

תקשורת ומחשוב - מטלה ששית

Packet Sniffing and Spoofing Lab

Task 1.1: Sniffing Packets

Task 1.1A:

לאחר הריצה של הקוד בשם **sys_A.py** (תמונה 1), על ידי root privilege עם הוספה של פקודת sudo לפני שם הקובץ, ניתן לראות את הפקודות ICMP (תמונה 2) שנשלחו מכתובת 10.0.2.6 ולשרת DNS של *google* (תמונה 3).



The screenshot shows a code editor window with the following details:

- Title Bar:** The title bar displays "task_1_1_A.py" and the path "~Desktop".
- File Menu:** The top-left corner has an "Open" button and a file icon.
- Save Button:** The top-right corner has a "Save" button, a three-dot menu icon, and a close (X) button.
- Code Content:** The main area contains a Python script:

```
1#!/usr/bin/env python3
2from scapy.all import *
3
4def print_pkt(pkt):
5    pkt.show()
6
7pkt = sniff(iface=['br-6a709a0e8789', 'enp0s3'], filter='icmp', prn=print_pkt)
```
- Status Bar:** The bottom status bar shows "Python 3" with a dropdown arrow, "Tab Width: 8" with a dropdown arrow, "Ln 7, Col 48" with a dropdown arrow, and "INS" indicating the current mode.

```
^[[A^C[01/01/22]seed@VM:~/Desktop$ chmod a+x task_1_1_A.py
[01/01/22]seed@VM:~/Desktop$ sudo python3 task_1_1_A.py
###[ Ethernet ]###
    dst      = 52:54:00:12:35:00
    src      = 08:00:27:de:a2:20
    type     = IPv4
###[ IP ]###
    version   = 4
    ihl       = 5
    tos       = 0x0
    len       = 84
    id        = 15909
    flags     = DF
    frag      = 0
    ttl       = 64
    proto     = icmp
    chksum   = 0xe06e
    src       = 10.0.2.6
    dst       = 8.8.8.8
    \options  \
###[ ICMP ]###
    type      = echo-request
    code      = 0
    chksum   = 0x2622
    id        = 0x6
    seq       = 0x1
###[ Raw ]###
    load      = 'G\r\xd0a\x00\x00\x00\x00\x00\xf2\x94\t\x00\x00\x00\x00\x00\x00\x00\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !%"$%&!'()*+,.../01234567'
```

```
[01/01/22] seed@VM:~/Desktop$ 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=117 time=43.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=50.5 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=44.1 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=44.2 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=117 time=44.4 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=117 time=43.9 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=117 time=44.2 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=117 time=43.8 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=117 time=44.3 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=117 time=43.5 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=117 time=44.0 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=117 time=43.4 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=117 time=43.9 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=117 time=43.6 ms
64 bytes from 8.8.8.8: icmp_seq=15 ttl=117 time=43.6 ms
64 bytes from 8.8.8.8: icmp_seq=16 ttl=117 time=44.0 ms
64 bytes from 8.8.8.8: icmp_seq=17 ttl=117 time=43.0 ms
64 bytes from 8.8.8.8: icmp_seq=18 ttl=117 time=48.3 ms
64 bytes from 8.8.8.8: icmp_seq=19 ttl=117 time=43.4 ms
64 bytes from 8.8.8.8: icmp_seq=20 ttl=117 time=44.0 ms
64 bytes from 8.8.8.8: icmp_seq=21 ttl=117 time=43.7 ms
64 bytes from 8.8.8.8: icmp_seq=22 ttl=117 time=44.1 ms
64 bytes from 8.8.8.8: icmp_seq=23 ttl=117 time=46.3 ms
64 bytes from 8.8.8.8: icmp_seq=24 ttl=117 time=44.1 ms
64 bytes from 8.8.8.8: icmp_seq=25 ttl=117 time=43.6 ms
64 bytes from 8.8.8.8: icmp_seq=26 ttl=117 time=43.7 ms
```

ריצת התוכנית בלי שימוש ב- `root privilege`, ככלمر בלי שימוש בפקודת `sudo` גורמת לשגיאה. אך אם רצים להסניף פקודות אනחן צריים להריץ ב- `root privilege` על מנת שתהיה לו גישה למשק שבו אナンחן מורים את הקוד.

```
[01/01/22]seed@VM:~/Desktop$ chmod a+x task_1_1_A.py
[01/01/22]seed@VM:~/Desktop$ python3 task_1_1_A.py
Traceback (most recent call last):
  File "task_1_1_A.py", line 7, in <module>
    pkt = sniff(iface=['br-6a709a0e8789', 'enp0s3'], 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 894, in _run
    sniff_sockets.update()
  File "/usr/local/lib/python3.8/dist-packages/scapy/sendrecv.py", line 895, in <genexpr>
    (L2socket(type=ETH_P_ALL, iface=ifname, *arg, **karg),
  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
```

Task 1.1B:

1. תפיסת פקודות ICMP בלבד ע"י שימוש ב-BPF

```
[01/01/22]seed@VM:~/Desktop$ chmod a+x task_1_1_A.py
[01/01/22]seed@VM:~/Desktop$ sudo python3 task_1_1_A.py
###[ Ethernet 1]##
dst      = 52:54:00:12:35:00
src      = 08:00:27:de:a2:20
type     = IPv4
###[ IP ]##
version  = 4
ihl      = 5
tos      = 0x0
len      = 84
id       = 61219
flags    = DF
frag     = 0
ttl      = 64
proto    = icmp
chksum   = 0x2f70
src      = 10.0.2.6
dst      = 8.8.8.8
'options' \
###[ ICMP ]##
type     = echo-request
code    = 0
checksum = 0xccaa
id      = 0x8
seq     = 0xa
###[ Raw ]##
load    = '\xd0\x1a\xd0\x0a\x00\x00\x00\x00\x00\x0b\xf7\x05\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !%"$&\'(*)+,-/01234567'
###[ Ethernet 1]##
```

seed@VM: ~/Desktop

```
64 bytes from 8.8.8.8: icmp_seq=12 ttl=117 time=44.2 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=117 time=43.5 ms
64 bytes from 8.8.8.8: icmp_seq=14 ttl=117 time=43.7 ms
64 bytes from 8.8.8.8: icmp_seq=15 ttl=117 time=43.7 ms
64 bytes from 8.8.8.8: icmp_seq=16 ttl=117 time=44.0 ms
64 bytes from 8.8.8.8: icmp_seq=17 ttl=117 time=43.3 ms
64 bytes from 8.8.8.8: icmp_seq=18 ttl=117 time=43.3 ms
64 bytes from 8.8.8.8: icmp_seq=19 ttl=117 time=45.2 ms
64 bytes from 8.8.8.8: icmp_seq=20 ttl=117 time=43.4 ms
^C
--- 8.8.8.8 ping statistics ---
20 packets transmitted, 20 received, 0% packet loss, time 19035ms
rtt min/avg/max/mdev = 43.167/43.815/45.161/0.459 ms
[01/01/22]seed@VM:~/Desktop$ 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=117 time=43.6 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=43.9 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=44.1 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=43.9 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=117 time=44.4 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=117 time=45.3 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=117 time=44.1 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=117 time=43.3 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=117 time=43.3 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=117 time=44.2 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=117 time=44.4 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=117 time=44.0 ms
64 bytes from 8.8.8.8: icmp_seq=13 ttl=117 time=44.8 ms
^C
--- 8.8.8.8 ping statistics ---
13 packets transmitted, 13 received, 0% packet loss, time 12098ms
rtt min/avg/max/mdev = 43.253/44.089/45.273/0.540 ms
```

Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
1	2022-01-01 03:1...	10.0.2.6	8.8.8.8	ICMP	98	Echo (ping) request id=0x0006, seq=1/256, ttl=64 (reply in 2)
2	2022-01-01 03:1...	8.8.8.8	10.0.2.6	ICMP	98	Echo (ping) reply id=0x0006, seq=1/256, ttl=117 (request in 1)
3	2022-01-01 03:1...	10.0.2.6	8.8.8.8	ICMP	98	Echo (ping) request id=0x0006, seq=2/512, ttl=64 (reply in 4)
4	2022-01-01 03:1...	8.8.8.8	10.0.2.6	ICMP	98	Echo (ping) reply id=0x0006, seq=2/512, ttl=117 (request in 3)
5	2022-01-01 03:1...	10.0.2.6	8.8.8.8	ICMP	98	Echo (ping) request id=0x0006, seq=3/768, ttl=64 (reply in 6)
6	2022-01-01 03:1...	8.8.8.8	10.0.2.6	ICMP	98	Echo (ping) reply id=0x0006, seq=3/768, ttl=117 (request in 5)
7	2022-01-01 03:1...	10.0.2.6	8.8.8.8	ICMP	98	Echo (ping) request id=0x0006, seq=4/1024, ttl=64 (reply in 8)
8	2022-01-01 03:1...	8.8.8.8	10.0.2.6	ICMP	98	Echo (ping) reply id=0x0006, seq=4/1024, ttl=117 (request in 7)

2. תפיסת פקודות הבאות מכתובת IP מסוימת עם port destination 23

```

task_1_1_A.py
[01/01/22]seed@VM:~/Desktop$ sudo python3 task_1_1_A.py
###[ Ethernet ]##
    dst      = 52:54:00:12:35:00
    src      = 08:00:27:de:a2:20
    type     = IPv4
###[ IP ]##
    version   = 4
    ihl       = 5
    tos       = 0x10
    len       = 60
    id        = 64937
    flags     = DF
    frag      = 0
    ttl       = 64
    proto     = tcp
    chksum   = 0x20ed
    src       = 10.0.2.6
    dst       = 8.8.8.8
    options   =
###[ TCP ]##
    sport     = 56154
    dport     = telnet
    seq       = 1312901517
    ack       = 0
    dataofs   = 10
    reserved  = 0
    flags     = S
    window    = 64240
    checksum  = 0x1c44
    urgptr   = 0
    options   = [ ('MSS', 1460), ('SACKOK', b''), ('Timestamp', (33925
56547, 0)), ('NOP', None), ('WScale', 7)]
```

[01/01/22]seed@VM:~/Desktop\$ icmp_seq=58 ttl=117 time=43.6 ms
64 bytes from 8.8.8.8: icmp_seq=59 ttl=117 time=43.4 ms
64 bytes from 8.8.8.8: icmp_seq=60 ttl=117 time=269 ms
64 bytes from 8.8.8.8: icmp_seq=61 ttl=117 time=43.5 ms
64 bytes from 8.8.8.8: icmp_seq=62 ttl=117 time=44.3 ms
64 bytes from 8.8.8.8: icmp_seq=63 ttl=117 time=44.4 ms
64 bytes from 8.8.8.8: icmp_seq=64 ttl=117 time=43.3 ms
64 bytes from 8.8.8.8: icmp_seq=65 ttl=117 time=43.8 ms
64 bytes from 8.8.8.8: icmp_seq=66 ttl=117 time=43.5 ms
64 bytes from 8.8.8.8: icmp_seq=67 ttl=117 time=43.3 ms
64 bytes from 8.8.8.8: icmp_seq=68 ttl=117 time=44.2 ms
64 bytes from 8.8.8.8: icmp_seq=69 ttl=117 time=43.5 ms
64 bytes from 8.8.8.8: icmp_seq=70 ttl=117 time=43.4 ms
64 bytes from 8.8.8.8: icmp_seq=71 ttl=117 time=43.1 ms
64 bytes from 8.8.8.8: icmp_seq=72 ttl=117 time=43.6 ms
64 bytes from 8.8.8.8: icmp_seq=73 ttl=117 time=43.8 ms
64 bytes from 8.8.8.8: icmp_seq=74 ttl=117 time=44.4 ms
64 bytes from 8.8.8.8: icmp_seq=75 ttl=117 time=43.8 ms
64 bytes from 8.8.8.8: icmp_seq=76 ttl=117 time=44.4 ms
64 bytes from 8.8.8.8: icmp_seq=77 ttl=117 time=43.7 ms
64 bytes from 8.8.8.8: icmp_seq=78 ttl=117 time=44.4 ms
64 bytes from 8.8.8.8: icmp_seq=79 ttl=117 time=46.2 ms
^C
--- 8.8.8.8 ping statistics ---
79 packets transmitted, 79 received, 0% packet loss, time 78443ms
rtt min/avg/max/mdev = 43.059/46.741/269.383/25.213 ms
[01/01/22]seed@VM:~/Desktop\$ telnet 8.8.8.8
Trying 8.8.8.8...
^C
[01/01/22]seed@VM:~/Desktop\$ telnet 8.8.8.8
Trying 8.8.8.8...
^C

Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
1	2022-01-01 04:2...	10.0.2.6	8.8.8.8	TCP	74	56154 -> 23 [SYN] Seq=1312901517 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TStamp=3392549436 TSecr=0 WS=128
2	2022-01-01 04:2...	10.0.2.6	8.8.8.8	TCP	74	[TCP Retransmission] 56154 -> 23 [SYN] Seq=1312901517 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TStamp=3392550468 TSecr=0 WS=128
3	2022-01-01 04:2...	10.0.2.6	8.8.8.8	TCP	74	[TCP Retransmission] 56154 -> 23 [SYN] Seq=1312901517 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TStamp=3392552484 TSecr=0 WS=128

Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_de:a2:20 (08:00:27:de:a2:20), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
Internet Protocol Version 4, Src: 10.0.2.6, Dst: 8.8.8.8
Transmission Control Protocol, Src Port: 56154, Dst Port: 23, Seq: 1312901517, Len: 0

3. תפיסת פקודות שבאות או נשלחות מ- subnet 128.230.0.0/16

```

task_1_1_A.py
[01/01/22]seed@VM:~/Desktop$ sudo python3 task_1_1_A.py
###[ ICMP ]##
    type      = Echo (ping)
    code      = 0
    checksum  = 98
    id        = 0x0012
    seq       = 1/256
    ttl       = 64 (no response found!)
    type      = Echo (ping)
    code      = 0
    checksum  = 98
    id        = 0x0012
    seq       = 2/512
    ttl       = 64 (no response found!)
    type      = Echo (ping)
    code      = 0
    checksum  = 98
    id        = 0x0012
    seq       = 3/768
    ttl       = 64 (no response found!)
    type      = Echo (ping)
    code      = 0
    checksum  = 98
    id        = 0x0012
    seq       = 4/1024
    ttl       = 64 (no response found!)
###[ NTP ]##
    version   = 4
    mode      = client
    origin    = 91.189.91.157
    recode   = 128.230.0.0
    stratum  = 0
    precision= 0
    offset   = 0.0
    delay    = 0.0
    dispersion= 0.0
    refid    = NTP Version 4, client
    flags    = 0
    auth     = 0
    keyid   = 0
    associd = 0
    authdata= 0
```

Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
1	2022-01-01 04:4...	10.0.2.6	128.230.0.0	ICMP	98	Echo (ping) request id=0x0012, seq=1/256, ttl=64 (no response found!)
2	2022-01-01 04:4...	10.0.2.6	128.230.0.0	ICMP	98	Echo (ping) request id=0x0012, seq=2/512, ttl=64 (no response found!)
3	2022-01-01 04:4...	10.0.2.6	128.230.0.0	ICMP	98	Echo (ping) request id=0x0012, seq=3/768, ttl=64 (no response found!)
4	2022-01-01 04:4...	10.0.2.6	128.230.0.0	ICMP	98	Echo (ping) request id=0x0012, seq=4/1024, ttl=64 (no response found!)
5	2022-01-01 04:4...	10.0.2.6	91.189.91.157	NTP	99	NTP Version 4, client
6	2022-01-01 04:4...	10.0.2.6	128.230.0.0	ICMP	98	Echo (ping) request id=0x0012, seq=5/1280, ttl=64 (no response found!)
7	2022-01-01 04:4...	91.189.91.157	10.0.2.6	NTP	99	NTP Version 4, server

Frame 4: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_de:a2:20 (08:00:27:de:a2:20), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
Internet Protocol Version 4, Src: 10.0.2.6, Dst: 128.230.0.0
Internet Control Message Protocol

```

seed@VM: ~/Desktop$ chmod a+x task_1_1_A.py
[01/01/22]seed@VM:~/Desktop$ sudo python3 task_1_1_A.py
###[ Ethernet ]###
dst      = 52:54:00:12:35:00
src      = 08:00:27:de:a2:20
type     = IPv4
###[ IP ]###
version  = 4
ihl      = 5
tos      = 0x0
len      = 84
id       = 22282
flags    = DF
frag     = 0
ttl      = 64
proto    = icmp
chksum   = 0x56b3
src      = 10.0.2.6
dst      = 128.230.0.0
\options \
###[ ICMP ]###
type     = echo-request
code     = 0
chksum   = 0x87e6
id       = 0x12
seq     = 0x7
###[ Raw ]###
load    = '^#\xd0a\x00\x00\x00\x00\x00y\x8\t\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f !#$%&('*+,-.01234567'

```

```

64 bytes from 8.8.8.8: icmp_seq=6 ttl=117 time=43.1 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=117 time=44.0 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=117 time=43.4 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=117 time=44.4 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=117 time=44.1 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=117 time=46.8 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=117 time=44.1 ms
^Z
[2]+ Stopped ping 8.8.8.8
[01/01/22]seed@VM:~/Desktop$ 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=117 time=43.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=44.2 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=43.9 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=43.3 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=117 time=43.7 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=117 time=44.3 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=117 time=44.3 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=117 time=46.2 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=117 time=43.9 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=117 time=43.7 ms
64 bytes from 8.8.8.8: icmp_seq=11 ttl=117 time=44.7 ms
64 bytes from 8.8.8.8: icmp_seq=12 ttl=117 time=44.3 ms
^Z
[3]+ Stopped ping 8.8.8.8
[01/01/22]seed@VM:~/Desktop$ ping 128.230.0.0
PING 128.230.0.0 (128.230.0.0) 56(84) bytes of data.
^C
-- 128.230.0.0 ping statistics --
22 packets transmitted, 0 received, 100% packet loss, time 21503ms

```

Task 1.2: Spoofing ICMP Packets

לאחר הריצה של הקוד בשם **Task_1_2.py** (תמונה 1), ניתן לראות שעשינו ע"י הסnipר של פקודות ICMP מהסעיף הקודם. השתמשנו כ-source IP בכתובת 10.2.0.6 של המכונה שאנו עבדים אליה וכ-destination את שרת DNS של גול (תמונה 2).

```

task_1_2.py
task_1_1_A.py
1from scapy.all import *
2
3a = IP()
4a.src = '10.0.2.6'
5a.dst = '8.8.8.8'
6b = ICMP()
7p = a/b
8send(p)
9ls(a)

```

```

[01/01/22]seed@VM:~/Desktop$ chmod a+x task_1_2.py
[01/01/22]seed@VM:~/Desktop$ sudo python3 task_1_2.py
.
Sent 1 packets.
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 ()>
0 ()>)
frag    : BitField (13 bits)         = 0           (0)
ttl     : ByteField                 = 64          (64)
proto   : ByteEnumField            = 0           (0)
chksum  : XShortField              = None        (None)
src     : SourceIPField            = '10.0.2.6'  (None)
dst     : DestIPField              = '8.8.8.8'   (None)
options : PacketListField          = []          ([])

[01/01/22]seed@VM:~/Desktop$ 

```

```

[01/01/22]seed@VM:~/Desktop$ sudo python3 task_1_1_A.py
###[ Ethernet ]###
dst      = 52:54:00:12:35:00
src      = 08:00:27:de:a2:20
type     = IPv4
###[ IP ]###
version  = 4
ihl      = 5
tos      = 0x0
len      = 28
id       = 1
flags    =
frag     = 0
ttl      = 64
proto    = icmp
chksum   = 0x5ecb
src      = 10.0.2.6
dst      = 8.8.8.8
\options \
###[ ICMP ]###
type     = echo-request
code     = 0
chksum   = 0xf7ff
id       = 0x0
seq     = 0x0
###[ Ethernet ]###
dst      = 08:00:27:de:a2:20
src      = 52:54:00:12:35:00
type     = IPv4
###[ IP ]###
version  = 4

```

No.	Time	Source	Destination	Protocol	Length	Info
6	2022-01-01 05:5...	8.8.8.8	127.0.0.1	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response ...
20	2022-01-01 05:5...	8.8.8.8	127.0.0.1	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=64 (no response ...
Frame 6: 44 bytes on wire (352 bits), 44 bytes captured (352 bits) on interface any, id 0						
Linux cooked capture						
Internet Protocol Version 4, Src: 8.8.8.8, Dst: 127.0.0.1						
Internet Control Message Protocol						

Task 1.3: Traceroute

בשאלה הזאת עלינו למצוא את `traceroute` כלומר כמה הנטבים שמספרדים בין הכתובת ה-IP של הכתובת ה-IP של היעד. על מנת לחשב `traceroute` עד השרת DNS של `Facebook` עם הכתובת IP 12.134.31.12, הוספנו לקטע קוד שמצויר (תמונה 1) בתרגיל כמה דברים לקוד שלנו `task_1_3.py`, משנהו בוליאני שיאפשר לנו בעצם לדעת אם הגענו לכתובת היעד או שאנוחנו עדין בדרך, רצמו בולאה בטוחה עד 50 (מספר מספיק גבוהה) ובעבור כל איטרציה בדקנו אם הכתובת המוחית שווה לכתובת היעד עם `timeout` מוגדר. (תמונה 2). ניתן לראות שהΤΤΤ הנדרש עד להגעה היעד הוא 9.

The screenshot shows a development environment with three windows:

- Code Editor:** Shows the Python script `task_1_3.py` which performs a traceroute to the IP 12.134.31.12. It uses Scapy to send ICMP echo requests with decreasing TTL values and prints the responses. The script includes imports for scapy.all, defines a `success` class, and a main loop for 50 iterations.
- Terminal 1:** Shows the command `sudo python3 task_1_3.py` being run, and the output indicating 9 successful packets received.
- Terminal 2:** Shows the command `sudo python3 task_1_3.py` again, with the output showing 9 successful packets received.
- Wireshark Capture:** Shows a packet list from Wireshark. It lists 20 captured frames, all of which are ICMP Echo requests (Type 8, Code 0) sent to the destination 12.134.31.12. The TTL field in the first few frames shows values decreasing from 255 down to 1, while later frames show TTL values of 0 and 1. The `Info` column for many frames indicates "Time-to-live exceeded".

No.	Time	Source	Destination	Protocol	Length	Info
3	2022-01-01 08:4.. 10.0.2.6	129.134.31.12	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=1 (no response found!)	
4	2022-01-01 08:4.. 10.0.2.1	10.0.2.6	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)	
5	2022-01-01 08:4.. 10.0.2.6	129.134.31.12	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=2 (no response found!)	
6	2022-01-01 08:4.. 31.168.13.77	10.0.2.6	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)	
7	2022-01-01 08:4.. 10.0.2.6	129.134.31.12	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=3 (no response found!)	
8	2022-01-01 08:4.. 212.179.16.121	10.0.2.6	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)	
9	2022-01-01 08:4.. 10.0.2.6	129.134.31.12	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=4 (no response found!)	
10	2022-01-01 08:4.. 212.179.124.85	10.0.2.6	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)	
11	2022-01-01 08:4.. 10.0.2.6	129.134.31.12	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=5 (no response found!)	
12	2022-01-01 08:4.. 10.91.99.6	10.0.2.6	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)	
13	2022-01-01 08:4.. 10.0.2.6	129.134.31.12	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=6 (no response found!)	
14	2022-01-01 08:4.. 157.240.74.60	10.0.2.6	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)	
15	2022-01-01 08:4.. 10.0.2.6	129.134.31.12	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=7 (no response found!)	
16	2022-01-01 08:4.. 129.134.36.117	10.0.2.6	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)	
17	2022-01-01 08:4.. 10.0.2.6	129.134.31.12	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=8 (no response found!)	
18	2022-01-01 08:4.. 157.240.39.73	10.0.2.6	ICMP	72	Time-to-live exceeded (Time to live exceeded in transit)	
19	2022-01-01 08:4.. 10.0.2.6	129.134.31.12	ICMP	44	Echo (ping) request id=0x0000, seq=0/0, ttl=9 (reply in 20)	
20	2022-01-01 08:4.. 129.134.31.12	10.0.2.6	ICMP	62	Echo (ping) reply id=0x0000, seq=0/0, ttl=56 (request in 19)	

The Wireshark capture shows 20 frames. The first 19 are ICMP echo requests (Type 8, Code 0) sent to the destination 129.134.31.12. The TTL field in the first few frames shows values decreasing from 255 down to 1, while later frames show TTL values of 0 and 1. The `Info` column for many frames indicates "Time-to-live exceeded". The 20th frame is an ICMP echo reply (Type 0, Code 0) sent back to the source 10.0.2.6, indicating that the final hop (the destination server) responded.

Task 1.4: Sniffing and then Spoofing

1.2.3.4 : השתמשתי בשתי מכונות וירטואליות, הראשה עם הכתובת **10.0.2.6** ש"תוקף" ע"י spoofing המכונה השנייה עם הכתובת **10.0.2.7** שולחת ping ל**1.2.3.4** שהוא host שלא קיים ברשת אינטרנט. ניתן לראות בתמונה בוירשאך את הפרטוקול ARP שהוא בעצם שואל ברשת למי שיר את כתובות היעד.

```

task_1_4.py
~/Desktop
seed@VM: ~/Desktop
[01/01/22]seed@VM:~/Desktop$ chmod a+x task_1_4.py
[01/01/22]seed@VM:~/Desktop$ sudo python3 task_1_4.py
***** BEFORE SPOOFING *****
>> IP SRC: 10.0.2.7
>> IP DST: 1.2.3.4
***** AFTER SPOOFING *****
>> IP SRC: 1.2.3.4
>> IP DST: 10.0.2.7

***** BEFORE SPOOFING *****
>> IP SRC: 10.0.2.7
>> IP DST: 1.2.3.4
***** AFTER SPOOFING *****
>> IP SRC: 1.2.3.4
>> IP DST: 10.0.2.7

***** BEFORE SPOOFING *****
>> IP SRC: 10.0.2.7
>> IP DST: 1.2.3.4
***** AFTER SPOOFING *****
>> IP SRC: 1.2.3.4
>> IP DST: 10.0.2.7

[01/01/22]seed@VM:~/Desktop$ ./task_1_4.py
***** BEFORE SPOOFING *****
>> IP SRC: 10.0.2.7
>> IP DST: 1.2.3.4
***** AFTER SPOOFING *****
>> IP SRC: 1.2.3.4
>> IP DST: 10.0.2.7

```

```

1#!/usr/bin/python3
2from scapy.all import *
3
4def spoofing(pkt):
5    if ICMP in pkt and pkt[ICMP].type == 8:      #is a request
6        print('***** BEFORE SPOOFING *****')
7        print(' >> IP SRC: ', pkt[IP].src)
8        print(' >> IP DST: ', pkt[IP].dst)
9
10   ip = IP(src=pkt[IP].dst, dst=pkt[IP].src, ihl=pkt[IP].ihl)
11   icmp = ICMP(type=0, id=pkt[ICMP].id, seq=pkt[ICMP].seq)
12   data = pkt[Raw].load
13   spoofed_pkt = ip/icmp/data
14
15   print('***** AFTER SPOOFING *****')
16   print(' >> IP SRC: ', spoofed_pkt[IP].src)
17   print(' >> IP DST: ', spoofed_pkt[IP].dst)
18   print("\n")
19   send(spoofed_pkt, verbose=0)
20
21pkt = sniff(iface=['br-6a709a0e8789', 'enp0s3'], filter='icmp and src host 10.0.2.7', prn=spoofing)

```

```

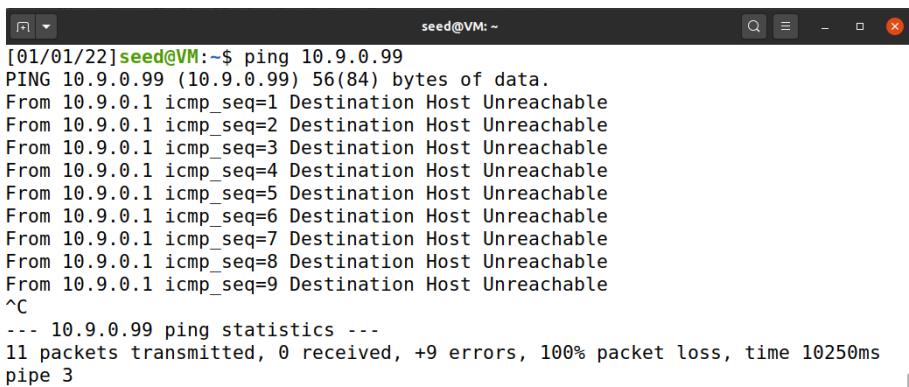
[01/01/22]seed@VM:~$ ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
64 bytes from 1.2.3.4: icmp_seq=1 ttl=64 time=54.5 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=64 time=15.3 ms
64 bytes from 1.2.3.4: icmp_seq=3 ttl=64 time=22.3 ms
64 bytes from 1.2.3.4: icmp_seq=4 ttl=64 time=15.5 ms
64 bytes from 1.2.3.4: icmp_seq=5 ttl=64 time=27.0 ms
64 bytes from 1.2.3.4: icmp_seq=6 ttl=64 time=23.1 ms
64 bytes from 1.2.3.4: icmp_seq=7 ttl=64 time=24.4 ms
64 bytes from 1.2.3.4: icmp_seq=8 ttl=64 time=32.6 ms
64 bytes from 1.2.3.4: icmp_seq=9 ttl=64 time=24.2 ms
^C
--- 1.2.3.4 ping statistics ---
11 packets transmitted, 9 received, 18.1818% packet loss, time 10166ms
rtt min/avg/max/mdev = 15.281/26.533/54.450/11.085 ms

```

Source	Destination	Protocol	Length	Info
14:1... 10.0.2.7	1.2.3.4	ICMP	98	Echo (ping) request id=0x0004, seq=1/256, ttl=64
14:1... PcsCompu_de:a2:20	Broadcast	ARP	60	Who has 10.0.2.7? Tell 10.0.2.6
14:1... PcsCompu_4c:2b:79	PcsCompu_de:a2:20	ARP	42	10.0.2.7 is at 08:00:27:4c:2b:79
14:1... 1.2.3.4	10.0.2.7	ICMP	98	Echo (ping) reply id=0x0004, seq=1/256, ttl=64
14:1... 10.0.2.7	1.2.3.4	ICMP	98	Echo (ping) request id=0x0004, seq=2/512, ttl=64
14:1... 1.2.3.4	10.0.2.7	ICMP	98	Echo (ping) reply id=0x0004, seq=2/512, ttl=64
14:1... 10.0.2.7	1.2.3.4	ICMP	98	Echo (ping) request id=0x0004, seq=3/768, ttl=64
14:1... 1.2.3.4	10.0.2.7	ICMP	98	Echo (ping) reply id=0x0004, seq=3/768, ttl=64

Frame 10: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface enp0s3, id 0
 Ethernet II, Src: PcsCompu_de:a2:20 (08:00:27:de:a2:20), Dst: PcsCompu_4c:2b:79 (08:00:27:4c:2b:79)
 Internet Protocol Version 4, Src: 1.2.3.4, Dst: 10.0.2.7
 Internet Control Message Protocol

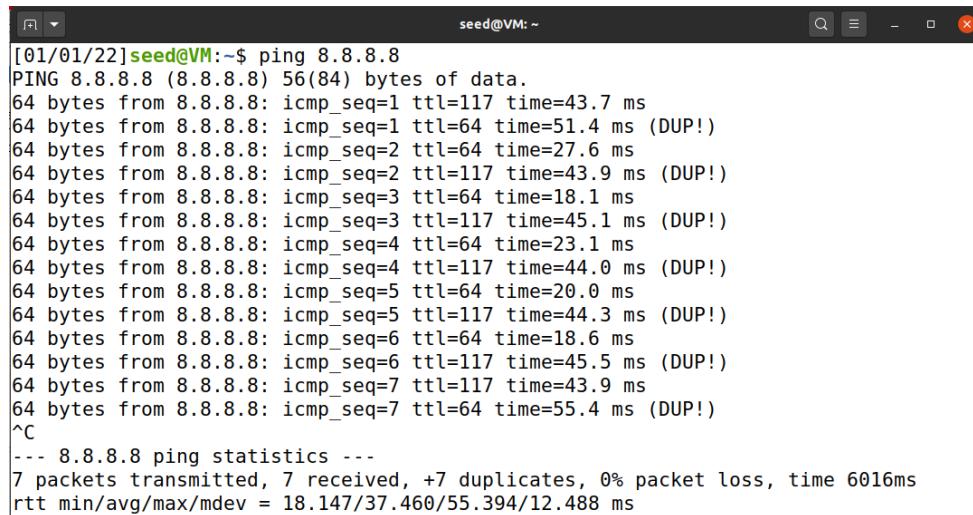
10.9.0.99: באופן דומה השתמשו באוטם מכונות וירטואליות. במקרה זה host לא קיים ב-LAN אך לא משנה כמה ping נשלח לא מקבל תשובה כי הוא בלתי ניתן להשגה.



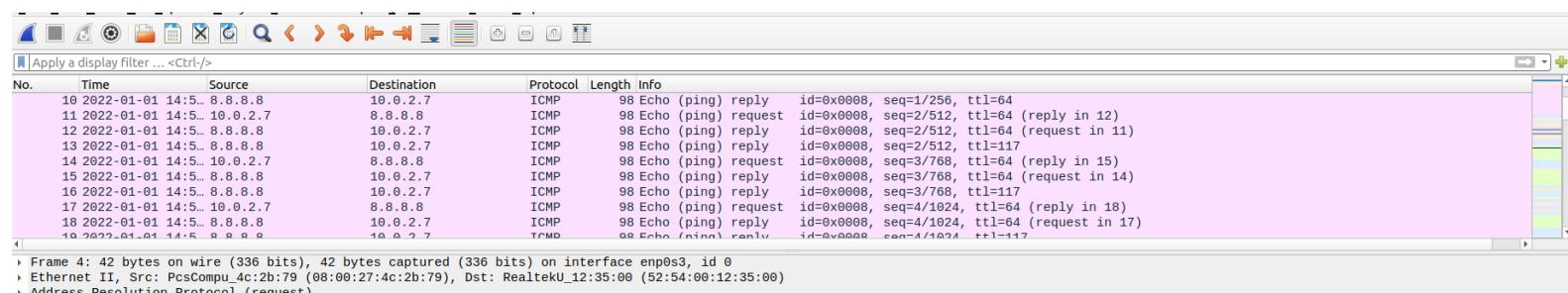
```
[01/01/22]seed@VM:~$ ping 10.9.0.99
PING 10.9.0.99 (10.9.0.99) 56(84) bytes of data.
From 10.9.0.1 icmp_seq=1 Destination Host Unreachable
From 10.9.0.1 icmp_seq=2 Destination Host Unreachable
From 10.9.0.1 icmp_seq=3 Destination Host Unreachable
From 10.9.0.1 icmp_seq=4 Destination Host Unreachable
From 10.9.0.1 icmp_seq=5 Destination Host Unreachable
From 10.9.0.1 icmp_seq=6 Destination Host Unreachable
From 10.9.0.1 icmp_seq=7 Destination Host Unreachable
From 10.9.0.1 icmp_seq=8 Destination Host Unreachable
From 10.9.0.1 icmp_seq=9 Destination Host Unreachable
^C
--- 10.9.0.99 ping statistics ---
11 packets transmitted, 0 received, +9 errors, 100% packet loss, time 10250ms
pipe 3
```

Current filter: arp						
No.	Time	Source	Destination	Protocol	Length	Info
5	2022-01-01 14:3...	PcsCompu_de:a2:20	RealtekU_12:35:00	ARP	60	Who has 10.0.2.1? Tell 10.0.2.6
6	2022-01-01 14:3...	RealtekU_12:35:00	PcsCompu_de:a2:20	ARP	60	10.0.2.1 is at 52:54:00:12:35:00

8.8.8.8: באן דומה למקרים קודמים השתמשו באוטם מכונות וירטואליות. בוגוד לכתחות הקודמות, באמצעות קיימת ברשת אינטרנט, אך לאחר הפעלתה ה-*spoofing* מהמכונה הראשונה ושליחת ping 8.8.8.8 שנקלב תשובות כפולות (*Duplicated response*) מכיוון שמקבלים תשובה גם מהשרת שלנו וגם ה-*spoofing*



```
[01/01/22]seed@VM:~$ 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=117 time=43.7 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=64 time=51.4 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=2 ttl=64 time=27.6 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=117 time=43.9 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=3 ttl=64 time=18.1 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=117 time=45.1 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=4 ttl=64 time=23.1 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=117 time=44.0 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=5 ttl=64 time=20.0 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=117 time=44.3 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=6 ttl=64 time=18.6 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=117 time=45.5 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=7 ttl=117 time=43.9 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=64 time=55.4 ms (DUP!)
^C
--- 8.8.8.8 ping statistics ---
7 packets transmitted, 7 received, +7 duplicates, 0% packet loss, time 6016ms
rtt min/avg/max/mdev = 18.147/37.460/55.394/12.488 ms
```



No.	Time	Source	Destination	Protocol	Length	Info
10	2022-01-01 14:5..	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) reply id=0x0008, seq=1/256, ttl=64
11	2022-01-01 14:5..	10.0.2.7	8.8.8.8	ICMP	98	Echo (ping) request id=0x0008, seq=2/512, ttl=64 (reply in 12)
12	2022-01-01 14:5..	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) reply id=0x0008, seq=2/512, ttl=64 (request in 11)
13	2022-01-01 14:5..	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) reply id=0x0008, seq=2/512, ttl=117
14	2022-01-01 14:5..	10.0.2.7	8.8.8.8	ICMP	98	Echo (ping) request id=0x0008, seq=3/768, ttl=64 (reply in 15)
15	2022-01-01 14:5..	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) reply id=0x0008, seq=3/768, ttl=64 (request in 14)
16	2022-01-01 14:5..	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) reply id=0x0008, seq=3/768, ttl=117
17	2022-01-01 14:5..	10.0.2.7	8.8.8.8	ICMP	98	Echo (ping) request id=0x0008, seq=4/1024, ttl=64 (reply in 18)
18	2022-01-01 14:5..	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) reply id=0x0008, seq=4/1024, ttl=64 (request in 17)
19	2022-01-01 14:5..	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) reply id=0x0008, seq=4/1024, ttl=117

Frame 4: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_4c:2b:79 (08:00:27:4c:2b:79), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
Address Resolution Protocol (request)

Task 2.1: Writing Packet Sniffing Program

Task 2.1A: Understanding How a Sniffer Works

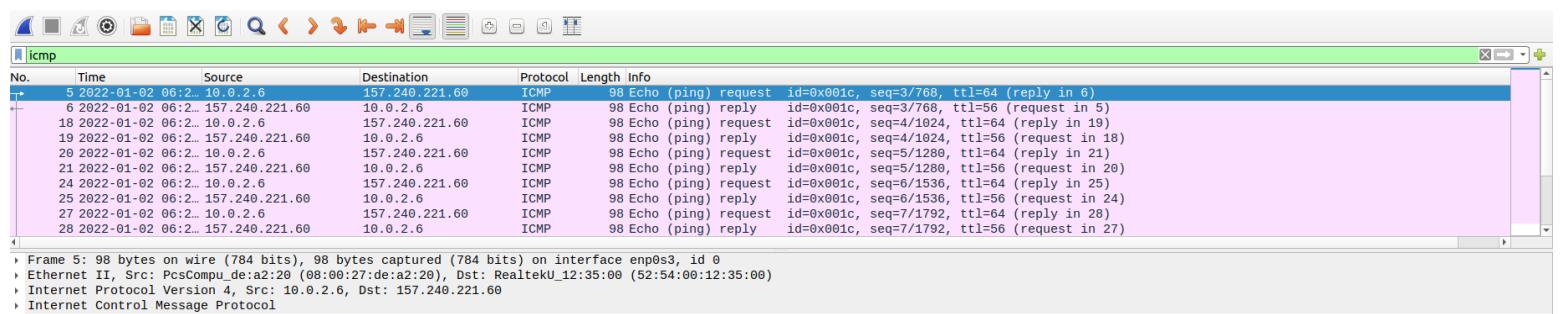
בסעיף זהה, השתמשנו בסניפר שכתבנו במתלה 5 שמסנן פקודות עם פרוטוקול ICMP. כפף שנייתן לראות בצילום מסך המכונה וירטואלית מפעילה את הסניפר **c_A.c**, ושולחים ping לכתובת IP 157.240.221.60 שהוא שיכת ל- WhatsApp אחר, והסניפר קולט את הפקודות הרלוונטיות אליו. בנוסף ניתן לראותם פקודות בוירשאך.

```
[01/02/22]seed@VM:~/Desktop$ gcc task_2_1_A.c -lpcap -o test
[01/02/22]seed@VM:~/Desktop$ sudo ./test
***** PACKET SNIFFING *****
>> PROTOCOL: ICMP
>> PACKET #0
>> SRC_IP: 157.240.221.60
>> DST_IP: 10.0.2.6

>> PROTOCOL: ICMP
>> PACKET #1
>> SRC_IP: 10.0.2.6
>> DST_IP: 157.240.221.60

>> PROTOCOL: ICMP
>> PACKET #2
>> SRC_IP: 157.240.221.60
>> DST_IP: 10.0.2.6
```

```
[01/02/22]seed@VM:~/Desktop$ ping 157.240.221.60
PING 157.240.221.60 (157.240.221.60) 56(84) bytes of data.
64 bytes from 157.240.221.60: icmp_seq=1 ttl=56 time=64.1 ms
64 bytes from 157.240.221.60: icmp_seq=2 ttl=56 time=64.3 ms
64 bytes from 157.240.221.60: icmp_seq=3 ttl=56 time=63.7 ms
64 bytes from 157.240.221.60: icmp_seq=4 ttl=56 time=73.1 ms
64 bytes from 157.240.221.60: icmp_seq=5 ttl=56 time=66.0 ms
64 bytes from 157.240.221.60: icmp_seq=6 ttl=56 time=63.1 ms
64 bytes from 157.240.221.60: icmp_seq=7 ttl=56 time=63.2 ms
64 bytes from 157.240.221.60: icmp_seq=8 ttl=56 time=64.1 ms
64 bytes from 157.240.221.60: icmp_seq=9 ttl=56 time=63.1 ms
^C
--- 157.240.221.60 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8010ms
rtt min/avg/max/mdev = 63.098/64.960/73.145/3.010 ms
```



הסנהה על ידי ספריית `pcap` משתמשת באربע פונקציות עיקריות: **Question 1**

pcap_open_live: פונקציה הזאת פותחת `device` כדי לתחוף את הפקות, היא מקבלת ארבעה ארגומנטים, הראשון הוא השם של הממשק השני הוא המספר המקסימלי של בתים לטאיפיסת פקודות, השלישי מציין שהממשק יעבור למצוב מוקרת הרביעי נותן את פסק הזמן לקריאה באלפיות שניות והאחרון מחזיר שגיאה ומוגדר רק כאשר `pcap_open_live` נכשלת.

pcap_compile: הפעקציה הזאת תיצור את הפילטר שלנו עבור הסניפר, היא מורכבת מארבעה פרמטרים. הראשון מצביע על תיאור של פקטה שנפתחה שהוחזר מ- `pcap_open_live`. השני מצביע על מבנה `bpf_program` אשר ישלים על ידי `pcap_compile`. השלישי מכיל את פרמטר המסנן, הרביעי שולט על מבצעים אופטימייזיה בקוד המתקבל ואחרון מציין את `netmask` של הרשות.

- **pcap_setfilter**: פונקציה זו מפעילה את המסן שיצרנו ב-*pcap_compile*. היא קודם מקבלת מצביע על תיאור של פקטה שנטרפה שהוחזר מ-*pcap_open_live* וגם מצביע על מבנה *.pcap_compile bpf_program*.

- **pcap_loop**: פונקציה זו מבצעת את התהיליך של תפיסת פקטות. יש לה מצביע על תיאור של פקטה שנטרפה שהוחזר מ-*pcap_open_live*, זה ישמש לאחסון נתונים. הארגומנט השני מצביע את המספר המקסימלי של פקטות לעיבוד לפני ההחזרה, השלישי הוא פונקציית *callback* לטיפול בחבילה ואחרון מצביע את הארגומנט הראשון שייעבור *callback*.

Question 2: ספריית *pcap* צריכה לגשת למשק של הרשת, היא משתמשת ב-*raw socket*, בנוסף התהיליך צריך לróż ב-*promiscuous mode* لكن ניתן לבצע את הפעולות האלה אך ורק ב-*root privilege*. אם ננסה להריץ את התוכנית בלי ה-*root privilege* התוכנית תזורך שגיאה.

Question 3: מצב מופקר (*promiscuous mode*) מאפשר לסניפר של הרשת לגשת למשק ולנתונים להעיבר את כל התעבורה מהרשות ולא רק את התעבורה שהרשות הייתה אמורה לקבל. המצב מופקר מוגדר בפרמטר השלישי של הפונקציה *pcap_open_live*.

כאשר מתחילה את הארגומנט ל-1 יש לסניפר גישה לכל התעבורה ומתקבל חבילות. כאשר אנחנו מגדירים את הפרמטר ל-0 ומעבירים את המכונה במצב Deny בהגדרות, הסניפר לא יכול פקטות.

Task 2.1B: Writing Filters

:ICMP.

השתמשנו בקוד מהסעיף הקודם כדי להסניף פקטות ICMP, שינו את הפילטר כי אנחנו צריכים להסניף פקטות בין שני host ספציפיים.

Char filter_exp[] = "icmp and src host 10.2.0.6 and dst host 8.8.8.8"

הרכזנו את הקוד *task_2_1_B_ICMP.c* מהמכונה השניה.

```
seed@VM: ~/Desktop          seed@VM: ~/Desktop
[01/02/22]seed@VM:~/Desktop$ gcc task_2_1_B_ICMP.c -lpcap -o test
[01/02/22]seed@VM:~/Desktop$ sudo ./test
***** PACKET SNIFFING *****
>> PROTOCOL: ICMP
>> PACKET #0
>> SRC_IP: 10.0.2.7
>> DST_IP: 8.8.8.8

>> PROTOCOL: ICMP
>> PACKET #1
>> SRC_IP: 8.8.8.8
>> DST_IP: 10.0.2.7

>> PROTOCOL: ICMP
>> PACKET #2
>> SRC_IP: 10.0.2.7
>> DST_IP: 8.8.8.8

>> PROTOCOL: ICMP
>> PACKET #3
>> SRC_IP: 8.8.8.8
>> DST_IP: 10.0.2.7
```

```
[01/02/22]seed@VM:~$ 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=115 time=48.9 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=67.4 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=49.8 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=115 time=43.8 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=115 time=57.9 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=115 time=46.9 ms
```

No.	Time	Source	Destination	Protocol	Length	Info
1	2022-01-02 09:3...	10.0.2.7	8.8.8.8	ICMP	98	Echo (ping) request id=0x000e, seq=1/256, ttl=64 (reply in progress)
2	2022-01-02 09:3...	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) reply id=0x000e, seq=1/256, ttl=115 (request in progress)
3	2022-01-02 09:3...	10.0.2.7	8.8.8.8	ICMP	98	Echo (ping) request id=0x000e, seq=2/512, ttl=64 (reply in progress)
4	2022-01-02 09:3...	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) reply id=0x000e, seq=2/512, ttl=115 (request in progress)
5	2022-01-02 09:3...	10.0.2.7	8.8.8.8	ICMP	98	Echo (ping) request id=0x000e, seq=3/768, ttl=64 (reply in progress)
6	2022-01-02 09:3...	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) reply id=0x000e, seq=3/768, ttl=115 (request in progress)
7	2022-01-02 09:3...	10.0.2.7	8.8.8.8	ICMP	98	Echo (ping) request id=0x000e, seq=4/1024, ttl=64 (reply in progress)
8	2022-01-02 09:3...	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) reply id=0x000e, seq=4/1024, ttl=115 (request in progress)

Frame 10: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface enp0s3, id 0
 Ethernet II, Src: RealtekU_12:35:00 (52:54:00:12:35:00), Dst: PcsCompu_4c:2b:79 (08:00:27:4c:2b:79)
 Internet Protocol Version 4, Src: 8.8.8.8, Dst: 10.0.2.7
 Internet Control Message Protocol

השתמשנו בקוד מהסעיף הקודם כדי להסניף פקודות TCP, שינוינו את הפילטר כי אנחנו צריכים להסניף פקודות TCP עם פורט יעד בטווח 10-100

Char filter_exp[] = "tcp and portrange 10-100"

הרכזנו את הקוד **c** מהמכונה וירטואלית **10.2.0.6** ושלחנו **telnet 8.8.8.8** ופעם ראשונה עברו פורט 50 (יש תעבורת) ופעם שנייה עברו פורט 110.(אין תעבורת)



```

seed@VM: ~/Desktop
[01/02/22] seed@VM:~/Desktop$ gcc task_2_1_B_TCP.c -lpcap -o test2
[01/02/22] seed@VM:~/Desktop$ sudo ./test2
***** PACKET SNIFFING *****
>> PROTOCOL: TCP
>> PACKET #0
>> SRC_IP: 10.0.2.6
>> DST_IP: 8.8.8.8

>> PROTOCOL: TCP
>> PACKET #1
>> SRC_IP: 10.0.2.6
>> DST_IP: 8.8.8.8

>> PROTOCOL: TCP
>> PACKET #2
>> SRC_IP: 10.0.2.6
>> DST_IP: 8.8.8.8

^Z
[22]+ Stopped sudo ./test2
[01/02/22] seed@VM:~/Desktop$ gcc task_2_1_B_TCP.c -lpcap -o test2
[01/02/22] seed@VM:~/Desktop$ sudo ./test2
***** PACKET SNIFFING *****

^Z
[23]+ Stopped sudo ./test2
[01/02/22] seed@VM:~/Desktop$ [REDACTED]

```

The terminal on the right shows a telnet session starting at port 50 and then connecting to port 110.

Task 2.1C: Sniffing Passwords

השתמשנו בקוד להסניף פקודות TCP אך ביצענו כמה שינויים והוסף על מנת לתפוס את הסיסמה (**dees**) של מכונה וירטואלית שאנו חובדים בה. קודם כל בפילטר השארם את:

char filter_exp[] = "proto TCP and dst portrange 10-100"

בנוסף הוספנו לתוכנית **task_2_C.c** את ה-**TCP header** עם כל הפרמטרים שלו וצמו על לולאה על **payload** כדי להסניף את התעבורת של **telnet** והנתונים שנשלחו מהמכונה **Telnet.10.0.2.7** בaczם מפצל את הסיסמה שמוכנת בנתונים ושולח אותם בספר פקודות אחד לכל תוו מהסיסמה.

The image shows two terminal windows. The left window displays network traffic analysis output, including source and destination IP addresses and ports. The right window shows a telnet session connected to port 10.0.2.7, displaying a welcome message from an Ubuntu 20.04.1 LTS server, system documentation links, and update information.

```

.....4.(...
>> PROTOCOL: TCP
>> IP SRC: 10.0.2.6
>> SRC PORT: 53952
>> IP DST: 10.0.2.7
>> DST PORT: 23
.....6;(...d
>> PROTOCOL: TCP
>> IP SRC: 10.0.2.6
>> SRC PORT: 53952
>> IP DST: 10.0.2.7
>> DST PORT: 23
.....7.(...e
>> PROTOCOL: TCP
>> IP SRC: 10.0.2.6
>> SRC PORT: 53952
>> IP DST: 10.0.2.7
>> DST PORT: 23
.....7.(...e
>> PROTOCOL: TCP
>> IP SRC: 10.0.2.6
>> SRC PORT: 53952
>> IP DST: 10.0.2.7
>> DST PORT: 23
.....8.(...s
>> PROTOCOL: TCP
>> IP SRC: 10.0.2.6
>> SRC PORT: 53952
>> IP DST: 10.0.2.7

[01/02/22]seed@VM:~$ telnet 10.0.2.7
Trying 10.0.2.7...
Connected to 10.0.2.7.
Escape character is '^].
Ubuntu 20.04.1 LTS
VM login: seed
Password:
Welcome to Ubuntu 20.04.1 LTS (GNU/Linux 5.4.0-54-generic x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

9 updates can be installed immediately.
9 of these updates are security updates.

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

```

Task 2.2: Spoofing

Task 2.2.A: Write a spoofing program

כפי שניתן לראות בצילומי מסך, ההרכנו את הקוד **task_2_2_A.c** שעשוה מהמכונה ירטואליית *spoofed* שלנו 10.0.2.6 לשרת של גугл 8.8.8.8, ניתן לראות בוירשאך שהתוכנית שולחה את הפקטה

```

void send_raw_ip_packet(struct ipheader* ip) {
    struct sockaddr_in dest_info;
    int enable = 1;
    //Step1: Create a raw network socket
    int sock = socket(AF_INET, SOCK_RAW, IPPROTO_RAW);

    //Step2: Set Socket option
    setsockopt(sock, IPPROTO_IP, IP_HDRINCL, &enable, sizeof(enable));

    //Step3: Provide destination information
    dest_info.sin_family = AF_INET;
    dest_info.sin_addr = ip->iph_destip;

    //Step4: Send the packet out
    printf("***** SENDING SPOOFED PACKET *****\n");
    if (sendto(sock, ip, ntohs(ip->iph_len), 0, (struct sockaddr *)&dest_info, sizeof(dest_info)) < 0) {
        printf("%sError: failed sending message !", "");
    }
    else{

        printf("\t>> IP SOURCE: %s\n", inet_ntoa(ip->iph_sourceip));
        printf("\t>> IP DEST: %s\n", inet_ntoa(ip->iph_destip));
        printf("\n");
    }
    close(sock);
}

```

```

int main() {

    char buffer[PACKET_LEN];
    memset(buffer, 0, PACKET_LEN);

    // Fill in the ICMP header
    struct icmpheader *icmp = (struct icmpheader *) (buffer + sizeof(struct ipheader));

    // ICMP type 8 for request and 0 for replay
    icmp->icmp_type = 8;

    // Calculate checksum
    icmp->icmp_chksum = 0;
    icmp->icmp_chksum = in_cksum((unsigned short *)icmp, sizeof(struct icmpheader));

    //Fill in the IP header
    struct ipheader *ip = (struct ipheader *) buffer;
    ip->iph_ver = 4; [01/04/22] seed@VM:~/Desktop$ sudo ./test
    ip->iph_ihl = 5; ***** SENDING SPOOFED PACKET *****
    ip->iph_tos = 16; >> IP SOURCE: 8.8.8.8
    ip->iph_ttl = 128; >> IP DEST: 10.0.2.6
    ip->iph_sourceip.s_addr = inet_addr("8.8.8.8");
    ip->iph_destip.s_addr = inet_addr("10.0.2.6");
    ip->iph_protocol = IPPROTO_ICMP;
    ip->iph_len = htons(sizeof(struct ipheader) + sizeof(struct icmpheader));

    send_raw_ip_packet (ip);

    return 0;
}

```

Apply a display filter ...<Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
7	2022-01-04 14:4...	8.8.8.8	10.0.2.6	ICMP	98	Echo (ping) reply id=0x000e, seq=1/256, ttl=117 (request in 6)
8	2022-01-04 14:4...	10.0.2.6	8.8.8.8	ICMP	98	Echo (ping) request id=0x000e, seq=2/512, ttl=64 (reply in 9)

Task 2.2B: Spoof an ICMP Echo Request.

הקוד שכתבנו בסעיף הקודם הוא spoof של Echo Request עם הצלומי מסך המצורפים.

Question 4

כן, ניתן להגדיר את אורך של ה-*header* IP לערך שרירותי ללא קשר לגודל החבילה האמיתית. בסופו של דבר גודל חבילה משתנה לגודל האמיתית לא משנה מה שינוי המתכונת.

Question 5

לא, אין צורך, מערכת הפעלה מחשבת בלבד שדה זה.

Question 6

על מנת שכל תוכנית תקבל גישה לחומרה ותוכל לעבור למצב מופקר חיבים להשתמש ב- *root privilege* אם נרצה להריץ תוכניות ב- *raw socket*. בנוסף ברגע שיוציאים *raw socket* אפשר לבחר אויה פרוט שורצים אבל יש חוק ברשותות שאי אפשר לעשות *bind* לפורט שהוא פחות 1024 בלי הרשות *root* ולכן במקרה זהה בغالbisicooui יכול לבחר פורט נמוך יותר ונבקש הרשות *root*. כמובן במקרה שאין הרשות תחסם הגישה ותיזיק גישה ב*bind*.

4.3 Task 2.3: Sniff and then Spoof

בסעיף זהה, כתבו תוכנית בשם task_2_3.c שמבצעת גם sniffing וגם spoofing על הפקודות. השתמשנו בני מכונות וירטואליות הראשונה עם כתובת IP 10.0.2.6 שמננה מריצים את התוכנית, כפי שנitinן לראות על הצלום מסך, מופיעות על הטרמינל פקודות אחריו -sniff ו-sniff dns ל-Shell DNS של גוג'ל מקור IP, כתובות יעד IP ופרוטוקול. עם המכונה השנייה אנחנו שולחים ping לשרת DNS של גוג'ל 8.8.8.8. על ידי הווירשאך נוכל לראות את כל הפקודות ICMP שנשלחו והתקבלו.

```
seed@VM: ~/Desktop
```

```
[01/05/22]seed@VM:~/Desktop$ gcc task_2_3.c -lpcap -o test
```

```
[01/05/22]seed@VM:~/Desktop$ sudo ./test
***** SNIFFING PACKET *****
```

```
>> IP SRC:10.0.2.7
>> IP DST:8.8.8.8
>> PROTOCOL: ICMP
```

```
***** SENDING SPOOFED PACKET *****
```

```
>> IP SRC:8.8.8.8
>> IP DST:10.0.2.7
>> PROTOCOL: ICMP
```

```
***** SNIFFING PACKET *****
```

```
>> IP SRC:8.8.8.8
>> IP DST:10.0.2.7
>> PROTOCOL: ICMP
```

```
***** SENDING SPOOFED PACKET *****
```

```
>> IP SRC:10.0.2.7
>> IP DST:8.8.8.8
```

```
***** SNIFFING PACKET *****
```

```
>> IP SRC:8.8.8.8
>> IP DST:10.0.2.7
>> PROTOCOL: ICMP
```

```
[01/05/22]seed@VM:~/Desktop$ 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=115 time=48.6 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=59.6 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=1191 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=1192 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=1240 ms (DUP!)
```

```
***** SNIFFING PACKET *****
```

```
[01/05/22]seed@VM:~/Desktop$ 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=115 time=48.6 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=59.6 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=1191 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=1 ttl=128 time=1192 ms (DUP!)
64 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=1240 ms (DUP!)
```

No.	Time	Source	Destination	Protocol	Length	Info
70	2022-01-05 05:2...	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) reply id=0x000d, seq=4/1024, ttl=115
71	2022-01-05 05:2...	PcsCompu_4c:2b:79	RealtekU_12:35:00	ARP	42	Who has 10.0.2.1? Tell 10.0.2.7
72	2022-01-05 05:2...	RealtekU_12:35:00	PcsCompu_4c:2b:79	ARP	60	10.0.2.1 is at 52:54:00:12:35:00
73	2022-01-05 05:2...	10.0.2.7	8.8.8.8	ICMP	98	Echo (ping) request id=0x000d, seq=1/256, ttl=128 (reply)
74	2022-01-05 05:2...	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) reply id=0x000d, seq=1/256, ttl=128 (request)
75	2022-01-05 05:2...	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) reply id=0x000d, seq=1/256, ttl=128
76	2022-01-05 05:2...	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) reply id=0x000d, seq=1/256, ttl=128
77	2022-01-05 05:2...	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) reply id=0x000d, seq=1/256, ttl=128
78	2022-01-05 05:2...	8.8.8.8	10.0.2.7	ICMP	98	Echo (ping) request id=0x000d, seq=2/512, ttl=128 (reply)
79	2022-01-05 05:2...	10.0.2.7	8.8.8.8	ICMP	98	Echo (ping) reply id=0x000d, seq=2/512, ttl=128

Frame 71: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface enp0s3, id 0
Ethernet II, Src: PcsCompu_4c:2b:79 (08:00:27:4c:2b:79), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
Address Resolution Protocol (request)