

דו"ח מסכם :

מגשים :

עומר מימון -211896774-קבוצה:0061305-05

גל עמנו -213385842-קבוצה : 0061305-03

עודי גאנם-325155315-קבוצה :0061305-06

חלק 1: בניית בסיס הנתונים וביצוע Encapsulation

1.מבוא.

בשלב זה של הפרויקט, המטרה הייתה לדמות תהליך של אריזת נתונים (Encapsulation) משכבת האפליקציה ועד לשכבת הרשת. התהליך כלל חילוף נתונים אמיתיים מרשת האינטרנט, ארגון במבנה נתונים מסוג CSV, ושימוש בקוד Python כדי להזריק אותם חזרה לרשת.

2.יצירת בסיס הנתונים(CSV)

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

- **מבנה השדות:** הקובץ כולל שדות חובה כגון :
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 הגולמי.

dst_app	src_app	message	app_proto	timestamp	msg_id
web_client_brov	web_server	GET / HTTP	HTTP	23.29007	1194
web_client_brov	web_server	GET /online HTTP	HTTP	26.74492	1416
client_brov	web_server	HTTP/1.1 HTTP	HTTP	26.95271	1419
web_client_brov	web_server	GET /online HTTP	HTTP	26.95739	1421
client_brov	web_server	HTTP/1.1 HTTP	HTTP	27.1618	1434
web_client_brov	web_server	GET /favicon HTTP	HTTP	27.20786	1438
client_brov	web_server	HTTP/1.1 HTTP	HTTP	27.4113	1444

3. תהליך ה Encapsulation-בקוד

השתמשנו בספריית Panda לקריאת הנתונים ובספריית Scapy לביצוע האריזה. התהליך התבצע בשלוש שכבות:

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

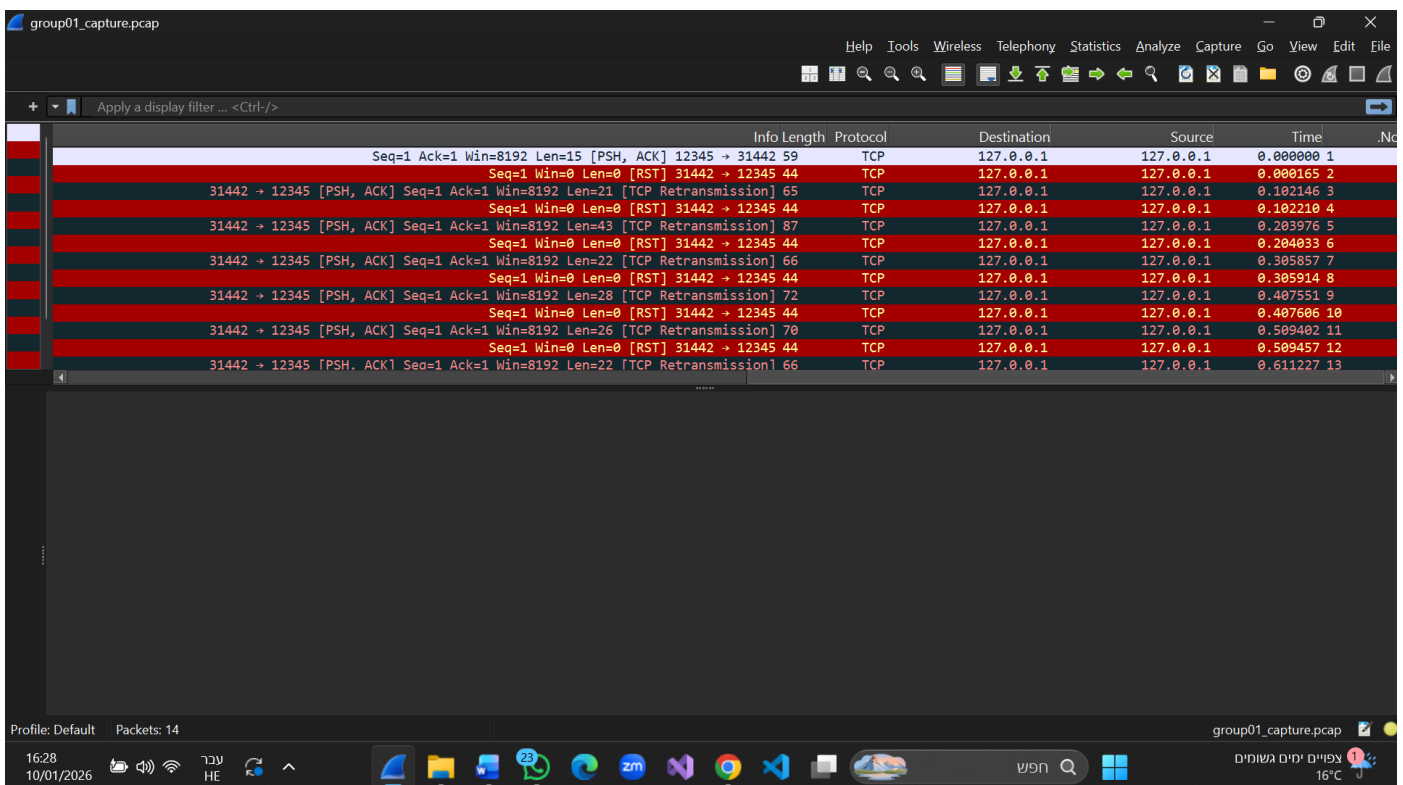
2. **Transport Layer (TCP):** הוספת כותרת TCP עם פורט יעד 80, המזוהה כפורט הסטנדרטי של פרוטוקול HTTP.

3. **Network Layer (IP):** הוספת כותרת IP עם כתובת יעד Local (127.0.0.1 (Loopback) המאפשרת למחשב לשלוח את החבילות לעצמו לצורך בדיקה.

4. ניתוח תוצאות בWireshark.

לאחר הרצת הקוד, בוצעה לכידה בממשק ה Loopback ב-Wireshark.

- **זיהוי חבילות:** ניתן לראות כי החבילות נשלחו במרווחים של שנייה אחת (באמצעות `time.sleep(1))`.
- **אימות הנתונים:** בלחיצה על חבילה ב Wireshark, תחת לשונית ה Data-ניתן לראות את הטקסט המדויק שהופיע בקובץ ה CSV. הדבר מוכיח כי תהליך ה Encapsulation בוצע בהצלחה והמידע הועבר משכבת האפליקציה דרך הסטאק של מערכת ההפעלה.



חלק 2 : מערכת בנק דיגיטלית

א. הסבר כללי על המערכת ומבנה הקוד

בחרנו לפתח מערכת המדמה בנק דיגיטלי במודל Client-Server מעל פרוטוקול TCP. המערכת מאפשרת ניהול משתמשים במקביל, (Multithreading) ביצוע פעולות כספיות בזמן אמת ושמירת הנתונים בבסיס נתונים קבוע. כערך מוסף, בחרנו להטמיע גם ממשק משתמש גרפי (GUI) עבור הלקוח. מאחר שמימוש הממשק אינו חלק מדרישות הליבה להבנת הפרוטוקול, צירפנו בנוסף את קובץ הלקוח המקורי לצורך בחינת הלוגיקה בלבד.

מבנה הקוד:

- **server.py**-מנהל את הקשר מול בסיס הנתונים, מאזין לחיבורים חדשים ויוצר Thread נפרד לכל לקוח.
- **clientGUI.py**-אחראי על שליחת פקודות מובנות לשרת והצגת התגובות למשתמש.
- **bank_data.pkl**-קובץ בסיס הנתונים מבוסס מילונים בפורמט Pickle השומר יתרות וסיסמאות.

ב. הוראות התקנה והרצה

1. **התקנת ספריות:** יש להריץ בטרמינל את הפקודה להתקנת הממשק הגרפי:
pip install customtkinter
2. **הרצת השרת:** פתח טרמינל והרץ את הפקודה - python server.py השרת יתחיל להאזין בפורט 65432.
3. **הרצת הלקוח:** פתח טרמינל נוסף והרץ את הפקודה python client.py
4. **זיהוי:** התחבר עם שם משתמש קיים או צור משתמש חדש הנתונים יישמרו אוטומטית בקובץ

ג. דוגמאות קלט ופלט

- **קלט:** המשתמש מזין username: yossi, password: 1234
- **פלט שרת:** השרת מחזיר LOGIN_OK 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 תעבורה (עד שכבת הרשת)

היישום פועל מעל מודל ה-TCP/IP-

1. **שכבת האפליקציה:** הנתונים נשלחים כמחרוזות טקסט מקודדות למשל: (TRANSFER:dana:100)
2. **שכבת התעבור:** שימוש בפרוטוקול **TCP** מתבצע חיבור (Handshake) להבטחת הגעת הנתונים לפי הסדר וללא שגיאות. נעשה שימוש בפורט 65432
3. **שכבת הרשת:** שימוש בפרוטוקול **IP** הנתונים נשלחים לכתובת ה Loopback המקומית 127.0.0.1 (localhost)

חלק 3 : תיאור שימוש AI

א. מטרות השימוש

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

1. **סנכרון פרוטוקול התקשורת:** וידוא ששליחת הנתונים מהלקוח והקריאה בשרת מתואמות כרונולוגית) מניעת. (Deadlocks
2. **תיקון באגים לוגיים:** זיהוי חוסר התאמה בין פורמט הפקודות בשרת לבין הקלט בלקוח.
3. **שיפור ממשק המשתמש:** קבלת הנחיות לעבודה עם ספריית customtkinter ליצירת ממשק ויזואלי מודרני.
4. **תיעוד:** סיוע בניסוח הסברים טכניים על מבנה בסיס הנתונים וניתוח שכבות הרשת.

ב. דוגמאות פרומפטים

- "יש לי שרת שמצפה לפקודה בפורמט LOGIN:user:pass אבל הלקוח שולח רק את השם, איך אני מסנכרן ביניהם?"
- "הסבר לי מה המשמעות של השימוש ב Lock-בתוך השרת ולמה זה קריטי כשיש כמה לקוחות".
- "למה אני מקבל ModuleNotFoundError על customtkinter ואיך אני מתקין את זה בתוך ה Terminal-של"?PyCharm
- "בצע Trace על שלב ה Authentication-וראה איפה ה recv-וה send-לא מתואמים".

