

## תרגיל 1 – מבוא לרשתות תקשורת

מגישות : יעל שמחיס 209009604, ולינוי דוארי 318416344.

### הערה –

את חלק א' של התרגיל ביצענו על המחשב של לינוי, עליו מותקן linux mint בתצורת dual boot. לצערנו, עקב תקלת חומרה במחשב זה, נאלצנו בחלק ב' לעבור למחשב של יעל, עליו הותקנו 2 מכונות וירטואליות (ubuntu).

### חלק א

2. הרצנו את השרת והלקוח והפעלנו את wireshark, כדי לסנן את החבילות שהלקוח שלח לשרת והשרת שלח ללקוח כתבנו בשורת ה- "apply a display filter" שזה שורת החיפוש את המסננים הבאים: `ip.addr == 127.0.0.1 && udp.port == 12345`. שמרנו את החבילות שסיננו בקובץ `trace1.pcapng` שמצורף להגשה.

3. בקוד השרת יש שימוש בפורט 12345. הלקוח שולח לשרת הודעה שמכילה את שמותינו ותעודות הזהות שלנו. לאחר שהשרת מקבל את ההודעה, הוא שולח אותה כפי שהיא חזרה ללקוח (למעשה זהו שרת echo).

| No. | Time        | Source    | Destination | Protocol | Length | Info                 |
|-----|-------------|-----------|-------------|----------|--------|----------------------|
| 1   | 0.000000000 | 127.0.0.1 | 127.0.0.1   | UDP      | 73     | 56294 → 12345 Len=31 |
| 2   | 0.000224962 | 127.0.0.1 | 127.0.0.1   | UDP      | 73     | 12345 → 56294 Len=31 |

הלקוח שולח את ההודעה שלו לפורט 12345 (הוא יודע אותו מראש), זה חיוני כדי לשלוח את ההודעה לשרת הרצוי. כפי שניתן לראות בתמונה לעיל, ראשית הלקוח (פורט 56294) שולח הודעה לשרת שמאזין על פורט 12345. לאחר זמן לא רב השרת (שקיבל את ההודעה שנשלחה אליו מהלקוח) מחזיר הודעה לפורט של הלקוח.

נציין כי התקשורת (העברת החבילות) בין השרת והלקוח, קרי התקשורת בין הפורטים השונים (שממומשים על ידי תקשורת בין סוקטים) מתבצעת בשכבת התעבורה. פורטים של TCP וגם של UDP מוגדרים בשכבת התעבורה במודל ה-OSI.

לפי מבחנן 1 ראינו שאחד מתפקידיה של שכבת התעבורה היא לדאוג כיצד המחשב המקבל יודע לאיזה מבין האפליקציות הרבות הרצות על המחשב החבילה מיועדת אליה. זה בדיוק מתבטא בתקשורת בין השרת והלקוח כפי שראינו בwireshark.

4. ניתן לראות בwireshark שכתובת IP המקור וכתובת IP היעד הן localhost (127.0.0.1). ניתן לראות זאת גם בצילום מסך הבא:

| No. | Time        | Source    | Destination | Protocol | Length | Info                 |
|-----|-------------|-----------|-------------|----------|--------|----------------------|
| 1   | 0.000000000 | 127.0.0.1 | 127.0.0.1   | UDP      | 73     | 56294 → 12345 Len=31 |
| 2   | 0.000224962 | 127.0.0.1 | 127.0.0.1   | UDP      | 73     | 12345 → 56294 Len=31 |

Frame 1: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0  
Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)  
Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1  
User Datagram Protocol, Src Port: 56294, Dst Port: 12345  
Data (31 bytes)

ניתן לראות בצילום מסך הבא כי לאחר הרצת פקודת ifconfig כתובת ה IP המשווייכת לכרטיס הרשת הינה גם 127.0.0.1:

```
(venv) linoy@LD:~/PycharmProjects/reshatot$ ifconfig
enp2s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.0.12 netmask 255.255.255.0 broadcast 10.0.0.255
    inet6 fe80::acfd:b1a:d9d4:448a prefixlen 64 scopeid 0x20<link>
    ether 54:e1:ad:eb:f0:24 txqueuelen 1000 (Ethernet)
    RX packets 50213 bytes 42081971 (42.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 29940 bytes 3085766 (3.0 MB)
    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 2201 bytes 217314 (217.3 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2201 bytes 217314 (217.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlp3s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.68.107 netmask 255.255.255.0 broadcast 192.168.68.255
    inet6 fe80::2de2:b319:f142:ca2 prefixlen 64 scopeid 0x20<link>
    ether e4:70:b8:fa:83:04 txqueuelen 1000 (Ethernet)
    RX packets 1325 bytes 84780 (84.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 256 bytes 31255 (31.2 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

## חלק ב

ראשית, נעזרנו בקבצים שסופקו – parent.txt, ips.txt על מנת לקנפג את "בסיס הנתונים" של שרת האב ושל שרת הבן בהתאמה. ביצענו את ההרצה באופן הבא:

בשלב הראשון, הרצנו על **מכונה וירטואלית א' (ipv4 = 10.0.2.4)** את שרת האבא כך:

```
yael@Yael:~/Downloads$ python3 server.py 55555 -l -l parent.txt
```

בשלב השני, הרצנו על **מכונה וירטואלית א' את השרת (הסטנדרטי, שרת הבן) כך:**

```
yael@Yael:~/Downloads$ python3 server.py 12345 127.0.0.1 55555 ips.txt
```

לבסוף, הרצנו על **מכונה וירטואלית ב' (ipv4 = 10.0.2.15)** את הלקוח כך:

```
yael@Yael:~/Downloads$ python3 client.py 10.0.2.4 12345
```

יש לציין שהשרת שבנינו מתפקד בדומה ל**שרת DNS**, וכראוי לאחד כזה, בהינתן דומיין של אתר (למשל, biu.ac.il) שהתקבל כבקשה מהלקוח, השרת שלנו צפוי להחזיר את כתובת ה-ip של האתר המבוקש. תחילה הזנו את הבקשה הבאה (ע"י הלקוח במכונה וירטואלית ב'):

```
yael@Yael:~/Downloads$ python3 client.py 10.0.2.4 12345
www.biu.ac.il
1.2.3.4
```

בשורה השלישית בתמונה הנ"ל רואים את הפלט שהתקבל ע"י קוד הלקוח והודפס לקונסולה. כעת, ע"י תוכנת ה-wireshark, ביצענו סינון בהתאם לכתובת ה-ip של השרת ובהתאם לפורט של השרת (שרת הבן). את הסינון שביצענו ואת התוצאות לאחר הסינון ניתן לראות בתמונה הבאה:

| ip.addr == 10.0.2.4 && udp.port == 12345 |      |              |             |           |                      |
|------------------------------------------|------|--------------|-------------|-----------|----------------------|
|                                          | Time | Source       | Destination | Protocol  | Length Info          |
|                                          | 109  | 31.140489305 | 10.0.2.15   | 10.0.2.4  | UDP 62 54223 → 12345 |
|                                          | 110  | 31.151726429 | 10.0.2.4    | 10.0.2.15 | UDP 55 12345 → 54223 |

בשורה 109 לעיל ניתן לראות את החבילה שנשלחה מהלקוח (בפורט 54223) אל השרת (בפורט 12345). בשורה 110 ניתן לראות את התגובה – את החבילה שנשלחה מהשרת שלנו אל הלקוח.

נשים לב שלפי בניית הקוד שלנו, יכול להתקיים **אחד מהתרחישים הבאים**:

- הלקוח הזין בקשה עם דומיין של אתר שנמצא בקובץ המיפויים של השרת ips.txt. במקרה זה, השרת שלנו יחזיר תשובה ישירות עם ה-ip המבוקש, כפי שהופיע בקובץ.
- הלקוח הזין בקשה עם דומיין של אתר **שאינו** נמצא בקובץ המיפויים של השרת ips.txt. במקרה זה, השרת שלנו יפנה לשרת האב (שפועל תחת API זהה כמובן) בבקשה לקבלת ה-IP של הדומיין המבוקש. שרת האב, שקונפג בעזרת קובץ המיפויים parent.txt, יחזיר את ה-IP המבוקש לשרת הסטנדרטי. כעת, שרת הבן

"ילמד" את המיפוי החדש (יוסיף אותו לקובץ ips.txt עד אשר יסתיים ה-TTL) ויחזיר את התשובה המתאימה ללקוח.

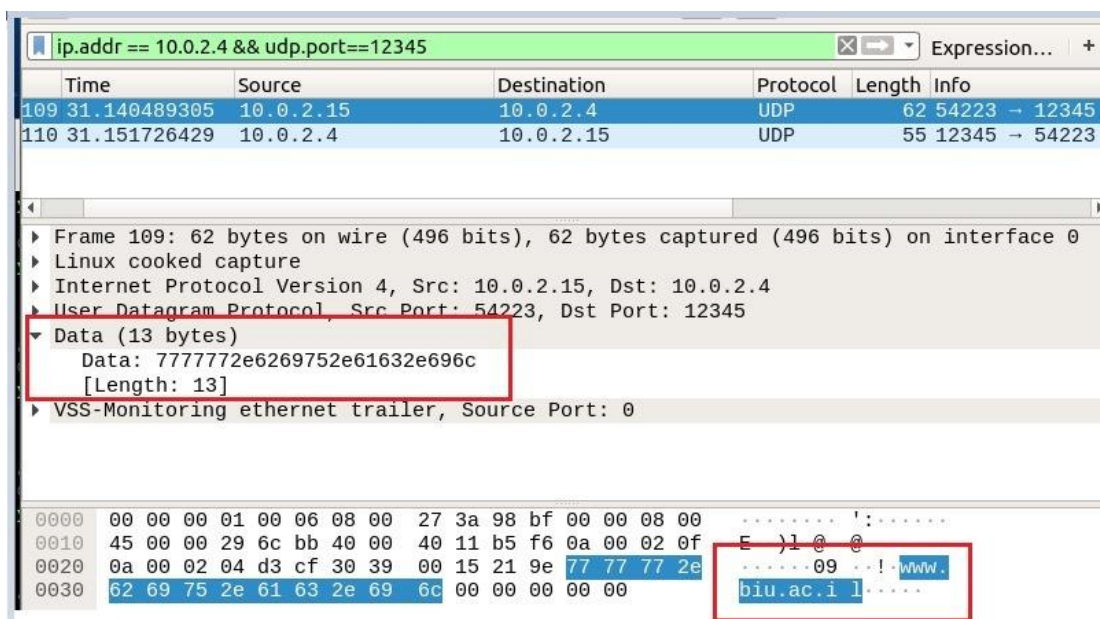
ג. הלקוח הזין בקשה עם דומיין של אתר שהשרת שלנו "למד" קודם לכן. חשוב לציין שבמקרה זה עדיין לא הסתיים ה-TTL של הבקשה שנלמדה (אחרת היא הייתה נמחקת מקובץ ה-ips.txt והשרת שלנו היה צריך ללמוד אותה מחדש). במקרה זה, השרת שלנו יחזיר את כתובת ה-IP המבוקשת, בהתאם לרשומה שלמד קודם לכן משרת האב. מסיבה זו מקרה ג' דומה מאוד למקרה א'.

כעת נרצה להראות איך נבנו ההודעות שהלקוח שלח בכל אחת מהשכבות במודל שלמדנו, בהתאם לשלושת המקרים השונים שתיארנו לעיל. נראה זאת גם באופן זהה עבור ההודעות ששלחו השרת ושרת האב.

**מקרה א':**

1. לקוח ← שרת

**שכבת האפליקציה:** המידע שמיוצג בשכבה זו הוא ההודעה שנשלחה מהלקוח. ניתן לראות בריבועים המסומנים באדום בתמונה למטה את הבקשה, הדומיין שהזין הלקוח ([www.biu.ac.il](http://www.biu.ac.il)) – תחת הלשונית data בתוכנת ה-wireshark. המידע שהלקוח שלח מופיע (משמאל למטה) גם בצורה מקודדת הקסאדצימלית.



**שכבת התעבורה:** המידע שמיוצג בשכבה זו הוא הפורטים השונים בהם השתמשנו בהרצת השרתים והלקוח. חשוב לציין כי הפורט של הלקוח נבחר רנדומלית בהתאם לפורטים התפוסים / הפנויים במ"ה ברגע ההרצה הספציפי. מטרת הפורטים היא ליצור תקשורת מבוססת סוקטים בין הלקוח לשרת. ניתן לראות זאת בתמונה הבאה, תחת הלשונית User Datagram Protocol:

| ip.addr == 10.0.2.4 && udp.port==12345 |           |             |          |        |               |
|----------------------------------------|-----------|-------------|----------|--------|---------------|
| Time                                   | Source    | Destination | Protocol | Length | Info          |
| 109 31.140489305                       | 10.0.2.15 | 10.0.2.4    | UDP      | 62     | 54223 → 12345 |
| 110 31.151726429                       | 10.0.2.4  | 10.0.2.15   | UDP      | 55     | 12345 → 54223 |

|                                                                                       |
|---------------------------------------------------------------------------------------|
| ▶ Frame 109: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0 |
| ▶ Linux cooked capture                                                                |
| ▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.4                          |
| ▶ User Datagram Protocol, Src Port: 54223, Dst Port: 12345                            |
| Source Port: 54223                                                                    |
| Destination Port: 12345                                                               |
| Length: 21                                                                            |
| Checksum: 0x219e [unverified]                                                         |
| [Checksum Status: Unverified]                                                         |
| [Stream index: 6]                                                                     |
| ▶ Data (13 bytes)                                                                     |
| ▶ VSS-Monitoring ethernet trailer, Source Port: 0                                     |

|      |                         |                         |                 |
|------|-------------------------|-------------------------|-----------------|
| 0000 | 00 00 00 01 00 06 08 00 | 27 3a 98 bf 00 00 08 00 | .....':.....    |
| 0010 | 45 00 00 29 6c bb 40 00 | 40 11 b5 f6 0a 00 02 0f | E..)l.@. @..... |
| 0020 | 0a 00 02 04 d3 cf 30 39 | 00 15 21 9e 77 77 77 2e | ....09...!.www. |
| 0030 | 62 69 75 2e 61 63 2e 69 | 6c 00 00 00 00 00 00 00 | biu.ac.i l..... |

בלשונית זו ניתן לראות את 4 השדות כפי שלמדנו בכיתה –

Source port (54223) – הפורט של הלקוח.

Destination port (12345) – הפורט של השרת.

Length (21) – גודל החבילה הכולל.

Checksum – שתפקידו זיהוי שגיאות.

**שכבת הרשת:** המידע שמיוצג בשכבה זו הוא כתובת ה-IPv4 של הלקוח (המקור) ושל השרת (היעד). בתמונה הבאה, תחת לשונית Internet Protocol Version 4:

| ip.addr == 10.0.2.4 && udp.port==12345 |           |             |          |        |               |
|----------------------------------------|-----------|-------------|----------|--------|---------------|
| Time                                   | Source    | Destination | Protocol | Length | Info          |
| 109 31.140489305                       | 10.0.2.15 | 10.0.2.4    | UDP      | 62     | 54223 → 12345 |
| 110 31.151726429                       | 10.0.2.4  | 10.0.2.15   | UDP      | 55     | 12345 → 54223 |

|                                                                 |
|-----------------------------------------------------------------|
| ▶ Linux cooked capture                                          |
| ▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.4    |
| 0100 .... = Version: 4                                          |
| .... 0101 = Header Length: 20 bytes (5)                         |
| ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) |
| Total Length: 41                                                |
| Identification: 0x6cbb (27835)                                  |
| Flags: 0x4000, Don't fragment                                   |
| Time to live: 64                                                |
| Protocol: UDP (17)                                              |
| Header checksum: 0xb5f6 [validation disabled]                   |
| [Header checksum status: Unverified]                            |
| Source: 10.0.2.15                                               |
| Destination: 10.0.2.4                                           |

|      |                         |                         |                 |
|------|-------------------------|-------------------------|-----------------|
| 0000 | 00 00 00 01 00 06 08 00 | 27 3a 98 bf 00 00 08 00 | .....':.....    |
| 0010 | 45 00 00 29 6c bb 40 00 | 40 11 b5 f6 0a 00 02 0f | E..)l.@. @..... |
| 0020 | 0a 00 02 04 d3 cf 30 39 | 00 15 21 9e 77 77 77 2e | ....09...!.www. |
| 0030 | 62 69 75 2e 61 63 2e 69 | 6c 00 00 00 00 00 00 00 | biu.ac.i l..... |

**שכבת הערוץ:** המידע שמיוצג בשכבה זו הוא כתובת ה-MAC של המקור, שמופיעה תחת הלשונית Linux cooked capture. **גילוי נאות** – חיפשנו בגוגל איך ניתן לראות



את כתובות ה-MAC ב-wireshark שהותקן על המכונה הוירטואלית (לינוקס). קריאה קצרה בלינק הבא סיפקה את התשובה עבורנו - <https://wiki.wireshark.org/SLL>

להלן צילום מסך מתוך הלינק הנ"ל:

When capturing from the "any" device, or from one of those other devices, in Linux, the libpcap doesn't supply the link-layer header for the real "hardware protocol" like Ethernet, but instead supplies a fake link-layer header for this pseudo-protocol.

Wireshark packet capture screenshot showing a UDP packet from 10.0.2.15 to 10.0.2.4. The packet details pane shows a Linux cooked capture header with a source MAC address of 08:00:27:3a:98:bf. The packet bytes pane shows the raw data of the packet.

2. שרת ← לקוח

החבילה שנשלחה מהשרת ללקוח דומה לחבילה הקודמת (מהלקוח לשרת) מבחינת האופן שבו השכבות השונות מיוצגות ע"י המידע בחבילה. כלומר, המידע המיוצג בהודעה שהחזיר השרת ללקוח דומה מאוד למה שפירטנו ב-1, אך עם שינויים קלים כמפורט להלן:

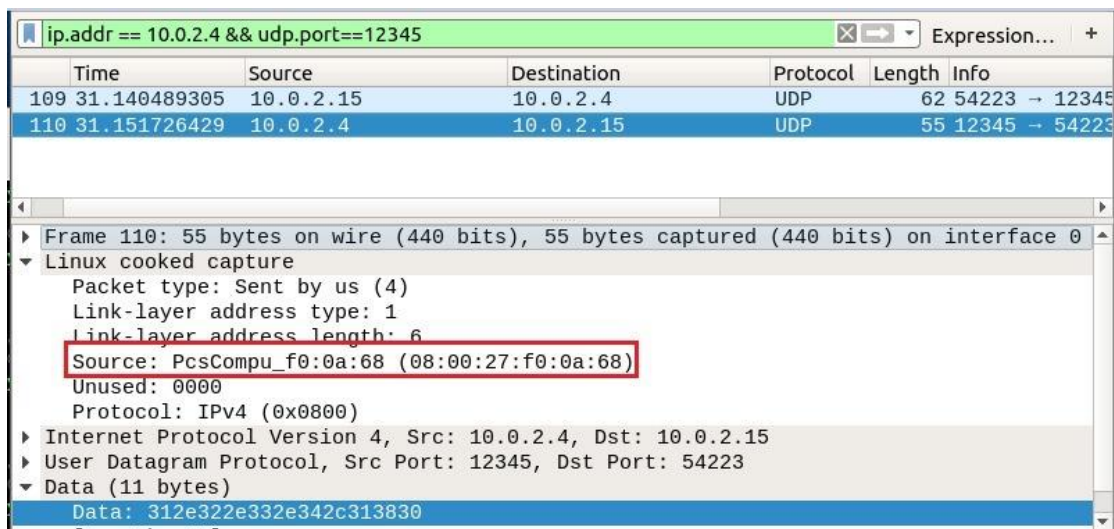
Wireshark packet capture screenshot showing a UDP packet from 10.0.2.4 to 10.0.2.15. The packet details pane shows a Linux cooked capture header with a source MAC address of 08:00:27:3a:98:bf. The packet bytes pane shows the raw data of the packet.

**שכבת האפליקציה:** ניתן לראות כמסומן בריבוע האדום התחתון את תגובת השרת כפי שגשלה ללקוח.

**שכבת התעבורה:** חל היפוך מבחינת פורט המקור ופורט היעד לעומת מה שראינו ב-1, כפי שניתן לראות בריבוע האדום האמצעי.

**שכבת הרשת:** באופן דומה, חל היפוך מבחינת כתובת ה-IP של המקור לעומת כתובת ה-IP של היעד לעומת מה שראינו ב-1, כפי שניתן לראות בריבוע האדום העליון.

**שכבת הערוץ:** בתמונה הבאה ניתן לראות שכתובת ה-MAC של המקור השתנתה:



| Time             | Source    | Destination | Protocol | Length | Info          |
|------------------|-----------|-------------|----------|--------|---------------|
| 109.31.140489305 | 10.0.2.15 | 10.0.2.4    | UDP      | 62     | 54223 → 12345 |
| 110.31.151726429 | 10.0.2.4  | 10.0.2.15   | UDP      | 55     | 12345 → 54223 |

Frame 110: 55 bytes on wire (440 bits), 55 bytes captured (440 bits) on interface 0

Linux cooked capture

- Packet type: Sent by us (4)
- Link-layer address type: 1
- Link-layer address length: 6
- Source: PcsCompu\_f0:0a:68 (08:00:27:f0:0a:68)
- Unused: 0000
- Protocol: IPv4 (0x0800)
- Internet Protocol Version 4, Src: 10.0.2.4, Dst: 10.0.2.15
- User Datagram Protocol, Src Port: 12345, Dst Port: 54223
- Data (11 bytes)
- Data: 312e322e332e342c313830

**מקרה ב':**

```
yael@Yael:~/Downloads$ python3 client.py 10.0.2.4 12345
www.biu.ac.il
1.2.3.4
mail.google.co.il
9.9.9.9
```

כאמור, במקרה זה הזין הלקוח בקשה לקבלת IP של דומיין (mail.google.co.il) שאינו נמצא בקובץ ips.txt. השרת שלנו פנה לשרת האב בבקשה לקבלת הרשומה המבוקשת, למד אותה, והחזיר את תשובתו ללקוח (9.9.9.9).



```
File Edit Search Options Help
www.biu.ac.il,1.2.3.4,180
mail.biu.ac.il,1.2.3.5,240
biu.ac.il,1.2.3.4,180
mail.google.co.il,9.9.9.9,240
```

ניתן לראות כי התווספה לקובץ ips.txt שורה חדשה שמכילה את הרשומה החדשה שהשרת למד.

| No.  | Time          | Source    | Destination | Protocol | Length | Info                 |
|------|---------------|-----------|-------------|----------|--------|----------------------|
| 109  | 31.1404893... | 10.0.2.15 | 10.0.2.4    | UDP      | 62     | 54223 → 12345 Len=13 |
| 110  | 31.1517264... | 10.0.2.4  | 10.0.2.15   | UDP      | 55     | 12345 → 54223 Len=11 |
| 4183 | 2656.42848... | 10.0.2.15 | 10.0.2.4    | UDP      | 62     | 54223 → 12345 Len=17 |
| 4184 | 2656.45732... | 127.0.0.1 | 127.0.0.1   | UDP      | 61     | 45977 → 55555 Len=17 |
| 4185 | 2656.49901... | 127.0.0.1 | 127.0.0.1   | UDP      | 55     | 55555 → 45977 Len=11 |
| 4186 | 2656.51586... | 10.0.2.4  | 10.0.2.15   | UDP      | 55     | 12345 → 54223 Len=11 |

בתצלום הנ"ל ניתן לראות סינון שונה מזה שראינו קודם לכן. הסינון כעת הוא לפי הפורט של שבת האב או לפי הפורט של שרת הבן. ניתן לראות בשורות 109,110 את אותן הבקשות כפי שראינו במקרה א'. בשורה 4183 רואים את הבקשה החדשה ששלחנו (מקרה ב') מהלקוח לשרת. ניתן גם לראות את הפנייה לשרת האב (פורט 55555) ואת התשובה שהוא מחזיר (שורה 4185). לבסוף, בשורה 4186, השרת שלנו מחזיר את התשובה ללקוח.

## 1. לקוח ← שרת

| No.  | Time          | Source    | Destination | Protocol | Length | Info                 |
|------|---------------|-----------|-------------|----------|--------|----------------------|
| 109  | 31.1404893... | 10.0.2.15 | 10.0.2.4    | UDP      | 62     | 54223 → 12345 Len=13 |
| 110  | 31.1517264... | 10.0.2.4  | 10.0.2.15   | UDP      | 55     | 12345 → 54223 Len=11 |
| 4183 | 2656.42848... | 10.0.2.15 | 10.0.2.4    | UDP      | 62     | 54223 → 12345 Len=17 |
| 4184 | 2656.45732... | 127.0.0.1 | 127.0.0.1   | UDP      | 61     | 45977 → 55555 Len=17 |
| 4185 | 2656.49901... | 127.0.0.1 | 127.0.0.1   | UDP      | 55     | 55555 → 45977 Len=11 |
| 4186 | 2656.51586... | 10.0.2.4  | 10.0.2.15   | UDP      | 55     | 12345 → 54223 Len=11 |

Frame 4183: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0

- Linux cooked capture
- Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.4
- User Datagram Protocol, Src Port: 54223, Dst Port: 12345
- Data (17 bytes)
- VSS-Monitoring ethernet trailer, Source Port: 0
- Src Port: 0

```

0000  00 00 00 01 00 06 08 00 27 3a 98 bf 00 00 08 00  ....!...
0010  45 00 00 2d 22 32 40 00 40 11 00 7c 0a 00 02 0f  E...-2@.@...
0020  0a 00 02 04 d3 cf 30 39 00 19 a4 88 6d 61 69 6c  ....09....mail
0030  2e 67 6f 6f 67 6c 65 2e 63 6f 2e 69 6c 00      .google.co.il
  
```

**שכבת האפליקציה:** כאמור, הלקוח שולח בקשה עבור הדומיין mail.google.co.il. מידע זה בא לידי ביטוי כפי שניתן לראות בלשונית ה-DATA ובריבוע האדום התחתון בתמונה.

**שכבת התעבורה:** ניתן לראות שפורט המקור הוא הפורט של הלקוח (54223) ופורט היעד הוא הפורט של השרת (12345).

**שכבת הרשת:** כתובות ה-IP של הלקוח (10.0.2.15) ושל השרת (10.0.2.4).

**שכבת הערוץ:** כפי שראינו קודם לכן, בלשונית ה-linux cooked capture ניתן לראות את כתובת ה-MAC של המקור (הלקוח):



Expression... +

udp.port==55555 || udp.port==12345

| No.  | Time          | Source    | Destination | Protocol | Length | Info                 |
|------|---------------|-----------|-------------|----------|--------|----------------------|
| 109  | 31.1404893... | 10.0.2.15 | 10.0.2.4    | UDP      | 62     | 54223 → 12345 Len=13 |
| 110  | 31.1517264... | 10.0.2.4  | 10.0.2.15   | UDP      | 55     | 12345 → 54223 Len=11 |
| 4183 | 2656.42848... | 10.0.2.15 | 10.0.2.4    | UDP      | 62     | 54223 → 12345 Len=17 |
| 4184 | 2656.45732... | 127.0.0.1 | 127.0.0.1   | UDP      | 61     | 45977 → 55555 Len=17 |
| 4185 | 2656.49901... | 127.0.0.1 | 127.0.0.1   | UDP      | 55     | 55555 → 45977 Len=11 |
| 4186 | 2656.51586... | 10.0.2.4  | 10.0.2.15   | UDP      | 55     | 12345 → 54223 Len=11 |

Frame 4183: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interface 0

Linux cooked capture

Packet type: Unicast to us (0)

Link-layer address type: 1

Link-layer address length: 6

Source: PcsCompu\_3a:98:bf (08:00:27:3a:98:bf)

Unused: 0000

Protocol: IPv4 (0x0800)

Internet Protocol Version 4, Src: 10.0.2.15, Dst: 10.0.2.4

User Datagram Protocol, Src Port: 54223, Dst Port: 12345

|      |                         |                         |       |                     |
|------|-------------------------|-------------------------|-------|---------------------|
| 0000 | 00 00 00 01 00 06 08 00 | 27 3a 98 bf 00 00       | 08 00 | .....':... ..       |
| 0010 | 45 00 00 2d 22 32 40 00 | 40 11 00 7c 0a 00       | 02 0f | E...-2@. @.. ....   |
| 0020 | 0a 00 02 04 d3 cf 30 39 | 00 19 a4 88 6d 61 69 6c |       | .....09 ....mail    |
| 0030 | 2e 67 6f 6f 67 6c 65 2e | 63 6f 2e 69 6c 00       |       | ......google. co.il |

Frame (frame) 62 bytes

Packets: 5357, Displayed: 6 (0.1%)

Profile: Default

## 2. שרת ← שרת אב

Expression... +

udp.port==55555 || udp.port==12345

| No.  | Time          | Source    | Destination | Protocol | Length | Info                 |
|------|---------------|-----------|-------------|----------|--------|----------------------|
| 109  | 31.1404893... | 10.0.2.15 | 10.0.2.4    | UDP      | 62     | 54223 → 12345 Len=13 |
| 110  | 31.1517264... | 10.0.2.4  | 10.0.2.15   | UDP      | 55     | 12345 → 54223 Len=11 |
| 4183 | 2656.42848... | 10.0.2.15 | 10.0.2.4    | UDP      | 62     | 54223 → 12345 Len=17 |
| 4184 | 2656.45732... | 127.0.0.1 | 127.0.0.1   | UDP      | 61     | 45977 → 55555 Len=17 |
| 4185 | 2656.49901... | 127.0.0.1 | 127.0.0.1   | UDP      | 55     | 55555 → 45977 Len=11 |
| 4186 | 2656.51586... | 10.0.2.4  | 10.0.2.15   | UDP      | 55     | 12345 → 54223 Len=11 |

Frame 4184: 61 bytes on wire (488 bits), 61 bytes captured (488 bits) on interface 0

Linux cooked capture

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

User Datagram Protocol, Src Port: 45977, Dst Port: 55555

Data (17 bytes)

|      |                         |                         |                     |
|------|-------------------------|-------------------------|---------------------|
| 0000 | 00 00 03 04 00 06 00 00 | 00 00 00 00 00 00 08 00 | ..... ..            |
| 0010 | 45 00 00 2d e8 36 40 00 | 40 11 54 87 7f 00 00 01 | .....@. @T.....     |
| 0020 | 7f 00 00 01 b3 99 d9 03 | 00 19 fe 2c 6d 61 69 6c | ..... ..,mail       |
| 0030 | 2e 67 6f 6f 67 6c 65 2e | 63 6f 2e 69 6c 00       | ......google. co.il |

**שכבת האפליקציה:** בתחתית התמונה לעיל ניתן לראות את הבקשה לדומיין, כפי שמועברת מהשרת שלנו אל השרת אב.

**שכבת התעבורה:** ניתן לראות פנייה מפורט 45977 לפורט 55555 של שרת האב.

**שכבת הרשת:** נשים לב שמכיוון ש-2 השרתים רצים על אותה מכונה וירטואלית, כתובת ה-IP של המקור ושל היעד זהות בשלב זה.

**שכבת הערוץ:** בדומה לסעיפים הקודמים, גם כאן ניתן לראות את כתובת ה-MAC של המקור. נשים לב שכתובת ה-MAC הפעם מופיעה כאפסים, כפי שניתן לראות בתמונה הבאה:

| No.  | Time          | Source    | Destination | Protocol | Length | Info                 |
|------|---------------|-----------|-------------|----------|--------|----------------------|
| 109  | 31.1404893... | 10.0.2.15 | 10.0.2.4    | UDP      | 62     | 54223 → 12345 Len=13 |
| 110  | 31.1517264... | 10.0.2.4  | 10.0.2.15   | UDP      | 55     | 12345 → 54223 Len=11 |
| 4183 | 2656.42848... | 10.0.2.15 | 10.0.2.4    | UDP      | 62     | 54223 → 12345 Len=17 |
| 4184 | 2656.45732... | 127.0.0.1 | 127.0.0.1   | UDP      | 61     | 45977 → 55555 Len=17 |
| 4185 | 2656.49901... | 127.0.0.1 | 127.0.0.1   | UDP      | 55     | 55555 → 45977 Len=11 |
| 4186 | 2656.51586... | 10.0.2.4  | 10.0.2.15   | UDP      | 55     | 12345 → 54223 Len=11 |

|      |                         |                         |                 |
|------|-------------------------|-------------------------|-----------------|
| 0000 | 00 00 03 04 00 06 00 00 | 00 00 00 00 00 00 08 00 | .....           |
| 0010 | 45 00 00 2d e8 36 40 00 | 40 11 54 87 7f 00 00 01 | E...6@. @.T.... |
| 0020 | 7f 00 00 01 b3 99 d9 03 | 00 19 fe 2c 6d 61 69 6c | .....,mail      |
| 0030 | 2e 67 6f 6f 67 6c 65 2e | 63 6f 2e 69 6c          | .google.co.il   |

3. שרת אב ← שרת: דומה מאוד לשלב 2, רק בהיפוך של הפורטים של המקור והיעד. כמו כן, המידע בשכבת האפליקציה השתנה – שרת האב שלח את הרשומה הרלוונטית לשרת הבן, כפי שנראה בתמונה הבאה:

| No.  | Time          | Source    | Destination | Protocol | Length | Info                 |
|------|---------------|-----------|-------------|----------|--------|----------------------|
| 109  | 31.1404893... | 10.0.2.15 | 10.0.2.4    | UDP      | 62     | 54223 → 12345 Len=13 |
| 110  | 31.1517264... | 10.0.2.4  | 10.0.2.15   | UDP      | 55     | 12345 → 54223 Len=11 |
| 4183 | 2656.42848... | 10.0.2.15 | 10.0.2.4    | UDP      | 62     | 54223 → 12345 Len=17 |
| 4184 | 2656.45732... | 127.0.0.1 | 127.0.0.1   | UDP      | 61     | 45977 → 55555 Len=17 |
| 4185 | 2656.49901... | 127.0.0.1 | 127.0.0.1   | UDP      | 55     | 55555 → 45977 Len=11 |
| 4186 | 2656.51586... | 10.0.2.4  | 10.0.2.15   | UDP      | 55     | 12345 → 54223 Len=11 |

|      |                         |                         |       |                 |
|------|-------------------------|-------------------------|-------|-----------------|
| 0000 | 00 00 03 04 00 06 00 00 | 00 00 00 00 00 00 52 54 | 08 00 | .....RT..       |
| 0010 | 45 00 00 27 e8 39 40 00 | 40 11 54 8a 7f 00 00 01 |       | F...9@. @.T.... |
| 0020 | 7f 00 00 01 d9 03 b3 99 | 00 13 fe 26 39 2e 39 2e |       | .....&9.9.      |
| 0030 | 39 2e 39 2c 32 34 30    |                         |       | 9.9,240         |

4. שרת ← לקוח

