



---

## פרויקט גמר

---

קורס רשות תקשורת מהשטים



---

ניתוח תעבורת בפרוטוקול TCP/IP

---

מגישות:  
מאור סבטני - 213119282  
מאי-סולי סמכה - 212408025  
אופל אסרי - 212713390

מרצה:  
רוייטל מרבל

## דו"ח מסכם

### חלק 1 - אפיות נתונים ולכידת מנוט בעזרת Wireshark

#### 1. יצירת קובץ CSV - שכבת היישום:

בשלב הראשון בחרנו להשתמש בפרוטוקול DNS בפרויקט. יצרנו קובץ CSV המכיל שורות של הודעות מפרוטוקול DNS, כאשר כל שורה מייצגת הודעה אחת בראצ' התקשורת:

msg_id	app_protocol	src_port	dst_port	message	timestamp
1	DNS	55001	53	Query www.server	0.012
2	DNS	55001	53	Response client	0.024
3	DNS	55002	53	Query www.server	0.035
4	DNS	55002	53	Response client	0.048
5	DNS	55003	53	Query www.server	0.059
6	DNS	55003	53	Response client	0.073
7	DNS	55004	53	Query www.server	0.082
8	DNS	55004	53	Response client	0.096
9	DNS	55005	53	Query www.server	0.108

קובץ CSV כולל את השדות הבאים:

- msg\_id - מזהה הודעה
- app\_protocol - פרוטוקול היישום (DNS)
- src\_port - פורט מקור
- dst\_port - פורטיעד
- message - תוכן ההודעה (Query / Response)
- timestamp - זמן יחסית לשילוח ההודעה

הקובץ משמש כקלט למחברת Jupyter ומדמה תקשורת בין ללקוח DNS לשרת DNS. השדות src\_port ו dst\_port נכללו בקובץ CSV לצורך התאמת לשימוש במחברת Jupyter, אשר עושה שימוש ב포רטים חלק מתהלייך יצירה חבילות התעבורת. בנוסף, נעזרנו כמעט במעט בבינה המלאכותית.

## 2. עבודה במחברת Jupyter :

את קובץ ה-CSV טענו למחברת Jupyter. המחברת מרים כל שורה בקובץ ומדגימה את תהליכי האזיה של המידע בין שכבות הרשות :

- שכבת היחסים (Application) - הودעת DNS מתוך DNS
- שכבת התעבורה (Transport) - עטיפה ב프וטוקול TCP
- שכבת הרשת (Network) - עטיפה בפרוטוקול IP
- שכבת הקישור (Data Link) - שליחה דרך ממשק Loopback

### יצירת תעבורת רשות :

באמצעות Python וספריית Scrapy נוצרו חבילות TCP אמיטיות, כאשר תוכן ההודעות מתוך קובץ ה-CSV משמש כדאטה של החבילות. החבילות נשלחו דרך ממשק ה-Loopback (127.0.0.1) אל פורט יעד 53, לצורך הדמיית תקשורת DNS.

## לכידת התעבורה באמצעות Wireshark :

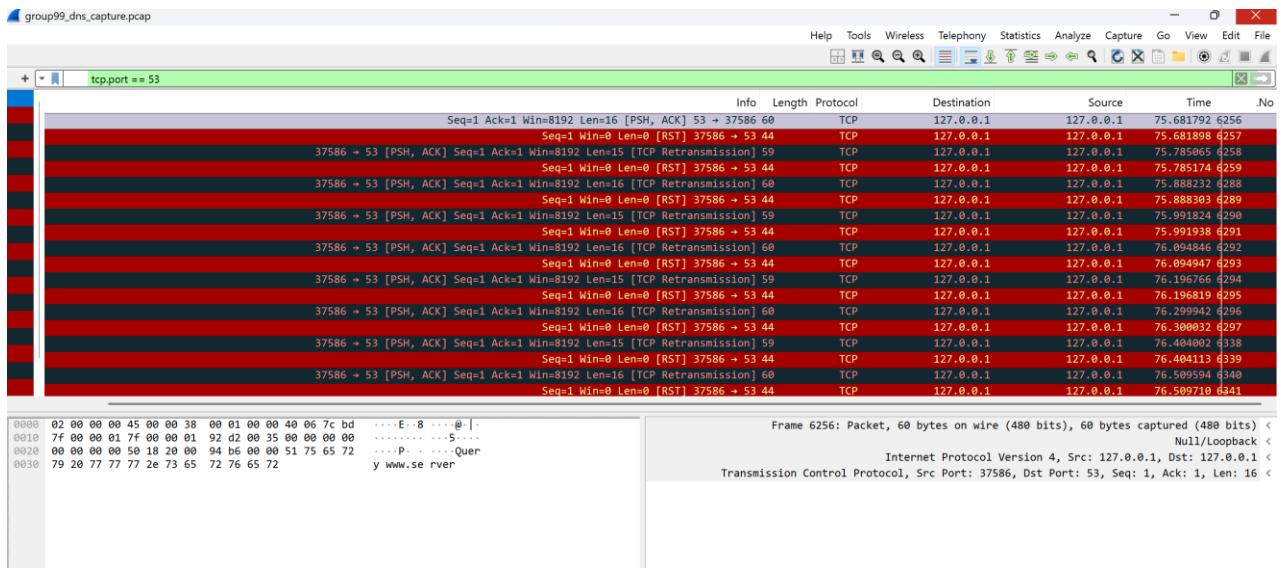
לכידת התעבורה בוצעה באמצעות Wireshark על ממשק Loopback Adapter. הגדרנו פילטר תצוגה : `tcp.port == 53` כך שייהי לנו קל יותר לגשת לפרוטוקול הרלוונטי לנו שהוא DNS. במהלך הרצת המחברת נלכדו חבילות TCP שנשלחו מకוד המחברת. באמצעות WiresharkLCDR נבדקו ושמרנו את התעבורה.

### ניתוח התעבורה שנלכדה :

בלכידה ראיינו מספר דברים חשובים בניהם : חבילות ב프וטוקול TCP, כתובות מקור ויעד : 127.0.0.1, פורט המקור ופורט יעד 53, החבילות כוללות Tag TCP מסוג PSH ו-ACK, המציגים את שליחת נתונים, ובנוסף, חבילות עם ה-tag RST, חבילות אלה נשלחות כמתואנה מהיעדר שירות DNS פעיל המאזין ב-TCP על פורט זה. נתונים אשר מייצגים את הودעות DNS (Query / Response).

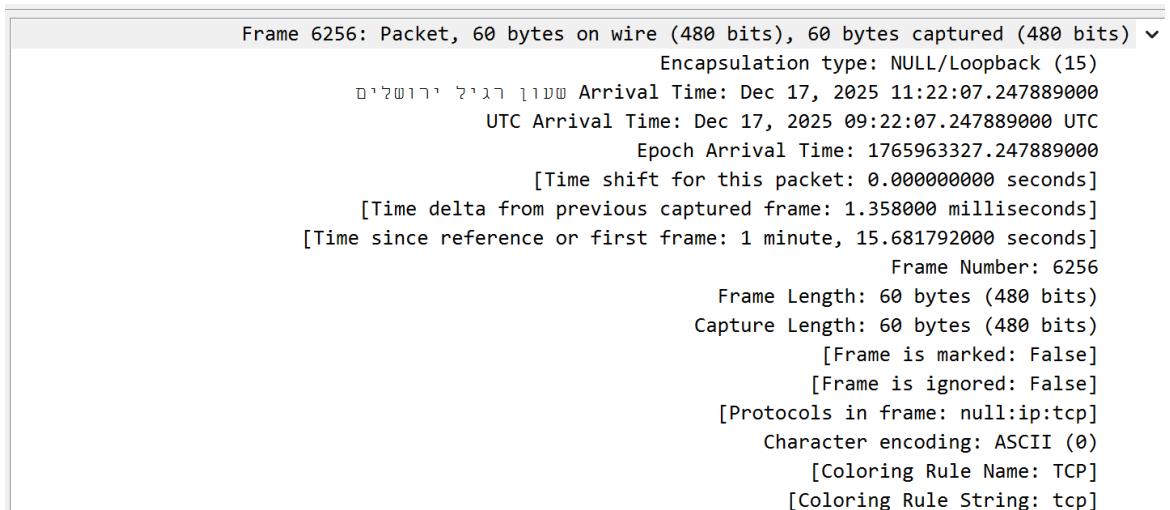
### מצורפים צילומי מסך לניתוח התעבורה שנלכדה :

חבילות TCP -> לפורט 53 (DNS)



החבילות המסווגנות באדום מעידות על חבילות הצלולות דגלי RST (אין שרת DNS שמאזין ב- TCP על ה-Loopback) או על תעבורת חריגה/כישלון חיבור. סימון זה מאפשר זיהוי מהיר של אופי התקשרות והבנת מצב החיבור.

### ניתוח ומעטפת הלויכידה:



צילום המסך מוצגת חבילת רשת (Frame) שנלכדה באמצעות Encapsulation type: Loopback, כפי שניתן לראות בשדה הלויכידה בוצעה על משק. אורץ החבילה הוא 60 בתים, והחבילה Nelcda במלואה. ברשימה הпротокולים ניתן לראות את תהליך האריזה (Encapsulation) Loopback-> IP -> TCP: (Encapsulation) המדגים את מעבר המידע משכבה התקשורת ועד שכבת התעבורת.

**ניתוח שכבת הרשת (IP):**

```

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 ▾
  Version: 4 = .... 0100
  Header Length: 20 bytes (5) = 0101 ....
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) <
    Total Length: 56
  Identification: 0x0001 (1)
  Flags: 0x0 = .... .000 <
  Fragment Offset: 0 = 0000 0000 0000 0...
    Time to Live: 64
    Protocol: TCP (6)
  Header Checksum: 0x7cbd [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 127.0.0.1
    Destination Address: 127.0.0.1
    [Stream index: 0]

```

בצילום המסך הבא מוצגת שכבת הרשת של החבילה בפרוטוקול. כתובות המקור וכתובות היעד זהות והן 127.0.0.1, דבר היכול להעיד על תעבורת פנימית. אורך כוורתת IP הוא 20 בתים, ללא פרוגמנטציה של חבילות (Fragment Offset = 0). ערך ה time to live הוא 64 ומצביע על חבילה תקנית שלא עברה דרך נטבים. השדה Protocol מצביע על TCP, וכן הנתונים מועברים לשכנת התעבורת.

 **ניתוח שכבת התעבורת (TCP):**

```

Transmission Control Protocol, Src Port: 37586, Dst Port: 53, Seq: 1, Ack: 1, Len: 16 ▾
  Source Port: 37586
  Destination Port: 53
  [Stream index: 54]
  [Stream Packet Number: 1]
  [Conversation completeness: Incomplete (40)] <
    [TCP Segment Len: 16]
  Sequence Number: 1      (relative sequence number)
    Sequence Number (raw): 0
  [Next Sequence Number: 17      (relative sequence number)]
  Acknowledgment Number: 1      (relative ack number)
    Acknowledgment number (raw): 0
  Header Length: 20 bytes (5) = .... 0101
  Flags: 0x018 (PSH, ACK) <
    Window: 8192
    [Calculated window size: 8192]
  [Window size scaling factor: -1 (unknown)]
  Checksum: 0x94b6 [unverified]
  [Checksum Status: Unverified]

```

בצילום המסך מוצגת שכבת התעבורת בפרוטוקול TCP. החבילה נשלחת מפורט מקור (פורט זמני של הלוקה) אל פורט יעד 53, המשמש לתקשורת DNS. מספר הרץ (Sequence Number) הוא 1, ואורך מקטע TCP הוא 16 בתים, המעיד על שליחת נתונים בפועל. דגלי ה-TCP המסומנים הם PSH ו-ACK, המעידים על העברת מידע מיידי ואישור קבלה של נתונים קודמים. נתונים אלו מצביעים על תקשורת TCP פעילה ותקינה כחלק מתהליך העברת הودעת DNS.

### **קישור בין CSV ללכידת Wireshark:**

כל חבילת TCP שנלכדה מייצגת הודעה DNS אחת מותוך קובץ ה-.CSV. ניתן קשר בין שורות כדוגמת `id=msg`, תוכן ההודעה (message) והזמן היחסי (timestamp) לבין החבילות שנצפו ב-Wireshark, ובכך לעקוב אחר תהליך העברת המידע משכבה היפיסום ועד לרשota.

## חלק 2 - כתיבת יישום רשות וניהוטה תעבורת של אותו יישום

### **1. מבוא - תיאור כללי של המערכת**

ב חלק זה התבקשנו לתוכנן מערכת צ'אט מבוססת Sockets תוך שימוש במטרה להבין את עקרונות התקשורת ברשת במודל של Client-Server. המערכת שיצרנו נכתבה בשפת פיתון אשר היא שפה קלה להבנה ומאפשרת לבצע את תכונות המודול בצורה נוחה לכל מותכנים. המערכת שיצרנו מאפשרת למספר לקוחות להתחבר לשרת בויזמנית, להזדהות באמצעות שם משתמש ייחודי, ולשלוח הודעות טקסט ללקוחות אחרים בזמן אמת. השרת אחראי לניהול החיבורים, לניטוב ההודעות בין הלקוחות.

### **2. מבנה המערכת**

#### **2.1 מבנה כללי**

התקשורת מתבצעת ב프וטוקול TCP, על מנת להבטיח אמינות, סדר והגעה מלאה של ההודעות. המערכת בניה משני רכיבים עיקריים :

- **Server** - השרתamazon לחיבורים נכנים, מנהל לקוחות ועביר את ה הודעות בין הלקוחות.
- **Client** - הלקוח אשר מתחבר לשרת, שולח ה הודעות ומקבל ה הודעות מלקוחות אחרים.

#### **2.2 הסבר על קובץ השרת (Server)**

בקוד שיצרנו השתמשנו בספריות socket, threading שימוש בספריות תקשורת וניהול תהליכי מקבילים לאפשרת ליצור תקשורת עם כמה משתמשים במקביל ללא הפרעות. לאחר הגדרת הספריות לשימוש יצרנו מילון CLIENTS אשר באמצעות השרת שיצרנו שומר את הלקוחות המשתמשים כרגע ובאמצעותו ממפה את התעבורת בין הלקוחות. בנוסף, יצרנו את SENDER\_LAST מטרת מנגן זו הנужד לעוזר לשרת לזכור מי שלח הודעה למי. אם מקבל הודעה ואין לו יעד מוגדר באותו רגע, השרת יאפשר לו פשוט להקליד הודעה והיא תישלח אוטומטית לאדם האחרון שפנה אליו. מטרת מנגן זו ליצר חווית משתמש רציפה. יתרה מזאת, הוספנו את מנגן זה LOCK בגלל שהשרת מנהל משתמשים רבים במקביל, עשוי להיווצר מצב שבו שני תהליכים (Threads) מנסים לעדכן את רשימת המשתמשים בדיקן, דבר המונע קריסות או שיבוש נתונים. לשרת יש כמה חשיבות בניהול התעborות, שלב ראשון - הרצה והמתנה למשתמשים. השרת פותח "שער" בכתובת שהחטנו (127.0.0.1) ופורט 8053, לאחר מכן השרת נכנס לולאה אינסופית אשר הוא ממතין לחיבור של לקוחות. כאשר מקבל השרת פותח עבورو THREAD ומשיך להמתין ללקוחות נוספים. בשלב הבא, השרת רושם את המשתמש - השרת מבקש מהלקוח להכנס שם משתמש בודק כי השם תקין או לא תפוס ולאחר מכן הוא מוסיף אותו למילון הלקוחות שיצרנו ומבקש ממנו לבחור ל��ח לדבר אליו. לאחר שהתחברו מספר לקוחות זהו שלב בו מערכת הziatlits עובדת, השרת שואל את הלקוח

עם מי הוא מתכוון לדבר, אם הלקוח קיים השיח מתחילה. בנוסף בשלב זה, השירות לא רק מעביר טקסט אלא גם מנתח את הטקסט. אם הלקוח משתמש בסימן '@' לפני שם של משתמש, השירות מזזה שמדובר ב'ティוג' של משתמש אחר. תכונה זו מאפשרת לשולח הודעה דחופה למשתמש אחר מוביל להפסיק את השיחה הנוכחי או להחליף את היעד הקבוע. במידה והלקוח אינו קיים, המשתמש נשאר בלולה וambil שוב שם לשיחה, ובמידה והלקוח מוביל exit הלוקוח מונתק מהשירות. ברגע שנבחר יעד השירות מעביר את מה שלוקוח א' כתוב ללקוח ב'. מצד המקבל מופיע שם הלוקוח השולח לפניו הודעה. בנוסף, במקרה ולקוח רוצה להחליף לקופה איתו הוא יכול לעשות זאת על ידי הקלדת //change אשר מחזיר את הלוקוח לבחור ל��וק לדבר אותו.

### 2.3 הסבר על קוביץ הלוקוח (Client)

תפקיד קוביץ הלוקוח הוא לאפשר למשתמש קצה להתחבר לשירות, לקבל ולשלוח הודעות בזמן אמת לאנשים אחרים המתחברים לשירות. כמו בקובץ השירות יש שימוש בספריות socket, threading. שבאמצעות מתחבר לשירות. השורה client.connect מטרתה ליצור חיבור בין הלוקוח לשירות. Thread-נפתח ומתחילה להמתין להודעות. המשמש מקליד שם/הודעה ב-, input-הSend-מעביר את זה לשירות, והשירות מחזיר תשובה שמודפסת על ידי ה-Thread. בתוכנת צ'אט יש בעיה-אם הלוקוח מחקה שהמשתמש יקליד הודעה (input) הוא לא יכול באותו זמן להකשב לשירות. כדי לפתור את בעיה זאת פיצלנו את הלולאות לשניים. receive\_messages מודפסת מיד. ובנוסף, מריםים את הלולאה האחורונה. אשר מחקה שהמשתמש יכתוב משהו כדי לשלוח אותו לשירות. כדי לאפשר זרימה של ההודעות השתמשנו ב-eu\_daemon=True והגדכנו שהוא תלו依 בתוכנית של שליחת ההודעות. משמעות הדבר - שברגע שנסגור את התוכנית הראשית (למשל על ידי סגירת החלון), ההזנה להודעות ייסגר אוטומטית. בלי זה, התוכנית עלולה להישאר "תקועה" ברקע. אם השירות קורס או הקשר מונתק, ה-tru-try-except-בתוכן פונקציית הקבלה תופס את השגיאה ועוצר את הלולאה בצורה נקייה.

### 3. הוראות התקינה והרכבת

#### 3.1 דרישות מוקדמות

- Python 3.x
- מערכת הפעלה Windows / Linux / macOS

#### 3.2 הרצת השירות

- .1. פתיחת חלון / CMD
- .2. ניוט לטיקיותuproject
- .3. להריץ : python server.py
- .4. לאחר ההריצה תופיע הודעה שהשירות מازין לפורט.

3.3 הרצת ל��חות

- .1. לפתוח חלון Terminal נוסף (לכל לקוח)
  - .2. להריץ: `python client.py`
  - .3. להזין שם משתמש ייחודי
  - .4. להתחילה התוכנות עם לקוחות נוספים

#### 4. דוגמאות קלט-פלט (Screenshots)

**צילום 1 - שרת ה-TCP פועל ומאזין לחיבורים נכנים, לאחר התחברות של מספר לקוחות.**

The screenshot shows a terminal window with the following content:

```
PS C:\Users\Maor\OneDrive - Holon Institute of Technology - פיקוורפ\הובנה [חולווש]\python server.py
SERVER LISTENING on port 8053...
[CONNECTED] maor
[CONNECTED] opal
[CONNECTED] may
[CONNECTED] bar
[CONNECTED] revital
```

**צלילום 2 - התחרבות ממספר לקוחות לשרת באמצעות היישום Client, הזדהות עם שם משתמש והמתנה לבחירת יעד לשיחת:**

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Maor\OneDrive - Holon Institute of Technology\python\client.py
Enter your username:
PS C:\Users\Maor\OneDrive - Holon Institute of Technology\python\client.py
PS C:\Users\Maor\OneDrive - Holon Institute of Technology\python\client.py
PS C:\Users\Maor\OneDrive - Holon Institute of Technology\python\client.py
Enter your username:
PS C:\Users\Maor\OneDrive - Holon Institute of Technology\python\client.py
PS C:\Users\Maor\OneDrive - Holon Institute of Technology\python\client.py
PS C:\Users\Maor\OneDrive - Holon Institute of Technology\python\client.py
Enter your username:
Opal

--> Enter username to chat with (or type 'exit' to quit):

```

### צילום 3 - התכונות דו-כיוונית בין שני לקוחות שונים עם מגנון תגובה אוטומטית:

מאור בוחרת לשולח למאי הודעה:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Maor\OneDrive - Holon Institute of Technology\python\client.py
Enter your username:
Maor

--> Enter username to chat with (or type 'exit' to quit):
May
--- Chat started with may. Type '/change' to switch user ---
Tip: send to a specific user using @username (example: hi @may)

Hey May:)

```

נפתח צ'אט - ההודעה מגיעה למאי, והוא עונה לה באופן אוטומטי ללא צורך בבחירה:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Maor\OneDrive - Holon Institute of Technology\python\client.py
Enter your username:
May

--> Enter username to chat with (or type 'exit' to quit):
[maor]: Hey May:)

Hey Maor, how are you today?

```

התשובה של Mai מגיעה חוזרת למאור - וגם מאור עונה למאי באופן אוטומטי:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Maor\OneDrive - Holon Institute of Technology\python\client.py
Enter your username:
Maor

--> Enter username to chat with (or type 'exit' to quit):
May
--- Chat started with may. Type '/change' to switch user ---
Tip: send to a specific user using @username (example: hi @may)

Hey May:)

[may]: Hey Maor, how are you today?

I'm fine, working on a special project. how about you?

```

המשך זרימת ההתכבות בין 2 הקליעינטים (מאור ומאי) :

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Maor\OneDrive - Holon Institute of Technology\holon\client> python client.py
Enter your username:
May
--> Enter username to chat with (or type 'exit' to quit):
[maor]: Hey May:)
Hey Maor, how are you today?
[maor]: I'm fine, working on a special project. how about you?
Sounds interesting, actually I'm working on something too:

```

צילום 4 - התכבות בין מסטר ללקוח במקביל, כולל ניתוב הודעות באמצעות @username והעברת הודעות דרך השירות:

שימוש ב-@ על מנת לפנות ספציפית ללקוח אחר בשירות :

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Maor\OneDrive - Holon Institute of Technology\holon\client> python client.py
Enter your username:
Maor
--> Enter username to chat with (or type 'exit' to quit):
May
--- Chat started with may. Type '/change' to switch user ---
Tip: send to a specific user using @username (example: hi @may)

Hey May:)

[may]: Hey Maor, how are you today?

I'm fine, working on a special project. how about you?

[may]: Sounds interesting, actually I'm working on something too:

Hey Opal, i miss you @opal

```

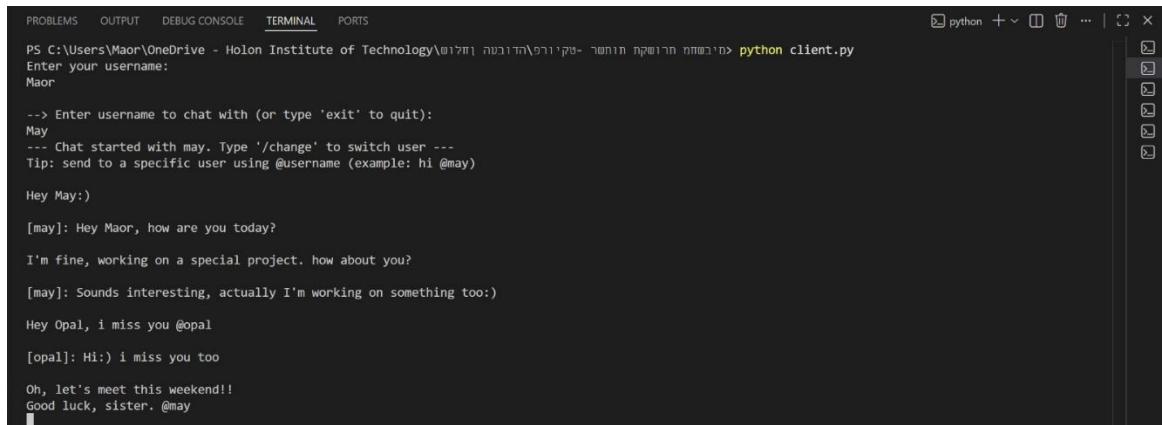
ניתן לראות שההודעה הגיעה ללקוח הספציפי (@ופל ולא למאי שנמצאת ביציאת עם מאור)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Maor\OneDrive - Holon Institute of Technology\holon\client> python client.py
Enter your username:
Opal
--> Enter username to chat with (or type 'exit' to quit):
[maor]: Hey Opal, i miss you
Hi:) i miss you too

```

מאור מקבלת הודעות מ-2 משתמשים, ועונה לשניהם:  
**תגובה אוטומטית לאופל** (הלקוח ששלח את ההודעה האחרונה ביצ'אט של מאור)  
**ותגובה ספציפית עם @ למאי**:



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Maor\OneDrive - Holon Institute of Technology\שולחן העבודה - פיקיירפ\הדגימה ו濟\client.py
Enter your username:
Maor

--> Enter username to chat with (or type 'exit' to quit):
May
--- Chat started with may. Type '/change' to switch user ---
Tip: send to a specific user using @username (example: hi @may)

Hey May:)

[may]: Hey Maor, how are you today?

I'm fine, working on a special project. how about you?

[may]: Sounds interesting, actually I'm working on something too:)

Hey Opal, i miss you @opal

[opal]: Hi:) i miss you too

Oh, let's meet this weekend!!
Good luck, sister. @may

```

**ניתן לראות שככל הודעה הגיעה לעיד המתאים לה:**



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Maor\OneDrive - Holon Institute of Technology\שולחן העבודה - פיקיירפ\הדגימה ו濟\client.py
Enter your username:
Opal

--> Enter username to chat with (or type 'exit' to quit):
[maor]: Hey Opal, i miss you

Hi:) i miss you too

[maor]: Oh, let's meet this weekend!!

```

המערכת תומכת בחיבור של לפחות 5 לקוחות במקביל, וניתנת להרחבת מספר לקוחות נוסף ללא שינוי מהותי בקוד.



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\Maor\OneDrive - Holon Institute of Technology\שולחן העבודה - פיקיירפ\הדגימה ו濟\client.py
Enter your username:
May

--> Enter username to chat with (or type 'exit' to quit):
[maor]: Hey May:)

Hey Maor, how are you today?

[maor]: I'm fine, working on a special project. how about you?

Sounds interesting, actually I'm working on something too:)

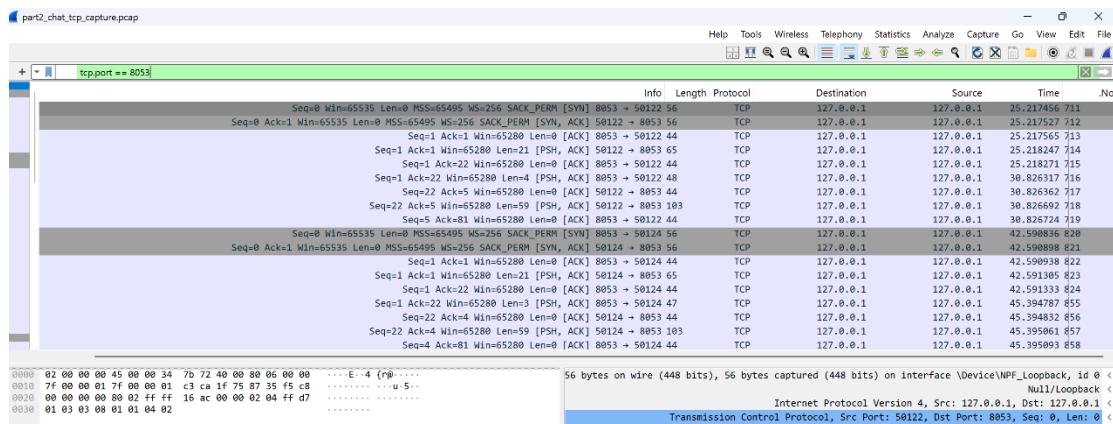
[maor]: Good luck, sister.

```

## 5. ניתוח תעבורת רשות (Wireshark) :

לצורך ניתוח תעבורת הרשות של מערכת הcz'אט, בוצעה לכידת תעבורת באמצעות תוכנת Wireshark במהלך הרצת השרת והקליניטים. הלכידה בוצעה על ממשק loopback, לאחר שהשתת והלקחות פעילים על אותו מחשב. לאחר סיום הלכידה בוצע סינון לפי הפורט של השרת (tcp.port == 8053) על מנת להטמך אך וرك בתעבורת היישום.

### צילום 1 - תעבורת TCP של יישום הcz'אט לאחר סינון לפי פורט השרת: 8053 :



בצילום זה ניתן לראות את כל חבילות TCP השויות לישום הcz'אט. התעבורת מסוננת לפי הפורט 8053, עליו נמצא השרת. ניתן להבחין בחבילות מסווג SYN, SYN-ACK ו-ACK המעודדות על הקמת חיבור TCP, וכן חבילות PSH/ACK המעדמות על העברת נתונים (הודעות cz'אט) בין הלקחות לשרת

### צילום 2 - שכבת הרשת (IP) :

<pre> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 Version: 4 = .... 0100 Header Length: 20 bytes (5) = 0101 .... Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) &lt; Total Length: 52 Identification: 0x7b72 (31602) Flags: 0x2, Don't fragment = .... .010 &lt; Fragment Offset: 0 = 0000 0000 0000 0... Time to Live: 128 Protocol: TCP (6) Header Checksum: 0x0000 [validation disabled] [Header checksum status: Unverified] Source Address: 127.0.0.1 Destination Address: 127.0.0.1 [Stream index: 0] </pre>
---

בצילום זה מוצגת שכבת הרשת (IP) של אחת החבילות. ניתן לראות כי מדובר בפרוטוקול IPv4 כאשר כתובות המקור וכתוות היעד הן 127.0.0.1, דבר המעיד על תקשורת מקומית (loopback).

שדה ה-TTL מוגדר לערך 128, והחטילה אינה מפורקת ( $= 0$ ). בנוסף, שדה ה- Fragment Offset מציין כי הנתונים מועברים ל프וטוקול TCP, בהתאם למימושם协议.

### צילום 3 - שכבת התעבורה (TCP)

```
Transmission Control Protocol, Src Port: 50122, Dst Port: 8053, Seq: 0, Len: 0
  Source Port: 50122
  Destination Port: 8053
  [Stream index: 4]
  [Stream Packet Number: 1]
  [Conversation completeness: Complete, WITH_DATA (31)] <
    [TCP Segment Len: 0]
    Sequence Number: 0      (relative sequence number)
    Sequence Number (raw): 2268460488
    [Next Sequence Number: 1      (relative sequence number)]
    Acknowledgment Number: 0
    Acknowledgment number (raw): 0
    Header Length: 32 bytes (8) = .... 1000
    Flags: 0x002 (SYN) <
      Window: 65535
      [Calculated window size: 65535]
      Checksum: 0x16ac [unverified]
      [Checksum Status: Unverified]
      Urgent Pointer: 0
    Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), No-Operation (N <
      [Timestamps] <
        [Client Contiguous Streams: 1]
        [Server Contiguous Streams: 1]
```

בצילום זה מוצגת שכבת התעבורה (TCP) של החטילה. ניתן לראות את פורט המקור (פורט זמן של הלקוח) ואת פורט היעד 8053, עליו נמצא השירות. החטילה מכילה דגל SYN, המעיד על תחילת תהליך הקמת חיבור TCP בין הלקוח לשרת. כמו כן מוצגים מספרי הרץ (Sequence Number) והחלון (Window Size), אשר משמשים את מגנוני האמינות ובקורת הזרימה של TCP. ניתוח התעבורה מראה כי מערכת הcz'אט עשויה שימוש בפרוטוקול TCP מעל IPv4 ומקיימת חיבורים אמינים בין הלקוחות לשרת. תהליכי הקמת החיבור, העברת הנתונים וניהול החיבורים תואמים את אופן הפעולה של הקוד שנכתב, ומדגימים תקשורת מלאה עד שכבת הרשת (IP כולל) ושכבת התעבורה (TCP).

### חלק 3 - שימוש במבנה מלאכותית במהלך העבודה

במהלך העבודה על הפרויקט נעשה שימוש חלקי במבנה מלאכותית, בהתאם להנחיות הקורס.

השימוש במבנה מלאכותית נעשה לצרכים הבאים :

- ייצרת נתוני דוגמה המודדים תüberות רשות (DNS) לצורך בניית קובץ CSV .
- סיווג בניסוח הסברים לדוח (הברחת מושגים, קיצור ניסוחים).
- בדיקה והבנה של פלטיהם מי-TCP (Frame, IP, TCP) Wireshark לצורך ניתוח נכון.
- סיווג חלקם בהסבירים על הספריות והפונקציות שניתן לשימוש לשם כתיבת הקוד.
- בנוסף, נעזרנו כדי לטפל בשגיאות בקוד.

דוגמאות לפרומפטים שנעשה בהם שימוש :

1. "תעזר לי ליצור קובץ CSV עם הודעות DNS "
2. "תסביר לי בקצרה מה המשמעות של TCP PSH ו-ACK שמופיעים ב-Wireshark".
3. "תעזר לי לנתח סיכום קצר וברור לניתוח שכבת TCP "
4. "תבדוק את תקינות הקוד שלי ותן לי הצעות לשיפור"
5. "איך אני יכולה לשפר את הפונקציה except, try בקובץ"
6. "תסביר לי מה ההבדל בין פורט 53 לפורט 8053"