

הוכחת פרוייקט רשתות תקשורת מחשבים - חלק 1

אבייטל יוחנןוב, נאור אנידיגר

מבוא

בפרויקט זהה המטרה הייתה לחת הודעות ברמת אפליקציה, להמיר אותן לפקודות IP/TCP בצורה ידנית ולראות איך הן נראות בפועל בתחום Wireshark. בנוסף היו צריכים ליצור קשר בין השורות בקובץ CSV לבין הפקודות שנוצרו ונקלטו בשרת.

טעינת קובץ CSV

בשלב הראשון טענו את קובץ CSV המכיל את ה הודעות האפליקציה המדומות גלישה ברשת (פרוטוקול HTTP), נערכנו בינה מלאכותית על מנת ל כתוב את ה הודעות האפליקציה. הטבלה שהתקבלה במחברת הציגה את כל השורות בפורמט הנדרש, כולל השדות:

msg_id, app_protocol, src_app, dst_app, message, timestamp.

Step 1 — Load Your CSV (Input)

1. Place your CSV file (e.g., `group05_http_input.csv`) in the same folder as this notebook.
2. The CSV must contain the following columns: `msg_id`, `app_protocol`, `src_app`, `dst_app`, `message`, `timestamp`.
3. In the next cell, set `CSV_PATH` to your file name and run the cell.
4. Verify that the preview shows your rows correctly.

```
#TODO: Load CSV file with messages into pandas DataFrame,  
  
# Replace 'path_to_your_file.csv' with the actual file path  
import pandas as pd  
filename = r"C:\Users\Avital\Desktop\רנה\בשיחודים\למען תקשורת מחשבים\רנה\bshichodim\groupAvitalNaor_http_input.csv"  
messages_df = pd.read_csv(filename)  
messages_df
```

[4] ✓ 0.0s

Python

	msg_id	app_protocol	src_app	dst_app	message	timestamp
0	1	HTTP	client_browser	web_server	GET /index.html	0.010
1	2	HTTP	web_server	client_browser	200 OK (index.html)	0.018
2	3	HTTP	client_browser	web_server	GET /styles.css	0.026
3	4	HTTP	web_server	client_browser	200 OK (styles.css)	0.034
4	5	HTTP	client_browser	web_server	POST /login username=avital	0.045
5	6	HTTP	web_server	client_browser	302 Redirect /home	0.052
6	7	HTTP	client_browser	web_server	GET /home	0.060
7	8	HTTP	web_server	client_browser	200 OK (home page)	0.072
8	9	HTTP	client_browser	web_server	GET /images/logo.png	0.083
9	10	HTTP	web_server	client_browser	200 OK (logo.png)	0.095
10	11	HTTP	client_browser	web_server	GET /logout	0.110
11	12	HTTP	web_server	client_browser	200 OK Logged out	0.121

בנייה מבנה הפקטה (Encapsulation)

הקוד במחברת יוצר באופן ידני את שבבות TCP ו-IP:

- יצירת TCP Header הכלל source port, destination port, sequence number, flags
- יצירת IP Header הכלל כתובות מקור/יעד TTL, Identification ו-IP.
- חיבור הכותרות יחד עם ה-Payload ליצור פקטה מלאה.

לאחר יצירת הפקטה ניתן היה לראות Hexdump שמציג את המבנה הקסדיימלי של כל השדות.

```
# Preview packet structure
src_ip = '127.0.0.1'
dst_ip = '127.0.0.1'
src_port = random.randint(1024, 65535)
dst_port = 12345
payload = b'Hello Packet (preview)'
pkt_preview = build_ip_header(src_ip, dst_ip, 20 + len(payload)) + build_tcp_header(src_ip, dst_ip, src_port, dst_port, payload) + payload
hexdump(pkt_preview)

[14] ✓ 0.0s Python
...
0000 45 00 00 3e b8 37 00 00 40 06 c4 80 7f 00 00 01 E..>.7..@.....
0010 7f 00 00 01 1c 3a 30 39 79 43 2a 4f 00 00 00 00 .....:09yC*0.....
0020 50 02 ff ff ad f1 00 00 48 65 6c 6c 6f 20 50 61 P.....Hello Pa
0030 63 6b 65 74 20 28 70 72 65 76 69 65 77 29 kett (preview)
```

שליחת הפקטות בפועל ולבידה ב-Wireshark

לאחר הבנת הפקטות, ביצענו את שלב השידור והלבידה:

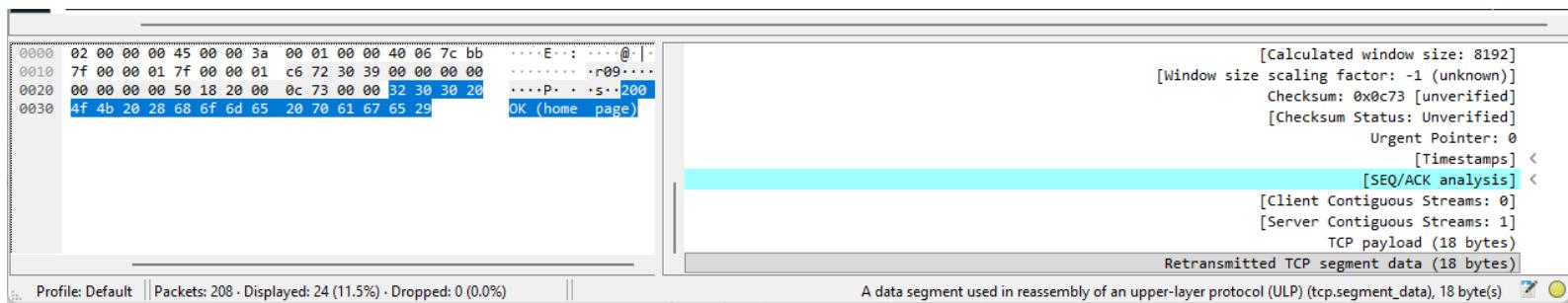
1. הפעילו את Wireshark והגדרכנו האזנה למשיק ה-Loopback (התעבורה הפנימית של המחשב).
2. הרצנו את לוallet השידור במחברת הקוד, שעברה על בל שורות ה-CSV- ושלחה אותן בזו אחר זו לרשת.

תוצאות הלבידה הראו בבירור את רצף הפקטות שנשלחו. ניתן לראות את כתובות המקור והיעד (127.0.0.1) ואת הפורטים שהוגדרו.

Info	Length	Protocol	Destination	Source	Time	.No
Seq=1 Ack=1 Win=18192 Len=15 [PSH, ACK] 12345 → 50802 59	TCP	127.0.0.1	127.0.0.1	127.0.0.1	157.645482 163	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=0 Len=0 [RST] 50802 → 12345 44	TCP	127.0.0.1	127.0.0.1	127.0.0.1	157.645753 164	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=18192 Len=19 [TCP Retransmission] 63	TCP	127.0.0.1	127.0.0.1	127.0.0.1	157.751686 165	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=15 [TCP Retransmission] 59	TCP	127.0.0.1	127.0.0.1	127.0.0.1	157.751796 166	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=18192 Len=19 [TCP Retransmission] 63	TCP	127.0.0.1	127.0.0.1	127.0.0.1	157.854010 167	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=18192 Len=19 [TCP Retransmission] 63	TCP	127.0.0.1	127.0.0.1	127.0.0.1	157.854062 168	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=18192 Len=19 [TCP Retransmission] 71	TCP	127.0.0.1	127.0.0.1	127.0.0.1	157.957858 169	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=18192 Len=27 [TCP Retransmission] 71	TCP	127.0.0.1	127.0.0.1	127.0.0.1	157.957954 170	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=18192 Len=0 [RST] 50802 → 12345 44	TCP	127.0.0.1	127.0.0.1	127.0.0.1	158.060093 171	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=18192 Len=18 [TCP Retransmission] 62	TCP	127.0.0.1	127.0.0.1	127.0.0.1	158.164309 173	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=0 Len=0 [RST] 50802 → 12345 44	TCP	127.0.0.1	127.0.0.1	127.0.0.1	158.164412 174	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=18192 Len=9 [TCP Retransmission] 53	TCP	127.0.0.1	127.0.0.1	127.0.0.1	158.268513 175	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=18192 Len=27 [TCP Retransmission] 62	TCP	127.0.0.1	127.0.0.1	127.0.0.1	158.268605 176	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=18192 Len=17 [TCP Retransmission] 62	TCP	127.0.0.1	127.0.0.1	127.0.0.1	158.370477 177	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=18192 Len=20 [TCP Retransmission] 64	TCP	127.0.0.1	127.0.0.1	127.0.0.1	158.370530 178	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=0 Len=0 [RST] 50802 → 12345 44	TCP	127.0.0.1	127.0.0.1	127.0.0.1	158.473829 179	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=18192 Len=17 [TCP Retransmission] 61	TCP	127.0.0.1	127.0.0.1	127.0.0.1	158.473900 180	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=18192 Len=11 [TCP Retransmission] 55	TCP	127.0.0.1	127.0.0.1	127.0.0.1	158.575724 182	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=18192 Len=17 [TCP Retransmission] 61	TCP	127.0.0.1	127.0.0.1	127.0.0.1	158.678646 184	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=18192 Len=17 [TCP Retransmission] 61	TCP	127.0.0.1	127.0.0.1	127.0.0.1	158.782897 185	
50802 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=0 Len=0 [RST] 50802 → 12345 44	TCP	127.0.0.1	127.0.0.1	127.0.0.1	158.782989 186	

ניתוח תוכן (Payload Analysis)

בחרנו באחת הפקודות שנלכדו על מנת לוודא שתוכן ההודעה עבר בצורה תקינה. בחרנו לדוגמא פקטה המבילה בקשה HTTP. בתחתית המסך בחלונית bytes ניתן לראות את הטקסט שנשלח "home page" (200 OK). זה מאשר שתהלייר האריזה הידני והשידור עבדו, ומהידע עבר משכבות האפליקציה ועד לשכבות הרשות בשלמותו.



קבצים מצורפים להגשה:

- CSV •
- Jupyter Notebook •
- pcap •