

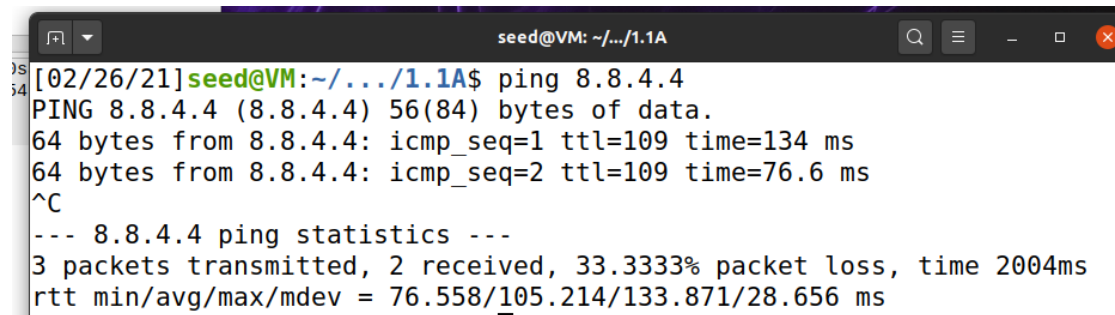
מטלת גמר

שאלה 1:

A.1:

א: תחילה נראה את הרצת קוד ה"sniffing" באמצעות `sudo` ונראה שהוא אכן פועל. (ניתן לראות בצילומים למטה)

Ping:

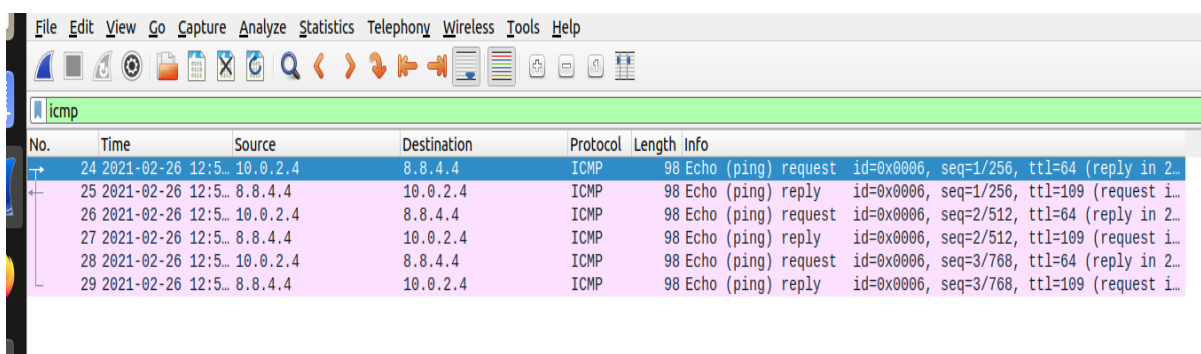
A screenshot of a terminal window with a dark background. The title bar shows 'seed@VM: ~/.../1.1A'. The terminal content shows a ping command being executed and its output. The output indicates that 3 packets were transmitted, 2 were received, and there was a 33.3333% packet loss. The round-trip times (rtt) are listed as 76.558, 105.214, and 133.871 ms.

```
[02/26/21]seed@VM:~/.../1.1A$ ping 8.8.4.4
PING 8.8.4.4 (8.8.4.4) 56(84) bytes of data.
64 bytes from 8.8.4.4: icmp_seq=1 ttl=109 time=134 ms
64 bytes from 8.8.4.4: icmp_seq=2 ttl=109 time=76.6 ms
^C
--- 8.8.4.4 ping statistics ---
3 packets transmitted, 2 received, 33.3333% packet loss, time 2004ms
rtt min/avg/max/mdev = 76.558/105.214/133.871/28.656 ms
```

תפיסת הפקטה:

```
[02/26/21]seed@VM:~/.../1.1A$ sudo python3 sniffer.py
###[ Ethernet ]###
dst      = 52:54:00:12:35:00
src      = 08:00:27:a1:65:99
type     = IPv4
###[ IP ]###
version  = 4
ihl      = 5
tos      = 0x0
len      = 84
id       = 35212
flags    = DF
frag     = 0
ttl      = 64
proto    = icmp
chksum   = 0x990d
src      = 10.0.2.4
dst      = 8.8.4.4
\options \
###[ ICMP ]###
type     = echo-request
code     = 0
chksum   = 0x6423
id       = 0x6
seq      = 0x1
```

צילומי WireShark ל ping הנ"ל:



The screenshot shows the Wireshark interface with the 'icmp' filter applied. The packet list pane displays five packets (24-29) showing a sequence of ping requests and replies. The packet details pane shows the structure of an ICMP Echo (ping) request and reply, including fields like ID, sequence number, and TTL.

No.	Time	Source	Destination	Protocol	Length	Info
24	2021-02-26 12:5...	10.0.2.4	8.8.4.4	ICMP	98	Echo (ping) request id=0x0006, seq=1/256, ttl=64 (reply in 2...
25	2021-02-26 12:5...	8.8.4.4	10.0.2.4	ICMP	98	Echo (ping) reply id=0x0006, seq=1/256, ttl=109 (request i...
26	2021-02-26 12:5...	10.0.2.4	8.8.4.4	ICMP	98	Echo (ping) request id=0x0006, seq=2/512, ttl=64 (reply in 2...
27	2021-02-26 12:5...	8.8.4.4	10.0.2.4	ICMP	98	Echo (ping) reply id=0x0006, seq=2/512, ttl=109 (request i...
28	2021-02-26 12:5...	10.0.2.4	8.8.4.4	ICMP	98	Echo (ping) request id=0x0006, seq=3/768, ttl=64 (reply in 2...
29	2021-02-26 12:5...	8.8.4.4	10.0.2.4	ICMP	98	Echo (ping) reply id=0x0006, seq=3/768, ttl=109 (request i...

ב: נראה שכשנריץ את תוכנת ה-"sniffer" ללא הפקודה sudo נקבל תקלה והתוכנה לא תעבוד, ניתן לראות הוכחה לכך בצילום המצורף למטה.

```
[02/26/21]seed@VM:~/.../1.1A$ python3 sniffer.py
Traceback (most recent call last):
  File "sniffer.py", line 6, in <module>
    pkt = sniff(iface='enp0s3', filter='icmp', prn=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
[02/26/21]seed@VM:~/.../1.1A$
```

הסבר:

נתבונן בתמונה המצורפת למעלה אשר מתארת לנו את התקלה הנוצרת כאשר אנו מריצים את תוכנת ה-"sniffing" בלי פקודת sudo.

נסתכל על החץ העליון ונשים לב שאנו מקבלים שגיאה כאשר אנו משתמשים ב raw socket וזה מכיוון שכאשר אנו רוצים להשתמש ב raw socket נצטרך לקבל הרשאות והרשאות אלו לא קיימות לנו ללא פקודת sudo.

לכן אם נריץ תוכנה זאת עם פקודת sudo היא תעבוד ואם נריץ ללא פקודה זו היא לא תעבוד.

B.1:

בשאלה זו קיבלנו שלוש משימות:

משימה א: לתפוס פקטות של ICMP: והתשובה לזה בסעיף ב-A1.

משימה ב: לתפוס פקטות של TCP שבאים מ IP אקראי ועם destport 23:

מה שעשינו בקוד זה נעזרנו בקוד של סעיף קודם אך רק שינינו לו את ה-filter כך שנקבל רק פקטות של TCP מה ip שבחרנו ומ-destport 23.

ועל מנת להראות את נכונות התוכנה זייפנו הודעה עם הפרטים המתבקשים.

משימה ג': לתפוס פקטות אשר באות או נכנסות ל subnet אקראי:

בתוכנה זאת נעזרנו בקוד של סעיף א אך תחילה הגדרנו את ה-subnet שאנו רוצים להקשיב
לה ואז יצרנו פילטר חדש אשר מכיל את כל כתובות ה-subnet, לאחר מכן הזנו בפילטר של
פונקצית sniffer את הפילטר שיצרנו.

א- Capture only the ICMP packet: תשובה לזה היא סעיף A.1
הרצת התוכנה באמצעות sudo

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

על מנת לבדוק עם הקוד עובד זייפנו הודעה להלן תמונות
של נכונות התוכנה:

```
[02/26/21] seed@VM: ~/.../1.1B$ sudo python3 TCP_mas.py  
.  
Sent 1 packets.
```

שליחת
ההודעה

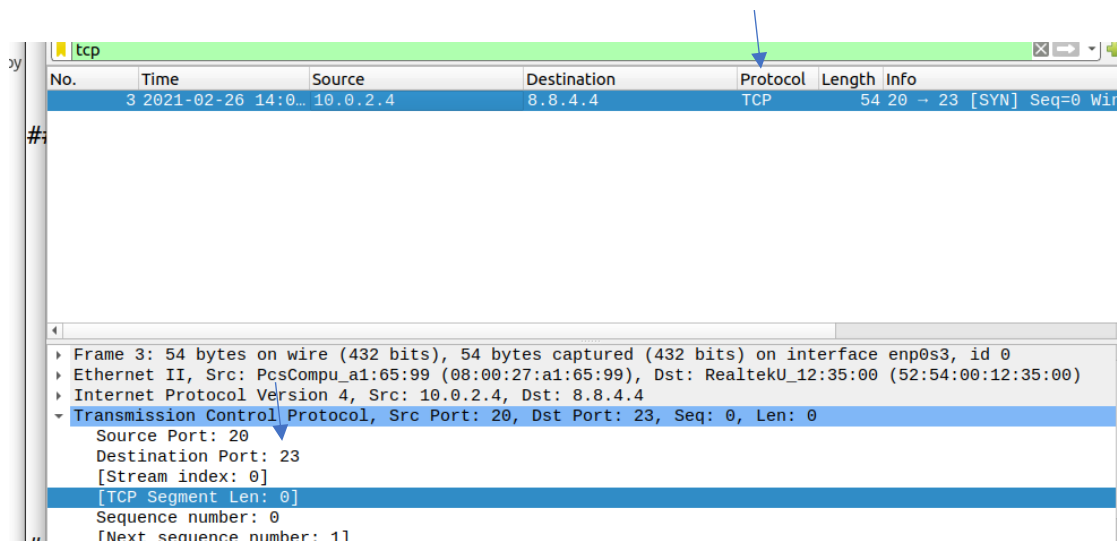
```
###[ Ethernet ]###  
  dst      = 52:54:00:12:35:00  
  src      = 08:00:27:a1:65:99  
  type     = IPv4
```

קבלת הפקטה

```
###[ IP ]###  
  version  = 4  
  ihl      = 5  
  tos      = 0x0  
  len      = 40  
  id       = 1  
  flags    =  
  frag     = 0  
  ttl      = 64  
  proto    = tcp  
  chksum   = 0x62c0  
  src      = 10.0.2.4  
  dst      = 8.8.4.4  
  \options \
```

```
###[ TCP ]###  
  sport    = ftp_data  
  dport    = telnet  
  seq      = 0  
  ack      = 0  
  dataofs  = 5  
  reserved = 0  
  flags    = S  
  window   = 8192  
  chksum   = 0x77a8  
  urgptr   = 0
```

:WhireShark



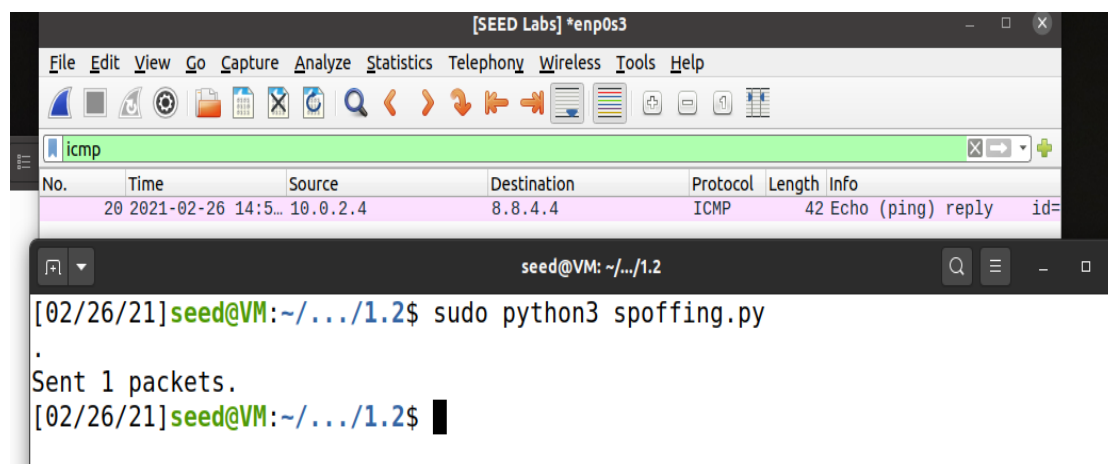
נשים לב שהפרוטוקול הינו TCP וה-port הוא אכן 23.

ג-Capture packets comes from or to go to a particular subnet:

להלן נכונות הקוד: נשים לב שתמונה הראשונה אנו מאזינים
 לכתובת 8.8.4.4 ובתמונה השנייה אנו מאזינים לכתובת
 8.8.4.3

וזו כאשר אנו מריצים את אותה תוכנה.

השינויים שנעשו בקוד הם הגדרת a.src ו-a.dst ולאחר שהגדרנו את הכתובת הקוד הנ"ל אכן יעשה הודעות spoofing נכונות הקוד בתמונה שמצורפת:
נשים לב שקיבלנו אך ורק reply



1.3: בשאלה זאת התבקשנו להחזיר מה ראוטרים עברנו עד הגעתנו לכתובת.

מה שעשינו בקוד זה שלחנו את ההודעה והגדרנו לה בשליחה ttl וגם timeout ,

ttl - התחלתי שהגדרנו הוא אחד אם לא המסלול שלנו יהיה ארוך מאחד אז נקבל שגיאה וה ans שנקבל יהיה שווה לnull , במקרה זה נקדם את ה-ttl באחד ונעשה זאת עד שנקבל את ה ttl המתבקש ונדפיס אותו.

יכולה להיבצר לנו תקלה נוספת שכשנשלח את ההודעה עם ttl מסוים ולא נקבל תשובה זאת אומרת נכנס ל"לולאה אין סופי"

במקרה זה נעזר ב- time out .

Time out: הגדרנו להיות את time out להיות שניה זאת
אומרת אם לא קיבלנו תשובה עד שניה משליחת ההודעה
נקבל שגיאה ואז נעשה את כל התהליך שוב.

נכונות הקוד:

נשים לב שב-wireShark נקבל שהttl הוא 18 וגם בתוכנה
שכתבנו.

icmp						
No.	Time	Source	Destination	Protocol	Length	Info
3	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=1 (no response f...
4	2021-02-26 15:3...	10.0.2.4	10.0.2.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
5	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=2 (no response f...
6	2021-02-26 15:3...	192.168.1.1	10.0.2.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
7	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=3 (no response f...
8	2021-02-26 15:3...	10.162.40.1	10.0.2.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
9	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=4 (no response f...
10	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=5 (no response f...
11	2021-02-26 15:3...	185.149.252.109	10.0.2.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
12	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=6 (no response f...
13	2021-02-26 15:3...	10.141.226.60	10.0.2.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
14	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=7 (no response f...
15	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=8 (no response f...
16	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=9 (no response f...
17	2021-02-26 15:3...	10.185.180.11	10.0.2.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
18	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=10 (no response ...
19	2021-02-26 15:3...	10.141.226.81	10.0.2.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
20	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=11 (no response ...
21	2021-02-26 15:3...	84.110.49.250	10.0.2.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
22	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=12 (no response ...
23	2021-02-26 15:3...	62.219.189.218	10.0.2.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
24	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=13 (no response ...
25	2021-02-26 15:3...	212.179.124.86	10.0.2.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
26	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=14 (no response ...
27	2021-02-26 15:3...	10.250.99.2	10.0.2.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
28	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=15 (no response ...
29	2021-02-26 15:3...	212.25.70.69	10.0.2.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
30	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=16 (no response ...
31	2021-02-26 15:3...	108.170.225.31	10.0.2.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
32	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=17 (no response ...
33	2021-02-26 15:3...	142.250.46.99	10.0.2.4	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
34	2021-02-26 15:3...	10.0.2.4	8.8.4.4	ICMP	42	Echo (ping) request id=0x0000, seq=0/0, ttl=18 (reply in 35)

```
Finished sending 1 packets.  
*  
Received 1 packets, got 1 answers, remaining 0 packets  
Begin emission:  
Finished sending 1 packets.  
*  
Received 1 packets, got 1 answers, remaining 0 packets  
Begin emission:  
Finished sending 1 packets.  
*  
Received 1 packets, got 1 answers, remaining 0 packets  
Begin emission:  
Finished sending 1 packets.  
*  
Received 1 packets, got 1 answers, remaining 0 packets  
Begin emission:  
Finished sending 1 packets.  
*  
Received 1 packets, got 1 answers, remaining 0 packets  
Begin emission:  
Finished sending 1 packets.  
*  
Received 1 packets, got 1 answers, remaining 0 packets  
Begin emission:  
Finished sending 1 packets.  
*  
Received 1 packets, got 1 answers, remaining 0 packets  
Begin emission:  
Finished sending 1 packets.  
*  
Received 1 packets, got 1 answers, remaining 0 packets  
ttl is: 18  
—
```

1.4: בשאלה זו התבקשנו לבנות עוד מכונה אשר תהייה באותו lan שלנו ולהאזין למכונה זו וכאשר היא שולחת הודעת פינג לכתובת כל שהיא אני אענה לה מהמכונה השנייה. בין אם המכונה קיימת או לא,

תחילה מצאנו את ה interface של מכונה ולאחר מכן הגדרנו לתוכנה שלנו לאילו interfaces להאזין ואם וכשקיבלנו שם הודעה שלחנו אותה לפונקציה שבנינו שפונקציה זאת שולחת הודעת reply בחזרה לאותה מכונה שהוא קיבל ממנה את הודעת request

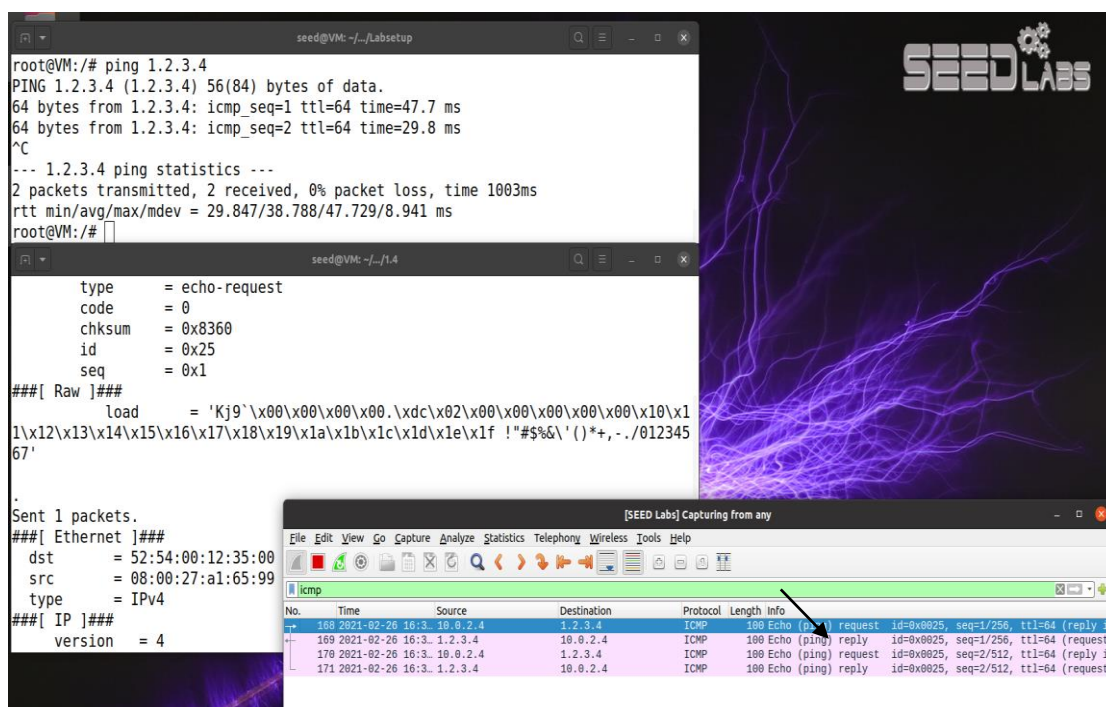
הוכחת נכונות התוכנה מצורפת בתמונות:

כתובת ראשונה:

1.2.3.4

כתובת אשר לא קיימת באינטרנט!

נשים לב כי שמכונה קיבלה הודעת reply אחת שהיא ההודעה שאנחנו שלחנו.



The screenshot displays two windows from a virtual machine environment. The top window is a terminal with the following output:

```
root@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=47.7 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=64 time=29.8 ms
^C
--- 1.2.3.4 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 29.847/38.788/47.729/8.941 ms
root@VM:/#
```

The bottom window shows a packet capture interface with the following table:

No.	Time	Source	Destination	Protocol	Length	Info
168	2021-02-26 16:31:10.024	10.0.2.4	1.2.3.4	ICMP	100	Echo (ping) request id=0x0025, seq=1/256, ttl=64 (reply in 169)
169	2021-02-26 16:31:10.024	1.2.3.4	10.0.2.4	ICMP	100	Echo (ping) reply id=0x0025, seq=1/256, ttl=64 (request in 170)
170	2021-02-26 16:31:10.024	10.0.2.4	1.2.3.4	ICMP	100	Echo (ping) request id=0x0025, seq=2/512, ttl=64 (reply in 171)
171	2021-02-26 16:31:10.024	1.2.3.4	10.0.2.4	ICMP	100	Echo (ping) reply id=0x0025, seq=2/512, ttl=64 (request in 172)

כתובת שנייה: 10.9.0.99 כתובת אשר לא קיימת ב-lan.

מכיוון שאנו לא מכירים את כתובת זו נשלח הודעת arp ומכיוון שכתובת זאת לא נמצאת ב-lan לא נקבל מאף אחד הודעה חזר שמודיע לנו איפה כתובת זו נמצאת ולכן לא נוציא פינג ומכיוון שלא הוצאנו פינג אז המכונה השנייה שמאזינה למכונה זו לא תחזיר לה הודעת request
ניתן לראות זאת בתמונה המצורפת:

```

root@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
^C

```

No.	Time	Source	Destination	Protocol	Length	Info
2	2021-02-26 17:1...	fe80::42:9bff:fe45:...	ff02::fb	MDNS	205	Standard query 0x0000 PTR _nfs._tcp.local, "QM" quest
3	2021-02-26 17:1...	fe80::9068:7bff:feb...	ff02::fb	MDNS	205	Standard query 0x0000 PTR _nfs._tcp.local, "QM" quest
4	2021-02-26 17:1...	02:42:9b:45:43:b7		ARP	44	Who has 10.9.0.99? Tell 10.9.0.1
5	2021-02-26 17:1...	02:42:9b:45:43:b7		ARP	44	Who has 10.9.0.99? Tell 10.9.0.1
6	2021-02-26 17:1...	02:42:9b:45:43:b7		ARP	44	Who has 10.9.0.99? Tell 10.9.0.1
7	2021-02-26 17:1...	02:42:9b:45:43:b7		ARP	44	Who has 10.9.0.99? Tell 10.9.0.1
8	2021-02-26 17:1...	02:42:9b:45:43:b7		ARP	44	Who has 10.9.0.99? Tell 10.9.0.1
9	2021-02-26 17:1...	02:42:9b:45:43:b7		ARP	44	Who has 10.9.0.99? Tell 10.9.0.1
10	2021-02-26 17:1...	02:42:9b:45:43:b7		ARP	44	Who has 10.9.0.99? Tell 10.9.0.1
11	2021-02-26 17:1...	02:42:9b:45:43:b7		ARP	44	Who has 10.9.0.99? Tell 10.9.0.1
12	2021-02-26 17:1...	10.9.0.1	10.9.0.1	ICMP	128	Destination unreachable (Host unreachable)
13	2021-02-26 17:1...	10.9.0.1	10.9.0.1	ICMP	128	Destination unreachable (Host unreachable)
14	2021-02-26 17:1...	10.9.0.1	10.9.0.1	ICMP	128	Destination unreachable (Host unreachable)

כתובת שלישית: 8.8.8.8

כתובת אשר קיימת ומכיוון שכתובת זו קיימת אנו נצטרך לקבל שני הודעות reply אחת מהאתר עצמו והשנייה

שאלה 1: בסניפר שכתבנו השתמשנו שלוש פעמים בספריית pcap פעם ראשונה בפונקציה pcap_open_live

אחת מהארגומנטים שהפונקציה מקבל היא האינטר פייס שאנו רוצים להאזין לו ובעצם אנו משתמשים בפונקציה זאת על מנת להאזין לאינטר פייס שהכנסו לפונקציה והיא פותחת אותנו להאזנה, נשים לב שלפונקציה זאת יש flag אשר פותח אותנו לפרמיסקיס מוד (off-0 on-1).

פעם שנייה שהשתמשנו בספריית pcap היא בפונקציה pcap_compile פונקציה זאת בעצם מקמפת לנו את הפילטר שאנו רוצים להאזין לו כגון icmp,tcp וכו

פעם שלישית שהשתמשנו בספריית pcap היא בפונקציה pcap_sniffer , אנו משתמשים בפונקציה זו לאחל שקימפלנו סניפר בפונקציה הקודמת ואנו רוצים להאזין לתנועה ספציפית(אנחנו גם יכולים להאזין להכל תלוי במה אנו רוצים לתפוס).

פעם רביעית שאנו משתמשים בספריית pcap היא בפונקציה pcap_loop , לאחר שאנו תופסים את הפקטה פונקציה זאת שולחת את הפקטה לפונקציה שאנו הגדרנו, אנו יכולים לקבל בפונקציה שהגדרנו את הפקטה ושם לנתח את הפקטה ובגלל זה אנו משתמשים בפונקציה זו.

פעם רביעית שאנו משתמשים בספריית pcap היא בפונקציה pcap_close , אנו נשתמש בפונקציה זו כאשר אנו לא נרצה להאזין יותר ונרצה לסגור את הערוץ אליו אנו מאזינים.

שאלה 2:

אנו צריכים root privilege מכיוון שבפונקציה pcap_open_live אנו פותחים raw_socket וכאשר אנו

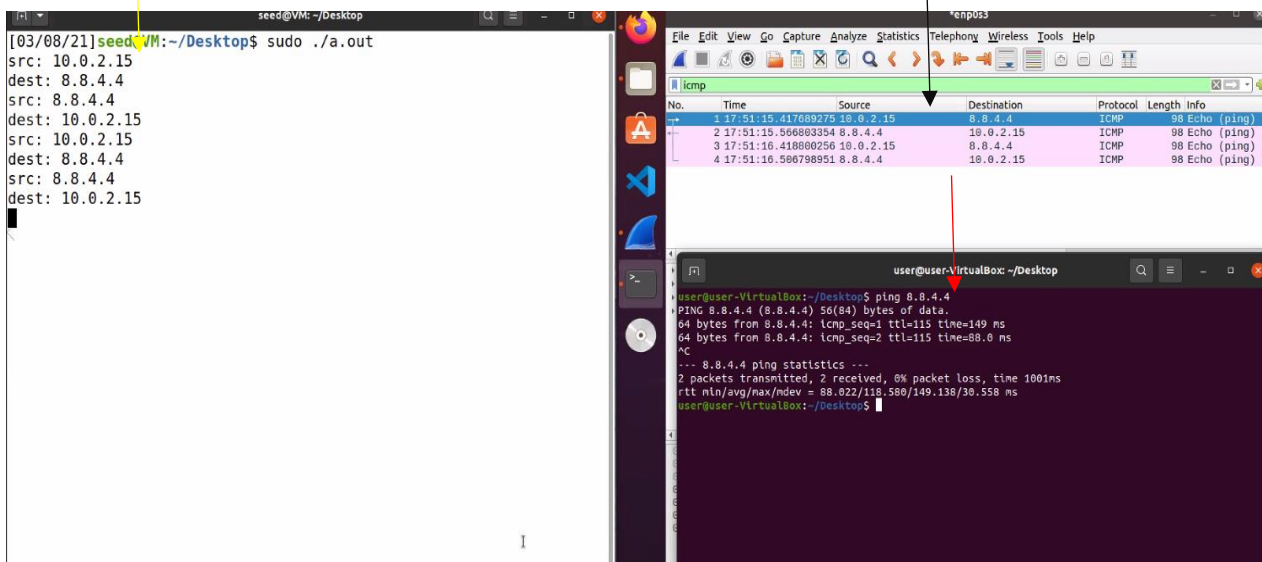
פותרים raw_socket זה בעצם פותר לנו גישה לכרטיס הרשת וכדי לעשות זאת אנו צריכים הרשאות root זאת אומרת כל פעם שנפח raw_socket דבר זה נותן לנו גישה לדברים משמעותיים שאנו לא רוצים שכל משתמש יוכל לגשת אליהם אלה אם כן הוא יש לו גישה של root ולכן אם נפעיל את תוכנית הסניפר שלנו ללא sudo התוכנית תיפול בפונקציה pcap_open_live .

שאלה 3:

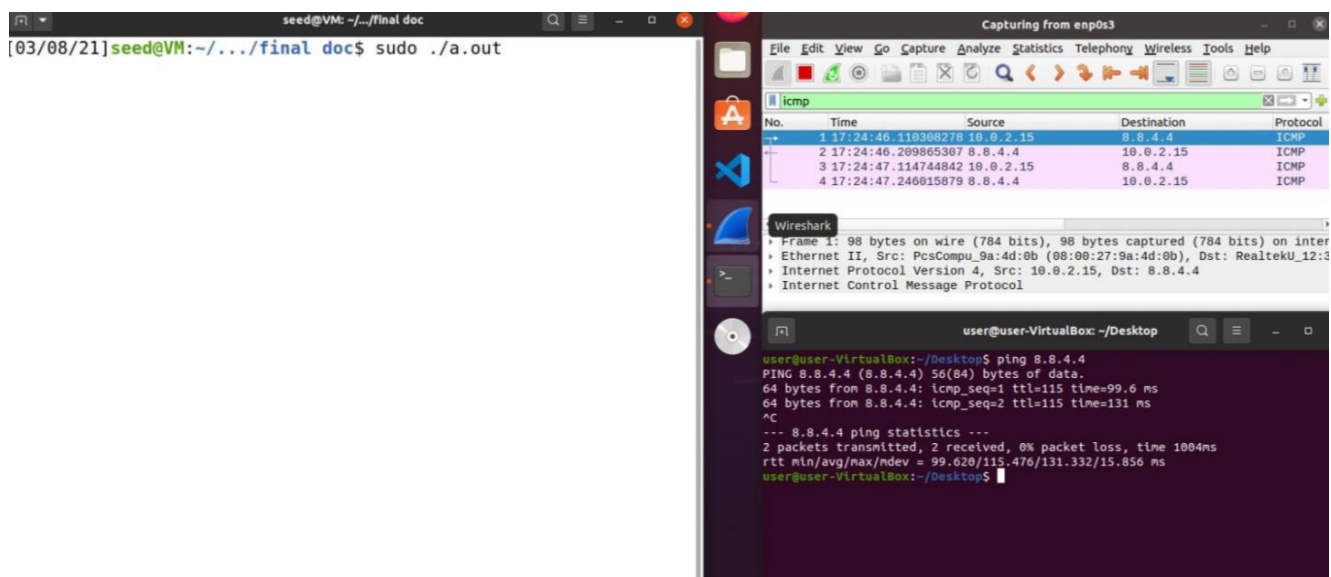
בשאלה זו התבקשנו להריץ את תוכנית הסניפר שכבתנו כשפעם אחת אנו במצב פרמיסקיס מוד ופעם לא ולהראות את ההבדל .

על מנת להראות הבדל אנו צריכים שיהיה עוד מחשב על אותו NIC ואז להראות שכאשר אנו במצב פרמיסקיס מוד אנו מצליחים לתפוס את הפקטה שלו אחרת לא.

ניתן לראות בתמונה המצורפת מצד שמאל את תוכנית הסניפר שלנו שיושבת ב VM נפרד ואכן מסניפה את אותם פרטים שיש ב VM מימין, כעת נראה שכאשר נכבה את הפרמיסקס מוד לא נצליח להסניף פקטות.



בתמונה המצורפת כבינו את הפרמיסקיס מוד וניתן לראות שאכן לא הוסנפו שום פקטות מצד שמאל תוכנת הסניפר שלנו היושבת במח נפרד ובצד שמאל המח השניה אשר שולחת את הודעת הפינג



:2.1B

(1)אנו צריכים לתפוס שתי פקטות בין שתי כתובות ספציפיות ניתן לראות את התפיסה בתמונות המצורפות:

הכתובות שהגדרו ככתובות שאנו רוצים לתפוס הם 8.8.8.8-8.8.4.4

נשים לב בתמונה ששלחנו פינג לשתי הכתובות הנ"ל ופעם אחת שלחנו ל walla.co.il את הפקטה של וואלה לא תפסו בעוד שאת האחרות כן תפסנו.

בתמונה ניתן לראות שליחת הפינג :

```
[03/02/21]seed@VM:~/.../2.1B$ ping 8.8.4.4
PING 8.8.4.4 (8.8.4.4) 56(84) bytes of data.
64 bytes from 8.8.4.4: icmp_seq=1 ttl=109 time=79.4 ms
^C
--- 8.8.4.4 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 79.372/79.372/79.372/0.000 ms
[03/02/21]seed@VM:~/.../2.1B$ 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=109 time=52.3 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=109 time=49.4 ms
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 49.387/50.863/52.340/1.476 ms
[03/02/21]seed@VM:~/.../2.1B$ ping walla.co.il
PING walla.co.il (13.225.255.40) 56(84) bytes of data.
64 bytes from server-13-225-255-40.tlv50.r.cloudfront.net (13.225.255.40): icmp_
seq=1 ttl=240 time=24.3 ms
64 bytes from server-13-225-255-40.tlv50.r.cloudfront.net (13.225.255.40): icmp_
seq=2 ttl=240 time=16.6 ms
64 bytes from server-13-225-255-40.tlv50.r.cloudfront.net (13.225.255.40): icmp_
seq=3 ttl=240 time=17.0 ms
^C
--- walla.co.il ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 16.644/19.324/24.328/3.541 ms
[03/02/21]seed@VM:~/.../2.1B$
```

בתמונה המצורפת ניתן לראות את תפיסת הפקטות (אך ורק של הכתובות 8.8.8.8,8.8.4.4)

```
seed@VM: ~/.../2.1B
[03/02/21] seed@VM: ~/.../2.1B$ sudo ./a.out
src: 10.0.2.4
dest: 8.8.4.4
src: 8.8.4.4
dest: 10.0.2.4
src: 10.0.2.4
dest: 8.8.8.8
src: 8.8.8.8
dest: 10.0.2.4
src: 10.0.2.4
dest: 8.8.8.8
src: 8.8.8.8
dest: 10.0.2.4
```

ובצילומי הwireShark ניתן שאכן שלחנו הודעות
request וקיבלנו הודעות reply מהכתובות הנ"ל(ניתן לראות
פה גם את ההודעה ששלחנו לוואלה):

[SEED Labs] Capturing from any

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Current filter: icmp

No.	Time	Source	Destination	Protocol	Length	Info
25	2021-03-02 12:3...	10.0.2.4	8.8.4.4	ICMP	100	Echo (ping) request id=0x002b, seq=1/256, ttl=64
26	2021-03-02 12:3...	8.8.4.4	10.0.2.4	ICMP	100	Echo (ping) reply id=0x002b, seq=1/256, ttl=10
692	2021-03-02 12:3...	10.0.2.4	8.8.8.8	ICMP	100	Echo (ping) request id=0x002c, seq=1/256, ttl=64
693	2021-03-02 12:3...	8.8.8.8	10.0.2.4	ICMP	100	Echo (ping) reply id=0x002c, seq=1/256, ttl=10
694	2021-03-02 12:3...	10.0.2.4	8.8.8.8	ICMP	100	Echo (ping) request id=0x002c, seq=2/512, ttl=64
695	2021-03-02 12:3...	8.8.8.8	10.0.2.4	ICMP	100	Echo (ping) reply id=0x002c, seq=2/512, ttl=10
716	2021-03-02 12:4...	10.0.2.4	13.225.255.40	ICMP	100	Echo (ping) request id=0x002d, seq=1/256, ttl=64
717	2021-03-02 12:4...	13.225.255.40	10.0.2.4	ICMP	100	Echo (ping) reply id=0x002d, seq=1/256, ttl=24
725	2021-03-02 12:4...	10.0.2.4	13.225.255.40	ICMP	100	Echo (ping) request id=0x002d, seq=2/512, ttl=64
726	2021-03-02 12:4...	13.225.255.40	10.0.2.4	ICMP	100	Echo (ping) reply id=0x002d, seq=2/512, ttl=24
727	2021-03-02 12:4...	10.0.2.4	13.225.255.40	ICMP	100	Echo (ping) request id=0x002d, seq=3/768, ttl=64
728	2021-03-02 12:4...	13.225.255.40	10.0.2.4	ICMP	100	Echo (ping) reply id=0x002d, seq=3/768, ttl=24

2) בשאלה זו אנו צריכים לתפוס פקטות tcp מפורט 10-100 בתמונה המצורפת ניתן לראות שאנו תופסים אך ורק פרקטות אלו ומדפיסים את ה src ו dest שלהם

```

^C
[03/02/21]seed@VM:~/.../2.1B$ gcc sniff_range.c -lpcap
[03/02/21]seed@VM:~/.../2.1B$ sudo ./a.out
src: 10.0.2.4
dest: 34.107.221.82
src: 34.107.221.82
dest: 10.0.2.4
src: 10.0.2.4
dest: 34.107.221.82
src: 10.0.2.4
dest: 34.107.221.82
src: 34.107.221.82
dest: 10.0.2.4
src: 34.107.221.82
dest: 10.0.2.4
src: 10.0.2.4
dest: 34.107.221.82
src: 10.0.2.4
dest: 34.107.221.82

```

Protocol: TCP (6)

Header checksum: 0x8fa4 [validation disabled]

[Header checksum status: Unverified]

Source: 10.0.2.4

Destination: 34.107.221.82

Transmission Control Protocol, Src Port: 50070, Dst Port: 80, Seq: 725100851, Len: 0

Source Port: 50070

Destination Port: 80

[Stream index: 7]

[TCP Segment Len: 0]

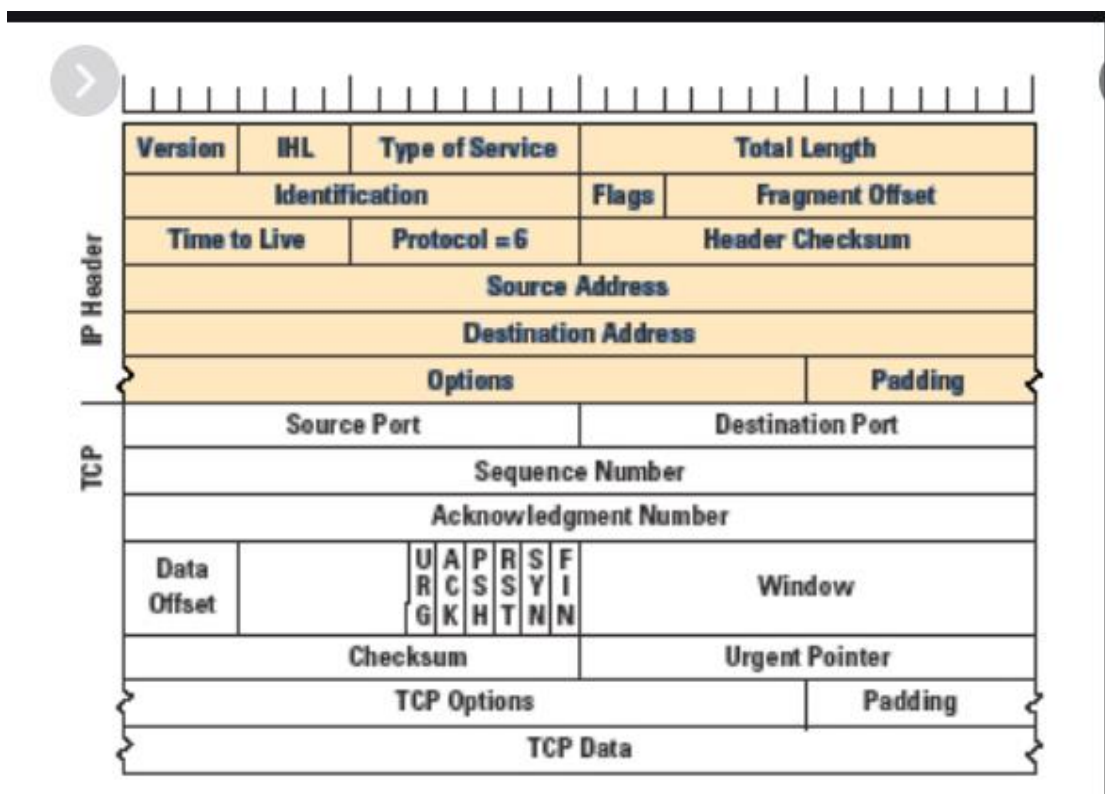
בתמונה הנוספת ניתן לראות תעבורת tcp במחשב עם פורט שהוא לא 10-100

ושהסינפר לא תפס אותו: מכיוון שהתמונה המצורפת היא מאותה הרצה של התמונה למעלה נוכל להסתכל על הפקטות שהסינפר תפס ולא נמצא את פקטה זו.

387	2021-03-02 13:4...	10.0.2.4	157.240.1.23	TCP	56 57012 → 443 [AC
388	2021-03-02 13:4...	216.58.212.206	10.0.2.4	TCP	62 443 → 59122 [AC
389	2021-03-02 13:4...	157.240.1.23	10.0.2.4	TLSv1.3	254 Application Dat
390	2021-03-02 13:4...	10.0.2.4	157.240.1.23	TCP	56 57012 → 443 [AC
391	2021-03-02 13:4...	216.58.212.206	10.0.2.4	TLSv1.3	1486 Server Hello, C
392	2021-03-02 13:4...	216.58.212.206	10.0.2.4	TCP	1516 443 → 59122 [AC
393	2021-03-02 13:4...	10.0.2.4	216.58.212.206	TCP	56 59122 → 443 [AC
394	2021-03-02 13:4...	10.0.2.4	216.58.212.206	TCP	56 59122 → 443 [AC
395	2021-03-02 13:4...	216.58.212.206	10.0.2.4	TLSv1.3	62 Application Dat
396	2021-03-02 13:4...	10.0.2.4	216.58.212.206	TCP	56 59122 → 443 [AC
397	2021-03-02 13:4...	10.0.2.4	157.240.1.23	TLSv1.3	120 Change Cipher S
398	2021-03-02 13:4...	10.0.2.4	157.240.1.23	TLSv1.3	226 Application Dat
399	2021-03-02 13:4...	157.240.1.23	10.0.2.4	TCP	62 443 → 57012 [AC

Header checksum: 0xa254 [validation disabled]
[Header checksum status: Unverified]
Source: 216.58.212.206
Destination: 10.0.2.4
Transmission Control Protocol, Src Port: 443, Dst Port: 59122, Seq: 4602602, Ack: 3140242976, L
Source Port: 443
Destination Port: 59122

2.1C: בשאלה זו אנו צריכים לתפוס את הסיסמא שאנו מכניסים ל telnet אותה, בכתיבת הקוד נעזרנו המון במבנה של tcp_ip:



הסבר הקוד מצורף בקוד כעת נראה את הצילומים של הרצת הקוד ואת התוצאה שתקבלה באישי

הסיסמה שהזנו היא dees ונראה שזה אכן מה שקיבלנו:

```
seed@VM: ~/2.1C
[03/02/21]seed@VM:~/2.1C$ gcc -g telnet.c -lpcap
[03/02/21]seed@VM:~/2.1C$ sudo ./a.out
dees

[03/02/21]seed@VM:~/2.1C$ telnet localhost
Trying 127.0.0.1...
Connected to localhost.
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

0 updates can be installed immediately.
0 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
Your Hardware Enablement Stack (HWE) is supported until April 2025.
Last login: Tue Mar  2 14:41:21 EST 2021 from localhost on pts/2
```

כעת נראה את נכונות הקוד ב-wireShark

54	2021-03-02 14:4...	127.0.0.1	127.0.0.1	TELNET	78 Telnet Data ...
56	2021-03-02 14:4...	127.0.0.1	127.0.0.1	TELNET	69 Telnet Data ...
58	2021-03-02 14:4...	127.0.0.1	127.0.0.1	TELNET	69 Telnet Data ...

- ▶ TCP Option - No-Operation (NOP)
- ▶ TCP Option - No-Operation (NOP)
- ▼ TCP Option - Timestamps: TSval 136658418, TSecr 136658416
 - Kind: Time Stamp Option (8)
 - Length: 10
 - Timestamp value: 136658418
 - Timestamp echo reply: 136658416
- ▶ [SEQ/ACK analysis]
- ▶ [Timestamps]
- TCP payload (10 bytes)

▼ Telnet

Data: Password:

56	2021-03-02 14:4...	127.0.0.1	127.0.0.1	TELNET	69 Telnet Data ...
58	2021-03-02 14:4...	127.0.0.1	127.0.0.1	TELNET	69 Telnet Data ...

- ▶ TCP Option - No-Operation (NOP)
- ▶ TCP Option - No-Operation (NOP)
- ▼ TCP Option - Timestamps: TSval 136658844, TSecr 136658418
 - Kind: Time Stamp Option (8)
 - Length: 10
 - Timestamp value: 136658844
 - Timestamp echo reply: 136658418
- ▶ [SEQ/ACK analysis]
- ▶ [Timestamps]
- TCP payload (1 byte)

▼ Telnet

Data: d

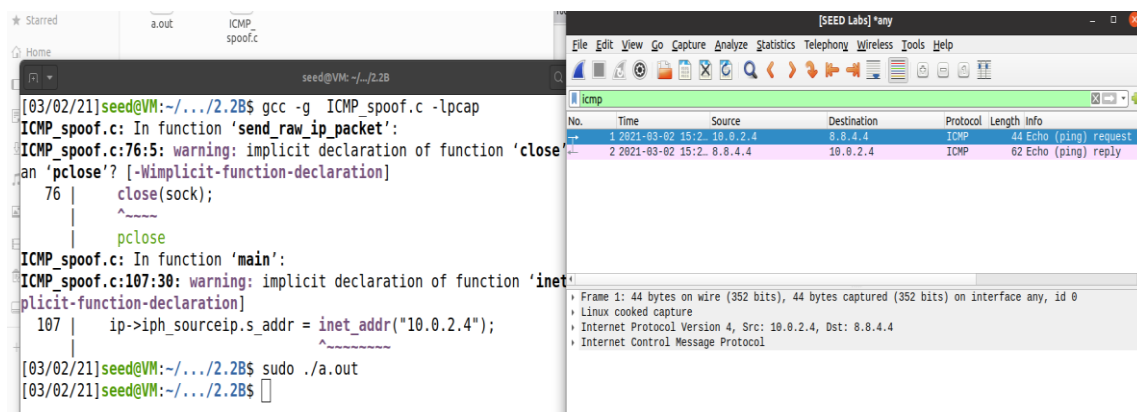
58	2021-03-02 14:4...	127.0.0.1	127.0.0.1	TELNET	69 Telnet Data ...
60	2021-03-02 14:4...	127.0.0.1	127.0.0.1	TELNET	69 Telnet Data ...
62	2021-03-02 14:4...	127.0.0.1	127.0.0.1	TELNET	69 Telnet Data ...
64	2021-03-02 14:4...	127.0.0.1	127.0.0.1	TELNET	70 Telnet Data ...
<ul style="list-style-type: none"> ▸ TCP Option - No-Operation (NOP) ▸ TCP Option - No-Operation (NOP) ▾ TCP Option - Timestamps: TSval 136659153, TSecr 136658885 <ul style="list-style-type: none"> Kind: Time Stamp Option (8) Length: 10 Timestamp value: 136659153 Timestamp echo reply: 136658885 ▸ [SEQ/ACK analysis] ▸ [Timestamps] TCP payload (1 byte) 					
<ul style="list-style-type: none"> ▾ Telnet <ul style="list-style-type: none"> Data: e 					

60	2021-03-02 14:4...	127.0.0.1	127.0.0.1	TELNET	69 Telnet Data ...
62	2021-03-02 14:4...	127.0.0.1	127.0.0.1	TELNET	69 Telnet Data ...
64	2021-03-02 14:4...	127.0.0.1	127.0.0.1	TELNET	70 Telnet Data ...
<ul style="list-style-type: none"> ▸ TCP Option - No-Operation (NOP) ▸ TCP Option - No-Operation (NOP) ▾ TCP Option - Timestamps: TSval 136659335, TSecr 136659153 <ul style="list-style-type: none"> Kind: Time Stamp Option (8) Length: 10 Timestamp value: 136659335 Timestamp echo reply: 136659153 ▸ [SEQ/ACK analysis] ▸ [Timestamps] TCP payload (1 byte) 					
<ul style="list-style-type: none"> ▾ Telnet <ul style="list-style-type: none"> Data: e 					

62	2021-03-02 14:4...	127.0.0.1	127.0.0.1	TELNET	69 Telnet Data ...
64	2021-03-02 14:4...	127.0.0.1	127.0.0.1	TELNET	70 Telnet Data ...
<ul style="list-style-type: none"> ▸ TCP Option - No-Operation (NOP) ▸ TCP Option - No-Operation (NOP) ▾ TCP Option - Timestamps: TSval 136659582, TSecr 136659335 <ul style="list-style-type: none"> Kind: Time Stamp Option (8) Length: 10 Timestamp value: 136659582 Timestamp echo reply: 136659335 ▸ [SEQ/ACK analysis] ▸ [Timestamps] TCP payload (1 byte) 					
<ul style="list-style-type: none"> ▾ Telnet <ul style="list-style-type: none"> Data: s 					

בשאלה זו התבקשנו לעשות ספופינג של icmp echo ובמידה והצלחנו נראה הודעה reply מהסרבר ששלחנו אליו , פירוט על כתיבת הקוד נמצא בקוד כעת נראה הוכחת נכונות של הקוד בעזרת wireShark

שלחנו את ההודעה מהאי פי שלנו לאי פי 8.8.4.4:



```
[03/02/21]seed@VM:~/2.2B$ gcc -g ICMP_spoof.c -lpcap
ICMP_spoof.c: In function 'send_raw_ip_packet':
ICMP_spoof.c:76:5: warning: implicit declaration of function 'close'
an 'pclose'? [-Wimplicit-function-declaration]
76 |     close(sock);
    |     ^~~~~~
    |     pclose
ICMP_spoof.c: In function 'main':
ICMP_spoof.c:107:30: warning: implicit declaration of function 'inet_pton'
[implicit-function-declaration]
107 |     ip->iph_sourceip.s_addr = inet_addr("10.0.2.4");
    |                               ^~~~~~
[03/02/21]seed@VM:~/2.2B$ sudo ./a.out
[03/02/21]seed@VM:~/2.2B$
```

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-03-02 15:2...	10.0.2.4	8.8.4.4	ICMP	44	Echo (ping) request
2	2021-03-02 15:2...	8.8.4.4	10.0.2.4	ICMP	62	Echo (ping) reply

request

reply

שאלה 4: התשובה היא כן! אנו אכן יכולים להציב length שרירותי אבל זה לא משנה מכיוון שה-ip_length חוזר לגודל המקורי שלו בלי שום קשר לגודל שהוגדר בתוכנית.

שאלה 5: התשובה לשאלה זו היא לא! איננו צריכים לחשב את checksum כי יש באפשרותנו את האופציה לבקש ממערכת ההפעלה לחשב את ה checksum

שאלה 6:

אנו חייבים להשתמש ב- root privilege מכיוון שאנו משתמשים בrawsocket וכשאנו

פותחים raw_socket זה בעצם פותח לנו גישה לכרטיס הרשת וכדי לעשות זאת אנו צריכים הרשאות root זאת אומרת כל פעם שנפח raw_socket דבר זה נותן לנו גישה לדברים משמעותיים שבד"כ מערכת ההפעלה אחראית

עליהם שאנו לא רוצים שכל משתמש יוכל לגשת אליהם אלה
אם כן הוא יש לו גישה של root

ולכן אם נפעיל את תוכנית ה- sniffer_spoofers שלנו ללא
sudo התוכנית תיפול בפונקציה socket() מכיוון ששם נפתח
הrawsocket.

2.3:

בשאלה זו התבקשנו לבנות תוכנה אשר מאזינה לתנועה
בסאב נאט שלנו וכאשר היא צופה בהודעת פינג בין אם
ה dest קיים או לא היא מחזירה הודעת reply, נעזרנו בדוקר
על מנת לפתוח עוד מחשב שיושב על הסאב נאט שלנו
ושלחנו ממנו הודעת פינג ל1.2.3.4 ip.

קו זה אינו קיים ברשת אבל בתמונה המצורפת נשים לב
שלאחר ששלחנו הודעת פינג נקבל הודעת reply מאותו ip:

The image shows a terminal window on the left and a Wireshark packet capture window on the right.

Terminal Window:

```
seed@VM: ~/Labsetup
[03/03/21]seed@VM:~/Labsetup$ gcc -g sniffing_spoofing.c -lpcap
[03/03/21]seed@VM:~/Labsetup$ sudo ./a.out
^C
[03/03/21]seed@VM:~/Labsetup$ gcc -g spoof_sniff.c -lpcap
[03/03/21]seed@VM:~/Labsetup$ sudo ./a.out
^C

root@VM:~# ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
^C
--- 1.2.3.4 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4085ms

root@VM:~# sudo ping 1.2.3.4
bash: sudo: command not found
root@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=570 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=64 time=593 ms
64 bytes from 1.2.3.4: icmp_seq=3 ttl=64 time=618 ms
^C
--- 1.2.3.4 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 569.626/593.584/618.086/19.787 ms
root@VM:~# ping 1.2.3.4
```

Wireshark Window:

The packet list shows ICMP Echo (ping) requests and replies. The packet details pane shows the selected packet (No. 1) as an Internet Control Message Protocol (ICMP) Echo (ping) request.

No.	Time	Source	Destination	Protocol	Length	Info
1	2021-03-03 14:11:10.0.2.4	10.0.2.4	1.2.3.4	ICMP	100	Echo (ping) request
2	2021-03-03 14:11:10.0.2.4	10.0.2.4	1.2.3.4	ICMP	100	Echo (ping) reply
3	2021-03-03 14:11:10.0.2.4	10.0.2.4	1.2.3.4	ICMP	100	Echo (ping) request
4	2021-03-03 14:11:10.0.2.4	10.0.2.4	1.2.3.4	ICMP	100	Echo (ping) reply
5	2021-03-03 14:11:10.0.2.4	10.0.2.4	1.2.3.4	ICMP	100	Echo (ping) request
6	2021-03-03 14:11:10.0.2.4	10.0.2.4	1.2.3.4	ICMP	100	Echo (ping) reply

The packet details pane for the selected packet (No. 1) shows:

- Frame 1: 100 bytes on wire (800 bits), 100 bytes captured (800 bits) on interface any, id 0
- Linux cooked capture
- Internet Protocol Version 4, Src: 10.0.2.4, Dst: 1.2.3.4
- Internet Control Message Protocol

The packet bytes pane shows the raw data of the ICMP Echo (ping) request.