

פרויקט ברשתות תקשורת

מגשים:

לירוי לייבל (212289987)

מיכאל וקנין (207973363)

רועי מימון (207972290)

חלק 1: אריזה ולכידת מנות (Encapsulation & Capture)

1. יצירת קובץ הנתונים (CSV)

בשלב הראשון של הפרויקט, נדרשנו לייצר תעבורה מדומיית של אפליקציה. לצורך כך, יצרנו קובץ CSV (בשם group212289987_http_input.csv) המכיל רשימה של הודעות בפרוטוקול HTTP, הקובץ נוצר בסיוע כלי AI כדי להבטיח פורמט תקין ונתונים מגוונים, והוא כולל את השדות הבאים: מזהה הודעה, פרוטוקול, אפליקציית מקור, אפליקציית יעד תוכן ההודעה וחותמת זמן.

תהליך אריזת המנות (Encapsulation)

באמצעות מחברת Jupyter ביצענו סימולציה ידנית של תהליך הכימוס (Encapsulation) המתרחש במערכת ההפעלה:

1. **שכבת היישום (Application):** חילצנו את המידע הגולמי (ההודעה) מתוך ה-CSV.
2. **שכבת התעבורה (Transport):** לכל הודעה הוספנו כותרת (Header) של TCP/UDP הכוללת את פורט המקור ופורט היעד.
3. **שכבת הרשת (Network):** עטפנו את המגזר בכותרת IP הכוללת את כתובות ה-IP של המקור והיעד.
4. **שידור:** שלחנו את המנות המוכנות דרך ממשק הרשת, כך שתוכנת Wireshark תוכל ללכוד אותן.

ניתוח לכידת התעבורה (Wireshark)

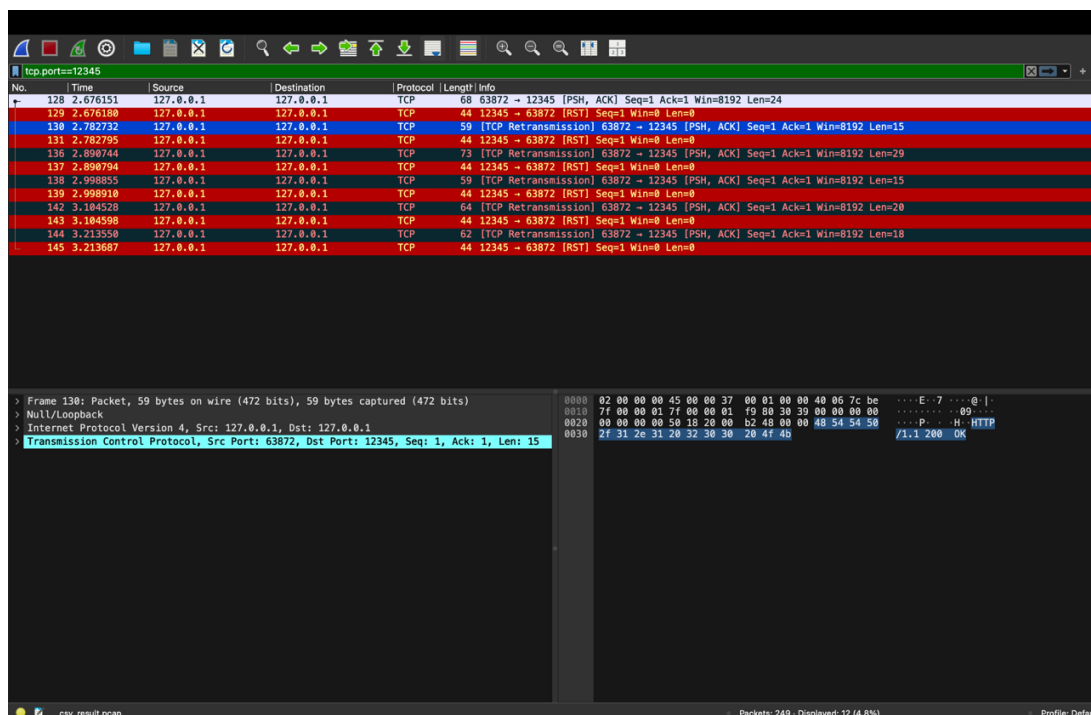
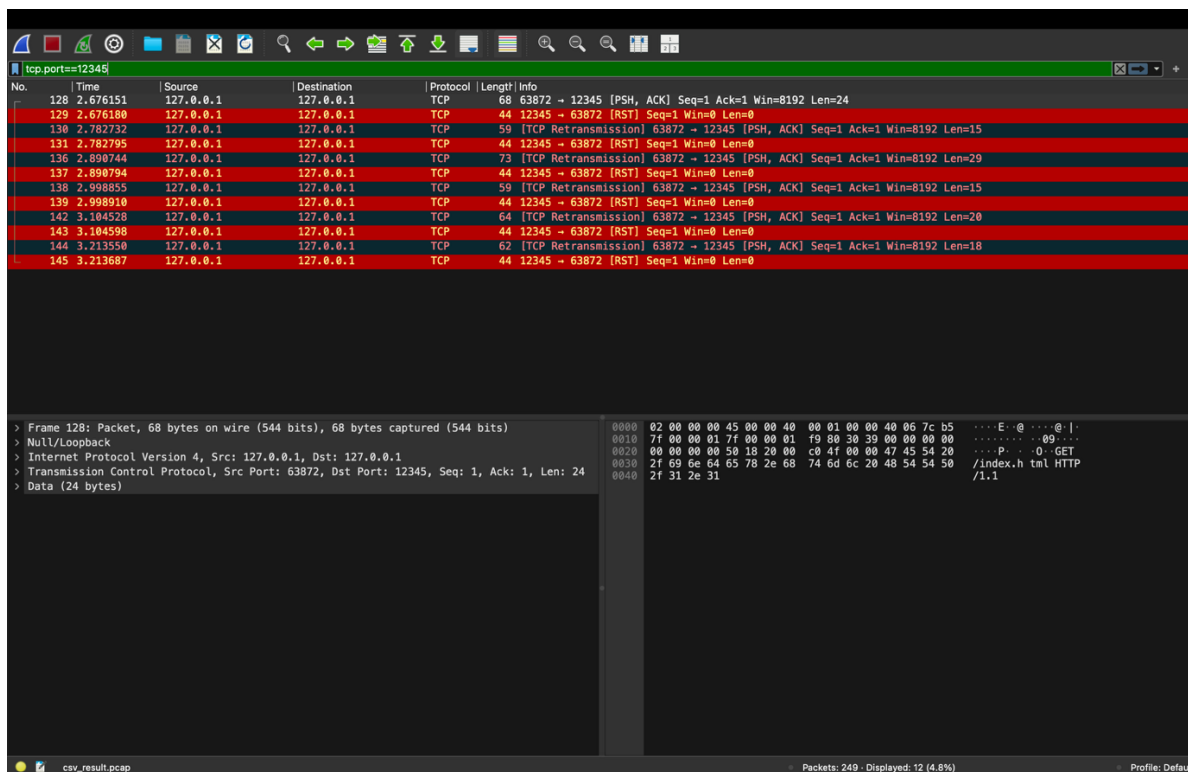
לאחר הרצת המחברת, ניתחנו את קובץ הלכידה csv_result.pcap.

זיהוי הפרוטוקול: Wireshark זיהה את המנות כ-TCP/HTTP-בהתבסס על הפורטים שהוגדרו.

תוכן המנה (Payload): בתוך גוף המנה, ניתן לראות בבירור את הטקסט שגולקח מקובץ ה-CSV.

תובנה: התרגיל המחיש כיצד כל שכבה מוסיפה את המידע הדרוש לה (Headers) ללא תלות בתוכן המידע עצמו.

לכידת מנות ה- CSV ב Wireshark:



חלק 2: יישום צ'אט וניתוח תעבורה

מבנה המערכת והקוד

פיתחנו מערכת צ'אט מבוססת Sockets ב-Python הפועלת בארכיטקטורת שרת-לקוח (Client-Server). המערכת אינה משתמשת בספריות ניהול שרתים חיצוניות, אלא מממשת את התקשורת באופן ישיר.

רכיבי המערכת:

1. השרת (server.py) :

מנהל את כל התעבורה

ריבוי תהליכים (Multi-threading): עבור כל לקוח שמתחבר, השרת פותח Thread נפרד. מנגנון זה מאפשר לשרת לטפל במספר רב של לקוחות במקביל (מעל 5 כנדרש) מבלי לחסום את התקשורת עם לקוחות אחרים.

השרת מנהל רשימת משתמשים מחוברים ומבצע ניתוב (Routing) של הודעות מלקוח אחד לאחר.

2. הלקוח (client.py / Gui_Client.py) :

פותח חיבור TCP לשרת ומפריד בין תהליכי השידור והקליטה באמצעות Threads, מה שמאפשר קבלת הודעות בזמן אמת.

ממשק גרפי (GUI): כחלק מבנוס, פיתחנו ממשק משתמש באמצעות ספריית tkinter המקל על השימוש, מציג את היסטוריית הצ'אט ומפריד ויזואלית בין הודעות נכנסות ליוצאות.

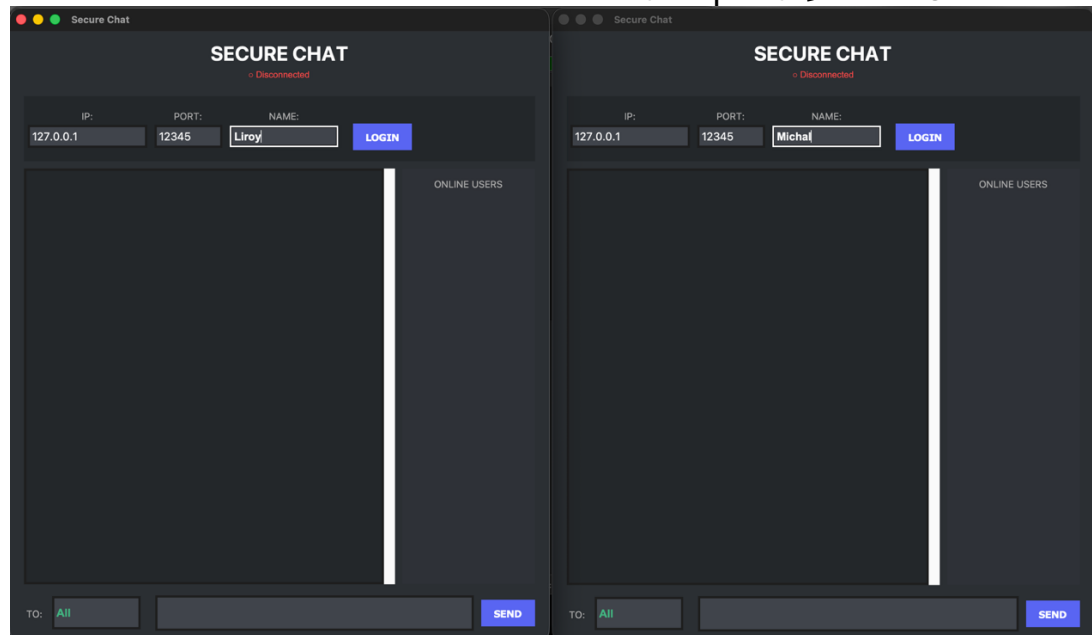
הוראות התקנה והרצה:

המערכת מתבססת על ספריות סטנדרטיות של Python ואינה דורשת התקנות מיוחדות.

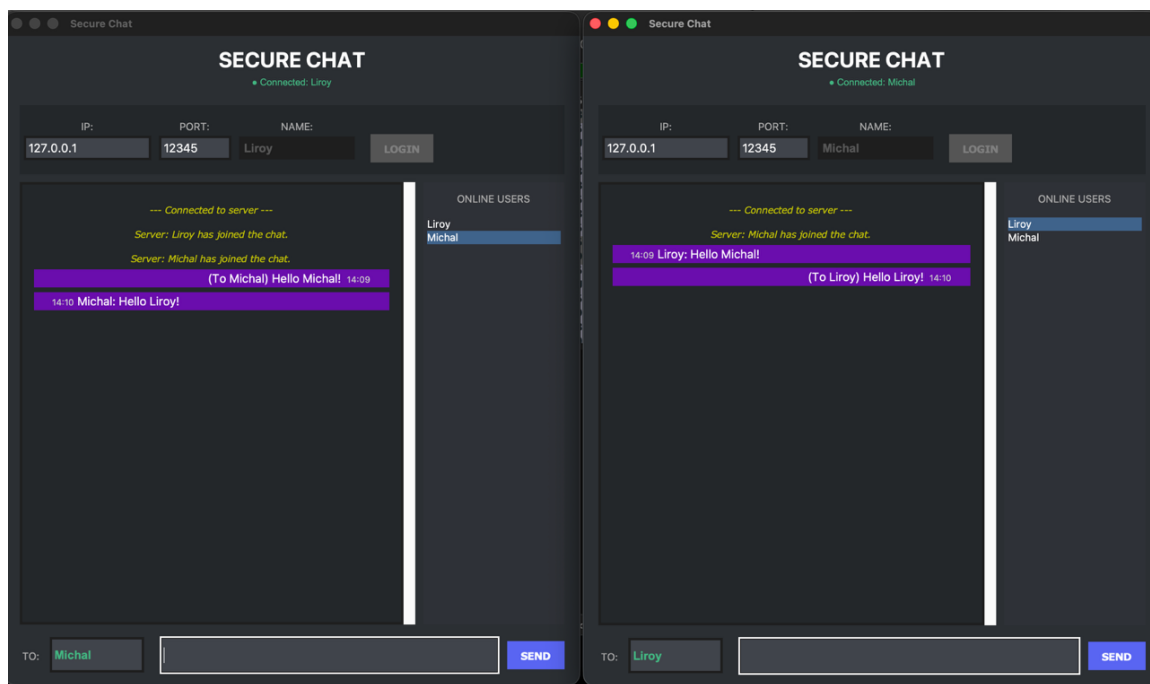
1. יש להפעיל את השרת באמצעות הפקודה: `python server.py`. השרת ידפיס הודעה כי הוא מאזין לחיבורים.
2. עבור כל משתמש, יש להפעיל את הלקוח: `python Gui_Client.py` :
3. בחלון שנפתח, יש להזין שם משתמש ולהתחיל בשיחה.

דוגמאות קלט ופלט:

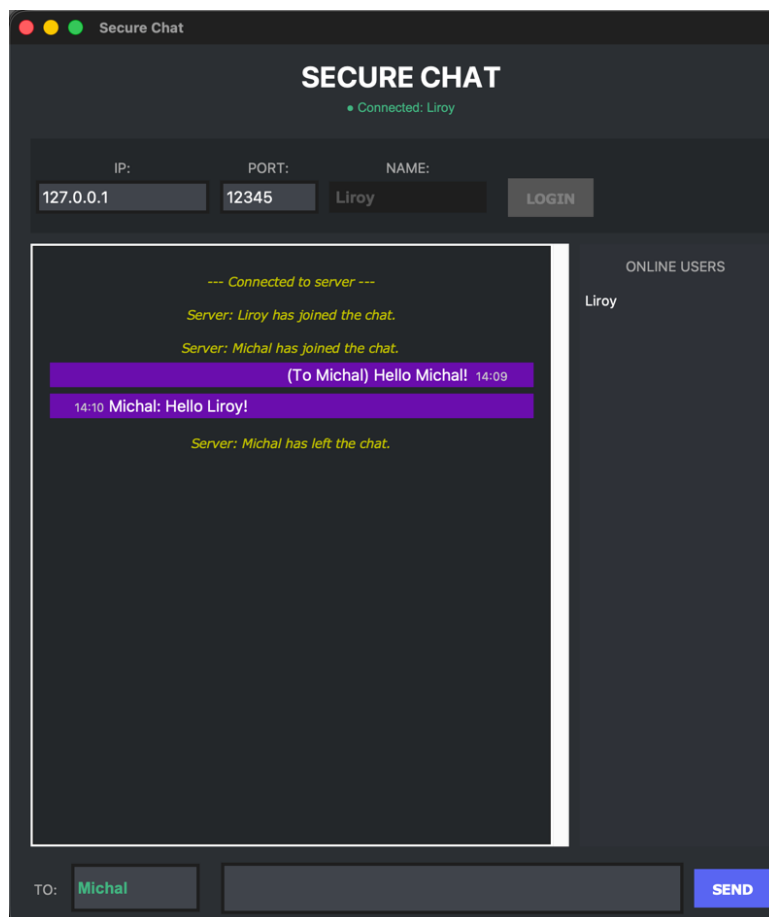
א. ממשק המשתמש (GUI) בתמונה הבאה ניתן לראות את חלון הצ'אט שפיתחנו. הממשק מאפשר חיבור נוח וצפייה בהודעות בזמן אמת.



ב. תרחיש שיחה בין לקוחות בתרחיש זה הפעלנו שני לקוחות במקביל. ההודעה שנשלחה מלקוח א' התקבלה מיידית אצל לקוח ב'.



ג. לוג השרת השרת מציג בזמן אמת את הסטטוס של המחוברים והודעות מערכת כולל התראה על עזיבה והצטרפות לצ'אט.



ניתוח תעבורת היישום

בניתוח קובץ הלכידה Chat_Analyze.pcap ניתן לראות את שלבי הפרוטוקול: TCP

1. יצירת הקשר (Handshake): רואים את תהליך ה SYN, SYN-ACK, ACK בפתיחת התוכנה.
2. העברת מידע (Data Transfer): הודעות הצ'אט (כגון שמות המשתמשים והתוכן) מועברות במנות עם דגל, PSH (Push) המורה על העברה מיידית לשכבת האפליקציה. מיד לאחר מכן מתקבל אישור ACK.

3. סיום קשר: בעת סגירת החלון, נשלחת מנת FIN/RST לסגירת ה-Socket

הודעת הצטרפות:

The image displays two screenshots of the Wireshark network protocol analyzer. The top screenshot shows a list of captured packets, with packet 16 selected. The bottom screenshot provides a detailed view of packet 16, which is a Transmission Control Protocol (TCP) segment. The packet details pane shows the following information:

- Frame 16: Packet, 91 bytes on wire (728 bits), 91 bytes captured (728 bits)
- Null/Loopback
- Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
- Transmission Control Protocol, Src Port: 12345, Dst Port: 61018, Seq: 1, Ack: 6, Len: 35
- Data (35 bytes)

The packet data section shows the raw bytes of the TCP segment, which is a standard query response for a PTR record. The data is displayed in hexadecimal and ASCII format. The ASCII column shows the text "WE:Q u: Server: Liroy ha s joined the cha t.", which is a message from a user named Liroy.

הודעות בין משתמשים:

Apply a display filter ... <34/>

No.	Time	Source	Destination	Protocol	Length	Info
33	137.492006	127.0.0.1	127.0.0.1	TCP	56	61019 → 12345 [ACK] Seq=7 Ack=61 Win=408320 Len=0 TSval=3654444784 TSecr=1036117591
34	150.007138	10.100.102.26	10.100.102.255	UDP	76	57621 → 57621 Len=44
35	160.409390	127.0.0.1	127.0.0.1	TCP	76	61018 → 12345 [PSH, ACK] Seq=6 Ack=113 Win=408320 Len=20 TSval=1373033354 TSecr=1464158800
36	160.409455	127.0.0.1	127.0.0.1	TCP	56	12345 → 61018 [ACK] Seq=113 Ack=26 Win=408320 Len=0 TSval=1464181717 TSecr=1373033354
37	160.409574	127.0.0.1	127.0.0.1	TCP	87	12345 → 61019 [PSH, ACK] Seq=61 Ack=7 Win=408320 Len=31 TSval=1036140508 TSecr=3654444784
38	160.409598	127.0.0.1	127.0.0.1	TCP	56	61019 → 12345 [ACK] Seq=7 Ack=92 Win=408320 Len=0 TSval=3654467701 TSecr=1036140508
39	172.762773	127.0.0.1	127.0.0.1	TCP	74	61019 → 12345 [PSH, ACK] Seq=7 Ack=92 Win=408320 Len=18 TSval=3654480054 TSecr=1036140508
40	172.762840	127.0.0.1	127.0.0.1	TCP	56	12345 → 61019 [ACK] Seq=92 Ack=25 Win=408320 Len=0 TSval=1036152861 TSecr=3654480054
41	172.762941	127.0.0.1	127.0.0.1	TCP	87	12345 → 61018 [PSH, ACK] Seq=113 Ack=26 Win=408320 Len=31 TSval=1464194071 TSecr=1373033354
42	172.762963	127.0.0.1	127.0.0.1	TCP	56	61018 → 12345 [ACK] Seq=26 Ack=144 Win=408320 Len=0 TSval=1373045708 TSecr=1464194071
43	180.007962	10.100.102.26	10.100.102.255	UDP	76	57621 → 57621 Len=44
44	186.297309	127.0.0.1	127.0.0.1	TCP	56	61019 → 12345 [FIN, ACK] Seq=25 Ack=92 Win=408320 Len=0 TSval=3654493589 TSecr=1036152861
45	186.297373	127.0.0.1	127.0.0.1	TCP	56	12345 → 61019 [ACK] Seq=92 Ack=26 Win=408320 Len=0 TSval=1036166396 TSecr=3654493589
46	186.297587	127.0.0.1	127.0.0.1	TCP	90	12345 → 61018 [PSH, ACK] Seq=144 Ack=26 Win=408320 Len=34 TSval=1464207605 TSecr=1373045708
47	186.297617	127.0.0.1	127.0.0.1	TCP	56	61018 → 12345 [ACK] Seq=26 Ack=178 Win=408320 Len=0 TSval=1373059242 TSecr=1464207605
48	186.297632	127.0.0.1	127.0.0.1	TCP	56	12345 → 61019 [FIN, ACK] Seq=92 Ack=26 Win=408320 Len=0 TSval=1036166396 TSecr=3654493589
49	186.297639	127.0.0.1	127.0.0.1	TCP	73	12345 → 61018 [PSH, ACK] Seq=178 Ack=26 Win=408320 Len=17 TSval=1464207605 TSecr=1373059242
50	186.297666	127.0.0.1	127.0.0.1	TCP	56	61019 → 12345 [ACK] Seq=26 Ack=93 Win=408320 Len=0 TSval=3654493589 TSecr=1036166396
51	186.297670	127.0.0.1	127.0.0.1	TCP	56	61018 → 12345 [ACK] Seq=26 Ack=195 Win=408320 Len=0 TSval=1373059242 TSecr=1464207605
52	210.008868	10.100.102.26	10.100.102.255	UDP	76	57621 → 57621 Len=44
53	215.846177	127.0.0.1	127.0.0.1	TCP	56	61018 → 12345 [FIN, ACK] Seq=26 Ack=195 Win=408320 Len=0 TSval=1373088791 TSecr=1464207605
54	215.846246	127.0.0.1	127.0.0.1	TCP	56	12345 → 61018 [ACK] Seq=195 Ack=27 Win=408320 Len=0 TSval=1464237154 TSecr=1373088791

> Frame 41: Packet, 87 bytes on wire (696 bits), 87 bytes captured (696 bits)
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 12345, Dst Port: 61018, Seq: 113, Ack: 26, Len: 31
Data (31 bytes)
285072697661746529204d696368616c3a2048656c6cf204c69726f79210a
[Length: 31]

0000 02 00 00 00 45 00 00 53 00 00 40 00 40 06 00 00E: S...@...
0010 7f 00 00 01 7f 00 00 01 30 39 ee 5a 8c 6b eb e609 Z k...
0020 f0 b3 03 48 80 18 18 ec fe 47 00 00 01 01 08 0a6.....G.....
0030 57 45 d4 17 51 d6 d3 8a 28 50 72 69 76 61 74 65 WE...Q... (Private
0040 29 20 4c 69 72 6f 79 21 0a 20 48 65 6c 6c 6f 20 4d Michal : Hello M
0050 4c 69 72 6f 79 21 0a Liroy!

Chat_Analyze.pcap Packets: 56 Profile: Default

Apply a display filter ... <34/>

No.	Time	Source	Destination	Protocol	Length	Info
29	137.491991	127.0.0.1	127.0.0.1	TCP	56	61019 → 12345 [ACK] Seq=7 Ack=37 Win=408320 Len=0 TSval=3654444784 TSecr=1036117591
30	137.491997	127.0.0.1	127.0.0.1	TCP	80	12345 → 61018 [PSH, ACK] Seq=89 Ack=6 Win=408320 Len=24 TSval=1464158800 TSecr=1373010437
31	137.492001	127.0.0.1	127.0.0.1	TCP	80	12345 → 61019 [PSH, ACK] Seq=37 Ack=7 Win=408320 Len=24 TSval=1036117591 TSecr=3654444784
32	137.492004	127.0.0.1	127.0.0.1	TCP	56	61018 → 12345 [ACK] Seq=6 Ack=113 Win=408320 Len=0 TSval=1373010437 TSecr=1464158800
33	137.492006	127.0.0.1	127.0.0.1	TCP	56	61019 → 12345 [ACK] Seq=7 Ack=61 Win=408320 Len=0 TSval=3654444784 TSecr=1036117591
34	150.007138	10.100.102.26	10.100.102.255	UDP	76	57621 → 57621 Len=44
35	160.409390	127.0.0.1	127.0.0.1	TCP	76	61018 → 12345 [PSH, ACK] Seq=6 Ack=113 Win=408320 Len=20 TSval=1373033354 TSecr=1464158800
36	160.409455	127.0.0.1	127.0.0.1	TCP	56	12345 → 61018 [ACK] Seq=113 Ack=26 Win=408320 Len=0 TSval=1464181717 TSecr=1373033354
37	160.409574	127.0.0.1	127.0.0.1	TCP	87	12345 → 61019 [PSH, ACK] Seq=61 Ack=7 Win=408320 Len=31 TSval=1036140508 TSecr=3654444784
38	160.409598	127.0.0.1	127.0.0.1	TCP	56	61019 → 12345 [ACK] Seq=7 Ack=92 Win=408320 Len=0 TSval=3654467701 TSecr=1036140508
39	172.762773	127.0.0.1	127.0.0.1	TCP	74	61019 → 12345 [PSH, ACK] Seq=7 Ack=92 Win=408320 Len=18 TSval=3654480054 TSecr=1036140508
40	172.762840	127.0.0.1	127.0.0.1	TCP	56	12345 → 61019 [ACK] Seq=92 Ack=25 Win=408320 Len=0 TSval=1036152861 TSecr=3654480054
41	172.762941	127.0.0.1	127.0.0.1	TCP	87	12345 → 61018 [PSH, ACK] Seq=113 Ack=26 Win=408320 Len=31 TSval=1464194071 TSecr=1373033354
42	172.762963	127.0.0.1	127.0.0.1	TCP	56	61018 → 12345 [ACK] Seq=26 Ack=144 Win=408320 Len=0 TSval=1373045708 TSecr=1464194071
43	180.007962	10.100.102.26	10.100.102.255	UDP	76	57621 → 57621 Len=44
44	186.297309	127.0.0.1	127.0.0.1	TCP	56	61019 → 12345 [FIN, ACK] Seq=25 Ack=92 Win=408320 Len=0 TSval=3654493589 TSecr=1036152861
45	186.297373	127.0.0.1	127.0.0.1	TCP	56	12345 → 61019 [ACK] Seq=92 Ack=26 Win=408320 Len=0 TSval=1036166396 TSecr=3654493589
46	186.297587	127.0.0.1	127.0.0.1	TCP	90	12345 → 61018 [PSH, ACK] Seq=144 Ack=26 Win=408320 Len=34 TSval=1464207605 TSecr=1373045708
47	186.297617	127.0.0.1	127.0.0.1	TCP	56	61018 → 12345 [ACK] Seq=26 Ack=178 Win=408320 Len=0 TSval=1373059242 TSecr=1464207605
48	186.297632	127.0.0.1	127.0.0.1	TCP	56	12345 → 61019 [FIN, ACK] Seq=92 Ack=26 Win=408320 Len=0 TSval=1036166396 TSecr=3654493589
49	186.297639	127.0.0.1	127.0.0.1	TCP	73	12345 → 61018 [PSH, ACK] Seq=178 Ack=26 Win=408320 Len=17 TSval=1464207605 TSecr=1373059242

> Frame 37: Packet, 87 bytes on wire (696 bits), 87 bytes captured (696 bits)
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 12345, Dst Port: 61019, Seq: 61, Ack: 7, Len: 31
Data (31 bytes)
69 63 88 61 6c 21 0a Michal!
0000 02 00 00 00 45 00 00 53 00 00 40 00 40 06 00 00E: S...@...
0010 7f 00 00 01 7f 00 00 01 30 39 ee 5b 5b 8b f2 1f09 [[...
0020 cf ba 36 c4 80 18 18 ec fe 47 00 00 01 01 08 0a6.....G.....
0030 3d c2 3f dc d9 d2 66 f0 28 50 72 69 76 61 74 65 =?...f... (Private
0040 29 20 4c 69 72 6f 79 21 0a 20 48 65 6c 6c 6f 20 4d Michal : Hello M
0050 69 63 88 61 6c 21 0a Michal!

Chat_Analyze.pcap Packets: 56 Profile: Default

הודעת עזיבת משתמש מהצ'אט:

The image shows a Wireshark network traffic capture. The top pane displays a list of 56 packets. The bottom pane shows the details of the selected packet (No. 46), which is a TCP segment from 127.0.0.1 to 127.0.0.1, port 12345 to 61018. The data field contains a 34-byte message in UTF-8 encoding.

No.	Time	Source	Destination	Protocol	Length	Info
35	160.409390	127.0.0.1	127.0.0.1	TCP	76	61018 → 12345 [PSH, ACK] Seq=6 Ack=113 Win=408320 Len=20 TSval=1373033354 TSecr=1464158800
36	160.409455	127.0.0.1	127.0.0.1	TCP	56	12345 → 61018 [ACK] Seq=113 Ack=26 Win=408320 Len=0 TSval=1464181717 TSecr=1373033354
37	160.409574	127.0.0.1	127.0.0.1	TCP	87	12345 → 61019 [PSH, ACK] Seq=61 Ack=7 Win=408320 Len=31 TSval=1036140508 TSecr=3654444784
38	160.409598	127.0.0.1	127.0.0.1	TCP	56	61019 → 12345 [ACK] Seq=7 Ack=92 Win=408320 Len=0 TSval=3654467701 TSecr=1036140508
39	172.762773	127.0.0.1	127.0.0.1	TCP	74	61019 → 12345 [PSH, ACK] Seq=7 Ack=92 Win=408320 Len=18 TSval=3654480054 TSecr=1036140508
40	172.762848	127.0.0.1	127.0.0.1	TCP	56	12345 → 61019 [ACK] Seq=92 Ack=25 Win=408320 Len=0 TSval=1036152861 TSecr=3654480054
41	172.762941	127.0.0.1	127.0.0.1	TCP	87	12345 → 61018 [PSH, ACK] Seq=113 Ack=26 Win=408320 Len=31 TSval=1464194071 TSecr=1373033354
42	172.762963	127.0.0.1	127.0.0.1	TCP	56	61018 → 12345 [ACK] Seq=26 Ack=144 Win=408320 Len=0 TSval=1373045708 TSecr=1464194071
43	180.007962	10.100.102.26	10.100.102.255	UDP	76	57621 → 57621 Len=44
44	186.297309	127.0.0.1	127.0.0.1	TCP	56	61019 → 12345 [FIN, ACK] Seq=25 Ack=92 Win=408320 Len=0 TSval=3654493589 TSecr=1036152861
45	186.297373	127.0.0.1	127.0.0.1	TCP	56	12345 → 61019 [ACK] Seq=92 Ack=26 Win=408320 Len=0 TSval=1036166396 TSecr=3654493589
46	186.297587	127.0.0.1	127.0.0.1	TCP	90	12345 → 61018 [PSH, ACK] Seq=144 Ack=26 Win=408320 Len=34 TSval=1464207605 TSecr=1373045708
47	186.297617	127.0.0.1	127.0.0.1	TCP	56	61018 → 12345 [ACK] Seq=26 Ack=178 Win=408320 Len=0 TSval=1373059242 TSecr=1464207605
48	186.297632	127.0.0.1	127.0.0.1	TCP	56	12345 → 61019 [FIN, ACK] Seq=92 Ack=26 Win=408320 Len=0 TSval=1036166396 TSecr=3654493589
49	186.297639	127.0.0.1	127.0.0.1	TCP	73	12345 → 61018 [PSH, ACK] Seq=178 Ack=26 Win=408320 Len=17 TSval=1464207605 TSecr=1373059242
50	186.297666	127.0.0.1	127.0.0.1	TCP	56	61019 → 12345 [ACK] Seq=26 Ack=93 Win=408320 Len=0 TSval=3654493589 TSecr=1036166396
51	186.297678	127.0.0.1	127.0.0.1	TCP	56	61018 → 12345 [ACK] Seq=26 Ack=195 Win=408320 Len=0 TSval=1373059242 TSecr=1464207605
52	210.008868	10.100.102.26	10.100.102.255	UDP	76	57621 → 57621 Len=44
53	215.846177	127.0.0.1	127.0.0.1	TCP	56	61018 → 12345 [FIN, ACK] Seq=26 Ack=195 Win=408320 Len=0 TSval=1373088791 TSecr=1464207605
54	215.846246	127.0.0.1	127.0.0.1	TCP	56	12345 → 61018 [ACK] Seq=195 Ack=27 Win=408320 Len=0 TSval=1464237154 TSecr=1373088791
55	215.846482	127.0.0.1	127.0.0.1	TCP	56	12345 → 61018 [FIN, ACK] Seq=195 Ack=27 Win=408320 Len=0 TSval=1464237154 TSecr=1373088791
56	215.846537	127.0.0.1	127.0.0.1	TCP	56	61018 → 12345 [ACK] Seq=27 Ack=196 Win=408320 Len=0 TSval=1373088791 TSecr=1464237154

> Frame 46: Packet, 90 bytes on wire (720 bits), 90 bytes captured (720 bits)
> Null/Loopback
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
> Transmission Control Protocol, Src Port: 12345, Dst Port: 61018, Seq: 144, Ack: 26, Len: 3
> Data (34 bytes)
Data: 5365727665723a204d696368616c20686173206c6566742074686520636861742e0a
[Length: 34]

0000 02 00 00 00 45 00 00 56 00 00 40 00 00 06 00 00 ... E . V . . @ . . .
0010 7f 00 00 01 7f 00 00 01 30 39 ee 5a 8c 6b ec 05 09 : Z . k . .
0020 f0 b3 03 40 00 18 10 ec fe 4a 00 00 01 01 00 0a ... H j
0030 57 46 08 15 51 d7 03 cc 53 65 72 76 65 72 3a 20 W f : 0 S e r v e r :
0040 4d 69 63 68 61 6c 20 68 61 73 20 6c 65 66 74 20 M i c h a l h a s l e f t
0050 74 68 65 20 63 68 61 74 2e 0a t h e c h a t . .

Data (data.data), 34 bytes Packets: 56 Profile: Default

חלק 3: שימוש בבינה מלאכותית (AI)

במסגרת העבודה נעשה שימוש בכלי הבינה המלאכותית Google Gemini ככלי עזר לפיתוח ולמידה. השימוש בוצע באופן מבוקר תוך בדיקת התוצרים.

מטרות השימוש ודוגמאות לפרומפטים:

1. **יצירת נתונים (חלק 1):** (יצירת קובץ ה CSV-עם נתוני דמה ריאליסטיים לסימולציה.

Prompt Example: "Create a CSV file with columns: msg_id, app_protocol, src_app, dst_app, message, timestamp. Generate 10 rows representing HTTP GET requests."

2. **פיתוח השרת (Server):** סיוע בכתיבת שלד הקוד לטיפול בריבוי לקוחות. (Multi-threading)

Prompt Example: "Write a Python TCP server using 'socket' and 'threading' to handle multiple clients simultaneously."

3. **עיצוב הממשק (GUI):** עזרה בבניית הממשק הגרפי ב Tkinter-וסידור האלמנטים בחלון.

Prompt Example: "Create a GUI chat client in Python using tkinter with a scrollable text area and an input field"