Len=0 TSval=1707799849…
19 3.077497277 10.9.0.6 10.9.0.5 TELNET 68 Telnet Data ...
20 3.077507305 10.9.0.5 10.9.0.6 TCP 66 55548 → 23 [ACK] Seq=3 Ack=214 Win=501 Len=0 TSval=1707799849…
21 3.624654314 10.9.0.5 10.9.0.6 TELNET 67 Telnet Data ...
22 3.626629032 10.9.0.6 10.9.0.5 TELNET 257 Telnet Data ...
23 3.626662257 10.9.0.5 10.9.0.6 TCP 66 55548 → 23 [ACK] Seq=4 Ack=405 Win=501 Len=0 TSval=1707800398…
24 4.091879830 1.2.3.4 10.9.0.6 ICMP 98 Echo (ping) reply id=0x0077, seq=4/1024, ttl=20 (request i…
25 5.210918001 02:42:0a:09:00:06 02:42:f3:49:6b:9d ARP 42 Who has 10.9.0.1? Tell 10.9.0.6
26 5.210995518 02:42:f3:49:6b:9d 02:42:0a:09:00:06 ARP 42 10.9.0.1 is at 02:42:f3:49:6b:9d
Frame 1: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface br-cefa40c1837f, id 0
Ethernet II, Src: 02:42:0a:09:00:05 (02:42:0a:09:00:05), Dst: 02:42:0a:09:00:06 (02:42:0a:09:00:06)
Internet Protocol Version 4, Src: 10.9.0.5, Dst: 10.9.0.6
Transmission Control Protocol, Src Port: 55548, Dst Port: 23, Seq: 1, Ack: 1, Len: 2
Telnet
0000 02 42 0a 09 00 06 02 42 0a 09 00 05 08 00 45 10 ·B·····B ····· E·
0010 00 36 30 92 40 00 40 06 f6 03 0a 09 00 05 0a 09 ·60·@·@· ·····
0020 00 06 d8 fc 00 17 71 4f 9a a8 93 05 11 97 80 18 ·····qO ·····
0030 01 f5 14 45 00 00 01 01 08 0a 65 ca e9 24 66 7f ···E···· ··e··$f·
0040 59 a4 0d 00 Y···


Activities
Terminal
Sep 16 14:37
seed@VM: ~
seed@VM: ~/Labs...
seed@VM: ~
seed@VM: ~
seed@VM: ~
seed@VM: ~
64 bytes from 1.2.3.4: icmp_seq=6 ttl=20 time=910 ms
64 bytes from 1.2.3.4: icmp_seq=7 ttl=20 time=924 ms
^C
--- 1.2.3.4 ping statistics ---
8 packets transmitted, 7 received, 12.5% packet loss, time 7045ms
rtt min/avg/max/mdev = 816.468/878.738/923.963/36.669 ms
seed@26e9353073ff:~$ ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
64 bytes from 1.2.3.4: icmp_seq=1 ttl=20 time=1010 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=20 time=1024 ms
64 bytes from 1.2.3.4: icmp_seq=3 ttl=20 time=1030 ms
^C
--- 1.2.3.4 ping statistics ---
4 packets transmitted, 3 received, 25% packet loss, time 3064ms
rtt min/avg/max/mdev = 1010.018/1021.300/1029.988/8.356 ms, pipe 2
seed@26e9353073ff:~$


Activities
Terminal
Sep 16 14:37
seed@VM: ~
seed@VM: ~/Labsetup2
seed@VM: ~
seed@VM: ~
seed@VM: ~
seed@VM: ~
Protocol: ICMP
From: 1.2.3.4
To: 10.9.0.6
Protocol: ICMP
^C
seed-attacker:PES1UG21CS924:Navya:/volumes
$>./sniffspoof
From: 10.9.0.6
To: 1.2.3.4
Protocol: ICMP
From: 10.9.0.6
To: 1.2.3.4
Protocol: ICMP
From: 1.2.3.4
To: 10.9.0.6
Protocol: ICMP
From: 1.2.3.4
To: 10.9.0.6
Protocol: ICMP
From: 10.9.0.6
To: 1.2.3.4
Protocol: ICMP
From: 10.9.0.6
To: 1.2.3.4
Protocol: ICMP
From: 1.2.3.4
To: 10.9.0.6
Protocol: ICMP
From: 1.2.3.4
To: 10.9.0.6
Protocol: ICMP