דוח מסכם: פרויקט תקשורת מחשבים - חלק 1

מגישים: נועה צדוק 214396434, לירן שטרנברג 211895578 ונעם רחמים 212551121

תאריך: 07.01.2026

קישור לGITHUB

מהו פרוטוקול HTTP?

פרוטוקול HTTP (HyperText Transfer Protocol) הוא פרוטוקול שכבת האפליקציה במודל ה-OSI, המשמש להעברת מסמכי היפר-טקסט (כגון HTML) ברחבי האינטרנט. הפרוטוקול פועל במודל Client-Server (לקוח-שרת):

- הלקוח (Client): סקריפט, כמו קובץ ה-Python, שולח בקשה לשרת.
- השרת (Server): מאזין לבקשות, מעבד אותן ומחזיר תשובה הכוללת קוד סטטוס (כגון 200 OK או 404 Not Found) ואת המידע המבוקש.

HTTP מסתמך על פרוטוקול TCP בשכבת התעבורה כדי להבטיח אמינות, סדר ושלמות הנתונים. כל בקשת HTTP מתחילה ב - Three-Way Handshake של TCP לפני שהמידע עצמו מועבר.

יצירת קובץ הCSV:

לצורך ביצוע הפרויקט, יצרנו קובץ נתונים בשם group212551121_http_input.csv.

- אופן יצירת הקובץ נוצר באמצעות שימוש בChatGPT עם הודעות המדמות תעבורת HTTP חוקית.
- מבנה הקובץ :הקובץ מכיל את השדות msg_id, app_protocol, src_app, dst_app, message, timestamp. שדות אלו משמשים כבסיס לסימולציית המידע שעובר ברשת.
- תהליך הEncapsulation והרצת הJupiter:

  בשלב זה השתמשנו במחברת (raw_tcp_ip_notebook.ipynb) כדי לדמות את תהליך אריזת הנתונים כפי שמתרחש במודל הTCP/IP

  תהליך Encapsulation הקורה בקוד במחברת לא רק "שולח" את ההודעה, אלא מדמה את תהליך ההוספה של Headers בכל שכבה, בדומה למערכת הפעלה אמיתית:

  - Headers - הם מידע טכני שמודבק לתחילת ההודעה כדי שהמחשבים ידעו איך לטפל בה.

תהליך זה מדמה התנהגות של Client המייצר תעבורה, אותה אנו רואים לאחר מכן נלכדת בכלי ה-Wireshark.

- **שכבת היישום** - הקוד לוקח את ה-Msg מה-CSV למשל את ההודעה Hello Packet ומוסיף לו כותרות של פרוטוקול HTTP.
- **שכבת התעבורה** - המידע נעטף ב TCP Header - בשלב זה מוגדרים הSrc_port והDes_port וכן דגלים ראשוניים.
- **שכבת הרשת** - החבילה מקבלת IP Header עם כתובות המקור והיעד בפרויקט זה: Loopback 127.0.0.1
- בסיום הEncapsulation הסקריפט שולח את החבילה המוכנה לרשת, שם קולטים אותה.

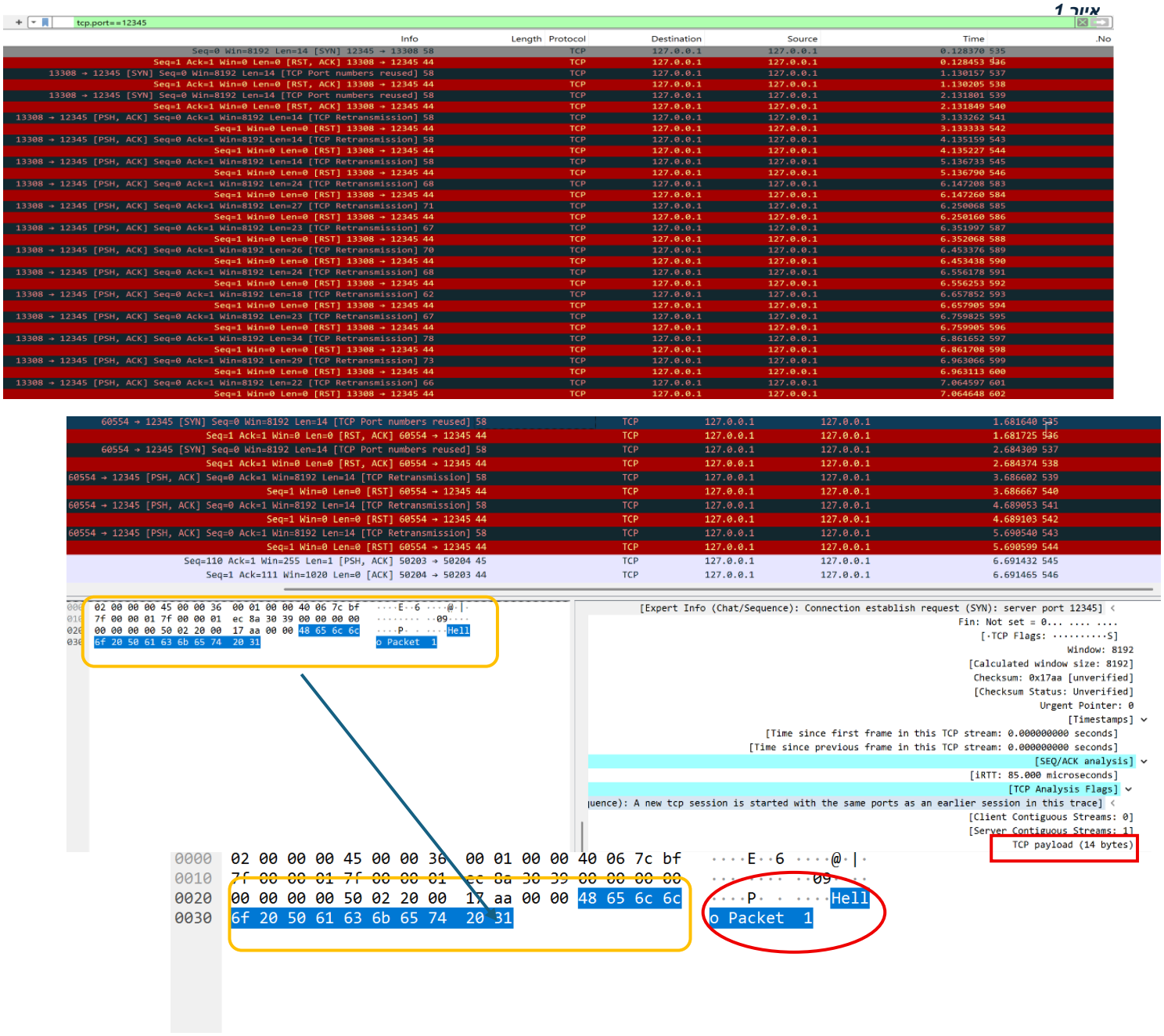
תהליך הלכידה בWIRESHARK:

כדי ללכוד את התעבורה שנוצרה על ידי הסקריפט, ביצענו את הפעולות הבאות ב - Wireshark:

1. פתחנו את תוכנת Wireshark ובחרנו במשק הרשת Adapter for loopback traffic capture מאחר והתקשורת היא מקומית בין תהליכים באותו מחשב.
2. הפעלנו מסנן תצוגה **tcp.port == 12345** מטרת המסנן היא לבודד רק את התעבורה הקשורה לפרויקט ולהתעלם מרעשי רקע של המחשב.
3. הרצנו את הסקריפט במחברת JUPITER ושמרנו את התוצאה בקובץ pcap.

ניתוח התעבורה ב-WIRESHARK:[1]

איור 1



[1] איור 1 – צילום של התעבורה במהלך הרצת סקריפט הJUPITER

בצילום המסך המצורף, ניתן לראות ניתוח של כשל בתקשורת TCP מצב של(Port Unreachable) .

בתמונה ניתן לראות ניסיון יצירת קשר בין שני פורטים על גבי המחשב המקומי 127.0.01:

- Source Port: 42407 - הלקוח שמנסה להתחבר - הסקריפט שאנחנו מריצים
- Destination Port: 12345 - השרת אליו מנסים לפנות
- SYN – דגל המבקש להתחיל חיבור TCP חדש.
- RESET = RST - דגל המסמן ניתוק מידי של הקשר. ה RST-מעיד על תקלה או על כך שהצד השני אינו פעיל. בתמונה, השרת שולח RST כתגובה ל SYN-של הלקוח.
- ACK – אישור שההתחברות הראשונית מהצד השני התבצעה ועד לרגע זה החיבור תקין.
- WIN - מנגנון בקרת זרימה המודיע לשולח כמה מקום פנוי יש למקבל.

<u>הסבר על הצבעים והדגלים שנלכדו:</u>

א. שורות עם רקע שחור וטקסט אדום:

שורות אלו מייצגות את הניסיון של הלקוח לשלוח מידע, ואת השידור החוזר עקב חוסר מענה תקין.

- SEQ = 1 - מודיע שהחיבור הראשוני הצליח מבחינת המספור, והמחשב שולח את הבית הראשון של המידע.
- LEN > 0 - בשורות אלו יש תוכן .כפי שניתן לראות ב ,Payload-ההודעה שנשלחת היא "Hello Packet 1".
- המשמעות :הלקוח מנסה לשלוח את ההודעה ,"Hello Packet 1" לא מקבל אישור (ACK) ומנסה לשלוח אותה שוב ושוב.

ב. שורות עם רקע אדום כהה :(RST - Reset) אלו הן תשובות ה"שרת" :

- RST,ACK: דגל Reset הודעה זו אומרת: "אין אף תהליך שמאזין בפורט 12345, נתק את השיחה מיד."
- WIN=0: גודל החלון הוא 0 מכיוון שהחיבור מבוטל, ולמחשב המקבל אין כוונה לקבל מידע נוסף.
- LEN = 0: הודעת שגיאה זו היא טכנית בלבד ואינה מכילה מידע משכבת האפליקציה.

```
Fragment Offset: 0 = 0000 0000 0000 0...
Time to Live: 64
Protocol: TCP (6)
Header Checksum: 0x7cb7 [validation disabled]
[Header checksum status: Unverified]
Source Address: 127.0.0.1
Destination Address: 127.0.0.1
[Stream index: 0]
nission Control Protocol, Src Port: 60554, Dst Port: 12345, Seq: 0, Ack: 1, Len: 22
Source Port: 60554
Destination Port: 12345
[Stream index: 9]
[Stream Packet Number: 27]
[Conversation completeness: Incomplete (45)]
RST: Present = .... .1..
FIN: Absent = .... 0...
Data: Present = ...1 ....
ACK: Present = ..1. ....
SYN-ACK: Absent = .0.. ....
SYN: Present = 1... ....
[Completeness Flags: R·DA·S]
```

ג. ניתוח מבנה החבילה (Packet Details): בבחינת חלונית הפרטים, ניתן לראות את חלוקת המידע לשכבות הפרוטוקול:

- <u>שכבת ה-(Internet Protocol) IP:</u>
  - Time to Live: הערך הוא 64. זהו ערך סטנדרטי המעיד שהחבילה לא עברה דרך ראוטרים, מה שמאשש שמדובר בתעבורה פנימית (Localhost).

- Header Checksum: מופיע כ-Unverified. זהו מצב תקין ולא שגיאה; מערכת ההפעלה מעבירה את חישוב "ספרת הביקורת" לכרטיס הרשת (Offloading) כדי לחסוך במשאבים, ולכן WIRESHARK (שלוכד את המידע לפני כן) לא רואה את הערך הסופי.

- <u>שכבת התעבורה (TCP):</u>
  - Ports: ניתן לראות בבירור את פורט המקור (60554) ופורט היעד (12345).
  - Len (Length): מציין את אורך המידע נטו (Payload) שנשלח בחבילה זו (22 בתים).

סיכום האירוע בתמונה 1:

- הלקוח -> פורט 42407 מנסה לשלוח מידע הודעת HTTP מתוך ה CSVלפורט 12345.
- מאחר ואין שום תהליך שמאזין בפורט 12345 (אין שרת), מערכת ההפעלה מחזירה הודעת RST.
- הלקוח, שלא קיבל סגירה מסודרת (FIN) אלא ניתוק אלים, מנסה לבצע Retransmission - שידור חוזר של המידע, עד שהוא מתייאש.