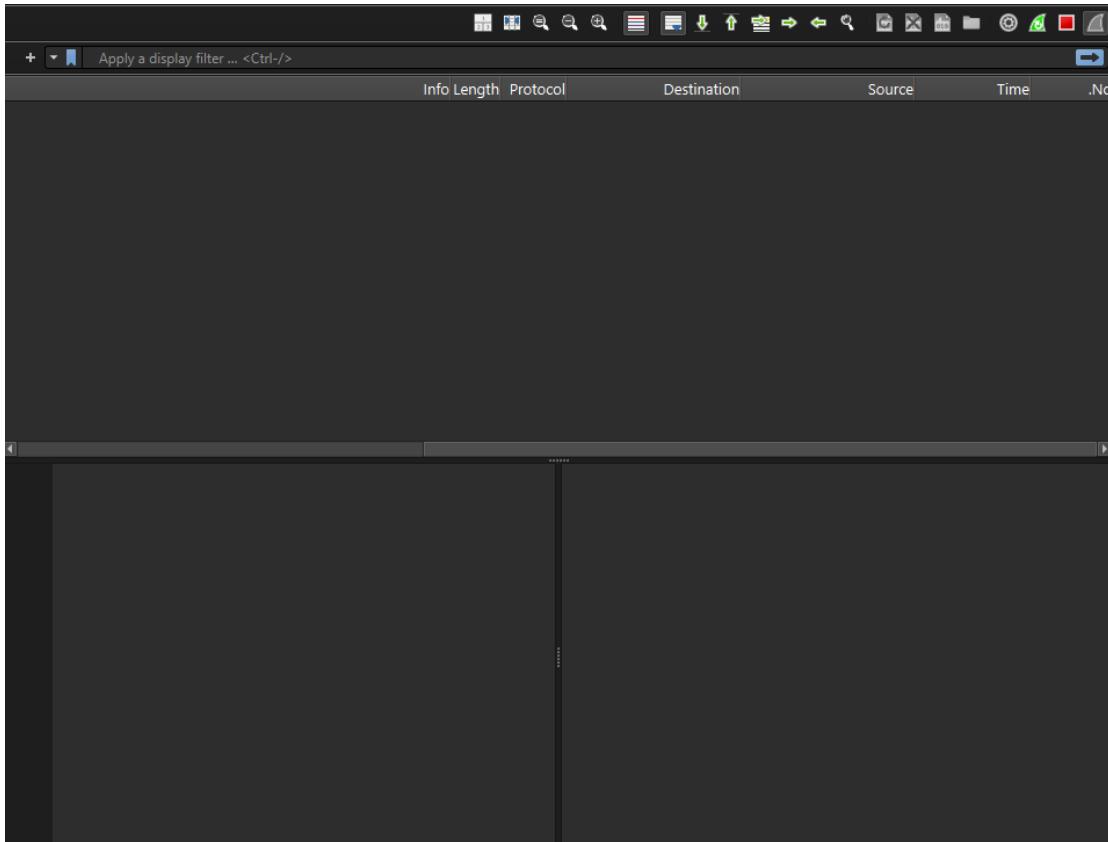


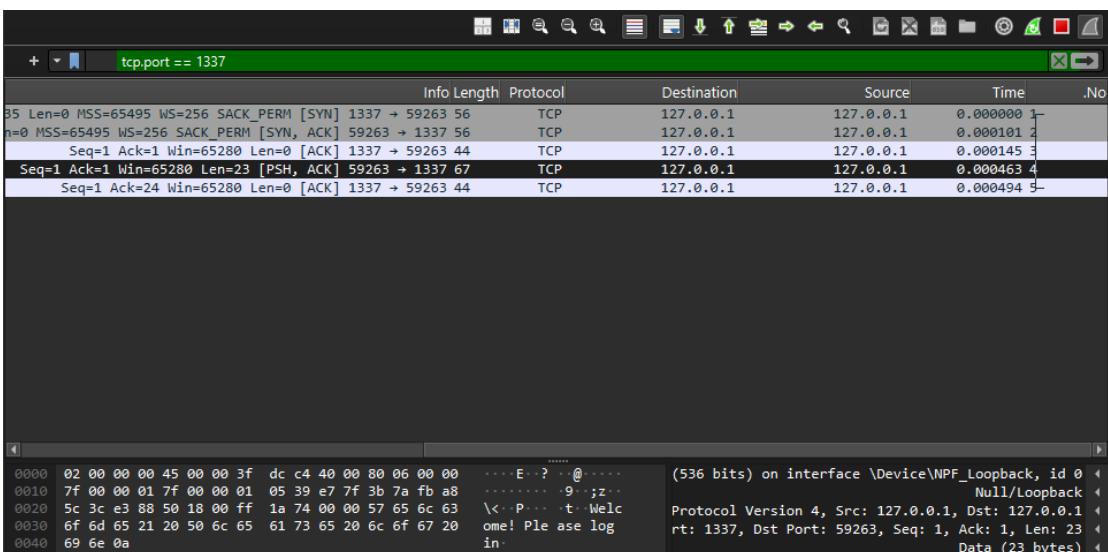
עבודה עם wireshark

תהליך ההתחברות –

1. הרצת הסקריפט של השרת – לא עוברות פקודות.



2. הרצת הסקריפט של הלקוח – יוצר חיבור TCP עם השרת (SYN, SYN-ACK, ACK) והשרת שולח הודעה "Welcome! Please log in" (ניתן לראות בתחתית התמונה) ולקוח מחזיר ACK.



3. הלקוח מזין שם משתמש (ניתן לראות בתמונה), השרת מחזיר ACK ולאחר מכן הלקוח שולח גם סיסמה, שם עלייה השרת מחזיר ACK.

No.	Date	Time	Source	Destination	Protocol	Info
35	Len=0 MSS=65495 WS=256 SACK_PERM	[SYN]	1337 → 59263	56	TCP	127.0.0.1
n=0	MSS=65495 WS=256 SACK_PERM	[SYN, ACK]	59263 → 1337	56	TCP	127.0.0.1
Seq=1	Ack=1 Win=65280 Len=0 [ACK]	1337 → 59263	44	TCP	127.0.0.1	0.000145 3
Seq=1	Ack=1 Win=65280 Len=23 [PSH, ACK]	59263 → 1337	67	TCP	127.0.0.1	0.000463 4
Seq=1	Ack=24 Win=65280 Len=0 [ACK]	1337 → 59263	44	TCP	127.0.0.1	0.000494 5
Seq=1	Ack=24 Win=65280 Len=11 [PSH, ACK]	1337 → 59263	55	TCP	127.0.0.1	393.956966 358
Seq=24	Ack=12 Win=65280 Len=0 [ACK]	59263 → 1337	44	TCP	127.0.0.1	393.957020 359
Seq=12	Ack=24 Win=65280 Len=21 [PSH, ACK]	1337 → 59263	65	TCP	127.0.0.1	393.957051 360
Seq=24	Ack=33 Win=65280 Len=0 [ACK]	59263 → 1337	44	TCP	127.0.0.1	393.957076 361
Seq=24	Ack=33 Win=65280 Len=25 [PSH, ACK]	59263 → 1337	69	TCP	127.0.0.1	393.957197 362
Seq=33	Ack=49 Win=65280 Len=0 [ACK]	1337 → 59263	44	TCP	127.0.0.1	393.957239 363

4. אחרי שהזנו פרטים נכונים, השרת שולח ללקוח הודעה התחברות (ניתן לראות אותה בתחלת התמונה) והלקוח מחזיר ACK.

No.	Date	Time	Source	Destination	Protocol	Info
35	Len=0 MSS=65495 WS=256 SACK_PERM	[SYN]	1337 → 59263	56	TCP	127.0.0.1
n=0	MSS=65495 WS=256 SACK_PERM	[SYN, ACK]	59263 → 1337	56	TCP	127.0.0.1
Seq=1	Ack=1 Win=65280 Len=0 [ACK]	1337 → 59263	44	TCP	127.0.0.1	0.000145 3
Seq=1	Ack=1 Win=65280 Len=23 [PSH, ACK]	1337 → 59263	67	TCP	127.0.0.1	0.000463 4
Seq=1	Ack=24 Win=65280 Len=0 [ACK]	59263 → 1337	44	TCP	127.0.0.1	0.000494 5
Seq=1	Ack=24 Win=65280 Len=11 [PSH, ACK]	1337 → 59263	55	TCP	127.0.0.1	393.956966 358
Seq=12	Ack=24 Win=65280 Len=21 [PSH, ACK]	1337 → 59263	65	TCP	127.0.0.1	393.957020 359
Seq=24	Ack=33 Win=65280 Len=0 [ACK]	59263 → 1337	44	TCP	127.0.0.1	393.957051 360
Seq=24	Ack=33 Win=65280 Len=25 [PSH, ACK]	59263 → 1337	69	TCP	127.0.0.1	393.957076 361
Seq=33	Ack=49 Win=65280 Len=0 [ACK]	1337 → 59263	44	TCP	127.0.0.1	393.957197 362
Seq=33	Ack=49 Win=65280 Len=0 [ACK]	1337 → 59263	44	TCP	127.0.0.1	393.957239 363

ברצף זה ניתן לראות איך פרוטוקול ההתחברות שיצרנו פועל – עם הרכבת הסקריפט של הלקוח, נשלחת בקשה להתחברות לשרת, שמחזיר את הודעה הנדרשת, ואחריה הלקוח שולח ברצף 2 פקודות ופרודות שמכילות את שם המשתמש והסיסמה (לפי הסדר הזה). אם המשתמש קיים בקובץ המשתמשים החוקים, הוא מקבל הודעה מתאימה מהשרת. אחרת, הוא מקבל הודעה נוספת נוספת להתחבר.

שליחת שאילתת מהלוקות –

tcp.port == 1337

No.	Date	Info	Length	Protocol	Destination	Source	Time	.No.
35	Len=0 MSS=65495 WS=256 SACK_PERM	[SYN]	1337 → 57372 56	TCP	127.0.0.1	127.0.0.1	0.000000 1	
n=0	MSS=65495 WS=256 SACK_PERM	[SYN, ACK]	57372 → 1337 56	TCP	127.0.0.1	127.0.0.1	0.000084 2	
Seq=1 Ack=1 Win=65280 Len=0 [ACK]	1337 → 57372 44	TCP	127.0.0.1	127.0.0.1	0.000114 3			
Seq=1 Ack=1 Win=65280 Len=23 [PSH, ACK]	57372 → 1337 67	TCP	127.0.0.1	127.0.0.1	0.000301 4			
Seq=1 Ack=24 Win=65280 Len=0 [ACK]	1337 → 57372 44	TCP	127.0.0.1	127.0.0.1	0.000330 5			
Seq=1 Ack=24 Win=65280 Len=11 [PSH, ACK]	1337 → 57372 55	TCP	127.0.0.1	127.0.0.1	5.833409 6			
Seq=24 Ack=12 Win=65280 Len=0 [ACK]	57372 → 1337 44	TCP	127.0.0.1	127.0.0.1	5.833452 7			
Seq=12 Ack=24 Win=65280 Len=21 [PSH, ACK]	1337 → 57372 65	TCP	127.0.0.1	127.0.0.1	5.833477 8			
Seq=24 Ack=33 Win=65280 Len=0 [ACK]	57372 → 1337 44	TCP	127.0.0.1	127.0.0.1	5.833492 9			
Seq=24 Ack=33 Win=65280 Len=25 [PSH, ACK]	57372 → 1337 69	TCP	127.0.0.1	127.0.0.1	5.833581 10			
Seq=33 Ack=49 Win=65280 Len=0 [ACK]	1337 → 57372 44	TCP	127.0.0.1	127.0.0.1	5.833618 11			
65280 Len=11[Malformed Packet] [PSH, ACK]	1337 → 57372 55	TCP	127.0.0.1	127.0.0.1	42.570029 18			
Seq=49 Ack=44 Win=65280 Len=0 [ACK]	57372 → 1337 44	TCP	127.0.0.1	127.0.0.1	42.570069 19			
Seq=49 Ack=44 Win=65280 Len=15 [PSH, ACK]	57372 → 1337 59	TCP	127.0.0.1	127.0.0.1	42.570245 20			
Seq=44 Ack=64 Win=65280 Len=0 [ACK]	1337 → 57372 44	TCP	127.0.0.1	127.0.0.1	42.570277 21			

0000 02 00 00 00 45 00 00 37 e1 d1 40 00 80 06 00 00E..7 ..@..... (472 bits) on interface \Device\NPF_Loopback, id 0 ↓
0010 7f 00 00 01 7f 00 00 01 05 39 e0 1c 70 c2 b6 3e9 ..p..> Null/Loopback ↓
0020 7b 1c 7e 74 50 18 00 ff 14 04 00 00 74 68 65 20 { ~tP.....the Family: IP (2)
0030 6c 63 6d 20 69 73 3a 20 36 30 0a lcm is: 60 Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 ↓
: 1337, Dst Port: 57372, Seq: 49, Ack: 44, Len: 15 ↓
Data (15 bytes) ↓

בשורה 18 הלקוח שולח לשרת את השורה "lcm: 20 15".
בשורה 19, השרת שולח ACK ומבצע את החישוב.
בשורה 20 (מסומנת), השרת מחזיר את התשובה (60).
בשורה 21, הלקוח מחזיר ACK.

פרוטוקול TCP-קריטי בקרה זהה, כי השרת מעבד את המידע שנשלח מהלקוח
ומבצע עלייו חישובים. אם חלק מהמידע אובד בדרך (כמו שעולה לקרות ב프וטוקול
UDP), עלולה לחזור תשובה שגوية ללקוח. פרוטוקול TCP מבטיח שהמידע שנשלח מכל
צד יגיע במלואו ולפי הסדר ליעד.