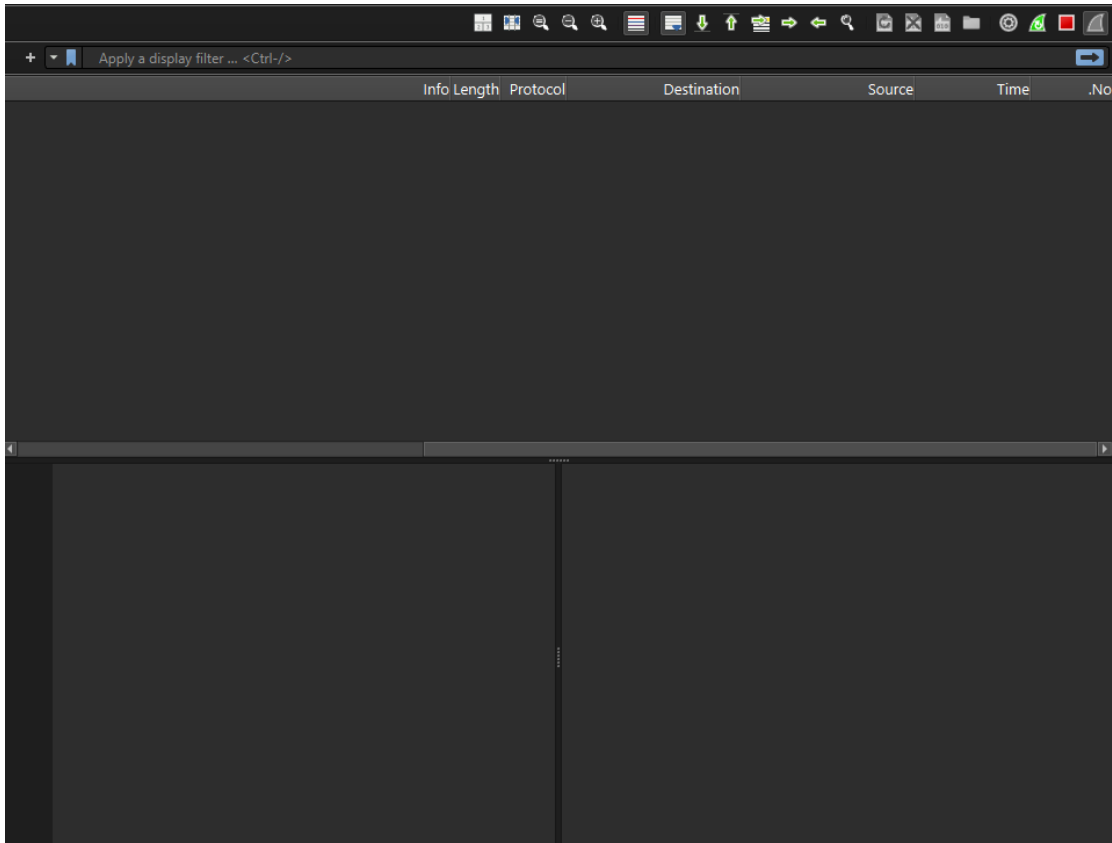


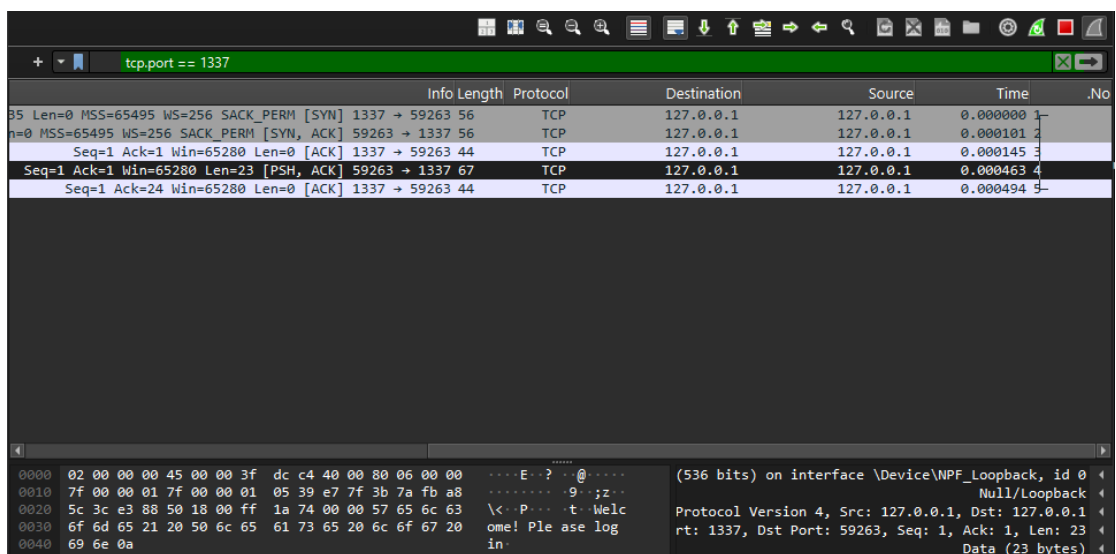
עבודה עם wireshark

תהליך ההתחברות –

1. הרצת הסקריפט של השרת – לא עוברות פקטות.



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



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

tcp.port == 1337									
No.	Time	Source	Destination	Protocol	Length	Info			
1	0.000000	127.0.0.1	127.0.0.1	TCP	56	1337 → 59263 56 MSS=65495 WS=256 SACK_PERM [SYN]			
2	0.000101	127.0.0.1	127.0.0.1	TCP	56	59263 → 1337 56 MSS=65495 WS=256 SACK_PERM [SYN, ACK]			
3	0.000145	127.0.0.1	127.0.0.1	TCP	44	Seq=1 Ack=1 Win=65280 Len=0 [ACK]			
4	0.000463	127.0.0.1	127.0.0.1	TCP	67	Seq=1 Ack=1 Win=65280 Len=23 [PSH, ACK]			
5	0.000494	127.0.0.1	127.0.0.1	TCP	44	Seq=1 Ack=24 Win=65280 Len=0 [ACK]			
558	393.956966	127.0.0.1	127.0.0.1	TCP	55	Seq=1 Ack=24 Win=65280 Len=11 [PSH, ACK]			
559	393.957020	127.0.0.1	127.0.0.1	TCP	44	Seq=24 Ack=12 Win=65280 Len=0 [ACK]			
560	393.957051	127.0.0.1	127.0.0.1	TCP	65	Seq=12 Ack=24 Win=65280 Len=21 [PSH, ACK]			
561	393.957076	127.0.0.1	127.0.0.1	TCP	44	Seq=24 Ack=33 Win=65280 Len=0 [ACK]			
562	393.957197	127.0.0.1	127.0.0.1	TCP	69	Seq=24 Ack=33 Win=65280 Len=25 [PSH, ACK]			
563	393.957239	127.0.0.1	127.0.0.1	TCP	44	Seq=33 Ack=49 Win=65280 Len=0 [ACK]			

0000	02 00 00 00 45 00 00 33	dd b4 40 00 80 06 00 00E-3-@-....	(440 bits) on interface \Device\NPF_{Loopback, id 0
0010	7f 00 00 01 7f 00 00 01	e7 7f 05 39 5c 3c e3 889\<..	Null/Loopback
0020	3b 7a fb bf 50 18 00 ff	71 26 00 00 55 73 65 72	;z-P-...q&-User	Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
0030	3a 20 6d 61 6f 7a 0a		: maoz	t: 59263, Dst Port: 1337, Seq: 1, Ack: 24, Len: 11
				Data (11 bytes)

4. אחרי שהוזנו פרטים נכונים, השרת שולח ללקוח הודעת התחברות (ניתן לראות אותה בתחתית התמונה) והלקוח מחזיר ACK.

tcp.port == 1337										
No.	Time	Source	Destination	Protocol	Length	Info				
1	0.000000	127.0.0.1	127.0.0.1	TCP	56	1337 → 59263 56 MSS=65495 WS=256 SACK_PERM [SYN]				
2	0.000101	127.0.0.1	127.0.0.1	TCP	56	59263 → 1337 56 MSS=65495 WS=256 SACK_PERM [SYN, ACK]				
3	0.000145	127.0.0.1	127.0.0.1	TCP	44	Seq=1 Ack=1 Win=65280 Len=0 [ACK]				
4	0.000463	127.0.0.1	127.0.0.1	TCP	67	Seq=1 Ack=1 Win=65280 Len=23 [PSH, ACK]				
5	0.000494	127.0.0.1	127.0.0.1	TCP	44	Seq=1 Ack=24 Win=65280 Len=0 [ACK]				
558	393.956966	127.0.0.1	127.0.0.1	TCP	55	Seq=1 Ack=24 Win=65280 Len=11 [PSH, ACK]				
559	393.957020	127.0.0.1	127.0.0.1	TCP	44	Seq=24 Ack=12 Win=65280 Len=0 [ACK]				
560	393.957051	127.0.0.1	127.0.0.1	TCP	65	Seq=12 Ack=24 Win=65280 Len=21 [PSH, ACK]				
561	393.957076	127.0.0.1	127.0.0.1	TCP	44	Seq=24 Ack=33 Win=65280 Len=0 [ACK]				
562	393.957197	127.0.0.1	127.0.0.1	TCP	69	Seq=24 Ack=33 Win=65280 Len=25 [PSH, ACK]				
563	393.957239	127.0.0.1	127.0.0.1	TCP	44	Seq=33 Ack=49 Win=65280 Len=0 [ACK]				

000002 00 00 00 45 00 00 41dd b8 40 00 80 06 00 0000107f 00 00 01 7f 00 00 0105 39 e7 7f 3b 7a fb bf00205c 3c e3 a8 50 18 00 ff80 3a 00 00 48 69 20 6d003061 6f 7a 2c 20 67 6f 6f64 20 74 6f 20 73 65 65004020 79 6f 75 0a

.....E-A-@-.....9-;z-..<-P-...:Hi maoz, goo d to see you

(552 bits) on interface \Device\NPF_{Loopback, id 0}Null/LoopbackProtocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1: 1337, Dst Port: 59263, Seq: 24, Ack: 33, Len: 25Data (25 bytes)

ברצף זה ניתן לראות איך פרוטוקול ההתחברות שיצרנו פועל – עם הרצת הסקריפט של הלקוח, נשלחת בקשת התחברות לשרת, שמחזיר את ההודעה הנדרשת, ואחריה הלקוח שולח ברצף 2 פקטות נפרדות שמכילות את שם המשתמש והסיסמה (לפי הסדר הזה). אם המשתמש קיים בקובץ המשתמשים החוקיים, הוא מקבל הודעה מתאימה מהשרת. אחרת, הוא מקבל הודעות נוספות להתחבר.

שליחת שאלתה מהלקוח –

tcp.port == 1337									
No.	Time	Source	Destination	Protocol	Length	Info			
1	0.000000	127.0.0.1	127.0.0.1	TCP	56	1337 → 57372 [SYN, SACK_PERM, MSS=65495, WS=256]			
2	0.000084	127.0.0.1	127.0.0.1	TCP	56	57372 → 1337 [ACK, Seq=1, Ack=1, Win=65280, Len=0]			
3	0.000114	127.0.0.1	127.0.0.1	TCP	44	1337 → 57372 [ACK, Seq=1, Ack=1, Win=65280, Len=23]			
4	0.000301	127.0.0.1	127.0.0.1	TCP	67	57372 → 1337 [PSH, ACK, Seq=1, Ack=24, Win=65280, Len=0]			
5	0.000330	127.0.0.1	127.0.0.1	TCP	44	1337 → 57372 [ACK, Seq=1, Ack=24, Win=65280, Len=0]			
6	5.833409	127.0.0.1	127.0.0.1	TCP	55	1337 → 57372 [PSH, ACK, Seq=24, Ack=12, Win=65280, Len=11]			
7	5.833452	127.0.0.1	127.0.0.1	TCP	44	57372 → 1337 [ACK, Seq=24, Ack=12, Win=65280, Len=0]			
8	5.833477	127.0.0.1	127.0.0.1	TCP	65	1337 → 57372 [PSH, ACK, Seq=12, Ack=24, Win=65280, Len=21]			
9	5.833492	127.0.0.1	127.0.0.1	TCP	44	57372 → 1337 [ACK, Seq=24, Ack=33, Win=65280, Len=0]			
10	5.833581	127.0.0.1	127.0.0.1	TCP	69	57372 → 1337 [PSH, ACK, Seq=24, Ack=33, Win=65280, Len=25]			
11	5.833618	127.0.0.1	127.0.0.1	TCP	44	1337 → 57372 [ACK, Seq=33, Ack=49, Win=65280, Len=0]			
18	42.570029	127.0.0.1	127.0.0.1	TCP	55	1337 → 57372 [PSH, ACK, Seq=5280, Len=11] [Malformed Packet]			
19	42.570069	127.0.0.1	127.0.0.1	TCP	44	57372 → 1337 [ACK, Seq=49, Ack=44, Win=65280, Len=0]			
20	42.570245	127.0.0.1	127.0.0.1	TCP	59	57372 → 1337 [PSH, ACK, Seq=49, Ack=44, Win=65280, Len=15]			
21	42.570277	127.0.0.1	127.0.0.1	TCP	44	1337 → 57372 [ACK, Seq=44, Ack=64, Win=65280, Len=0]			

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 ה-TCP קריטי במקרה הזה, כי השרת מעבד את המידע שנשלח מהלקוח ומבצע עליו חישובים. אם חלק מהמידע אובד בדרך (כמו שעלול לקרות בפרוטוקול UDP), עלולה לחזור תשובה שגויה ללקוח. פרוטוקול TCP מבטיח שהמידע שנשלח מכל צד יגיע במלואו ולפי הסדר ליעד.