Ethical Hacking

Sniffing & Spoofing Lab

Niccolò Borgioli
Enkeleda Bardhi

Università degli Studi di Padova

Dipartimento Matematica

Contents

I. Lab Setup

- Creating the Network
- Network Details

II. Task 1.1

- Code
- Question A
- Question B

III. Task 1.2

- Code
- Execution
- Wireshark

IV. Task 1.3

- Code
- Execution

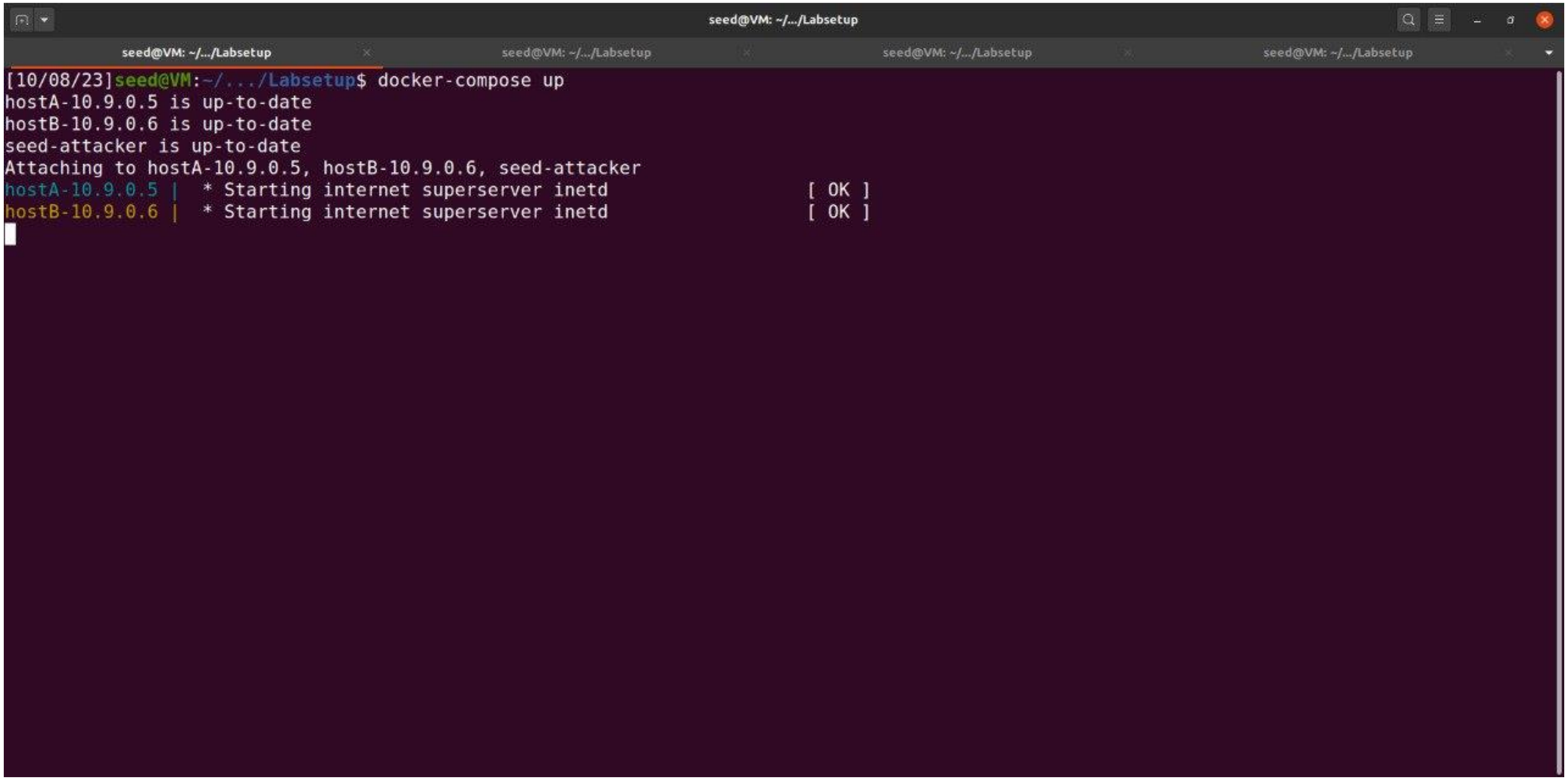
V. Task 1.4

- Code
- Host 1.2.3.4
- Host 10.9.0.99
- Host 8.8.8.8

Lab Setup

Creating the Network

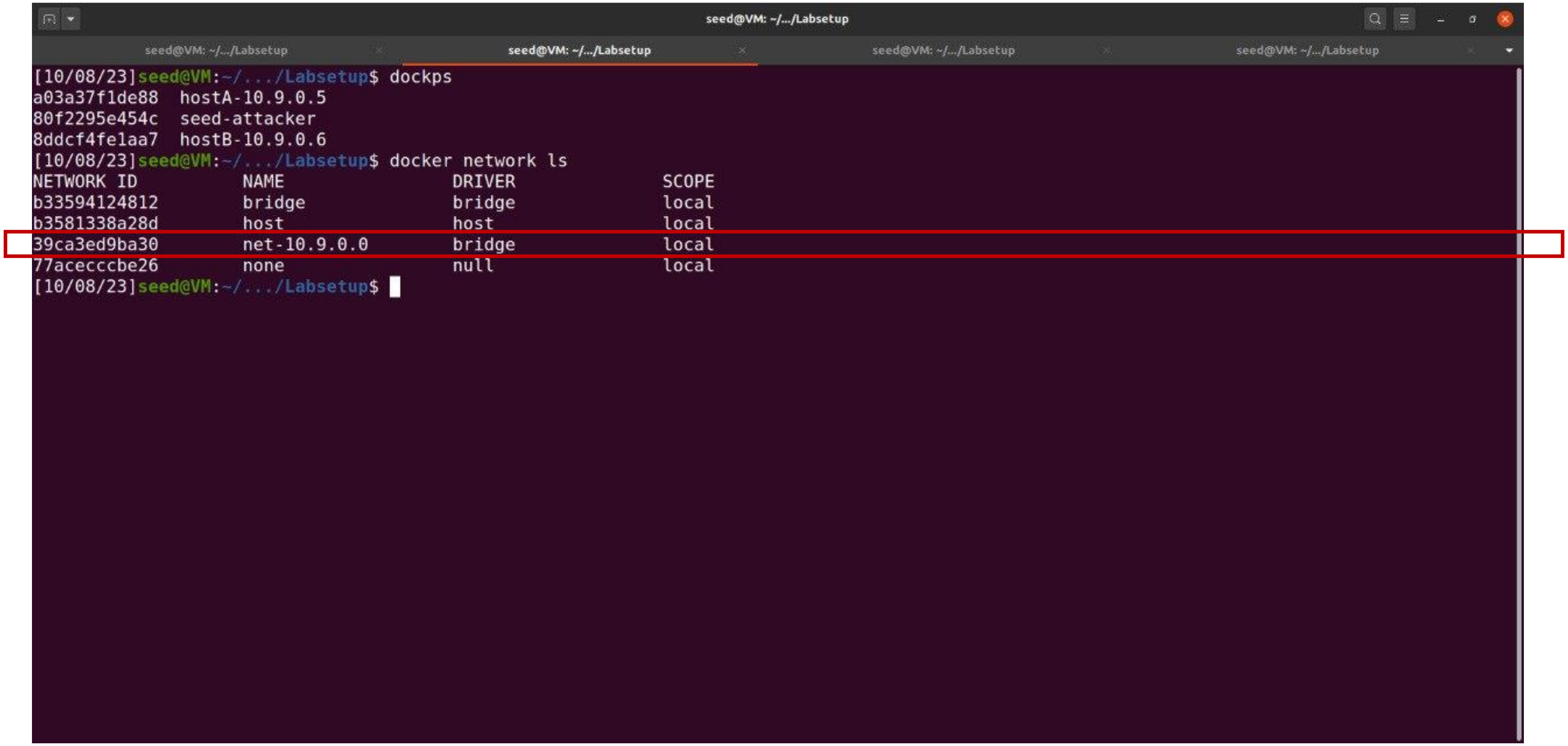
- **$ docker-compose build** to build the containers
- **$ docker-compose up** to start the containers



```
[10/08/23]seed@VM:~/.../Labsetup$ docker-compose up
hostA-10.9.0.5 is up-to-date
hostB-10.9.0.6 is up-to-date
seed-attacker is up-to-date
Attaching to hostA-10.9.0.5, hostB-10.9.0.6, seed-attacker
hostA-10.9.0.5 |  * Starting internet superserver inetd                  [ OK ]
hostB-10.9.0.6 |  * Starting internet superserver inetd                  [ OK ]
```

Network Details

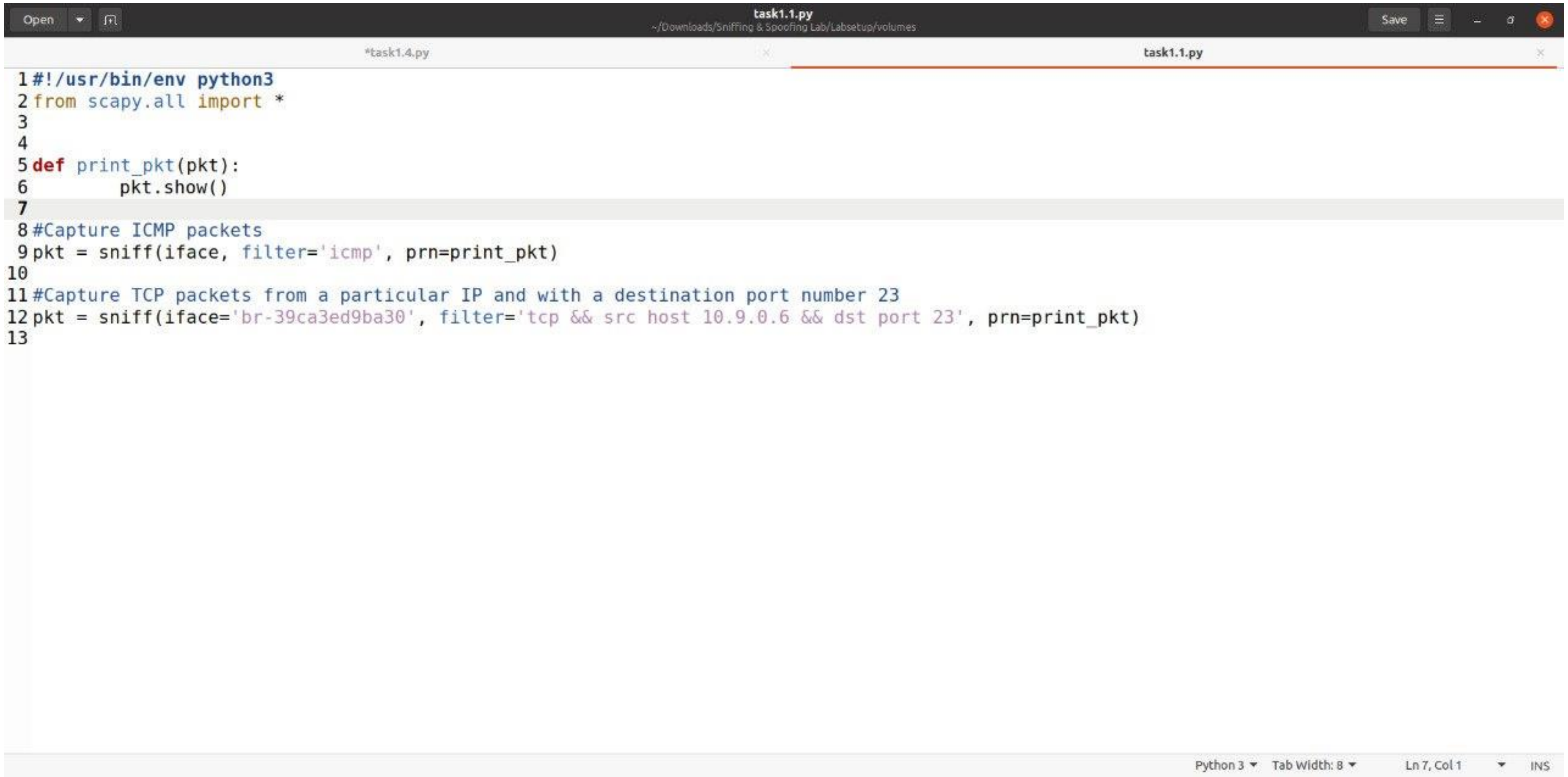
- **$ dockps** to find out the the IDs of the containers
- **$ docker network ls** to find out the network IDs



```
[10/08/23]seed@VM:~/.../Labsetup$ dockps
a03a37f1de88  hostA-10.9.0.5
80f2295e454c  seed-attacker
8ddcf4fe1aa7  hostB-10.9.0.6
[10/08/23]seed@VM:~/.../Labsetup$ docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
b33594124812        bridge              bridge              local
b3581338a28d        host                host                local
39ca3ed9ba30        net-10.9.0.0        bridge              local
77acecccbe26        none                null                local
[10/08/23]seed@VM:~/.../Labsetup$ 
```

Task 1.1

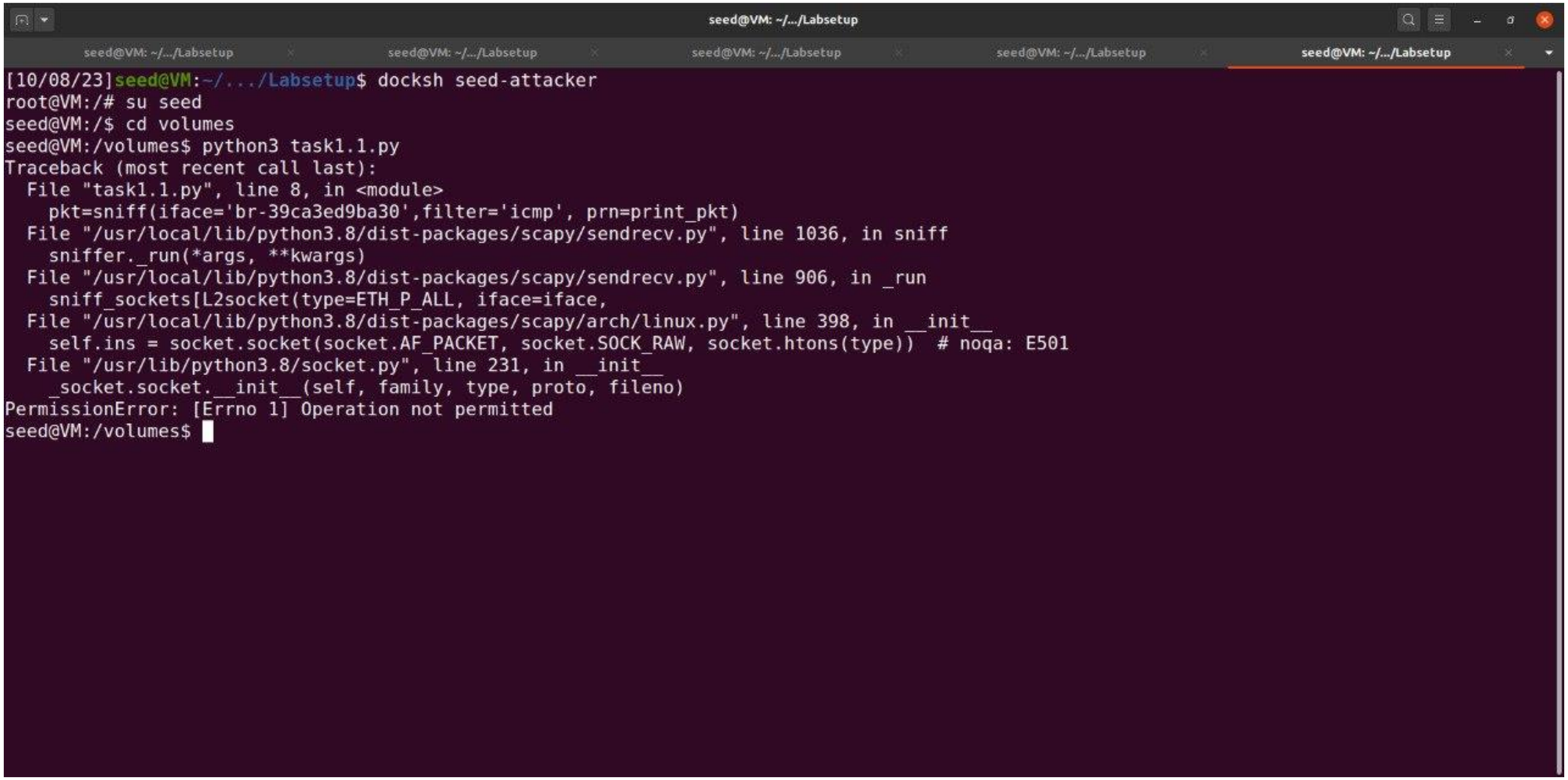
Code



```
#!/usr/bin/env python3
from scapy.all import *


def print_pkt(pkt):
        pkt.show()

#Capture ICMP packets
pkt = sniff(iface, filter='icmp', prn=print_pkt)

#Capture TCP packets from a particular IP and with a destination port number 23
pkt = sniff(iface='br-39ca3ed9ba30', filter='tcp && src host 10.9.0.6 && dst port 23', prn=print_pkt)

```

Question A

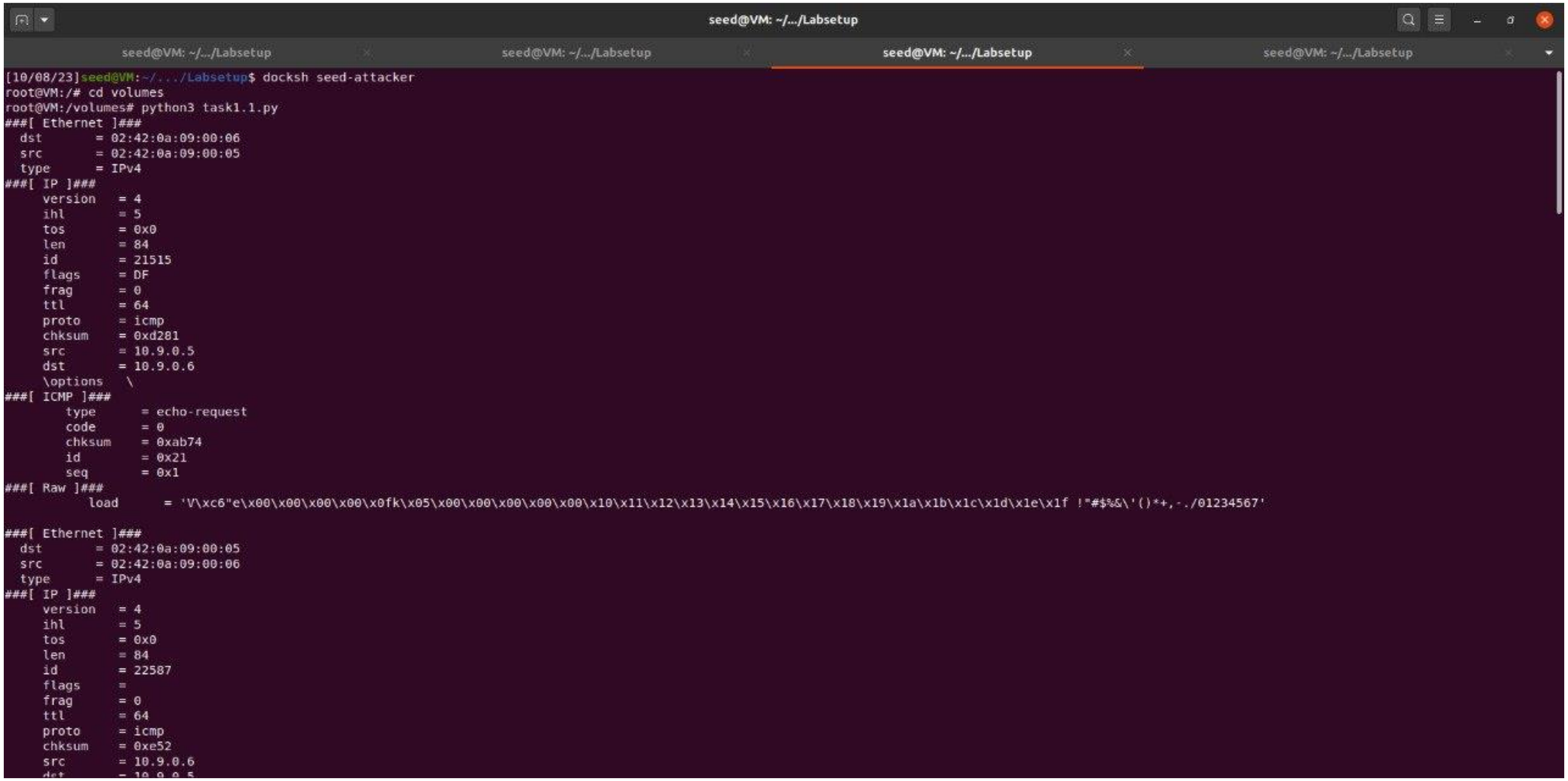
- Python script is attempting to use a raw socket to capture network packets
- Script typically needs to run with root privileges



```
[10/08/23]seed@VM:~/.../Labsetup$ docksh seed-attacker
root@VM:/# su seed
seed@VM:/$ cd volumes
seed@VM:/volumes$ python3 task1.1.py
Traceback (most recent call last):
  File "task1.1.py", line 8, in <module>
    pkt=sniff(iface='br-39ca3ed9ba30',filter='icmp', 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$ 
```

Question B (1)

- Capture only the ICMP packets

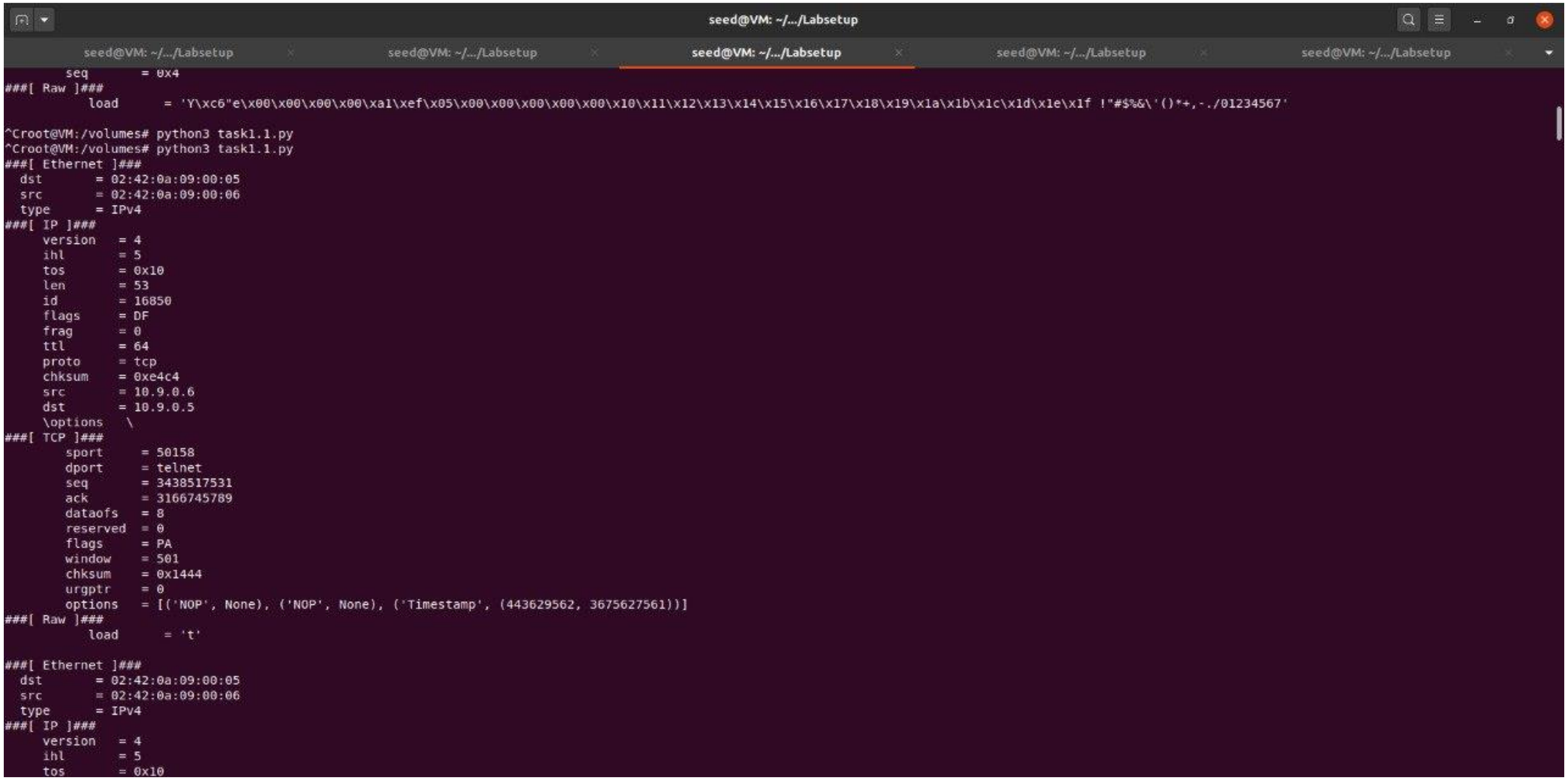


```
[10/08/23]seed@VM:~/.../Labsetup$ docksh seed-attacker
root@VM:/# cd volumes
root@VM:/volumes# python3 task1.1.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:06
  src       = 02:42:0a:09:00:05
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 21515
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0xd281
     src       = 10.9.0.5
     dst       = 10.9.0.6
     \options   \
###[ ICMP ]###
        type      = echo-request
        code      = 0
        chksum    = 0xab74
        id        = 0x21
        seq       = 0x1
###[ Raw ]###
           load      = 'V\xc6"e\x00\x00\x00\x00\x0fk\x05\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       = 02:42:0a:09:00:06
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 84
     id        = 22587
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0xe52
     src       = 10.9.0.6
     dst       = 10.9.0.5
```

* In *HostA-10.9.0.5* I ran the command **$ ping 10.9.0.6**

Question B (2)

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



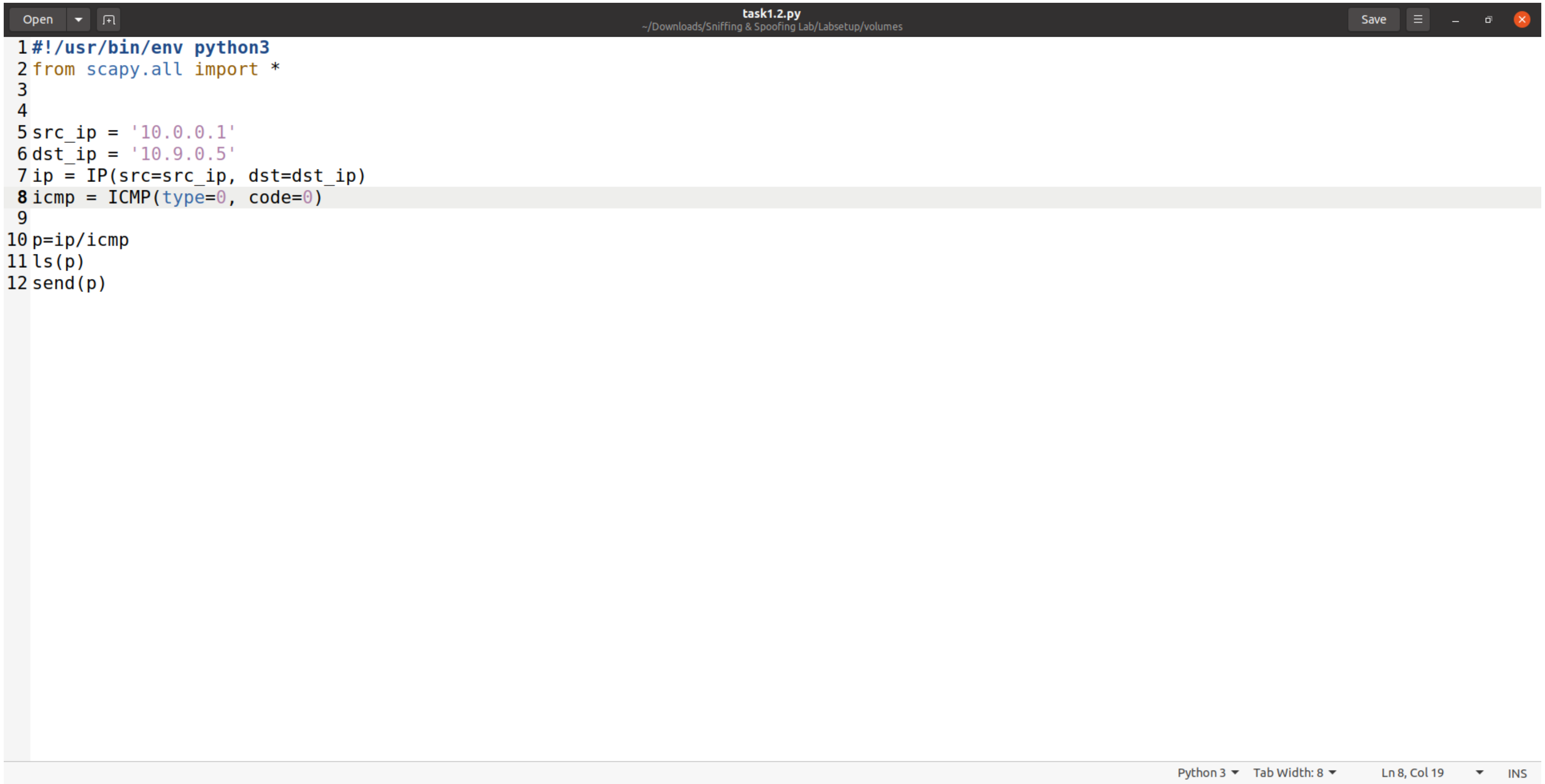
```
        seq       = 0x4
###[ Raw ]###
           load      = 'Y\xc6"e\x00\x00\x00\x00\xa1\xef\x05\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !"#$%&\'()*+,-./01234567'

^Croot@VM:/volumes# python3 task1.1.py
^Croot@VM:/volumes# python3 task1.1.py
###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:0a:09:00:06
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x10
     len       = 53
     id        = 16850
     flags     = DF
     frag      = 0
     ttl       = 64
     proto     = tcp
     chksum    = 0xe4c4
     src       = 10.9.0.6
     dst       = 10.9.0.5
     \options   \
###[ TCP ]###
        sport     = 50158
        dport     = telnet
        seq       = 3438517531
        ack       = 3166745789
        dataofs   = 8
        reserved  = 0
        flags     = PA
        window    = 501
        chksum    = 0x1444
        urgptr    = 0
        options   = [('NOP', None), ('NOP', None), ('Timestamp', (443629562, 3675627561))]
###[ Raw ]###
           load      = 't'

###[ Ethernet ]###
  dst       = 02:42:0a:09:00:05
  src       = 02:42:0a:09:00:06
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x10
```

* In *HostB-10.9.0.6* I ran the command **$ telnet 10.9.0.5**

Task 1.2

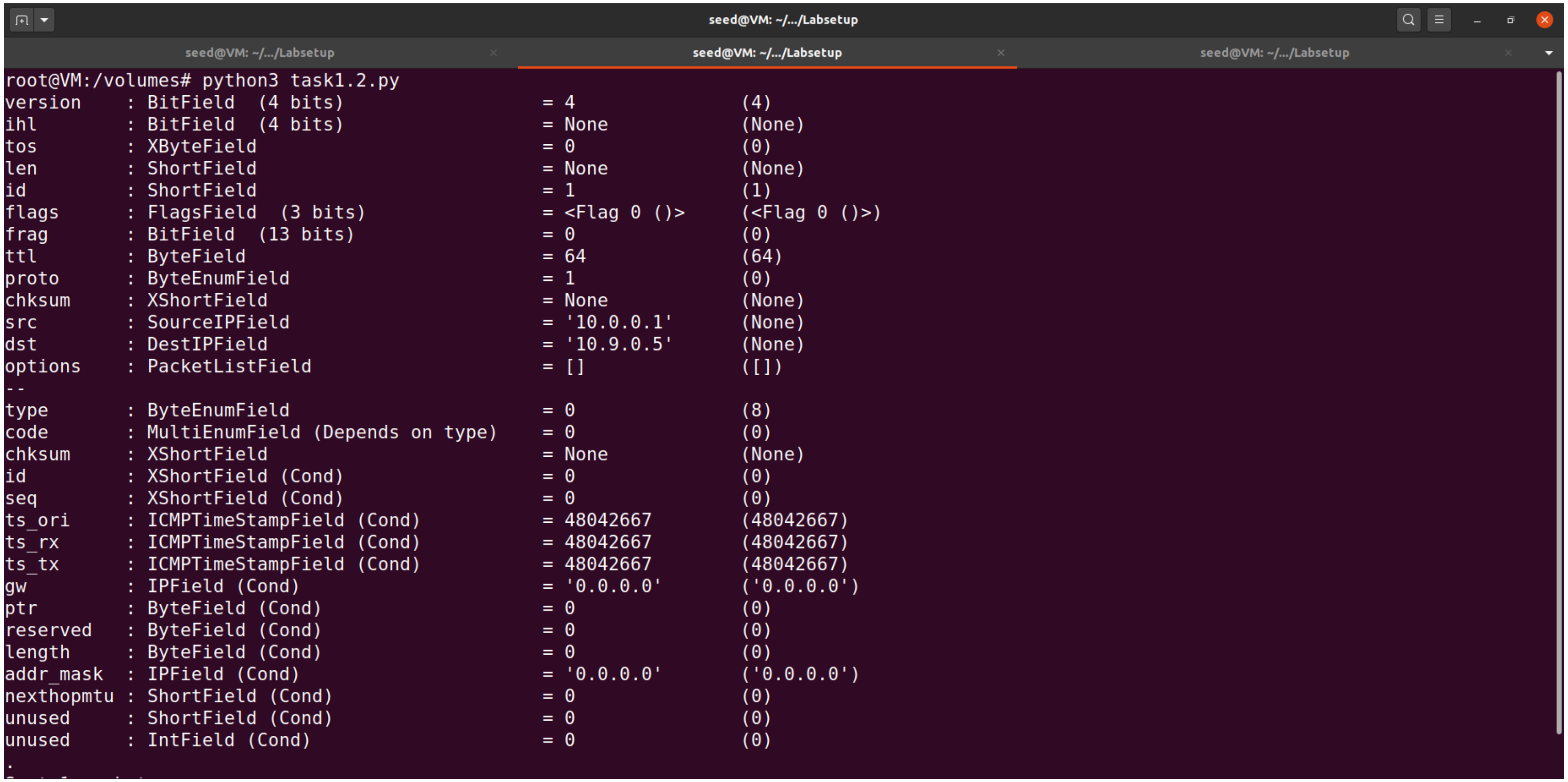
Code

- Sends an echo reply from 10.0.0.1 to 10.9.0.5



```python
#!/usr/bin/env python3
from scapy.all import *


src_ip = '10.0.0.1'
dst_ip = '10.9.0.5'
ip = IP(src=src_ip, dst=dst_ip)
icmp = ICMP(type=0, code=0)

p=ip/icmp
ls(p)
send(p)
```

Execution

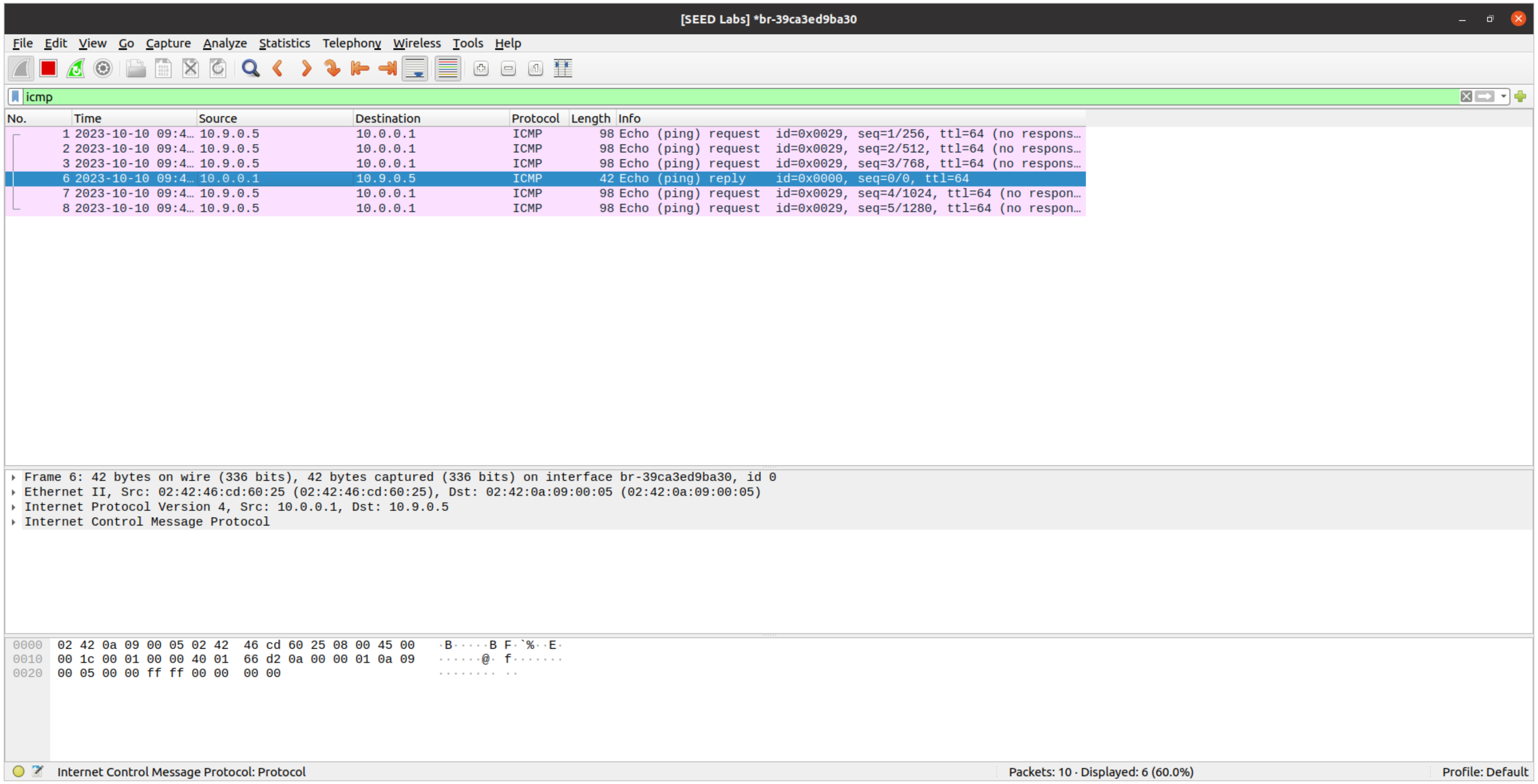


```
root@VM:/volumes# python3 task1.2.py
version    : BitField  (4 bits)                  = 4               (4)
ihl        : BitField  (4 bits)                  = None            (None)
tos        : XByteField                          = 0               (0)
len        : ShortField                          = None            (None)
id         : ShortField                          = 1               (1)
flags      : FlagsField  (3 bits)                = <Flag 0 ()>     (<Flag 0 ()>)
frag       : BitField  (13 bits)                 = 0               (0)
ttl        : ByteField                           = 64              (64)
proto      : ByteEnumField                       = 1               (0)
chksum     : XShortField                         = None            (None)
src        : SourceIPField                       = '10.0.0.1'      (None)
dst        : DestIPField                         = '10.9.0.5'      (None)
options    : PacketListField                     = []              ([])
--
type       : ByteEnumField                       = 0               (8)
code       : MultiEnumField (Depends on type)    = 0               (0)
chksum     : XShortField                         = None            (None)
id         : XShortField (Cond)                  = 0               (0)
seq        : XShortField (Cond)                  = 0               (0)
ts_ori     : ICMPTimeStampField (Cond)           = 48042667        (48042667)
ts_rx      : ICMPTimeStampField (Cond)           = 48042667        (48042667)
ts_tx      : ICMPTimeStampField (Cond)           = 48042667        (48042667)
gw         : IPField (Cond)                      = '0.0.0.0'       ('0.0.0.0')
ptr        : ByteField (Cond)                    = 0               (0)
reserved   : ByteField (Cond)                    = 0               (0)
length     : ByteField (Cond)                    = 0               (0)
addr_mask  : IPField (Cond)                      = '0.0.0.0'       ('0.0.0.0')
nexthopmtu : ShortField (Cond)                   = 0               (0)
unused     : ShortField (Cond)                   = 0               (0)
unused     : IntField (Cond)                     = 0               (0)
.
```

DIPARTIMENTO
MATEMATICA

Wireshark

- Using *HostA-10.9.0.5* I ran the command **$ ping 10.0.0.1**



[SEED Labs] *br-39ca3ed9ba30
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

icmp

```
No.    Time               Source     Destination  Protocol Length Info
     1 2023-10-10 09:4… 10.9.0.5   10.0.0.1     ICMP       98 Echo (ping) request  id=0x0029, seq=1/256, ttl=64 (no respons…
     2 2023-10-10 09:4… 10.9.0.5   10.0.0.1     ICMP       98 Echo (ping) request  id=0x0029, seq=2/512, ttl=64 (no respons…
     3 2023-10-10 09:4… 10.9.0.5   10.0.0.1     ICMP       98 Echo (ping) request  id=0x0029, seq=3/768, ttl=64 (no respons…
     6 2023-10-10 09:4… 10.0.0.1   10.9.0.5     ICMP       42 Echo (ping) reply    id=0x0000, seq=0/0, ttl=64
     7 2023-10-10 09:4… 10.9.0.5   10.0.0.1     ICMP       98 Echo (ping) request  id=0x0029, seq=4/1024, ttl=64 (no respon…
     8 2023-10-10 09:4… 10.9.0.5   10.0.0.1     ICMP       98 Echo (ping) request  id=0x0029, seq=5/1280, ttl=64 (no respon…
```

```
› Frame 6: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface br-39ca3ed9ba30, id 0
› Ethernet II, Src: 02:42:46:cd:60:25 (02:42:46:cd:60:25), Dst: 02:42:0a:09:00:05 (02:42:0a:09:00:05)
› Internet Protocol Version 4, Src: 10.0.0.1, Dst: 10.9.0.5
› Internet Control Message Protocol
```

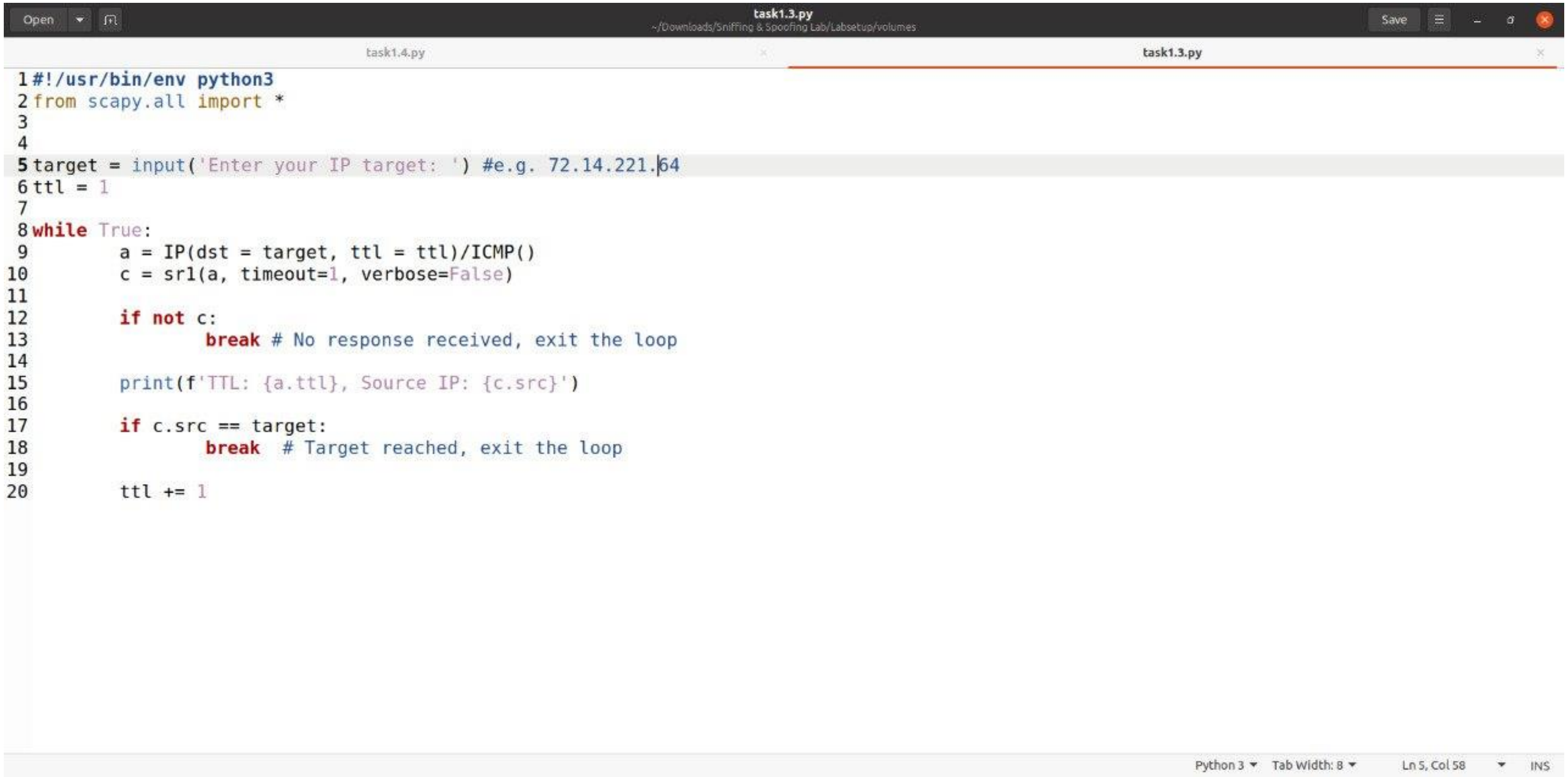
```
0000  02 42 0a 09 00 05 02 42  46 cd 60 25 08 00 45 00   ·B·····B F·`%··E·
0010  00 1c 00 01 00 00 40 01  66 d2 0a 00 00 01 0a 09   ······@· f·······
0020  00 05 00 00 ff ff 00 00  00 00                     ········ ··
```

Internet Control Message Protocol: Protocol   Packets: 10 · Displayed: 6 (60.0%)   Profile: Default

Task 1.3

Code

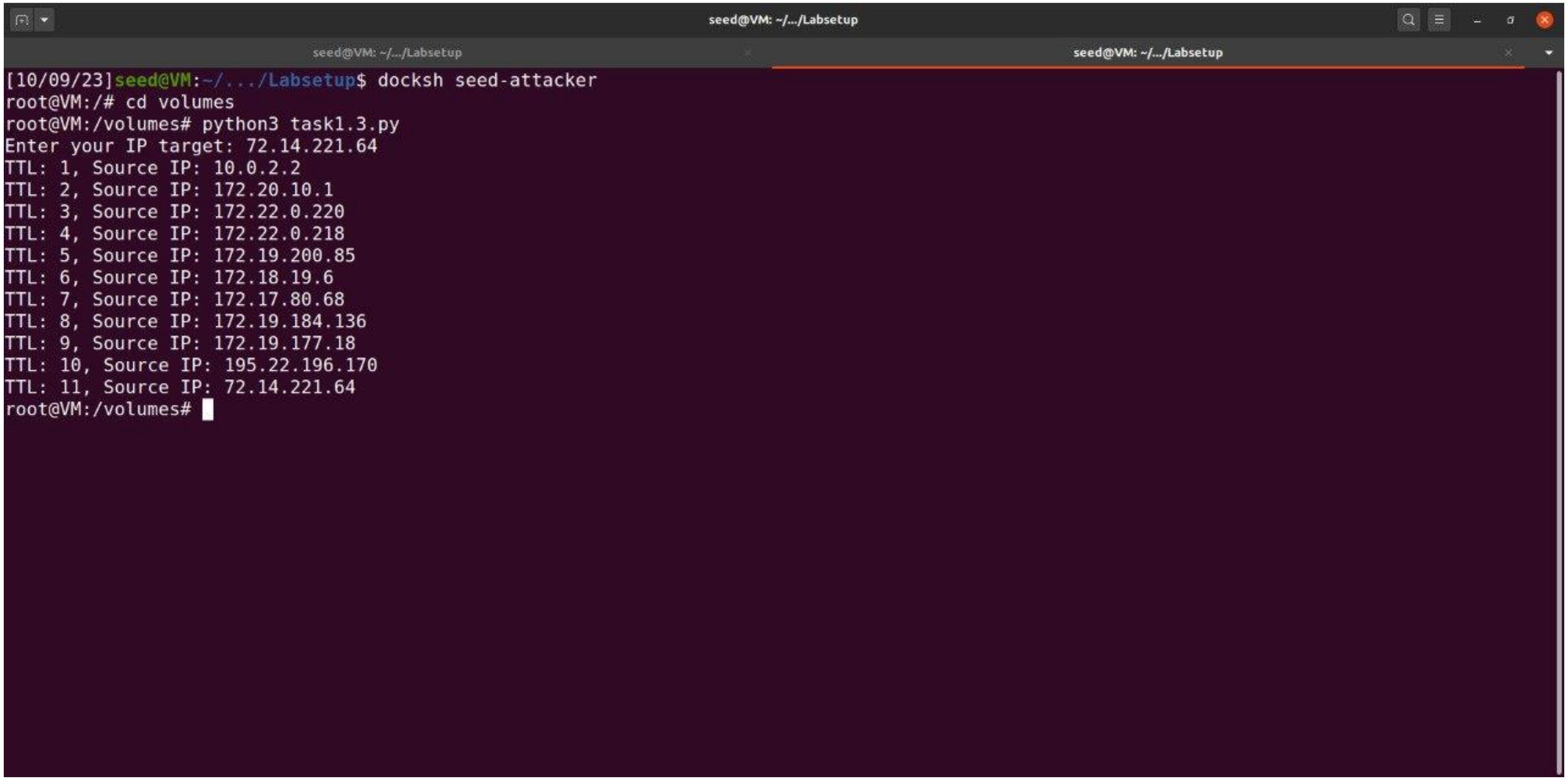
- Tracerouting



```
#!/usr/bin/env python3
from scapy.all import *


target = input('Enter your IP target: ') #e.g. 72.14.221.64
ttl = 1

while True:
        a = IP(dst = target, ttl = ttl)/ICMP()
        c = sr1(a, timeout=1, verbose=False)

        if not c:
                break # No response received, exit the loop

        print(f'TTL: {a.ttl}, Source IP: {c.src}')

        if c.src == target:
                break  # Target reached, exit the loop

        ttl += 1
```

Python 3 ▾ Tab Width: 8 ▾ Ln 5, Col 58 ▾ INS

Execution

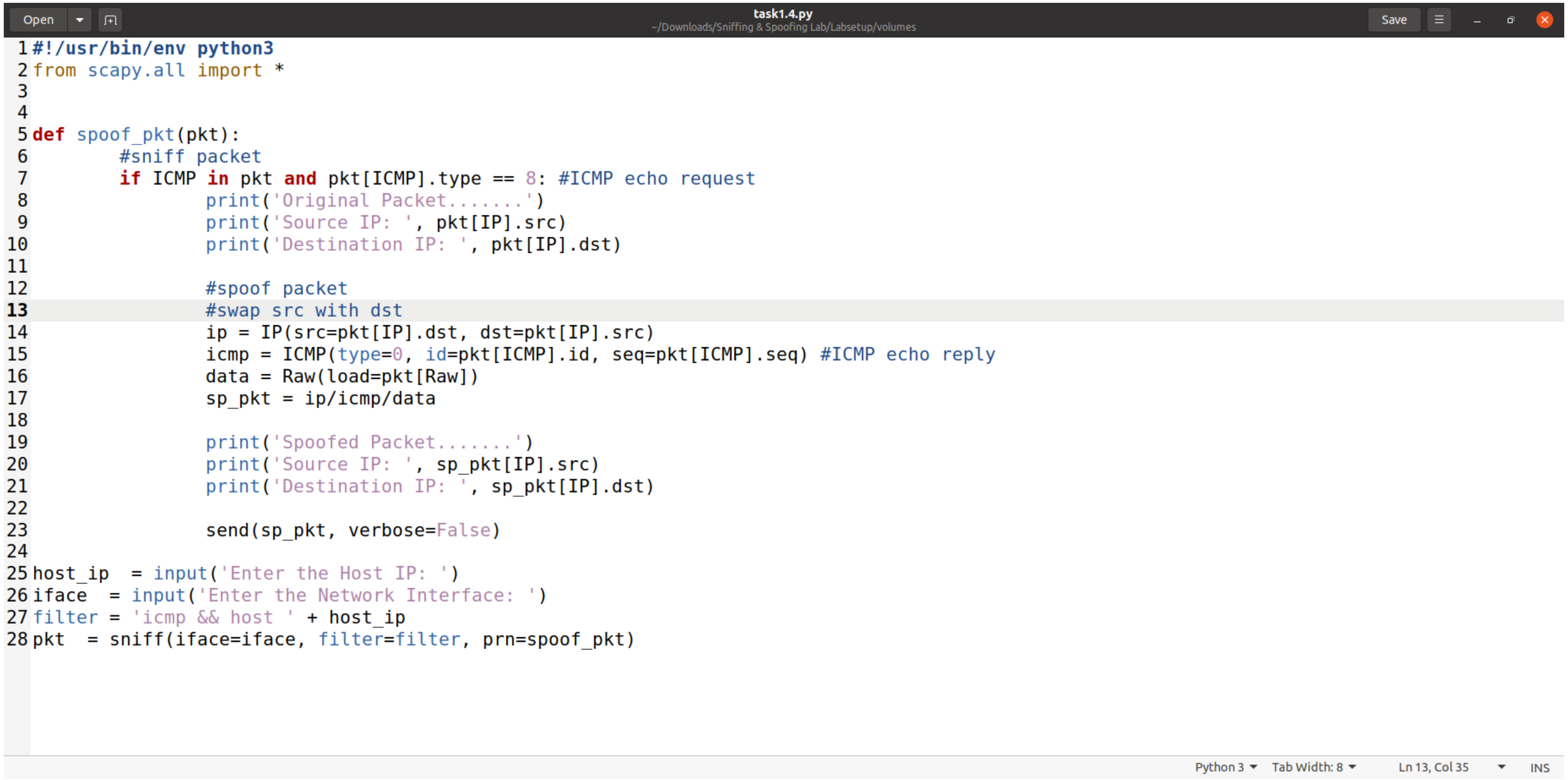


```
[10/09/23]seed@VM:~/.../Labsetup$ docksh seed-attacker
root@VM:/# cd volumes
root@VM:/volumes# python3 task1.3.py
Enter your IP target: 72.14.221.64
TTL: 1, Source IP: 10.0.2.2
TTL: 2, Source IP: 172.20.10.1
TTL: 3, Source IP: 172.22.0.220
TTL: 4, Source IP: 172.22.0.218
TTL: 5, Source IP: 172.19.200.85
TTL: 6, Source IP: 172.18.19.6
TTL: 7, Source IP: 172.17.80.68
TTL: 8, Source IP: 172.19.184.136
TTL: 9, Source IP: 172.19.177.18
TTL: 10, Source IP: 195.22.196.170
TTL: 11, Source IP: 72.14.221.64
root@VM:/volumes# 
```

Task 1.4

Code

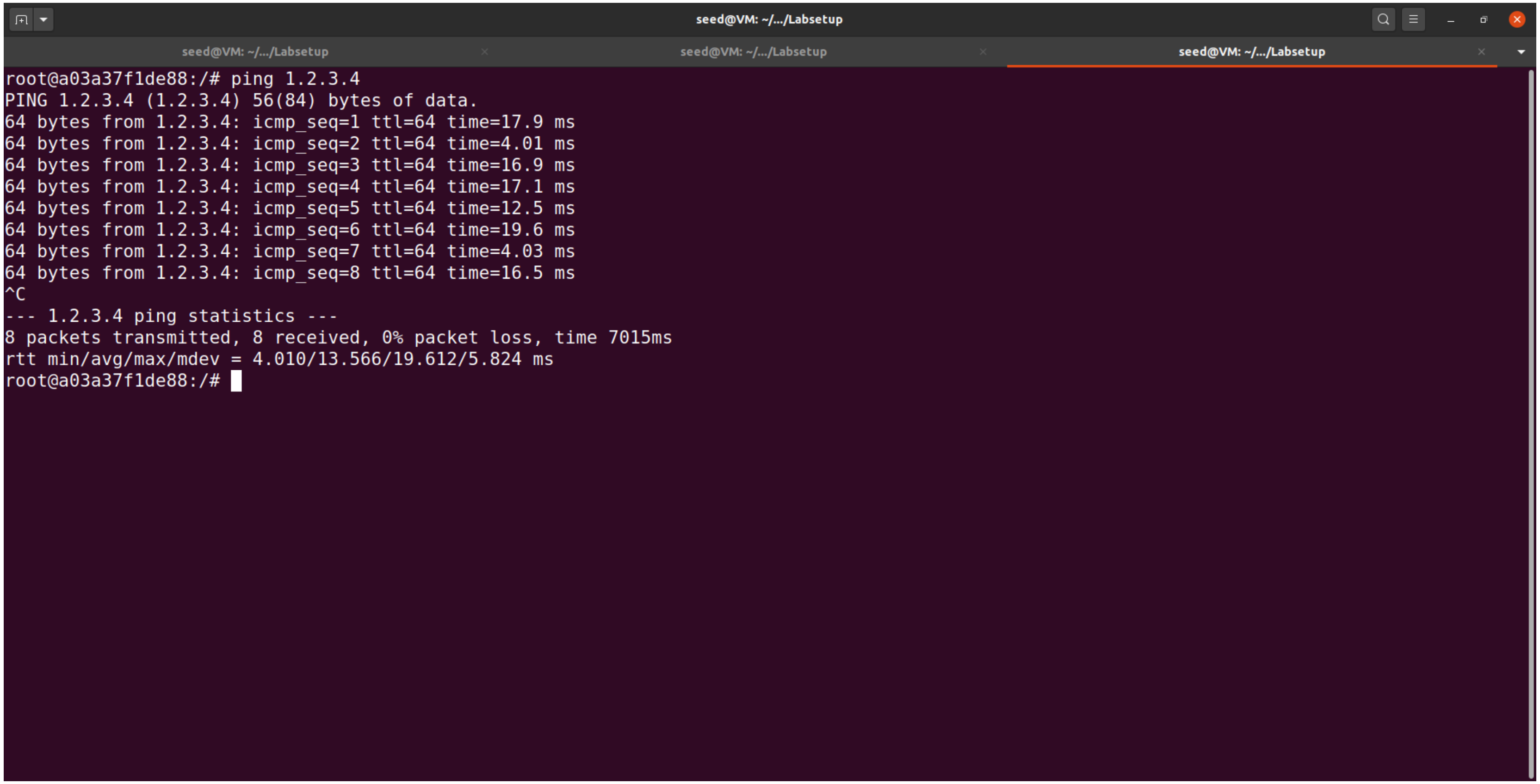
- Sniffing + Spoofing



```python
#!/usr/bin/env python3
from scapy.all import *


def spoof_pkt(pkt):
        #sniff packet
        if ICMP in pkt and pkt[ICMP].type == 8: #ICMP echo request
                print('Original Packet.......')
                print('Source IP: ', pkt[IP].src)
                print('Destination IP: ', pkt[IP].dst)

                #spoof packet
                #swap src with dst
                ip = IP(src=pkt[IP].dst, dst=pkt[IP].src)
                icmp = ICMP(type=0, id=pkt[ICMP].id, seq=pkt[ICMP].seq) #ICMP echo reply
                data = Raw(load=pkt[Raw])
                sp_pkt = ip/icmp/data

                print('Spoofed Packet.......')
                print('Source IP: ', sp_pkt[IP].src)
                print('Destination IP: ', sp_pkt[IP].dst)

                send(sp_pkt, verbose=False)

host_ip  = input('Enter the Host IP: ')
iface  = input('Enter the Network Interface: ')
filter = 'icmp && host ' + host_ip
pkt  = sniff(iface=iface, filter=filter, prn=spoof_pkt)
```

Host 1.2.3.4 - Ping

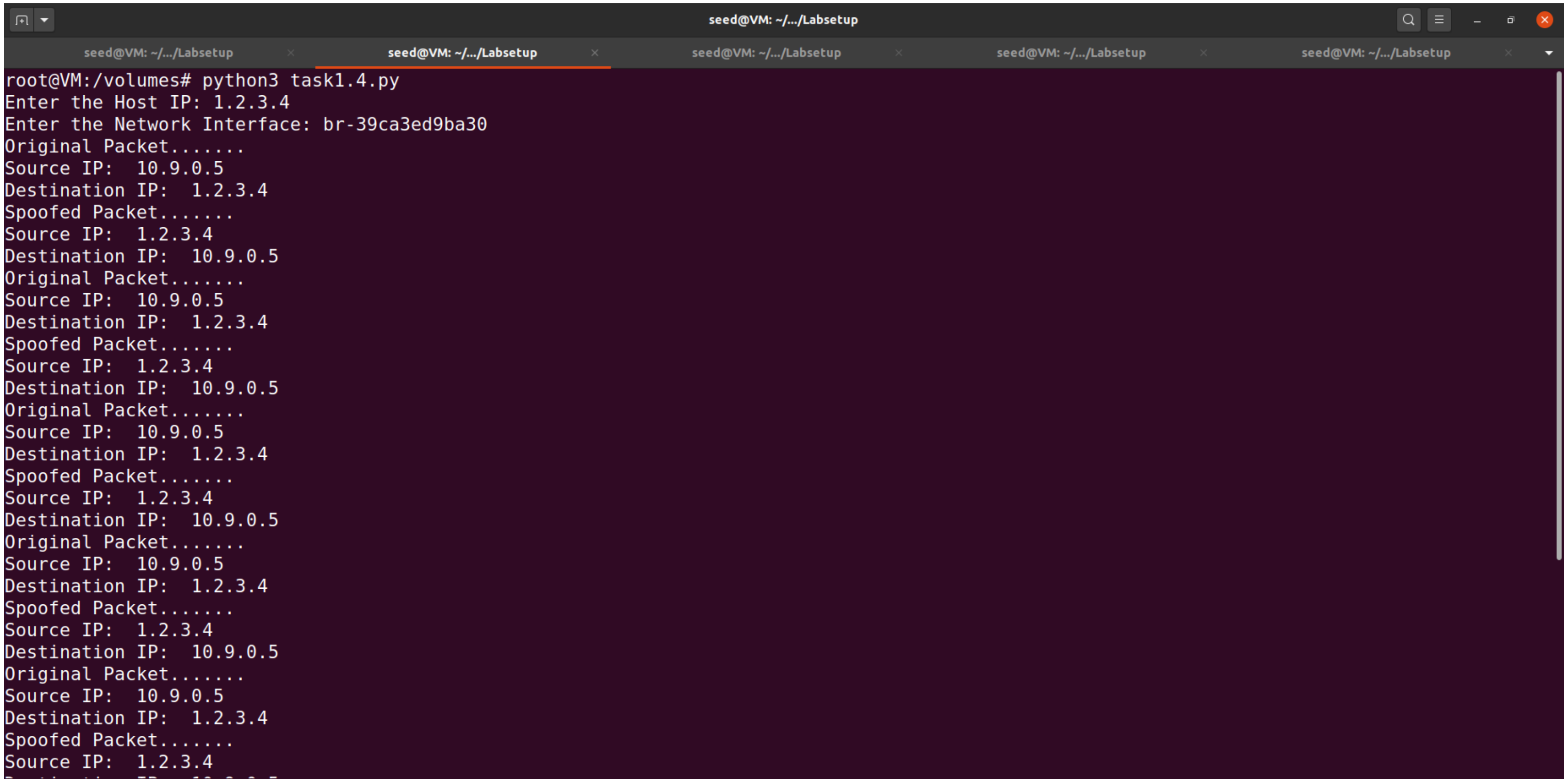
- Non-existing host on the Internet
- **$ ip route get 1.2.3.4** -> *10.9.0.1 dev eth0 src 10.9.0.5 uid 0*



```
root@a03a37f1de88:/# 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=64 time=17.9 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=64 time=4.01 ms
64 bytes from 1.2.3.4: icmp_seq=3 ttl=64 time=16.9 ms
64 bytes from 1.2.3.4: icmp_seq=4 ttl=64 time=17.1 ms
64 bytes from 1.2.3.4: icmp_seq=5 ttl=64 time=12.5 ms
64 bytes from 1.2.3.4: icmp_seq=6 ttl=64 time=19.6 ms
64 bytes from 1.2.3.4: icmp_seq=7 ttl=64 time=4.03 ms
64 bytes from 1.2.3.4: icmp_seq=8 ttl=64 time=16.5 ms
^C
--- 1.2.3.4 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7015ms
rtt min/avg/max/mdev = 4.010/13.566/19.612/5.824 ms
root@a03a37f1de88:/# 
```

Host 1.2.3.4 - Spoofing

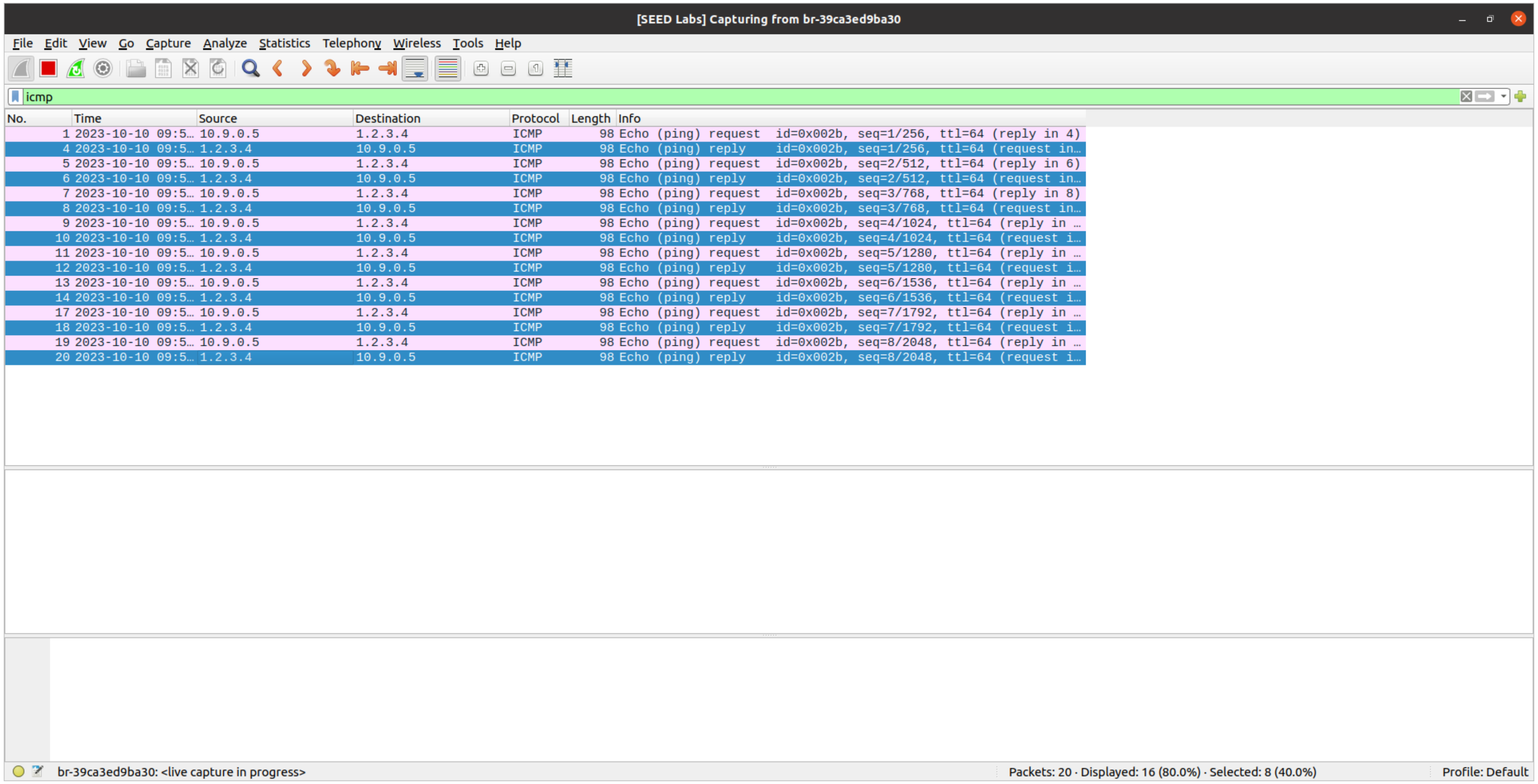
- Spoofing works because traffic is sent via the gateway with IP address 10.9.0.1



```
root@VM:/volumes# python3 task1.4.py
Enter the Host IP: 1.2.3.4
Enter the Network Interface: br-39ca3ed9ba30
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
```

Host 1.2.3.4 - Wireshark

- Consequently, we can see echo requests with their corresponding (spoofed) echo replies


[SEED Labs] Capturing from br-39ca3ed9ba30
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

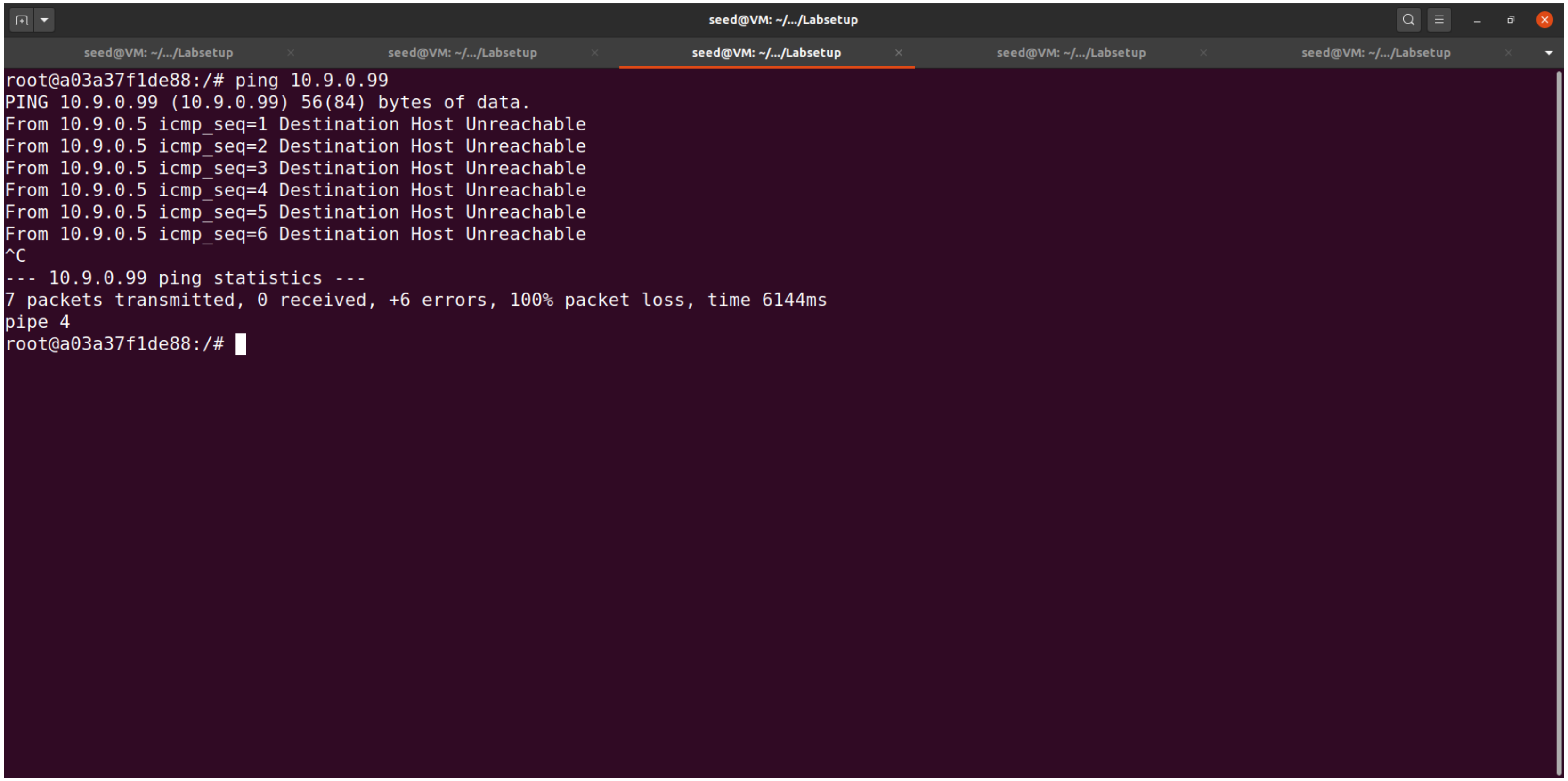
icmp

No.	Time	Source	Destination	Protocol	Length	Info
1	2023-10-10 09:5…	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x002b, seq=1/256, ttl=64 (reply in 4)
4	2023-10-10 09:5…	1.2.3.4	10.9.0.5	ICMP	98	Echo (ping) reply id=0x002b, seq=1/256, ttl=64 (request in…
5	2023-10-10 09:5…	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x002b, seq=2/512, ttl=64 (reply in 6)
6	2023-10-10 09:5…	1.2.3.4	10.9.0.5	ICMP	98	Echo (ping) reply id=0x002b, seq=2/512, ttl=64 (request in…
7	2023-10-10 09:5…	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x002b, seq=3/768, ttl=64 (reply in 8)
8	2023-10-10 09:5…	1.2.3.4	10.9.0.5	ICMP	98	Echo (ping) reply id=0x002b, seq=3/768, ttl=64 (request in…
9	2023-10-10 09:5…	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x002b, seq=4/1024, ttl=64 (reply in …
10	2023-10-10 09:5…	1.2.3.4	10.9.0.5	ICMP	98	Echo (ping) reply id=0x002b, seq=4/1024, ttl=64 (request i…
11	2023-10-10 09:5…	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x002b, seq=5/1280, ttl=64 (reply in …
12	2023-10-10 09:5…	1.2.3.4	10.9.0.5	ICMP	98	Echo (ping) reply id=0x002b, seq=5/1280, ttl=64 (request i…
13	2023-10-10 09:5…	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x002b, seq=6/1536, ttl=64 (reply in …
14	2023-10-10 09:5…	1.2.3.4	10.9.0.5	ICMP	98	Echo (ping) reply id=0x002b, seq=6/1536, ttl=64 (request i…
17	2023-10-10 09:5…	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x002b, seq=7/1792, ttl=64 (reply in …
18	2023-10-10 09:5…	1.2.3.4	10.9.0.5	ICMP	98	Echo (ping) reply id=0x002b, seq=7/1792, ttl=64 (request i…
19	2023-10-10 09:5…	10.9.0.5	1.2.3.4	ICMP	98	Echo (ping) request id=0x002b, seq=8/2048, ttl=64 (reply in …
20	2023-10-10 09:5…	1.2.3.4	10.9.0.5	ICMP	98	Echo (ping) reply id=0x002b, seq=8/2048, ttl=64 (request i…

br-39ca3ed9ba30: <live capture in progress>   Packets: 20 · Displayed: 16 (80.0%) · Selected: 8 (40.0%)   Profile: Default

Host 10.9.0.99 - Ping

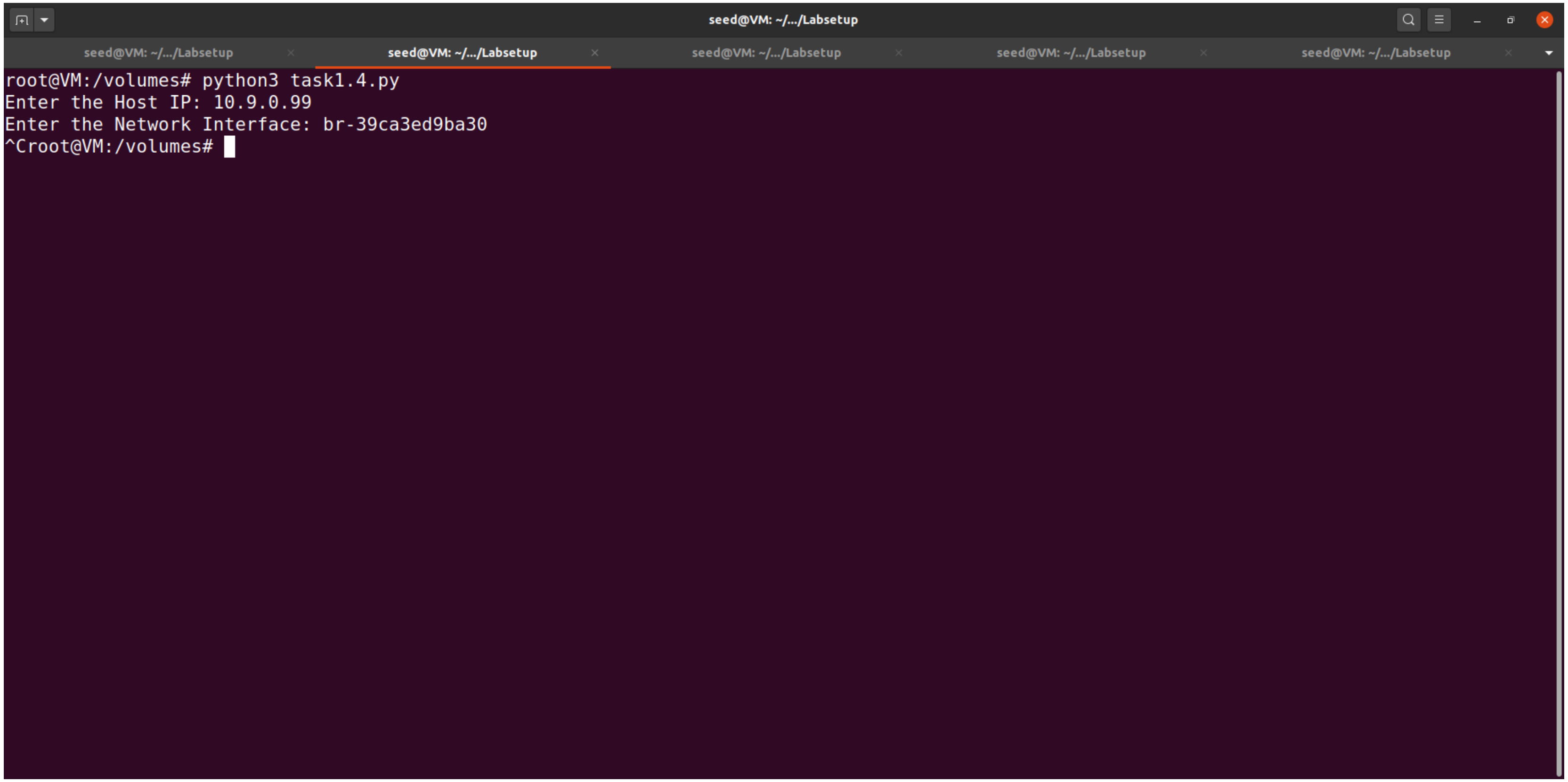
- Non-existing host on the LAN
- No echo requests are sent



```
root@a03a37f1de88:/# ping 10.9.0.99
PING 10.9.0.99 (10.9.0.99) 56(84) bytes of data.
From 10.9.0.5 icmp_seq=1 Destination Host Unreachable
From 10.9.0.5 icmp_seq=2 Destination Host Unreachable
From 10.9.0.5 icmp_seq=3 Destination Host Unreachable
From 10.9.0.5 icmp_seq=4 Destination Host Unreachable
From 10.9.0.5 icmp_seq=5 Destination Host Unreachable
From 10.9.0.5 icmp_seq=6 Destination Host Unreachable
^C
--- 10.9.0.99 ping statistics ---
7 packets transmitted, 0 received, +6 errors, 100% packet loss, time 6144ms
pipe 4
root@a03a37f1de88:/# 
```

Host 10.9.0.99 - Spoofing

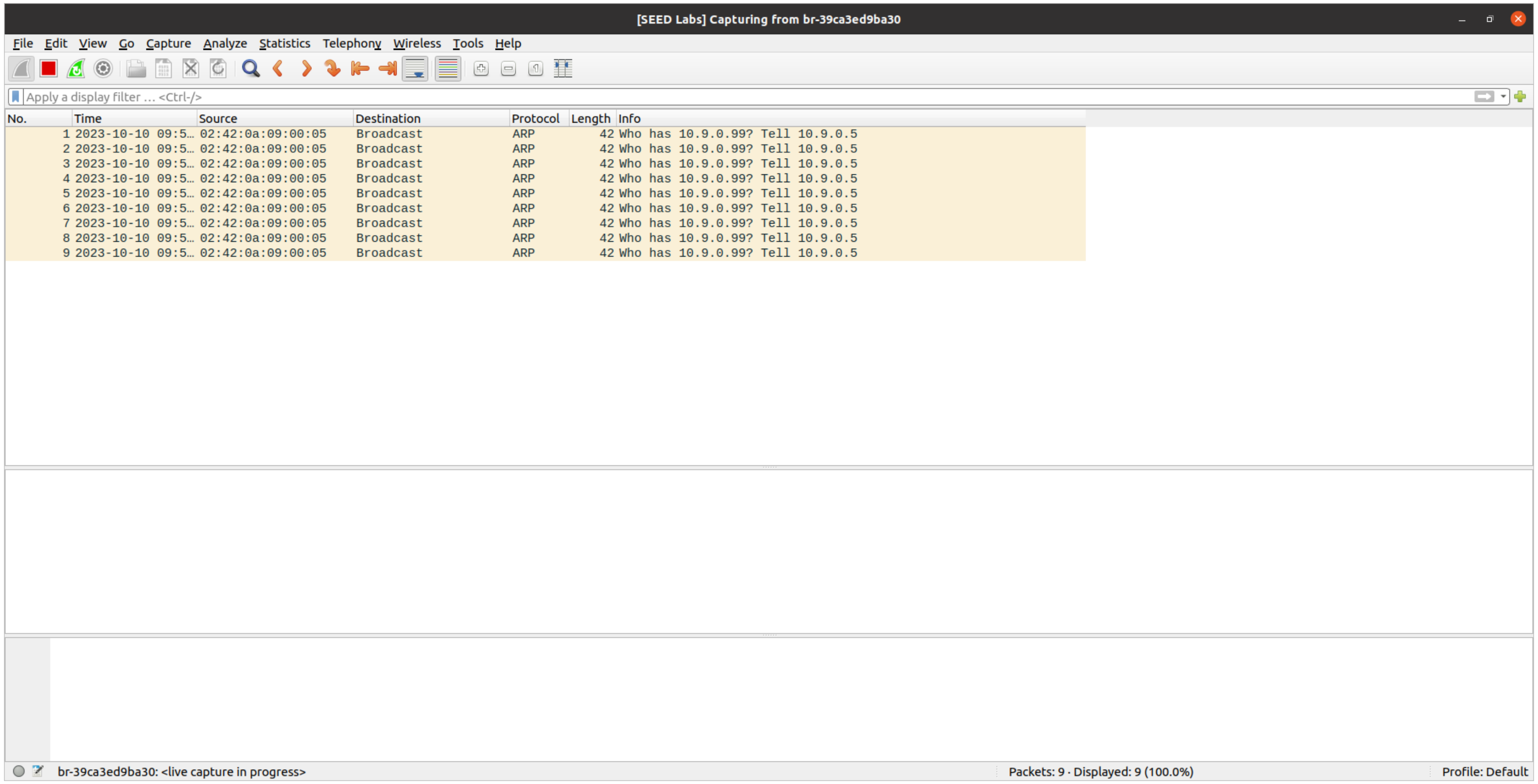
- No packets spoofed



```
root@VM:/volumes# python3 task1.4.py
Enter the Host IP: 10.9.0.99
Enter the Network Interface: br-39ca3ed9ba30
^Croot@VM:/volumes# 
```

Host 10.9.0.99 - Wireshark

- The sender device creates an ARP packet and then broadcasts to all devices in the same local area network
- No ARP Reply because the receiver does not exist

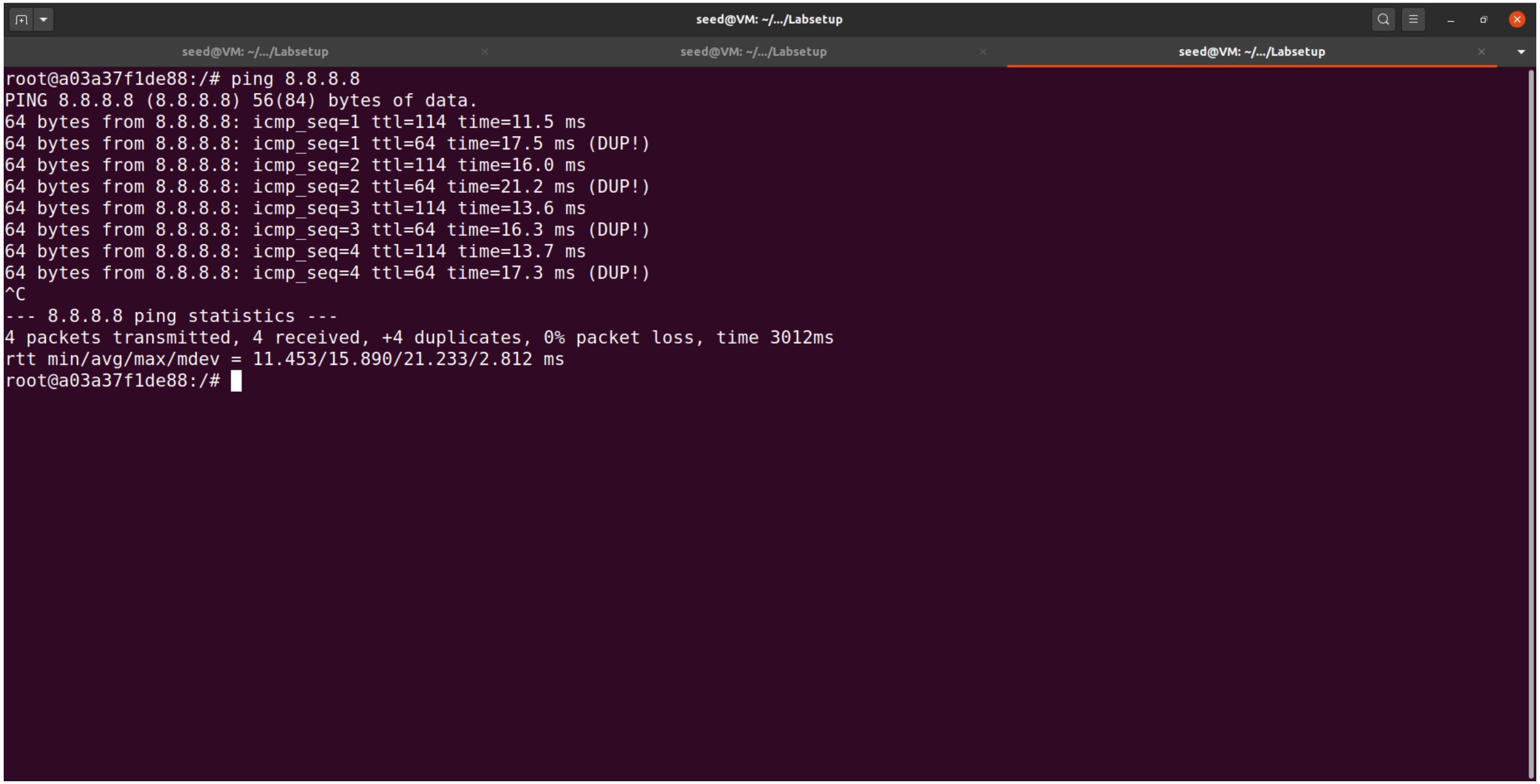

[SEED Labs] Capturing from br-39ca3ed9ba30
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time                 Source             Destination   Protocol Length Info
    1 2023-10-10 09:5… 02:42:0a:09:00:05  Broadcast     ARP          42 Who has 10.9.0.99? Tell 10.9.0.5
    2 2023-10-10 09:5… 02:42:0a:09:00:05  Broadcast     ARP          42 Who has 10.9.0.99? Tell 10.9.0.5
    3 2023-10-10 09:5… 02:42:0a:09:00:05  Broadcast     ARP          42 Who has 10.9.0.99? Tell 10.9.0.5
    4 2023-10-10 09:5… 02:42:0a:09:00:05  Broadcast     ARP          42 Who has 10.9.0.99? Tell 10.9.0.5
    5 2023-10-10 09:5… 02:42:0a:09:00:05  Broadcast     ARP          42 Who has 10.9.0.99? Tell 10.9.0.5
    6 2023-10-10 09:5… 02:42:0a:09:00:05  Broadcast     ARP          42 Who has 10.9.0.99? Tell 10.9.0.5
    7 2023-10-10 09:5… 02:42:0a:09:00:05  Broadcast     ARP          42 Who has 10.9.0.99? Tell 10.9.0.5
    8 2023-10-10 09:5… 02:42:0a:09:00:05  Broadcast     ARP          42 Who has 10.9.0.99? Tell 10.9.0.5
    9 2023-10-10 09:5… 02:42:0a:09:00:05  Broadcast     ARP          42 Who has 10.9.0.99? Tell 10.9.0.5
```

br-39ca3ed9ba30: <live capture in progress>   Packets: 9 · Displayed: 9 (100.0%)   Profile: Default

Host 8.8.8.8 - Ping

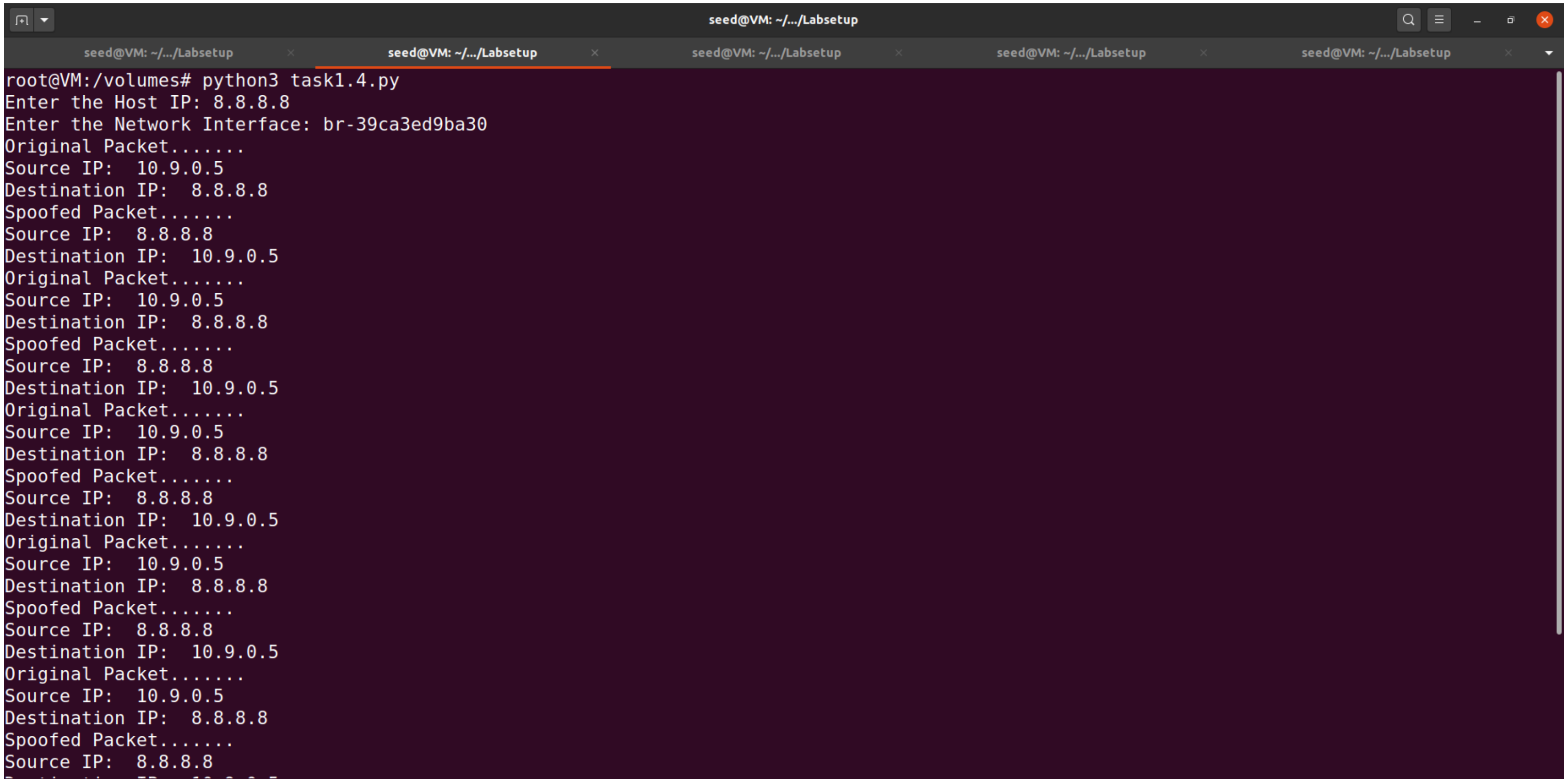
- Existing host on the Internet



```
root@a03a37f1de88:/# 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=114 time=11.5 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=64 time=17.5 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=2 ttl=114 time=16.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=64 time=21.2 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=3 ttl=114 time=13.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=64 time=16.3 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=4 ttl=114 time=13.7 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=64 time=17.3 ms (DUP!)
^C
--- 8.8.8.8 ping statistics ---
4 packets transmitted, 4 received, +4 duplicates, 0% packet loss, time 3012ms
rtt min/avg/max/mdev = 11.453/15.890/21.233/2.812 ms
root@a03a37f1de88:/# 
```

Host 8.8.8.8 - Spoofing

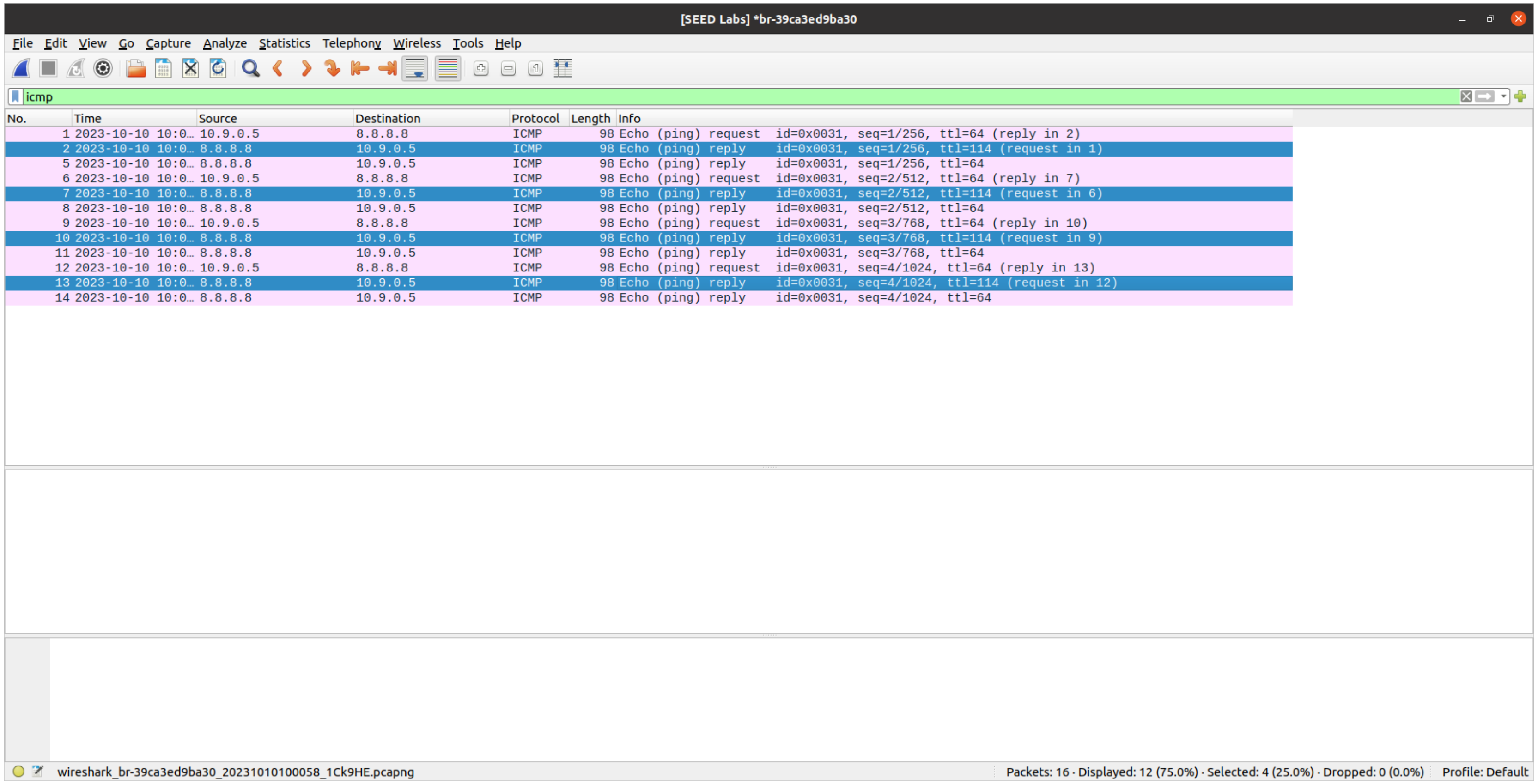
- By exchanging *src* and *dst* with each other, the spoofing attack creates duplicates



```
root@VM:/volumes# python3 task1.4.py
Enter the Host IP: 8.8.8.8
Enter the Network Interface: br-39ca3ed9ba30
Original Packet.......
Source IP:  10.9.0.5
Destination IP:  8.8.8.8
Spoofed Packet.......
Source IP:  8.8.8.8
Destination IP:  10.9.0.5
Original Packet.......
Source IP:  10.9.0.5
Destination IP:  8.8.8.8
Spoofed Packet.......
Source IP:  8.8.8.8
Destination IP:  10.9.0.5
Original Packet.......
Source IP:  10.9.0.5
Destination IP:  8.8.8.8
Spoofed Packet.......
Source IP:  8.8.8.8
Destination IP:  10.9.0.5
Original Packet.......
Source IP:  10.9.0.5
Destination IP:  8.8.8.8
Spoofed Packet.......
Source IP:  8.8.8.8
Destination IP:  10.9.0.5
Original Packet.......
Source IP:  10.9.0.5
Destination IP:  8.8.8.8
Spoofed Packet.......
Source IP:  8.8.8.8
```

Host 8.8.8.8 - Wireshark

- As we can see, each echo request have two echo replies
- The blue ones are the duplicates



icmp

```
No.  Time              Source      Destination  Protocol  Length Info
   1 2023-10-10 10:0… 10.9.0.5    8.8.8.8      ICMP        98 Echo (ping) request  id=0x0031, seq=1/256, ttl=64 (reply in 2)
   2 2023-10-10 10:0… 8.8.8.8     10.9.0.5     ICMP        98 Echo (ping) reply    id=0x0031, seq=1/256, ttl=114 (request in 1)
   5 2023-10-10 10:0… 8.8.8.8     10.9.0.5     ICMP        98 Echo (ping) reply    id=0x0031, seq=1/256, ttl=64
   6 2023-10-10 10:0… 10.9.0.5    8.8.8.8      ICMP        98 Echo (ping) request  id=0x0031, seq=2/512, ttl=64 (reply in 7)
   7 2023-10-10 10:0… 8.8.8.8     10.9.0.5     ICMP        98 Echo (ping) reply    id=0x0031, seq=2/512, ttl=114 (request in 6)
   8 2023-10-10 10:0… 8.8.8.8     10.9.0.5     ICMP        98 Echo (ping) reply    id=0x0031, seq=2/512, ttl=64
   9 2023-10-10 10:0… 10.9.0.5    8.8.8.8      ICMP        98 Echo (ping) request  id=0x0031, seq=3/768, ttl=64 (reply in 10)
  10 2023-10-10 10:0… 8.8.8.8     10.9.0.5     ICMP        98 Echo (ping) reply    id=0x0031, seq=3/768, ttl=114 (request in 9)
  11 2023-10-10 10:0… 8.8.8.8     10.9.0.5     ICMP        98 Echo (ping) reply    id=0x0031, seq=3/768, ttl=64
  12 2023-10-10 10:0… 10.9.0.5    8.8.8.8      ICMP        98 Echo (ping) request  id=0x0031, seq=4/1024, ttl=64 (reply in 13)
  13 2023-10-10 10:0… 8.8.8.8     10.9.0.5     ICMP        98 Echo (ping) reply    id=0x0031, seq=4/1024, ttl=114 (request in 12)
  14 2023-10-10 10:0… 8.8.8.8     10.9.0.5     ICMP        98 Echo (ping) reply    id=0x0031, seq=4/1024, ttl=64
```

wireshark_br-39ca3ed9ba30_20231010100058_1Ck9HE.pcapng   Packets: 16 · Displayed: 12 (75.0%) · Selected: 4 (25.0%) · Dropped: 0 (0.0%)   Profile: Default

Thanks for Your Attention

Niccolò Borgioli
Enkeleda Bardhi

Università degli Studi di Padova

Ethical Hacking

A.Y. 2023 - 2024

Dipartimento Matematica