

שאלה מספר 1:

UDP Server

```
1 from socket import socket, AF_INET, SOCK_DGRAM
2
3 s = socket(AF_INET, SOCK_DGRAM)
4 source_ip = '0.0.0.0'
5 source_port = 12345
6 s.bind((source_ip, source_port))
7 while True:
8     data, sender_info = s.recvfrom(2048)
9     print "Message: ", data, " from: ", sender_info
10    s.sendto(data.upper(), sender_info)
```

1. מייבא ממודל ה־Socket את פונקציות פתיחת הסוקט מ־socket את סוג כתובות האיפ'י שאנחנו נתעסק בהן, IPv4 מ־AF_INET ו־SOCK_DGRAM נותן לנו את פרוטוקל ה־UDP שבו נשתמש.
3. נפתח סוקט חדש שמקבל כתובות איפ'י מסוג IPv4, שעובד בפרוטוקל UDP.
- 4,5. נגדיר את כתובת האיפ'י להיות 0.0.0.0, כלומר השרת מאזין על כל כתובות האיפ'י של כל כרטיסי הרשת הלוקליים. ובחר פורט רנדומלי שאינו בשימוש להאזין לו (12345).
6. נקשר את כתובת האיפ'י והפורט שבחרנו לסוקט שפתחנו כדי שנאזין דרכו.
7. נריץ לולאה אינסופית בכדי להריץ את השרת ללא הפסקה (אלא אם כן סגרנו בכח את התכנית)
8. נשמור את המידע שמתקבל לסוקט (נאפשר לו לקבל עד 2048 בייטים), ה־data זו הודעה עצמה, בסטרינג, ו־sender_info זו הכתובת שממנה התקבלה ההודעה.
9. נדפיס את ההודעה שהתקבלה ואת כתובת המקור שלה
10. כעת נשלח את אותה ההודעה, לאחר שנמיר אותה לאותיות גדולות, חזרה אל השולח.

UDP Client

```
1 from socket import socket, AF_INET, SOCK_DGRAM
2
3 s = socket(AF_INET, SOCK_DGRAM)
4 dest_ip = '127.0.0.1'
5 dest_port = 12345
6 msg = raw_input("Message to send: ")
7 while not msg == 'quit':
8     s.sendto(msg, (dest_ip, dest_port))
9     data, sender_info = s.recvfrom(2048)
10    print "Server sent: ", data
11    msg = raw_input("Message to send: ")
12 s.close()
```

1. מייבא ממודל Socket את פונקציות פתיחת הסוקט מsocket את סוג כתובות האיפי שאנחנו נתעסק בהן, IPv4 מAF_INET SOCK_DGRAM נותן לנו את פרוטוקל הUDP שבו נשתמש.
3. נפתח סוקט חדש שמקבל כתובות איפי מסוג IPv4, שעובד בפרוטוקל UDP.
- 4,5. כתובת האיפי של שרת היעד שלנו, במקרה זה היא כתובת הloopback כיוון שהשרת נמצא על אותה המכונה. כמו כן כתובת הפורט ששרת היעד מאזינה עליה.
6. נקבל כאינפוט מהמשתמש את ההודעה שהוא רוצה לשלוח.
7. כעת, עד שלא נשלח הודעה של quit, התוכנה תבקש מהמשתמש הודעות, ותשלח אותן לשרת.
8. אנחנו שולחים דרך הסוקט את ההודעה. נעביר לפונקציה השליחה כארגומנטים את ההודעה ואת כתובת האיפי והפורט שהשרת מאזין דרכן.
9. נקבל דרך הסוקט שלנו את התשובה של השרת, נשמור אותו למשתנה data שהוא תוכן ההודעה sender_info שהם פרטי השרת שענה לנו.
- 10,11. נדפיס את ההודעה שקיבלנו (שהיא ההודעה ששלחנו, מומרת לאותיות גדולות) ואז נבקש מהמשתמש את ההודעה הבאה שהוא רוצה לשלוח ונחזור שוב על התהליך.
12. כאשר המשתמש שלח הודעת quit ויצאנו מהלולאה, נסגור את הסוקט ונפנה את הפורט שלנו כיוון שאין בו יותר שימוש, והתכנית נסגרה.

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

שאלה מספר 2:

א.

The screenshot shows the Wireshark interface with a packet capture on interface *enp0s3. The packet list pane displays several packets, with packet 3629 selected. The packet details pane for packet 3629 shows the following structure:

- Frame 3629: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
- Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_7d:1b:f7 (08:00:27:7d:1b:f7)
- Internet Protocol Version 4, Src: 34.250.241.160, Dst: 10.0.2.15
- Transmission Control Protocol, Src Port: 443, Dst Port: 54426, Seq: 3133431295, Ack: 676, Len: 0

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```
0000 08 00 27 7d 1b f7 52 54 00 12 35 02 08 00 45 00  ..'}..RT ..5...E.
0010 00 28 56 23 00 00 ff 06 45 03 22 fa f1 a0 0a 00  .(V#.... E"...
0020 02 0f 01 bb d4 9a 00 00 00 00 fd a8 f9 27 50 14  .....P.
0030 00 00 c2 00 00 00 00 00 00 00 00 00 00 00 00  .....
```

מספר החבילות שהוסנפו בפתיחת שני אתרים הוא 3629.

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

ב.

The image shows a Wireshark packet capture window titled '*enp0s3'. The filter bar at the top shows 'udp'. The packet list pane displays several packets, with packet 3547 selected. The packet details pane shows the structure of the selected packet: Frame 3547: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface 0. The packet is an Ethernet II frame with source PcsCompu_7d:1b:f7 and destination IPv4mcast_fb (01:00:5e:00:00:fb). It is an Internet Protocol Version 4 packet with source 10.0.2.15 and destination 224.0.0.251. The payload is a User Datagram Protocol packet with source port 5353 and destination port 5353. The protocol is Multicast Domain Name System (query). The packet bytes pane shows the raw data in hexadecimal and ASCII. The status bar at the bottom indicates that 3629 packets were captured, 276 were displayed (7.6%), and 0 were dropped (0.0%).

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|--------------|------------------------|-------------|----------|--------|-----------------------|
| 3436 | 12.839886961 | 10.0.2.15 | 10.0.0.138 | DNS | 91 | Standard query 0x... |
| 3437 | 12.853948539 | 10.0.0.138 | 10.0.2.15 | DNS | 154 | Standard query res... |
| 3438 | 12.855002479 | 10.0.0.138 | 10.0.2.15 | DNS | 142 | Standard query res... |
| 3475 | 12.890840818 | 10.0.2.15 | 10.0.0.138 | DNS | 86 | Standard query 0x... |
| 3476 | 12.890975176 | 10.0.2.15 | 10.0.0.138 | DNS | 86 | Standard query 0x... |
| 3478 | 12.906722592 | 10.0.0.138 | 10.0.2.15 | DNS | 133 | Standard query res... |
| 3479 | 12.910702407 | 10.0.0.138 | 10.0.2.15 | DNS | 145 | Standard query res... |
| 3528 | 13.979113090 | fe80::e33e:2b90:bf7... | ff02::fb | MDNS | 107 | Standard query 0x... |
| 3547 | 14.181917429 | 10.0.2.15 | 224.0.0.251 | MDNS | 87 | Standard query 0x... |

Frame 3547: 87 bytes on wire (696 bits), 87 bytes captured (696 bits) on interface 0

Ethernet II, Src: PcsCompu_7d:1b:f7 (08:00:27:7d:1b:f7), Dst: IPv4mcast_fb (01:00:5e:00:00:fb)

Internet Protocol Version 4, Src: 10.0.2.15, Dst: 224.0.0.251

User Datagram Protocol, Src Port: 5353, Dst Port: 5353

Multicast Domain Name System (query)

0000 01 00 5e 00 00 fb 08 00 27 7d 1b f7 08 00 45 00 ..A.... '}....E.
0010 00 49 16 b4 40 00 ff 11 77 e5 0a 00 02 0f e0 00 .I..@... w.....
0020 00 fb 14 e9 14 e9 00 35 ed 50 00 00 00 00 025 .P.....
0030 00 00 00 00 00 00 05 5f 69 70 70 73 04 5f 74 63_ ipps._tc
0040 70 05 6c 6f 63 61 6c 00 00 0c 00 01 04 5f 69 70 p.local._ip
0050 70 c0 12 00 0c 00 01 p.....

User Datagram Protocol: Protocol Packets: 3629 Displayed: 276 (7.6%) Dropped: 0 (0.0%) Profile: Default

כאשר הפעלנו את הפילטר כדי שיראה רק חבילות שנשלחו בפרוטוקל UDP, ראינו שרק 7.6% מסך החבילות נשלחו בפרוטוקל הזה (276 מתוך 3629). כמו כן אפשר לראות שהרוב המוחלט, אם לא כל החבילות, הן Queries לשרתי DNS.

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז. 317301596

ג.

```
dima@dima-VirtualBox: ~  
File Edit View Search Terminal Help  
dima@dima-VirtualBox:~$ ifconfig  
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255  
    inet6 fe80::e33e:2b90:bf7f:8db0 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:7d:1b:f7 txqueuelen 1000 (Ethernet)  
    RX packets 6579 bytes 5955359 (5.9 MB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 3596 bytes 437448 (437.4 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 771 bytes 64961 (64.9 KB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 771 bytes 64961 (64.9 KB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
dima@dima-VirtualBox:~$
```

כתובת האי"פי ברשת היא 10.0.2.15 כאשר מדובר על פרוטוקול IPv4 ו־fe80::e33e:2b90:bf7f:8db0 כאשר מדובר בפרוטוקול IPv6.

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

.ד

The image shows a Wireshark packet capture analysis of a DNS query. The packet list at the top shows a query from 10.0.2.15 to 10.0.0.138 on port 53. The packet details pane shows the Ethernet II, IP, and DNS query structure. The packet bytes pane shows the raw data in hexadecimal and ASCII.

| No. | Time | Source | Destination | Protocol | Length | Source Port | Dest. Port | Info |
|------|-------------|------------|-------------|----------|--------|-------------|------------|----------------|
| 1446 | 7.835828783 | 10.0.2.15 | 10.0.0.138 | DNS | 91 | 39067 | 53 | Standard query |
| 1468 | 8.380429783 | 10.0.2.15 | 10.0.0.138 | DNS | 83 | 34554 | 53 | Standard query |
| 1469 | 8.380552406 | 10.0.2.15 | 10.0.0.138 | DNS | 83 | 57153 | 53 | Standard query |
| 1470 | 8.395705933 | 10.0.0.138 | 10.0.2.15 | DNS | 233 | 53 | 57153 | Standard query |
| 1471 | 8.395720786 | 10.0.0.138 | 10.0.2.15 | DNS | 174 | 53 | 34554 | Standard query |
| 1472 | 8.464445608 | 10.0.2.15 | 10.0.0.138 | DNS | 87 | 52586 | 53 | Standard query |
| 1473 | 8.464559968 | 10.0.2.15 | 10.0.0.138 | DNS | 87 | 56447 | 53 | Standard query |
| 1474 | 8.478820083 | 10.0.0.138 | 10.0.2.15 | DNS | 193 | 53 | 52586 | Standard query |
| 1475 | 8.479860224 | 10.0.0.138 | 10.0.2.15 | DNS | 213 | 53 | 56447 | Standard query |
| 1476 | 8.480023227 | 10.0.2.15 | 10.0.0.138 | DNS | 99 | 56866 | 53 | Standard query |
| 1477 | 8.494737585 | 10.0.0.138 | 10.0.2.15 | DNS | 183 | 53 | 56866 | Standard query |
| 1478 | 8.619004641 | 10.0.2.15 | 10.0.0.138 | DNS | 85 | 59145 | 53 | Standard query |
| 1479 | 8.619149206 | 10.0.2.15 | 10.0.0.138 | DNS | 85 | 46161 | 53 | Standard query |
| 1480 | 8.632971163 | 10.0.0.138 | 10.0.2.15 | DNS | 141 | 53 | 59145 | Standard query |
| 1481 | 8.633868273 | 10.0.0.138 | 10.0.2.15 | DNS | 196 | 53 | 46161 | Standard query |

Frame 1469: 83 bytes on wire (664 bits), 83 bytes captured (664 bits) on interface 0

Ethernet II, Src: PcsCompu_7d:1b:f7 (08:00:27:7d:1b:f7), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)

Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.0.138

User Datagram Protocol, Src Port: 57153, Dst Port: 53

Domain Name System (query)

0000 52 54 00 12 35 02 08 00 27 7d 1b f7 08 00 45 00 RT..5... '}'...E.
 0010 00 45 18 f6 40 00 40 11 0b 1a 0a 00 02 0f 0a 00 .E..@.@.....
 0020 00 8a df 41 00 35 00 31 16 db c5 f1 01 00 00 01 ...A.5.1.....
 0030 00 00 00 00 00 01 01 7a 04 79 6e 65 74 02 63 6fz..ynet.co

Domain Name System: Protocol Packets: 3629 - Displayed: 270 (7.4%) - Dropped: 0 (0.0%) Profile: Default

- החבילה שבחרתי נשלחה מהמחשב שלי אל השרת. ניתן לקבוע זאת כיוון שכתובת האיפ'י שמוגדרת כמקור החבילה היא כתובת האיפ'י שהראנו שמשייכת אליי בסעיף הקודם (10.0.2.15).
- ניתן לראות שהחבילה נשלחה מפורט 57153 אל פורט 53. הלקוח האזין לפורט 57153, שהוא פורט המקור והשרת האזין לפורט 53. (הסופתי עמודות פורט מקור/יעד לשם נוחות)
- כתובת האיפ'י של השולח היא 10.0.2.15 וכתובת האיפ'י של היעד היא 10.0.0.138
- כתובת הMAC של השולח היא 08:00:27:7d:1b:f7 והכתובת של היעד היא 52:54:00:12:35:02

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז. 317301596

ה.

The image shows a Wireshark packet capture window titled 'packets.pcapng'. The main pane displays a list of network packets. Packet 1470 is highlighted in blue and has a red rectangle around it. The packet details pane on the right shows the structure of packet 1470: Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and Domain Name System (response). The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

| No. | Time | Source | Destination | Protocol | Length | Source Port | Dest. Port | Info |
|------|-------------|----------------|----------------|----------|--------|-------------|------------|------------------|
| 1465 | 7.991009133 | 182.50.136.239 | 10.0.2.15 | TCP | 60 | 80 | 39456 | 80 → 39456 [FIN] |
| 1466 | 8.032623051 | 10.0.2.15 | 182.50.136.239 | TCP | 54 | 39458 | 80 | 39458 → 80 [ACK] |
| 1467 | 8.035650603 | 10.0.2.15 | 182.50.136.239 | TCP | 54 | 39456 | 80 | 39456 → 80 [ACK] |
| 1468 | 8.380429783 | 10.0.2.15 | 10.0.0.138 | DNS | 83 | 34554 | 53 | Standard query |
| 1469 | 8.380552406 | 10.0.2.15 | 10.0.0.138 | DNS | 83 | 57153 | 53 | Standard query |
| 1470 | 8.395705933 | 10.0.0.138 | 10.0.2.15 | DNS | 233 | 53 | 57153 | Standard query |
| 1471 | 8.395720786 | 10.0.0.138 | 10.0.2.15 | DNS | 174 | 53 | 34554 | Standard query |
| 1472 | 8.464445608 | 10.0.2.15 | 10.0.0.138 | DNS | 87 | 52586 | 53 | Standard query |
| 1473 | 8.464559968 | 10.0.2.15 | 10.0.0.138 | DNS | 87 | 56447 | 53 | Standard query |
| 1474 | 8.478820083 | 10.0.0.138 | 10.0.2.15 | DNS | 193 | 53 | 52586 | Standard query |
| 1475 | 8.479860224 | 10.0.0.138 | 10.0.2.15 | DNS | 213 | 53 | 56447 | Standard query |
| 1476 | 8.480023227 | 10.0.2.15 | 10.0.0.138 | DNS | 99 | 56866 | 53 | Standard query |
| 1477 | 8.494737585 | 10.0.0.138 | 10.0.2.15 | DNS | 183 | 53 | 56866 | Standard query |
| 1478 | 8.619004641 | 10.0.2.15 | 10.0.0.138 | DNS | 85 | 59145 | 53 | Standard query |
| 1479 | 8.619149206 | 10.0.2.15 | 10.0.0.138 | DNS | 85 | 46161 | 53 | Standard query |

Frame 1470: 233 bytes on wire (1864 bits), 233 bytes captured (1864 bits) on interface 0
Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_7d:1b:f7 (08:00:27:7d:1b:f7)
Internet Protocol Version 4, Src: 10.0.0.138, Dst: 10.0.2.15
User Datagram Protocol, Src Port: 53, Dst Port: 57153
Domain Name System (response)

0000 08 00 27 7d 1b f7 52 54 00 12 35 02 08 00 45 00 ..}.RT..5...E.
0010 00 db 4a db 00 00 40 11 18 9f 0a 00 00 8a 0a 00 ...J...@.....
0020 02 0f 00 35 df 41 00 c7 a8 f2 c5 f1 81 80 00 01 ...5.A.....
0030 00 02 00 01 00 01 01 7a 04 79 6e 65 74 02 63 6fz..ynet.co

– החבילה המתאימה לחבילה הקודמת היא זו שמסומנת, ניתן לראות שהחצים הם בין החבילה הקודמת לזו, וכן מקור ויעד האיפי והפורט הפוכים מהקודמת.

– הפורט שהחבילה נשלחה ממנו הוא פורט 53 והפורט שאליו החבילה נשלחה הוא פורט 57153.

– כתובת האיפי של המקור היא 10.0.0.138 וכתובת האיפי של היעד היא 10.0.2.15

– כתובת ה-MAC של המקור היא 52:54:00:12:35:02 וכתובת היעד היא 08:00:27:7d:1b:f7

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז. 317301596

packets.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.src == 10.0.2.15 and dns

| No. | Time | Source | Destination | Protocol | Length | Source Port | Dest. Port | Info |
|-----|-------------|-----------|-------------|----------|--------|-------------|------------|----------------|
| 3 | 2.392947564 | 10.0.2.15 | 10.0.0.138 | DNS | 81 | 46893 | 53 | Standard query |
| 4 | 2.393086706 | 10.0.2.15 | 10.0.0.138 | DNS | 81 | 44403 | 53 | Standard query |
| 7 | 2.410836064 | 10.0.2.15 | 10.0.0.138 | DNS | 86 | 54749 | 53 | Standard query |
| 8 | 2.411149758 | 10.0.2.15 | 10.0.0.138 | DNS | 86 | 36284 | 53 | Standard query |
| 10 | 2.450138710 | 10.0.2.15 | 10.0.0.138 | DNS | 97 | 57012 | 53 | Standard query |
| 11 | 2.450314139 | 10.0.2.15 | 10.0.0.138 | DNS | 97 | 52453 | 53 | Standard query |
| 13 | 2.464527828 | 10.0.2.15 | 10.0.0.138 | DNS | 86 | 47667 | 53 | Standard query |
| 14 | 2.464626589 | 10.0.2.15 | 10.0.0.138 | DNS | 86 | 54368 | 53 | Standard query |
| 16 | 2.471644486 | 10.0.2.15 | 10.0.0.138 | DNS | 87 | 49838 | 53 | Standard query |
| 17 | 2.471843415 | 10.0.2.15 | 10.0.0.138 | DNS | 87 | 49333 | 53 | Standard query |
| 20 | 2.482144222 | 10.0.2.15 | 10.0.0.138 | DNS | 88 | 34214 | 53 | Standard query |
| 21 | 2.482317616 | 10.0.2.15 | 10.0.0.138 | DNS | 88 | 36008 | 53 | Standard query |
| 24 | 2.492292855 | 10.0.2.15 | 10.0.0.138 | DNS | 85 | 43345 | 53 | Standard query |
| 25 | 2.492430210 | 10.0.2.15 | 10.0.0.138 | DNS | 85 | 39134 | 53 | Standard query |
| 28 | 2.517161948 | 10.0.2.15 | 10.0.0.138 | DNS | 85 | 56755 | 53 | Standard query |
| 29 | 2.517284958 | 10.0.2.15 | 10.0.0.138 | DNS | 85 | 41268 | 53 | Standard query |
| 32 | 2.518607621 | 10.0.2.15 | 10.0.0.138 | DNS | 88 | 36150 | 53 | Standard query |
| 33 | 2.518723361 | 10.0.2.15 | 10.0.0.138 | DNS | 88 | 46173 | 53 | Standard query |
| 36 | 2.570979199 | 10.0.2.15 | 10.0.0.138 | DNS | 92 | 54251 | 53 | Standard query |
| 38 | 2.585762678 | 10.0.2.15 | 10.0.0.138 | DNS | 93 | 35966 | 53 | Standard query |
| 39 | 2.585867545 | 10.0.2.15 | 10.0.0.138 | DNS | 93 | 34636 | 53 | Standard query |
| 52 | 2.782089663 | 10.0.2.15 | 10.0.0.138 | DNS | 88 | 46217 | 53 | Standard query |
| 53 | 2.782210668 | 10.0.2.15 | 10.0.0.138 | DNS | 88 | 35301 | 53 | Standard query |
| 69 | 3.269302705 | 10.0.2.15 | 10.0.0.138 | DNS | 85 | 51507 | 53 | Standard query |
| 208 | 4.797531172 | 10.0.2.15 | 10.0.0.138 | DNS | 81 | 38728 | 53 | Standard query |
| 209 | 4.797687855 | 10.0.2.15 | 10.0.0.138 | DNS | 81 | 58138 | 53 | Standard query |

Frame 3: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0
 Ethernet II, Src: PcsCompu_7d:1b:f7 (08:00:27:7d:1b:f7), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
 Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.0.138
 User Datagram Protocol, Src Port: 46893, Dst Port: 53
 Domain Name System (query)

```

0000  52 54 00 12 35 02 08 00 27 7d 1b f7 08 00 45 00  RT..5...'}....E.
0010  00 43 16 58 40 00 40 11 0d ba 0a 00 02 0f 0a 00  .C.X@.@.....
0020  00 8a b7 2d 00 35 00 2f 16 d9 63 6a 01 00 00 01  ....5./..cj....
0030  00 00 00 00 00 01 06 67 69 74 68 75 62 03 63 6f  ....g ithub.co
0040  6d 00 00 01 00 01 00 00 29 02 00 00 00 00 00 00  m.....).....
0050  00
  
```

Domain Name System: Protocol Packets: 3629 · Displayed: 139 (3.8%) Profile: Default

כעת סיננו לפי החבילות שנשלחו מהמחשב שלי (אייפי המקור הוא 10.0.2.15) ושרן בפרוטוקול DNS. ניתן לראות שבכל query שנשלח מהמחשב שלי אל היעד, היעד של שרת DNS תמיד מאזין על פורט 53, ואילו כל שאילתא שאני שולח, נשלחת מסוקט עם פורט רנדומלי. המשמעות כאמור היא שתעבורת ה-DNS query עוברת תמיד דרך פורט 53 כשהיא מגיעה לשרת או יוצאת ממנו.

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

1.

The image shows a Wireshark packet capture window titled 'packets.pcapng'. A filter is applied: `ip.dst == 10.0.2.15 and dns`. The packet list shows 36 DNS queries from source 10.0.0.138 to destination 10.0.2.15, all using source port 53 and destination port 44403. The selected packet (No. 5) is expanded to show details: Ethernet II, Internet Protocol Version 4, User Datagram Protocol, and Domain Name System (response). The packet bytes pane shows the raw data for the DNS response, including the query ID, flags, and the response data.

| No. | Time | Source | Destination | Protocol | Length | Source Port | Dest. Port | Info |
|-----|-------------|------------|-------------|----------|--------|-------------|------------|----------------|
| 5 | 2.408871775 | 10.0.0.138 | 10.0.2.15 | DNS | 168 | 53 | 44403 | Standard query |
| 6 | 2.410465892 | 10.0.0.138 | 10.0.2.15 | DNS | 113 | 53 | 46893 | Standard query |
| 12 | 2.464015520 | 10.0.0.138 | 10.0.2.15 | DNS | 271 | 53 | 57012 | Standard query |
| 15 | 2.466235809 | 10.0.0.138 | 10.0.2.15 | DNS | 228 | 53 | 52453 | Standard query |
| 18 | 2.480251271 | 10.0.0.138 | 10.0.2.15 | DNS | 363 | 53 | 47667 | Standard query |
| 19 | 2.481518026 | 10.0.0.138 | 10.0.2.15 | DNS | 151 | 53 | 54368 | Standard query |
| 22 | 2.485581059 | 10.0.0.138 | 10.0.2.15 | DNS | 144 | 53 | 49838 | Standard query |
| 23 | 2.491120734 | 10.0.0.138 | 10.0.2.15 | DNS | 156 | 53 | 49333 | Standard query |
| 26 | 2.515334155 | 10.0.0.138 | 10.0.2.15 | DNS | 118 | 53 | 54749 | Standard query |
| 27 | 2.516798784 | 10.0.0.138 | 10.0.2.15 | DNS | 173 | 53 | 36284 | Standard query |
| 30 | 2.518151615 | 10.0.0.138 | 10.0.2.15 | DNS | 116 | 53 | 36008 | Standard query |
| 31 | 2.518161709 | 10.0.0.138 | 10.0.2.15 | DNS | 104 | 53 | 34214 | Standard query |
| 34 | 2.569062421 | 10.0.0.138 | 10.0.2.15 | DNS | 136 | 53 | 43345 | Standard query |
| 35 | 2.570642345 | 10.0.0.138 | 10.0.2.15 | DNS | 191 | 53 | 39134 | Standard query |
| 37 | 2.585302024 | 10.0.0.138 | 10.0.2.15 | DNS | 163 | 53 | 54251 | Standard query |
| 40 | 2.599262752 | 10.0.0.138 | 10.0.2.15 | DNS | 170 | 53 | 34636 | Standard query |
| 51 | 2.781520826 | 10.0.0.138 | 10.0.2.15 | DNS | 109 | 53 | 35966 | Standard query |
| 66 | 3.263974752 | 10.0.0.138 | 10.0.2.15 | DNS | 194 | 53 | 35301 | Standard query |
| 67 | 3.264623794 | 10.0.0.138 | 10.0.2.15 | DNS | 170 | 53 | 46217 | Standard query |
| 70 | 3.285338688 | 10.0.0.138 | 10.0.2.15 | DNS | 113 | 53 | 51507 | Standard query |
| 210 | 4.891240072 | 10.0.0.138 | 10.0.2.15 | DNS | 97 | 53 | 38728 | Standard query |
| 211 | 4.896149703 | 10.0.0.138 | 10.0.2.15 | DNS | 160 | 53 | 58138 | Standard query |
| 227 | 5.077496638 | 10.0.0.138 | 10.0.2.15 | DNS | 178 | 53 | 51379 | Standard query |
| 228 | 5.079157017 | 10.0.0.138 | 10.0.2.15 | DNS | 237 | 53 | 39381 | Standard query |
| 230 | 5.094287248 | 10.0.0.138 | 10.0.2.15 | DNS | 169 | 53 | 49922 | Standard query |
| 326 | 5.314690026 | 10.0.0.138 | 10.0.2.15 | DNS | 144 | 53 | 55491 | Standard query |

Frame 5: 168 bytes on wire (1344 bits), 168 bytes captured (1344 bits) on interface 0
 Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_7d:1b:f7 (08:00:27:7d:1b:f7)
 Internet Protocol Version 4, Src: 10.0.0.138, Dst: 10.0.2.15
 User Datagram Protocol, Src Port: 53, Dst Port: 44403
 Domain Name System (response)

0000 08 00 27 7d 1b f7 52 54 00 12 35 02 08 00 45 00 ...}..RT..5...E.
 0010 00 9a 45 b2 00 00 40 11 1e 09 0a 00 00 8a 0a 00 ...E...@.....
 0020 02 0f 00 35 ad 73 00 86 f2 23 22 a1 81 80 00 01 ...5.s...#".....
 0030 00 00 00 01 00 01 06 67 69 74 68 75 62 03 63 6fg ithub.co
 0040 6d 00 00 1c 00 01 c0 0c 00 06 00 01 00 00 00 de m.....
 0050 00 4b 07 6e 73 2d 31 37 30 37 09 61 77 73 64 6e ..K.ns-17 07·awsdn

אם נסתכל בתעבורה כאשר כתובת האייפי שלנו היא היעד, ניתן לראות אשורור לסענה הקודמת. כל התעבורה של שאליתאות DNS יוצאת מפורט 53, ומגיעה אלינו כאשר אנחנו מאזינים דרך פורט רנדומלי.

מוטב גם לציין כי כל חבילה בשאליתא עוברת בפורטוקול הUDP. לפי בדיקה באינטרנט מסתבר כי DNS Queries משתמשים בUDP כל עוד גודל החבילה לא עולה על 512 בייטים (וכפי שניתן לראות מהצילום, אף חבילה לא עברה את הגודל הזה – לפי עמודת הLength). אם והייתה שאליתא שהתשובה אליה גדולה מהגודל הזה, אז ככל הנראה היא הייתה עוברת בפורטוקול TCP, אלא אם כן בוצע איזשהו UDP extension המאפשר שליחה של חבילות גדולות יותר.

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

.T

packets.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

udp.port == 53 Expression...

| No. | Time | Source | Destination | Protocol | Length | Source Port | Dest. Port | Info |
|------|--------------|------------|-------------|----------|--------|-------------|------------|----------------|
| 3115 | 11.519711308 | 10.0.2.15 | 10.0.0.138 | DNS | 89 | 34027 | 53 | Standard query |
| 3116 | 11.519817788 | 10.0.2.15 | 10.0.0.138 | DNS | 89 | 57129 | 53 | Standard query |
| 3124 | 11.534683776 | 10.0.0.138 | 10.0.2.15 | DNS | 143 | 53 | 42897 | Standard query |
| 3125 | 11.535859772 | 10.0.0.138 | 10.0.2.15 | DNS | 198 | 53 | 34442 | Standard query |
| 3128 | 11.538472015 | 10.0.0.138 | 10.0.2.15 | DNS | 201 | 53 | 57129 | Standard query |
| 3129 | 11.538484764 | 10.0.0.138 | 10.0.2.15 | DNS | 146 | 53 | 34027 | Standard query |
| 3290 | 12.520703052 | 10.0.2.15 | 10.0.0.138 | DNS | 90 | 36385 | 53 | Standard query |
| 3291 | 12.520800260 | 10.0.2.15 | 10.0.0.138 | DNS | 90 | 59364 | 53 | Standard query |
| 3302 | 12.538172393 | 10.0.0.138 | 10.0.2.15 | DNS | 169 | 53 | 36385 | Standard query |
| 3305 | 12.540115063 | 10.0.0.138 | 10.0.2.15 | DNS | 224 | 53 | 59364 | Standard query |
| 3327 | 12.597676148 | 10.0.2.15 | 10.0.0.138 | DNS | 86 | 60928 | 53 | Standard query |
| 3328 | 12.597804027 | 10.0.2.15 | 10.0.0.138 | DNS | 86 | 41356 | 53 | Standard query |
| 3339 | 12.612749453 | 10.0.0.138 | 10.0.2.15 | DNS | 102 | 53 | 60928 | Standard query |
| 3340 | 12.614442945 | 10.0.0.138 | 10.0.2.15 | DNS | 114 | 53 | 41356 | Standard query |
| 3431 | 12.824284332 | 10.0.2.15 | 10.0.0.138 | DNS | 95 | 38030 | 53 | Standard query |
| 3432 | 12.824450260 | 10.0.2.15 | 10.0.0.138 | DNS | 95 | 37776 | 53 | Standard query |
| 3433 | 12.838483654 | 10.0.0.138 | 10.0.2.15 | DNS | 170 | 53 | 38030 | Standard query |
| 3434 | 12.838499134 | 10.0.0.138 | 10.0.2.15 | DNS | 158 | 53 | 37776 | Standard query |
| 3435 | 12.839743134 | 10.0.2.15 | 10.0.0.138 | DNS | 91 | 39067 | 53 | Standard query |
| 3436 | 12.839886961 | 10.0.2.15 | 10.0.0.138 | DNS | 91 | 51587 | 53 | Standard query |
| 3437 | 12.853948539 | 10.0.0.138 | 10.0.2.15 | DNS | 154 | 53 | 39067 | Standard query |
| 3438 | 12.855002479 | 10.0.0.138 | 10.0.2.15 | DNS | 142 | 53 | 51587 | Standard query |
| 3475 | 12.890840818 | 10.0.2.15 | 10.0.0.138 | DNS | 86 | 58041 | 53 | Standard query |
| 3476 | 12.890975176 | 10.0.2.15 | 10.0.0.138 | DNS | 86 | 50949 | 53 | Standard query |
| 3478 | 12.906722592 | 10.0.0.138 | 10.0.2.15 | DNS | 133 | 53 | 58041 | Standard query |
| 3479 | 12.910702407 | 10.0.0.138 | 10.0.2.15 | DNS | 145 | 53 | 50949 | Standard query |

Frame 5: 168 bytes on wire (1344 bits), 168 bytes captured (1344 bits) on interface 0
 Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_7d:1b:f7 (08:00:27:7d:1b:f7)
 Internet Protocol Version 4, Src: 10.0.0.138, Dst: 10.0.2.15
 User Datagram Protocol, Src Port: 53, Dst Port: 44403
 Domain Name System (response)

```

0000  08 00 27 7d 1b f7 52 54 00 12 35 02 08 00 45 00  ..}.RT..5...E.
0010  00 9a 45 b2 00 00 40 11 1e 09 0a 00 00 8a 0a 00  ..E...@. ....
0020  02 0f 00 35 ad 73 00 86 f2 23 22 a1 81 80 00 01  ...5.s..#"....
0030  00 00 00 01 00 01 06 67 69 74 68 75 62 03 63 6f  ....g ithub.co
0040  6d 00 00 1c 00 01 c0 0c 00 06 00 01 00 00 00 de  m.....
0050  00 4b 07 6e 73 2d 31 37 30 37 09 61 77 73 64 6e  -K.ns-17 07-awsdn
  
```

packets.pcapng Packets: 3629 · Displayed: 270 (7.4%) Profile: Default

משני הסעיפים הקודמים ניתן להסיק שנוכל לסנן את כל חבילות שנשלחו בפורטוקול DNS אם נסנן לפי כל החבילות שנשלחו ל או מפורט 53 לפי UDP. ואכן, אם נשווה את הצילום של הסינון למעלה אל הצילום בתת סעיף ד' נוכל לראות כי מספר החבילות המוצגות זהה.

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

שאלה 3:

הרצתי את Clientn על מכונה וירטואלית שמריצה Ubuntu על הדסקטופ, ואת צד השרת על לפטופ שמריץ גם הוא Ubuntu.

זו התעבורה שנתפסה בWireshark בצד הלקוח לאחר שליחת ארבעה DNS Queries: (שתי פקטות לא רלוונטיות נתפסו לכן סיננתי לפי UDP)

Wireshark capture of UDP traffic on interface enp0s3. The packet list shows 10 packets, with the first 8 being DNS queries. The packet details pane shows the structure of a User Datagram Protocol (UDP) packet, including Ethernet II, Internet Protocol Version 4, and User Datagram Protocol fields. The packet bytes pane shows the raw data in hexadecimal and ASCII.

| No. | Time | Source | Destination | Protocol | Length | Source Port | Dest. Port | Info |
|-----|--------------|-----------|-------------|----------|--------|-------------|------------|----------------------|
| 1 | 0.000000000 | 10.0.2.15 | 10.0.0.10 | UDP | 50 | 34964 | 12345 | 34964 → 12345 Len=8 |
| 2 | 0.101827444 | 10.0.0.10 | 10.0.2.15 | UDP | 60 | 12345 | 34964 | 12345 → 34964 Len=10 |
| 3 | 3.868663655 | 10.0.2.15 | 10.0.0.10 | UDP | 59 | 34964 | 12345 | 34964 → 12345 Len=17 |
| 4 | 3.892123418 | 10.0.0.10 | 10.0.2.15 | UDP | 60 | 12345 | 34964 | 12345 → 34964 Len=7 |
| 7 | 7.667943631 | 10.0.2.15 | 10.0.0.10 | UDP | 51 | 34964 | 12345 | 12345 → 34964 Len=9 |
| 8 | 7.685012945 | 10.0.0.10 | 10.0.2.15 | UDP | 60 | 12345 | 34964 | 34964 → 12345 Len=7 |
| 9 | 11.266358641 | 10.0.2.15 | 10.0.0.10 | UDP | 54 | 34964 | 12345 | 34964 → 12345 Len=12 |
| 10 | 11.366441282 | 10.0.0.10 | 10.0.2.15 | UDP | 60 | 12345 | 34964 | 12345 → 34964 Len=9 |

Frame 1: 50 bytes on wire (400 bits), 50 bytes captured (400 bits) on interface 0
Ethernet II, Src: PcsCompu_7d:1b:f7 (08:00:27:7d:1b:f7), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.0.10
User Datagram Protocol, Src Port: 34964, Dst Port: 12345
Data (8 bytes)

0000 52 54 00 12 35 02 08 00 27 7d 1b f7 08 00 45 00 RT..5... '}....E.
0010 00 24 bd 10 40 00 40 11 67 a0 0a 00 02 0f 0a 00 .\$.@.@.g.....
0020 00 0a 88 94 30 39 00 10 16 3a 6d 6f 72 65 2e 69 ...09...:more.i
0030 70 73 ps

– ניתן לראות שלאחר 4 queries קיבלנו 8 חבילות, כלומר לכל שאליתא יש חבילה ששולחת את הבקשה, וחבילה שמקבלים שמחזירה את התשובה.

– פורט המקור הוא 34964, שזהו פורט רנדומלי שקושר לסוקט של הקליינט. הוא לא השתנה כיוון שלא סגרנו את הסוקט עד ליציאת התוכנה. במידה והיינו פותחים את הקליינט מחדש ומבצעים עוד queries, הן הינו נשלחות דרך פורט רנדומלי אחר. פורט היעד, 12345, הוא הפורט שנקבע ונשלח כארגומנט לצד השרת בעת פתיחת השרת.

– אייפי המקור הוא האיפי של המכונה הוירטואלית ברשת (10.0.2.15) ואיפי היעד (10.0.0.10) הוא האיפי של המחשב השני ברשת שהעברנו כארגומנט לקליינט אחרי שבדקנו מה האיפי עצמו ע"י הרצת ifconfig במחשב של השרת.

מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז. 317301596

הפלט של צד הלקוח הוא:

```
dima@dima-VirtualBox: ~/csprojects/Networks/ex1
File Edit View Search Terminal Help
dima@dima-VirtualBox:~/csprojects/Networks/ex1$ python2.7 UDPClient.py 10.0.0.10 12345
Find IP of: more.ips
IP Address: 11.111.1.1
Find IP of: mail.google.co.il
IP Address: 9.9.9.9
Find IP of: biu.ac.il
IP Address: 1.2.3.4
Find IP of: google.co.il
IP Address: 13.37.0.1
Find IP of: quit
dima@dima-VirtualBox:~/csprojects/Networks/ex1$
```

נתבונן יותר לעומק באחד מהQueries ובתשובה של השרת אליו:
נקח את החבילה השניה שנשלחה מצד הלקוח

The image shows a Wireshark packet capture analysis. The main window displays a list of packets, with packet 3 selected. The packet details pane shows the following information:

- Frame 3: 59 bytes on wire (472 bits), 59 bytes captured (472 bits) on interface 0
- Ethernet II, Src: PcsCompu_7d:1b:f7 (08:00:27:7d:1b:f7), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
- Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.0.10
- User Datagram Protocol, Src Port: 34964, Dst Port: 12345
- Data (17 bytes)

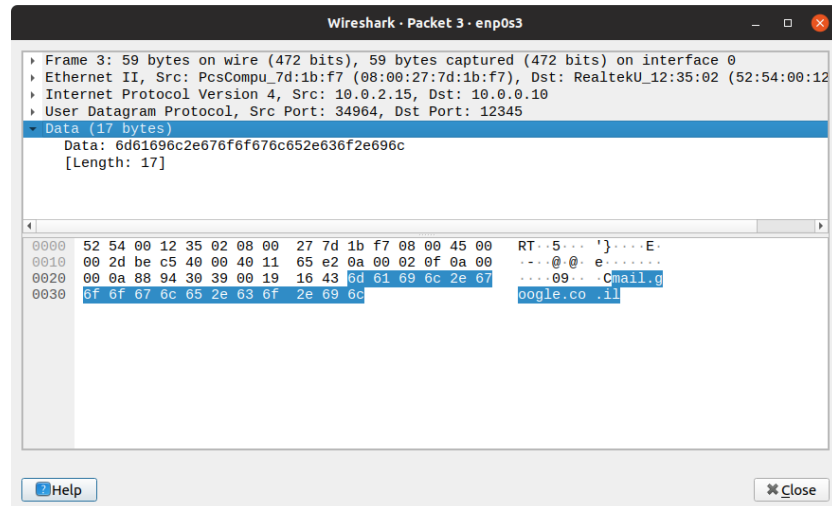
The packet bytes pane shows the following data:

```
0000 52 54 00 12 35 02 08 00 27 7d 1b f7 08 00 45 00  RT...'}...E.
0010 00 2d be c5 40 00 40 11 65 e2 0a 00 02 0f 0a 00  ....@.@.e.....
0020 00 0a 88 94 30 39 00 19 16 43 6d 61 69 6c 2e 67  ....09...Cmail.g
0030 6f 6f 67 6c 65 2e 63 6f 2e 69 6c                oogle.co.il
```

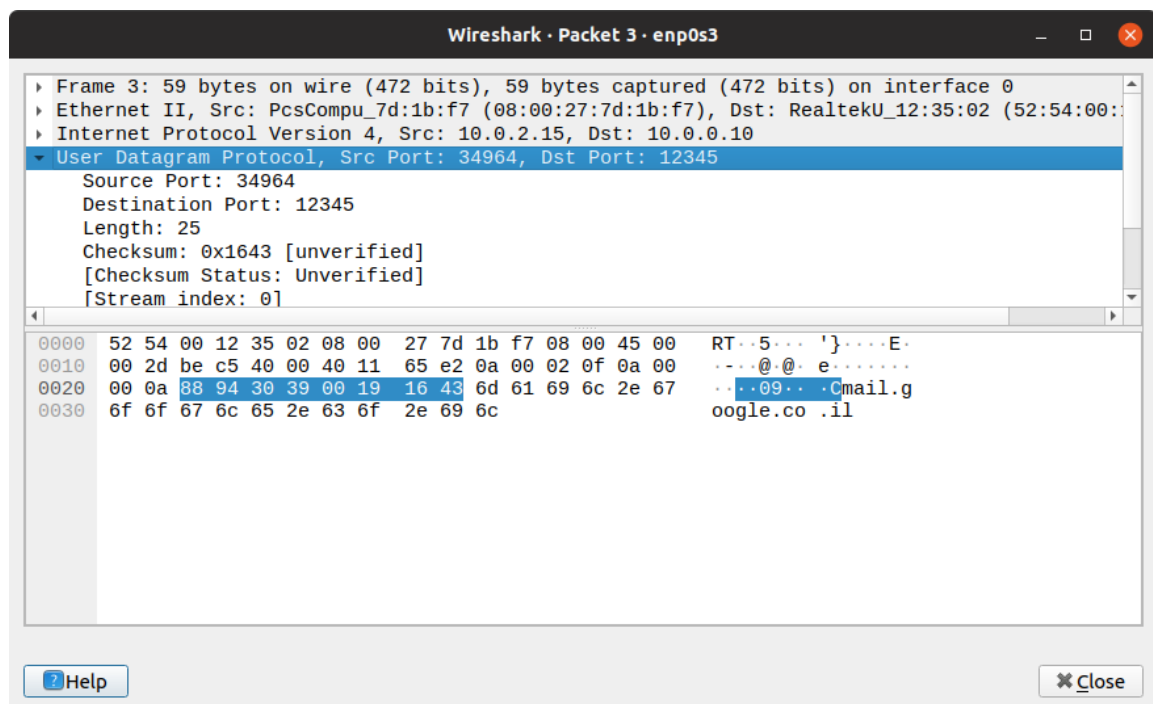
מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז. 317301596

ניתן לראות שהמידע שהחבילה מעבירה, במקרה שלנו הכתובת של האתר שאנחנו מעוניינים לקבל את האייפי שלה, היא החלק האחרון בפקטה שמורכבת מheaders שמורכבים עליה בכל שכבות התעבורה.



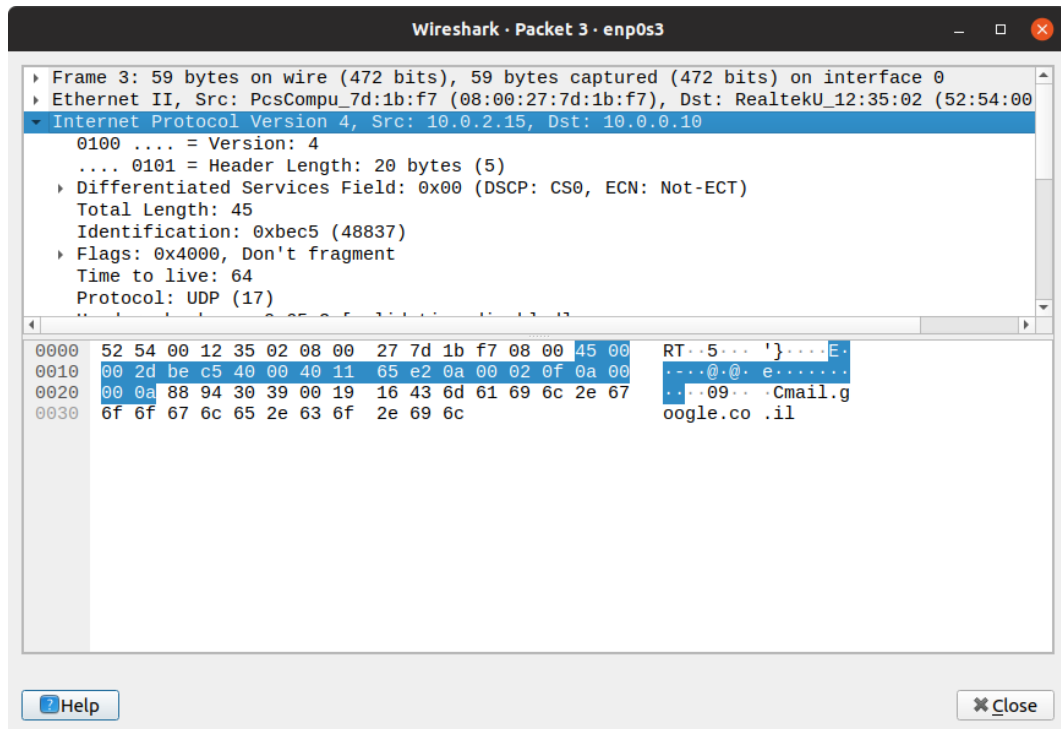
החלק של המידע מועבר בApplication Layer פשוט למחרוזת של בתים שתועבר. לאחר מכן אנחנו יורדים לTransport Layer, בה מתווסף למידע שאנחנו שולחים האדר בו מוגדר שהוא נשלח בפרוטוקול UDP ומתווסף לו להאדר פורט המקור והיעד. כמו כן מתווסף צקסאם של ההאדר וגודל החבילה בבייטים, כעת פיסת המידע נקראת Segment.



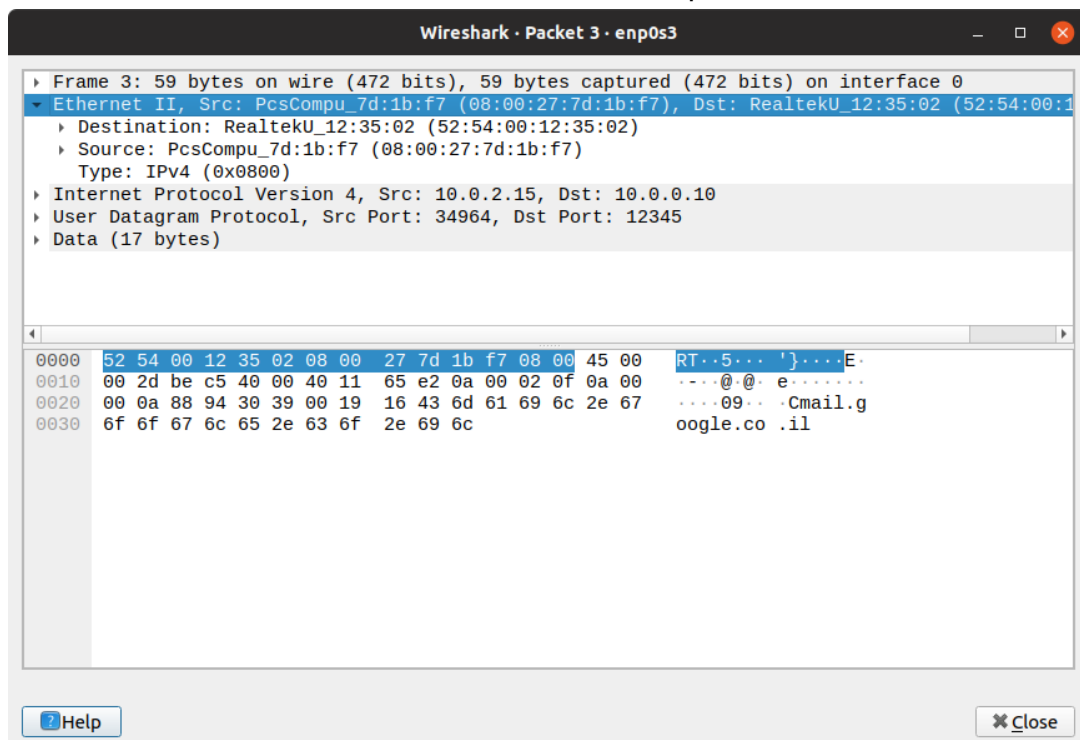
מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז. 317301596

אחרי שכבת התעבורה אנחנו עוברים ל-Network Layer. כעת מתווסף למידע שלנו האדר שמגדיר מאיזה IP ולאיזה IP המידע נשלח וכן האם מדובר בIPv4 או בIPv6, ה-Segment כעת נקרא Packet. ניתן לראות גם שבתוך השכבה מתווסף ה-TTL של החבילה.



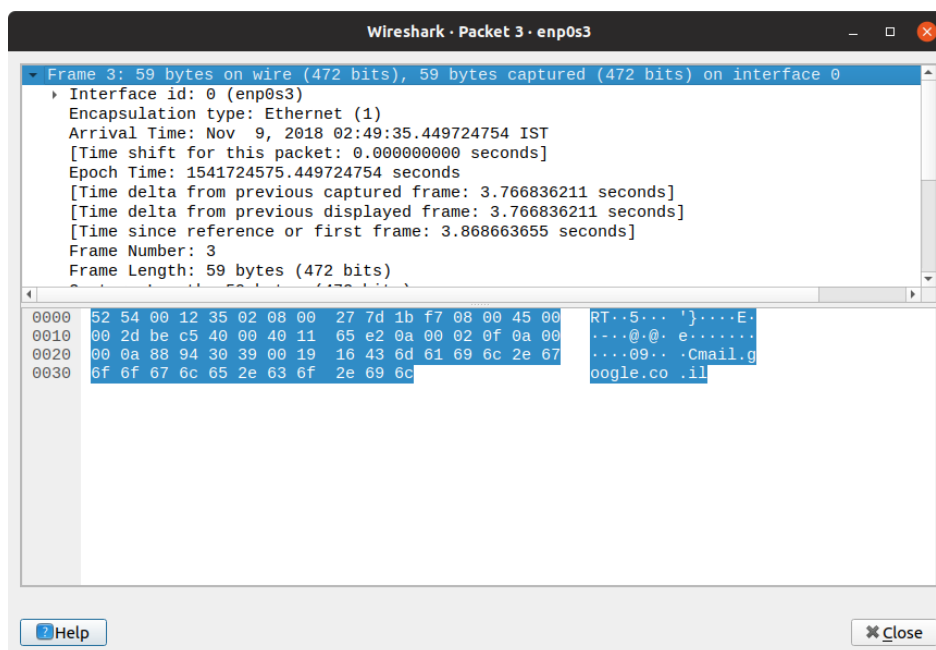
אחרי שכבת הרשת אנחנו עוברים ל-Link Layer. בשכבה זו מתבצע מיפוי לכתובות MAC. ניתן לראות את כתובת ה-MAC של המקור ושל היעד:



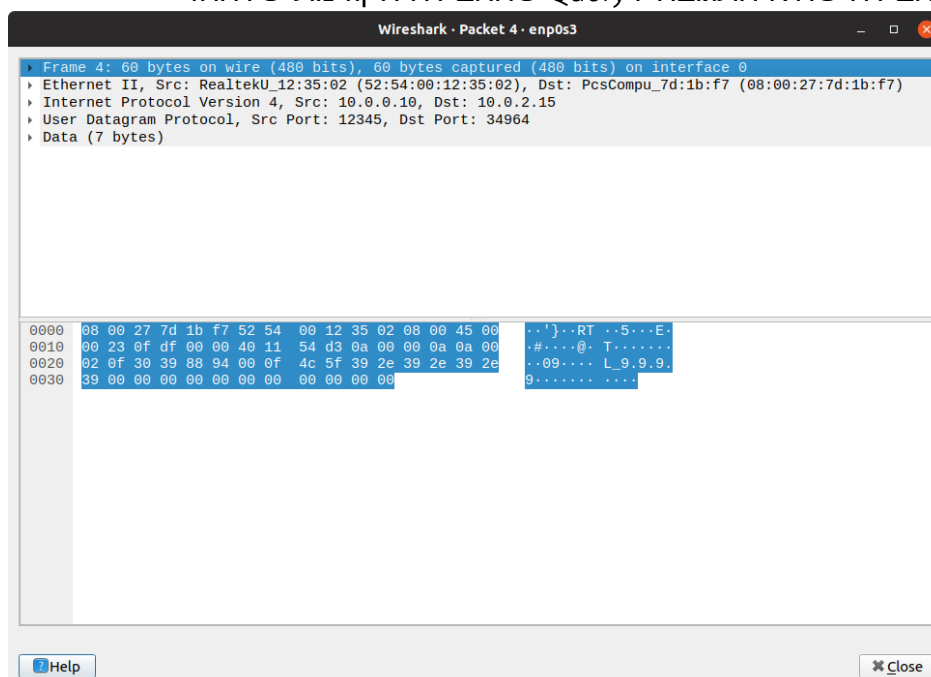
מבוא לרשתות תקשורת – תרגיל 1

דמיטרי זינקביץ ת.ז 317301596

לאחר שכבת הלינק נוצר לנו הFrame שמורכב מכל השכבות והוא זה שמועבר בסופו של דבר אל היעד שלנו.



נתבונן בחבילה שהיא התגובה לQuery שהחבילה הקודמת שלחה:



ניתן לראות שעיקר הדברים די דומה, החבילה מורכבת מהמידע (7 בייטים של המחרוזת של האיפ'י 9.9.9.9), היא נשלחת לפורט של הקליינט (34964) מהפורט של השרת (12345), כנ"ל לגבי כתובות האיפ'י והMAC.

אפשר בעצם להבין שסידור החבילה במודל הTCP/IP יוצר מעין היררכיה שמסדרת את המעבר של החבילות בין המקור ליעד, כך שזה מהווה סטנדרט מסוים בתהליך התקשורת בין השרת ללקוח. תהליך הQuery and Answer הוא פשוט כמו שהוא נשמע, הלקוח שולח שאלה לשרת מסוים דרך חבילה, והשרת עונה לו באותה הצורה.