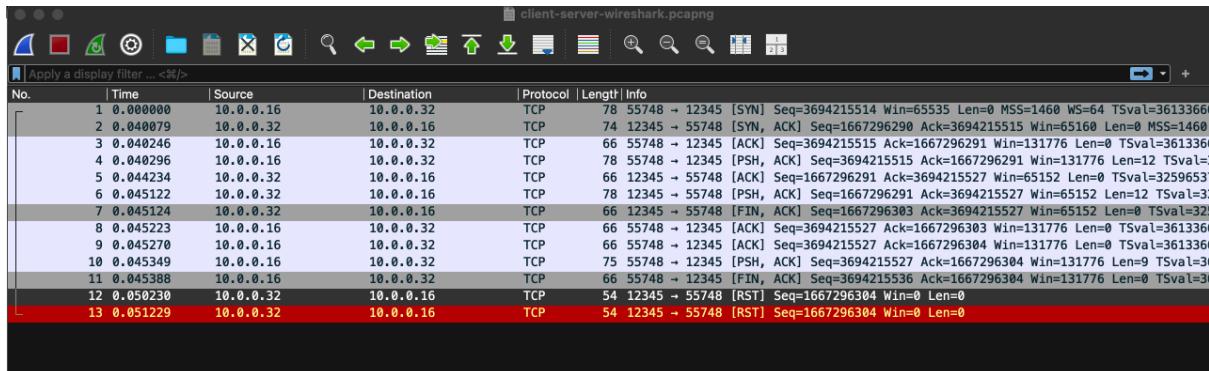


## רשתות תרגיל 2 חלק 1

מצורף כאן צילום מסך של תקשורת בין שני מחשבים שונים. הלקוח (10.0.0.16) מרים את הקובץ `.tcp_server.py` אשר פונה לשרת (10.0.0.32) הרים את הקובץ `.tcp_client.py`



### (Three-Way Handshake)

תהליך זה מתרחש בabiliaות 1 עד 3. זהו תהליך לחיצת היד המשולשת.

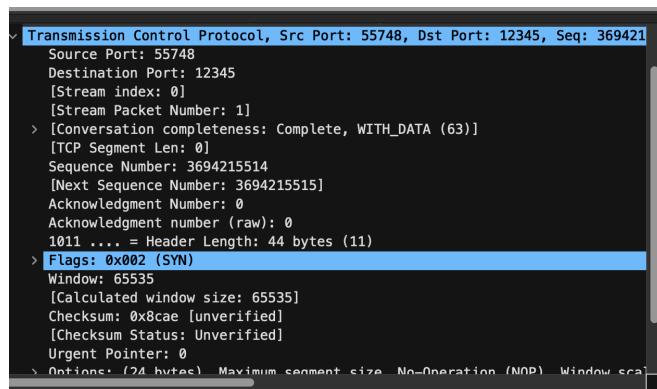
#### 1. שלב ראשון (Packet 1) - שליחת SYN:

- הלקוח (10.0.0.16) יוזם את החיבור ושולח הודעה SYN לשרת (10.0.0.32).
- מספרים סידוריים: המissor מגיריל מספר סידורי ההתחלתי.

$$\text{Seq} = 3694215514 \quad ■$$

$$\text{Ack} = 0 \quad ■$$

- משמעותו: אני רוצה לפתוח איתך שיחה, המissor הסידורי שלי מתחילה ב- "3694215514".



#### 2. שלב שני (Packet 2) - שליחת SYN, ACK:

- תיאור: השרת מקבל את הבקשה, ומחייב הודעה המכילה גם SYN (כדי לפתח ערזע משלו) וגם ACK (אישור הבקשה של הלקוח).

- מספרים סידוריים:

$$\text{Seq} = 1667296290 \quad ■$$

$$\text{Ack} = 3694215515 \quad ■$$

(היחסוב הוא: Seq של הלקוח + 1).

- משמעות: קיבלתי את ההודעה שלך והוספתי לה 1, ואני גם רוצה לפתח איתך שיחה במספר הסידורי שלו.

```

    > Transmission Control Protocol, Src Port: 12345, Dst Port: 55748, Seq: 166729
      Source Port: 12345
      Destination Port: 55748
      [Stream index: 0]
      [Stream Packet Number: 2]
    > [Conversation completeness: Complete, WITH_DATA (63)]
      [TCP Segment Len: 0]
      Sequence Number: 1667296290
      [Next Sequence Number: 1667296291]
      Acknowledgment Number: 3694215515
      1010 .... = Header Length: 40 bytes (10)
    > Flags: 0x012 (SYN, ACK)
      Window: 65160
      [Calculated window size: 65160]
      Checksum: 0x2eaf [unverified]
      [Checksum Status: Unverified]
  
```

- שלב שלישי (Packet 3) - שליחת ACK:  
הלקוח מאשר את קבלת ה-SYN של השרת ע"י שליחת ACK. החיבורicut הוקם.  
מספרים סידוריים:
- Seq (התקדם ב-1 לעומת חבילת 1, כי דגל ה-SYN צורך מספר סידורי אחד).
- Ack (הчисלוב הוא: ה-Seq של השרת + 1).

```

> Frame 3: Packet, 66 bytes on wire (528 bits), 66 bytes captured (528 bits) o
> Ethernet II, Src: de:6e:31:1c:f5:de (de:6e:31:1c:f5:de), Dst: Intel_46:ba:9a
> Internet Protocol Version 4, Src: 10.0.0.16, Dst: 10.0.0.32
> Transmission Control Protocol, Src Port: 55748, Dst Port: 12345, Seq: 369421
  Source Port: 55748
  Destination Port: 12345
  [Stream index: 0]
  [Stream Packet Number: 3]
> [Conversation completeness: Complete, WITH_DATA (63)]
  [TCP Segment Len: 0]
  Sequence Number: 3694215515
  [Next Sequence Number: 3694215515]
  Acknowledgment Number: 1667296291
  1000 .... = Header Length: 32 bytes (8)
> Flags: 0x010 (ACK)
  Window: 2059
  [Calculated window size: 131776]
  [Window size scaling factor: 64]
  Checksum: 0x53d0 [unverified]
  
```

## העברת נתונים וניתוח המספרים

### 1. שליחת מידע (Packet 4):

- תיאור: הלקוח (10.0.0.16) שולח מידע לשרת. ניתן לראות את הדגל שմבוקש להעביר את המידע לאפליקציה מיד.
- גודל המידע:  $12 = \text{Len} (\text{Length})$  או  $\text{Info}$ .
- מספר סידורי:  $\text{Seq} = 3694215515$

The screenshot shows a TCP packet captured by Wireshark. The details pane displays the following information:

- Sequence Number: 3694215515
- [Next Sequence Number: 3694215527]
- Acknowledgment Number: 1667296291
- 1000 .... = Header Length: 32 bytes (8)
- > Flags: 0x018 (PSH, ACK)
- Window: 2059
- [Calculated window size: 131776]
- [Window size scaling factor: 64]
- Checksum: 0x4960 [unverified]
- [Checksum Status: Unverified]
- Urgent Pointer: 0
- > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
- > [Timestamps]
- > [SEQ/ACK analysis]
- [Client Contiguous Streams: 1]
- [Server Contiguous Streams: 1]
- TCP payload (12 bytes)

The bytes pane shows the raw hex and ASCII data of the packet, starting with:

0000	f4	06	69	46	ba	9a	de	6e	31	1c	f5	de	08	00	45	00	..iF..n1.....E..
0010	00	40	00	00	40	00	40	06	26	89	0a	00	00	10	0a	00	@...@. @.....
0020	00	20	d9	c4	30	39	dc	31	41	5b	63	60	ec	23	80	18	..09.1 A[c`.#..
0030	08	0b	49	60	00	00	01	01	08	0a	d7	5f	97	5d	c2	4a	.I.....[.J..
0040	5e	9b	59	6f	61	76	20	45	6c	68	61	64	61	64			^Yoav E lhadad

### 2. אישור קבלת מידע (Packet 5):

- תיאור: השרת (10.0.0.32) מאשר את קבלת המידע.
- חישוב ה-Ack: השרת מבצע חישוב פשוט: המספר הסידורי הקודם שקיבל + גודל המידע שקיבל.
- החישוב במספרים:
  - $3694215515$  (ה-Seq של חבילה 4)
  - $+ 12$  (ה-Len של חבילה 4)
  - $= 3694215527$
- לכן חבילה 5 אנו רואים  $\text{Ack} = 3694215527$ . זהו האישור שהמידע התקבל בשלםותיו והשרת מצפה לבית הבא במספר זה.

```

!Stream Packet Number: 5]
> [Conversation completeness: Complete, WITH_DATA (63)]
[TCP Segment Len: 0]
Sequence Number: 1667296291
[Next Sequence Number: 1667296291]
Acknowledgment Number: 3694215527
1000 .... = Header Length: 32 bytes (8)
> Flags: 0x010 (ACK)
Window: 509
[Calculated window size: 65152]
[Window size scaling factor: 128]
Checksum: 0x59cd [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
> Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
> [Timestamps]
> [SEQ/ACK analysis]
[Client Contiguous Streams: 1]
[Server Contiguous Streams: 1]

```

client-server-wireshark.pcapng

### 3. תשובה השרת (Packet 6)

- תיאור: השרת (10.0.0.32) שולח חזרה ללקוח את המחרוזת "YOAV ELHADAD".
- ה Seq הוא 1667296291 (זהה ל-ACK הקודם, כי השרת עדין לא שלח מידע, רק אישר).
- ה Ack הוא 3694215527 (השרת עדין מצפה לבית ה-27, כולם מאשר שקיבל את השם).
- האורך הוא 12 בתים

```

* Transmission Control Protocol, Src Port: 12345, Dst Port: 55748, Seq: 1, Ack: 13, Len: 12
Source Port: 12345
Destination Port: 55748
[Stream index: 0]
[Stream Packet Number: 6]
> [Conversation completeness: Complete, WITH_DATA (63)]
[TCP Segment Len: 12]
Sequence Number: 1 (relative sequence number)
Sequence Number (raw): 1667296291
[Next Sequence Number: 13 (relative sequence number)]
Acknowledgment Number: 13 (relative ack number)
Acknowledgment number (raw): 3694215527
1000 .... = Header Length: 32 bytes (8)
> Flags: 0x018 (PSH, ACK)
Window: 509
[Calculated window size: 65152]
[Window size scaling factor: 128]
Checksum: 0xcffd [unverified]
[Checksum Status: Unverified]
Urgent Pointer: 0
> Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
> [Timestamps]
> [SEQ/ACK analysis]
[Client Contiguous Streams: 1]
[Server Contiguous Streams: 1]
TCP payload (12 bytes)

```

### 4. שליחת תעודת זהות (Packet 8)

- תיאור: הלקוח (10.0.0.16) שולח את ההודעה השנייה המכילה את תעודת הזהות: "315853793".

הчисוב במספרים:

- Seq=3694215527, המספר הסידורי הקודם של הלקוח בחבילה 4 היה 3694215515, שלחנו 12 בתים ולקס  $3694215515 + 12 = 3694215527$
- Ack=1667296303, הלקוח קיבל מהשרת את תשובה Echo סובב ה-6 שהייתה באורך 12 בתים והתחילה ב 1667296291 אך הלקוח מאשר קבלת ומצבה לו:  $1667296303 + 12 = 1667296291$ .

## האורך הוא 9 בתים ■

```
▼ Transmission Control Protocol, Src Port: 55748, Dst Port: 12345, Seq: 13, Ack: 13, Len: 0
  Source Port: 55748
  Destination Port: 12345
  [Stream index: 0]
  [Stream Packet Number: 8]
  > [Conversation completeness: Complete, WITH_DATA (63)]
    [TCP Segment Len: 0]
    Sequence Number: 13      (relative sequence number)
    Sequence Number (raw): 3694215527
    [Next Sequence Number: 13      (relative sequence number)]
    Acknowledgment Number: 13      (relative ack number)
    Acknowledgment number (raw): 1667296303
    1000 .... = Header Length: 32 bytes (8)
  > Flags: 0x010 (ACK)
    Window: 2059
    [Calculated window size: 131776]
    [Window size scaling factor: 64]
    Checksum: 0x53af [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
  > Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
  > [Timestamps]
  > [SEQ/ACK analysis]
    [Client Contiguous Streams: 1]
    [Server Contiguous Streams: 1]
```

## חלק 3: סיום החיבור

1. ניסיון סיום (Packet 7):
  - הרשות (10.0.0.32) שולח FIN, ACK (10.0.0.32).
  - משמעות: הרשות מודיע "סימתי לשלוח מידע, אני רוצה לסגור את החיבור מציד".
2. סגירה מיידית / תקלה (Packet 12-13):
  - רואים הודעות של RST Reset (Packet 12-13). הסבר: במקום תחיליך סגירה מסודר (שבו הצד השני שולח FIN משול ו-ACK), נשלחה פקודה "Reset". זה קורה בדרך כלל כאשר צד אחד סוגר את התוכנה בכוח, או שהפורט כבר לא מאזין, והוא "שובר" את החיבור מיד במקום לסגור אותו בעדינות.

Frame 12: Packet, 54 bytes on wire (432 bits), 54 bytes captured (432 bits)  
Ethernet II, Src: Intel\_A6:ba:9a (f4:06:69:46:ba:9a), Dst: de:6e:31:1c:f5:de  
Internet Protocol Version 4, Src: 10.0.0.32, Dst: 10.0.0.16  
Transmission Control Protocol, Src Port: 12345, Dst Port: 55748, Seq: 1667296304, Ack: 1667296304 Win=0 Len=0

Index	Source	Destination	Protocol	Port	Sequence	Acknowledgment	Flags	Length
11	10.0.0.32	10.0.0.16	TCP	12345 → 55748	1667296304	1667296304	[FIN, ACK]	54
12	0.0.0.32	10.0.0.16	TCP	55748 → 12345	1667296304	1667296304	[RST]	0
13	0.0.0.32	10.0.0.16	TCP	55748 → 12345	1667296304	1667296304	[RST]	0

```
Source Port: 12345
Destination Port: 55748
[Stream index: 0]
[Stream Packet Number: 12]
> [Conversation completeness: Complete, WITH_DATA (63)]
[TCP Segment Len: 0]
Sequence Number: 1667296304
[Next Sequence Number: 1667296304]
Acknowledgment Number: 0
Acknowledgment number (raw): 0
0101 .... = Header Length: 20 bytes (5)
> Flags: 0x004 (RST)
Window: 0
[Calculated window size: 0]
[Window size scaling factor: 128]
```

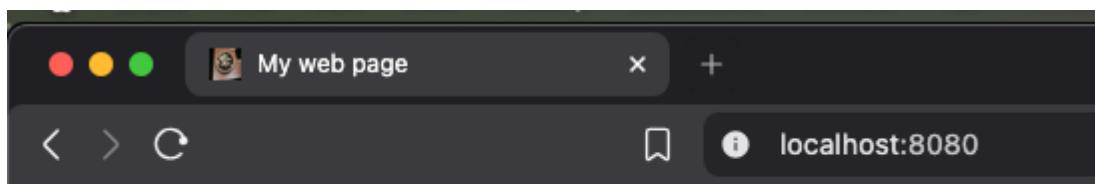
תומך:  
הקוד של הלקוח:

The screenshot shows a code editor interface with the following details:

- File Explorer:** Shows a project structure under "ASS2" named "part1" containing "client-server-wireshark.pcapng", "Dockerfile", "tcp\_client.py", and "tcp\_server.py".
- Code Editor:** Displays the contents of "tcp\_client.py".

```
part1 > 🐍 tcp_client.py > ...
1 import socket
2 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
3 s.connect(('10.0.0.32', 12345))
4 s.send(b'Yoav Elhadad')
5 data = s.recv(100)
6 print("Server sent: ", data)
7 s.send(b'315053793')
8 data = s.recv(100)
9 print("Server sent: ", data)
10 s.close()
```
- Terminal:** Shows the output of running the client twice.

```
● part1|> python3 ./tcp_client.py
Server sent: b'YOAV ELHADAD'
Server sent: b''
● part1|> python3 ./tcp_client.py
Server sent: b'YOAV ELHADAD'
Server sent: b''
○ part1|>
```



hello