

### רשתות ותקשורת - תרגיל בית 1

**שאלה 1:** נפרט ונסביר על כל אחת מהשורות הקוד בקובץ הראשון של פנינו שיוצר udp server

```
1  from socket import socket, AF_INET, SOCK_DGRAM
2  s = socket(AF_INET, SOCK_DGRAM)
3  source_ip = '127.0.0.1'
4  source_port = 123456
5  s.bind((source_ip, source_port))
6  while True:
7      data, sender_info = s.recvfrom(2048)
8      print "Message: ", data, " from: ", sender_info
9      s.sendto(data.upper(), sender_info)
```

- (1) בשורה הראשונה אנחנו מבצעים Import מתוך הספרייה socket לבאים:  
socket – פונקציה שמתירה ליצור סוקט – נקודת קצה של תקשורת  
IP version 4 – מסמל לנו שהכתובות הלוגיות יהיו בגרסת 4  
SOCK\_DGRAM – סוקט מסוג UDP
- (2) יוצרים סוקט על ידי הפונקציה socket עם הפרמטרים שהציגו קודם (כתובות זו גרסה 4 וסוקט  
סוג UDP) ושומרים את הסוקט במשתנה s
- (3) מגדירים את כתובת IP מקור – ומגדירים אותה להיות זהה ל-host מכיוון שביקשו מאיינו  
בתרגיל שגם השרת וגם הלקוח יריצו לوكאלית על המחשב
- (4) מגדירים כתובת פורט מקור - 123456 .
- (5) מבצעים Bind – פונקציה שמפעלת על הסוקט, ומקבלת 2 פרמטרים: כתובת IP מקור ומספר  
פורט מקור. אנחנו בעצם לוקחים את הסוקט שקיבל פורט אكريי ממערכת הפעלה ונקשר לו  
את ה포רט שאנו רוצים (הפורט שידוע ללקוחות כזזה של השרת).
- (6) מתחילה לולה שתמשיך לוחץ כל עוד הכל אמת
- (7) מבצעים על הסוקט את הפונקציה recvfrom , אנחנו בעצם מבקשים מהסוקט שיתן לנו חבילה  
אחת מתוך אלו שהגעו (טור כד' הגבלה של 2048 בתים). הפונקציה מחזירה לנו את ה-data,  
כלומר החבילה עצמה - sender\_info , כלומר מידע על השולח, כד' שהשרת ידע לאן להשב  
מדפים את תוכן החבילה, ולאחר מכן את המידע על השולח : כתובת IP ומספר PORT
- (8) מפעילים על הסוקט את הפונקציה sendto עם המידע שקיבלנו מהחבילה ועליה מפעילים את  
הfonktsיה שהופכת את תוכן ההודעה לאותיות גדולות, וכפרמטר שני את פרטי השולח.
- (9)

cut נפרט על כל אחת משורות הקוד בקובץ הראשון שלפנינו שיוצר udp client

```
1  from socket import socket, AF_INET, SOCK_DGRAM
2  s = socket(AF_INET, SOCK_DGRAM)
3  dest_ip = '127.0.0.1'
4  dest_port = 123456
5  msg = raw_input("Message to send: ")
6  while not msg == 'quit':
7      s.sendto(msg, (dest_ip, dest_port))
8      data, sender_info = s.recvfrom(2048)
9      print "Server sent: ", data
10     msg = raw_input("Message to send: ")
11 s.close()
```

(1) בשורה הראשונה אנחנו מבצעים Import Import הטעינה socket לברים:

(2) – פונקציה שמרתה ליצור סוקט – נקודת קצה של תקשורת

(3) AF\_INET – מסמל לנו שהכטבות הלוגיות יהיו בגרסת 4 IP version 4

(4) SOCK\_DGRAM – סוקט מסוג UDP

(5) יוצרים סוקט על ידי הפונקציה socket עם הפרמטרים שהציגו קודם (כתובות IP גרסה 4 וסוקט מסוג UDP) ושומרם את הסוקט במשתנה s

(6) מגדירים את כתובות IP יעד – ומגדירים אותה להיות זהה ל-local host מכיוון שביקשו מאייתנו

(7) בתרגיל שגם השרת וגם הלקוח ירצו לokaneית על המחשב

(8) מגדירים כתובות יעד- 123456 , זהה למספר הפורט של הלקוח

(9) הלקוח שולח הודעה בה מצהיר שהוא מעוניין לשЛОוח חביליה.

(10) רצים בתוך הלולאה כל עוד לא קיבלנו את ההודעה quest, כלומר שאין עוד חבילות לשЛОוח

(11) מפעלים על הסוקט את הפונקציה sendto בה מגדירים את תוכן החביליה שנשלחת, ואת פרטיה הנמען (השרת) כתובות IP ומספר פורט.

(12) מחכים למידע המתקבל חוזה מהשרת על ידי קר שמבצעים על הסוקט את הפונקציה recvform , אנחנו בעצם מבקשים מהסוקט שייתן לנו חבילה אחת מתוך אלו שהגיעו (טור CD')

(13) הגבלה של 2048 בתים). הפונקציה מחזירה לנו את ה-data, כלומר החביליה עצמה ו-

(14) sender\_info , כלומר מידע על השולח, כדי שהלקוח ידע לאן להשיב. השרת יחזיר לנו את

(15) ההודעה באותיות גדולות, כפי שהסבירתי בעמוד קודם.

(16) מדפסים את ההודעה שהתקבלה

(17) שוב נחזיר על קבלת הקלט, עד שנמצא מהלולאת while

(18) הלקוח מבצע close על הסוקט, מכיוון שישים את התקשרות עם השרת ולכן יפנה את הסוקט

(19) מקום) על מנת לאפשר לאפליקציות (לקוחות) אחריות להשתמש בסוקט שהתפנה.

## שאלה 2

A. מספר החבילות שהסנוו הוא 10770

Wireshark interface list:

- 10719 61.829839744 fe80::b2d5:9dff:fed... fe80::7e91:7ae0:faf... DHCPv6 1. Advertise XID: 0x410be8 CID: 0004acf629d664abde26bf5ef2e5544ec1
- 10720 61.821561238 fe80::b2d5:9dff:fed... fe80::7e91:7ae0:faf... DHCPv6 1. Advertise XID: 0x410be8 CID: 0004acf629d664abde26bf5ef2e5544ec1
- 10721 61.850922365 192.36.253.112 10.0.0.28 TCP 68 [TCP Keep-Alive ACK] 443 - 52746 [ACK] Seq=25671 Ack=1186 Win=30720 Len=0 TSval=409443363 TSecr=

Frame 1: 114 bytes on wire (912 bits), 114 bytes captured (912 bits) on interface 0

Protocol: Internet Protocol Version 4, Src: 10.0.0.28, Dst: 172.217.22.110

Transmission Control Protocol, Src Port: 66626, Dst Port: 443, Seq: 1, Ack: 1, Len: 46

Secure Sockets Layer

```

0000  00 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00
0001  45 00 00 00 45 00 45 00 00 00 00 00 00 00 00 00
0002  00 39 16 6c cc d2 01 bb 23 18 47 79 27 35 45 00
0003  80 18 01 a5 af 8d 00 00 01 01 08 00 d1 d0 b4 fb
0049  65 98 1c 58 17 03 03 00 20 00 00 00 00 00 00 00
0050  0b 0d 66 3b 78 9e ad 05 a7 15 9a 5c 1e 36 c1 94
0066  f6 f2 1c 78 b3 db 47 9a d5 fd de 12 a8 f0 82
0070  66 74

```

Packets: 10770 · Displayed: 10770 (100.0%) · Profile: Default

Packets: 10770 ·

ב. לאחר סינון חבילות שנשלחו על גבי פרוטוקול UDP בלבד גילינו שהו 1034 חבילות כאלה.

Wireshark interface list:

- 5 0.936133044 10.10.10.254 255.255.255.255 UDP 5 DHCP Discover - Transaction ID 0x2ecd433
- 7 3.278388442 10.0.0.7 255.255.255.255 UDP-LSP 1. Dropbox LAN sync Discovery Protocol
- 8 3.280468077 10.0.0.7 10.0.0.255 UDP-LSP 1. Dropbox LAN sync Discovery Protocol
- 9 12.088076863 10.10.19.254 255.255.255.255 DHCP 5. DHCP Discover - Transaction ID 0x6ce7bbdb
- 16 15.160608240 10.0.0.254 255.255.255.255 DHCP 5. DHCP Discover - Transaction ID 0x23844f59
- 17 18.0239390430 10.0.0.48 224.0.0.113 ALLJOY 1. VERSION 0 ISAT
- 18 18.025760844 10.0.0.48 10.0.0.255 ALLJOY 1. VERSION 0 ISAT
- 19 18.0278108935 10.0.0.48 224.0.0.113 ALLJOY 1. VERSION 1 ISAT
- 20 18.0280520835 10.0.0.48 10.0.0.255 ALLJOY 1. VERSION 1 ISAT
- 29 24.349489319 127.0.0.1 127.0.1.1 DNS 77 Standard query 0xb5ca A mail.google.com
- 30 24.349517102 127.0.0.1 127.0.1.1 DNS 77 Standard query 0x62f5 AAAA mail.google.com
- 31 24.349563845 10.0.0.28 10.0.0.138 DNS 77 Standard query 0x62e7 A mail.google.com
- 32 24.349664947 fda0:110a:8c3f:0:d0.. fda0:110a:8c3f:0:16.. DNS 77 Standard query 0x367e AAAA mail.google.com
- 33 24.349637169 10.0.0.28 10.0.0.138 DNS 77 Standard query 0xd261 A web.whatsapp.com
- 34 24.349730266 127.0.0.1 127.0.1.1 DNS 78 Standard query 0x13ac A drive.google.com
- 35 24.349755985 127.0.0.1 127.0.1.1 DNS 78 Standard query 0x656e AAAA drive.google.com
- 36 24.349762969 127.0.0.1 127.0.1.1 DNS 78 Standard query 0x656e AAAA drive.google.com
- 37 24.349804689 10.0.0.28 10.0.0.138 DNS 78 Standard query 0x221a A web.whatsapp.com
- 38 24.349804689 10.0.0.28 10.0.0.138 DNS 78 Standard query 0x221a A drive.google.com
- 39 24.349823962 10.0.0.28 10.0.0.138 DNS 78 Standard query 0xb98d AAAA drive.google.com
- 40 24.351279393 10.0.0.138 10.0.0.28 DNS 1. Standard query response 0x62e7 A mail.google.com CNAME googlegmail.l.google.com A 172.217.22.101
- 41 24.351666048 127.0.1.1 127.0.0.1 DNS 1. Standard query response 0xb5ca A mail.google.com CNAME googlegmail.l.google.com A 172.217.22.101
- 42 24.351950455 10.0.0.138 10.0.0.28 DNS 1. Standard query response 0x367e AAAA mail.google.com CNAME googlegmail.l.google.com AAAA 2a00:1450..
- 43 24.351999916 127.0.1.1 127.0.0.1 DNS 1. Standard query response 0x63f5 AAAA mail.google.com CNAME googlegmail.l.google.com AAAA 2a00:1450..
- 44 24.352266855 127.0.0.1 127.0.1.1 DNS 72 Standard query 0xeb62 A github.com
- 45 24.35282179 10.0.0.28 10.0.0.138 DNS 72 Standard query 0x7a0a A github.com
- 46 24.352428148 127.0.0.1 127.0.1.1 DNS 72 Standard query 0x2bd7 AAAA github.com
- 47 24.352428148 127.0.0.1 127.0.1.1 DNS 72 Standard query 0x2bd7 AAAA github.com
- 50 24.47784705 fda0:110a:8c3f:0:16.. fda0:110a:8c3f:0:d0.. DNS 1. Standard query response 0x4e6e A mail.google.com CNAME googlegmail.l.google.com A 172.217.22.101
- 51 24.832566632 127.0.0.1 127.0.1.1 DNS 88 Standard query 0x1e0d A tiles.services.mozilla.com
- 52 24.8316468856 10.0.0.28 10.0.0.138 DNS 88 Standard query 0x94f3 A tiles.services.mozilla.com

Frame 5: 592 bytes on wire (4736 bits), 592 bytes captured (4736 bits) on interface 0

Protocol: User Datagram Protocol, Src Port: 68, Dst Port: 67

Bootstrap Protocol (Discover)

```

0000  00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010  45 00 02 40 00 00 40 00 40 11 23 0a 0a 0a fe
0020  ff ff ff ff 00 44 00 43 02 2c 05 fa 01 01 06 00
0030  32 ee d4 33 ff 00 80 00 00 00 00 00 00 00 00 00
0039  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0049  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0059  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0069  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0079  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0089  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0099  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00a9  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00ba  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00ca  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00da  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00ea  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

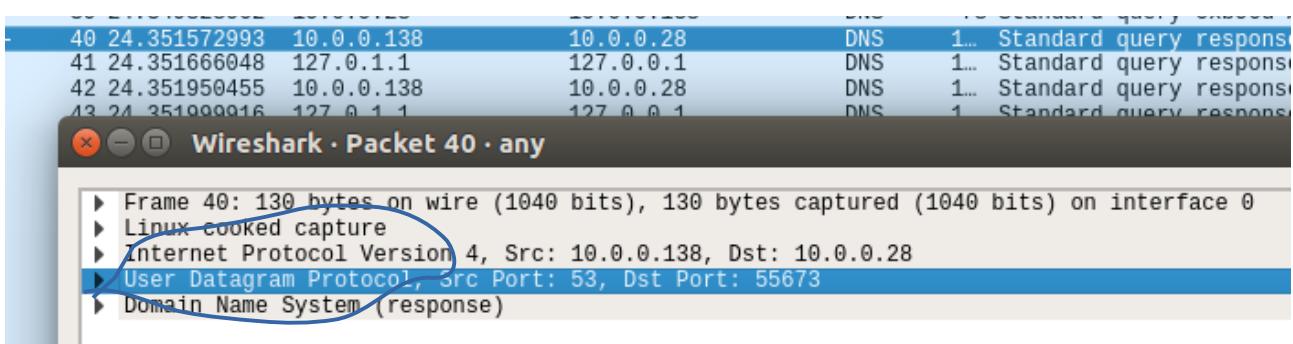
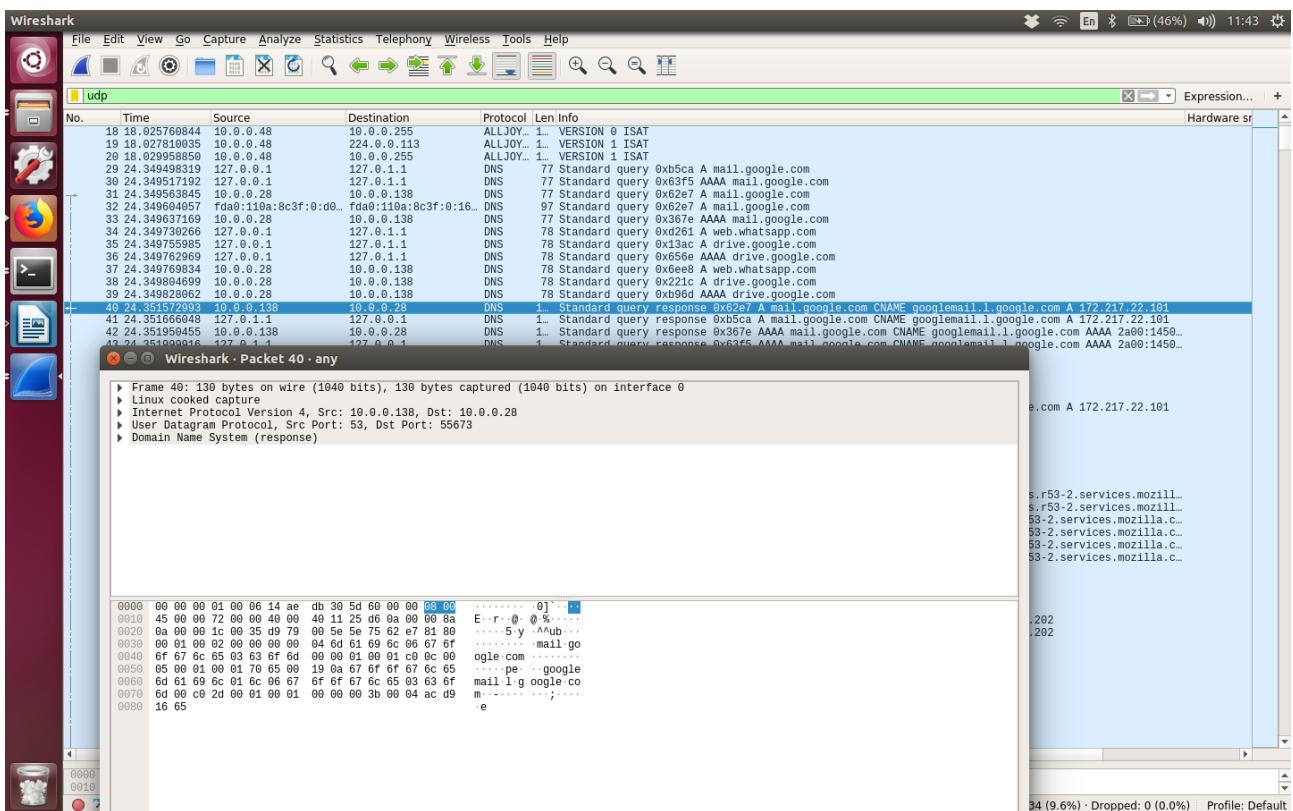
```

Packets: 10770 · Displayed: 1034 (9.6%) · Dropped: 0 (0.0%) · Profile: Default

Packets: 10770 · Displayed: 1034 (9.6%) ·

אם נלץ על אחת החבילות על מנת לקבל מידע נוסף

נראה שכאן מדובר בחבילות בעברית על גבי פרוטוקול UDP :



ג.על מנת לבדוק את כתובות ה IP בה משתמש המחשב כרגע נקליד את הפקודה ifconfig, ונקבל שכתובת ה IP הינה 10.0.0.28

```

Terminal File Edit View Search Terminal Help
coral2018@coral2018-S500CA:~$ ifconfig
enp3s0    Link encap:Ethernet HWaddr 74:d0:2b:16:2b:66
          UP BROADCAST MULTICAST MTU:1500 Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)
          Interrupt:19

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING MTU:65536 Metric:1
          RX packets:39131 errors:0 dropped:0 overruns:0 frame:0
          TX packets:39131 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2864521 (2.8 MB) TX bytes:2864521 (2.8 MB)

wlp2s0    Link encap:Ethernet HWaddr 6c:71:d9:3b:12:61
          inet addr:10.0.0.28 Bcast:10.0.0.255 Mask:255.255.255.0
          inet6 addr: fd00:110a:8c3f:0:2865:cdc3:4d09:49d0/64 Scope:Global
          inet6 addr: fe80::7e91:7ae0:fa69:5653/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:183632 errors:0 dropped:0 overruns:0 frame:0
          TX packets:106733 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:178721392 (178.7 MB) TX bytes:16578693 (16.5 MB)

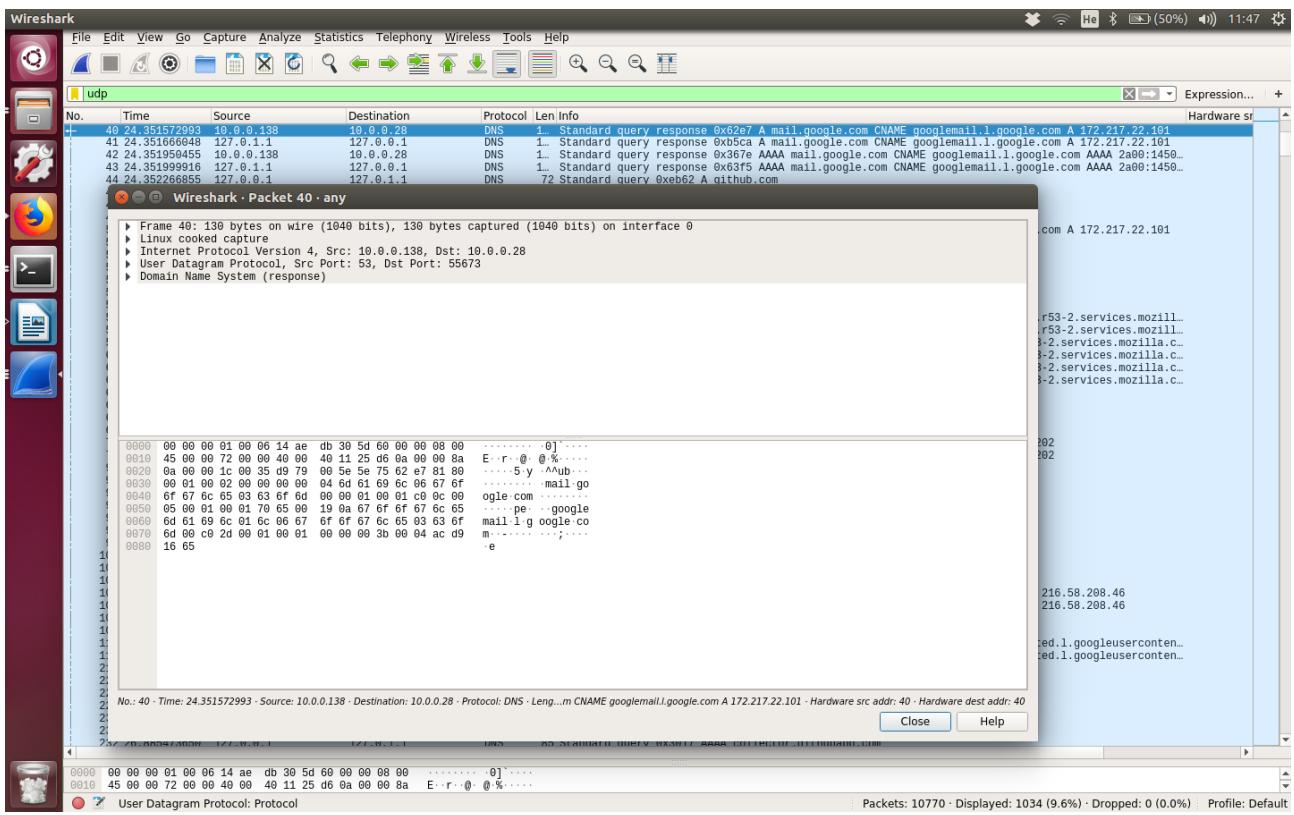
coral2018@coral2018-S500CA:~$ 

```

inet addr:10.0.0.28

ד. החבילה שבחרתי (חבילת DNS) ורצה על גבי פרוטוקול UDP ניתן לראות שהיא לא הגיעה מהמחשב שלנו אלא נשלחה אליו. ניתן להבין זאת על ידי קר שב-src מופיעה כתובות IP שאינה של המחשב שלנו (10.0.0.138), ובכתובות dest כתובות ה IP הינה של המחשב שלנו(10.0.0.28). לכן החבילה נשלחת אל המחשב שלנו.

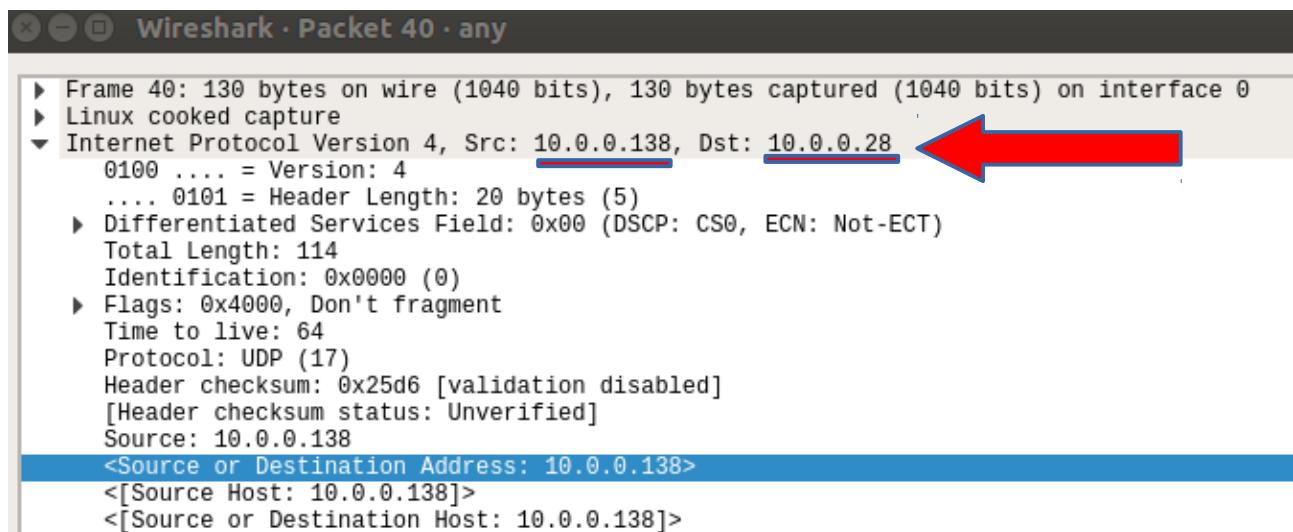
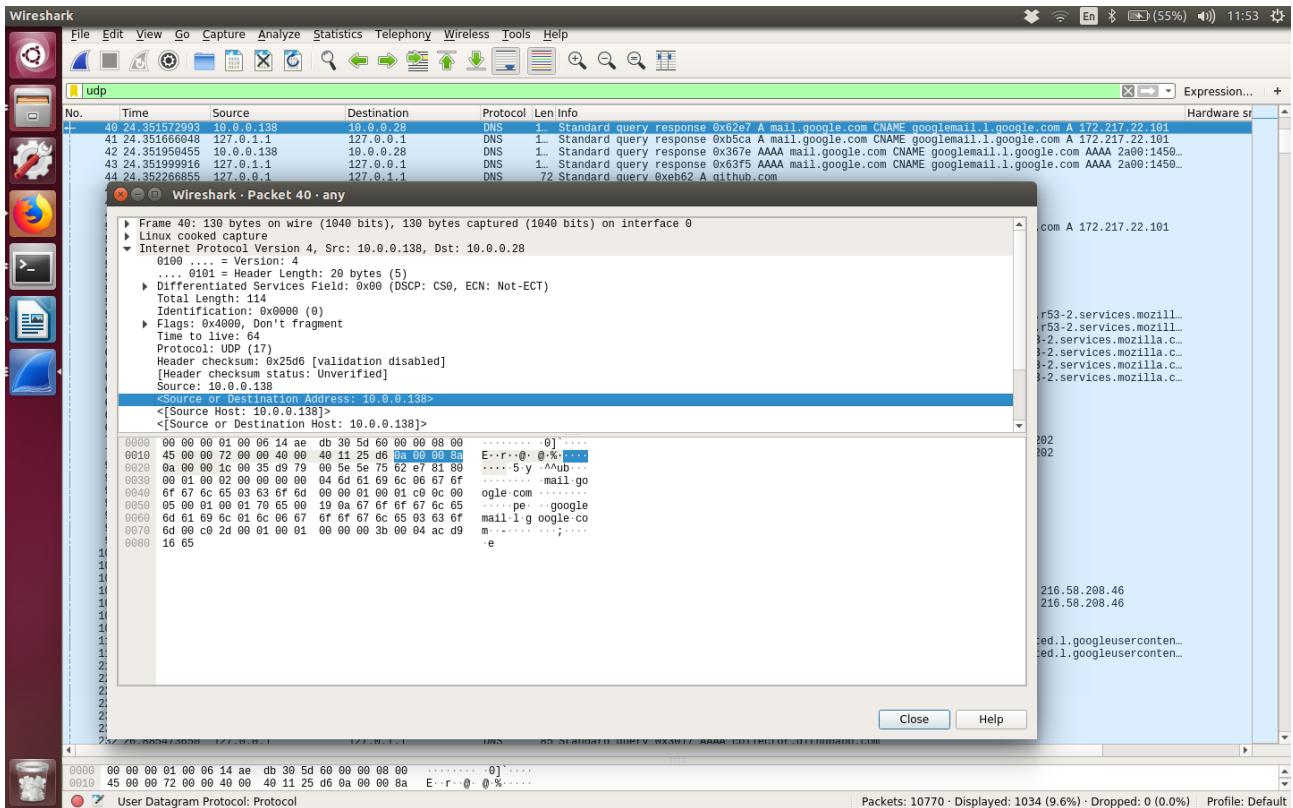
	Time	Source	Destination
40	24.351572993	10.0.0.138	10.0.0.28



נשים לב שהחbillה נשלחה מפורט מס' 53 אל פורט מס' 55673. קלומר הרשת האין בפורט 53 והלקוח האין בפורט 58358 (הלקוח זה המחשב שלנו).

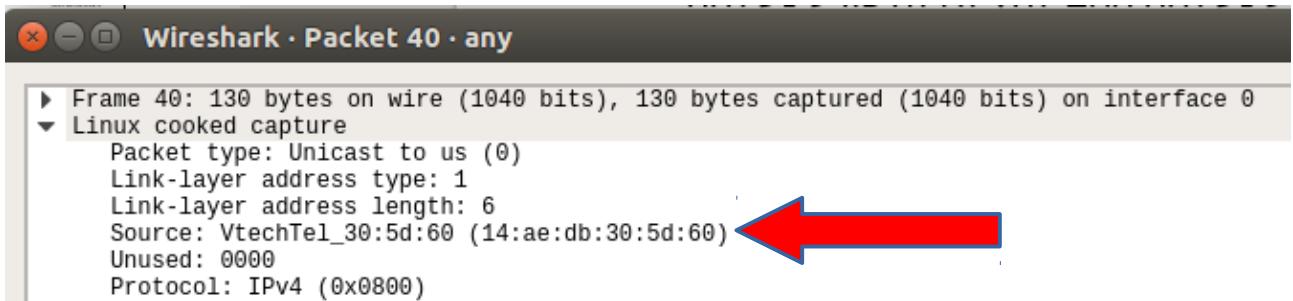


אכן במקרה שכבת הרשת ניתן לראות את כתובת הIP של השולח - הרשת - 10.0.0.138, וככתובת הIP של המქבל - המחשב שלנו (הלקוח) - 10.0.0.28.



כתובת MAC של השולח (של השרת) היא :

14:ae:db:30:5d:60



כתובת MAC של המקלט (הлокוט – המחשב שלנו) , נמצוא אותה על ידי מציאת החבילה ההפוכה הספציפית שמתאימה לו, זו שיצרה את התקשרות בין השרת למחשב שלנו:

No.	Time	Source	Destination	Protocol	Len Info
8	3.289468977	10.0.0.7	10.0.0.255	DB-LSP	1. Dropbox LAN sync Discovery Protocol
9	12.088976863	10.18.10.254	255.255.255.255	DHCP	5. DHCP Discover - Transaction ID 0x6ce7bbdb
16	16.09068244	10.18.10.254	255.255.255.255	DHCP	5. DHCP Discover - Transaction ID 0x23844f59
17	18.023939430	10.0.0.48	224.0.0.113	ALLJOY_	1. VERSION 0 ISAT
18	18.025768644	10.0.0.48	10.0.0.255	ALLJOY_	1. VERSION 0 ISAT
19	18.027819035	10.0.0.48	224.0.0.113	ALLJOY_	1. VERSION 1 ISAT
20	18.029998585	10.0.0.48	10.0.0.255	ALLJOY_	1. VERSION 1 ISAT
29	24.349498319	127.0.0.1	127.0.1.1	DNS	77 Standard query 0xb5ca A mail.google.com
30	24.349517192	127.0.0.1	127.0.1.1	DNS	77 Standard query 0x63f5 AAAA mail.google.com
31	24.349563845	10.0.0.28	10.0.0.138	DNS	77 Standard query 0x62e7 A mail.google.com
32	24.349604057	fda0:110a:8c3f:0:d0...	fda0:110a:8c3f:0:16...	DNS	97 Standard query 0x62e7 A mail.google.com
33	24.349637160	10.0.0.28	10.0.0.138	DNS	77 Standard query 0x367e AAAA mail.google.com
34	24.349739264	127.0.0.1	127.0.1.1	DNS	78 Standard query 0xd261 A web.whatsapp.com
35	24.349755985	127.0.0.1	127.0.1.1	DNS	78 Standard query 0x13ac A drive.google.com
36	24.349762965	127.0.0.1	127.0.1.1	DNS	78 Standard query 0x656e AAAA drive.google.com
37	24.349769834	10.0.0.28	10.0.0.138	DNS	78 Standard query 0x6ee8 A web.whatsapp.com
38	24.349804694	10.0.0.28	10.0.0.138	DNS	78 Standard query 0x221c A drive.google.com
39	24.349828862	10.0.0.28	10.0.0.138	DNS	78 Standard query 0x96d AAAA drive.google.com
40	24.351572993	10.0.0.138	10.0.0.28	DNS	1. Standard query response 0x62e7 A mail.google.com CNAME googlemail.l.google.com A 172.217.22.101 ← Red arrow
41	24.351666044	127.0.0.1	127.0.1.1	DNS	1. Standard query response 0xb5ca A mail.google.com CNAME googlemail.l.google.com A 172.217.22.101
42	24.351950455	10.0.0.138	10.0.0.28	DNS	1. Standard query response 0x367e AAAA mail.google.com CNAME googlemail.l.google.com AAAA 2a00:1450..
43	24.351999916	127.0.0.1	127.0.1.1	DNS	1. Standard query response 0x63f5 AAAA mail.google.com CNAME googlemail.l.google.com AAAA 2a00:1450..
44	24.352266850	127.0.0.1	127.0.1.1	DNS	72 Standard query 0xeb62 A github.com
45	24.352382170	10.0.0.28	10.0.0.138	DNS	72 Standard query 0x7a0a A github.com
46	24.352429149	127.0.0.1	127.0.1.1	DNS	72 Standard query 0xcbd7 AAAA github.com
47	24.352453479	10.0.0.28	10.0.0.138	DNS	72 Standard query 0x21a4 AAAA github.com
50	24.477814709	fda0:110a:8c3f:0:16...	fda0:110a:8c3f:0:d0...	DNS	1. Standard query response 0x62e7 A mail.google.com CNAME googlemail.l.google.com A 172.217.22.101
51	24.351666044	127.0.0.1	127.0.1.1	DNS	88 Standard query 0x1ede A tiles.services.mozilla.com
52	24.351666045	10.0.0.28	10.0.0.138	DNS	88 Standard query 0x94f3 A tiles.services.mozilla.com
53	24.351694772	127.0.0.1	127.0.1.1	DNS	88 Standard query 0x2ac9 AAAA tiles.services.mozilla.com
54	24.351738918	10.0.0.28	10.0.0.138	DNS	88 Standard query 0x5719 AAAA tiles.services.mozilla.com
55	24.351828671	127.0.0.1	127.0.1.1	DNS	88 Standard query 0x48ab A tiles.services.mozilla.com
56	24.351867890	10.0.0.28	10.0.0.138	DNS	88 Standard query 0x2a82 A tiles.services.mozilla.com
57	24.352278739	10.0.0.138	10.0.0.28	DNS	1. Standard query response 0x5719 AAAA tiles.services.mozilla.com CNAME tiles.r53-2.services.mozilla.com
58	24.352282944	127.0.0.1	127.0.1.1	DNS	1. Standard query response 0x2ac9 AAAA tiles.services.mozilla.com CNAME tiles.r53-2.services.mozilla.com
59	24.3523127535	10.0.0.138	10.0.0.28	DNS	2. Standard query response 0xa282 A tiles.services.mozilla.com CNAME tiles.r53-2.services.mozilla.com
60	24.3523170849	127.0.0.1	127.0.1.1	DNS	2. Standard query response 0x48a6 A tiles.services.mozilla.com CNAME tiles.r53-2.services.mozilla.com
63	24.3527763299	10.0.0.138	10.0.0.28	DNS	2. Standard query response 0x94f3 A tiles.services.mozilla.com CNAME tiles.r53-2.services.mozilla.com
64	24.3527850313	127.0.0.1	127.0.1.1	DNS	2. Standard query response 0x1ede A tiles.services.mozilla.com CNAME tiles.r53-2.services.mozilla.com
65	25.0253116917	10.0.0.28	255.255.255.255	DB-LSP	1. Dropbox LAN sync Discovery Protocol
67	25.027410271	10.0.0.28	10.0.0.255	DB-LSP	1. Dropbox LAN sync Discovery Protocol
68	25.038594867	127.0.0.1	127.0.1.1	DNS	89 Standard query 0xc8fd A safebrowsing.googleapis.com
69	25.038670479	10.0.0.28	10.0.0.138	DNS	89 Standard query 0x1c3f A safebrowsing.googleapis.com
70	25.056169596	10.0.0.138	10.0.0.28	DNS	1. Standard query response 0x1c3f A safebrowsing.googleapis.com A 172.217.16.202
71	25.056266190	127.0.0.1	127.0.1.1	DNS	1. Standard query response 0x1c3f A safebrowsing.googleapis.com A 172.217.16.202
73	25.231800783	127.0.0.1	127.0.1.1	DNS	77 Standard query 0xead2 A apis.google.com
94	25.232875352	10.0.0.28	10.0.0.138	DNS	77 Standard query 0x6548 A apis.google.com
95	25.231981144	127.0.0.1	127.0.1.1	DNS	77 Standard query 0x4cc1 A www.gstatic.com
96	25.232822837	10.0.0.28	10.0.0.138	DNS	77 Standard query 0x1cd7 A www.gstatic.com
97	25.232349165	127.0.0.1	127.0.1.1	DNS	77 Standard query 0x4df6 A ssl.gstatic.com

החבילה המתאימה היא חבילת מס' 31, ואם נסתכל לראות מה כתובת MAC של המוצא שלו (הлокוט – המחשב שלנו) קיבל :

6c:71:d9:3b:12:61

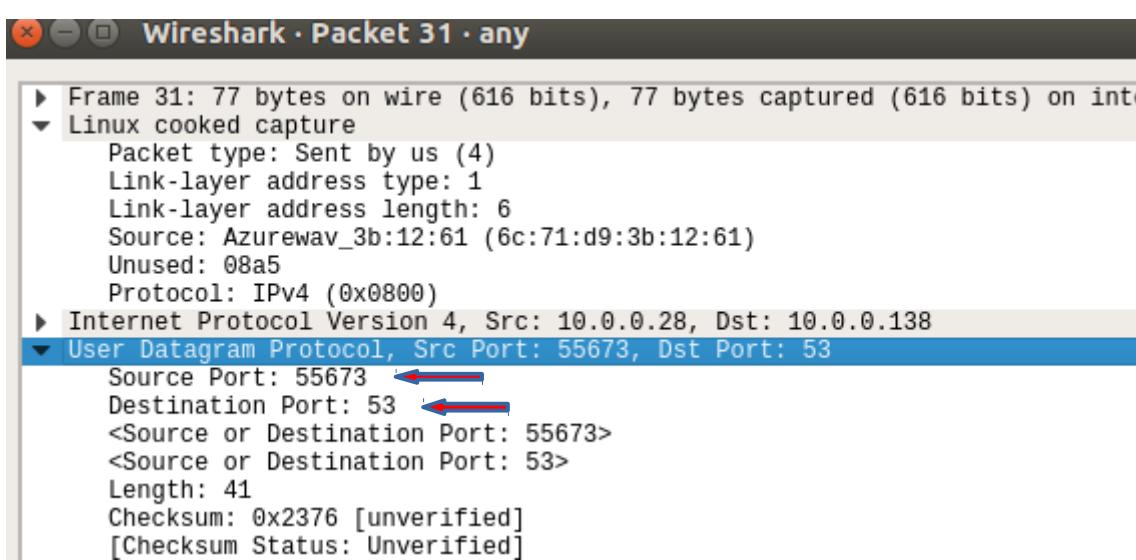
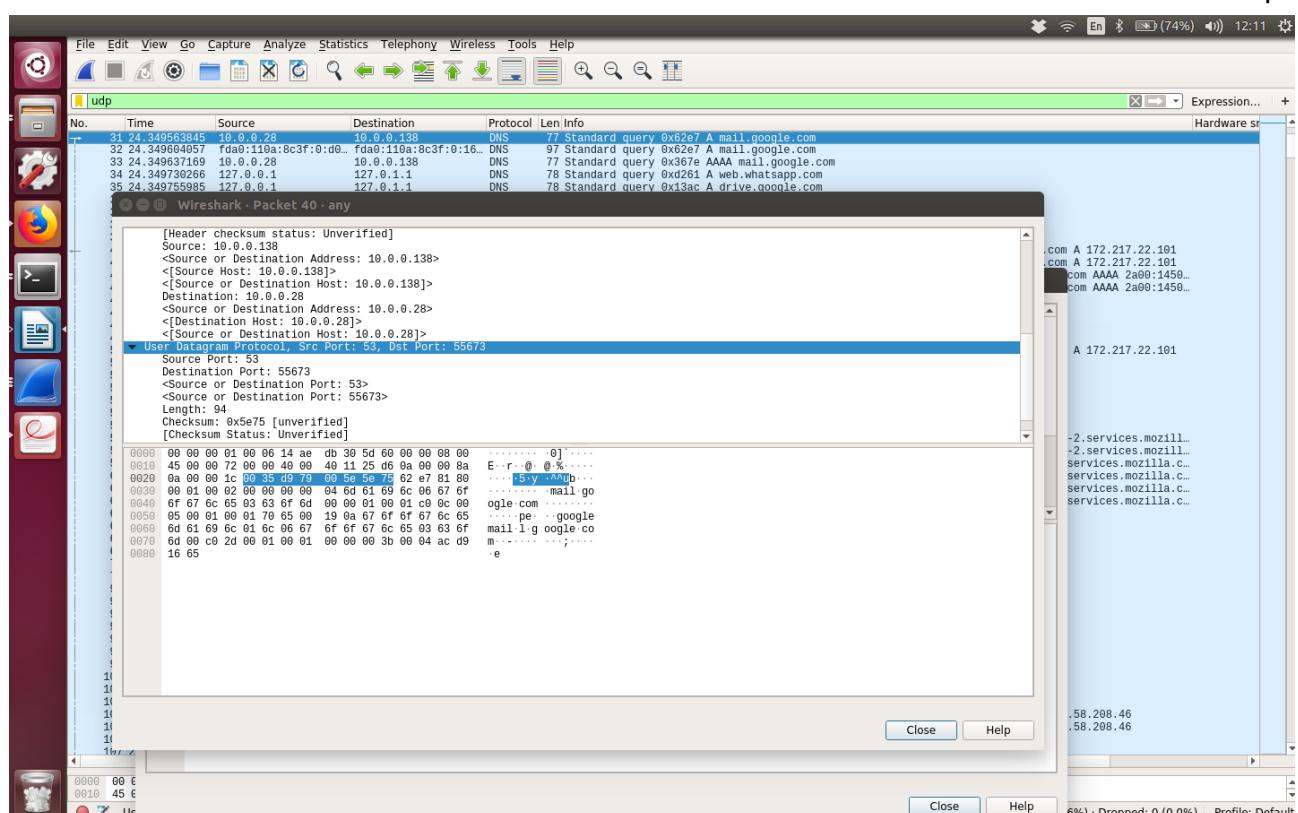
Frame 31: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface 0  
 Linux cooked capture  
 Packet type: Sent by us (4)  
 Link-layer address type: 1  
 Link-layer address length: 6  
 Source: Azurewav\_3b:12:61 (6c:71:d9:3b:12:61) ← Red arrow  
 Unused: 08a5

icut נחזור על סעיף ד רק בכיוון הփוף, כלומר עם החבילה ההפוכה, אותה הרأتي בעמוד קודם או' מצאת - חבילת מס' 31.

החבילה כמובן נשלחה מהמחשב שלנו אל השרת – כיון שגם החבילה ההפוכה שיצרה את התקשרות עם השרת, וניתן לראות זאת גם בתיאור החבילה, כתובות IP המוצא שלה הוא 10.0.0.28 וזהו המחשב שלנו, וכ כתובות היעד שלו הוא 10.0.0.138 שגם כתובות IP של השרת כפי שראינו קודם.

No.	Time	Source	Destination	Protocol	Len/Info
31	24.349563845	10.0.0.28	10.0.0.138	DNS	
32	24.349604057	10.0.0.28	10.0.0.138	DNS	

ניתן לראות שמספר פורט המוצא הינו 55673 (כפי שראינו בסעיף קודם שהוא מספר הפורט של המחשב שלנו – הלוקוח). מספר פורט היעד הוא 53 – כפי שראינו קודם שהוא מספר הפורט של השרת – היעד במקורה זה.



כתובת IP של השולח (הלקוט - המחשב שלנו) הינה 10.0.0.28 וכותבת ה IP של המქבל (השרת במקורה זה) הינה 10.0.0.138  
ניתן לראות זו באותה תמונה לעיל בשכבה הרשת -

```

Frame 31: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface
└─ Linux cooked capture
    └─ Internet Protocol Version 4, Src: 10.0.0.28, Dst: 10.0.0.138
        └─ User Datagram Protocol, Src Port: 55673, Dst Port: 53
            Source Port: 55673
            Destination Port: 53
            <Source or Destination Port: 55673>
            <Source or Destination Port: 53>
            Length: 41
            Checksum: 0x2376 [unverified]
            [Checksum Status: Unverified]
    
```

כתובת MAC של השולח הינה :

6c:71:d9:3b:12:61

ניתן לראות זאת בדיק באמצעות תמונה לעיל :

```

Frame 31: 77 bytes on wire (616 bits), 77 bytes captured (616 bits) on interface
└─ Linux cooked capture
    └─ Internet Protocol Version 4, Src: 10.0.0.28, Dst: 10.0.0.138
        └─ User Datagram Protocol, Src Port: 55673, Dst Port: 53
            Source Port: 55673
            Destination Port: 53
            <Source or Destination Port: 55673>
            <Source or Destination Port: 53>
            Length: 41
            Checksum: 0x2376 [unverified]
            [Checksum Status: Unverified]
    
```

את כתובת MAC של המქבל ניתן למצוא בפרט החבילה הפוכה בכתובת המוצא MAC שהוא :  
[14:ae:db:30:5d:60](#)

מה שמצאנו קודם :

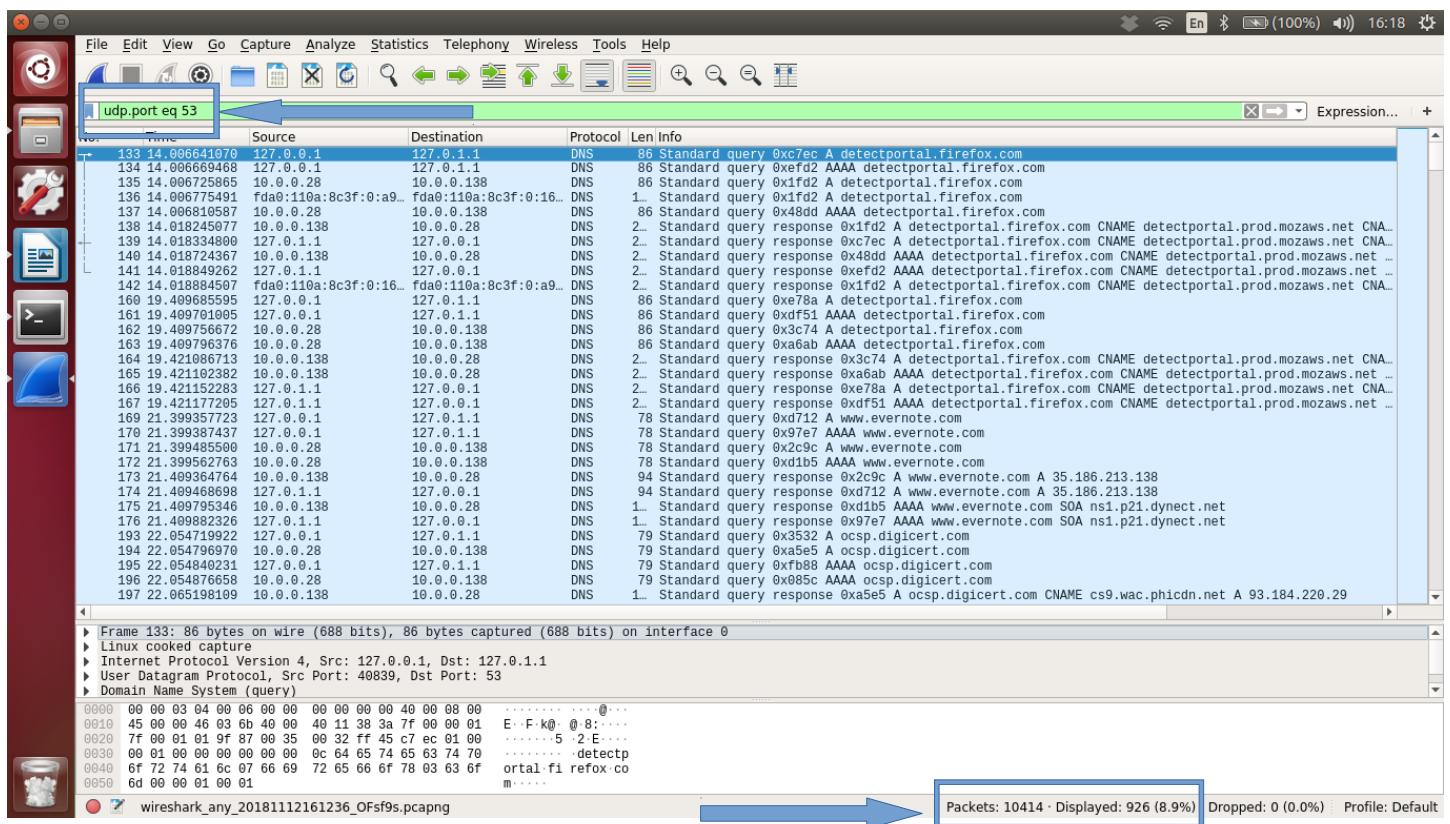
```

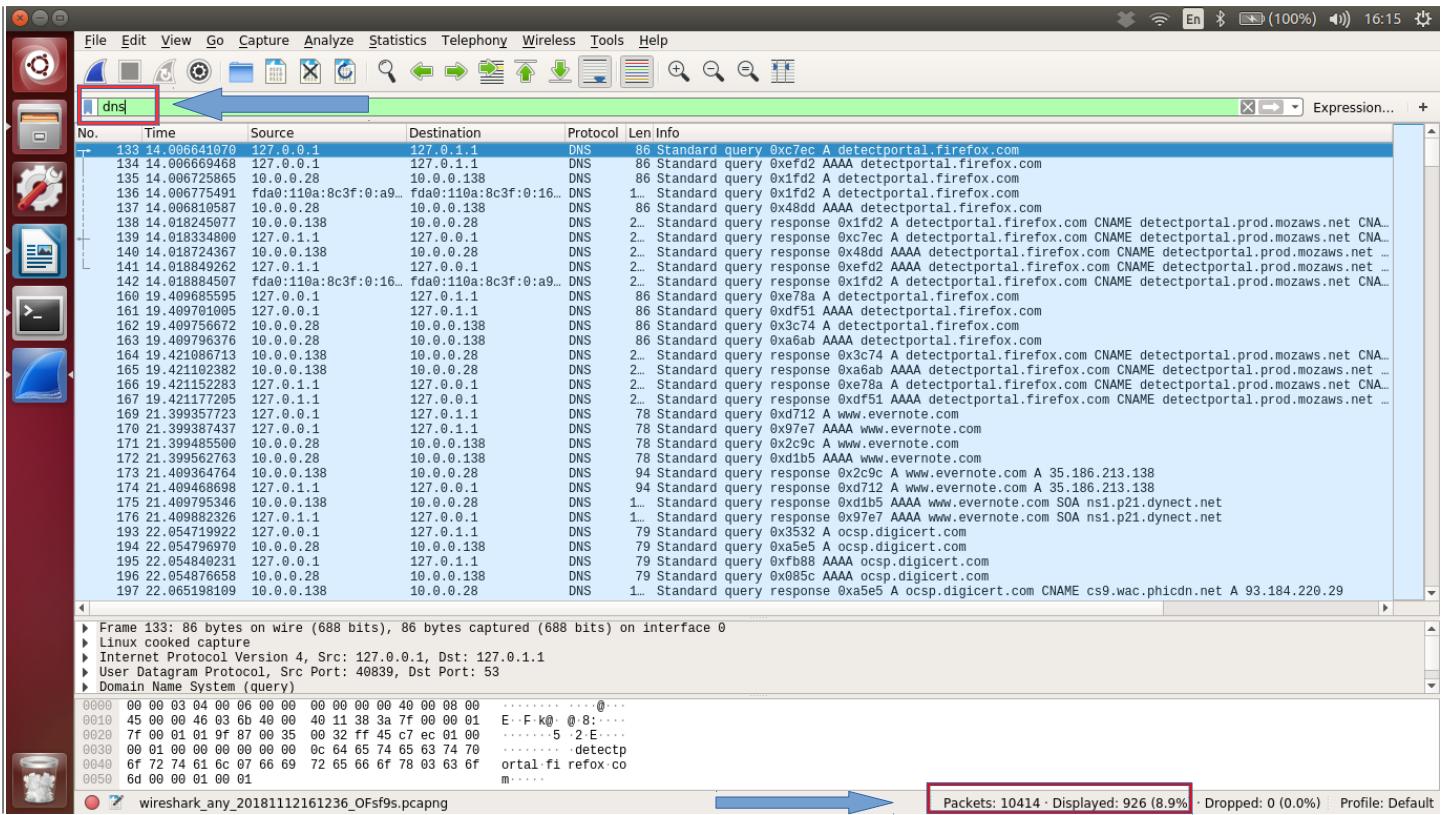
Frame 40: 130 bytes on wire (1040 bits), 130 bytes captured (1040 bits) on interface 0
└─ Linux cooked capture
    └─ Internet Protocol Version 4, Src: VtechTel_30:5d:60 (14:ae:db:30:5d:60)
        └─ User Datagram Protocol, Src Port: 55673, Dst Port: 53
            Source Port: 55673
            Destination Port: 53
            <Source or Destination Port: 55673>
            <Source or Destination Port: 53>
            Length: 41
            Checksum: 0x2376 [unverified]
            [Checksum Status: Unverified]
    
```

סעיף ז':

על ידי סינון החבילות שהמחשב שלנו קיבל על גבי פרוטוקול DNS ניתן לראות שתמיד מסטר פורט המוצא הינו 53, لكن אנחנו יכולים להסיק שרשת DNS מАЗין בפורט 53.

סעיף ז': נוכל לסנן לפי מסטר הפורט, כיוון שאנחנו יודעים ש DNS עובד עם פורט 53, נוכל להכנס לשורת הסינון תנאי שדורש רק חבילות שנשלחו מפורט 53 ו/או התקבלו מפורט 53. התנאי שנכנסים לשורת הסינון יהיה udp.port.53.קפס וכך נקבל אך ורק חבילות שנשלחו על גבי פרוטוקול DNS. נשים לב שאכן אם נכנס בשורת הסינון DNS נקבל בדיק את אותן הפקטות.





ואכן ניתן לראות שקיבלנו את אותן החבילות (אותו מספר של פקודות מוצגות!)

### שאלה 3

צרכנו שתי מכונות וירטואליות עבור השירותים (האב והבן) ועבור הלקוח.  
ניתן לראות שכתובת ה IP של השירותים הינה 10.0.2.5

```
client@client-VirtualBox: ~/Desktop/code/fwd1
client@client-VirtualBox:~/Desktop/code/fwd1$ ifconfig
enp0s3    Link encap:Ethernet HWaddr 08:00:27:52:83:fe
          inet addr:10.0.2.5 Bcast:10.0.2.255 Mask:255.255.255.0
          inet6 addr: fe80::2dd7:dec:e901:f1c2/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:30198 errors:0 dropped:0 overruns:0 frame:0
            TX packets:15164 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:37284270 (37.2 MB) TX bytes:2598079 (2.5 MB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:65536 Metric:1
            RX packets:5846 errors:0 dropped:0 overruns:0 frame:0
            TX packets:5846 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:386446 (386.4 KB) TX bytes:386446 (386.4 KB)

client@client-VirtualBox:~/Desktop/code/fwd1$
```

בתמונה השנייה ניתן לראות שכתובת ה IP של הלקוח היא 10.0.2.4

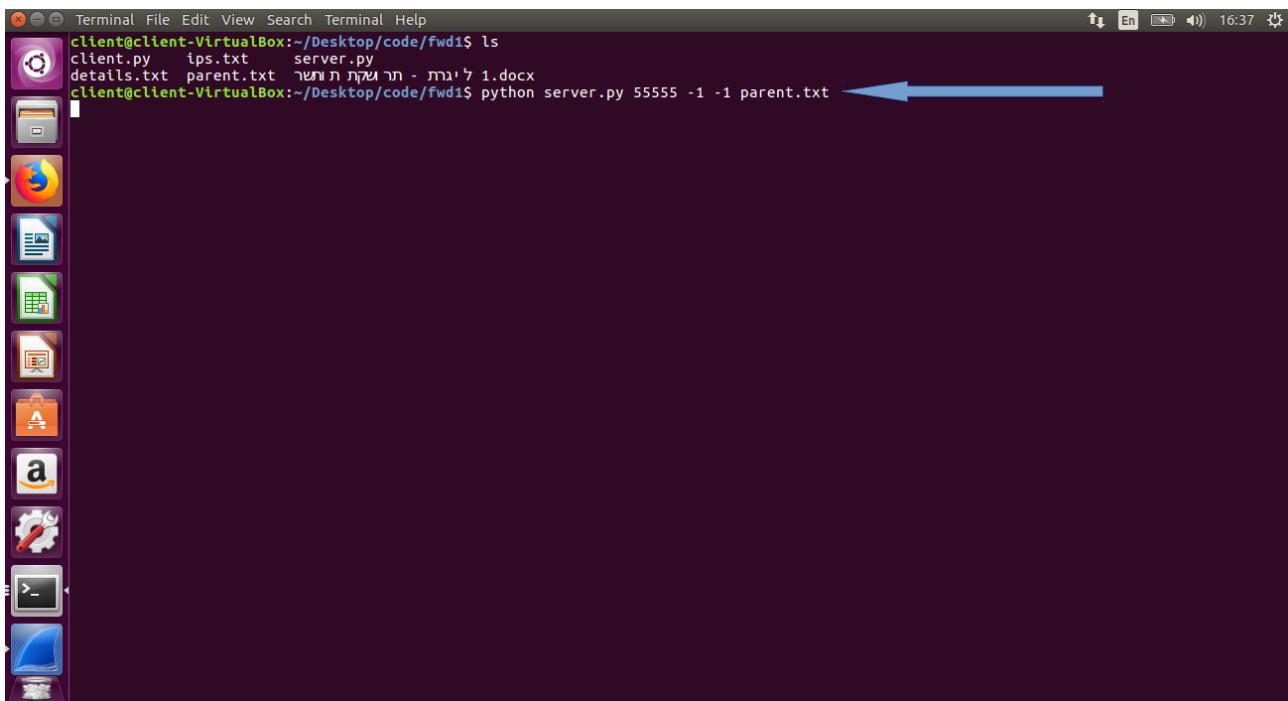
```
Terminal
client@client-VirtualBox: ~/Desktop/code
client@client-VirtualBox:~/Desktop/code$ ifconfig
enp0s3    Link encap:Ethernet HWaddr 08:00:27:92:e7:39
          inet addr:10.0.2.4 Bcast:10.0.2.255 Mask:255.255.255.0
          inet6 addr: fe80::1e04:fd72:f3a0:aca8/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:861 errors:0 dropped:0 overruns:0 frame:0
            TX packets:845 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:76301 (76.3 KB) TX bytes:69967 (69.9 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:65536 Metric:1
            RX packets:1652 errors:0 dropped:0 overruns:0 frame:0
            TX packets:1652 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:102980 (102.9 KB) TX bytes:102980 (102.9 KB)

client@client-VirtualBox:~/Desktop/code$
```

כעת נתאר את הרכבת הסקריפט שמאתחל את שרת האב והבן, ניתן לראות בתמונה כי הגדרנו את הפורט 55555 עבור שרת האב ואת הפורט 22222 עבור שרת הבן.

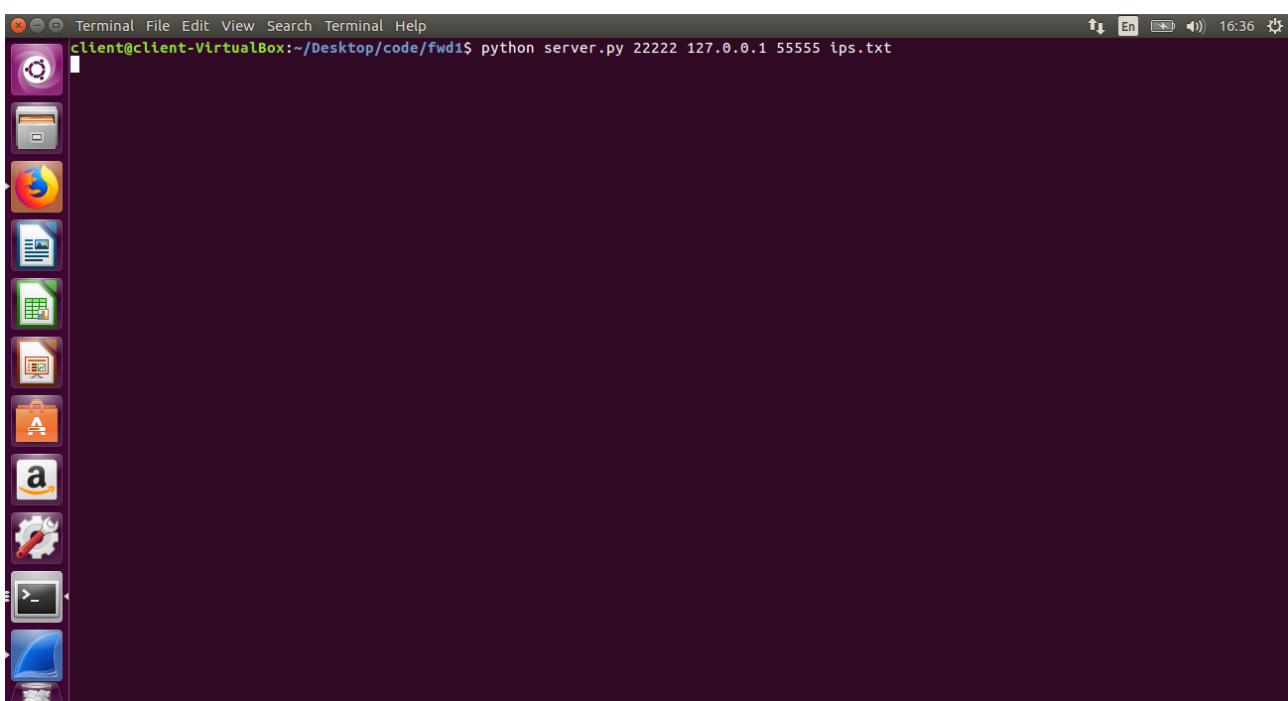
יצירת שרת האב על מכונה וירטואלית של השירותים:



```
client@client-VirtualBox:~/Desktop/code/fwd1$ ls
client.py  ips.txt  server.py
details.txt  parent.txt  1.docx
client@client-VirtualBox:~/Desktop/code/fwd1$ python server.py 55555 -1 -1 parent.txt
```

A screenshot of a terminal window titled "Terminal". The window shows a command-line interface with a dark background and light-colored text. The command entered is "python server.py 55555 -1 -1 parent.txt". A blue arrow points from the right side of the terminal window towards the command line, highlighting it. The terminal window has a standard window title bar with icons for minimize, maximize, and close, and a status bar at the top right showing "16:37". On the left side of the terminal, there is a vertical dock containing icons for various applications like a file manager, browser, and terminal.

יצירת שרת הבן על המכונה הווירטואלית של השירותים:

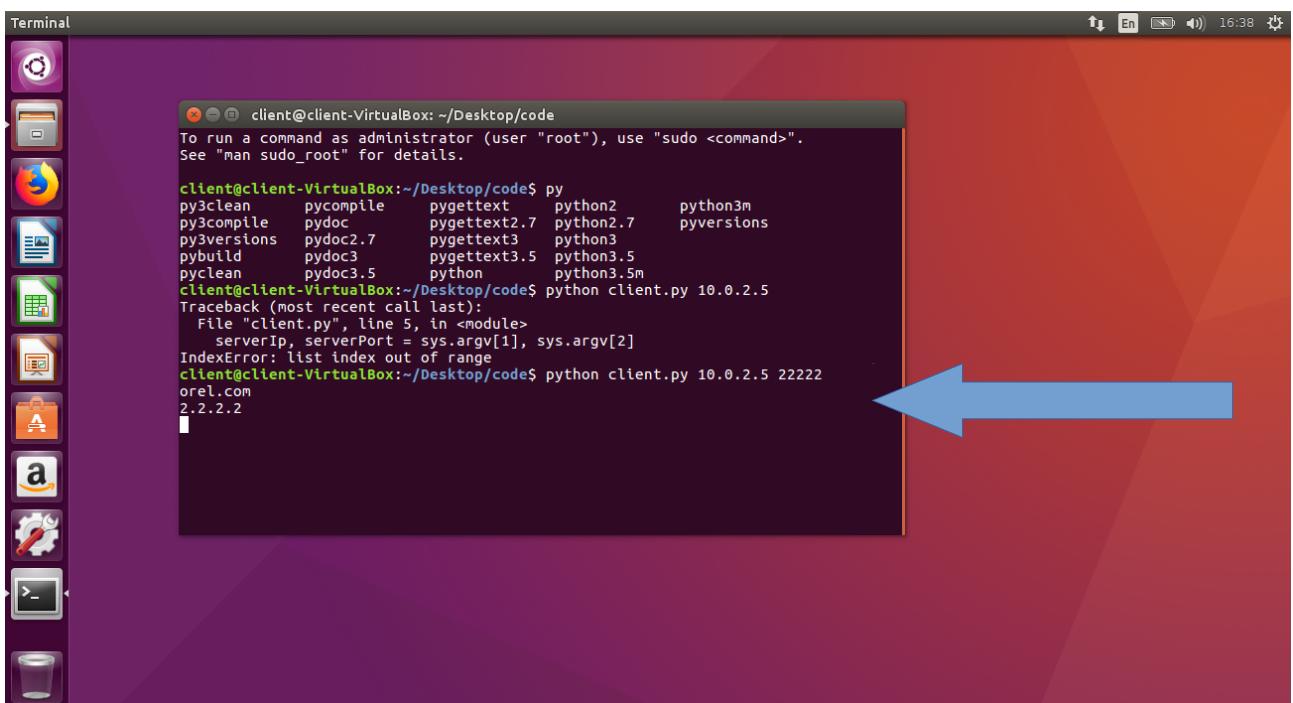


```
client@client-VirtualBox:~/Desktop/code/fwd1$ python server.py 22222 127.0.0.1 55555 ips.txt
```

A screenshot of a terminal window titled "Terminal". The window shows a command-line interface with a dark background and light-colored text. The command entered is "python server.py 22222 127.0.0.1 55555 ips.txt". The terminal window has a standard window title bar with icons for minimize, maximize, and close, and a status bar at the top right showing "16:36". On the left side of the terminal, there is a vertical dock containing icons for various applications like a file manager, browser, and terminal.

כעת נראה כיצד אתחלנו את הלקוח – על ידי הרצת סקריפט הקל'ינט, אשר מקבל כארגומנטים את כתובות ה IP ומספר הפורט של שרת הבן:  
 בנוסף ניתן לראות כי הכנסנו אינפוט של כתובות כלשהי שלא מופיעה במילון של שרת הבן (ips.txt)  
 וכן הוא שרת הבן פנה אל שרת האב על מנת לקבל את המיפוי כפי שנראה בהמשך.

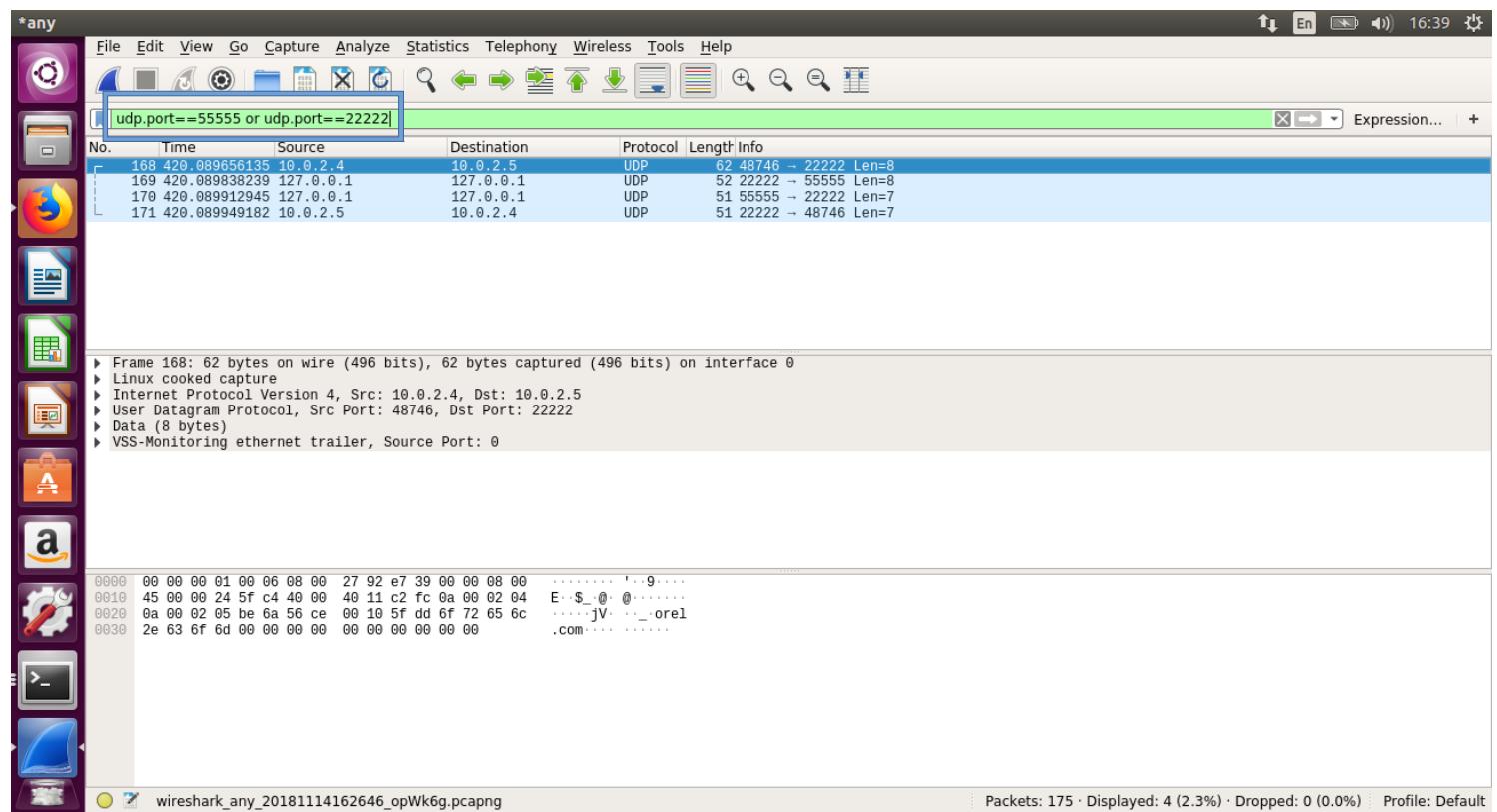
במקרה זה השתמשנו בlienק : oreI.com



נתאר כעת את סדר הפעולות בצורה מופשטת כפי שנראה בהמשך על ידי Wireshark.

- I. הלקוח פונה אל שרת הבן בבקשת לקלול כתובות IP עבור כתובות מסוימת
- II. מכיוון שלשרת הבן אין את כתובות ה IP במילון המיפוי, הוא פונה אל שרת האב.
- III. שרת האב מחזיר את תשובתו אל שרת הבן השומר את הכתובת החדשה אל תוך המילון מיפוי כתובות
- IV. שרת הבן מחזיר את התשובה ללקוח

נסתכל כעת על הנתונים בוירשאך עבור פעולה זו (בשורת הפליטור הכנסנו תנאי שיציג אך ורק את הפקודות שמעניינות אותנו – את הפקודות שהו בקשר בין השירותים ללקוח)  
 ניתן לראות שפעולה זו גרמה לארבע הנסנות כפי שתיארנו לעיל. נפרט כעת עבור כל פקטה ופקטה.



נתבונן ב**פקטה הראשונה**, המתארת את שליחת בקשת הליקוי אל שרת הבן.  
כפי שהסבירנו קודם הליקוי מבקש את כתובות ה IP של האתר oreI.com משרת הבן.  
אנו ידועים שאכן הליקוי פונה לשרת ציון שכותבת המוצא בהינה כתובת הליקוי : 10.0.2.4  
וכותבתה יעד היא הכתובת IP של שרת הבן : 10.0.2.5

עתה אנו בשכבות האפליקציה שכידע תקפידה הינו לסדר את המידע באופן כזה שהמחשב בצד  
המקבל יוכל לפענח את המידע.

בתמונה רואים את תוכן הודעה שהלkoת שלח לשרת הבן:

Wireshark screenshot showing captured traffic. The search bar at the top contains the expression `udp.port==55555 or udp.port==22222`. The fourth packet in the list is highlighted with a red arrow. The packet details pane shows the following information:

No.	Time	Source	Destination	Protocol	Length	Info
168	420.089656135	10.0.2.4	10.0.2.5	UDP	62	48746 → 22222 Len=8
169	420.089838239	127.0.0.1	127.0.0.1	UDP	52	22222 → 55555 Len=8
170	420.089912945	127.0.0.1	127.0.0.1	UDP	51	55555 → 22222 Len=7
171	420.089949182	10.0.2.5	10.0.2.4	UDP	51	22222 → 48746 Len=7

The packet bytes pane shows the raw data starting with `E..$.@. @.....jV...orel.com.....`.

אנחנוCut בשכבת התעבורה האחראית לביצוע וידוא שה-data יגיע אל האפליקציה הנcona מבין כל האפליקציות במחשב היעד. היא עשויה זאת על ידי שימוש במספר הפורט יעד ופורט המוצא. אך ניתן לראות שמספר פורט היעד הינו תואם למספר הפורט שהגדכנו לשרת האב : 55555 ומספר פורט המוצע הינו נבחר באופן אקראי על ידי המחשב של ה-client.

Wireshark screenshot showing captured traffic. The search bar at the top contains the expression `udp.port==55555 or udp.port==22222`. The fourth packet in the list is highlighted with a red arrow. The packet details pane shows the following information:

No.	Time	Source	Destination	Protocol	Length	Info
168	420.089656135	10.0.2.4	10.0.2.5	UDP	62	48746 → 22222 Len=8
169	420.089838239	127.0.0.1	127.0.0.1	UDP	52	22222 → 55555 Len=8
170	420.089912945	127.0.0.1	127.0.0.1	UDP	51	55555 → 22222 Len=7
171	420.089949182	10.0.2.5	10.0.2.4	UDP	51	22222 → 48746 Len=7

The packet bytes pane shows the raw data starting with `E..$.@. @.....jV...orel.com.....`.

כעת אנחנו בשכבה הרשת שאחראית לדאג שהחbillה תגע אל מחשב היעד הסופי.  
אנו רואים זאת בתמונה הבאה , שהר' כתובת המקור תואמת לכתובת ה IP של מחשב הלוקוט (10.0.2.4) וכותבת ה IP של היעד היא היא כתובת ה IP של שרת הבן (10.0.2.5)

No.	Time	Source	Destination	Protocol	Length	Info
168	420.089656135	10.0.2.4	10.0.2.5	UDP	62	48746 → 22222 Len=8
169	420.089838239	127.0.0.1	127.0.0.1	UDP	52	22222 → 55555 Len=8
170	420.089912945	127.0.0.1	127.0.0.1	UDP	51	55555 → 22222 Len=7
171	420.089949182	10.0.2.5	10.0.2.4	UDP	51	22222 → 48746 Len=7

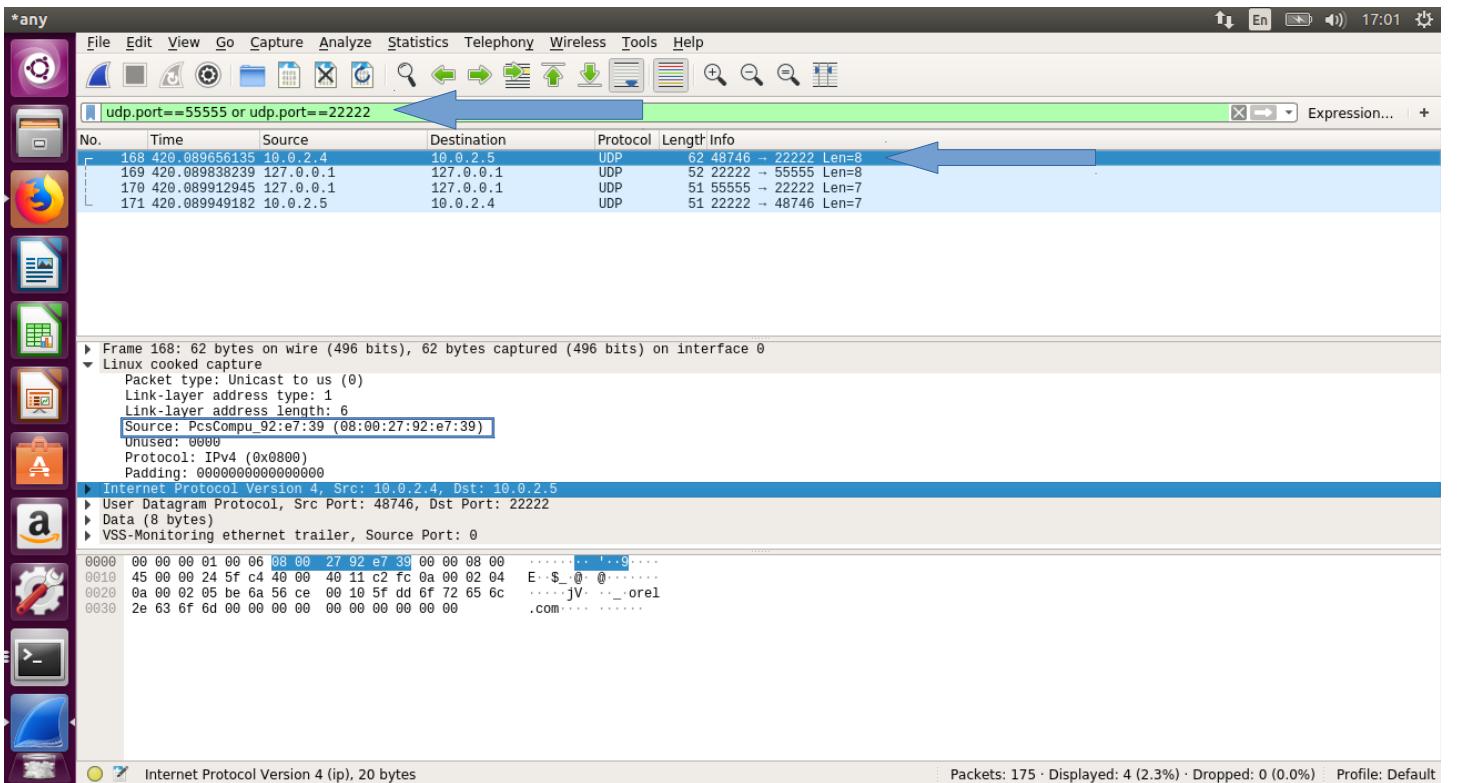
```

Frame 168: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0
  ▷ Linux cooked capture
  ▷ Internet Protocol Version 4 Src: 10.0.2.4, Dst: 10.0.2.5
  ▷ User Datagram Protocol, Src Port: 48746, Dst Port: 22222
  ▷ Data (8 bytes)
  ▷ VSS-Monitoring ethernet trailer, Source Port: 0

0000  00 00 00 01 00 06 08 00  27 92 e7 39 00 00 08 00  ....9...
0010  45 00 00 24 5f c4 40 00  40 11 c2 fc 0a 00 02 04  E-$@ @.....
0020  0a 00 02 05 be 6a 56 ce  00 10 5f dd 6f 72 65 6c  ...JV...orel
0030  2e 63 6f 6d 00 00 00 00  00 00 00 00 00 00 00 00 .com.....

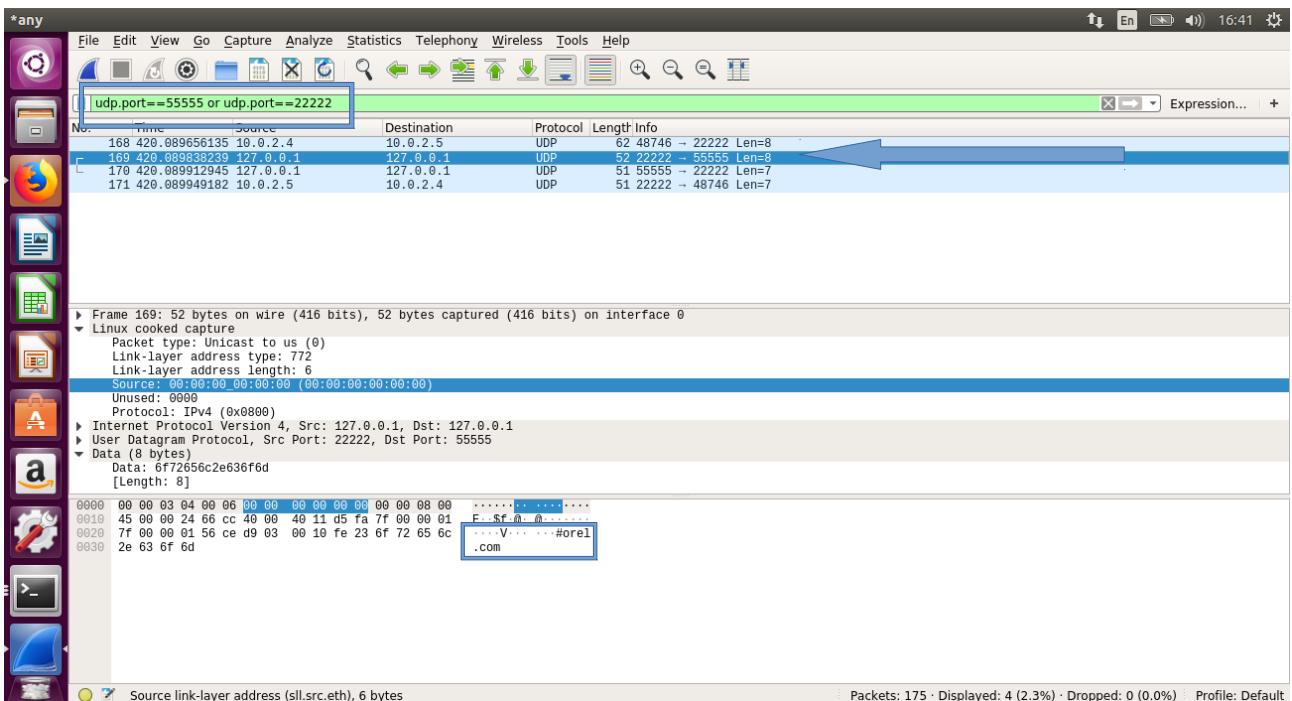
```

אנו כעת בשכבה הערוץ , אשר אחראית על ויזוא הגעת החbillה אל התחנה הקרובה ביותר הבאה (בדרכה לעד הסופי).  
אנו רואים בתמונה הבאה את הכתובת הפיזית (MAC) של המוצא - קלומר הכתובת הפיזית האחרונה שבאה בירקה החbillה בדרכה על היעד הסופי.

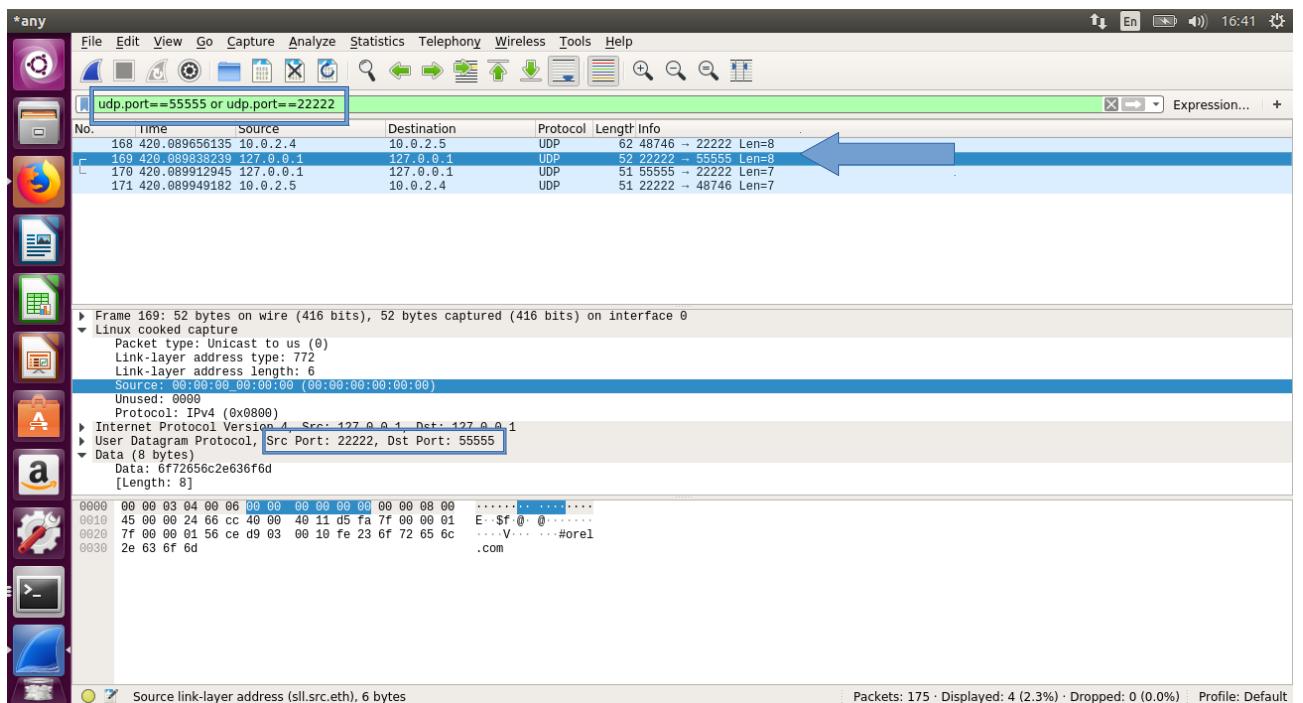


### נתבון בפקטה השנייה :

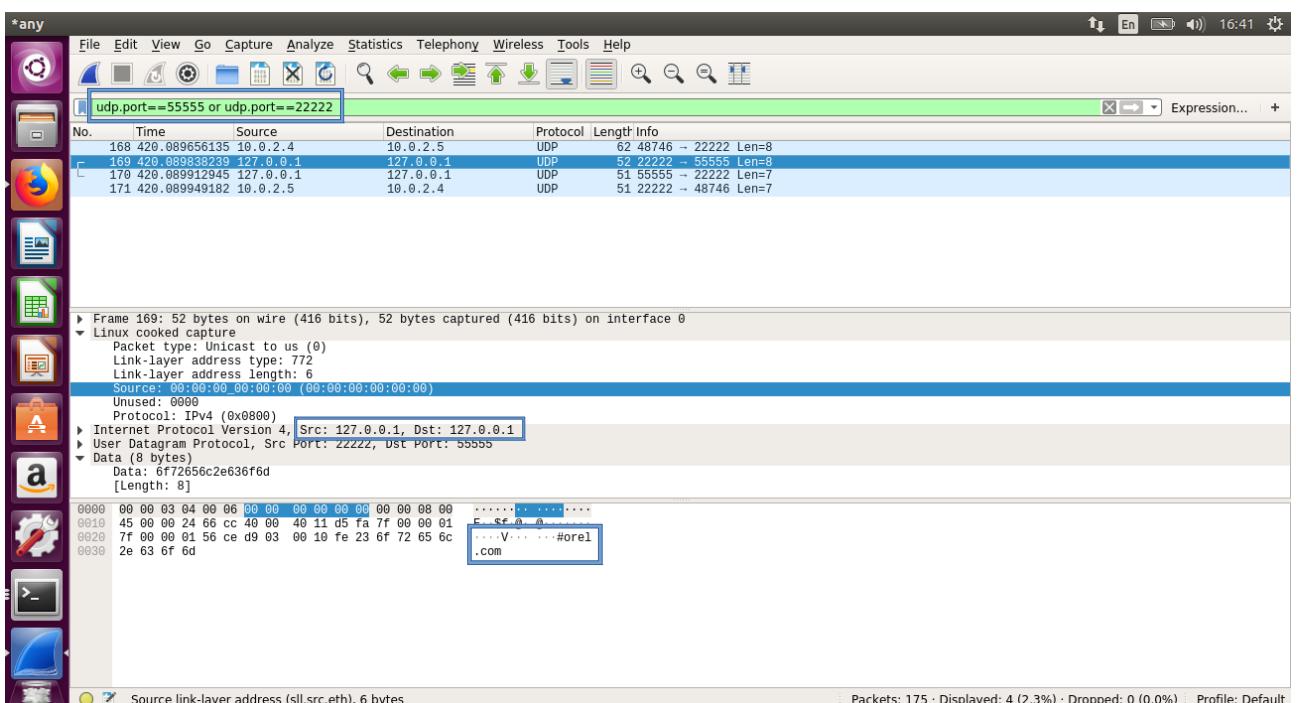
פקטה זו מတارد את המצב בו שרת הבן מבקש משרת האב את כתובת ה IP של האתר הנדרש.icut שוב, אנחנו בשכבה האפליקציה שcidou תקיפה הינו לסדר את המידע באופן זהה שהמחשב בצד המתקבל יכול לפעול את המידע. והפעם בתמונה רואים את תוכן הודעה ששרת הבן שולח לשרת האב.



אנחנוicut כעת בשכבה התעבורה האחראית לבצע וידוא שה-data יגיע אל האפליקציה הנcona מבין כל האפליקציות במחשב היעד. היא עשוה זאת על ידי שימוש במספר הפורט יעד ופורט המוצא. הפעם ניתן לראות בתמונה הבאה שמספר פורט המוצא תואם למספר הפורט שהגדכנו לשרת הבן – 22222 ומספר הפורט של היעד תואם למספר הפורט שהגדכנו לשרת האב 55555

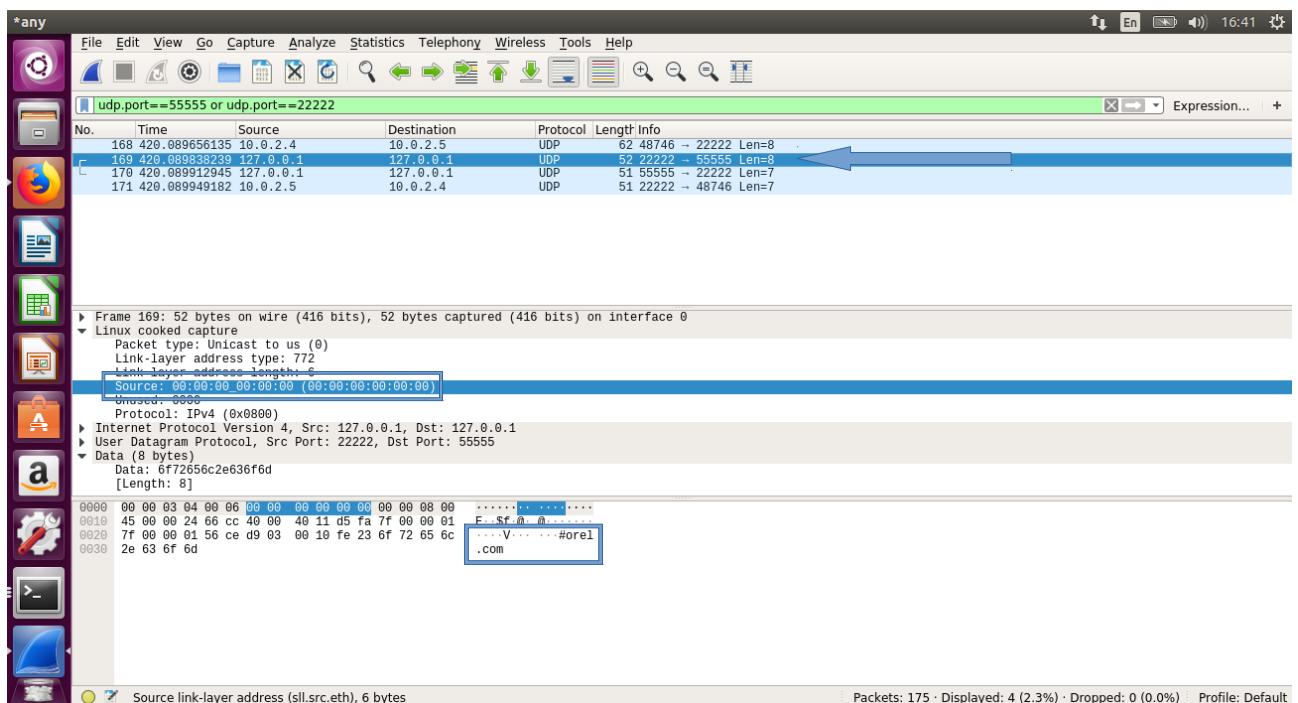


כעת אנחנו בשכבה הרשות שאחראית לדאג שהחbillה תגיע אל מחשב היעד הסופי. אנחנו רואים זאת בתמונה הבאה, שהרי כתובת המקור תואמת לכתובת ה IP של שרת הבן (127.0.0.1) וכותבת ה IP של היעד היא היא כתובות ה IP של שרת האב (127.0.0.1) ניתן לראות שכותבות ה IP זהות (הлокאלHOST) וזאת מכיוון שששת ה אפליקציות מתקשרות על אותו מחשב, ולכן נדרש לנו רק פורטים שונים על מנת ליצור תקשורת תקינה.



אנו כעת שוב בשכבה הערוץ, אשר אחראית על וידוא הגעת החbillה אל התחנה הקרובה ביותר הבאה (בדרכך) בדרך ליעד הסופי.

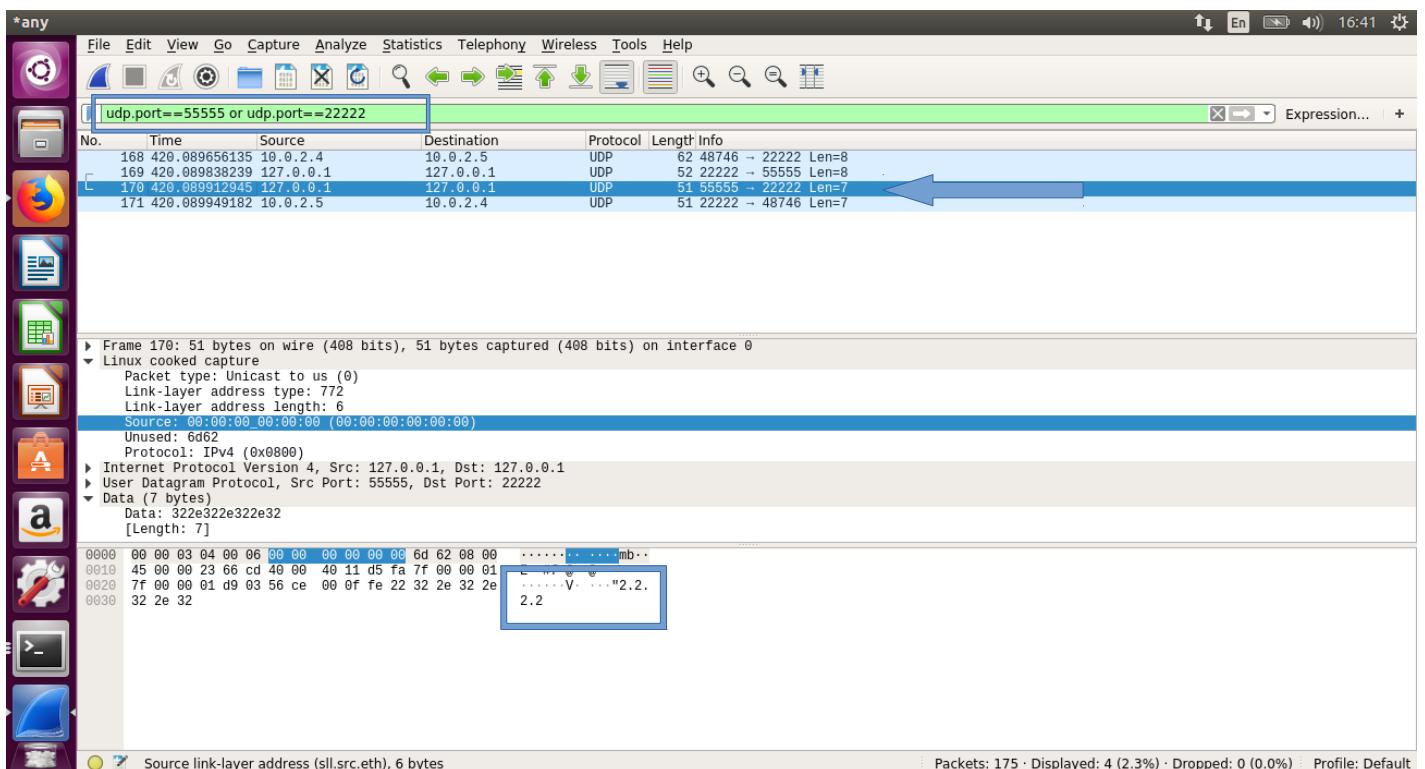
אנחנו רואים בתמונה הבאה את הכתובת הפיזית (MAC) של המזאא וכותבת הפיזית של היעד. נשים לב שהן שתיהן זהות וערוך הוא 00:00:00 שזו הכתובת MAC שמייצגת בקשת ARP, פרוטוקול שתפקידו לארח כתובת MAC של תחנה ברשע על פי כתובת IP שלה.



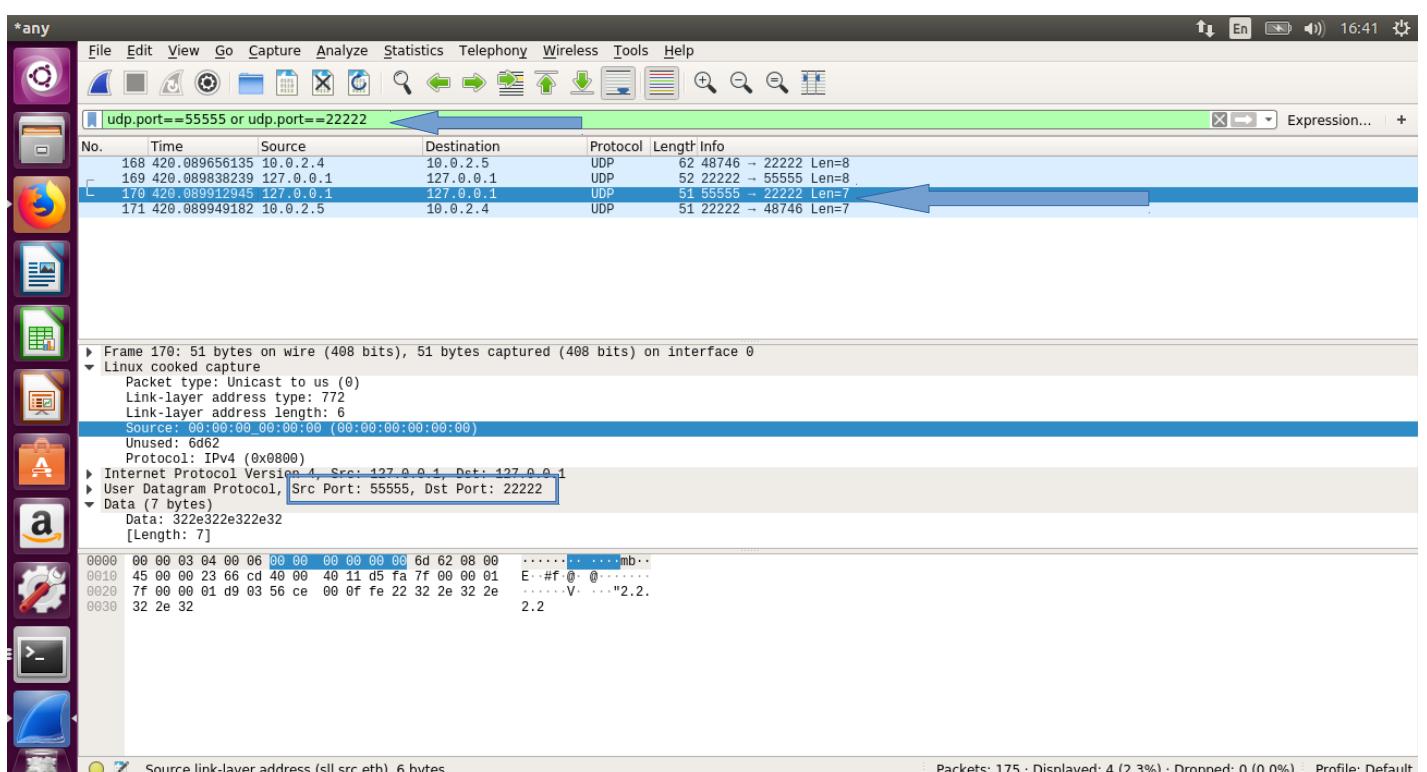
### cut נתבן בפקטה השלישית:

פקטה זו מייצגת את המצב בו שרת האב מוחזר את תשובות (כתובת IP) אל שרת הבן.

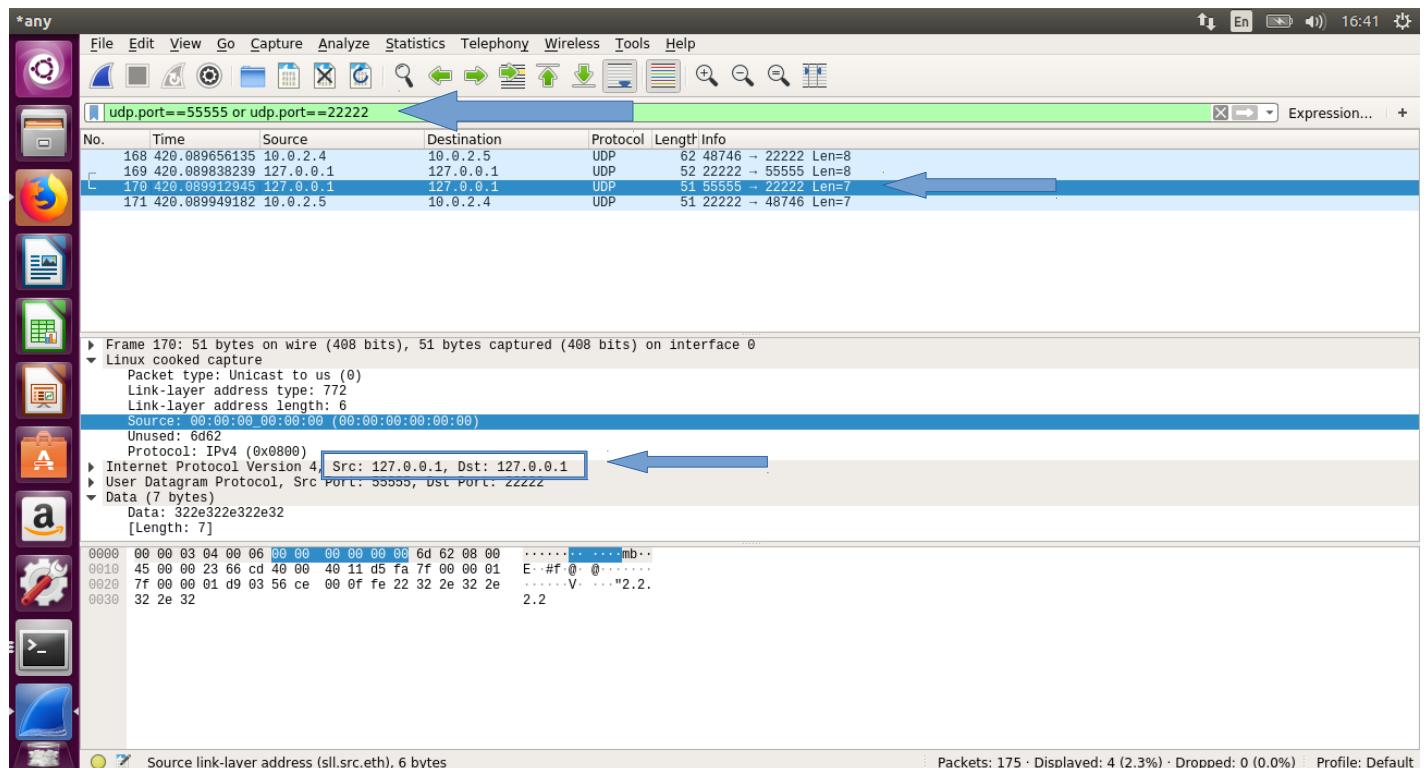
cut אנו בשכבה האפליקציה שכךיעו תקPIDה הינו לסדר את המידע באופן זה שהמחשב באצד המתקבל יכול לפעול את המידע. והפעם בתמונה רואים את תוכן הודעה ששרת האב שלוח לשרת הלוקו.



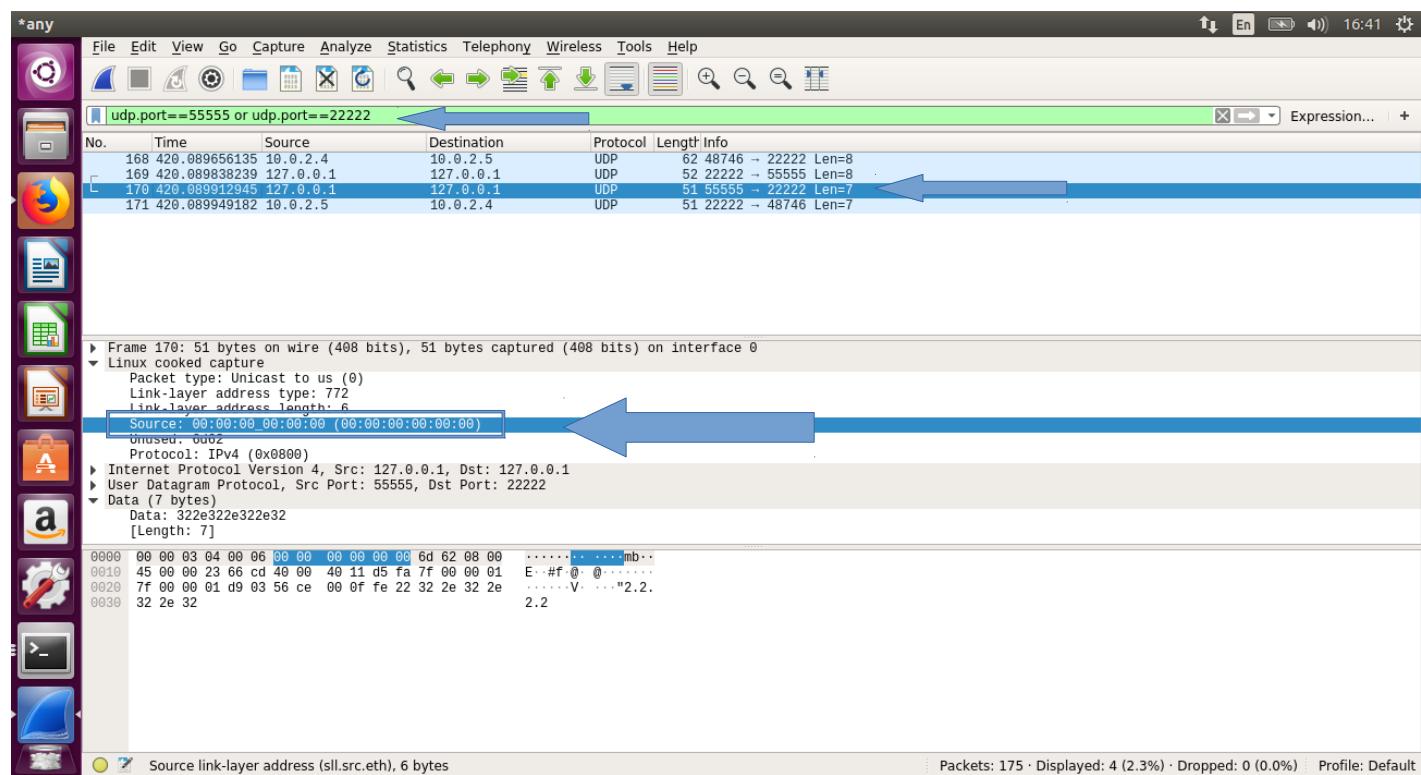
אנו כעת בשכבה התעבורה האחראית לביצוע וידוא שה-data הגיע אל האפליקציה הנכונה מבין כל האפליקציות במחשב היעד. היא עשויה זאת על ידי שימוש במספר הפורט יעד ופורט המוצא. הפעם ניתן לראות בתמונה הבאה שמספר פורט המוצא תואם למספר הפורט שהגדכנו לשרת האב – 55555 ומספר הפורט של היעד תואם למספר הפורט שהגדכנו לשרת הבן 22222



כעת אנחנו בשכבה הרכשת שאחראית לדאג שהחbillה תגיע אל מחשב היעד הסופי.  
אנו רואים זאת בתמונה הבאה , שהרוי כתובות המקור תואמת לכתובת ה IP של שרת האב (127.0.0.1) וככתובת ה IP של היעד היא היא כתובות ה IP של שרת הבן (127.0.0.1)  
ניתן לראות באפ"ו דומה לפאקטה שנייה ששכתובות ה IP זהות (הлокלhost) וזאת מכיוון ששתי האפליקציות מתקשרות על אותו מחשב, ולכן נדרש להן רק פורטים שונים על מנת ליצור תקשורת תקינה.



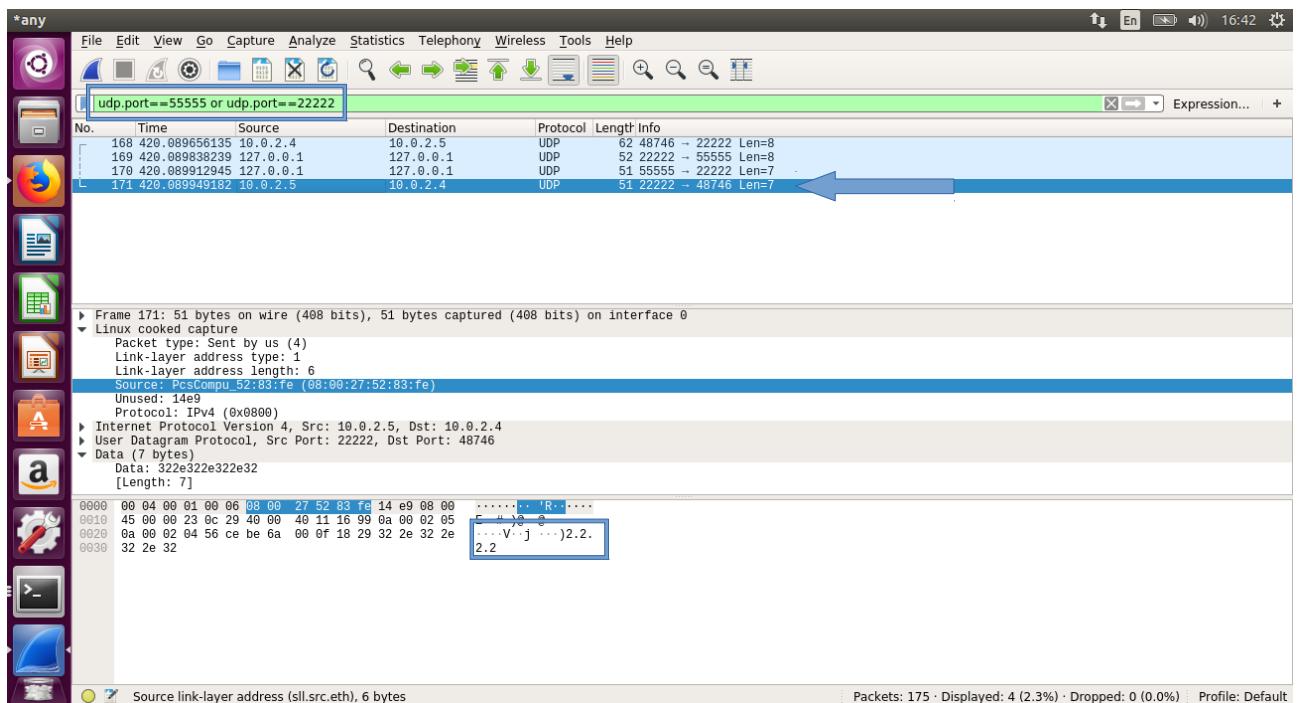
אנו רואים בתמונה הבאה את הכתובת הפיזית (MAC) של המוצא (שרת האב) וככתובת הפיזית של היעד (שרת הבן). נשים לב שהן שתיهن זהות וערך ה 00:00:00:00:00:00 שזוהה כתובות ה MAC שמייצגת בקשה ARP, פרוטוקול שתפקידו לאתר כתובת MAC של תחנה ברשף על פי כתובת ה IP שלה.



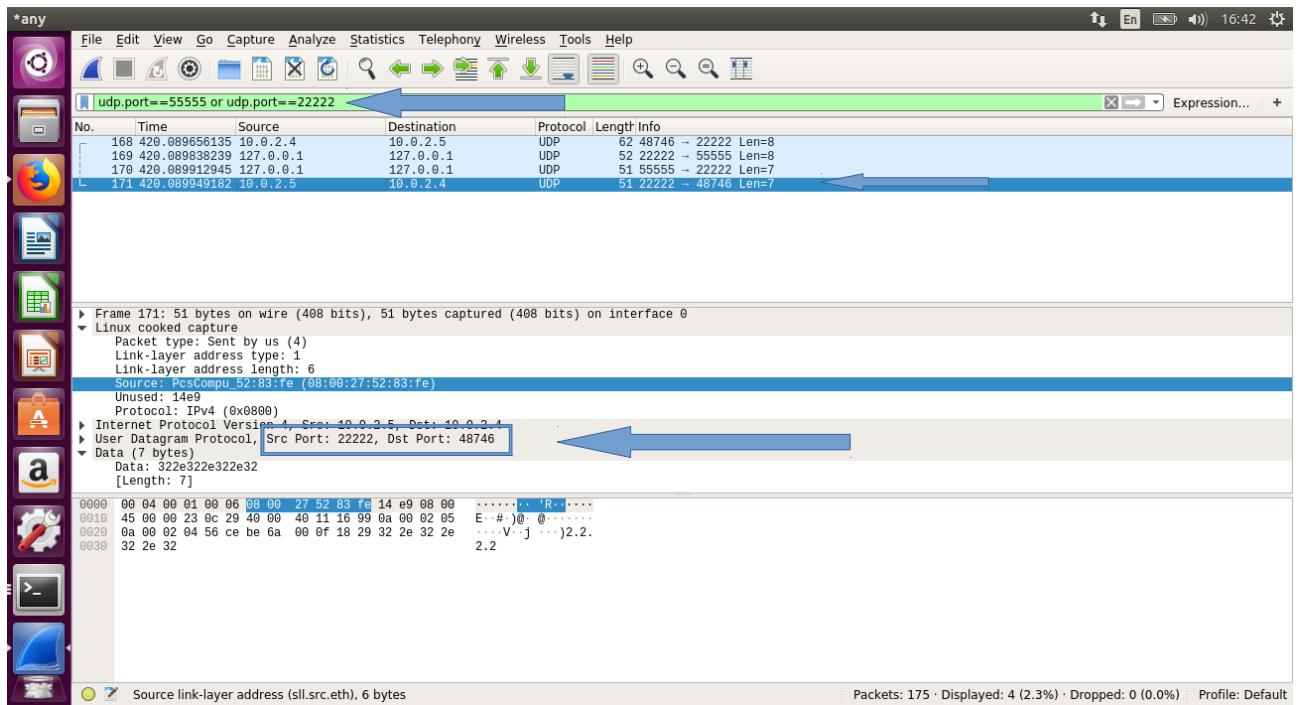
### נתבון כעת על **פקטה הרביעית**.

פקטה זו מ"יצגת את המצב בו שרת הבן מוחזיר את המידע אל הלקוט, לאחר ששמר את כתובות ה-IP החדש במיילון המיפוי שלו. מעתה ואילך כאשר הלקוט יבקש את כתובות ה-IP של כתובות זו שרת הבן לא יאלץ לפנות אל שרת האב כיון שהכתובת תהיה שמורה אצלו במיילון המיפוי כתובות.

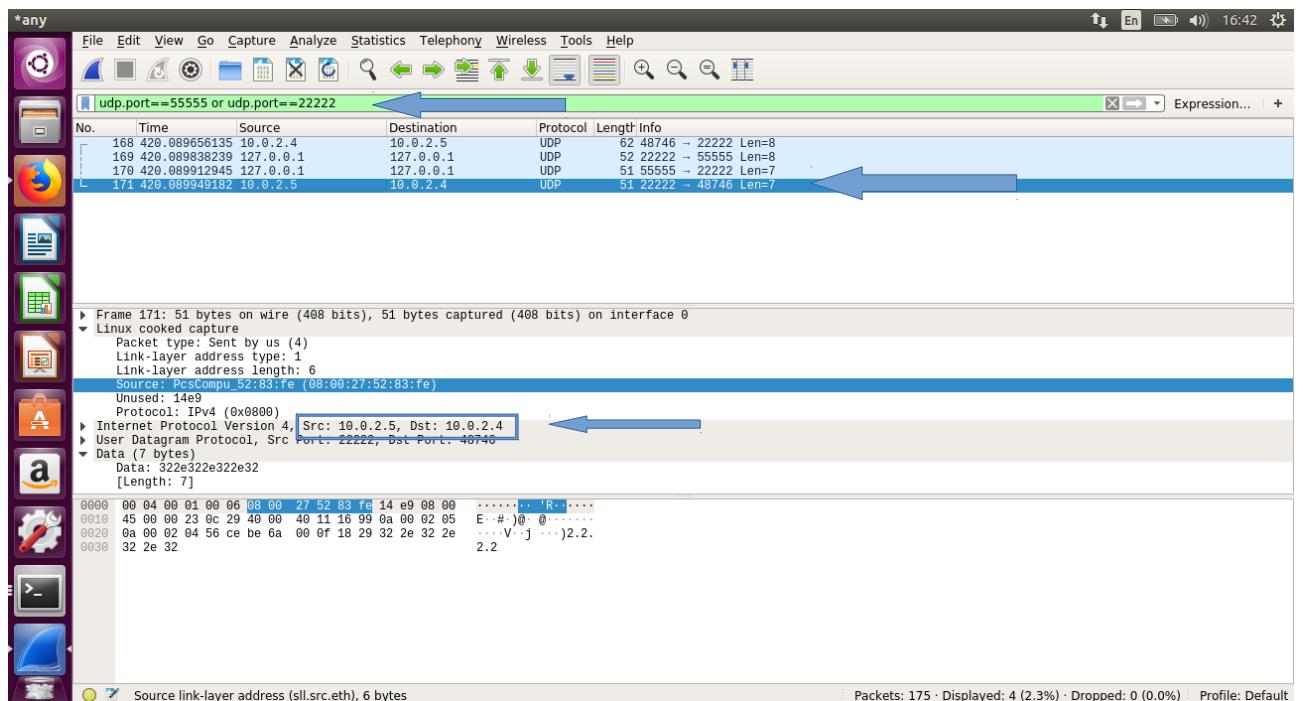
כעת אנחנו בשכבה האפליקציה שcidou תקפידה הינו לסדר את המידע באופן זהה שהמחשב בצד המתקבל יוכל לפעול את המידע. והפעם בתמונה רואים את תוכן הודעה ששרת הבן שלח ללקוט.



אנחנו כעת בשכבה התעבורה האחראית לביצוע וידוא שה-data הגיע הנכונה מבין כל האפליקציות במחשב היעד. היא עשויה זאת על ידי שימוש במספר הפורט יעד ופורט המוצא. הפעם ניתן לראות בתמונה הבאה שמספר פורט המוצא תואם למספר הפורט שהגדכנו לשרת הבן - 22222 ומספר הפורט של היעד תואם למספר הפורט שראינו קודם (בפקטה הראשונה) שנitinן ללקוט -



כעת אנחנו בשכבה הרכשת שאחראית לדאג שהחbillה תגעה אל מחשב היעד הסופי.  
אנחנו רואים זאת בתמונה הבאה , שהרי כתובות המקור תואמת לכתובת ה IP של שרת הבן (10.0.2.5)  
וכתוות ה IP של היעד היא היא כתובות ה IP של הלוקוח(10.0.2.4)



אנחנו כעתשוב בשכבת הארץז , אשר אחראית על וידוא הגעת החbillה אל התחנה הקרובה ביותר  
הבאה (בדרך) בדרכה ליעד הסופי.  
אנחנו רואים בתמונה הבאה את הכתובת הפיזית (MAC) של המוצא , שזו הכתובת MAC של  
התחנה האחרונה בו בקרה החbillה בדרכה אל היעד הסופי – השרת.

