

דו"ח מסכם:

מגישים :

עוורם מימון-211896774 -קבוצה:05-06

אל עמו-213385842-קבוצה : 0061305-03

עד גאנט-325155315-קבוצה: 0061305-06

חלק 1: בניית בסיס הנתונים וביוזו Encapsulation

1. מבוא

בשלב זה של הפרויקט, המטרה הייתה לדמות תהליך של אריזת נתונים (Encapsulation) משכבה האפליקציה ועד לשכבה הרשת. התהליך כלל חילוץ נתונים אמיטיים מרשת האינטרנט, ארגונם במבנה נתונים מסווג CSV ושימוש בקוד Python כדי להזירק אותם לצרעה לרשת.

2. יצירה בסיס הנתונים(CSV).

מאתר אינטרנט לא Wireshark-ב HTTP בסיס הנתונים הופק באמצעות ליצת תעבורת הנתונים שולבו בקובץ <http://neverssl.com> מאובטח כתובתו : group01_http_input.csv.

- **מבנה השדות:** הקובץ כולל שדות קבועים : msg_id, app_protocol, src_port, dst_port, message, timestamp
- **ניקוי נתונים:** בוצע ניקוי של עמודות מיותרות (כמו אורך החבילה בבתים) כדי להשאיר רק את תוכן האפליקציה הגלומי.

	dst_app	src_app	message	app_proto	timestamp	msg_id
1	web_server_brov	GET / HTT	23.29007	HTTP	1194	1
2	web_server_brov	GET /onlin	26.74492	HTTP	1416	2
3	client_brov	web_serve	26.95271	HTTP/1.1	1419	3
4	web_server_brov	GET /onlin	26.95739	HTTP	1421	4
5	client_brov	web_serve	27.1618	HTTP/1.1	1434	5
6	web_server_brov	GET /favic	27.20786	HTTP	1438	6
7	client_brov	web_serve	27.4113	HTTP/1.1	1444	7
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						

3.תהליך ה-Encapsulation-בקוד

השתמשנו בספריית **Panda** לקריאת הנתונים ובספריית **Scapy** לביצוע האזיה. התהליך התרחש בשלוש שכבות:

1. **Application Layer:** השדה message הוגדר כ-Payload (מטען) של החבילה.

2. **Transport Layer (TCP):** הוספה כוורת TCP עם פורט יעד 80, המזוינה כפורט הסטנדרטי של פרוטוקול HTTP.

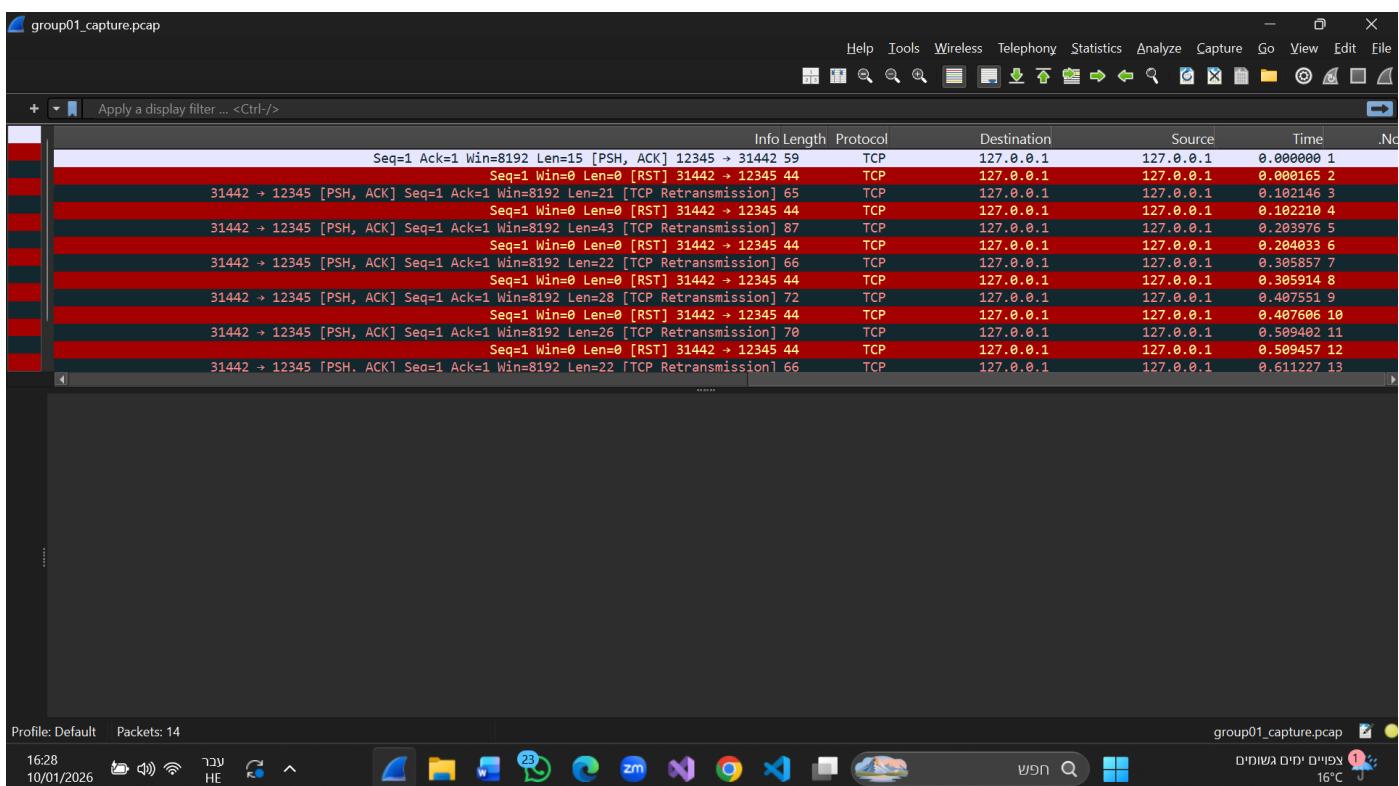
3. **Network Layer (IP):** הוספה כוורת IP עם כתובת יעד 127.0.0.1 (Local).

4. ניתוח תוצאות Wireshark

לאחר הריצת הקוד, בוצעה לצדעה במשק ה-Loopback-Wireshark.

- **זיהוי חבילות:** ניתן לראות כי חבילות נשלחו במרוחקים של שנייה אחת (באמצעות time.sleep(1)).

- **אימות הנתונים:** בלחיצה על חבילה ב-Wireshark, תחת לשונית ה-Data, ניתן לראות את הטקסט המדוק שփיע בקובץ CSV. הדבר מוכיח כי תהליך ה-Encapsulation בוצע בהצלחה והמידע הועבר משכבה האפליקציה דרך הסטאנך של מערכת הפעלה.



חלק 2 : מערכת בנק דיגיטלי

א. הסבר כללי על המערכת ומבנה הקוד

בחרנו לפתח מערכת המדממת בנק דיגיטלי במודל Client-Server מעלה פרוטוקול TCP. המערכת מאפשרת ניהול משתמשים במקביל (Multithreading) ביצוע פעולות כספיות בזמן אמת ושמירת הנתונים בסיס נתונים קבוע. בנוסף, בחרנו להטמע גם משקל משתמש גרפי (GUI). לאחר שמיושם המשקל איננו חלק מדרישות הליבה להבנת הפרוטוקול, צרפנו בנוסף את קובץ הלוקו המוקורי לצורך בחינת הלוגיקה בלבד.

מבנה הקוד:

- **server.py**-מנהל את הקשר מול בסיס הנתונים, מסזין לחיובים חדשים ויציר Thread נפרד לכל לקוח.
- **clientGUI.py**-אחראי על שליחת פקודות מובנות לשרת והציג התגובה למשתמש.
- **bank_data.pkl**-קובץ בסיס הנתונים מבוסס מיליוןים בפורמט Pickle השומר יתרות ויסודות.

ב. הוראות התקנה והרצה

1. **התקנת ספריות:** יש להריץ בטרמינל את הפקודה להתקנת המשק הגרפי:

`pip install customtkinter`

2. **הרצה השירות:** פתח טרמינל והרץ את הפקודה - `python server.py` הרשת יתחל להאזין בפורט 65432.

3. **הרצה הלוקו:** פתח טרמינל נוסף והרץ את הפקודה `python client.py`

4. **זיהוי:** התחבר עם שם משתמש קיימ או צור משתמש חדש הנתונים יישמרו אוטומטית בקובץ

ג. דוגמאות קלט ופלט

- **קלט:** המשתמש מזין username: yossi, password: 1234
- **פלט שירות:** הרשת מחזיר OK_LOGIN ומאפשר כניסה לתפריט.
- **קלט פעולה:** המשתמש בוחר "Deposit" ומזין את הסכום. 500.
- **פלט לקוח:** מוצגת הודעה: "Deposited 500.00. New balance: 1500.00."

ד. ניתוח תעבורת (עד שכבת הרשות)

היישום פועל מעל מודל ה-IP/TCP.

1. **שכבת האפליקציה:** הנתונים נשלחים כמחוזות טקסט מקודדות למשל :
(TRANSFER:dana:100)
2. **שכבת התעבורת:** שימוש ב프וטוקול TCP מתבצע חיבור (Handshake) להבטחת הגעת הנתונים לפי הסדר ולא שגיאות. נעשה שימוש בפורט 65432
3. **שכבת הרשות:** שימוש בפרוטוקול IP הנתונים נשלחים לכתובת ה-loopback המקומית (localhost) 127.0.0.1

חלק 3 : תיאור שימוש API

א. מטרות השימוש

במהלך הפROYKT נעזרנו בבינה מלאכותית (Gemini) עבור המטרות הבאות:

1. **סנכרון פרוטוקול התקשרות:** ידוא שליחת הנתונים מהלך והקריאה בשרת מתואמת כרונולוגית) מניעת Deadlocks).
2. **תיקון באגים לוגיים:** זיהוי חוסר התאמה בין פורתט הפקודות בשרת לבין הקלט בלקוח.
3. **שיפור ממתק המשתמש:** קבלת הנחיות לעבודה עם ספריית customtkinter לייצירת ממתק ויזואלי מודרני.
4. **тиיעוד:** סיווג בניסוח הסברים טכניים על מבנה בסיס הנתונים וניתוח שכבות הרשות.

ב. דוגמאות פרומפטים

- "יש לי שרת שמצויה לפקודה בפורמט LOGIN:user:pass:LOGIN אבל הלך שולח רק את השם, איך אני מסנכרן ביניהם?"
- "הסביר לי מה המשמעות של השימוש בLock-בתוך השרת ולמה זה קריטי כשיש כמה ל��חות".
- "למה אני מקבל FileNotFoundError ModuleNotFoundError על customtkinter ואיך אני מתקין את זה בתוך ה-terminal-של PyCharm?"
- "בצע Trace על שלב ה Authentication-Authorization-וראה איפה ה recv-וה send-לא מתואימים".

