

יעל אביעוז 207237421
שי גונדרסן 204753123

מבוא לרשתות תקשורת – תרגיל 2

חלק א

בתמונה הבאה ניתן לראות את התעבורה שהתקבלה לאחר הרצת קוד השרת והלקוח בפרוטוקול TCP. השתמשו במסנן "TCP" על מנת להסתכל רק על התעבורה הרלוונטית לנו.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000...	10.0.2.5	10.0.2.8	TCP	74	39364 → 12345 [SYN] Seq=0 Win=64240 Len=0 MSS...
2	0.000020...	10.0.2.8	10.0.2.5	TCP	74	12345 → 39364 [SYN, ACK] Seq=0 Ack=1 Win=6516...
3	0.000213...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345 [ACK] Seq=1 Ack=1 Win=64256 Len...
4	0.000429...	10.0.2.5	10.0.2.8	TCP	95	39364 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=6425...
5	0.000438...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364 [ACK] Seq=1 Ack=30 Win=65152 Le...
6	0.000507...	10.0.2.8	10.0.2.5	TCP	95	12345 → 39364 [PSH, ACK] Seq=1 Ack=30 Win=651...
7	0.000713...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345 [ACK] Seq=30 Ack=30 Win=64256 L...
8	0.000713...	10.0.2.5	10.0.2.8	TCP	88	39364 → 12345 [PSH, ACK] Seq=30 Ack=30 Win=64...
9	0.000722...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364 [ACK] Seq=30 Ack=52 Win=65152 L...
10	0.000768...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364 [FIN, ACK] Seq=30 Ack=52 Win=65...
11	0.000898...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345 [FIN, ACK] Seq=52 Ack=30 Win=64...
12	0.000903...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364 [ACK] Seq=31 Ack=53 Win=65152 L...
13	0.001011...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345 [ACK] Seq=53 Ack=31 Win=64256 L...

בתמונות הבאות ננתח ונסביר מהו המידע המוצג.

תמונה מספר 1

FileEditViewGoCaptureAnalyzeStatisticsToolsHelp

tcp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000...	10.0.2.5	10.0.2.8	TCP	74	39364 → 12345 [SYN] Seq=0 Win=64240 Len=0 MSS...
2	0.000020...	10.0.2.8	10.0.2.5	TCP	74	12345 → 39364 [SYN, ACK] Seq=0 Ack=1 Win=6516...
3	0.000213...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345 [ACK] Seq=1 Ack=1 Win=64256 Len...
4	0.000429...	10.0.2.5	10.0.2.8	TCP	95	39364 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=6425...
5	0.000438...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364 [ACK] Seq=1 Ack=30 Win=65152 Le...
6	0.000507...	10.0.2.8	10.0.2.5	TCP	95	12345 → 39364 [PSH, ACK] Seq=1 Ack=30 Win=651...

- Transmission Control Protocol, Src Port: 39364, Dst Port: 12345, Seq: 0, Len: 0

Source Port: 39364

Destination Port: 12345

[Stream index: 0]

[TCP Segment Len: 0]

Sequence number: 0 (relative sequence number)

Sequence number (raw): 3463997819

[Next sequence number: 1 (relative sequence number)]

Acknowledgment number: 0

Acknowledgment number (raw): 0

1010 = Header Length: 40 bytes (10)

- Flags: 0x002 (SYN)

000. = Reserved: Not set

0000	08 00 27	24 4c de 08 00	27 d2 8d 71 08 00 45 00	..\$.L... '...q...E..
0010	00 3c 52 b3 40 00 40 06	cf fc 0a 00 02 05 0a 00	..<R:@. @.	
0020	02 08 99 c4 30 39 ce 78	69 7b 00 00 00 00 a0 02	...09-x i{	
0030	fa f0 2b 5f 00 00 02 04	05 b4 04 02 08 0a f7 e2	...+_	

Wireshark 3.20.0 (128164350) x86_64-linux

Packets: 13 - Displayed: 13 (100.0%) - Decoded: 0 (0.0%)

בתמונה זו ניתן לראות חבילה ראשונה שנשלחת בין הלקוח שכתובת ה- IP שלו היא 10.0.2.5 אל השרת שכתובת ה- IP שלו היא 10.0.2.8. לחבילה הראשונה אין שכבת אפליקציה אלא רק TCP Header. דגל SYN דולק משום שאנו מבקשים להסתנכרן עם הצד השני (השרת), בנוסף נשים לב כי דגל ה- ACK אינו דולק כי אין מה לאשר כרגע.

בצהוב נוכל לראות את ה – Source Port, כלומר מהיכן נשלח המידע ובירוק נוכל לראות לאן הוא נשלח (ה – Destination Port).

בצבע אדום מסומן ה – Sequence number האמיתי (3463997819) שהוא המספר הסידורי של הבית הראשון בהודעה והוא זה שיופיע בחבילה (הוא נבחר ע"י מי ששלח את ה SYN).

בצבע סגול מסומן ה – Acknowledgment number האמיתי שמשמש כפידבק, כלומר אישור על כך שההודעה הקודמת הגיעה, אך מאחר וזו החבילה הראשונה אזי אין מה לאשר. (ה ACK נבחר ע"י המקבל ומכיוון ועוד לא קיבל כלום, השולח אינו יכול לדעת מספר זה)

תמונה מספר 2

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.2.5	10.0.2.8	TCP	74	39364 → 12345 [SYN] Seq=0 Win=64240 Len=0 MSS...
2	0.000020	10.0.2.8	10.0.2.5	TCP	74	12345 → 39364 [SYN, ACK] Seq=0 Ack=1 Win=6516...
3	0.000213	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345 [ACK] Seq=1 Ack=1 Win=64256 Len...
4	0.000429	10.0.2.5	10.0.2.8	TCP	95	39364 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=6425...
5	0.000438	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364 [ACK] Seq=1 Ack=30 Win=65152 Le...
6	0.000507	10.0.2.8	10.0.2.5	TCP	95	12345 → 39364 [PSH, ACK] Seq=1 Ack=30 Win=651...

Transmission Control Protocol, Src Port: 12345, Dst Port: 39364, Seq: 0, Ack: 1, Len: 0

Source Port: 12345

Destination Port: 39364

[Stream index: 0]

[TCP Segment Len: 0]

Sequence number: 0 (relative sequence number)

Sequence number (raw): 2012182080

[Next sequence number: 1 (relative sequence number)]

Acknowledgment number: 1 (relative ack number)

Acknowledgment number (raw): 3463997820

1010 = Header Length: 40 bytes (10)

Flags: 0x012 (SYN, ACK)

0000 = Reserved: Not set

0000 08 00 27 d2 8d 71 08 00 27 24 4c de 08 00 45 00 - - . . q . . '\$L . . . E -

0010 00 3c 00 00 40 00 40 06 22 b0 0a 00 02 08 0a 00 - < . @ . @ . "

0020 02 05 30 39 99 c4 77 ef 76 40 ce 78 69 7c a0 12 - . 09 . w . v @ . x i | . .

0030 fe 88 18 3b 00 00 02 04 05 b4 04 02 08 0a 88 05 - . . ;

תמונה זו מציגה את התעבורה כאשר השרת מקבל את החבילה הראשונה מהלקוח, גם חבילה זו מכילה רק TCP Header.

כתובת ה - IP של השרת היא 10.0.2.8 וכתובת ה - IP של הלקוח היא 10.0.2.5 כפי שניתן לראות תחת הכותרות של Source ו - Destination המופיעות מעלה.

נתייחס לדגלים שדולקים:

דגל SYN – בדומה לחבילה הראשונה אנו מסתנכרנים עם הצד השני (הלקוח).

דגל ACK – מאשר את קבלת החבילה שנשלחה מהלקוח לשרת.

בצהוב נוכל לראות את ה – Source Port, כלומר מהיכן נשלח המידע ובירוק נוכל לראות לאן הוא נשלח (ה – Destination Port).

בצבע אדום מסומן ה – Sequence number האמיתי שהוא המספר הסידורי של הבית הראשון בהודעה שאותו בחר השרת, ונשים לב שהוא שונה מהמספר אותו בחר הלקוח קודם בצילום מספר 1, וזאת כיוון שהמספר ישמש את השרת ליידע את הלקוח מאיזה מספר בית התחיל לשדר את המידע.

בצבע סגול מסומן ה – Acknowledgment number האמיתי שמשמש כפידבק, כלומר האם ההודעה הגיעה באופן תקין.

נשים לב ש ה - Acknowledgment number שווה בדיוק למספר ה - Sequence number שנבחר על ידי הלקוח בתמונה מספר 1 ועוד אחד. $3463997820 = 3463997819 + 1$ שזהו מספר Acknowledgment number שניתן לראות בצילום זה. וזאת על מנת לסנכרן מול הלקוח ולדעת מה הוא מצפה לקבל ואם יקבל מידע אחר יוכל לזהות את הבעיה.

ניתן לראות כי קיימים מספרים אמיתיים RAW וקיימים מספרים רלטיביים. הסמפר השני הוא רק תצוגה נוחה יותר ע"י התוכנה אבל בפועל מה שמועבר זה המספרים האמיתיים.

נתייחס גם לשורה מספר 3, בה הלקוח שולח הודעה לשרת, גם חבילה זו מכילה רק TCP Header.

הדגל של ACK דולק משום שהוא משיב שהוא קיבל את החבילה שנשלחה מהשרת באופן תקין.

ה - Sequence number היחסי הוא 1 וה- Acknowledgment number היחסי הוא 1. מספר ה ACK האמיתי יהיה Sequence number של השרת ועוד 1.

כעת הסתיים תהליך "תחיצת היד" שבו למעשה התבצע סנכרון של הלקוח מול השרת וכעת כל צד יודע מאיזה מספר בית הוא עתיד לקבל את ההודעה הבאה.

תמונה מספר 3

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000...	10.0.2.5	10.0.2.8	TCP	74	39364 → 12345 [SYN] Seq=0 Win=64240 Len=0 MSS...
2	0.000020...	10.0.2.8	10.0.2.5	TCP	74	12345 → 39364 [SYN, ACK] Seq=0 Ack=1 Win=6516...
3	0.000213...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345 [ACK] Seq=1 Ack=1 Win=64256 Len...
4	0.000429...	10.0.2.5	10.0.2.8	TCP	95	39364 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=6425...
5	0.000438...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364 [ACK] Seq=1 Ack=30 Win=65152 Le...
6	0.000507...	10.0.2.8	10.0.2.5	TCP	95	12345 → 39364 [PSH, ACK] Seq=1 Ack=30 Win=651...
7	0.000713...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345 [ACK] Seq=30 Ack=30 Win=64256 L...
8	0.000713...	10.0.2.5	10.0.2.8	TCP	88	39364 → 12345 [PSH, ACK] Seq=30 Ack=30 Win=64...

Source Port: 39364
Destination Port: 12345
[Stream index: 0]
[TCP Segment Len: 29]
Sequence number: 1 (relative sequence number)
Sequence number (raw): 3463997820
[Next sequence number: 30 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
Acknowledgment number (raw): 2012182081
1000 = Header Length: 32 bytes (8)
- Flags: 0x018 (PSH, ACK)

0020	02 08 99 c4 30 39 ce 78	69 7c 77 ef 76 41	80 18	...09·x i w·vA..
0030	01 f6 7c bf 00 00 01 01	08 0a f7 e2 0f cf 88 05	
0040	37 a2 53 68 61 69 20 47	75 6e 64 65 72 73 65 6e		7·Shai G undersen
0050	20 61 6e 64 20 59 61 65	6c 20 41 76 69 6f 7a		and Yae l Avioz

Data (data.data), 29 bytes

Packets: 13 · Displayed: 13 | 1

כאן אנו רואים את החבילה מהלקוח לשרת, לחבילה זו כבר יש שכבת אפליקציה לעומת שלושת החבילות הראשונות שנשלחו בין השרת ללקוח.

בחבילה זו שלחנו מידע מהלקוח לשרת – המידע מסומן באפור בהיר בתחתית התמונה וניתן לראות את רצף הביטים המייצגים מידע זה בסמוך אליו.

הדגלים הדולקים הם:

ACK על החבילה הקודמת ו - PSH שמציין שהלקוח שלח מידע בשכבת האפליקציה.

בצהוב נוכל לראות את ה - Source Port, כלומר מהיכן נשלח המידע ובירוק נוכל לראות לאן הוא נשלח (ה - Destination Port).

בצבע אדום מסומן ה - Sequence number האמיתי שעתה הוא 3463997820. בחבילה זו אנו שולחים מידע בגודל 29 בתים ולכן נצפה לראות כי בחבילה הבאה שהלקוח ישלח ה NUM SEQ ישתנה להיות $3463997820 + 29 = 3463997849$.

בצבע סגול מסומן ה - Acknowledgment number האמיתי ששווה ל - Sequence number של השרת ועוד 1 (2012182081). שהרי בשורה 3 נשלחה הודעת ACK.

בשורה מספר 5, השרת מקבל את ההודעה ואנו יודעים זאת כי דגל ה-ACK דולק. נתבונן לרגע במספר ACK היחסי בשורה זו שהוא 29, וזאת כיוון שנשלחו 29 בתים של מידע (השמות שלנו כולל רווחים) ולכן השרת מודיע כי קיבל מידע, והמידע הבא שיצפה לקבל הוא מהבית ה-30 והלאה.

תמונה מספר 4

File

Edit

View

Go

Capture

Analyze

Statistics

Telephony

Wireless

Tools

Help

tcp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000...	10.0.2.5	10.0.2.8	TCP	74	39364 → 12345 [SYN] Seq=0 Win=64240 Len=0 MSS...
2	0.000020...	10.0.2.8	10.0.2.5	TCP	74	12345 → 39364 [SYN, ACK] Seq=0 Ack=1 Win=6516...
3	0.000213...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345 [ACK] Seq=1 Ack=1 Win=64256 Len...
4	0.000429...	10.0.2.5	10.0.2.8	TCP	95	39364 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=6425...
5	0.000438...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364 [ACK] Seq=1 Ack=30 Win=65152 Le...
6	0.000507...	10.0.2.8	10.0.2.5	TCP	95	12345 → 39364 [PSH, ACK] Seq=1 Ack=30 Win=651...
7	0.000713...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345 [ACK] Seq=30 Ack=30 Win=64256 L...
8	0.000713...	10.0.2.5	10.0.2.8	TCP	88	39364 → 12345 [PSH, ACK] Seq=30 Ack=30 Win=64...

Transmission Control Protocol, Src Port: 12345, Dst Port: 39364, Seq: 1, Ack: 30, Len: 29

Source Port: 12345

Destination Port: 39364

[Stream index: 0]

[TCP Segment Len: 29]

Sequence number: 1 (relative sequence number)

Sequence number (raw): 2012182081

[Next sequence number: 30 (relative sequence number)]

Acknowledgment number: 30 (relative ack number)

Acknowledgment number (raw): 3463997849

1000 = Header Length: 32 bytes (8)

Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0020	02	05	30	39	99	c4	77	ef	76	41	ce	78	69	99	80	18	...	09	...	w	...	va	...	xi	
0030	01	fd	18	50	00	00	01	01	08	0a	88	05	37	a3	f7	e2	
0040	0f	cf	53	48	41	49	20	47	55	4e	44	45	52	53	45	4e	
0050	20	41	4e	44	20	59	41	45	4c	20	41	56	49	4f	5a	

Data (data.data), 29 bytes

Packets: 13 · Displayed: 13 (100.0%) · Dropped: 0 (0.0%)

ואכן, בהמשך לתמונה מספר 3, ניתן לראות כי השרת מודיע ללקוח כי קיבל הודעת הלקוח והכניס אותה תחת ה-ACK הנוכחי, שהוא ה-SEQ שחישבנו בתמונה הקודמת 3463997849.

בתמונה זו, לאחר שהשרת קיבל את ההודעה עם השמות שלנו, ביקשנו ממנו להחזיר את השמות באותיות גדולות ולכן כאן גם דגל ה-ACK דולק וגם דגל ה-PSH דולק כי הוא שולח לנו מידע.

את המידע שנשלח ניתן לראות בתחתית התמונה בצבע אפור, בסמוך לביטים המייצגים אותו.

בצהוב נוכל לראות את ה-Source Port, כלומר מהיכן נשלח המידע ובירוק נוכל לראות לאן הוא נשלח (ה-Destination Port).

בצהוב נוכל לראות את ה-Source Port, כלומר מהיכן נשלח המידע ובירוק נוכל לראות לאן הוא נשלח (ה-Destination Port).

בצבע אדום מסומן ה-Sequence number וניתן לראות כי הוא לא השתנה מאז שורה מספר 4 וזאת כיוון שהשרת לא שלח מידע נוסף בזמן הזה. (למעט +1 על הודעת הסנכון בשורה 2). היות והשרת שולח מידע בגודל 29 בתים לעבר הלקוח, נצפה כי בהודעת ה-ACK שהלקוח ישלח לשרת, יופיע המספר הקודם לו בתוספת 29 בתים, כלומר $2012182110 = 2012182081 + 29$.


בצבע סגול מסומן ה-Acknowledgment number שהוא אכן שווה ל-ACK בשורה 5 שכן, לא התקבל מידע נוסף מאז והוא "מתזכר" את הלקוח במספר שלו.

ואכן, בשורה מספר 7, הלקוח מאשר את קבלת החבילה ולכן דגל ה-ACK דולק, ובנוסף ניתן לראות כי הלקוח מאשר קבלת המידע ע"י כך שהוא שולח ACK עם מספר יחסי של 30.

*בתמונה הבאה נראה כיצד זה בא לידי ביטוי במספרים האמיתיים ולא הרלטיביים.

תמונה מספר 5

File Edit View Go Capture Analyze Statistics Help



tcp

No.	Time	Source	Destination	Protocol	Length	Info
40	0.000429...	10.0.2.5	10.0.2.8	TCP	95	39364 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=6425...
50	0.000438...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364 [ACK] Seq=1 Ack=30 Win=65152 Le...
60	0.000507...	10.0.2.8	10.0.2.5	TCP	95	12345 → 39364 [PSH, ACK] Seq=1 Ack=30 Win=651...
70	0.000713...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345 [ACK] Seq=30 Ack=30 Win=64256 L...
80	0.000713...	10.0.2.5	10.0.2.8	TCP	88	39364 → 12345 [PSH, ACK] Seq=30 Ack=30 Win=64...
90	0.000722...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364 [ACK] Seq=30 Ack=52 Win=65152 L...
100	0.000768...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364 [FIN, ACK] Seq=30 Ack=52 Win=65...
110	0.000898...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345 [FIN, ACK] Seq=52 Ack=30 Win=64...

- Transmission Control Protocol, Src Port: 39364, Dst Port: 12345, Seq: 30, Ack: 30, Len: 22

Source Port: 39364

Destination Port: 12345

[Stream index: 0]

[TCP Segment Len: 22]

Sequence number: 30 (relative sequence number)

Sequence number (raw): 3463997849

[Next sequence number: 52 (relative sequence number)]

Acknowledgment number: 30 (relative ack number)

Acknowledgment number (raw): 2012182110

1000 ... = Header Length: 32 bytes (8)

0020	02 08 99 c4 30 39 ce 78	69 99 77 ef 76 5e 80 18	...09.x i.w.v^..
0030	01 f6 89 c5 00 00 01 01	08 0a f7 e2 0f d0 88 05
0040	37 a3 32 30 34 37 35 33	31 32 33 20 2f 2f 20 32	7.204753 123 // 2
0050	30 37 32 33 37 34 32 31		07237421

Data (data.data), 22 bytes

Packets: 13 · Displayed: 13 (100.0%) · Dropped: 0 (0.0%)

בהמשך לשורה מספר 7, אכן ניתן לראות כי המספר ACK האמיתי 2012182110 שזה בדיוק הבדל של 29 בתים מהפעם האחרונה בה נשלח מידע לכיוון השרת. לאחר שקיבלנו מהשרת את התשובה להודעה שנשלחה והלקוח אישר שהמידע הגיע באופן תקין, אנו שולחים את ההודעה הבאה המכילה את מספרי תעודות הזהות שלנו. את המידע ניתן לראות בתחתית התמונה בצבע אפור ולצידו את הביטים המייצגים את המידע הכתוב. מאחר והלקוח שולח הודעה חדשה דגל ה - PSH דולק וגם דגל ה - ACK דולק שמאשר את קבלת החבילה האחרונה שנשלחה מהשרת. בצהוב נוכל לראות את ה - Source Port, כלומר מהיכן נשלח המידע ובירוק נוכל לראות לאן הוא נשלח (ה - Destination Port).

בצבע אדום מסומן ה - Sequence number האמיתי שהוא נשאר זהה ל - 3463997849, שזהו הניתוח שעשינו עבור תמונה 3. היות ואנו שולחים מידע בגודל 22 בתים, נצפה כי SEQ הבא של הלקוח יהיה מהמספר 3463997871. והACK של השרת יהיה גם כן אותו המספר (במידה והכל הגיע תקין) בצבע סגול מסומן ה - Acknowledgment number ואכן ניתן לראות כי הוא מתאים לניתוח שבצענו עבור תמונה מספר 4.

בשורה 9, השרת מאשר את קבלת החבילה ואנו רואים זאת באמצעות הדגל ACK שדולק.

תמונה מספר 6

No.	Time	Source	Destination	Protocol	Length	Info
7	0.000713...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345 [ACK] Seq=30 Ack=30 Win=64256 L...
8	0.000713...	10.0.2.5	10.0.2.8	TCP	88	39364 → 12345 [PSH, ACK] Seq=30 Ack=30 Win=64...
9	0.000722...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364 [ACK] Seq=30 Ack=52 Win=65152 L...
10	0.000768...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364 [FIN, ACK] Seq=30 Ack=52 Win=65...
11	0.000898...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345 [FIN, ACK] Seq=52 Ack=30 Win=64...
12	0.000903...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364 [ACK] Seq=31 Ack=53 Win=65152 L...
13	0.001011...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345 [ACK] Seq=53 Ack=31 Win=64256 L...

- Transmission Control Protocol, Src Port: 12345, Dst Port: 39364, Seq: 30, Ack: 52, Len: 0	
Source Port: 12345	
Destination Port: 39364	
[Stream index: 0]	
[TCP Segment Len: 0]	
Sequence number: 30 (relative sequence number)	
Sequence number (raw): 2012182110	
[Next sequence number: 31 (relative sequence number)]	
Acknowledgment number: 52 (relative ack number)	
Acknowledgment number (raw): 3463997871	
1000 = Header Length: 32 bytes (8)	

0000	08 00 27 d2 8d 71 08 00	27 24 4c de 08 00 45 00	..'.q.. '\$L...E.
0010	00 34 80 6e 40 00 40 06	a2 49 0a 00 02 08 0a 00	.4.n@.@. .I.....
0020	02 05 30 39 99 c4 77 ef	76 5e ce 78 69 af 80 11	..09..w. v^..xi...
0030	01 fd 18 33 00 00 01 01	08 0a 88 05 37 a3 f7 e2	...3.... ..7...

בתמונה זו השרת רוצה לסגור את החיבור עם הלקוח ולכן הוא שולח ללקוח הודעה עם הדגל FIN, כלומר לשרת אין יותר מה לשלוח ללקוח מעבר ל-Sequence number 2012182110 ולכן הוא סוגר את הסוקט אצלו. דגל ה-ACK שדולק מציין אישור על הודעה שנשלחה קודם מהלקוח לשרת עם מספרי תעודות הזהות שלנו – אנו יודעים זאת לפי Acknowledgment number לפי הניתוח שעשינו עבור תמונה מספר 5. בצהוב נוכל לראות את ה – Source Port, כלומר מהיכן נשלח המידע ובירוק נוכל לראות לאן הוא נשלח (ה – Destination Port). בצבע אדום מסומן ה – Sequence number, נצפה כי בהודעה הבאה שהשרת ישלח ללקוח (וזו יהיה רק הודעת ACK משורה 12) כי ה-SEQ יהיה +1 רק עבור קבלת הודעת הסגירה (אומנם אין תצלום להמשך הזה אבל אפשר לראות כי ה-SEQ היחסי השתנה מ 30 בשורה 10, ל 31 בשורה 12. בצבע סגול מסומן ה – Acknowledgment number. גם כאן יקרה אותו תהליך כמו שנכתב למעלה.

תמונה מספר 7

No.	Time	Source	Destination	Protocol	Length	Info
70.000713...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345	[ACK] Seq=30 Ack=30 Win=64256 L...
80.000713...	10.0.2.5	10.0.2.8	TCP	88	39364 → 12345	[PSH, ACK] Seq=30 Ack=30 Win=64...
90.000722...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364	[ACK] Seq=30 Ack=52 Win=65152 L...
100.000768...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364	[FIN, ACK] Seq=30 Ack=52 Win=65...
110.000898...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345	[FIN, ACK] Seq=52 Ack=30 Win=64...
120.000903...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39364	[ACK] Seq=31 Ack=53 Win=65152 L...
130.001011...	10.0.2.5	10.0.2.8	TCP	66	39364 → 12345	[ACK] Seq=53 Ack=31 Win=64256 L...

Transmission Control Protocol, Src Port: 39364, Dst Port: 12345, Seq: 52, Ack: 30, Len: 0	
Source Port: 39364	
Destination Port: 12345	
[Stream index: 0]	
[TCP Segment Len: 0]	
Sequence number: 52 (relative sequence number)	
Sequence number (raw): 3463997871	
[Next sequence number: 53 (relative sequence number)]	
Acknowledgment number: 30 (relative ack number)	
Acknowledgment number (raw): 2012182110	
1000 = Header Length: 32 bytes (8)	

0000	08 00 27 24 4c de 08 00	27 d2 8d 71 08 00 45 00	.. '\$L... '...q...E..
0010	00 34 52 b8 40 00 40 06	cf ff 0a 00 02 05 0a 00	..4R..@..@..
0020	02 08 99 c4 30 39 ce 78	69 af 77 ef 76 5e 80 1109..x i.w.v^..
0030	01 f6 a4 ea 00 00 01 01	08 0a f7 e2 0f d0 88 05

בשורה זו ניתן לראות כי הלקוח שלח לשרת חבילה הוא מאשר את ההתנקות, לכן דגל ה – FIN דולק ובנוסף דגל ה – ACK דולק. חשוב להדגיש כי דגל ה ACK תמיד דולק למעט בהודעה ההראשונה, גם אם לא נשלחה הודעה משמעותית, וזאת על מנת "לתזכר" את הצד השני מאיזה בית הוא מצפה לקבל וכדי לוודא שלא הגיעו חבילות נוספות בטעות מאותו הפורט.

בצהוב נוכל לראות את ה – Source Port, כלומר מהיכן נשלח המידע ובירוק נוכל לראות לאן הוא נשלח (ה – Destination Port).

בצבע אדום מסומן ה – Sequence number, נשים לב כי ה SEQ אכן השתנה למספר אשר חישבנו בתמונה מספר 5. נצפה כי בהודעה הבאה שהלקוח ישלח לשרת (וזוה יהיה רק הודעת ACK משורה 13) כי ה SEQ יהיה +1 רק עבור קבלת הודעת הסגירה (אומנם אין תצלום להמשך הזה אבל אפשר לראות כי ה SEQ היחסי השתנה מ 52 בשורה 11, ל 52 בשורה 13. בצבע סגול מסומן ה – Acknowledgment number גם כאן יקרה אותו תהליך כמו שנכתב למעלה.

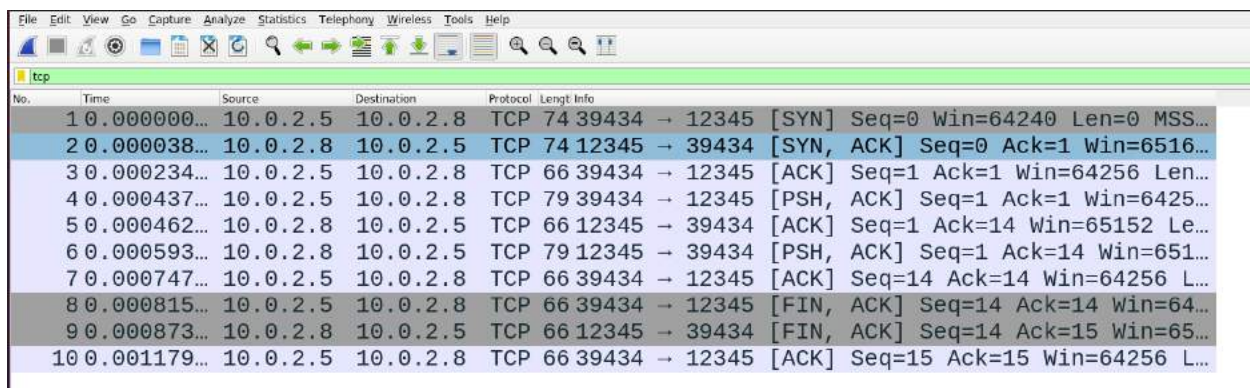
שורות 12 ו – 13 הן בעצם אישור של הלקוח לשרת ולהפך על ההתנתקות ביניהם.

הרצת קבצים V 1-4

גרסה 1:

השרת פותח סוקט ויכול להאזין לבקשה אחת בבאפר שלו, ואז מחכה לקבלת לקוח חדש.
הלקוח מבקש להתחבר לשרת (כאן למעשה מתבצעת "לחיצת היד" והסנכרון בין הלקוח והשרת).
השרת מאשר ומדפיס למסך מאיזו כתובת הגיעה ההודעה.
השרת נכנס ללולאה שתמשיך כל עוד נשאר מידע לקרוא מתוך הבאפר. כלומר השרת ממשיך לטפל באותו הלקוח ולא מתפנה לטיפול בלקוחות אחרים בינתיים.
השרת מצפה לקבל הודעה.
הלקוח שולח הודעה: Hello, world!.
השרת מדפיס למסך את תוכן ההודעה שהתקבלה.
השרת שולח חזרה את אותה הודעה רק באותיות גדולות.
הלקוח מקבל את ההודעה.
הלקוח סוגר את החיבור מול השרת.
השרת יסיים לקרוא את הבאפר ויסגור את החיבור מול הלקוח.

צילום מסך:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000...	10.0.2.5	10.0.2.8	TCP	74	39434 → 12345 [SYN] Seq=0 Win=64240 Len=0 MSS...
2	0.000038...	10.0.2.8	10.0.2.5	TCP	74	12345 → 39434 [SYN, ACK] Seq=0 Ack=1 Win=6516...
3	0.000234...	10.0.2.5	10.0.2.8	TCP	66	39434 → 12345 [ACK] Seq=1 Ack=1 Win=64256 Len...
4	0.000437...	10.0.2.5	10.0.2.8	TCP	79	39434 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=6425...
5	0.000462...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39434 [ACK] Seq=1 Ack=14 Win=65152 Le...
6	0.000593...	10.0.2.8	10.0.2.5	TCP	79	12345 → 39434 [PSH, ACK] Seq=1 Ack=14 Win=651...
7	0.000747...	10.0.2.5	10.0.2.8	TCP	66	39434 → 12345 [ACK] Seq=14 Ack=14 Win=64256 L...
8	0.000815...	10.0.2.5	10.0.2.8	TCP	66	39434 → 12345 [FIN, ACK] Seq=14 Ack=14 Win=64...
9	0.000873...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39434 [FIN, ACK] Seq=14 Ack=15 Win=65...
10	0.001179...	10.0.2.5	10.0.2.8	TCP	66	39434 → 12345 [ACK] Seq=15 Ack=15 Win=64256 L...

בצילום זה ניתן לראות, כמו בחלק הקודם, את התעבורה בין הלקוח לשרת.
שתי החבילות הראשונות מבטאות למעשה את "לחיצת היד" המתבצעת. בשורה הראשונה הלקוח מבקש להתחבר ובשורה השנייה השרת שולח אישר על בקשת הלקוח להתחברות ובנוסף שולח בקשת התחברות ללקוח (מילה נכונה יותר מאשר התחברות היא הסנכרונות, שם כל אחד ישלח את הSEQUENCE NUMBER שממנו הוא מתחיל)

שורה 3 הלקוח מאשר את בקשת הסנכרון של השרת.
שורה 4 הלקוח שולח את ההודעה הנ"ל.
שורה 5 השרת מאשר קבלת ההודעה.
שורה 6 השרת שולח את אותה הודעה רק באותיות גדולות.
שורה 7 הלקוח מאשר קבלת ההודעה.
שורה 8 הלקוח סגר את החיבור ולכן שולח הודעת FIN לשרת.
שורה 9 השרת מאשר קבלת סגירת החיבור מצד השרת ושולח הודעת FIN ללקוח.
שורה 10 הלקוח מאשר קבלת סגירת החיבור מצד השרת.
כעת נראה שני צילומי מסך נבחרים של ההודעות שנשלחו:

עבור שורה 4:

```

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp
No. Time Source Destination Protocol Length Info
4 0.000437... 10.0.2.5 10.0.2.8 TCP 79 39434 -> 12345 [PSH, ACK] Seq=1 Ack=1 W
Sequence number: 1 (relative sequence number)
Sequence number (raw): 3829307602
[Next sequence number: 14 (relative sequence number)]
Acknowledgment number: 1 (relative ack number)
Acknowledgment number (raw): 1768710239
1000 .... = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
window size value: 502
[Calculated window size: 64256]
[Window size scaling factor: 128]
Checksum: 0x5313 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]
[Timestamps]
TCP payload (13 bytes)
Data (13 bytes)
0010 00 41 d0 b2 40 00 40 06 51 f8 0a 00 02 05 0a 00 A..@. Q.....
0020 02 08 9a 0a 30 39 e4 3e 98 d2 69 6c 60 5f 80 18 ...09->..il`_..
0030 01 f6 53 13 00 00 01 01 08 0a f8 0d 87 ac 88 30 ..S.....0
0040 af 5a 48 65 6c 6c 6f 2c 20 57 6f 72 6c 64 21 zHello, world!

```

ניתן לראות כי בתחילת ה-TCP דלוקים שני דגלים. האחד הוא ACK. ניתן להבחין כי ה-ACK הרלטיבי הוא 1 והוא זהה ל-ACK בשורה 3 וזאת מכיוון שבין שורה 3 ל-4 הלקוח לא קיבל מידע נוסף. הדגל השני הוא PSH שמעיד כי נשלח DATA שאותו שכבת האפליקציה צריכה לקבל. ואכן, ניתן לראות בצד שמאל למטה כי אכן הועבר DATA בגודל 13 בתים ובצד ימין למטה ניתן לראות את תוכן ההודעה. בנוסף:

עבור שורה 5:

```

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp
No. Time Source Destination Protocol Length Info
5 0.000462... 10.0.2.8 10.0.2.5 TCP 66 12345 -> 39434 [ACK] Seq=1 Ack=14 Win=65152 Le...
[Stream index: 0]
[TCP Segment Len: 0]
Sequence number: 1 (relative sequence number)
Sequence number (raw): 1768710239
[Next sequence number: 1 (relative sequence number)]
Acknowledgment number: 14 (relative ack number)
Acknowledgment number (raw): 3829307615
1000 .... = Header Length: 32 bytes (8)
Flags: 0x010 (ACK)
Window size value: 509
[Calculated window size: 65152]
[Window size scaling factor: 128]
Checksum: 0x1833 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]
[Timestamps]
0010 00 34 b6 ee 40 00 40 06 6b c9 0a 00 02 08 0a 00 4..@. k.....
0020 02 05 30 39 9a 0a 69 6c 60 5f e4 3e 98 df 80 10 ...09..il`_>...
0030 01 fd 18 33 00 00 01 01 08 0a 88 30 af 5b f8 0d ...3.....0.[
0040 87 ac

```

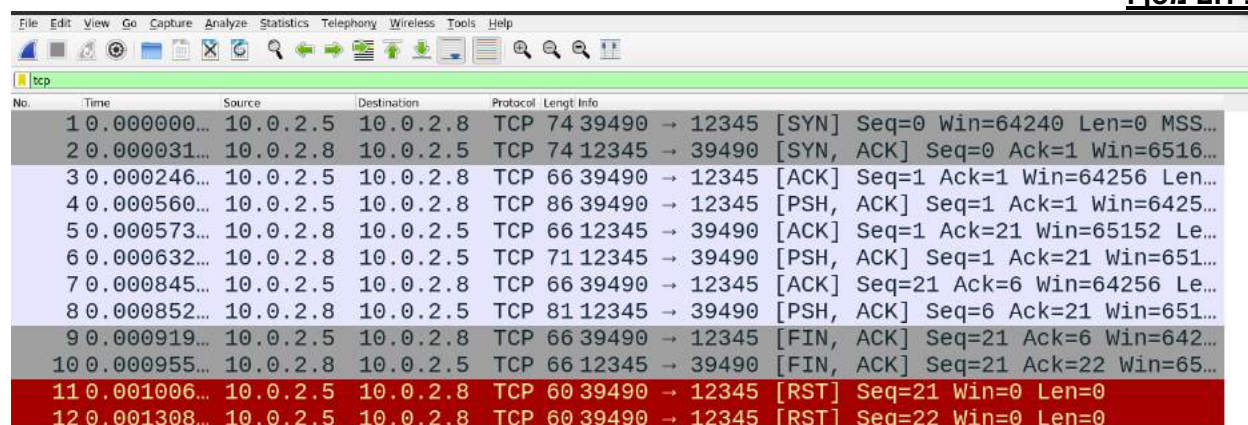
ניתן לראות כי השרת שולח ACK ללקוח המביע קבלת הודעה. בשורה 2 ה-ACK הרלטיבי היה 1 וכעת, לראות כי ה-ACK הרלטיבי התעדכן ועומד על 14 שכן הוא קיבל DATA בגודל 13 בתים ומעתה יצפה לקבל מידע מהבית מספר 14 והלאה.

גרסה 2:

בגרסה זו, ישנם שני הבדלים. בצד השרת גודל הבאפר הוא 5 (לעומת פעם קודמת שהיה 1024). שינוי בגודל הבאפר יוביל לכך שהשרת יקרא רק 5 בתים מתוך ההודעה שנשלחה. שאר הבתים של ההודעה ישארו בחלק של ה-receive buffer בצד השרת. ולכן בלולאה בכל פעם השרת יקרא 5 בתים מתוך הבאפר, ישלח אותם באותיות גדולות, ולאחר סיום קריאת הבאפר יסגור את החיבור. בצד הלקוח, השוני הוא שההודעה מכילה יותר בתים.

*לכן הציפייה שלנו היא שהלקוח ישלח הודעה אחת לשרת שמכילה DATA ואילו השרת ישלח כמות הודעות, אשר מכילות DATA, ככמות הבתים לחלק ל-5 ערך עליון. (במקרה שלנו ההודעה מכילה 20 בתים ולכן השרת ישלח 4 הודעות עם אותיות גדולות חזרה ללקוח) (כמובן שלא לקחנו בחשבון את כמות ה-ACK-ים שכמובן תעלה כי על כל הודעה TCP ישלח כי הוא קיבל)

צילום מסך:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000...	10.0.2.5	10.0.2.8	TCP	74	39490 → 12345 [SYN] Seq=0 Win=64240 Len=0 MSS...
2	0.000031...	10.0.2.8	10.0.2.5	TCP	74	12345 → 39490 [SYN, ACK] Seq=0 Ack=1 Win=6516...
3	0.000246...	10.0.2.5	10.0.2.8	TCP	66	39490 → 12345 [ACK] Seq=1 Ack=1 Win=64256 Len...
4	0.000560...	10.0.2.5	10.0.2.8	TCP	86	39490 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=6425...
5	0.000573...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39490 [ACK] Seq=1 Ack=21 Win=65152 Le...
6	0.000632...	10.0.2.8	10.0.2.5	TCP	71	12345 → 39490 [PSH, ACK] Seq=1 Ack=21 Win=651...
7	0.000845...	10.0.2.5	10.0.2.8	TCP	66	39490 → 12345 [ACK] Seq=21 Ack=6 Win=64256 Le...
8	0.000852...	10.0.2.8	10.0.2.5	TCP	81	12345 → 39490 [PSH, ACK] Seq=6 Ack=21 Win=651...
9	0.000919...	10.0.2.5	10.0.2.8	TCP	66	39490 → 12345 [FIN, ACK] Seq=21 Ack=6 Win=642...
10	0.000955...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39490 [FIN, ACK] Seq=21 Ack=22 Win=65...
11	0.001006...	10.0.2.5	10.0.2.8	TCP	60	39490 → 12345 [RST] Seq=21 Win=0 Len=0
12	0.001308...	10.0.2.5	10.0.2.8	TCP	60	39490 → 12345 [RST] Seq=22 Win=0 Len=0

נתבונן בשורות המעניינות : 4, 6, 8, 11, 12
שורה 4 הלקוח שולח הודעה בעלת 20 בתים.
שורה 6 השרת קרא 5 בתים מהבאפר ושולח ללקוח את 5 בתים הללו באותיות גדולות.
שורה 8 השרת שולח את יתר 15 בתים באותיות גדולות ללקוח. (בניגוד לציפייה שלנו!)
שורה 11 הלקוח מנסה לשלוח לשרת ACK על קבלת ההודעה שנשלחה בשורה 8 אך השרת כבר ניתק את החיבור.
שורה 12 הלקוח מנסה לשלוח לשרת ACK על קבלת הודעת התנקות בשורה 10 אך השרת כבר ניתק את החיבור.

כעת נראה שני צילומי מסך נבחרים של ההודעות שנשלחו:

שורה 6:

No.	Time	Source	Destination	Protocol	Length	Info
4	0.000560...	10.0.2.5	10.0.2.8	TCP	86	39490 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=6425...
5	0.000573...	10.0.2.8	10.0.2.5	TCP	66	12345 → 39490 [ACK] Seq=1 Ack=21 Win=65152 Le...
6	0.000632...	10.0.2.8	10.0.2.5	TCP	71	12345 → 39490 [PSH, ACK] Seq=1 Ack=21 Win=651...

• Frame 6: 71 bytes on wire (568 bits), 71 bytes captured (568 bits)
 • Ethernet II, Src: PcsCompu_24:4c:de (08:00:27:24:4c:de), Dst: PcsCompu_d2:8d:71 (08:00:27:24:4c:de)
 • Internet Protocol Version 4, Src: 10.0.2.8, Dst: 10.0.2.5
 • Transmission Control Protocol, Src Port: 12345, Dst Port: 39490, Seq: 1, Ack: 21, Len: 5
 • Data (5 bytes)
 Data: 574f524c44
 [Length: 5]

0010	00 39 a2 23 40 00 40 06	80 8f 0a 00 02 08 0a 00	..9.#@-@
0020	02 05 30 39 9a 42 bf 5d	91 79 7c 9e 0c a8 80 18	..09.B.] .y
0030	01 fd 18 38 00 00 01 01	08 0a 88 62 47 c0 f8 3f	..8.....bg-?
0040	1f fb 57 4f 52 4c 44		..WORLD

Data (data.data), 5 bytes

אכן, כמו שציפינו השרת שולח 5 בתים באותיות גדולות

שורה 8:

No.	Time	Source	Destination	Protocol	Length	Info
7	0.000845...	10.0.2.5	10.0.2.8	TCP	66	39490 → 12345 [ACK] Seq=21 Ack=6 Win=64256 Le...
8	0.000852...	10.0.2.8	10.0.2.5	TCP	81	12345 → 39490 [PSH, ACK] Seq=6 Ack=21 Win=651...
9	0.000919...	10.0.2.5	10.0.2.8	TCP	66	39490 → 12345 [FIN, ACK] Seq=21 Ack=6 Win=642...

• Frame 8: 81 bytes on wire (648 bits), 81 bytes captured (648 bits)
 • Ethernet II, Src: PcsCompu_24:4c:de (08:00:27:24:4c:de), Dst: PcsCompu_d2:8d:71 (08:00:27:d2:8d:71)
 • Internet Protocol Version 4, Src: 10.0.2.8, Dst: 10.0.2.5
 • Transmission Control Protocol, Src Port: 12345, Dst Port: 39490, Seq: 6, Ack: 21, Len: 15
 • Data (15 bytes)
 Data: 212048454c4c4f2c20574f524c4421
 [Length: 15]

0020	02 05 30 39 9a 42 bf 5d	91 7e 7c 9e 0c a8 80 18	..09.B.] -
0030	01 fd 18 42 00 00 01 01	08 0a 88 62 47 c0 f8 3f	..B.....bg-?
0040	1f fb 21 20 48 45 4c 4c	4f 2c 20 57 4f 52 4c 44	..! HELL O, WORLD
0050	21		!

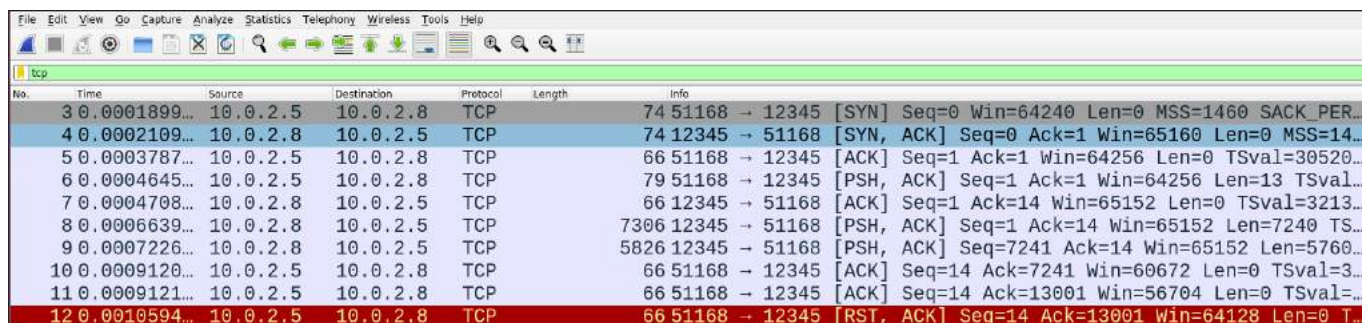
Data (data.data), 15 bytes

שלא כמו שציפינו, השרת שולח ללקוח 15 בתים באותיות גדולות (ניתן לראות בצד ימין למטה את תוכן ההודעה) נסביר זאת. פרוטוקול TCP שולח בתים ולא הודעות ולכן שולח כמות בתים כממות MSS. (לא רואים בתמונה אבל כמובן שהMSS גדול 5 בתים, יותר לכיוון 1460 בתים) בזמן שעבר בין שליחת ההודעה בשורה 6 לבין קבלת ACK עליה מהלקוח בשורה 7, הלולאה בשכבת האפליקציה הספיקה לקרוא את 15 הבתים הנותרים מהבאפר ופרוטוקול הTCP הספיק להכין את החבילה הבאה לשליחה ושלה אותה.

גרסה 3:

בגרסה זו, הלקוח שולח מידע בגודל 13 בתים, השרת יחזיר ללקוח 13 בתים*1000 באותיות גדולות. ואז הלקוח קורא 1024 בתים מהבאפר שלו, מדפיס אותם ושוב פעם.
*הציפייה שלנו היא שהשרת ישלח כמות חבילות גדולה אשר שווה למספר הבתים הכולל חלקי MSS, כי פרוטוקול TCP ינסה לחלק את המידע לסגמנטים.

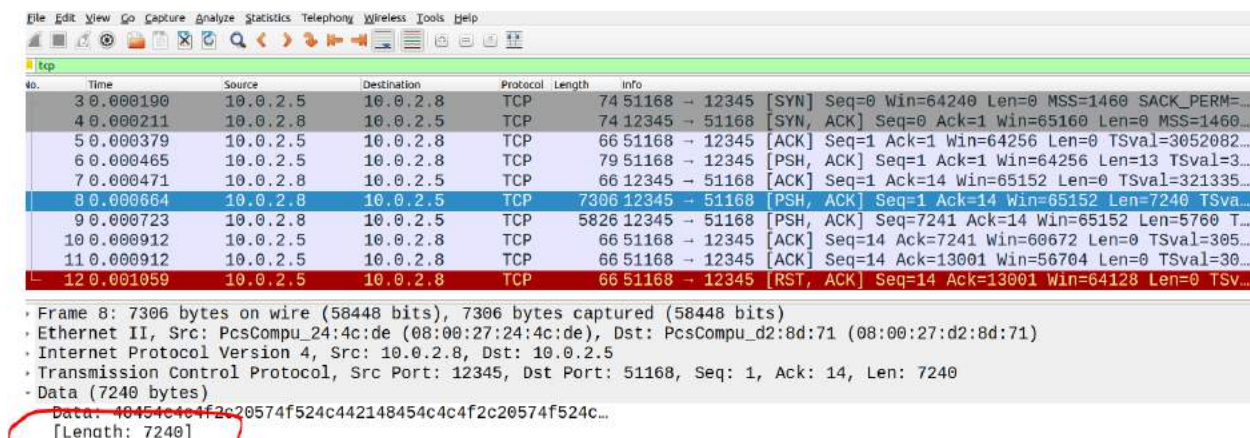
צילום מסך:



No.	Time	Source	Destination	Protocol	Length	Info
3	0.0001899...	10.0.2.5	10.0.2.8	TCP	74	51168 → 12345 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM...
4	0.0002109...	10.0.2.8	10.0.2.5	TCP	74	12345 → 51168 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=14...
5	0.0003787...	10.0.2.5	10.0.2.8	TCP	66	51168 → 12345 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=30520...
6	0.0004645...	10.0.2.5	10.0.2.8	TCP	79	51168 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=13 TSval=...
7	0.0004708...	10.0.2.8	10.0.2.5	TCP	66	12345 → 51168 [ACK] Seq=1 Ack=14 Win=65152 Len=0 TSval=3213...
8	0.0006639...	10.0.2.8	10.0.2.5	TCP	7306	12345 → 51168 [PSH, ACK] Seq=1 Ack=14 Win=65152 Len=7240 TS...
9	0.0007226...	10.0.2.8	10.0.2.5	TCP	5826	12345 → 51168 [PSH, ACK] Seq=7241 Ack=14 Win=65152 Len=5760...
10	0.0009120...	10.0.2.5	10.0.2.8	TCP	66	51168 → 12345 [ACK] Seq=14 Ack=7241 Win=60672 Len=0 TSval=3...
11	0.0009121...	10.0.2.5	10.0.2.8	TCP	66	51168 → 12345 [ACK] Seq=14 Ack=13001 Win=56704 Len=0 TSval=...
12	0.0010594...	10.0.2.5	10.0.2.8	TCP	66	51168 → 12345 [RST, ACK] Seq=14 Ack=13001 Win=64128 Len=0 T...

לאחר ביסוס החיבור, בשורה 6 הלקוח שולח לשרת חבילה בגודל 13 בתים. בניגוד לציפייה שלנו, לא נשלחו המון חבילות אל הלקוח אלא רואים כי בשורות 8 וגם 9 השרת שולח הודעה ארוכה עם המון בתים. לאחר שיח עם חמי בנושא הוסבר כי ישנם מקרים בהם wireshark מראה את כמות הבתים הסופית שיצאה מכרטיס רשת מסוים אבל בפועל המאחורי הקלעים TCP כן מפרק את המידע לסגמנטים בגודל MSS.

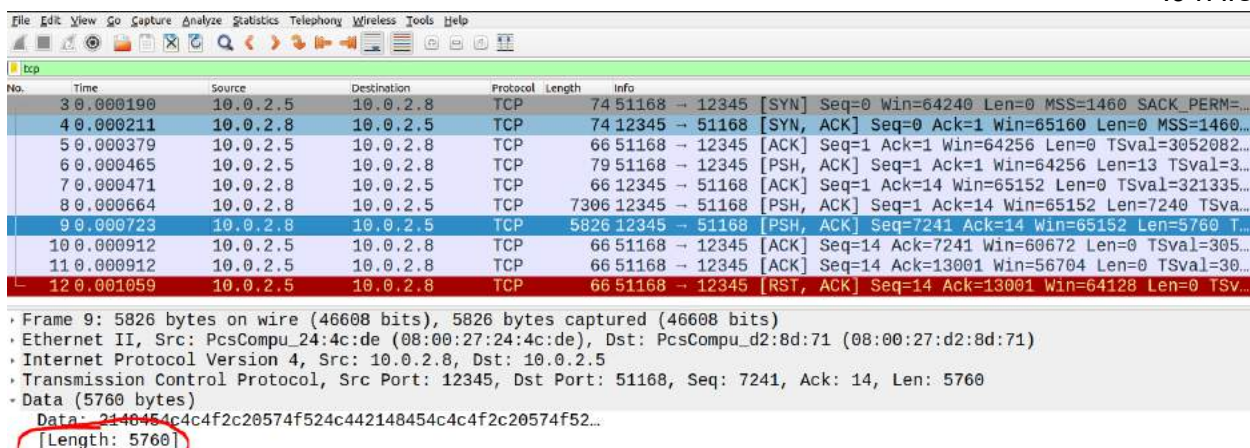
בשורה 12 אנו עדים לכך שהשרת סגר את החיבור מול הלקוח לפני הזמן, שכן הלקוח לא מסוגל לשלוח ACK על החבילה האחרונה שהתקבלה אצלו.
נראה כעת כי אכן כמות הבתים הסופית שנשלחה היא אכן 13,000 בתים:
שורה 8:



No.	Time	Source	Destination	Protocol	Length	Info
3	0.000190	10.0.2.5	10.0.2.8	TCP	74	51168 → 12345 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=...
4	0.000211	10.0.2.8	10.0.2.5	TCP	74	12345 → 51168 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460...
5	0.000379	10.0.2.5	10.0.2.8	TCP	66	51168 → 12345 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3052082...
6	0.000465	10.0.2.5	10.0.2.8	TCP	79	51168 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=13 TSval=3...
7	0.000471	10.0.2.8	10.0.2.5	TCP	66	12345 → 51168 [ACK] Seq=1 Ack=14 Win=65152 Len=0 TSval=321335...
8	0.000664	10.0.2.8	10.0.2.5	TCP	7306	12345 → 51168 [PSH, ACK] Seq=1 Ack=14 Win=65152 Len=7240 TSva...
9	0.000723	10.0.2.8	10.0.2.5	TCP	5826	12345 → 51168 [PSH, ACK] Seq=7241 Ack=14 Win=65152 Len=5760 T...
10	0.000912	10.0.2.5	10.0.2.8	TCP	66	51168 → 12345 [ACK] Seq=14 Ack=7241 Win=60672 Len=0 TSval=305...
11	0.000912	10.0.2.5	10.0.2.8	TCP	66	51168 → 12345 [ACK] Seq=14 Ack=13001 Win=56704 Len=0 TSval=30...
12	0.001059	10.0.2.5	10.0.2.8	TCP	66	51168 → 12345 [RST, ACK] Seq=14 Ack=13001 Win=64128 Len=0 TSV...

Frame 8: 7306 bytes on wire (58448 bits), 7306 bytes captured (58448 bits)
Ethernet II, Src: PcsCompu_24:4c:de (08:00:27:24:4c:de), Dst: PcsCompu_d2:8d:71 (08:00:27:d2:8d:71)
Internet Protocol Version 4, Src: 10.0.2.8, Dst: 10.0.2.5
Transmission Control Protocol, Src Port: 12345, Dst Port: 51168, Seq: 1, Ack: 14, Len: 7240
Data (7240 bytes)
Data: 40454c4c4f2c20574f524c442148454c4c4f2c20574f524c...
[Length: 7240]

שורה 9:



No.	Time	Source	Destination	Protocol	Length	Info
3	0.000190	10.0.2.5	10.0.2.8	TCP	74	51168 → 12345 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=...
4	0.000211	10.0.2.8	10.0.2.5	TCP	74	12345 → 51168 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460...
5	0.000379	10.0.2.5	10.0.2.8	TCP	66	51168 → 12345 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=3052082...
6	0.000465	10.0.2.5	10.0.2.8	TCP	79	51168 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=13 TSval=3...
7	0.000471	10.0.2.8	10.0.2.5	TCP	66	12345 → 51168 [ACK] Seq=1 Ack=14 Win=65152 Len=0 TSval=321335...
8	0.000664	10.0.2.8	10.0.2.5	TCP	7306	12345 → 51168 [PSH, ACK] Seq=1 Ack=14 Win=65152 Len=7240 TSva...
9	0.000723	10.0.2.8	10.0.2.5	TCP	5826	12345 → 51168 [PSH, ACK] Seq=7241 Ack=14 Win=65152 Len=5760 T...
10	0.000912	10.0.2.5	10.0.2.8	TCP	66	51168 → 12345 [ACK] Seq=14 Ack=7241 Win=60672 Len=0 TSval=305...
11	0.000912	10.0.2.5	10.0.2.8	TCP	66	51168 → 12345 [ACK] Seq=14 Ack=13001 Win=56704 Len=0 TSval=30...
12	0.001059	10.0.2.5	10.0.2.8	TCP	66	51168 → 12345 [RST, ACK] Seq=14 Ack=13001 Win=64128 Len=0 TSV...

Frame 9: 5826 bytes on wire (46608 bits), 5826 bytes captured (46608 bits)
Ethernet II, Src: PcsCompu_24:4c:de (08:00:27:24:4c:de), Dst: PcsCompu_d2:8d:71 (08:00:27:d2:8d:71)
Internet Protocol Version 4, Src: 10.0.2.8, Dst: 10.0.2.5
Transmission Control Protocol, Src Port: 12345, Dst Port: 51168, Seq: 7241, Ack: 14, Len: 5760
Data (5760 bytes)
Data: 2140454c4c4f2c20574f524c442148454c4c4f2c20574f52...
[Length: 5760]

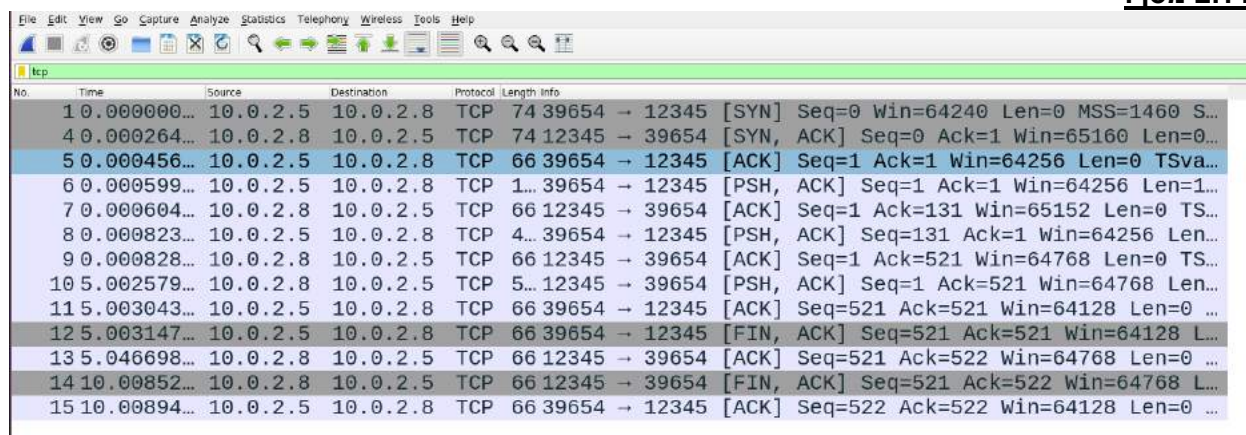
גרסה 4:

כעת, בצד השרת נוספה פקודת "שינה" מיד לפני קריאה מהבאפר בכל פעם.
בצד הלקוח אנו שולחים את ההודעה בעלת 13 הבתים אך מכפילים אותה כפול 10, כלומר שולחים בפועל 130 בתים. ומבצעים זאת 4 פעמים בזאת אחר זו. (סה"כ 520 בתים של מידע שיש להעביר)

*הציפייה שלנו:

פרוטוקול TCP ישלח ללא דיחוי את החבילה הראשונה המכילה 130 בתים. בזמן שימתין לקבלת ACK על החבילה, הוא יספיק לאגד את 390 הבתים הנוותרים וישלח אותם בחבילה אחת נוספת אל השרת. השרת יישן 5 שניות ועד שהוא מתעורר, הצליחו להגיע 520 בתים לתוך הבאפר. לכן כאשר הוא יעשה קריאה מהבאפר (שהפעם ניתן לקרוא 1024 בתים), באמת יצליח לקרוא 520 בתים. ימיר אותם לאותיות גדולות וישלח חבילה אחת ללקוח.

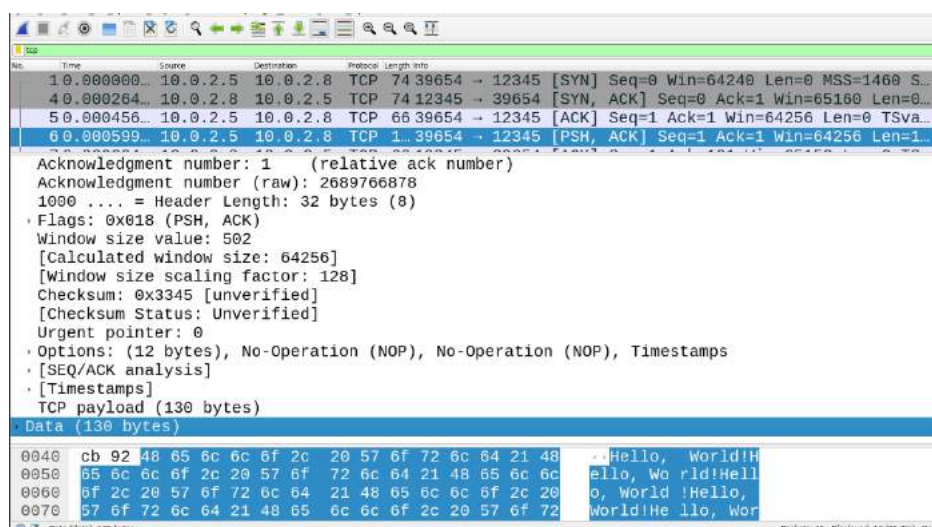
צילום מסך:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.2.5	10.0.2.8	TCP	74	39654 → 12345 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 S...
4	0.000264	10.0.2.8	10.0.2.5	TCP	74	12345 → 39654 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0...
5	0.000456	10.0.2.5	10.0.2.8	TCP	66	39654 → 12345 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSva...
6	0.000599	10.0.2.5	10.0.2.8	TCP	1...	39654 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=1...
7	0.000604	10.0.2.8	10.0.2.5	TCP	66	12345 → 39654 [ACK] Seq=1 Ack=131 Win=65152 Len=0 TS...
8	0.000823	10.0.2.5	10.0.2.8	TCP	4...	39654 → 12345 [PSH, ACK] Seq=131 Ack=1 Win=64256 Len...
9	0.000828	10.0.2.8	10.0.2.5	TCP	66	12345 → 39654 [ACK] Seq=1 Ack=521 Win=64768 Len=0 TS...
10	5.002579	10.0.2.8	10.0.2.5	TCP	5...	12345 → 39654 [PSH, ACK] Seq=1 Ack=521 Win=64768 Len...
11	5.003043	10.0.2.5	10.0.2.8	TCP	66	39654 → 12345 [ACK] Seq=521 Ack=521 Win=64128 Len=0 ...
12	5.003147	10.0.2.5	10.0.2.8	TCP	66	39654 → 12345 [FIN, ACK] Seq=521 Ack=521 Win=64128 L...
13	5.046698	10.0.2.8	10.0.2.5	TCP	66	12345 → 39654 [ACK] Seq=521 Ack=522 Win=64768 Len=0 ...
14	10.00852	10.0.2.8	10.0.2.5	TCP	66	12345 → 39654 [FIN, ACK] Seq=521 Ack=522 Win=64768 L...
15	10.00894	10.0.2.5	10.0.2.8	TCP	66	39654 → 12345 [ACK] Seq=522 Ack=522 Win=64128 Len=0 ...

נתבונן בשורות המעניינות: 6, 8, 10
שורה 6 הלקוח שולח 130 בתים לשרת.
שורה 8 הלקוח שולח 390 בתים לשרת וזאת מכיוון שבזמן שחיכה לקבלת הACK בשורה 7 הצליח לאגד עוד בתים לחבילה הבאה.
שורה 10 השרת יישן 5 שניות (ניתן לראות את ההבדל הזמנים בין שורה 9 לבין שורה 10) ושולח ללקוח חבילה בגודל 520 בתים.

כעת נראה שלושה צילומי מסך נבחרים של ההודעות שנשלחו:
שורה 6:



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.2.5	10.0.2.8	TCP	74	39654 → 12345 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 S...
4	0.000264	10.0.2.8	10.0.2.5	TCP	74	12345 → 39654 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0...
5	0.000456	10.0.2.5	10.0.2.8	TCP	66	39654 → 12345 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSva...
6	0.000599	10.0.2.5	10.0.2.8	TCP	1...	39654 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=64256 Len=1...

Acknowledgment number: 1 (relative ack number)
Acknowledgment number (raw): 2689766878
1000 = Header Length: 32 bytes (8)
Flags: 0x018 (PSH, ACK)
Window size value: 502
[Calculated window size: 64256]
[Window size scaling factor: 128]
Checksum: 0x3345 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
[SEQ/ACK analysis]
[Timestamps]
TCP payload (130 bytes)
Data (130 bytes)

Offset	Hex	ASCII
0040	cb 92 48 65 6c 6c 6f 2c 20 57 6f 72 6c 64 21 48	..Hello, World!H
0050	65 6c 6c 6f 2c 20 57 6f 72 6c 64 21 48 65 6c 6c	ello, Wo rld!Hell
0060	6f 2c 20 57 6f 72 6c 64 21 48 65 6c 6c 6f 2c 20	o, World !Hello,
0070	57 6f 72 6c 64 21 48 65 6c 6c 6f 2c 20 57 6f 72	World!He llo, Wor

שורה 8:

Wireshark packet capture details for packet 8:

- Frame 8:** 10.0.2.8 → 10.0.2.5 TCP 66.12345 → 39654 [ACK] Seq=1 Ack=521 Win=64768 Len=0 TS...
- Details:**
 - Acknowledgment number: 1 (relative ack number)
 - Acknowledgment number (raw): 2689766878
 - 1000 ... = Header Length: 32 bytes (8)
 - Flags: 0x018 (PSH, ACK)
 - Window size value: 502
 - [Calculated window size: 64256]
 - [Window size scaling factor: 128]
 - Checksum: 0xeb77 [unverified]
 - [Checksum Status: Unverified]
 - Urgent pointer: 0
 - Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
 - [SEQ/ACK analysis]
 - [Timestamps]
 - TCP payload (390 bytes)
 - Data (390 bytes)
- Data:**

```

0040  cb 93 48 65 6c 6c 6f 2c 20 57 6f 72 6c 64 21 48  .Hello, World!H
0050  65 6c 6c 6f 2c 20 57 6f 72 6c 64 21 48 65 6c 6c  ello, Wo rld!Hell
0060  6f 2c 20 57 6f 72 6c 64 21 48 65 6c 6c 6f 2c 20  o, World !Hello,
0070  57 6f 72 6c 64 21 48 65 6c 6c 6f 2c 20 57 6f 72  world!He llo, Wor

```

שורה 10:

Wireshark packet capture details for packet 10:

- Frame 10:** 10.0.2.8 → 10.0.2.5 TCP 66.12345 → 39654 [ACK] Seq=1 Ack=521 Win=64768 Len=0 TS...
- Details:**
 - Acknowledgment number: 521 (relative ack number)
 - Acknowledgment number (raw): 3050713326
 - 1000 ... = Header Length: 32 bytes (8)
 - Flags: 0x018 (PSH, ACK)
 - Window size value: 506
 - [Calculated window size: 64768]
 - [Window size scaling factor: 128]
 - Checksum: 0x1a3b [unverified]
 - [Checksum Status: Unverified]
 - Urgent pointer: 0
 - Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
 - [SEQ/ACK analysis]
 - [Timestamps]
 - TCP payload (520 bytes)
 - Data (520 bytes)
- Data:**

```

0040  54 2f 48 45 4c 4c 4f 2c 20 57 4f 52 4c 44 21 48  T/HELLO, WORLD!H
0050  45 4c 4c 4f 2c 20 57 4f 52 4c 44 21 48 45 4c 4c  ELLO, WO RLD!HELL
0060  4f 2c 20 57 4f 52 4c 44 21 48 45 4c 4c 4f 2c 20  o, WORLD !HELLO,
0070  57 4f 52 4c 44 21 48 45 4c 4c 4f 2c 20 57 4f 52  WORLD!HE LLO, WOR

```


חלק ב'

נתבונן בכמה הרצות של התרגיל בהן הדפדפן שולח בקשות שונות לשרת.

הרצה 1:

קלט דפדפן localhost:12346/

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	:::1	:::1	TCP	94	56988 → 12346 [SYN] Seq=0 Win=65476 Len...
2	0.000005	:::1	:::1	TCP	74	12346 → 56988 [RST, ACK] Seq=1 Ack=1 Wi...
3	0.000076	127.0.0.1	127.0.0.1	TCP	74	46654 → 12346 [SYN] Seq=0 Win=65495 Len...
4	0.000082	127.0.0.1	127.0.0.1	TCP	74	12346 → 46654 [SYN, ACK] Seq=0 Ack=1 Wi...
5	0.000088	127.0.0.1	127.0.0.1	TCP	66	46654 → 12346 [ACK] Seq=1 Ack=1 Win=655...
6	0.000209	:::1	:::1	TCP	94	56992 → 12346 [SYN] Seq=0 Win=65476 Len...
7	0.000211	:::1	:::1	TCP	74	12346 → 56992 [RST, ACK] Seq=1 Ack=1 Wi...
8	0.000234	127.0.0.1	127.0.0.1	TCP	74	46658 → 12346 [SYN] Seq=0 Win=65495 Len...
9	0.000237	127.0.0.1	127.0.0.1	TCP	74	12346 → 46658 [SYN, ACK] Seq=0 Ack=1 Wi...
10	0.000241	127.0.0.1	127.0.0.1	TCP	66	46658 → 12346 [ACK] Seq=1 Ack=1 Win=655...
11	0.0007818	127.0.0.1	127.0.0.1	HTTP	599	GET / HTTP/1.1
12	0.0007832	127.0.0.1	127.0.0.1	TCP	66	12346 → 46654 [ACK] Seq=1 Ack=534 Win=6...
13	0.0008009	127.0.0.1	127.0.0.1	HTTP	238	HTTP/1.1 200 OK
14	0.0008048	127.0.0.1	127.0.0.1	TCP	66	46654 → 12346 [ACK] Seq=534 Ack=173 Win...
15	0.159637	127.0.0.1	127.0.0.1	HTTP	512	GET /favicon.ico HTTP/1.1
16	0.159649	127.0.0.1	127.0.0.1	TCP	66	12346 → 46654 [ACK] Seq=173 Ack=980 Win...
17	0.159853	127.0.0.1	127.0.0.1	HTTP	23...	HTTP/1.1 200 OK
18	0.159945	127.0.0.1	127.0.0.1	TCP	66	46654 → 12346 [ACK] Seq=980 Ack=2476 Wi...
19	1.160836	127.0.0.1	127.0.0.1	TCP	66	12346 → 46654 [FIN, ACK] Seq=2476 Ack=9...
20	1.208415	127.0.0.1	127.0.0.1	TCP	66	46654 → 12346 [ACK] Seq=980 Ack=2477 Wi...
21	2.162990	127.0.0.1	127.0.0.1	TCP	66	12346 → 46658 [FIN, ACK] Seq=1 Ack=1 Wi...
22	2.163943	127.0.0.1	127.0.0.1	TCP	66	46658 → 12346 [ACK] Seq=1 Ack=2 Win=655...

*נתעלם משורות 1-2 וגם 6-7

ניתן לראות כי הדפדפן שולח שתי בקשות התחברות. אחת מפורט 46654 (שורה 3) והשניה מפורט 46658 (שורה 8).

מכיוון והשרת שלנו מטפל כל פעם בלקוח אחד ניתן להבין כי החבילות שנשלחו היו בין השרת לבין הפורט הראשון. (להלן פורט א') נתבונן בחבילות שנשלחו:

שורה 11 – אומנם לא רואים מאיפה לאן נשלחה החבילה כיוון שHTTP מסתיר זאת כאן אבל אם נתבונן בשורה מספר 12 נראה כי השרת היושב בפורט 12346 שולח ACK לפורט א' עם ACK NUMBER = 534 (רלטיבי כמובן), כלומר נשלח מידע מסוים. בנוסף ניתן לראות כי בשורה זו כתוב כי הדפדפן מעוניין בקובץ 'index.html'.
שורה 13 – השרת מעבד את המידע שקיבל בהודעה הקודמת ולכן החליט לשלוח ללקוח את הקובץ הרלוונטי שהוא index.html. תמונה להמחשה:

13	0.0008009	127.0.0.1	127.0.0.1	HTTP	238	HTTP/1.1 200 OK
Hypertext Transfer Protocol						
HTTP/1.1 200 OK\r\n						
Connection: keep-alive\r\n						
Content-Length: 108\r\n						
\r\n						
[HTTP response 1/2]						
0040	ef 4f	48 54 54 50 2f 31	2e 31	20 32 30 30 20 4f	HTTP/1.1 200 OK	
0050	4b 0d	0a 43 6f 6e 6e 65	63 74 69 6f 6e 3a	20 6b	K: Connection: k	
0060	65 65	70 2d 61 6c 69 76	65 0d 0a 43 6f 6e	74 65	eep-alive. Conte	
0070	6e 74	2d 4c 65 6e 67 74	68 3a 20 31 30 38	0d 0a	nt-Lengt h: 108.	
0080	0d 0a	3c 68 74 6d 6c 3e	0d 0a 09 3c 68 65	61 64	<html> ...<head	
0090	3e 0d	0a 09 09 3c 74 69	74 6c 65 3e 4d 79	20 77	>...<title>My w	
00a0	65 62	20 70 61 67 65 3c	2f 74 69 74 6c 65	3e 0d	eb page< /title>.	
00b0	0a 09	3c 2f 68 65 61 64	3e 0d 0a 09 3c 62	6f 64	..</head> ...<bod	
00c0	79 3e	0d 0a 09 09 3c 62	3e 3c 75 3e 68 65	6c 6c	y>...<u>hell	
00d0	6f 3c	2f 75 3e 3c 2f 62	3e 0d 0a 09 3c 2f	62 6f	o</u> ...</bo	
00e0	64 79	3e 0d 0a 3c 2f 68	74 6d 6c 3e 0d 0a		dy>...</h tml>..	

בירוק מסומן headern של HTTP הנשלח לדפדפן ובאדום ניתן לראות את הDATA הנשלח. המידע הזה הוא בדיוק המידע שנמצא ב index.html.

בשורה 15 – פורט א' מבצע בקשה נוספת לקבל את favicon.ico / ועל כן השרת עונה ACK.

שורה 17 – השרת שולח את המידע הידוע לו על הקובץ.

שורה 19 – השרת מסיים את התקשורת ומנתק את פורט א' וזאת מכיוון ולקוח זה לא שלח לו בקשה במשך 1

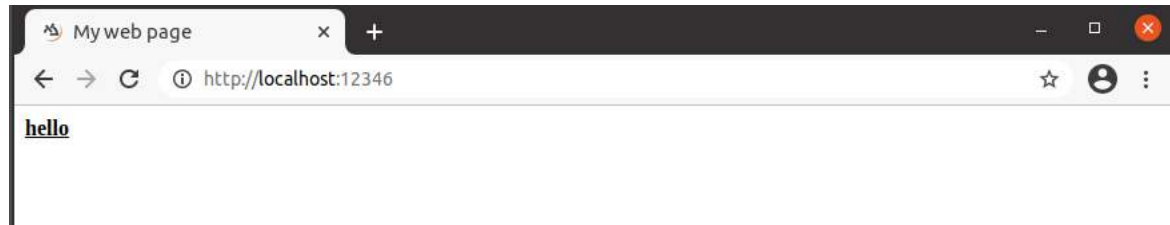
שניות. אפשר ממש לראות זאת ליד מספר השורה מצד שמאל, שהרי הבקשה האחרונה התבצעה בשורה 15

בזמן 0.159 והזמן בשורה 19 הינו 1.16

שורה 21 – כעת השרת מתפנה לטפל בחיבור הבא שהגיע מזמן מפורט 46658 אך ניתן לראות כי גם כאן השרת

המתין 1 שניות בין שורה 19 לעכשיו, ומכיוון שלא קיבל כל בקשה, מנתק גם כאן את החיבור.

תמונת פלט:



הרצה 2:

קלט דפדפן localhost:12346/a

No.	Time	Source	Destination	Protocol	Length	Info
3	0.000049	127.0.0.1	127.0.0.1	TCP	74	46688 → 12346 [SYN] Seq=0 Win=65495 Len...
4	0.000054	127.0.0.1	127.0.0.1	TCP	74	12346 → 46688 [SYN, ACK] Seq=0 Ack=1 Wi...
5	0.000059	127.0.0.1	127.0.0.1	TCP	66	46688 → 12346 [ACK] Seq=1 Ack=1 Win=655...
8	0.000217	127.0.0.1	127.0.0.1	TCP	74	46692 → 12346 [SYN] Seq=0 Win=65495 Len...
9	0.000222	127.0.0.1	127.0.0.1	TCP	74	12346 → 46692 [SYN, ACK] Seq=0 Ack=1 Wi...
10	0.000227	127.0.0.1	127.0.0.1	TCP	66	46692 → 12346 [ACK] Seq=1 Ack=1 Win=655...
11	0.007205	127.0.0.1	127.0.0.1	HTTP	600	GET /a HTTP/1.1
12	0.007223	127.0.0.1	127.0.0.1	TCP	66	12346 → 46688 [ACK] Seq=1 Ack=535 Win=6...
13	0.007375	127.0.0.1	127.0.0.1	TCP	109	12346 → 46688 [PSH, ACK] Seq=1 Ack=535 Wi...
14	0.007388	127.0.0.1	127.0.0.1	TCP	66	46688 → 12346 [ACK] Seq=535 Ack=44 Win=...
15	0.007439	127.0.0.1	127.0.0.1	TCP	66	12346 → 46688 [FIN, ACK] Seq=44 Ack=535...
16	0.008056	127.0.0.1	127.0.0.1	TCP	66	46688 → 12346 [FIN, ACK] Seq=535 Ack=45...
17	0.008065	127.0.0.1	127.0.0.1	TCP	66	12346 → 46688 [ACK] Seq=45 Ack=536 Win=...
22	0.100026	127.0.0.1	127.0.0.1	TCP	74	46696 → 12346 [SYN] Seq=0 Win=65495 Len...
23	0.100034	127.0.0.1	127.0.0.1	TCP	74	12346 → 46696 [SYN, ACK] Seq=0 Ack=1 Wi...
24	0.100042	127.0.0.1	127.0.0.1	TCP	66	46696 → 12346 [ACK] Seq=1 Ack=1 Win=655...
25	1.008154	127.0.0.1	127.0.0.1	TCP	66	12346 → 46692 [FIN, ACK] Seq=1 Ack=1 Wi...
26	1.011574	127.0.0.1	127.0.0.1	TCP	66	46692 → 12346 [ACK] Seq=1 Ack=2 Win=655...
27	2.012471	127.0.0.1	127.0.0.1	TCP	66	12346 → 46696 [FIN, ACK] Seq=1 Ack=1 Wi...
28	2.015474	127.0.0.1	127.0.0.1	TCP	66	46696 → 12346 [ACK] Seq=1 Ack=2 Win=655...

הדפדפן שולח 3 בקשות התחברות. שורה 3 פורט א' 46688, שורה 8 פורט ב' 46692, שורה 22 פורט ג' 46696. גם כאן, השרת מטפל בלקוחות לפי סדר הגעתם.

שורה 11 – פורט א' שולח בקשה עבור הקובץ /a. קובץ זה אינו קיים.

שורה 13 – השרת שולח לפורט א' כי הקובץ לא נמצא. תמונה להמחשה:

```
GET /a HTTP/1.1
Host: localhost:12346
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8

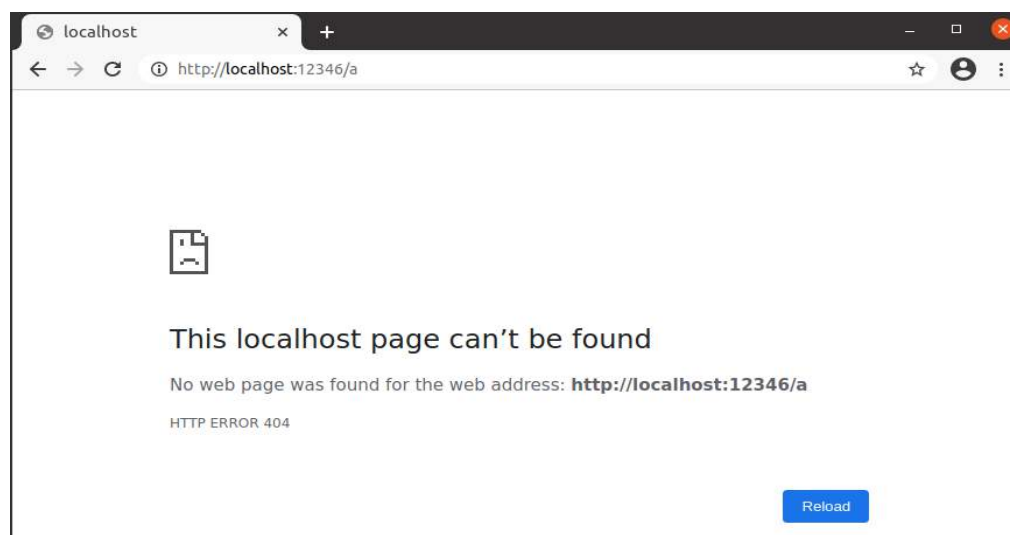
HTTP/1.1 404 Not Found
Connection: close
```

ולכן סוגר את החיבור בשורה 15.

שורה 25 – השרת סוגר את החיבור של פורט ב' כיוון שמאז שורה 15, כאשר סגר את החיבור הקודם ועבר לחיבור הבא, המתין 1 שניות ולא קיבל הודעה.

שורה 27 – השרת סוגר את החיבור של פורט ג' כי המתין שניה בין שורה 26 ושורה 27 מבלי לקבל מידע חדש.

תמונת הפלט:



הרצה 3:

קלט דפדפן localhost:12346/a/b/ref.html

No.	Time	Source	Destination	Protocol	Length	Info
3	0.000090	127.0.0.1	127.0.0.1	TCP	74	46702 → 12346 [SYN, Seq=0 Win=65495 Len...
4	0.000143	127.0.0.1	127.0.0.1	TCP	74	12346 → 46702 [SYN, ACK] Seq=0 Ack=1 Wi...
5	0.000150	127.0.0.1	127.0.0.1	TCP	66	46702 → 12346 [ACK] Seq=1 Ack=1 Win=655...
8	0.000244	127.0.0.1	127.0.0.1	TCP	74	46706 → 12346 [SYN] Seq=0 Win=65495 Len...
9	0.000249	127.0.0.1	127.0.0.1	TCP	74	12346 → 46706 [SYN, ACK] Seq=0 Ack=1 Wi...
10	0.000254	127.0.0.1	127.0.0.1	TCP	66	46706 → 12346 [ACK] Seq=1 Ack=1 Win=655...
11	0.016514	127.0.0.1	127.0.0.1	HTTP	611	GET /a/b/ref.html HTTP/1.1
12	0.016539	127.0.0.1	127.0.0.1	TCP	66	12346 → 46702 [ACK] Seq=1 Ack=546 Win=6...
13	0.017318	127.0.0.1	127.0.0.1	HTTP	791	HTTP/1.1 200 OK
14	0.017337	127.0.0.1	127.0.0.1	TCP	66	46702 → 12346 [ACK] Seq=546 Ack=726 Win...
17	0.126541	127.0.0.1	127.0.0.1	HTTP	524	GET /a/oh_no.jpg HTTP/1.1
18	0.126554	127.0.0.1	127.0.0.1	TCP	66	12346 → 46702 [ACK] Seq=726 Ack=1004 Wi...
19	0.127849	127.0.0.1	127.0.0.1	TCP	32...	12346 → 46702 [ACK] Seq=726 Ack=1004 Wi...
20	0.127870	127.0.0.1	127.0.0.1	TCP	66	46702 → 12346 [ACK] Seq=1004 Ack=33494 ...
21	0.127884	127.0.0.1	127.0.0.1	TCP	32...	12346 → 46702 [PSH, ACK] Seq=33494 Ack=...
22	0.127890	127.0.0.1	127.0.0.1	TCP	66	46702 → 12346 [ACK] Seq=1004 Ack=66262 ...
23	0.127901	127.0.0.1	127.0.0.1	HTTP	33...	HTTP/1.1 200 OK (JPEG JFIF image)
24	0.132480	127.0.0.1	127.0.0.1	TCP	66	46702 → 12346 [ACK] Seq=1004 Ack=69571 ...
25	0.134067	127.0.0.1	127.0.0.1	HTTP	526	GET /a/b/oh_no.jpg HTTP/1.1
26	0.134076	127.0.0.1	127.0.0.1	TCP	66	12346 → 46702 [ACK] Seq=69571 Ack=1464 ...
27	0.135568	127.0.0.1	127.0.0.1	TCP	32...	12346 → 46702 [ACK] Seq=69571 Ack=1464 ...
28	0.135594	127.0.0.1	127.0.0.1	TCP	32...	[TCP Window Full] 12346 → 46702 [PSH, A...
29	0.136674	127.0.0.1	127.0.0.1	TCP	66	46702 → 12346 [ACK] Seq=1464 Ack=135107...
30	0.136683	127.0.0.1	127.0.0.1	HTTP	23...	HTTP/1.1 200 OK (JPEG JFIF image)
31	0.136689	127.0.0.1	127.0.0.1	TCP	66	46702 → 12346 [ACK] Seq=1464 Ack=158847...

כאן נפתחו שני חיבורים מול השרת. בתמונה זו לא נצליח להראות את כל המידע העובר אבל ננסה להסביר זאת. פורט א' 46702 הוא הפורט הראשון ומולו השרת מתנהל. שורה 11 – הלקוח מבקש את הקובץ a/b/ref.html ובשורה 13 השרת שולח לו את הקובץ. תמונה להמחשה:

```
GET /a/b/ref.html HTTP/1.1
Host: localhost:12346
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.6
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8
exchange;v=b3;q=0.0
Sec-Fetch-Site: none
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8

HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 661

<html>
  <head>
    <title>My web page</title>
  </head>
  <body>
    <h2>Amazing:</h2>

    
    
    
    
    
    
    
    
    
    
    
    
  </body>
</html>
```

כעת בשורה 17, הלקוח לא סוגר את החיבור מול השרת אלא מבקש את הלינק הראשון מתוך רשימת הלינקים שהשרת שלח לו בשורה 13. ואכן, בשורה 23, ניתן לראות כי השרת שלח את הקובץ הרלוונטי לקישור המבוקש.

נשים לב לדבר מעניין שקרה כאן וזה פיצול החבילה שהשרת שלח לכמה סגמנטים היות והחבילה גדולה מדי כלומר, בשורה 19 נשלח סגמנט בגודל 32768 בתים וכל בשורה 21, ובשורה 23 נשלחו יתר הבתים, וכעת wireshark מעדכן אותנו כי כל הבתים הגיעו הבהצלחה לידען

23	0.127901	127.0.0.1	127.0.0.1	HTTP	3375 HTTP/1.1 200 OK (JPEG JFIF image)
24	0.132480	127.0.0.1	127.0.0.1	TCP	66 46702 → 12346 [ACK] Seq=1004 Ack=69571 ...
<ul style="list-style-type: none"> Frame 23: 3375 bytes on wire (27000 bits), 3375 bytes captured (27000 bits) Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00) Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 Transmission Control Protocol, Src Port: 12346, Dst Port: 46702, Seq: 66202, Ack: 1004, Len: 3309 [3 Reassembled TCP Segments (68845 bytes): #19(32768), #21(32768), #23(3309)] Hypertext Transfer Protocol <ul style="list-style-type: none"> HTTP/1.1 200 OK\r\n Connection: keep-alive\r\n Content-Length: 68779\r\n \r\n 					
[HTTP response 2/8]					

דבר מעניין נוסף שניתן לשים אליו לב זה כי בשורה 28 השרת שולח ללקוח חבילה והלקוח מודיע כי חלון הקבלה שלו כעת מלא. (אך כמובן שהוא יתרוקן די מהר כי הדפדפן עושה recv לחבילה עד להגעת החבילה הבאה בשורה 30)

80	0.176277	127.0.0.1	127.0.0.1	HTTP	3428 HTTP/1.1 200 OK (JPEG JFIF image)
81	0.176278	127.0.0.1	127.0.0.1	TCP	66 46702 → 12346 [ACK] Seq=3304 Ack=475089 Win=1244288 Len=0 TSv...
82	0.181188	127.0.0.1	127.0.0.1	HTTP	527 GET /a/b/oh_no5.jpg HTTP/1.1
83	0.182182	127.0.0.1	127.0.0.1	TCP	65549 12346 → 46702 [ACK] Seq=475089 Ack=3765 Win=65536 Len=65483 T...
84	0.182198	127.0.0.1	127.0.0.1	TCP	66 46702 → 12346 [ACK] Seq=3765 Ack=540572 Win=1375232 Len=0 TSv...
85	0.182210	127.0.0.1	127.0.0.1	HTTP	23859 HTTP/1.1 200 OK (JPEG JFIF image)
86	0.182216	127.0.0.1	127.0.0.1	TCP	66 46702 → 12346 [ACK] Seq=3765 Ack=564365 Win=1506176 Len=0 TSv...
87	1.182873	127.0.0.1	127.0.0.1	TCP	66 12346 → 46702 [FIN, ACK] Seq=564365 Ack=3765 Win=65536 Len=0 ...
94	1.225399	127.0.0.1	127.0.0.1	TCP	66 46702 → 12346 [ACK] Seq=3765 Ack=564366 Win=1506176 Len=0 TSv...

שיח זה בין הלקוח והשרת ממשיך עד לקבלת כל התמונות בצד הלקוח, ולכן הלקוח לא שולח יותר בקשות ולבסוף בשורה 87 ניתן לראות כי השרת המתין 1 שניות וסגר את החיבור.
*רק נדגיש כי כל הבקשות נעשו ע"י אותו הפורט ולא על ידי לקוחות שונים.
תצוגת הדפדפן תהיה:

