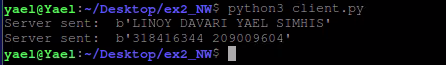
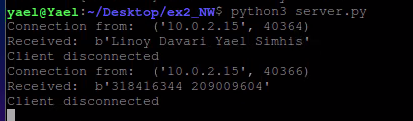
**דו"ח הגשה – תרגיל 2**

**חלק א':**

תחילה, הרצנו את קוד השרת והלקוח בפרוטוקול TCP מהמודל, באופן הבא:

 הרצת הלקוח (מכונה וירטואלית ב')

 הרצת השרת (מכונה וירטואלית א')

הרצנו את קוד השרת ממכונה וירטואלית א', ואת קוד הלקוח ממכונה וירטואלית ב'. למעשה הלקוח מבצע 2 פניות שונות לשרת - נשים לב שבהתחלה הלקוח פונה לשרת מפורט 45190, ובבקשה השנייה הוא פונה לשרת מפורט 45192.

בכל פניה, הלקוח מבצע חיבור של ממש לשרת, ואילו השרת (שהאזין לבקשות) מאשר את הבקשה, מתחבר ללקוח הרלוונטי (למשל, ממסד את החיבור בעזרת פקודת bind) ומטפל בבקשה. במקרה זה, זהו שרת echo ולכן הוא ידפיס את ההודעה המבוקשת – כפי שניתן לראות, הדפסנו תחילה את שמותינו ואחר כך את תעודות הזהות שלנו.

סך הכל היו שני חיבורים שונים, כאשר בכל חיבור הועברה מהלקוח לשרת בקשה אחת.

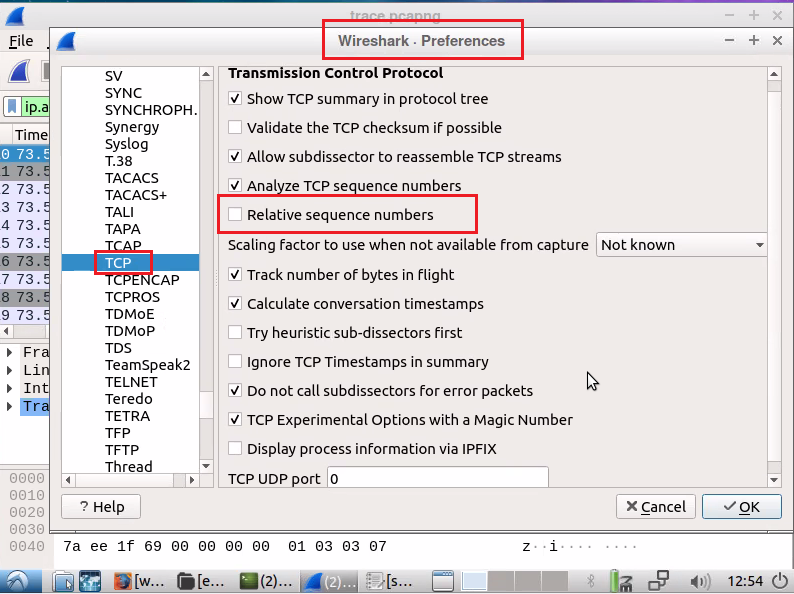
נשים לב שכתובת הIP של הלקוח היא – 10.0.2.15 כפי שניתן לראות בפלט שהתקבל מהשרת. בנוסף, כתובת הIP של השרת היא 10.0.2.4, ולפי כתובת זו ביצענו את הסינון.

באפליקציית wireshark המותקנת על מכונה וירטואלית א', סיננו את התעבורה שהתקבלה לפי הפילטר הבא:



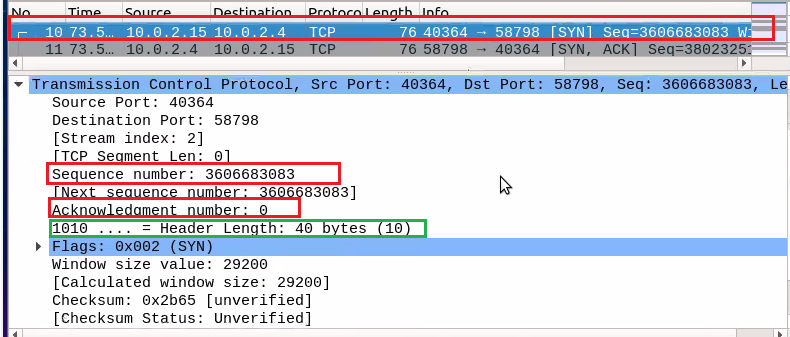
את הפאקטות שנתפסו שמרנו בקובץ trace.pcapng המצורף להגשה זו.

כפי שהתבקשנו, יש צורך בהתייחסות למספרים האמיתיים (של ערכי ה-sequence number וה-ack number). שמנו לב שבאופן דיפולטיבי, בתוכנת wireshark יש התייחסות לערכים היחסיים. שינינו זאת כך, דרך ממשק ה-Preferences:



כפי שלמדנו בהרצאות ובתרגולים, ה-**sequence number** מתאר את מס' ה-byte הראשון בחבילה הנשלחת וה-**ack. Number** מתאר את מס' ה-byte הבא בתור שהוא מצפה לקבל, כלומר קיבלנו מהשולח את ההודעה שעד byte זה.

כעת נביט בחבילה **הראשונה** שתפסנו בוויירשארק:

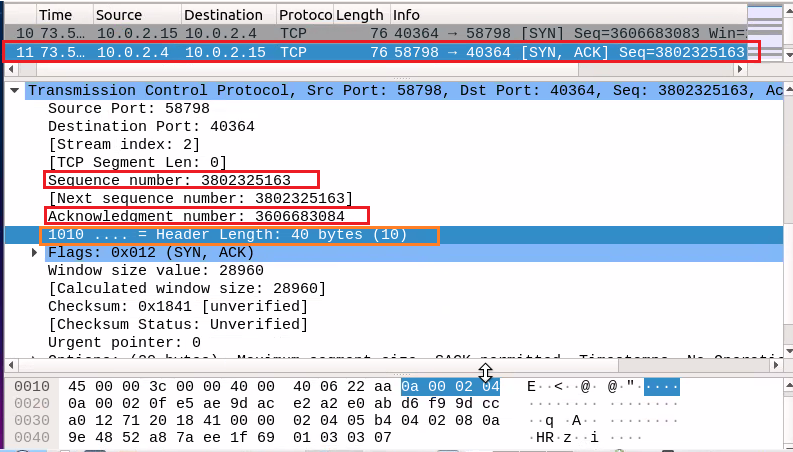


חבילה זו נשלחה מהלקוח לשרת, והיא למעשה מתארת את בקשת ההתחברות (פקודת connect בקוד הלקוח) של הלקוח אל השרת.

נשים לב שערך ה-ack number (ממורקר באדום) הוא 0, כי טרם התקבלו הודעות (זו ההודעה הראשונה). ערך ה-sequence number ממורקר באדום גם כן. תחילית ה-TCP, כפי שלמדנו בכיתה, מכיל את ה-sequence number ואת ה-ack number, ואורכו הוא 40 bytes, כפי שמופיע בצילום לעיל (בירוק).

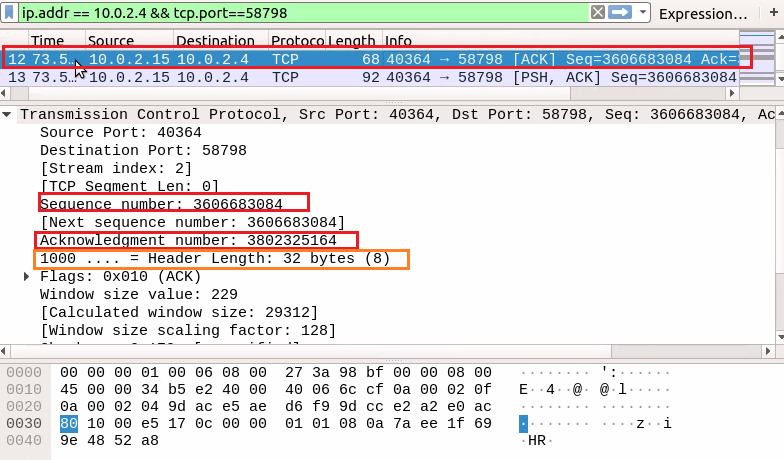
כעת נביט בחבילה **השנייה** שנתפסה:

חבילה זו נשלחה מהשרת אל הלקוח, והיא מתארת את פקודת ה-bind, כלומר את מיסוד ההתקשרות בין השרת ללקוח.



נשים לב לשינוי בערך ה-ack number לעומת החבילה הראשונה – כאן כבר הערך הוא לא 0, שכן כבר נשלחה הודעה מהלקוח לשרת קודם לחבילה זו. בנוסף, גם ערך ה-sequence number השתנה באופן טבעי, שכן נשלחה חבילה נוספת. ניתן לראות גם (בכתום) שערך תחילית ה-TCP נותר ללא שינוי.

כעת, נביט בחבילה **השלישית** שנתפסה:

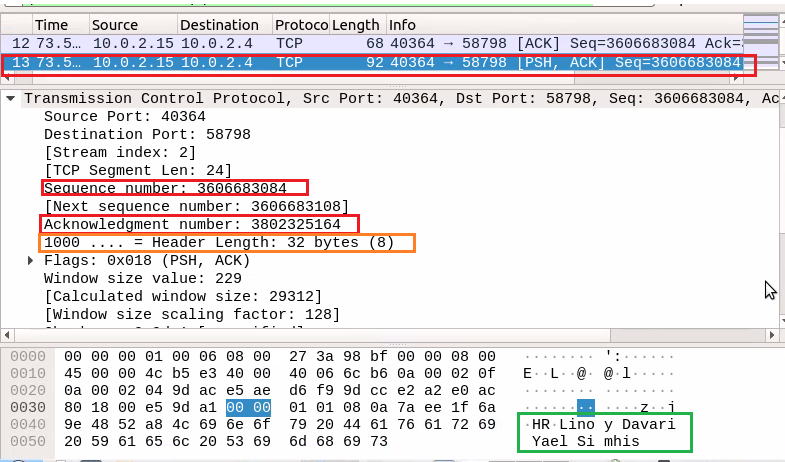


חבילה זו נשלחה מהלקוח לשרת ותפקידה היא "לאשרר" את החיבור שהשרת הסכים למסד בחבילה הקודמת. כלומר, שלב לחיצת הידיים הושלם – השרת והלקוח אישרו שניהם את החיבור. בהתאם למה שראינו בכיתה, נראה תהליך דומה גם בעת תהליך הניתוק של חיבור ה-TCP, בהמשך.

נשים לב כי ערך ה-sequence number **שונה** מערך זה שהיה בחבילות הקודמות. *הערה –* *ערך ה-sequence number בחבילות שיישלחו מהלקוח לשרת יהיה בעל "תחילית" שונה מערך ה-sequence number שבחבילות שיישלחו מהשרת ללקוח*. זה יתבטא גם בהמשך, עבור שאר החבילות.

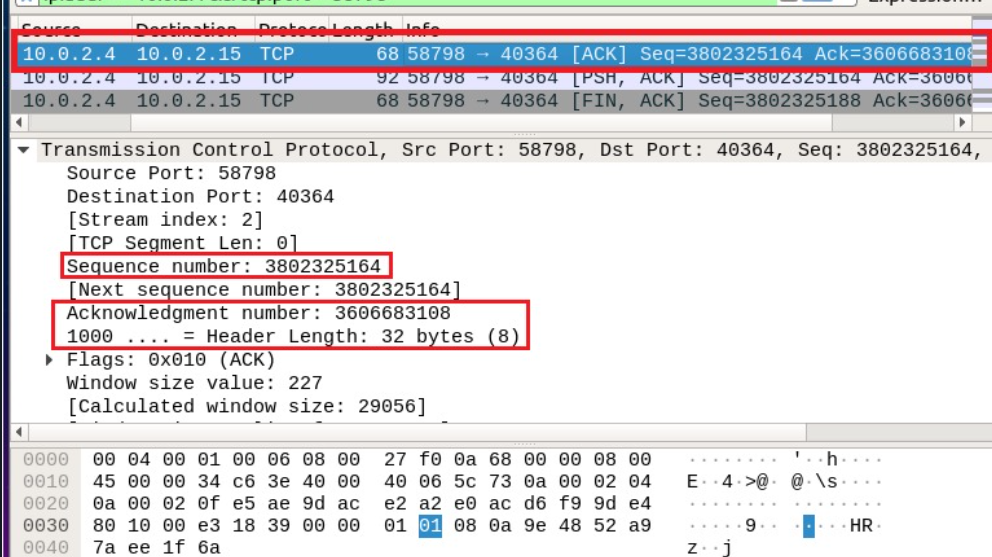
באופן דומה, גם ערך ה-ack number ואורך תחילית ה-TCP השתנו.

כעת, נביט בחבילה **הרביעית** שנתפסה:



חבילה זו גם נשלחה מהלקוח לשרת, והיא מכילה את המידע, כלומר את הבקשה עצמה. ניתן לראות את ה-data שנשלח במסגרת הירוקה לעיל. גם כאן השתנו (בהתאם להערה שציינו קודם לכן) ערכי ה-sequence number, ack number, ואורך תחילית ה-TCP נותר ללא שינוי (לעומת החבילה השלישית).

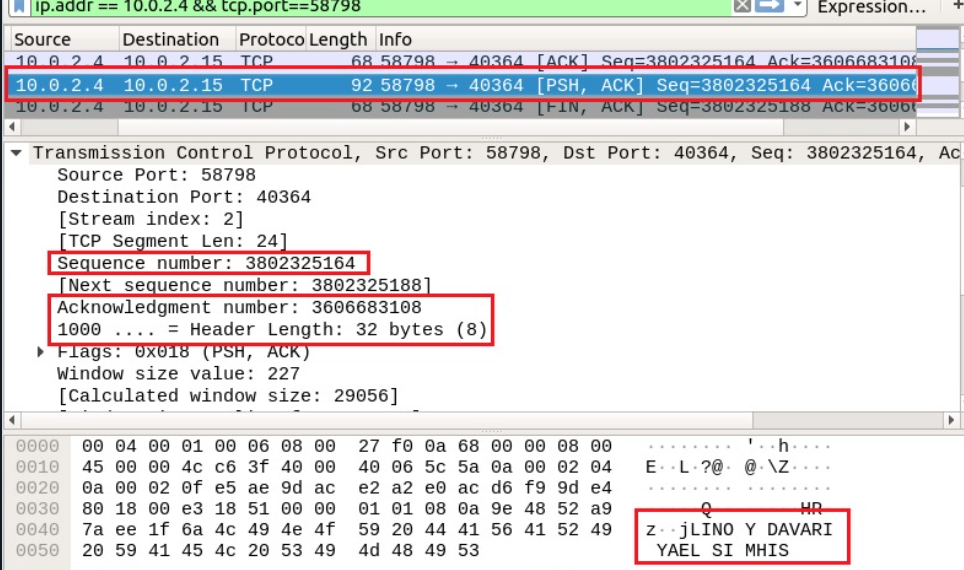
נביט כעת בחבילה **החמישית** שנתפסה:



חבילה זו נשלחה מהשרת אל הלקוח ותפקידה לאשר; כלומר השרת מכיר בכך שהלקוח שלח את הבקשה (בחבילה הקודמת, הרביעית) ומאשר שהיא הגיעה בשלמות אליו.

למעשה, זה תואם ללוגיקה שבקוד השרת, שכן הוא מקבל את הבקשה מהלקוח ע"י הפקודה client\_socket.recv(100). גם כאן רואים את השינוי שבערכי ה-sequence number, ה-ack number, אך אורך תחילית ה-TCP נותר ללא שינוי לעומת החבילה הקודמת (הרביעית).

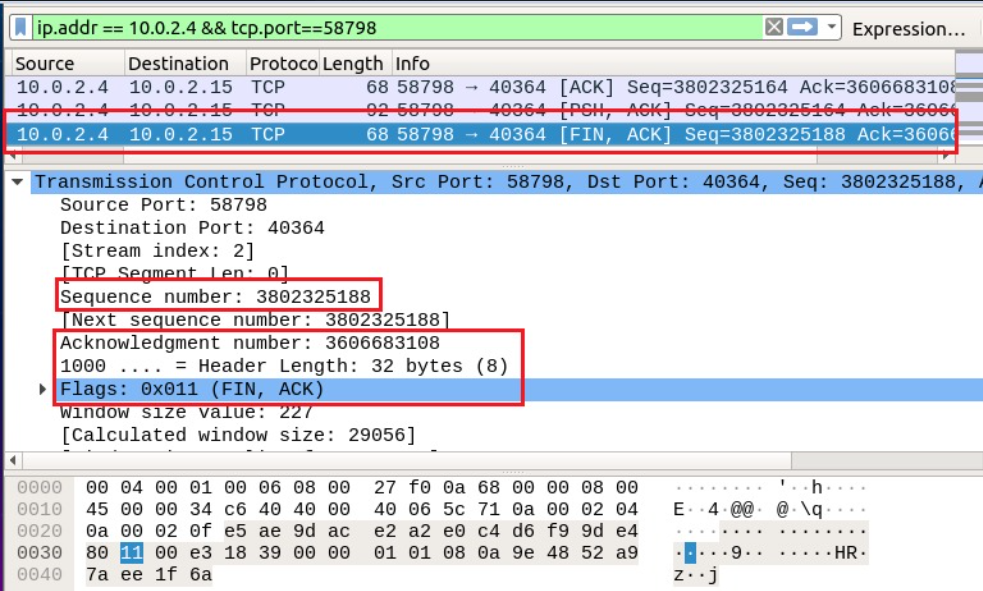
נביט כעת בחבילה **השישית** שנתפסה:



חבילה זו נשלחת גם היא מהשרת ללקוח, והיא מכילה את תשובת השרת בהתאם לבקשה. כאמור, זהו שרת echo, ולכן ניתן לראות במסגרת האדומה התחתונה שהשרת החזיר ללקוח את שמותינו כפי שנשלחו ע"י הלקוח.

נשים לב שלא חל שינוי לעומת החבילה הקודמת בערכי ה-ack number, ה-sequence number ובאורך תחילית ה-TCP, כפי שמודגש בשאר המסגרות האדומות לעיל.

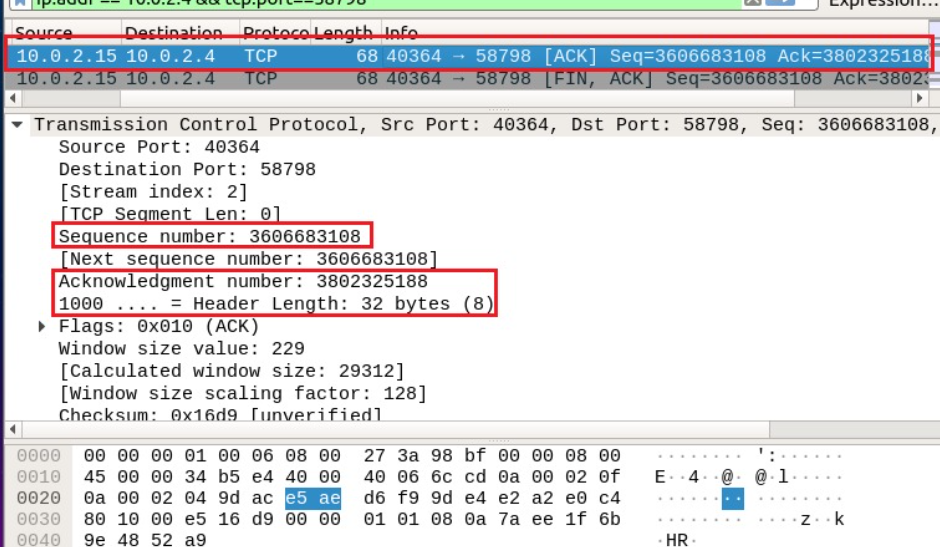
נביט בחבילה **השביעית** שנתפסה:



גם חבילה זו נשלחה מהשרת ללקוח. היא מתארת את העובדה שהשרת סוגר את החיבור שהיה לו עם הלקוח הזה, ולכן יש שינוי ב-FLAGS שנשלחו. בניגוד לחבילות הקודמות, כאן דולק גם הדגל FIN, שמציין את העובדה שהשרת רוצה לסגור את החיבור.

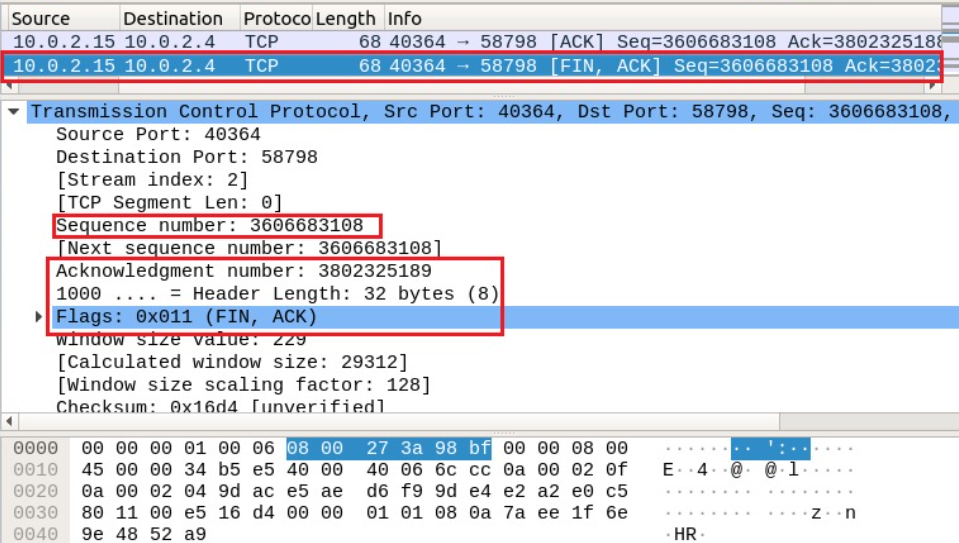
בנוסף, ערך ה-sequence number גדל, אך ערך ה-ack number נותר זהה (לעומת החבילה הקודמת, שכן לא התקבלה חבילה נוספת מהלקוח בין החבילה השישית והשביעית). אורך תחילית ה-TCP גם נותר זהה.

כעת, נביט בחבילה **השמינית** שנתפסה:



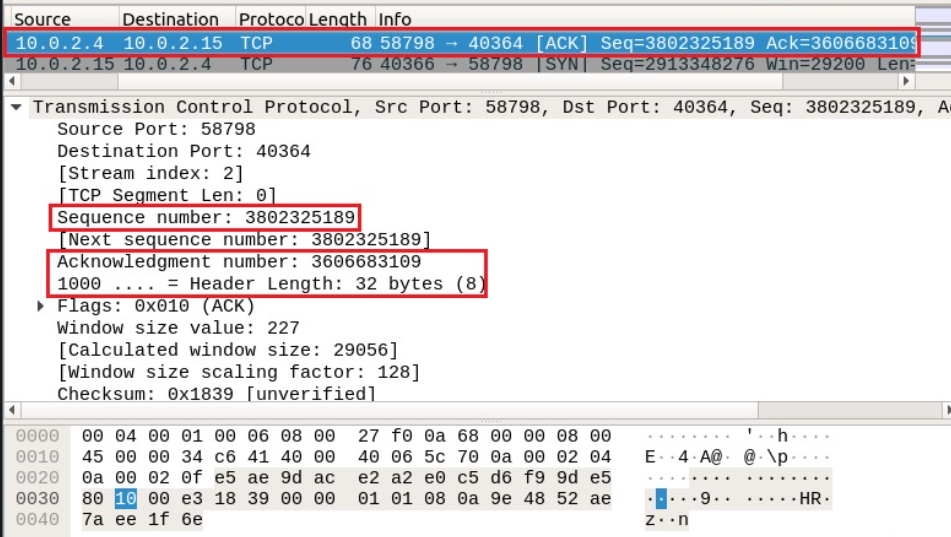
חבילה זו נשלחה מהלקוח לשרת ותפקידה הוא לאשר את קבלת התשובה מהשרת. גם כאן חל שינוי דומה בערכים שבמסגרות האדומות לעיל, באותו אופן שתיארנו קודם לכן.

כעת, נביט בחבילה **התשיעית** שנתפסה:



חבילה זו נשלחה מהלקוח לשרת. היא מתארת את סגירת החיבור שבין הלקוח לשרת, ולכן גם ערך דגל ה-FIN השתנה להיות 1 (דולק). גם כאן חל שינוי דומה בערכים שבמסגרות האדומות לעיל, באותו אופן שתיארנו קודם.

כעת, נביט בחבילה **העשירית** שנתפסה:

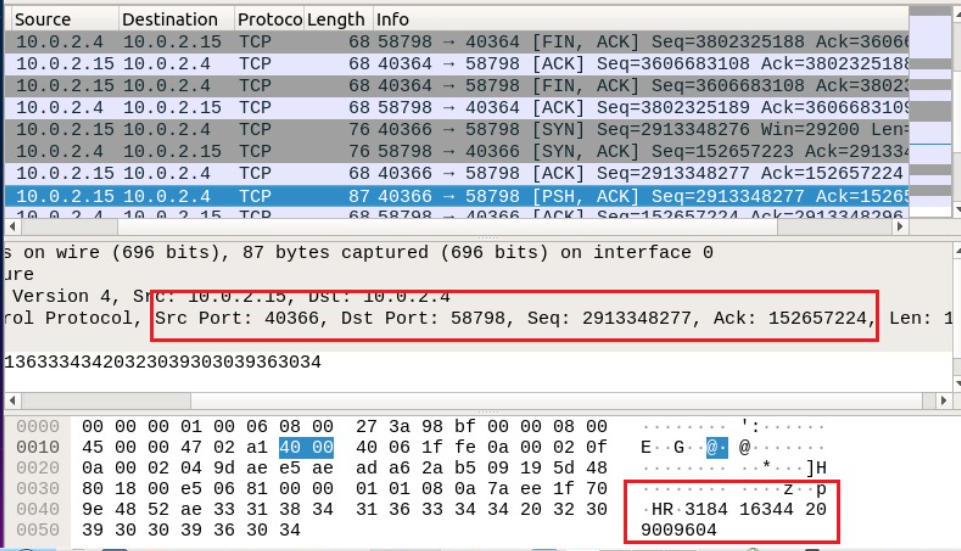


חבילה זו נשלחה מהשרת ללקוח, והיא מתארת את האישור – השרת מאשר (כלומר, שולח ACK) כי הוא קיבל את החבילה הקודמת של הלקוח (וכך החיבור ביניהם נסגר).

\*\*\*

בקובץ ה-trace.pcapng ניתן לראות כי סך הכל ישנן 20 חבילות, מתוכן תיארנו לעיל אודות ה-10 הראשונות. בקבצי השרת והלקוח שסופקו, ניתן לראות כי בהרצת קוד הלקוח יש 2 חיבורים שונים שנפתחים, "מתקשרים" עם השרת ונסגרים.

כלומר, 10 החבילות שתיארנו לעיל הן למעשה שיקוף ישיר, שלב שלב, של החיבור הראשון מתחילתו ועד סופו. 10 החבילות שלא תיארנו כאן (ומופיעות כמובן בקובץ המצורף להגשה) יכילו מידע דומה, ולכן מושמטות מכאן. הערכים השונים יהיו תוכן ההודעה (בחיבור השני הלקוח שולח את תעודות הזהות שלנו ולא את שמותינו) והפורט של הלקוח (שכן הוא מוקצה אוטומטית, ולכן טבעי שיהיה שינוי בין הפורט של הלקוח בחיבור הראשון לבין הפורט של הלקוח בחיבור השני). כמו כן יהיה שינוי בערכי ה-ack number וה-sequence number, כפי שניתן לראות בתמונה.



בתמונה שלעיל ניתן לראות את הפורט השונה של הלקוח בחיבור השני. נשים לב שפורט השרת נותר זהה כמובן. במסגרת האדומה התחתונה ניתן לראות את תעודות הזהות שלנו, כפי שנשלחו מהלקוח לשרת.

כעת, נדון ב-4 הגרסאות של קבצי השרת והלקוח – V1, V2, V3, V4.

נציין כי עשינו שימוש בגרסאות הנתונות בפייתון 2, שינינו אותן לגרסת פייתון 3 ורק לאחר מכן הרצנו את הקבצי הקוד השונים.