Computer Network Security

UE23CS343AB6

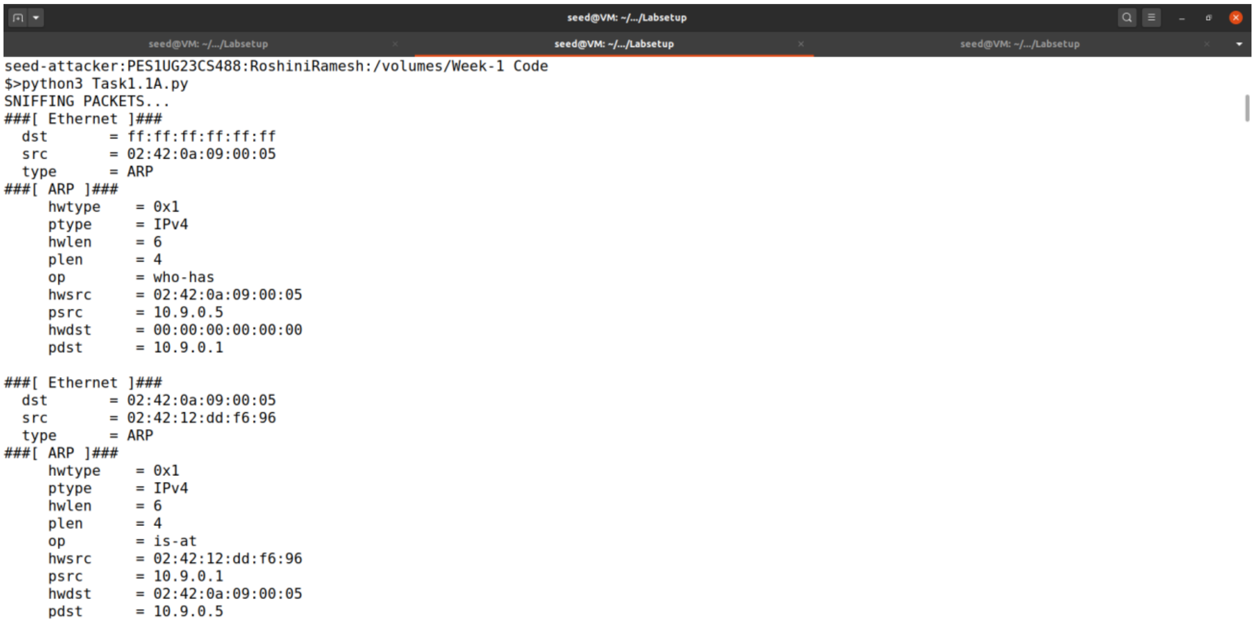
5th Semester, Academic Year 2023

Date: 12/08/2025

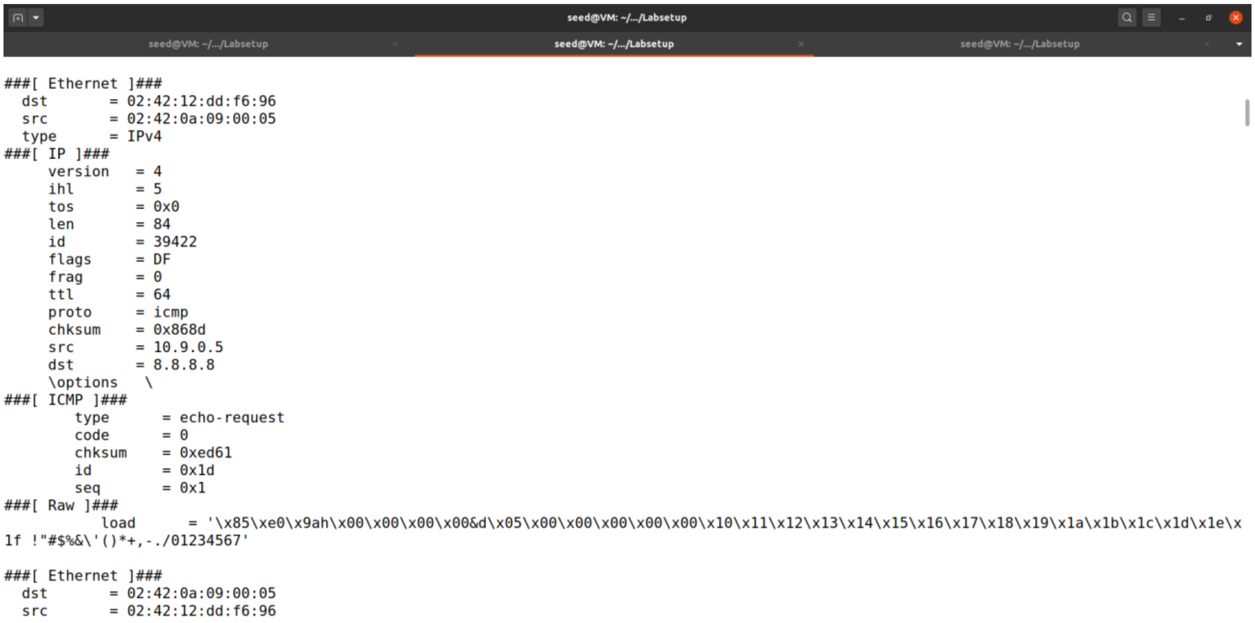
Name: Roshini Ramesh	SRN: PES1UG23CS488	Section: H

TASK 1.1A STEP-1: SNIFF IP PACKETS USING SCAPY (WITH ROOT PRIVILEGES)

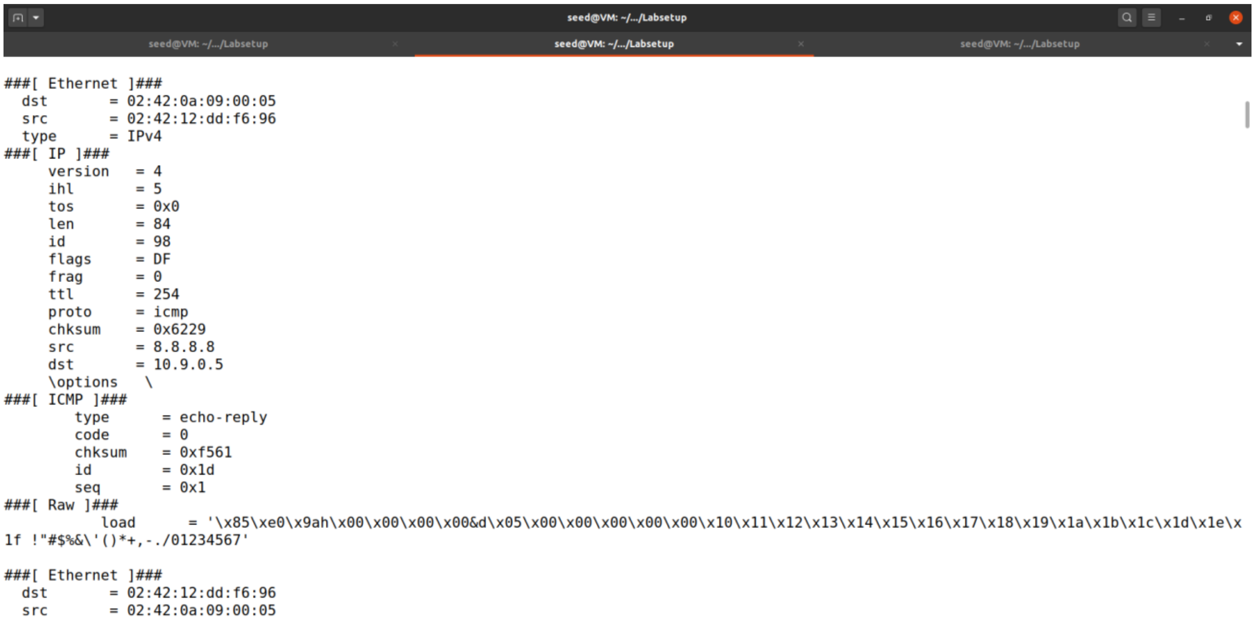
Terminal Output:



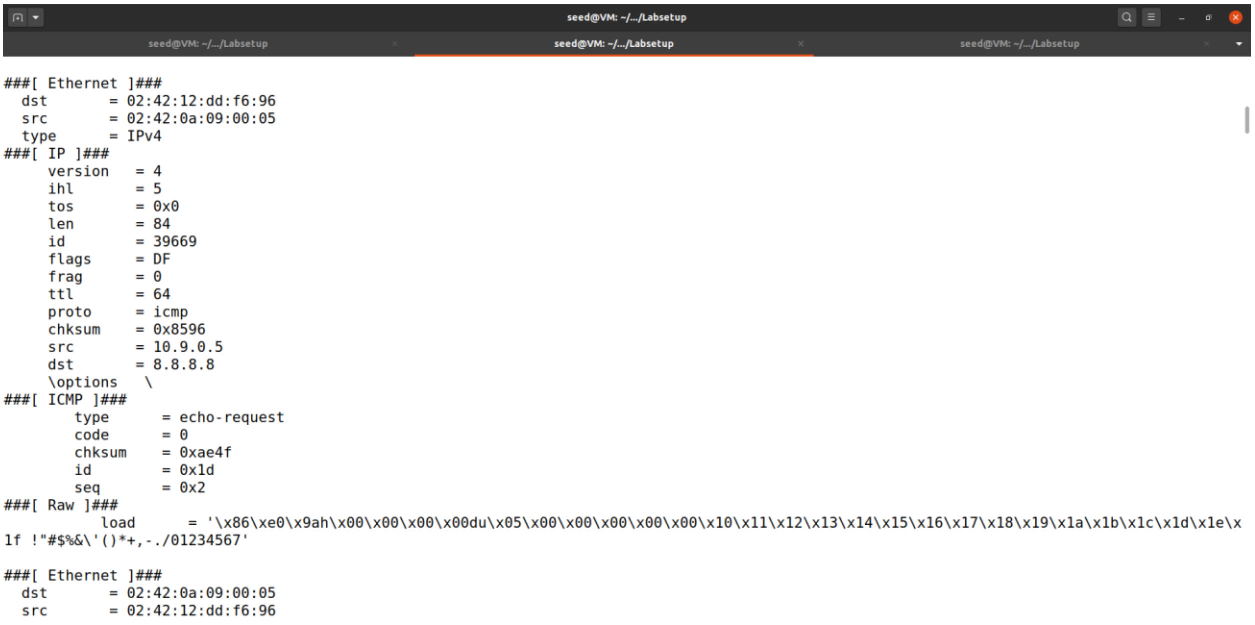
```
seed-attacker:PES1UG23CS488:RoshiniRamesh:/volumes/Week-1 Code
$>python3 Task1.1A.py
SNIFFING PACKETS...
###[ Ethernet ]###
  dst       = ff:ff:ff:ff:ff:ff
  src       = 02:42:0a:09:00:05
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = 6
     plen      = 4
     op        = who-has
     hwsrc     = 02:42:0a:09:00:05
     psrc      = 10.9.0.5
     hwdst     = 00:00:00:00:00:00
     pdst      = 10.9.0.1

###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:12:dd:f6:96
  type      = ARP
###[ ARP ]###
     hwtype    = 0x1
     ptype     = IPv4
     hwlen     = 6
     plen      = 4
     op        = is-at
     hwsrc     = 02:42:12:dd:f6:96
     psrc      = 10.9.0.1
     hwdst     = 02:42:0a:09:00:05
     pdst      = 10.9.0.5
```



```
###[ Ethernet ]###
  dst       = 02:42:12:dd:f6:96
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 39422
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x868d
     src       = 10.9.0.5
     dst       = 8.8.8.8
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0xed61
        id        = 0x1d
        seq       = 0x1
###[ Raw ]###
           load      = '\x85\xe0\x9ah\x00\x00\x00\x00&d\x05\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x
1f !"#$%&\'()*+,-./01234567'

###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:12:dd:f6:96
```

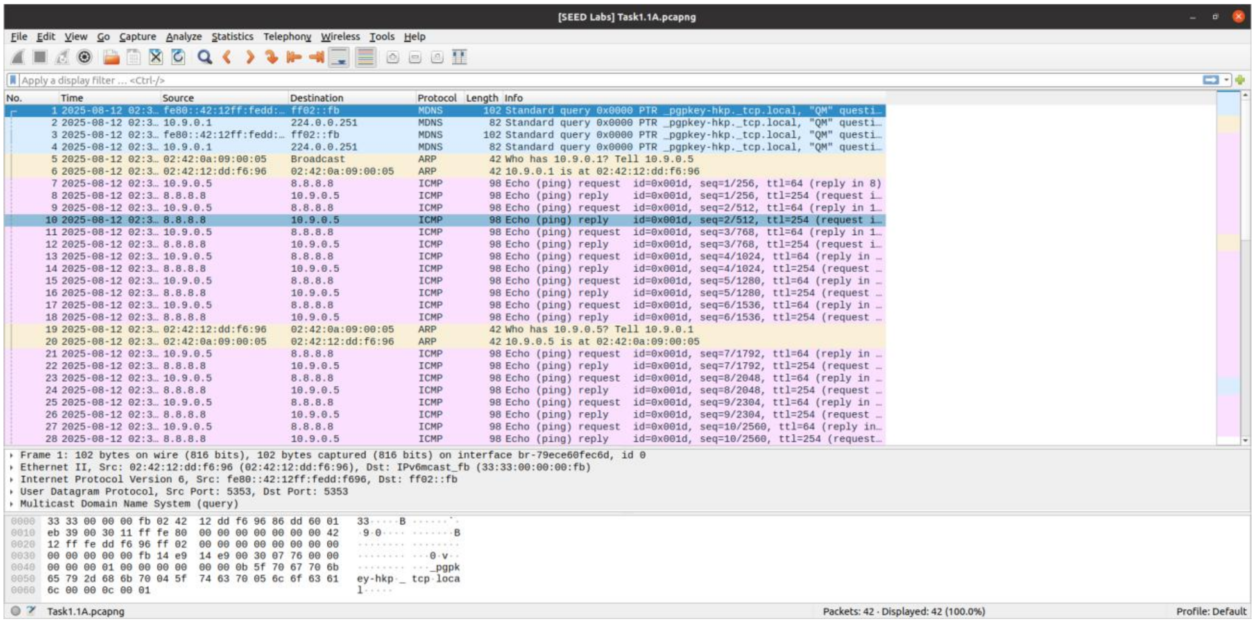


```
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:12:dd:f6:96
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 98
     flags     = DF
     frag      = 0
     ttl       = 254
     proto     = icmp
     chksum    = 0x6229
     src       = 8.8.8.8
     dst       = 10.9.0.5
     \options   \
###[ ICMP ]###
        type      = echo-reply
        code      = 0
        chksum    = 0xf561
        id        = 0x1d
        seq       = 0x1
###[ Raw ]###
           load      = '\x85\xe0\x9ah\x00\x00\x00\x00&d\x05\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x
1f !"#$%&\'()*+,-./01234567'

###[ Ethernet ]###
  dst       = 02:42:12:dd:f6:96
  src       = 02:42:0a:09:00:05
```



```
###[ Ethernet ]###
  dst       = 02:42:12:dd:f6:96
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 39669
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x8596
     src       = 10.9.0.5
     dst       = 8.8.8.8
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0xae4f
        id        = 0x1d
        seq       = 0x2
###[ Raw ]###
           load      = '\x86\xe0\x9ah\x00\x00\x00\x00du\x05\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x
1f !"#$%&\'()*+,-./01234567'

###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:12:dd:f6:96
```

Wireshark Output:


[SEED Labs] Task1.1A.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

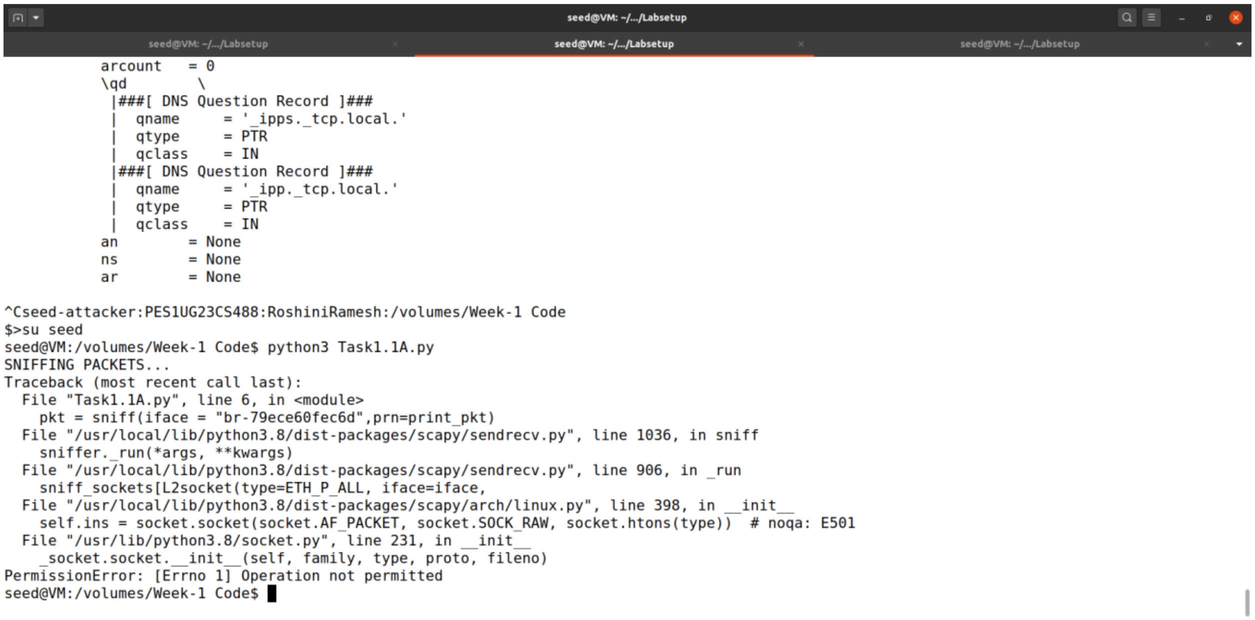
```
No.   Time              Source                 Destination        Protocol Length Info
    1 2025-08-12 02:3… fe80::42:12ff:fedd:… ff02::fb           MDNS      102 Standard query 0x0000 PTR _pgpkey-hkp._tcp.local, "QM" questi…
    2 2025-08-12 02:3… 10.9.0.1              224.0.0.251        MDNS       82 Standard query 0x0000 PTR _pgpkey-hkp._tcp.local, "QM" questi…
    3 2025-08-12 02:3… fe80::42:12ff:fedd:… ff02::fb           MDNS      102 Standard query 0x0000 PTR _pgpkey-hkp._tcp.local, "QM" questi…
    4 2025-08-12 02:3… 10.9.0.1              224.0.0.251        MDNS       82 Standard query 0x0000 PTR _pgpkey-hkp._tcp.local, "QM" questi…
    5 2025-08-12 02:3… 02:42:0a:09:00:05     Broadcast          ARP        42 Who has 10.9.0.1? Tell 10.9.0.5
    6 2025-08-12 02:3… 02:42:12:dd:f6:96     02:42:0a:09:00:05  ARP        42 10.9.0.1 is at 02:42:12:dd:f6:96
    7 2025-08-12 02:3… 10.9.0.5              8.8.8.8            ICMP       98 Echo (ping) request  id=0x001d, seq=1/256, ttl=64 (reply in 8)
    8 2025-08-12 02:3… 8.8.8.8               10.9.0.5           ICMP       98 Echo (ping) reply    id=0x001d, seq=1/256, ttl=254 (request i…
    9 2025-08-12 02:3… 10.9.0.5              8.8.8.8            ICMP       98 Echo (ping) request  id=0x001d, seq=2/512, ttl=64 (reply in 1…
   10 2025-08-12 02:3… 8.8.8.8               10.9.0.5           ICMP       98 Echo (ping) reply    id=0x001d, seq=2/512, ttl=254 (request i…
   11 2025-08-12 02:3… 10.9.0.5              8.8.8.8            ICMP       98 Echo (ping) request  id=0x001d, seq=3/768, ttl=64 (reply in 1…
   12 2025-08-12 02:3… 8.8.8.8               10.9.0.5           ICMP       98 Echo (ping) reply    id=0x001d, seq=3/768, ttl=254 (request i…
   13 2025-08-12 02:3… 10.9.0.5              8.8.8.8            ICMP       98 Echo (ping) request  id=0x001d, seq=4/1024, ttl=64 (reply in …
   14 2025-08-12 02:3… 8.8.8.8               10.9.0.5           ICMP       98 Echo (ping) reply    id=0x001d, seq=4/1024, ttl=254 (request …
   15 2025-08-12 02:3… 10.9.0.5              8.8.8.8            ICMP       98 Echo (ping) request  id=0x001d, seq=5/1280, ttl=64 (reply in …
   16 2025-08-12 02:3… 8.8.8.8               10.9.0.5           ICMP       98 Echo (ping) reply    id=0x001d, seq=5/1280, ttl=254 (request …
   17 2025-08-12 02:3… 10.9.0.5              8.8.8.8            ICMP       98 Echo (ping) request  id=0x001d, seq=6/1536, ttl=64 (reply in …
   18 2025-08-12 02:3… 8.8.8.8               10.9.0.5           ICMP       98 Echo (ping) reply    id=0x001d, seq=6/1536, ttl=254 (request …
   19 2025-08-12 02:3… 02:42:12:dd:f6:96     02:42:0a:09:00:05  ARP        42 Who has 10.9.0.5? Tell 10.9.0.1
   20 2025-08-12 02:3… 02:42:0a:09:00:05     02:42:12:dd:f6:96  ARP        42 10.9.0.5 is at 02:42:0a:09:00:05
   21 2025-08-12 02:3… 10.9.0.5              8.8.8.8            ICMP       98 Echo (ping) request  id=0x001d, seq=7/1792, ttl=64 (reply in …
   22 2025-08-12 02:3… 8.8.8.8               10.9.0.5           ICMP       98 Echo (ping) reply    id=0x001d, seq=7/1792, ttl=254 (request …
   23 2025-08-12 02:3… 10.9.0.5              8.8.8.8            ICMP       98 Echo (ping) request  id=0x001d, seq=8/2048, ttl=64 (reply in …
   24 2025-08-12 02:3… 8.8.8.8               10.9.0.5           ICMP       98 Echo (ping) reply    id=0x001d, seq=8/2048, ttl=254 (request …
   25 2025-08-12 02:3… 10.9.0.5              8.8.8.8            ICMP       98 Echo (ping) request  id=0x001d, seq=9/2304, ttl=64 (reply in …
   26 2025-08-12 02:3… 8.8.8.8               10.9.0.5           ICMP       98 Echo (ping) reply    id=0x001d, seq=9/2304, ttl=254 (request …
   27 2025-08-12 02:3… 10.9.0.5              8.8.8.8            ICMP       98 Echo (ping) request  id=0x001d, seq=10/2560, ttl=64 (reply in…
   28 2025-08-12 02:3… 8.8.8.8               10.9.0.5           ICMP       98 Echo (ping) reply    id=0x001d, seq=10/2560, ttl=254 (request…

› Frame 1: 102 bytes on wire (816 bits), 102 bytes captured (816 bits) on interface br-79ece60fec6d, id 0
› Ethernet II, Src: 02:42:12:dd:f6:96 (02:42:12:dd:f6:96), Dst: IPv6mcast_fb (33:33:00:00:00:fb)
› Internet Protocol Version 6, Src: fe80::42:12ff:fedd:f696, Dst: ff02::fb
› User Datagram Protocol, Src Port: 5353, Dst Port: 5353
› Multicast Domain Name System (query)

0000  33 33 00 00 00 fb 02 42  12 dd f6 96 86 dd 60 01   33·····B ······`·
0010  eb 39 00 30 11 ff fe 80  00 00 00 00 00 00 00 42   ·9·0···· ·······B
0020  12 ff fe dd f6 96 ff 02  00 00 00 00 00 00 00 00   ········ ········
0030  00 00 00 00 00 fb 14 e9  14 e9 00 30 07 76 00 00   ········ ···0·v··
0040  00 00 00 01 00 00 00 00  00 00 0b 5f 70 67 70 6b   ········ ···_pgpk
0050  65 79 2d 68 6b 70 04 5f  74 63 70 05 6c 6f 63 61   ey-hkp·_ tcp·loca
0060  6c 00 00 0c 00 01                                  l·····

Task1.1A.pcapng                                    Packets: 42 · Displayed: 42 (100.0%)                    Profile: Default
```

Explanation:

The python code given uses the sniff function to sniff packets on the network. This requires root privileges, which are given by default to the docker container. So, when host A pings 8.8.8.8, the attacker sniffs the ICMP echo request and reply, as seen in the Wireshark as well as terminal output. The contents of the packets are also clearly visible through this process.

TASK 1.1A STEP-2: SNIFF IP PACKETS USING SCAPY (WITHOUT ROOT PRIVILEGES)

Terminal Output:



```
           arcount   = 0
           \qd        \
            |###[ DNS Question Record ]###
            |  qname     = '_ipps._tcp.local.'
            |  qtype     = PTR
            |  qclass    = IN
            |###[ DNS Question Record ]###
            |  qname     = '_ipp._tcp.local.'
            |  qtype     = PTR
            |  qclass    = IN
           an        = None
           ns        = None
           ar        = None

^Cseed-attacker:PES1UG23CS488:RoshiniRamesh:/volumes/Week-1 Code
$>su seed
seed@VM:/volumes/Week-1 Code$ python3 Task1.1A.py
SNIFFING PACKETS...
Traceback (most recent call last):
  File "Task1.1A.py", line 6, in <module>
    pkt = sniff(iface = "br-79ece60fec6d",prn=print_pkt)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py", line 1036, in sniff
    sniffer._run(*args, **kwargs)
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py", line 906, in _run
    sniff_sockets[L2socket(type=ETH_P_ALL, iface=iface,
  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
seed@VM:/volumes/Week-1 Code$ █
```

Wireshark Output:

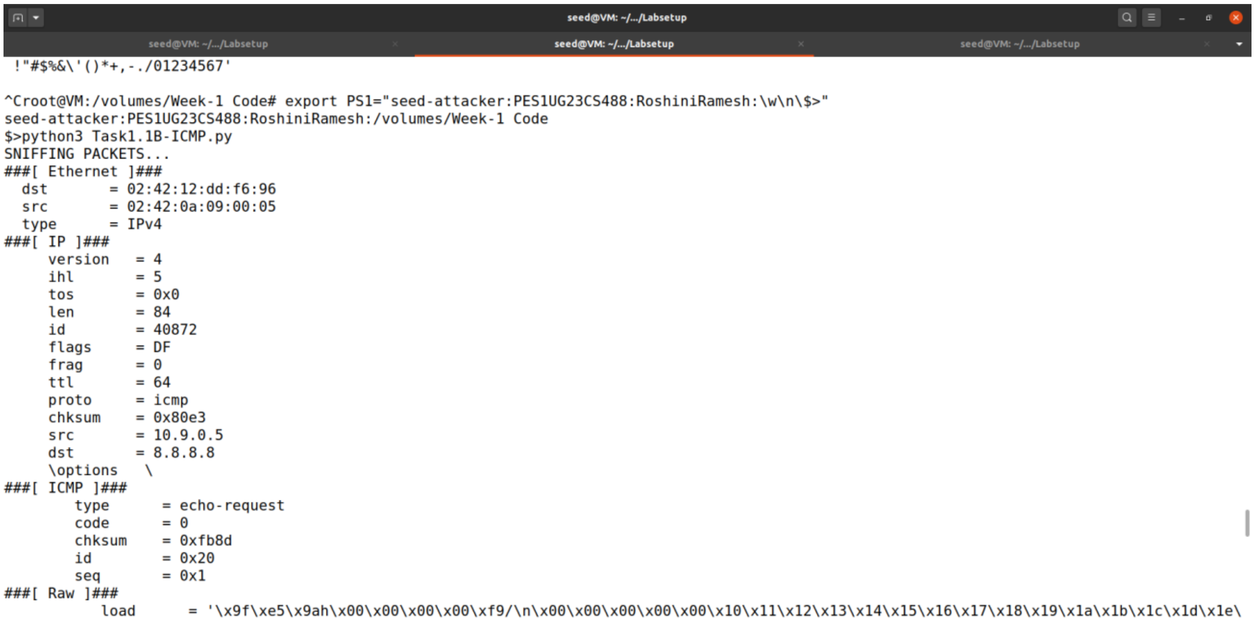
No Wireshark output.

Explanation:

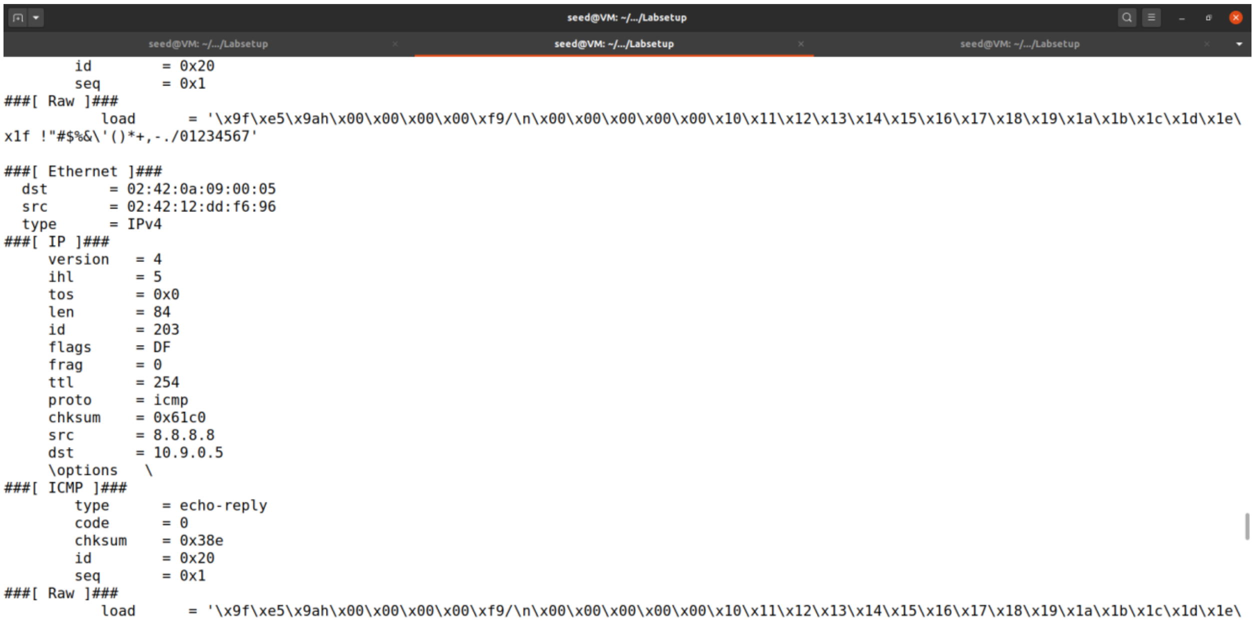
Before executing the code, root privileges are removed. So, when the program is run, we get PermissionError. Therefore, packet sniffing can be done only when attacker has root privileges.

TASK 1.1 B: APPLYING PACKET FILTERS TO CAPTURE ICMP PACKETS

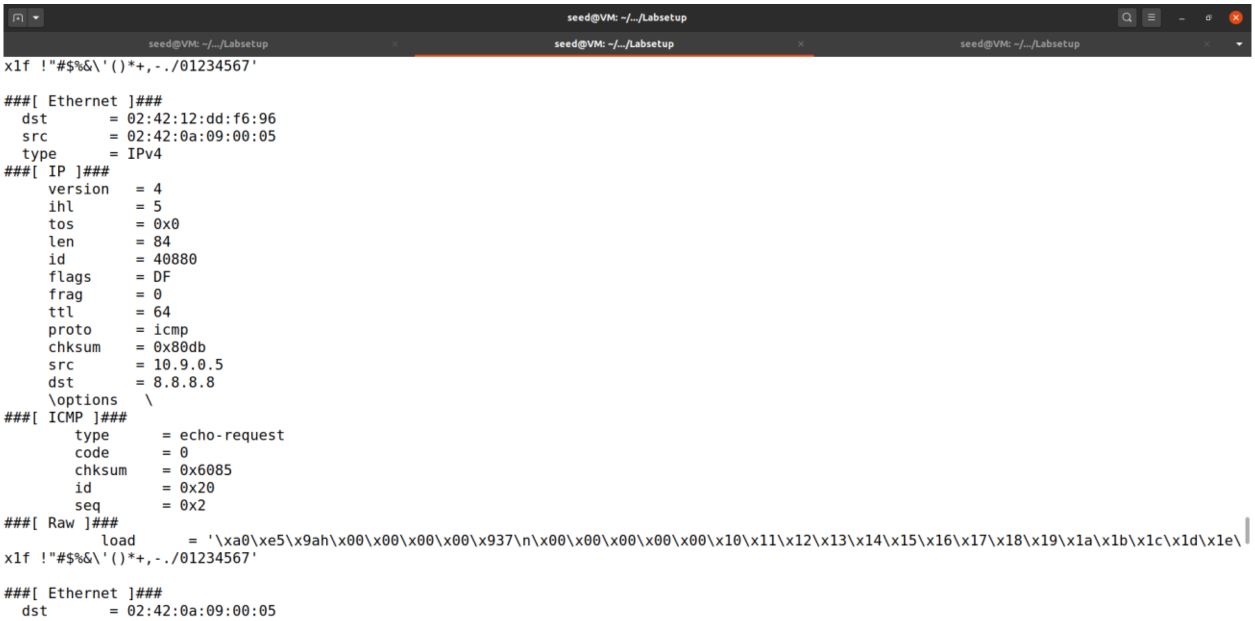
Terminal Output:



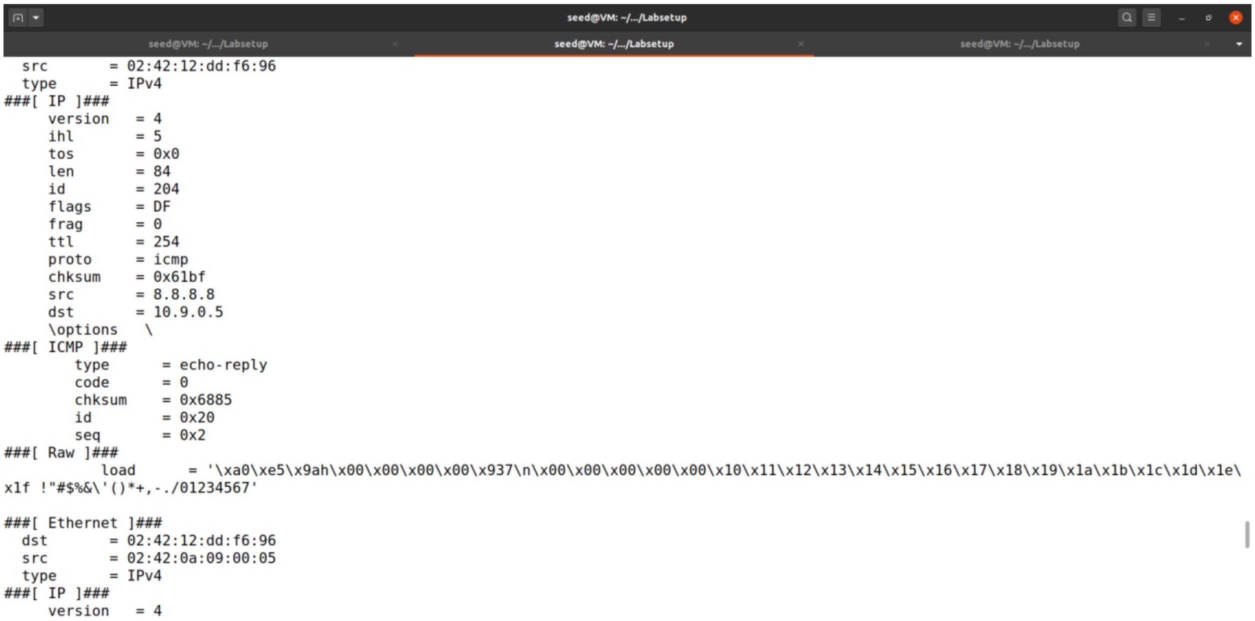
```
 !"#$%&\'()*+,-./01234567'

^Croot@VM:/volumes/Week-1 Code# export PS1="seed-attacker:PES1UG23CS488:RoshiniRamesh:\w\n\$>"
seed-attacker:PES1UG23CS488:RoshiniRamesh:/volumes/Week-1 Code
$>python3 Task1.1B-ICMP.py
SNIFFING PACKETS...
###[ Ethernet ]###
  dst       = 02:42:12:dd:f6:96
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 40872
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x80e3
     src       = 10.9.0.5
     dst       = 8.8.8.8
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0xfb8d
        id        = 0x20
        seq       = 0x1
###[ Raw ]###
           load      = '\x9f\xe5\x9ah\x00\x00\x00\x00\xf9/\n\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\
```



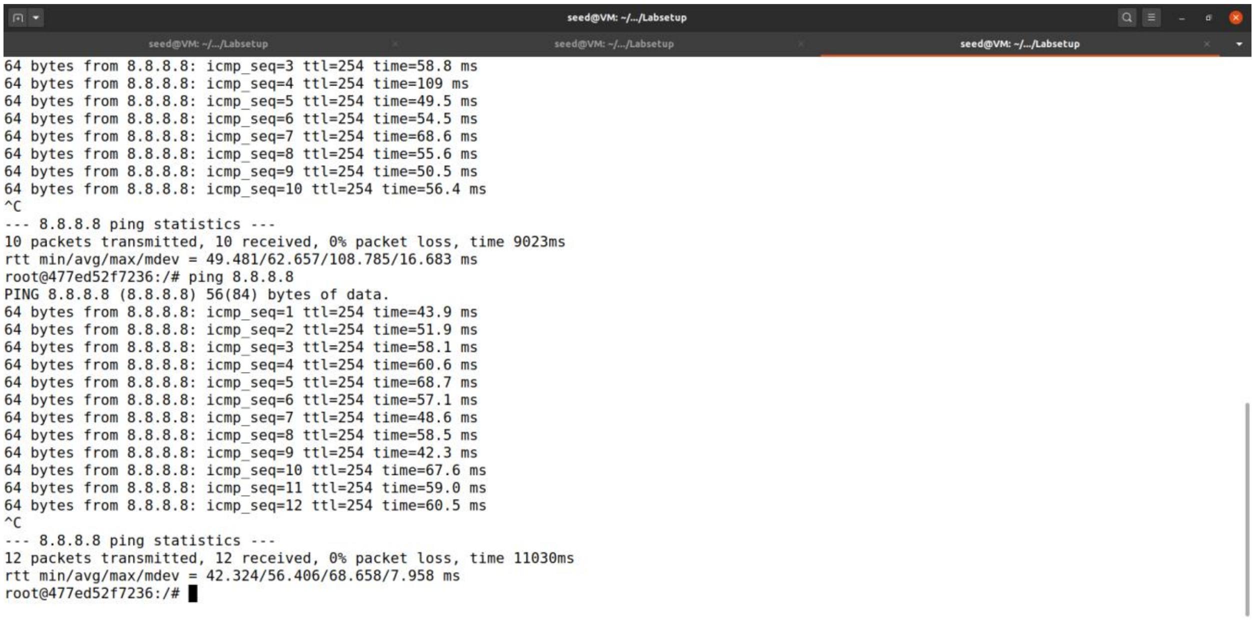
```
        id        = 0x20
        seq       = 0x1
###[ Raw ]###
           load      = '\x9f\xe5\x9ah\x00\x00\x00\x00\xf9/\n\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\
x1f !"#$%&\'()*+,-./01234567'

###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:12:dd:f6:96
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 203
     flags     = DF
     frag      = 0
     ttl       = 254
     proto     = icmp
     chksum    = 0x61c0
     src       = 8.8.8.8
     dst       = 10.9.0.5
     \options   \
###[ ICMP ]###
        type      = echo-reply
        code      = 0
        chksum    = 0x38e
        id        = 0x20
        seq       = 0x1
###[ Raw ]###
           load      = '\x9f\xe5\x9ah\x00\x00\x00\x00\xf9/\n\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\
```



```
x1f !"#$%&\'()*+,-./01234567'

###[ Ethernet ]###
  dst       = 02:42:12:dd:f6:96
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 40880
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x80db
     src       = 10.9.0.5
     dst       = 8.8.8.8
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0x6085
        id        = 0x20
        seq       = 0x2
###[ Raw ]###
           load      = '\xa0\xe5\x9ah\x00\x00\x00\x00\x937\n\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\
x1f !"#$%&\'()*+,-./01234567'

###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
```

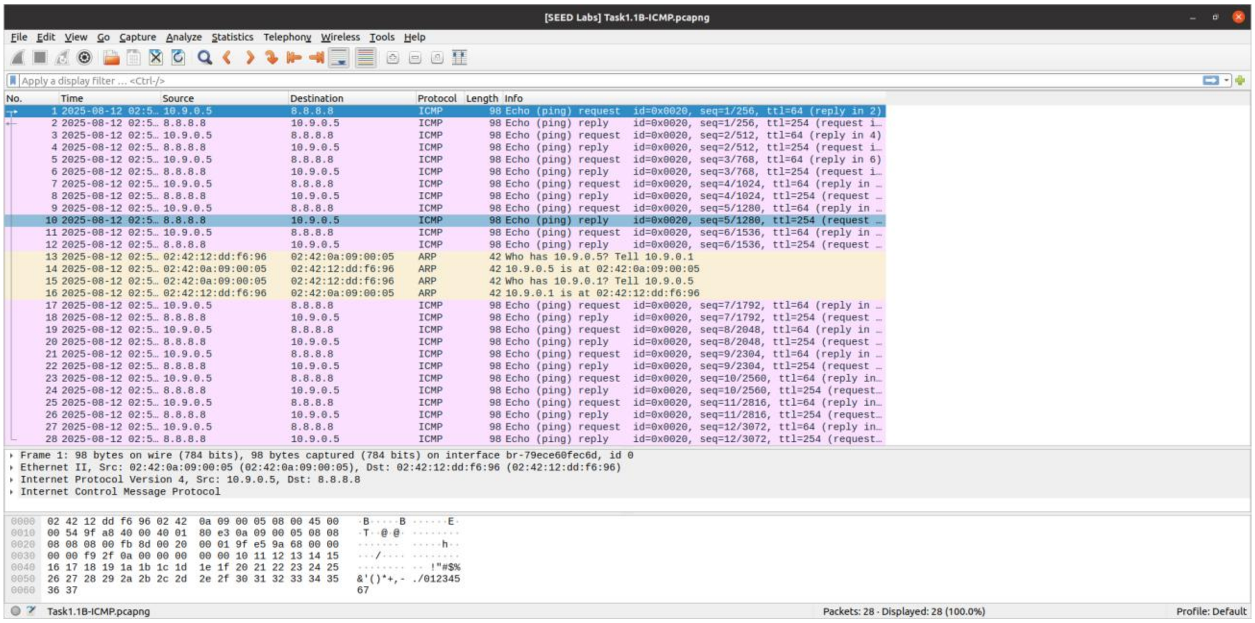


```
  src       = 02:42:12:dd:f6:96
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 204
     flags     = DF
     frag      = 0
     ttl       = 254
     proto     = icmp
     chksum    = 0x61bf
     src       = 8.8.8.8
     dst       = 10.9.0.5
     \options   \
###[ ICMP ]###
        type      = echo-reply
        code      = 0
        chksum    = 0x6885
        id        = 0x20
        seq       = 0x2
###[ Raw ]###
           load      = '\xa0\xe5\x9ah\x00\x00\x00\x00\x937\n\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\
x1f !"#$%&\'()*+,-./01234567'

###[ Ethernet ]###
  dst       = 02:42:12:dd:f6:96
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
```



```
64 bytes from 8.8.8.8: icmp_seq=3 ttl=254 time=58.8 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=254 time=109 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=254 time=49.5 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=254 time=54.5 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=254 time=68.6 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=254 time=55.6 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=254 time=50.5 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=254 time=56.4 ms
^C
--- 8.8.8.8 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9023ms
rtt min/avg/max/mdev = 49.481/62.657/108.785/16.683 ms
root@477ed52f7236:/# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=254 time=43.9 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=254 time=51.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=254 time=58.1 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=254 time=60.6 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=254 time=68.7 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=254 time=57.1 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=254 time=48.6 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=254 time=58.5 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=254 time=42.3 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=254 time=67.6 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=254 time=59.0 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=254 time=60.5 ms
^C
--- 8.8.8.8 ping statistics ---
12 packets transmitted, 12 received, 0% packet loss, time 11030ms
rtt min/avg/max/mdev = 42.324/56.406/68.658/7.958 ms
root@477ed52f7236:/# 
```

Wireshark Output:


[SEED Labs] Task1.1B-ICMP.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	2025-08-12 02:5…	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request id=0x0020, seq=1/256, ttl=64 (reply in 2)
2	2025-08-12 02:5…	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply id=0x0020, seq=1/256, ttl=254 (request i…
3	2025-08-12 02:5…	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request id=0x0020, seq=2/512, ttl=64 (reply in 4)
4	2025-08-12 02:5…	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply id=0x0020, seq=2/512, ttl=254 (request i…
5	2025-08-12 02:5…	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request id=0x0020, seq=3/768, ttl=64 (reply in 6)
6	2025-08-12 02:5…	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply id=0x0020, seq=3/768, ttl=254 (request i…
7	2025-08-12 02:5…	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request id=0x0020, seq=4/1024, ttl=64 (reply in …
8	2025-08-12 02:5…	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply id=0x0020, seq=4/1024, ttl=254 (request …
9	2025-08-12 02:5…	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request id=0x0020, seq=5/1280, ttl=64 (reply in …
10	2025-08-12 02:5…	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply id=0x0020, seq=5/1280, ttl=254 (request …
11	2025-08-12 02:5…	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request id=0x0020, seq=6/1536, ttl=64 (reply in …
12	2025-08-12 02:5…	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply id=0x0020, seq=6/1536, ttl=254 (request …
13	2025-08-12 02:5…	02:42:12:dd:f6:96	02:42:0a:09:00:05	ARP	42	Who has 10.9.0.5? Tell 10.9.0.1
14	2025-08-12 02:5…	02:42:0a:09:00:05	02:42:12:dd:f6:96	ARP	42	10.9.0.5 is at 02:42:0a:09:00:05
15	2025-08-12 02:5…	02:42:0a:09:00:05	02:42:12:dd:f6:96	ARP	42	Who has 10.9.0.1? Tell 10.9.0.5
16	2025-08-12 02:5…	02:42:12:dd:f6:96	02:42:0a:09:00:05	ARP	42	10.9.0.1 is at 02:42:12:dd:f6:96
17	2025-08-12 02:5…	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request id=0x0020, seq=7/1792, ttl=64 (reply in …
18	2025-08-12 02:5…	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply id=0x0020, seq=7/1792, ttl=254 (request …
19	2025-08-12 02:5…	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request id=0x0020, seq=8/2048, ttl=64 (reply in …
20	2025-08-12 02:5…	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply id=0x0020, seq=8/2048, ttl=254 (request …
21	2025-08-12 02:5…	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request id=0x0020, seq=9/2304, ttl=64 (reply in …
22	2025-08-12 02:5…	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply id=0x0020, seq=9/2304, ttl=254 (request …
23	2025-08-12 02:5…	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request id=0x0020, seq=10/2560, ttl=64 (reply in…
24	2025-08-12 02:5…	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply id=0x0020, seq=10/2560, ttl=254 (request…
25	2025-08-12 02:5…	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request id=0x0020, seq=11/2816, ttl=64 (reply in…
26	2025-08-12 02:5…	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply id=0x0020, seq=11/2816, ttl=254 (request…
27	2025-08-12 02:5…	10.9.0.5	8.8.8.8	ICMP	98	Echo (ping) request id=0x0020, seq=12/3072, ttl=64 (reply in…
28	2025-08-12 02:5…	8.8.8.8	10.9.0.5	ICMP	98	Echo (ping) reply id=0x0020, seq=12/3072, ttl=254 (request…

```
› Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface br-79ece60fec6d, id 0
› Ethernet II, Src: 02:42:0a:09:00:05 (02:42:0a:09:00:05), Dst: 02:42:12:dd:f6:96 (02:42:12:dd:f6:96)
› Internet Protocol Version 4, Src: 10.9.0.5, Dst: 8.8.8.8
› Internet Control Message Protocol
```

```
0000  02 42 12 dd f6 96 02 42  0a 09 00 05 08 00 45 00   ·B····B ······E·
0010  00 54 9f a8 40 00 40 01  80 e3 0a 09 00 05 08 08   ·T··@·@· ········
0020  08 08 08 00 fb 8d 00 20  00 01 9f e5 9a 68 00 00   ········ ·····h··
0030  00 00 f9 2f 0a 00 00 00  00 00 10 11 12 13 14 15   ···/···· ········
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ········ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

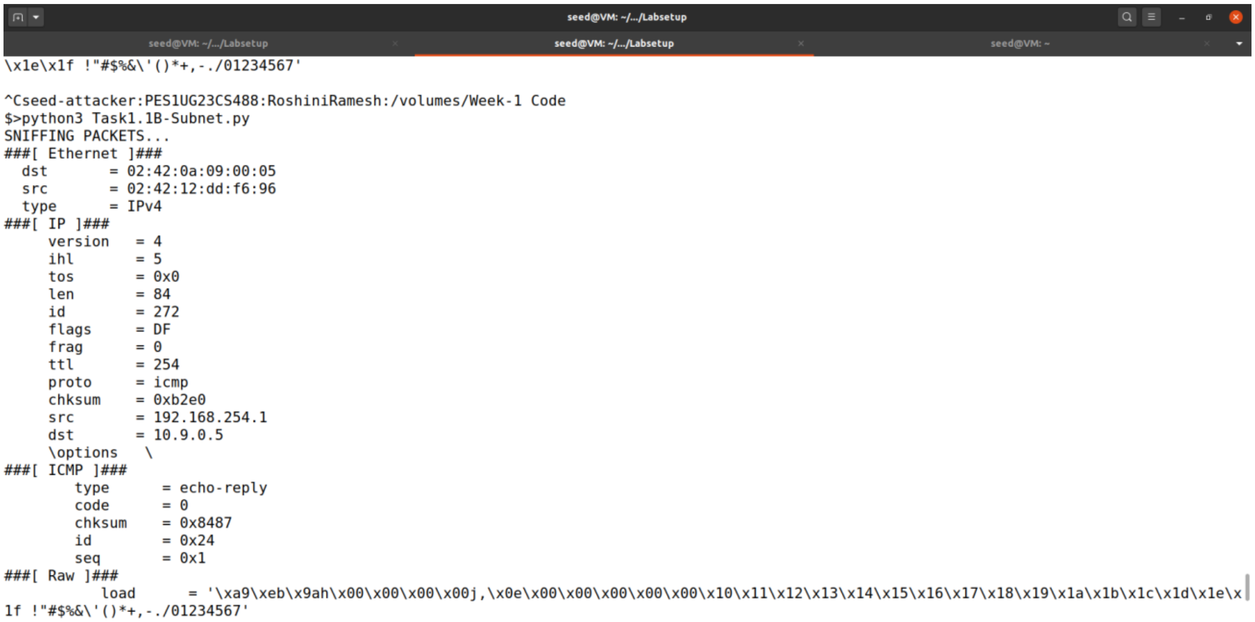
Task1.1B-ICMP.pcapng   Packets: 28 · Displayed: 28 (100.0%)   Profile: Default

Explanation:

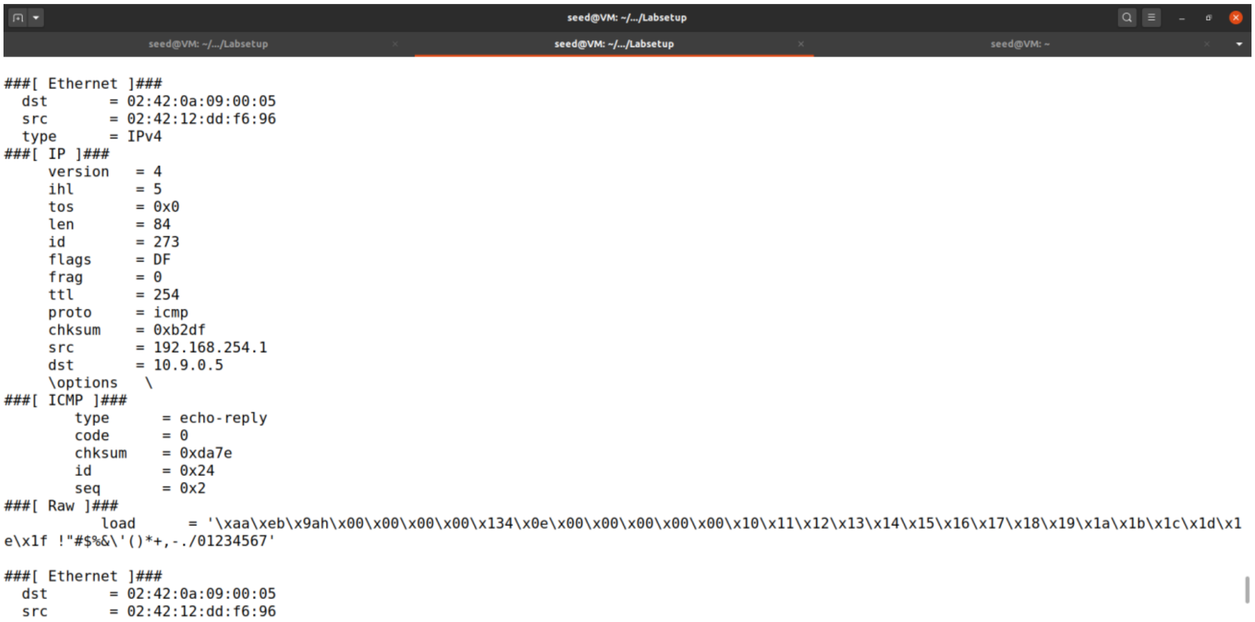
In the given code, the filter feature of sniff to capture only ICMP packets is used. Hence, both the terminal and Wireshark output display only sniffed ICMP echo request and reply packets which Host A sends while pinging 8.8.8.8.

TASK 1.1 B: APPLYING PACKET FILTERS TO CAPTURE SUBNET PACKETS

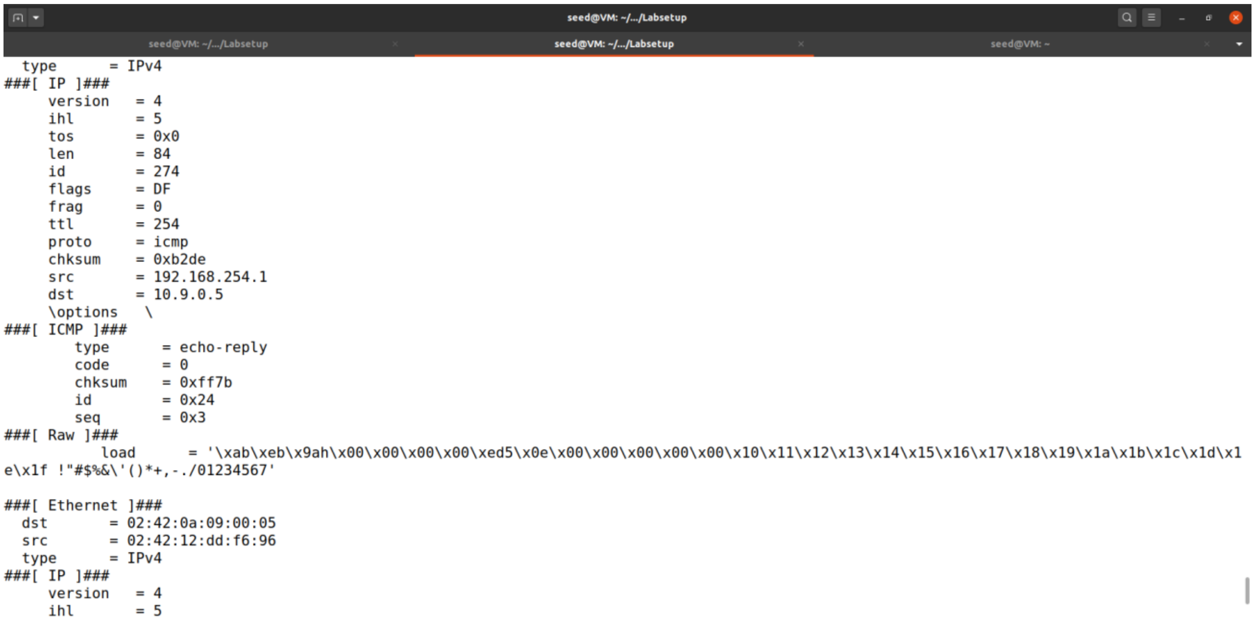
Terminal Output:



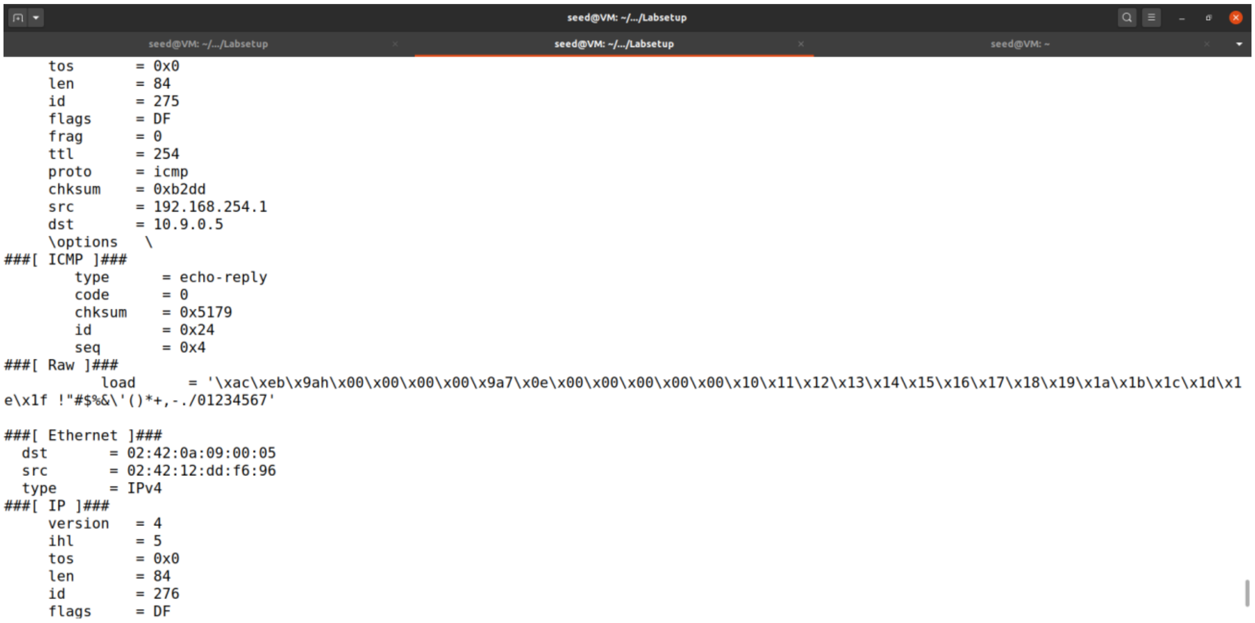
```
\x1e\x1f !"#$%&\'()*+,-./01234567'

^Cseed-attacker:PES1UG23CS488:RoshiniRamesh:/volumes/Week-1 Code
$>python3 Task1.1B-Subnet.py
SNIFFING PACKETS...
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:12:dd:f6:96
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 272
     flags     = DF
     frag      = 0
     ttl       = 254
     proto     = icmp
     chksum    = 0xb2e0
     src       = 192.168.254.1
     dst       = 10.9.0.5
     \options   \
###[ ICMP ]###
        type      = echo-reply
        code      = 0
        chksum    = 0x8487
        id        = 0x24
        seq       = 0x1
###[ Raw ]###
           load      = '\xa9\xeb\x9ah\x00\x00\x00\x00j,\x0e\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x
1f !"#$%&\'()*+,-./01234567'
```



```
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:12:dd:f6:96
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 273
     flags     = DF
     frag      = 0
     ttl       = 254
     proto     = icmp
     chksum    = 0xb2df
     src       = 192.168.254.1
     dst       = 10.9.0.5
     \options   \
###[ ICMP ]###
        type      = echo-reply
        code      = 0
        chksum    = 0xda7e
        id        = 0x24
        seq       = 0x2
###[ Raw ]###
           load      = '\xaa\xeb\x9ah\x00\x00\x00\x00\x134\x0e\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1
e\x1f !"#$%&\'()*+,-./01234567'

###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:12:dd:f6:96
```

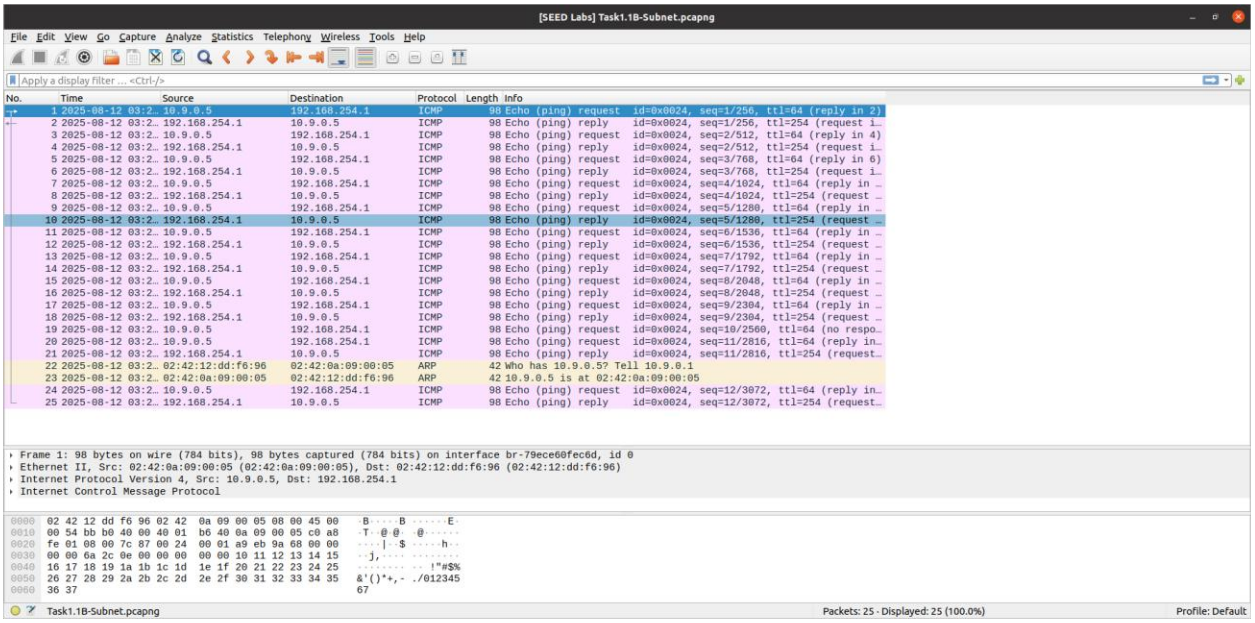


```
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 274
     flags     = DF
     frag      = 0
     ttl       = 254
     proto     = icmp
     chksum    = 0xb2de
     src       = 192.168.254.1
     dst       = 10.9.0.5
     \options   \
###[ ICMP ]###
        type      = echo-reply
        code      = 0
        chksum    = 0xff7b
        id        = 0x24
        seq       = 0x3
###[ Raw ]###
           load      = '\xab\xeb\x9ah\x00\x00\x00\x00\xed5\x0e\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1
e\x1f !"#$%&\'()*+,-./01234567'

###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:12:dd:f6:96
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
```



```
     tos       = 0x0
     len       = 84
     id        = 275
     flags     = DF
     frag      = 0
     ttl       = 254
     proto     = icmp
     chksum    = 0xb2dd
     src       = 192.168.254.1
     dst       = 10.9.0.5
     \options   \
###[ ICMP ]###
        type      = echo-reply
        code      = 0
        chksum    = 0x5179
        id        = 0x24
        seq       = 0x4
###[ Raw ]###
           load      = '\xac\xeb\x9ah\x00\x00\x00\x00\x9a7\x0e\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1
e\x1f !"#$%&\'()*+,-./01234567'

###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:12:dd:f6:96
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 276
     flags     = DF
```

Wireshark Output:


[SEED Labs] Task1.1B-Subnet.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>
Packets: 25 · Displayed: 25 (100.0%)
Profile: Default

```
No.  Time             Source             Destination        Protocol Length Info
   1 2025-08-12 03:2… 10.9.0.5           192.168.254.1      ICMP       98 Echo (ping) request  id=0x0024, seq=1/256, ttl=64 (reply in 2)
   2 2025-08-12 03:2… 192.168.254.1      10.9.0.5           ICMP       98 Echo (ping) reply    id=0x0024, seq=1/256, ttl=254 (request i…
   3 2025-08-12 03:2… 10.9.0.5           192.168.254.1      ICMP       98 Echo (ping) request  id=0x0024, seq=2/512, ttl=64 (reply in 4)
   4 2025-08-12 03:2… 192.168.254.1      10.9.0.5           ICMP       98 Echo (ping) reply    id=0x0024, seq=2/512, ttl=254 (request i…
   5 2025-08-12 03:2… 10.9.0.5           192.168.254.1      ICMP       98 Echo (ping) request  id=0x0024, seq=3/768, ttl=64 (reply in 6)
   6 2025-08-12 03:2… 192.168.254.1      10.9.0.5           ICMP       98 Echo (ping) reply    id=0x0024, seq=3/768, ttl=254 (request i…
   7 2025-08-12 03:2… 10.9.0.5           192.168.254.1      ICMP       98 Echo (ping) request  id=0x0024, seq=4/1024, ttl=64 (reply in …
   8 2025-08-12 03:2… 192.168.254.1      10.9.0.5           ICMP       98 Echo (ping) reply    id=0x0024, seq=4/1024, ttl=254 (request …
   9 2025-08-12 03:2… 10.9.0.5           192.168.254.1      ICMP       98 Echo (ping) request  id=0x0024, seq=5/1280, ttl=64 (reply in …
  10 2025-08-12 03:2… 192.168.254.1      10.9.0.5           ICMP       98 Echo (ping) reply    id=0x0024, seq=5/1280, ttl=254 (request …
  11 2025-08-12 03:2… 10.9.0.5           192.168.254.1      ICMP       98 Echo (ping) request  id=0x0024, seq=6/1536, ttl=64 (reply in …
  12 2025-08-12 03:2… 192.168.254.1      10.9.0.5           ICMP       98 Echo (ping) reply    id=0x0024, seq=6/1536, ttl=254 (request …
  13 2025-08-12 03:2… 10.9.0.5           192.168.254.1      ICMP       98 Echo (ping) request  id=0x0024, seq=7/1792, ttl=64 (reply in …
  14 2025-08-12 03:2… 192.168.254.1      10.9.0.5           ICMP       98 Echo (ping) reply    id=0x0024, seq=7/1792, ttl=254 (request …
  15 2025-08-12 03:2… 10.9.0.5           192.168.254.1      ICMP       98 Echo (ping) request  id=0x0024, seq=8/2048, ttl=64 (reply in …
  16 2025-08-12 03:2… 192.168.254.1      10.9.0.5           ICMP       98 Echo (ping) reply    id=0x0024, seq=8/2048, ttl=254 (request …
  17 2025-08-12 03:2… 10.9.0.5           192.168.254.1      ICMP       98 Echo (ping) request  id=0x0024, seq=9/2304, ttl=64 (reply in …
  18 2025-08-12 03:2… 192.168.254.1      10.9.0.5           ICMP       98 Echo (ping) reply    id=0x0024, seq=9/2304, ttl=254 (request …
  19 2025-08-12 03:2… 10.9.0.5           192.168.254.1      ICMP       98 Echo (ping) request  id=0x0024, seq=10/2560, ttl=64 (no respo…
  20 2025-08-12 03:2… 10.9.0.5           192.168.254.1      ICMP       98 Echo (ping) request  id=0x0024, seq=11/2816, ttl=64 (reply in…
  21 2025-08-12 03:2… 192.168.254.1      10.9.0.5           ICMP       98 Echo (ping) reply    id=0x0024, seq=11/2816, ttl=254 (request…
  22 2025-08-12 03:2… 02:42:12:dd:f6:96  02:42:0a:09:00:05  ARP        42 Who has 10.9.0.5? Tell 10.9.0.1
  23 2025-08-12 03:2… 02:42:0a:09:00:05  02:42:12:dd:f6:96  ARP        42 10.9.0.5 is at 02:42:0a:09:00:05
  24 2025-08-12 03:2… 10.9.0.5           192.168.254.1      ICMP       98 Echo (ping) request  id=0x0024, seq=12/3072, ttl=64 (reply in…
  25 2025-08-12 03:2… 192.168.254.1      10.9.0.5           ICMP       98 Echo (ping) reply    id=0x0024, seq=12/3072, ttl=254 (request…
```

```
› Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface br-79ece60fec6d, id 0
› Ethernet II, Src: 02:42:0a:09:00:05 (02:42:0a:09:00:05), Dst: 02:42:12:dd:f6:96 (02:42:12:dd:f6:96)
› Internet Protocol Version 4, Src: 10.9.0.5, Dst: 192.168.254.1
› Internet Control Message Protocol
```

```
0000  02 42 12 dd f6 96 02 42  0a 09 00 05 08 00 45 00   ·B·····B ······E·
0010  00 54 bb b0 40 00 40 01  b6 40 0a 09 00 05 c0 a8   ·T··@·@· ·@······
0020  fe 01 08 00 7c 87 00 24  00 01 a9 eb 9a 68 00 00   ····|··$ ·····h··
0030  00 00 6a 2c 0e 00 00 00  00 00 10 11 12 13 14 15   ··j,···· ········
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ········ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

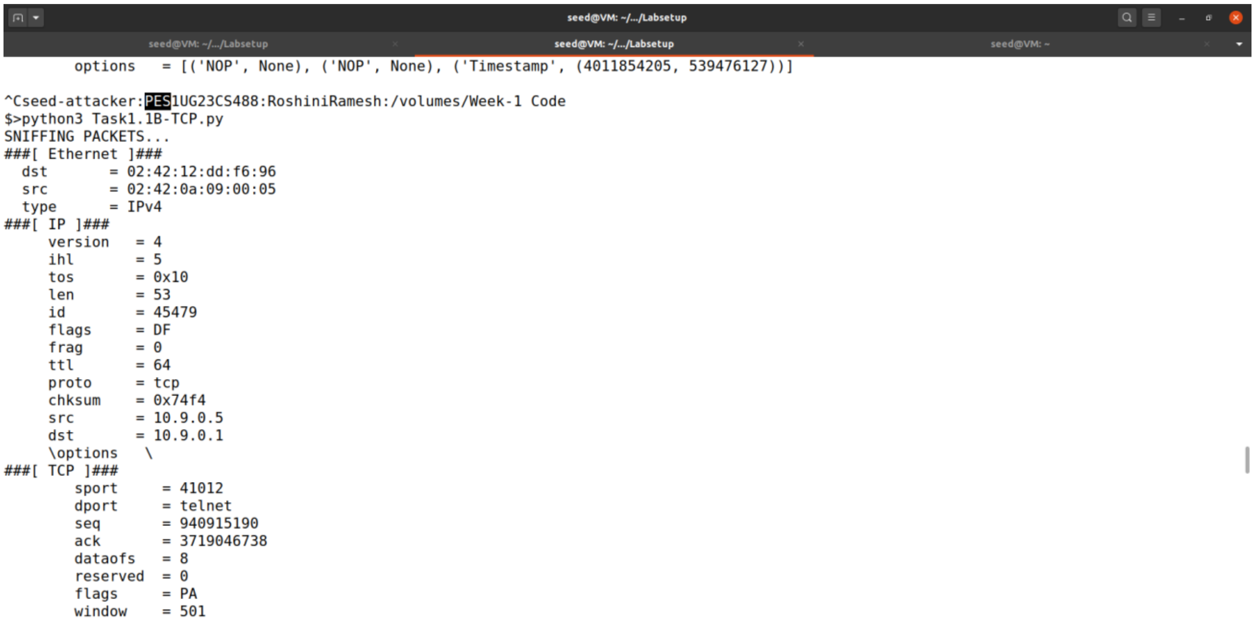
Task1.1B-Subnet.pcapng

Explanation:

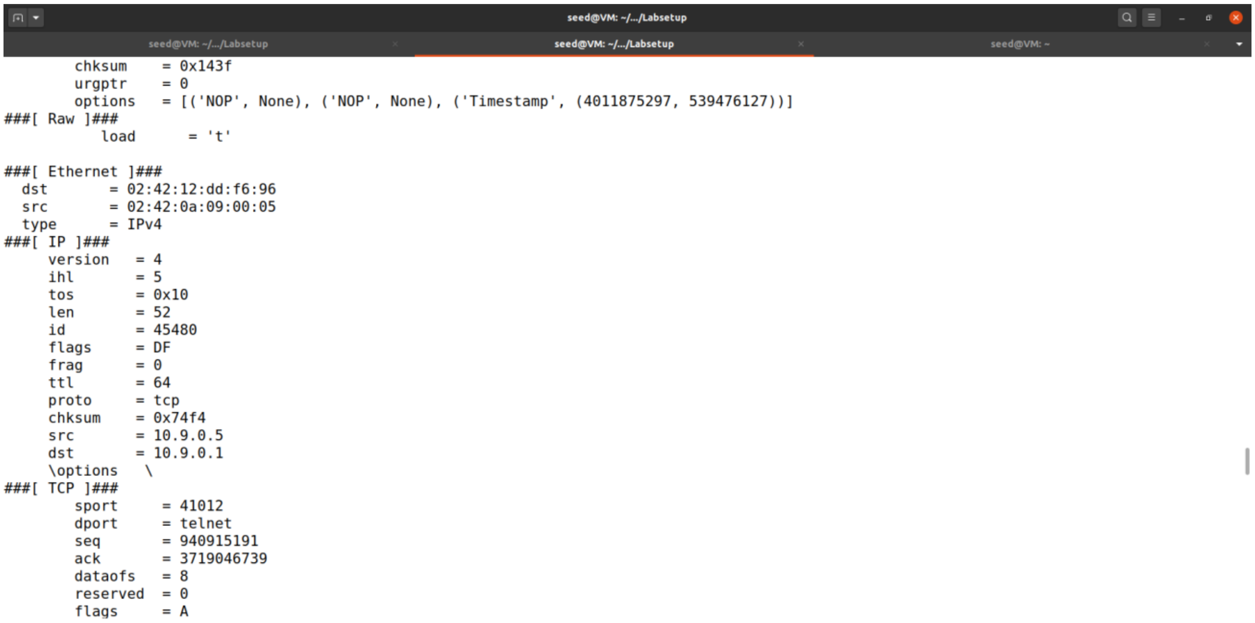
In the given code, the filter feature of sniff to capture only packets on a given subnet. Hence, both the terminal and Wireshark output display only sniffed ICMP echo request and reply packets which Host A sends while pinging 192.168.254.1.

TASK 1.1 B: APPLYING PACKET FILTERS TO CAPTURE TCP PACKETS USING TELNET

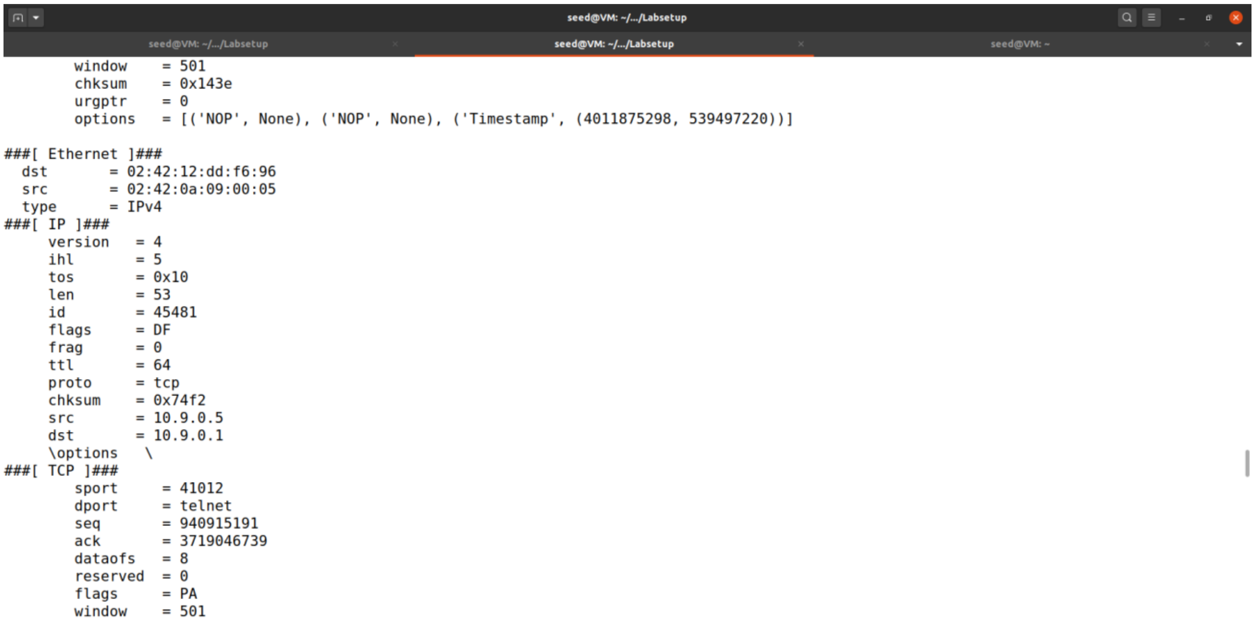
Terminal Output:



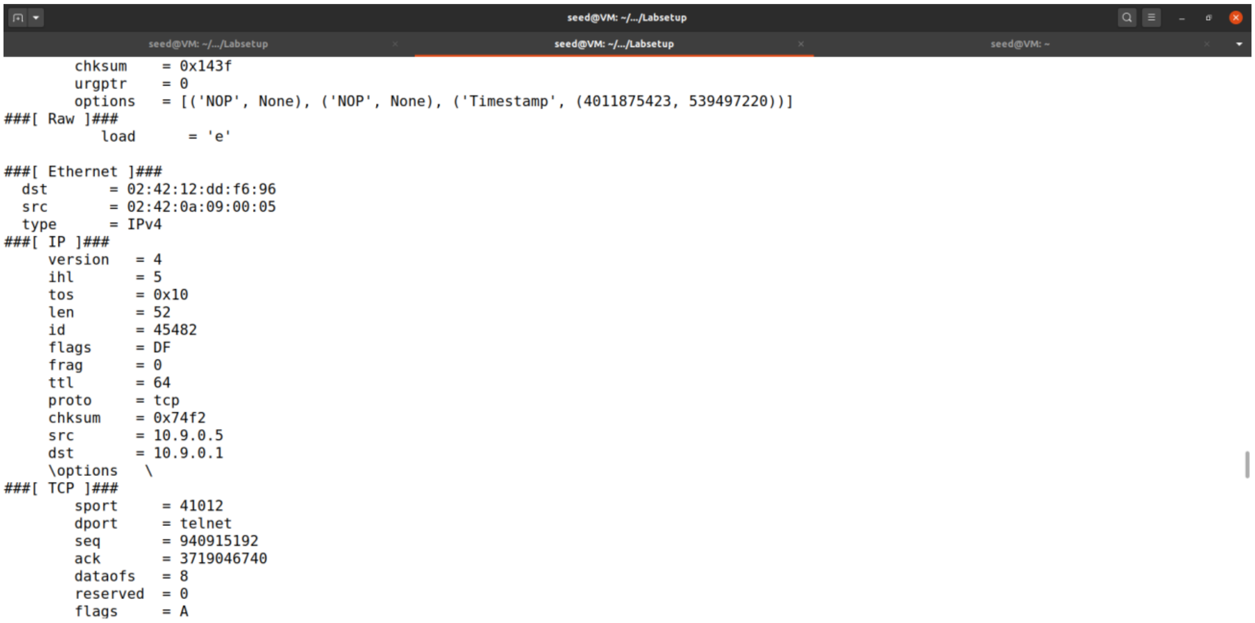
```
        options   = [('NOP', None), ('NOP', None), ('Timestamp', (4011854205, 539476127))]

^Cseed-attacker:PES1UG23CS488:RoshiniRamesh:/volumes/Week-1 Code
$>python3 Task1.1B-TCP.py
SNIFFING PACKETS...
###[ Ethernet ]###
  dst       = 02:42:12:dd:f6:96
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x10
     len       = 53
     id        = 45479
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = tcp
     chksum    = 0x74f4
     src       = 10.9.0.5
     dst       = 10.9.0.1
     \options   \
###[ TCP ]###
        sport     = 41012
        dport     = telnet
        seq       = 940915190
        ack       = 3719046738
        dataofs   = 8
        reserved  = 0
        flags     = PA
        window    = 501
```



```
        chksum    = 0x143f
        urgptr    = 0
        options   = [('NOP', None), ('NOP', None), ('Timestamp', (4011875297, 539476127))]
###[ Raw ]###
           load      = 't'

###[ Ethernet ]###
  dst       = 02:42:12:dd:f6:96
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x10
     len       = 52
     id        = 45480
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = tcp
     chksum    = 0x74f4
     src       = 10.9.0.5
     dst       = 10.9.0.1
     \options   \
###[ TCP ]###
        sport     = 41012
        dport     = telnet
        seq       = 940915191
        ack       = 3719046739
        dataofs   = 8
        reserved  = 0
        flags     = A
```

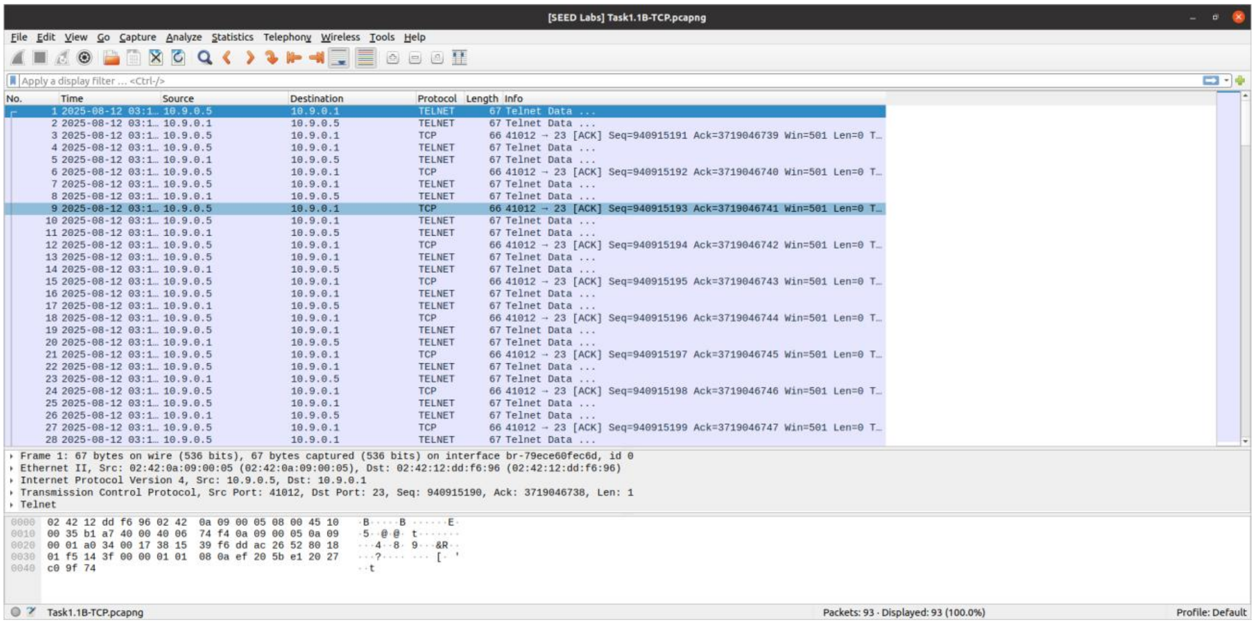


```
        window    = 501
        chksum    = 0x143e
        urgptr    = 0
        options   = [('NOP', None), ('NOP', None), ('Timestamp', (4011875298, 539497220))]

###[ Ethernet ]###
  dst       = 02:42:12:dd:f6:96
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x10
     len       = 53
     id        = 45481
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = tcp
     chksum    = 0x74f2
     src       = 10.9.0.5
     dst       = 10.9.0.1
     \options   \
###[ TCP ]###
        sport     = 41012
        dport     = telnet
        seq       = 940915191
        ack       = 3719046739
        dataofs   = 8
        reserved  = 0
        flags     = PA
        window    = 501
```



```
        chksum    = 0x143f
        urgptr    = 0
        options   = [('NOP', None), ('NOP', None), ('Timestamp', (4011875423, 539497220))]
###[ Raw ]###
           load      = 'e'

###[ Ethernet ]###
  dst       = 02:42:12:dd:f6:96
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x10
     len       = 52
     id        = 45482
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = tcp
     chksum    = 0x74f2
     src       = 10.9.0.5
     dst       = 10.9.0.1
     \options   \
###[ TCP ]###
        sport     = 41012
        dport     = telnet
        seq       = 940915192
        ack       = 3719046740
        dataofs   = 8
        reserved  = 0
        flags     = A
```

Wireshark Output:



[SEED Labs] Task1.1B-TCP.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TELNET	67	Telnet Data ...
2	2025-08-12 03:1…	10.9.0.1	10.9.0.5	TELNET	67	Telnet Data ...
3	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TCP	66	41012 → 23 [ACK] Seq=940915191 Ack=3719046739 Win=501 Len=0 T…
4	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TELNET	67	Telnet Data ...
5	2025-08-12 03:1…	10.9.0.1	10.9.0.5	TELNET	67	Telnet Data ...
6	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TCP	66	41012 → 23 [ACK] Seq=940915192 Ack=3719046740 Win=501 Len=0 T…
7	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TELNET	67	Telnet Data ...
8	2025-08-12 03:1…	10.9.0.1	10.9.0.5	TELNET	67	Telnet Data ...
9	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TCP	66	41012 → 23 [ACK] Seq=940915193 Ack=3719046741 Win=501 Len=0 T…
10	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TELNET	67	Telnet Data ...
11	2025-08-12 03:1…	10.9.0.1	10.9.0.5	TELNET	67	Telnet Data ...
12	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TCP	66	41012 → 23 [ACK] Seq=940915194 Ack=3719046742 Win=501 Len=0 T…
13	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TELNET	67	Telnet Data ...
14	2025-08-12 03:1…	10.9.0.1	10.9.0.5	TELNET	67	Telnet Data ...
15	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TCP	66	41012 → 23 [ACK] Seq=940915195 Ack=3719046743 Win=501 Len=0 T…
16	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TELNET	67	Telnet Data ...
17	2025-08-12 03:1…	10.9.0.1	10.9.0.5	TELNET	67	Telnet Data ...
18	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TCP	66	41012 → 23 [ACK] Seq=940915196 Ack=3719046744 Win=501 Len=0 T…
19	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TELNET	67	Telnet Data ...
20	2025-08-12 03:1…	10.9.0.1	10.9.0.5	TELNET	67	Telnet Data ...
21	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TCP	66	41012 → 23 [ACK] Seq=940915197 Ack=3719046745 Win=501 Len=0 T…
22	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TELNET	67	Telnet Data ...
23	2025-08-12 03:1…	10.9.0.1	10.9.0.5	TELNET	67	Telnet Data ...
24	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TCP	66	41012 → 23 [ACK] Seq=940915198 Ack=3719046746 Win=501 Len=0 T…
25	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TELNET	67	Telnet Data ...
26	2025-08-12 03:1…	10.9.0.1	10.9.0.5	TELNET	67	Telnet Data ...
27	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TCP	66	41012 → 23 [ACK] Seq=940915199 Ack=3719046747 Win=501 Len=0 T…
28	2025-08-12 03:1…	10.9.0.5	10.9.0.1	TELNET	67	Telnet Data ...

```
› Frame 1: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface br-79ece60fec6d, id 0
› Ethernet II, Src: 02:42:0a:09:00:05 (02:42:0a:09:00:05), Dst: 02:42:12:dd:f6:96 (02:42:12:dd:f6:96)
› Internet Protocol Version 4, Src: 10.9.0.5, Dst: 10.9.0.1
› Transmission Control Protocol, Src Port: 41012, Dst Port: 23, Seq: 940915190, Ack: 3719046738, Len: 1
› Telnet
```

```
0000  02 42 12 dd f6 96 02 42  0a 09 00 05 08 00 45 10   ·B·····B ······E·
0010  00 35 b1 a7 40 00 40 06  74 f4 0a 09 00 05 0a 09   ·5··@·@· t·······
0020  00 01 a0 34 00 17 38 15  39 f6 dd ac 26 52 80 18   ···4··8· 9···&R··
0030  01 f5 14 3f 00 00 01 01  08 0a ef 20 5b e1 20 27   ···?···· ··· [· '
0040  c0 9f 74                                           ··t
```

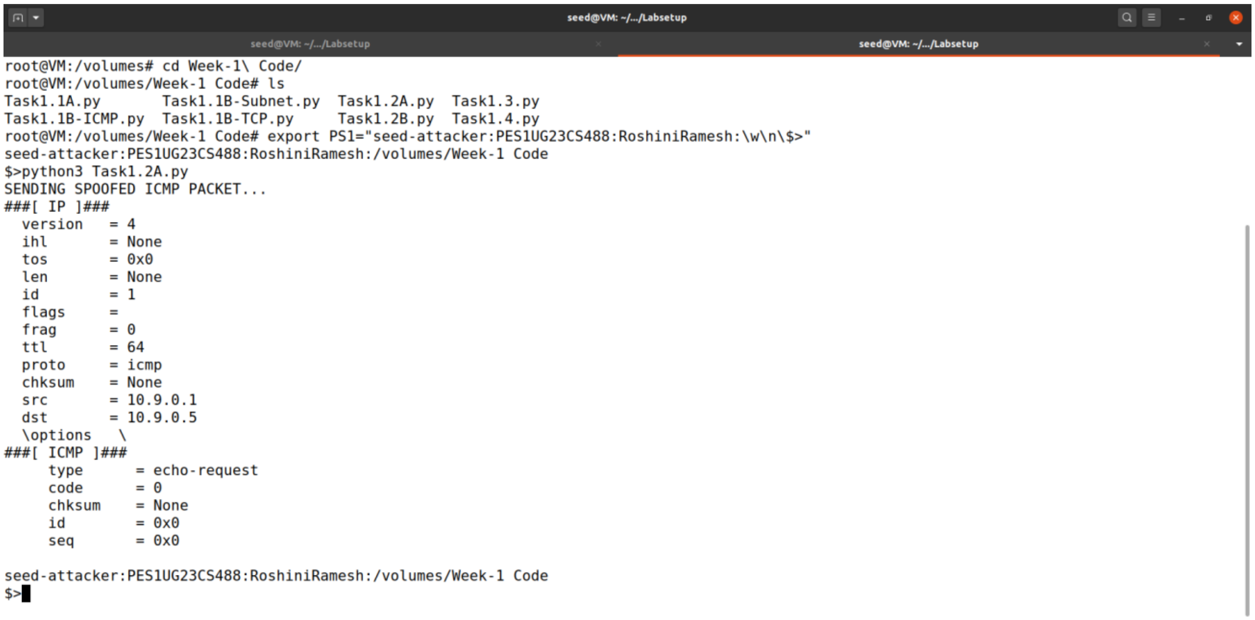
Task1.1B-TCP.pcapng   Packets: 93 · Displayed: 93 (100.0%)   Profile: Default

Explanation:

In the given code, the filter feature of sniff is used to capture only TCP packets. This is done while host A sets up a Telnet connection to 8.8.8.8. Telnet works on port 23.

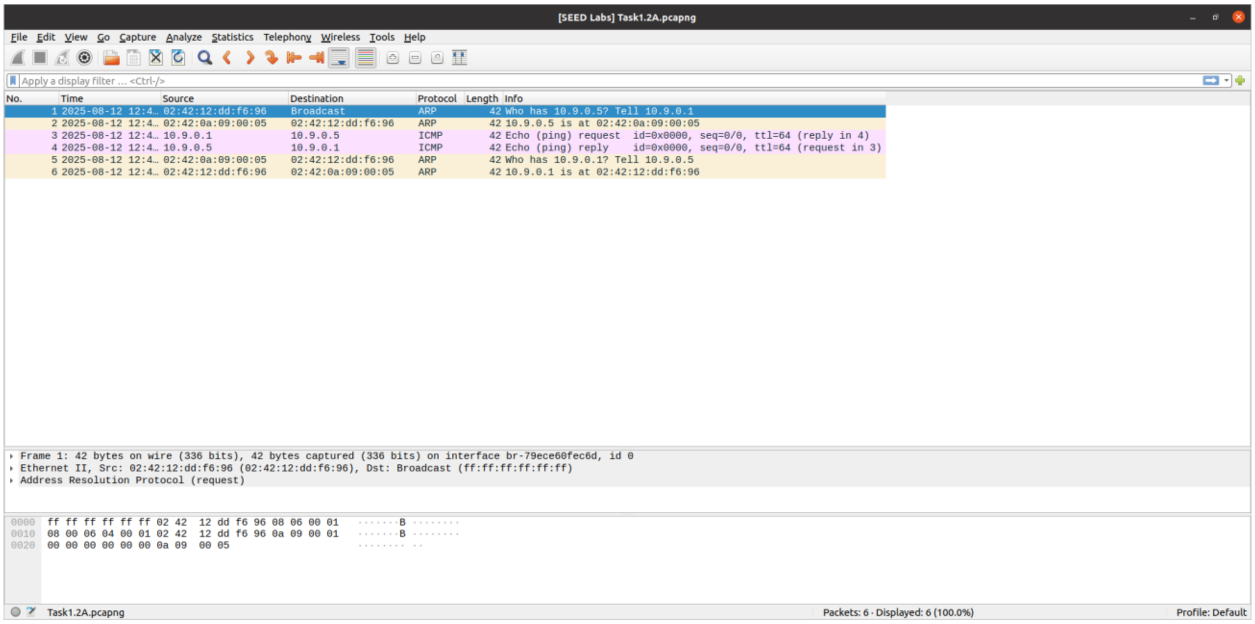
TASK 1.2A: PACKET SPOOFING

Terminal Output:



```
root@VM:/volumes# cd Week-1\ Code/
root@VM:/volumes/Week-1 Code# ls
Task1.1A.py       Task1.1B-Subnet.py  Task1.2A.py  Task1.3.py
Task1.1B-ICMP.py  Task1.1B-TCP.py     Task1.2B.py  Task1.4.py
root@VM:/volumes/Week-1 Code# export PS1="seed-attacker:PES1UG23CS488:RoshiniRamesh:\w\n\$>"
seed-attacker:PES1UG23CS488:RoshiniRamesh:/volumes/Week-1 Code
$>python3 Task1.2A.py
SENDING SPOOFED ICMP PACKET...
###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = icmp
  chksum    = None
  src       = 10.9.0.1
  dst       = 10.9.0.5
  \options   \
###[ ICMP ]###
     type      = echo-request
     code      = 0
     chksum    = None
     id        = 0x0
     seq       = 0x0

seed-attacker:PES1UG23CS488:RoshiniRamesh:/volumes/Week-1 Code
$>
```

Wireshark:



[SEED Labs] Task1.2A.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.  Time            Source             Destination        Protocol Length Info
   1 2025-08-12 12:4… 02:42:12:dd:f6:96  Broadcast          ARP        42 Who has 10.9.0.5? Tell 10.9.0.1
   2 2025-08-12 12:4… 02:42:0a:09:00:05  02:42:12:dd:f6:96  ARP        42 10.9.0.5 is at 02:42:0a:09:00:05
   3 2025-08-12 12:4… 10.9.0.1           10.9.0.5           ICMP       42 Echo (ping) request  id=0x0000, seq=0/0, ttl=64 (reply in 4)
   4 2025-08-12 12:4… 10.9.0.5           10.9.0.1           ICMP       42 Echo (ping) reply    id=0x0000, seq=0/0, ttl=64 (request in 3)
   5 2025-08-12 12:4… 02:42:0a:09:00:05  02:42:12:dd:f6:96  ARP        42 Who has 10.9.0.1? Tell 10.9.0.5
   6 2025-08-12 12:4… 02:42:12:dd:f6:96  02:42:0a:09:00:05  ARP        42 10.9.0.1 is at 02:42:12:dd:f6:96
```

```
› Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface br-79ece60fec6d, id 0
› Ethernet II, Src: 02:42:12:dd:f6:96 (02:42:12:dd:f6:96), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
› Address Resolution Protocol (request)
```

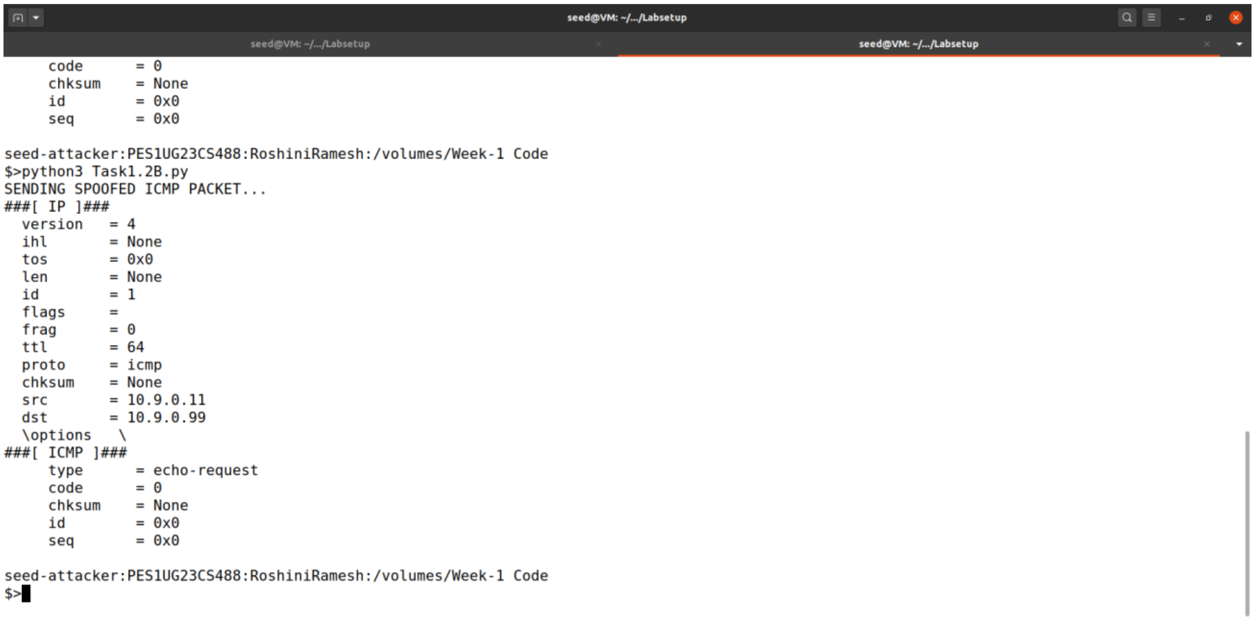
```
0000  ff ff ff ff ff ff 02 42  12 dd f6 96 08 06 00 01   ·······B ········
0010  08 00 06 04 00 01 02 42  12 dd f6 96 0a 09 00 01   ·······B ········
0020  00 00 00 00 00 00 0a 09  00 05                     ········ ··
```

Explanation:

In the given code, a spoofed ICMP echo request packet is sent using an IP on the network. The packet is being sent from 10.9.0.1 to 10.9.0.5. An ICMP echo reply is received, showing that spoofing has been successful.

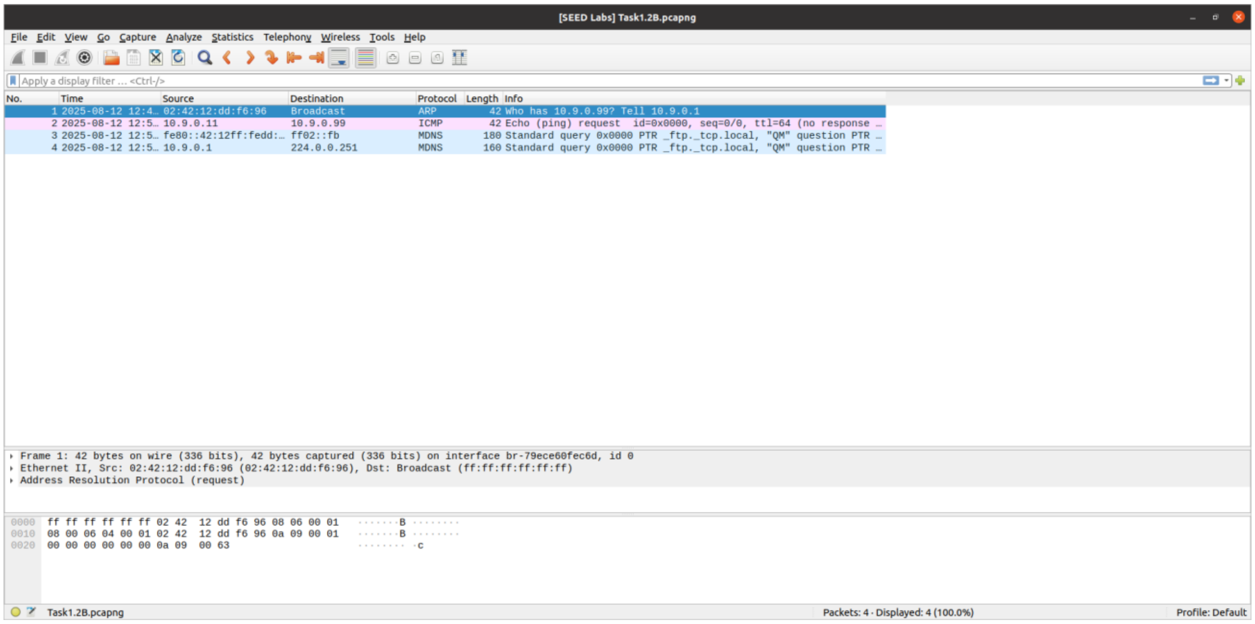
TASK 1.2B: PACKET SPOOFING

Terminal Output:



```
     code      = 0
     chksum    = None
     id        = 0x0
     seq       = 0x0

seed-attacker:PES1UG23CS488:RoshiniRamesh:/volumes/Week-1 Code
$>python3 Task1.2B.py
SENDING SPOOFED ICMP PACKET...
###[ IP ]###
  version   = 4
  ihl       = None
  tos       = 0x0
  len       = None
  id        = 1
  flags     =
  frag      = 0
  ttl       = 64
  proto     = icmp
  chksum    = None
  src       = 10.9.0.11
  dst       = 10.9.0.99
  \options   \
###[ ICMP ]###
     type      = echo-request
     code      = 0
     chksum    = None
     id        = 0x0
     seq       = 0x0

seed-attacker:PES1UG23CS488:RoshiniRamesh:/volumes/Week-1 Code
$>
```

Wireshark:



[SEED Labs] Task1.2B.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	2025-08-12 12:4…	02:42:12:dd:f6:96	Broadcast	ARP	42	Who has 10.9.0.99? Tell 10.9.0.1
2	2025-08-12 12:5…	10.9.0.11	10.9.0.99	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response …
3	2025-08-12 12:5…	fe80::42:12ff:fedd:…	ff02::fb	MDNS	180	Standard query 0x0000 PTR _ftp._tcp.local, "QM" question PTR …
4	2025-08-12 12:5…	10.9.0.1	224.0.0.251	MDNS	160	Standard query 0x0000 PTR _ftp._tcp.local, "QM" question PTR …

```
Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface br-79ece60fec6d, id 0
Ethernet II, Src: 02:42:12:dd:f6:96 (02:42:12:dd:f6:96), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Address Resolution Protocol (request)
```

```
0000  ff ff ff ff ff ff 02 42  12 dd f6 96 08 06 00 01   ·······B ········
0010  08 00 06 04 00 01 02 42  12 dd f6 96 0a 09 00 01   ·······B ········
0020  00 00 00 00 00 00 0a 09  00 63                     ········ ·c
```

Explanation:

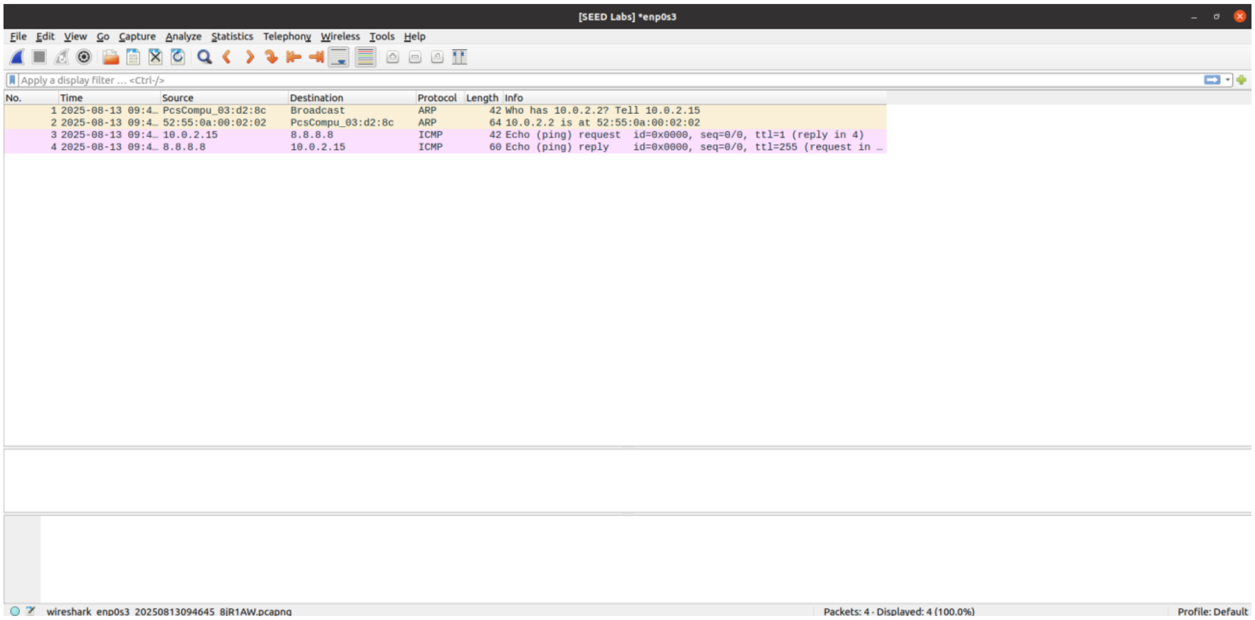
In the given code, a spoofed ICMP echo request packet is sent using an arbitrary IP (which is non-existent on the network). The packet is being sent from 10.9.0.11 to 10.9.0.99. An ICMP echo reply is received, showing that spoofing has been successful.

TASK 1.3: TRACEROUTE

Terminal Output:

```
seed-attacker:PES1UG23CS488:RoshiniRamesh:/volumes/Week-1 Code
$>python3 Task1.3.py 8.8.8.8
Traceroute 8.8.8.8
1 hops away:  8.8.8.8
Done 8.8.8.8
```

Wireshark:


[SEED Labs] *enp0s3
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	2025-08-13 09:4…	PcsCompu_03:d2:8c	Broadcast	ARP	42	Who has 10.0.2.2? Tell 10.0.2.15
2	2025-08-13 09:4…	52:55:0a:00:02:02	PcsCompu_03:d2:8c	ARP	64	10.0.2.2 is at 52:55:0a:00:02:02
3	2025-08-13 09:4…	10.0.2.15	8.8.8.8	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=1 (reply in 4)
4	2025-08-13 09:4…	8.8.8.8	10.0.2.15	ICMP	60	Echo (ping) reply id=0x0000, seq=0/0, ttl=255 (request in …

wireshark_enp0s3_20250813094645_8jR1AW.pcapng

Packets: 4 · Displayed: 4 (100.0%)

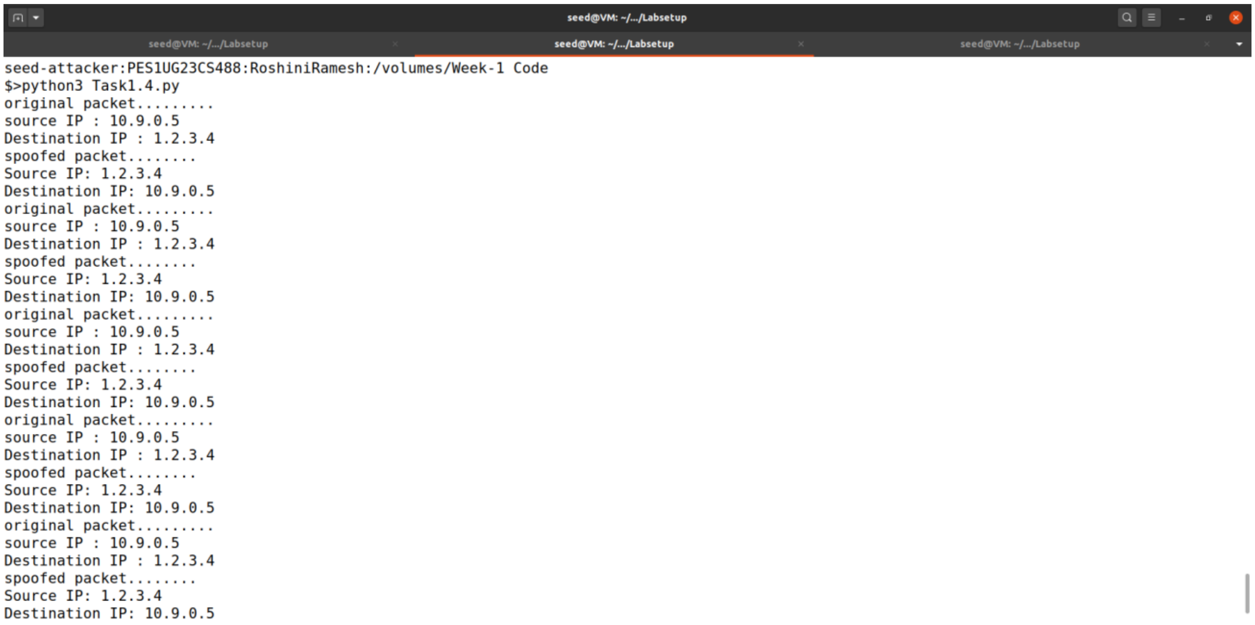
Profile: Default

Explanation:

In the given code, a simple traceroute program is implemented. The program sends ICMP Echo Requests with increasing TTL values. When ICMP Echo Request reaches the destination, destination sends ICMP Echo Reply. Here, Echo Request has been sent to 8.8.8.8 and after one hop, it responds with an Echo Reply as seen in the Wireshark output.

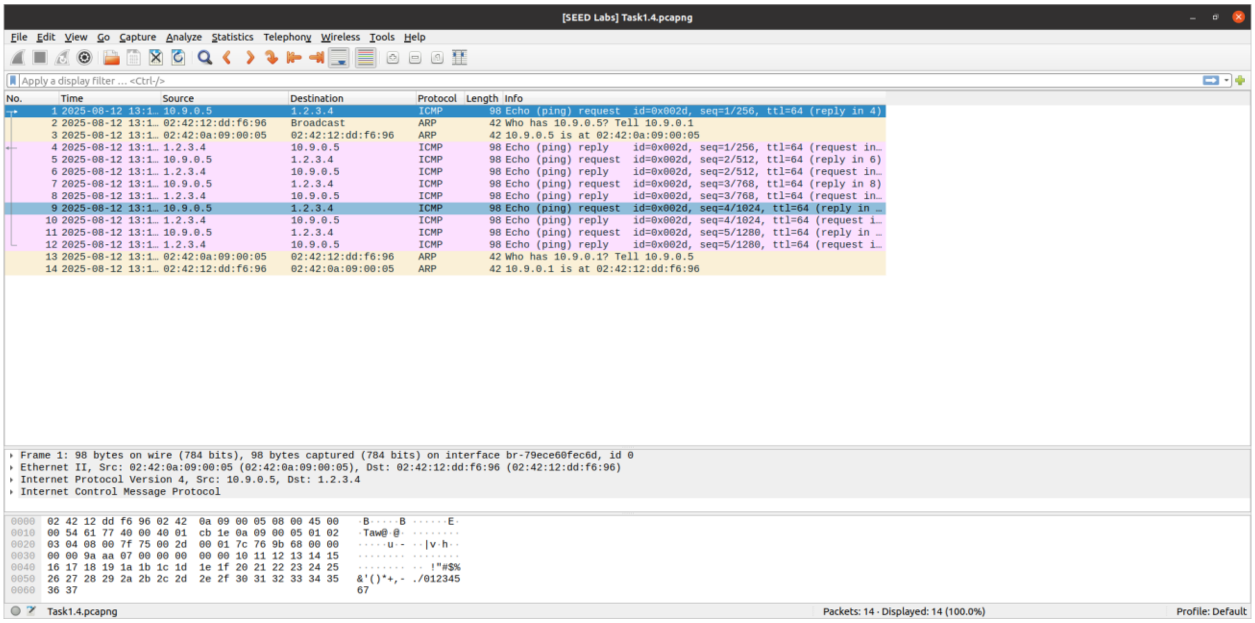
TASK 1.4: SNIFFING AND-THEN SPOOFING

Terminal Output:



```
seed-attacker:PES1UG23CS488:RoshiniRamesh:/volumes/Week-1 Code
$>python3 Task1.4.py
original packet.........
source IP : 10.9.0.5
Destination IP : 1.2.3.4
spoofed packet........
Source IP: 1.2.3.4
Destination IP: 10.9.0.5
original packet.........
source IP : 10.9.0.5
Destination IP : 1.2.3.4
spoofed packet........
Source IP: 1.2.3.4
Destination IP: 10.9.0.5
original packet.........
source IP : 10.9.0.5
Destination IP : 1.2.3.4
spoofed packet........
Source IP: 1.2.3.4
Destination IP: 10.9.0.5
original packet.........
source IP : 10.9.0.5
Destination IP : 1.2.3.4
spoofed packet........
Source IP: 1.2.3.4
Destination IP: 10.9.0.5
original packet.........
source IP : 10.9.0.5
Destination IP : 1.2.3.4
spoofed packet........
Source IP: 1.2.3.4
Destination IP: 10.9.0.5
```

Wireshark:


[SEED Labs] Task1.4.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	2025-08-12 13:1…	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x002d, seq=1/256, ttl=64 (reply in 4)
2	2025-08-12 13:1…	02:42:12:dd:f6:96	Broadcast	ARP	42	Who has 10.9.0.5? Tell 10.9.0.1
3	2025-08-12 13:1…	02:42:0a:09:00:05	02:42:12:dd:f6:96	ARP	42	10.9.0.5 is at 02:42:0a:09:00:05
4	2025-08-12 13:1…	1.2.3.4	10.9.0.5	ICMP	98	Echo (ping) reply id=0x002d, seq=1/256, ttl=64 (request in…
5	2025-08-12 13:1…	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x002d, seq=2/512, ttl=64 (reply in 6)
6	2025-08-12 13:1…	1.2.3.4	10.9.0.5	ICMP	98	Echo (ping) reply id=0x002d, seq=2/512, ttl=64 (request in…
7	2025-08-12 13:1…	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x002d, seq=3/768, ttl=64 (reply in 8)
8	2025-08-12 13:1…	1.2.3.4	10.9.0.5	ICMP	98	Echo (ping) reply id=0x002d, seq=3/768, ttl=64 (request in…
9	2025-08-12 13:1…	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x002d, seq=4/1024, ttl=64 (reply in …
10	2025-08-12 13:1…	1.2.3.4	10.9.0.5	ICMP	98	Echo (ping) reply id=0x002d, seq=4/1024, ttl=64 (request i…
11	2025-08-12 13:1…	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x002d, seq=5/1280, ttl=64 (reply in …
12	2025-08-12 13:1…	1.2.3.4	10.9.0.5	ICMP	98	Echo (ping) reply id=0x002d, seq=5/1280, ttl=64 (request i…
13	2025-08-12 13:1…	02:42:0a:09:00:05	02:42:12:dd:f6:96	ARP	42	Who has 10.9.0.1? Tell 10.9.0.5
14	2025-08-12 13:1…	02:42:12:dd:f6:96	02:42:0a:09:00:05	ARP	42	10.9.0.1 is at 02:42:12:dd:f6:96

```
Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface br-79ece60fec6d, id 0
Ethernet II, Src: 02:42:0a:09:00:05 (02:42:0a:09:00:05), Dst: 02:42:12:dd:f6:96 (02:42:12:dd:f6:96)
Internet Protocol Version 4, Src: 10.9.0.5, Dst: 1.2.3.4
Internet Control Message Protocol
```

```
0000  02 42 12 dd f6 96 02 42  0a 09 00 05 08 00 45 00   ·B·····B ······E·
0010  00 54 61 77 40 00 40 01  cb 1e 0a 09 00 05 01 02   ·Taw@·@· ········
0020  03 04 08 00 7f 75 00 2d  00 01 7c 76 9b 68 00 00   ·····u·- ··|v·h··
0030  00 00 9a aa 07 00 00 00  00 00 10 11 12 13 14 15   ········ ········
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ········ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

Explanation:

This combines all the concepts done so far. Here, Host A pings 1.2.3.4. This is sniffed by the attacker, who then sends a spoofed packet back to Host A. This is how spoofing attacks take place.