**רשתות תרגיל 2**

אופק ילוז 206666729

נועם כהן 209311620

חלק א': הרצנו את קוד הקליינט והסרבר מהתרגול, הנה ניתוח התעבורה המתקבלת ב wireshark:

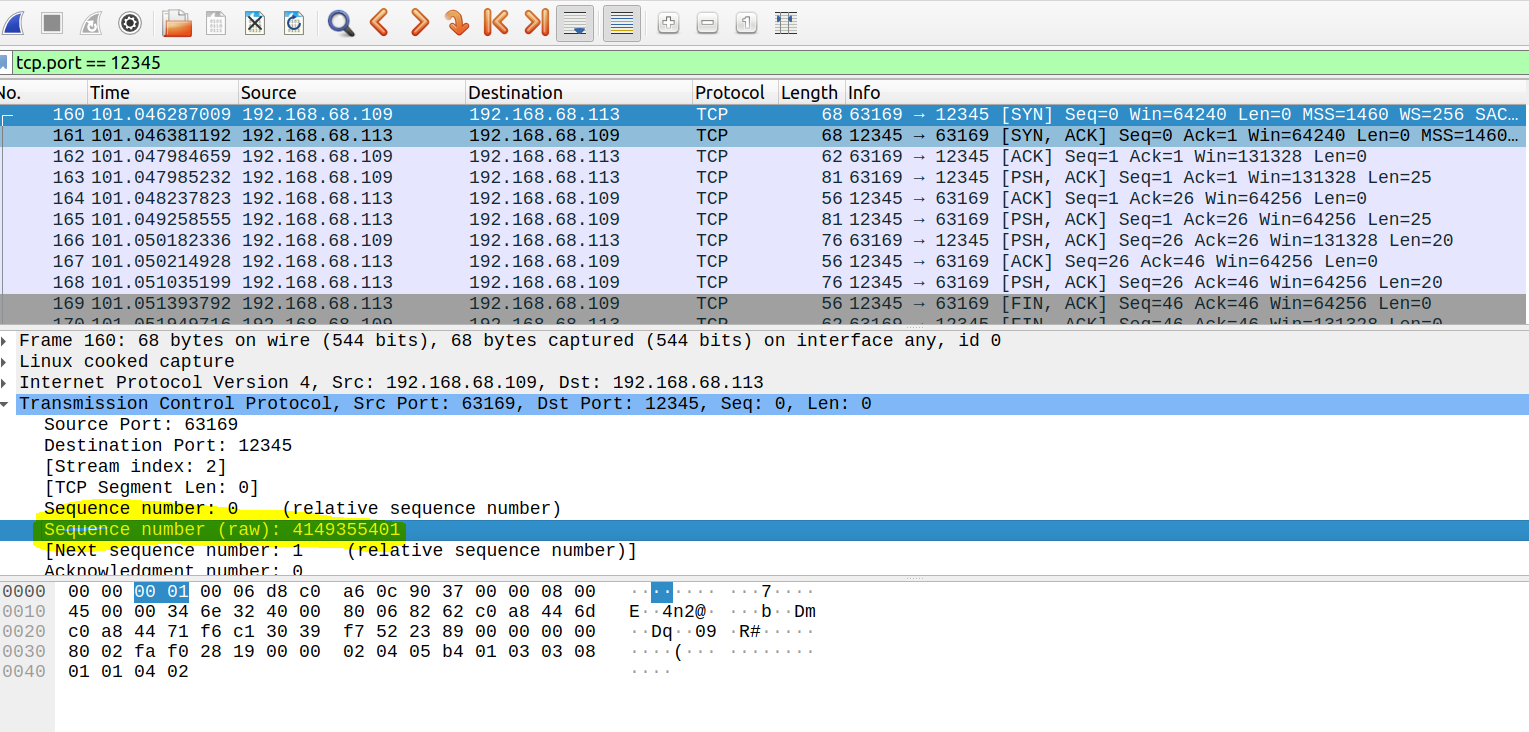
השרת (על מכונה וירטואלית – מערכת הפעלה Ubuntu) נמצא בפורט 12345 והלקוח נמצא ב 63169 (על מחשב ה Windows שמכיל את המכונה הוירטואלית)

להלן ניתוח התעבורה (בנוסף, בתוך הקובץ ClientServerWireshark.pcap מופיע קובץ ה pcap המלא):

בשניה 101 (מתחילת הsniffing), הלקוח שלח לשרת הודעת SYN. שזה בקשה ליצור חיבור מולו, שזה התחלת ה Handshake. ה Sequence Number שמופיע ב wireshark הוא

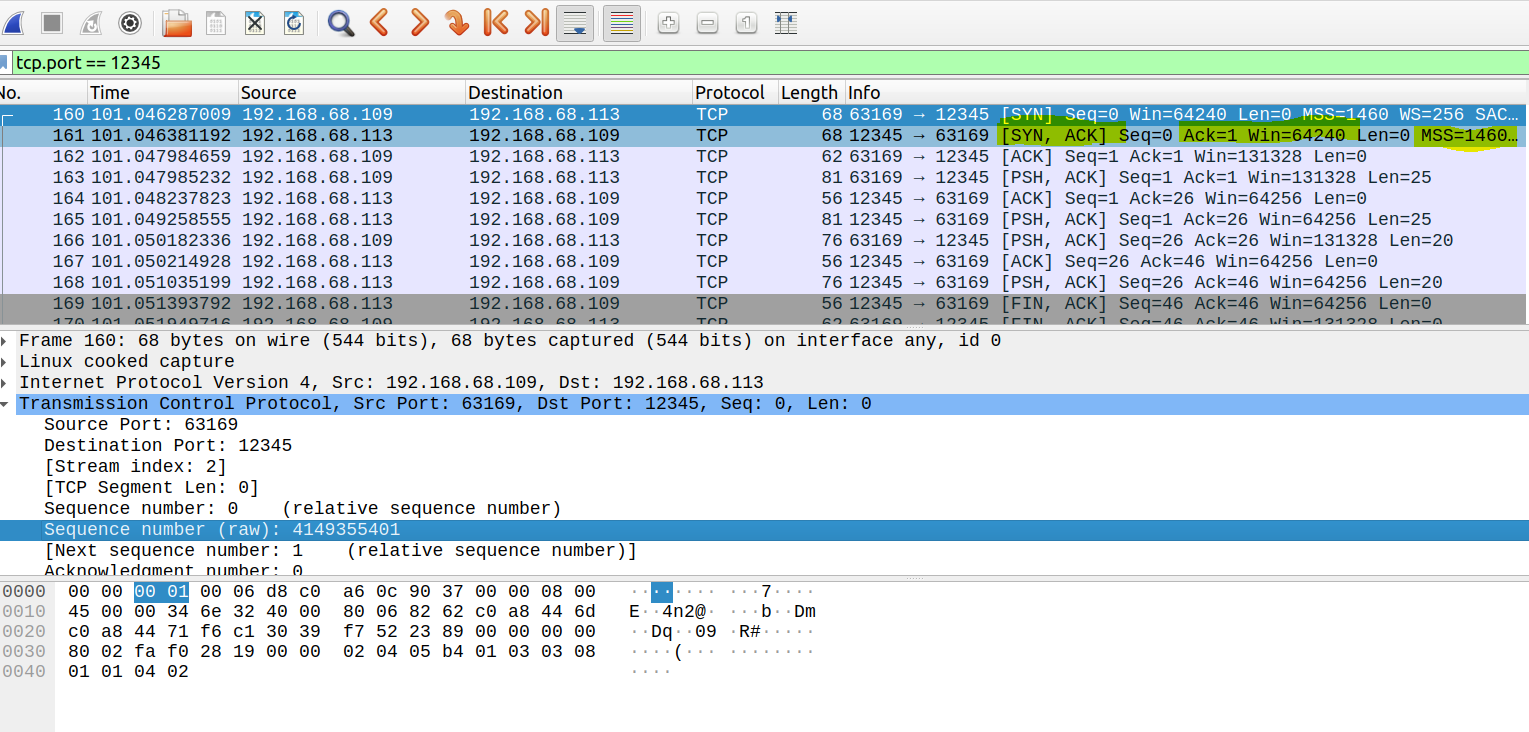
ה Wireshark מסמן את ה Sequence Number להיות 0, אבל בפועל הוא כותב בתוך ה raw שמסומן בצהוב את מספר הסגמנט הראשון האמיתי שהוא מספר ארוך יותר.

מספר ה Seq האמיתי הוא: 4148355401.



לאחר מכן, השרת שולח הודעת SYN, ACK ללקוח, כאשר ה ACK Number הוא 0 וגודל החלון הוא 64240, כמו כן, ה MSS הוא 1460.

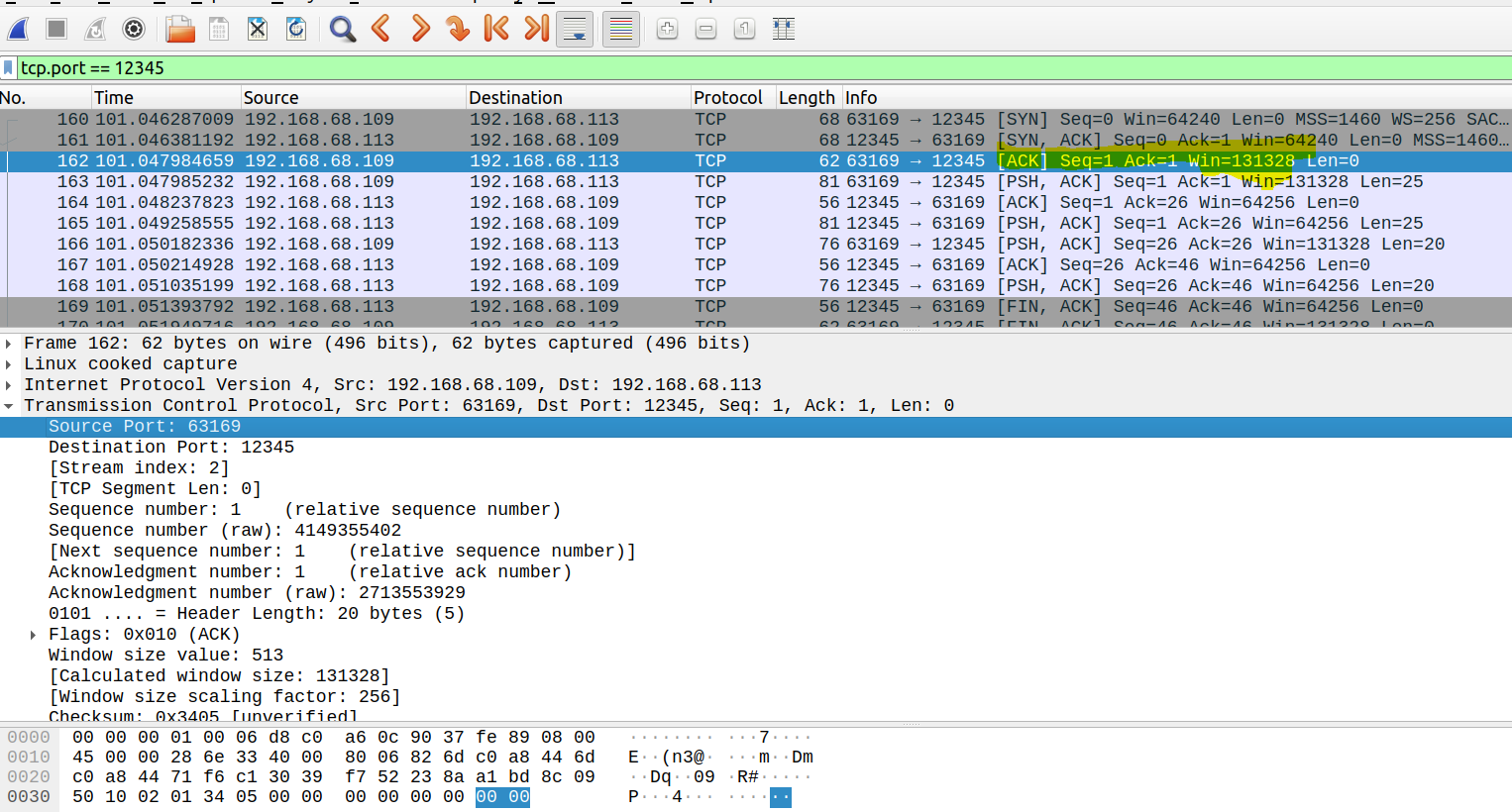
מספר ה Sequence Number האמיתי התעדכן להיות 4149355401. מספר ה ACK האמיתי הוא 2540863573.



לאחר מכן, הלקוח שולח לשרת הודעת ACK, כדי לסיים את ה handshake. מתקיים כי החלון של הלקוח הוא 131328, כמו כן, מספר ה Seq הפך ל 1, וגם ה ACK הפך ל1 (שהרי הלקוח קיבל SYN-ACK מהשרת).

מספר ה Sequence Number האמיתי התעדכן להיות 4149355402. מספר ה ACK האמיתי הוא 2713553929

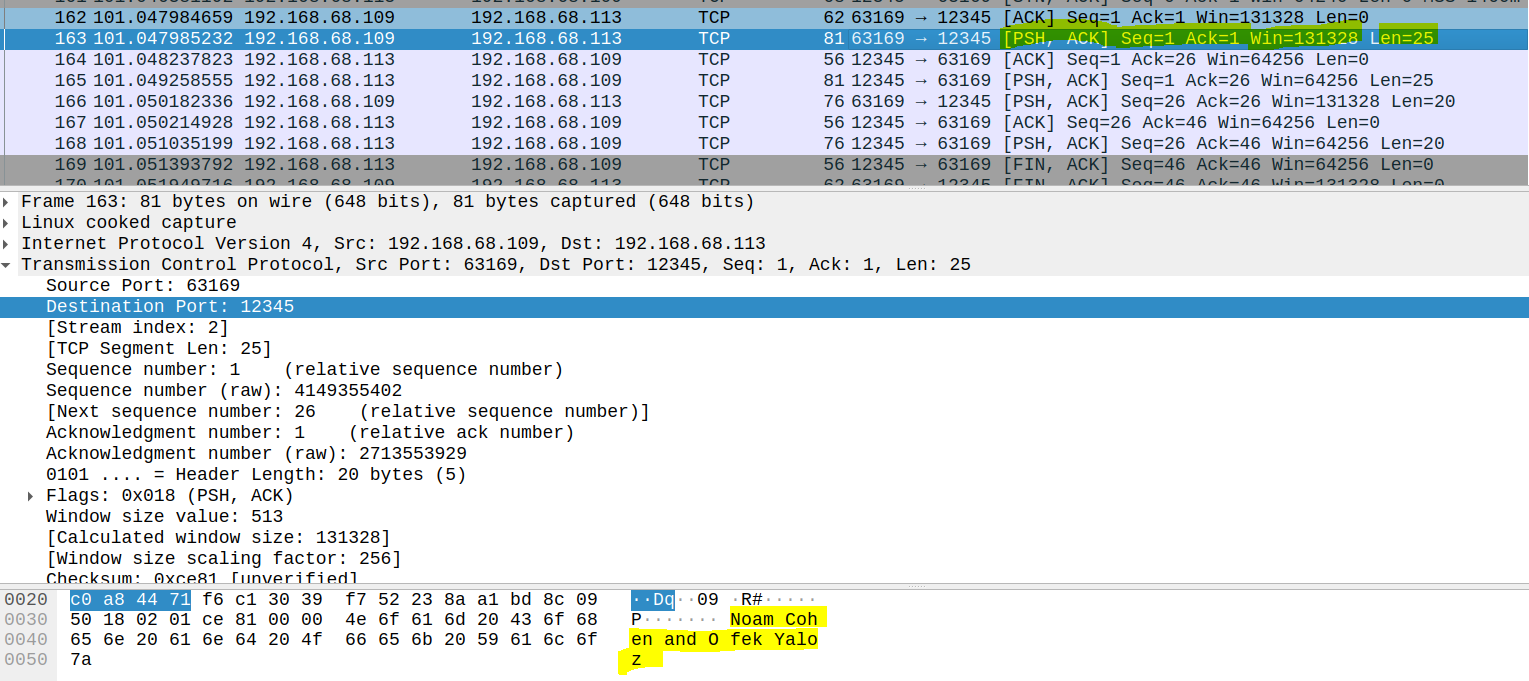
מספר ה ACK האמיתי התעדכן להיות 1.



לאחר מכן , הלקוח שולח לשרת הודעת PSH-ACK שמכילה את השמות שלנו: Noam Cohen and Ofek Yaloz, מספר הסגמנט הוא 1, ומספר הACK הוא 1 (כי מקודם השרת החזיר ACK). כמו כן, גודל החלון של הלקוח הוא 131328, אורך החבילה הוא 25.

מספר ה Sequence Number האמיתי הוא: 419355402

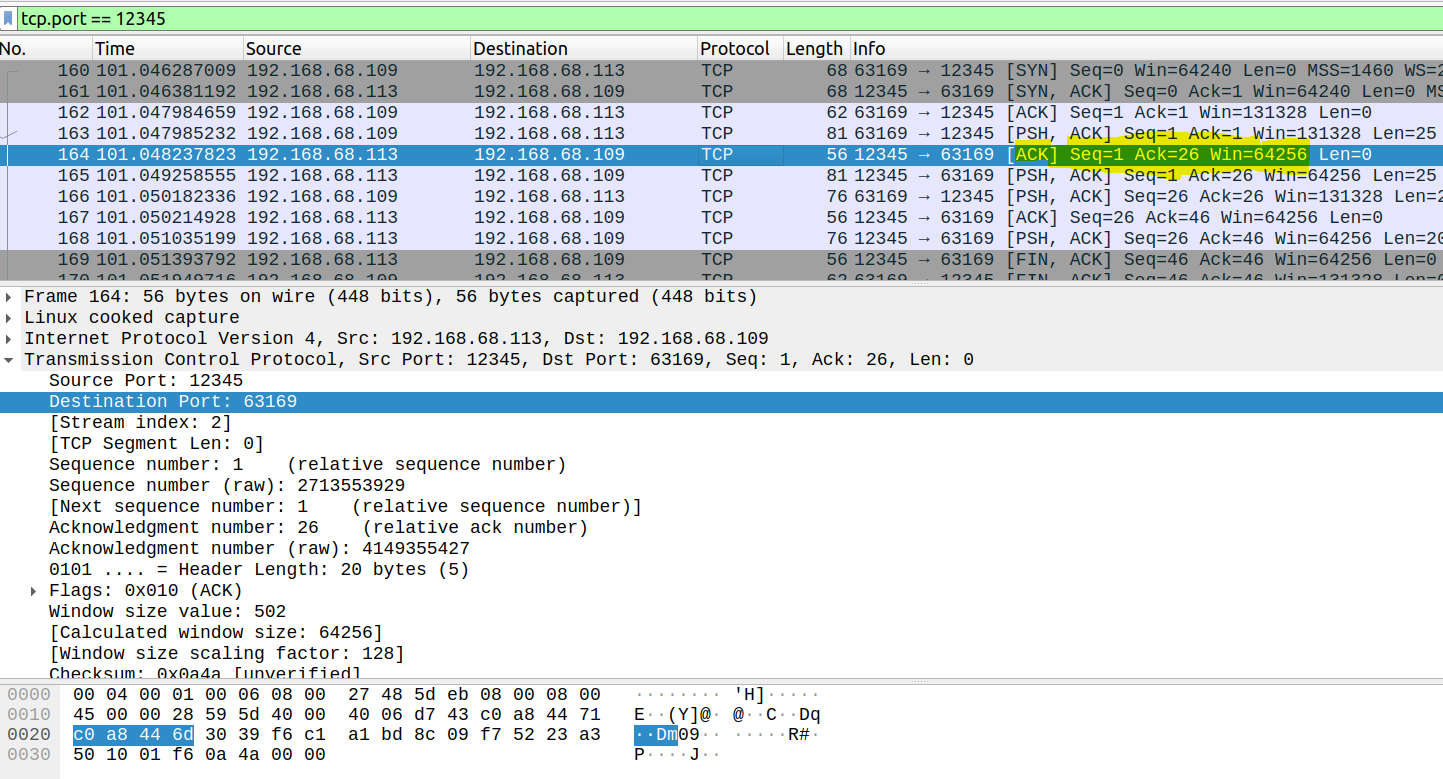
מספר ה ACK האמיתי הוא: 2713553929.



השרת מחזיר ללקוח הודעת ACK, כאשר ה ACK=26, וזה כיוון שמקודם הוא הReceived Buffer הכיל בית אחד (על ה ACK בעת ה handshake), וכעת הוא קיבל הודעה שאורכה 25 בתים, ולכן 1+25=26 וזה מספר ה ACK העדכני.

מספר ה sequence האמיתי הוא 2713553929

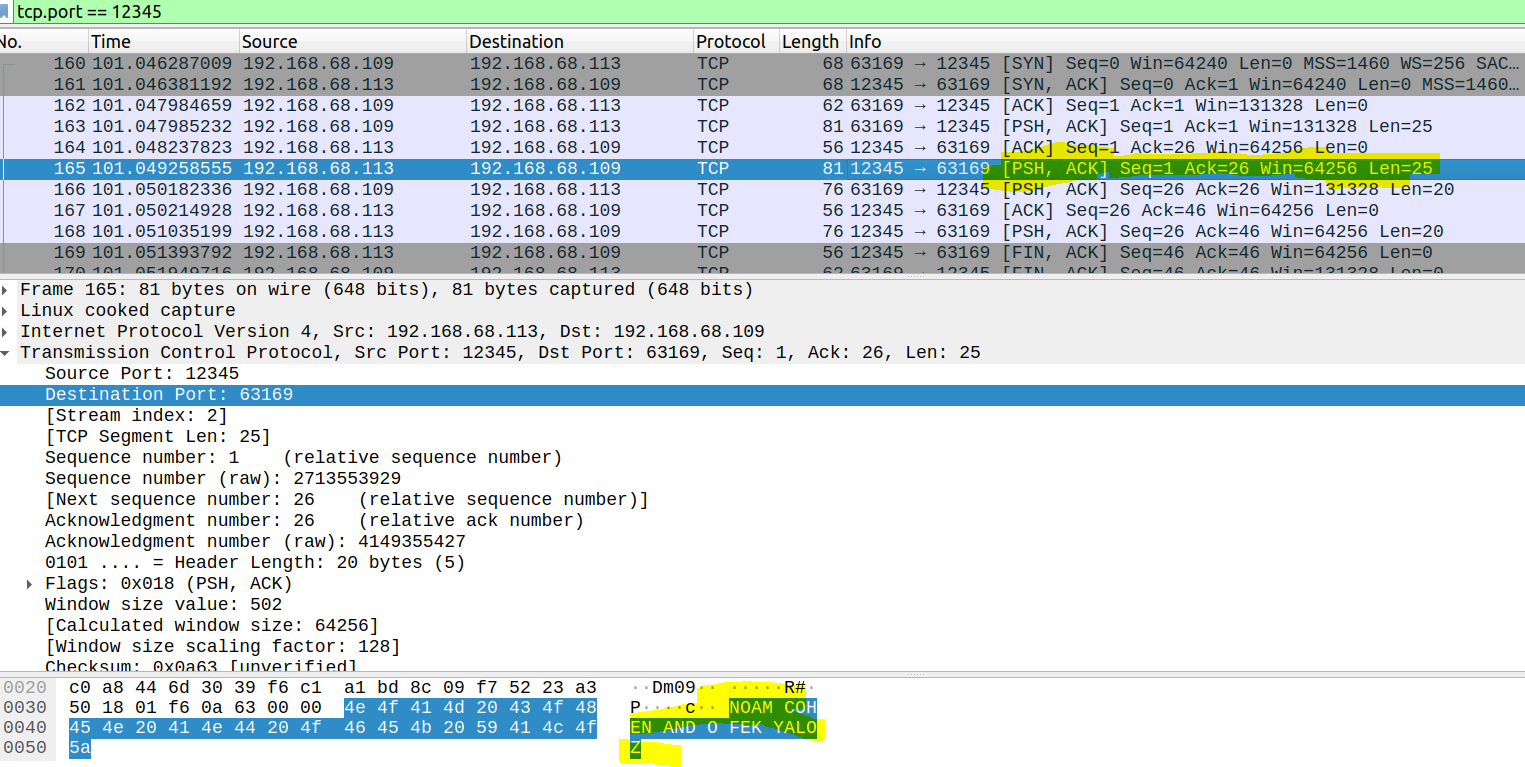
מספר ה ACK האמיתי הוא 4149355427.



לאחר מכן, השרת מחזיר ללקוח הודעה שמכילה את השמות שלה – כשהם עתה ב UPPER CASE, אזי מתקיים כי מספר הסגמנט בשרת הוא 1, (כי הוא שלח בינתיים רק SYN-ACK בהתחלה), כמו כן, גודל החבילה הוא 25, בנוסף, גודל החלון נשאר 64256.

מספר ה Sequence Number האמיתי הוא: 2713553929

מספר ה ACK האמיתי הוא: 4149355427

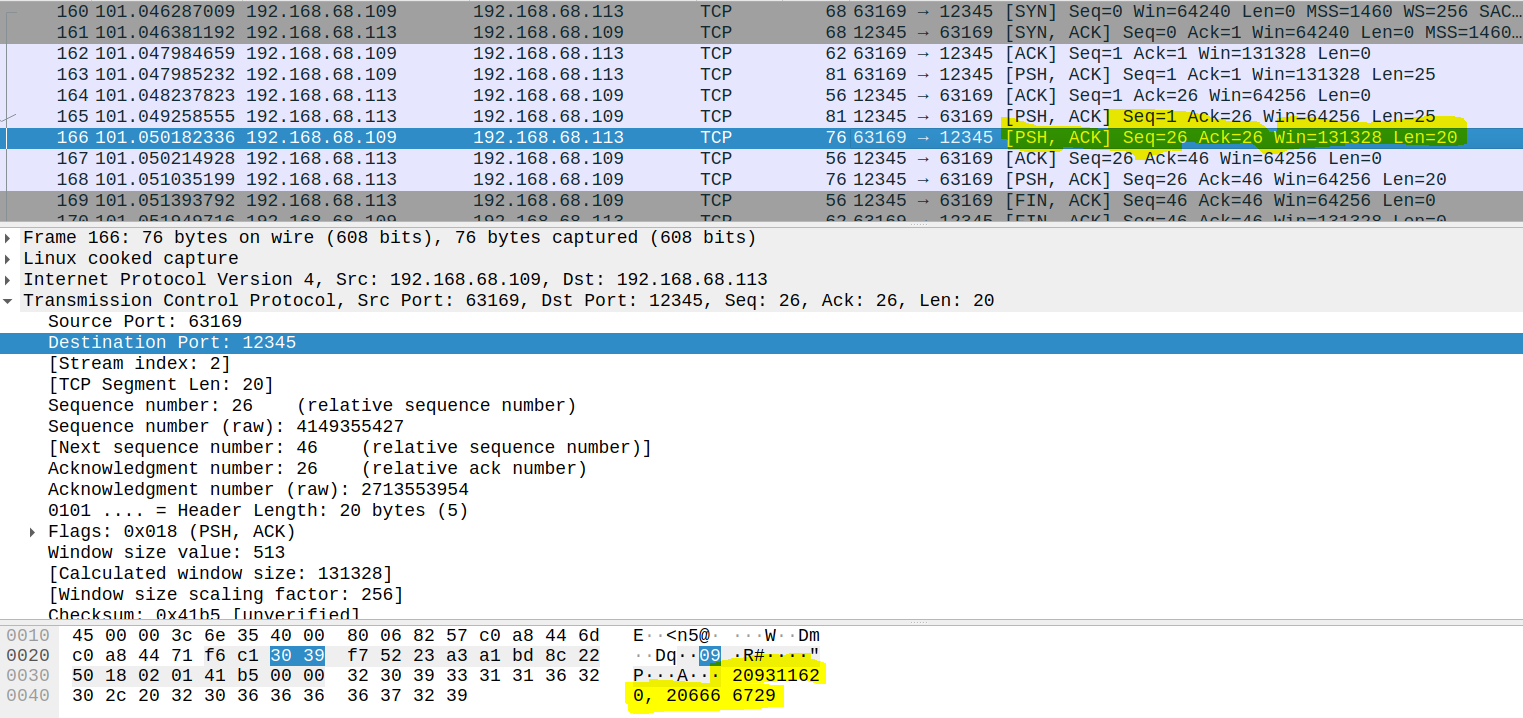


לאחר מכן, הלקוח שולח לשרת הודעה, כעת מספר הסגמנט הסידורי הוא 26 (כי מקודם הוא שלח הודעה עם השמות), מספר ה ACK הוא 26 (כי השרת שלח גם את השמות כשהם באותיות גדולות),

מספר ה Sequence Number הוא 4149355427

מספר ה ACK האמיתי הוא 2713553954

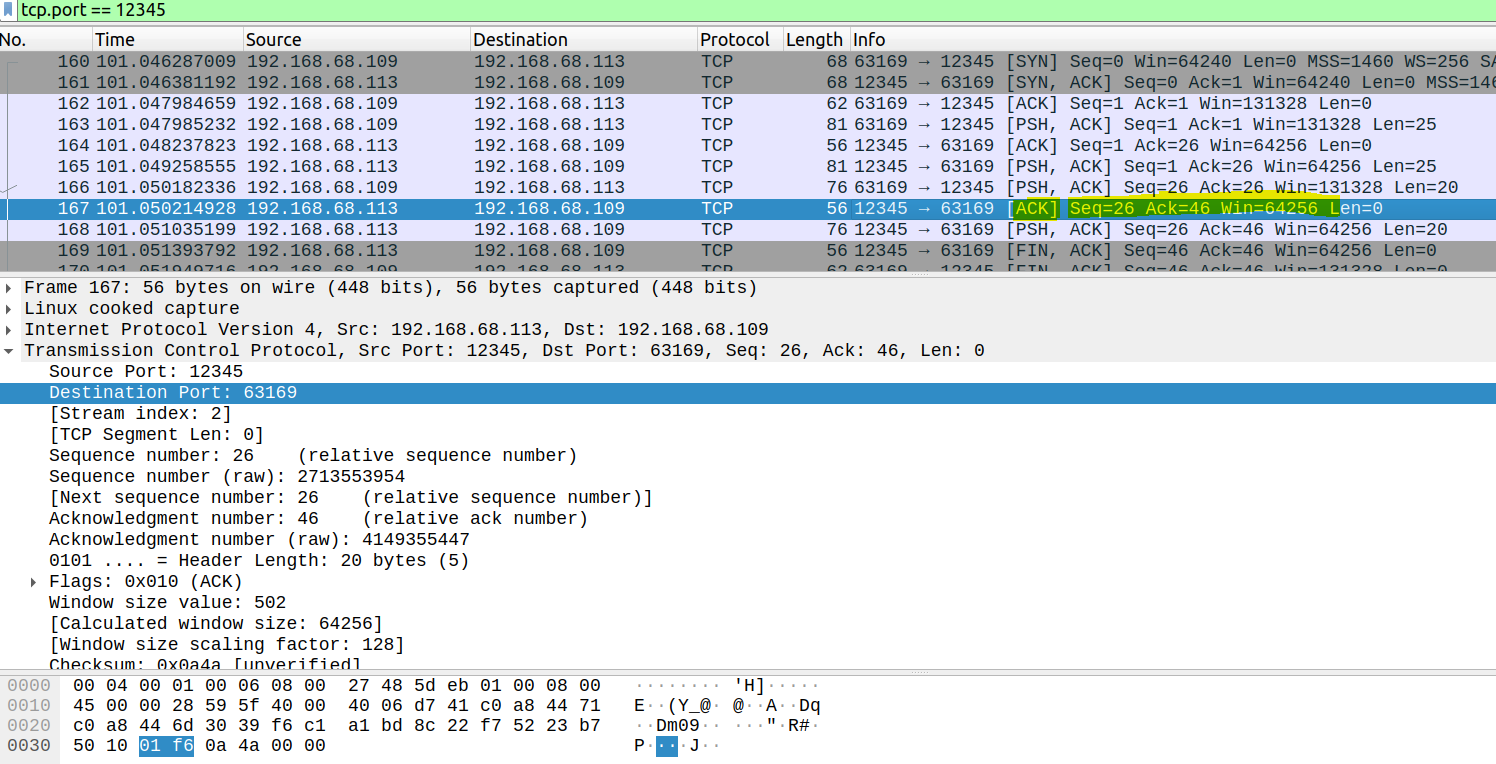
וגודל החלון שלו נשאר זהה – 131328, ואורך החבילה היא 20, בתוכן החבילה, ניתן לראות את תעודת הזהות שלנו:



עכשיו השרת מחזיר הודעת ACK כאשר ACK=46 שהוא קיבל את ההודעה עם תעודת הזהות, הסיבה שה ACK הוא 46, כי כבר השרת קיבל חבילות מהלקוח: של SYN (נספר כאחד) + שמות שלנו (25) תעודות זהות (20). כמו כן, גודל החלון נשאר זהה – 64256:

מספר ה Sequence Number האמיתי הוא: 2713553954.

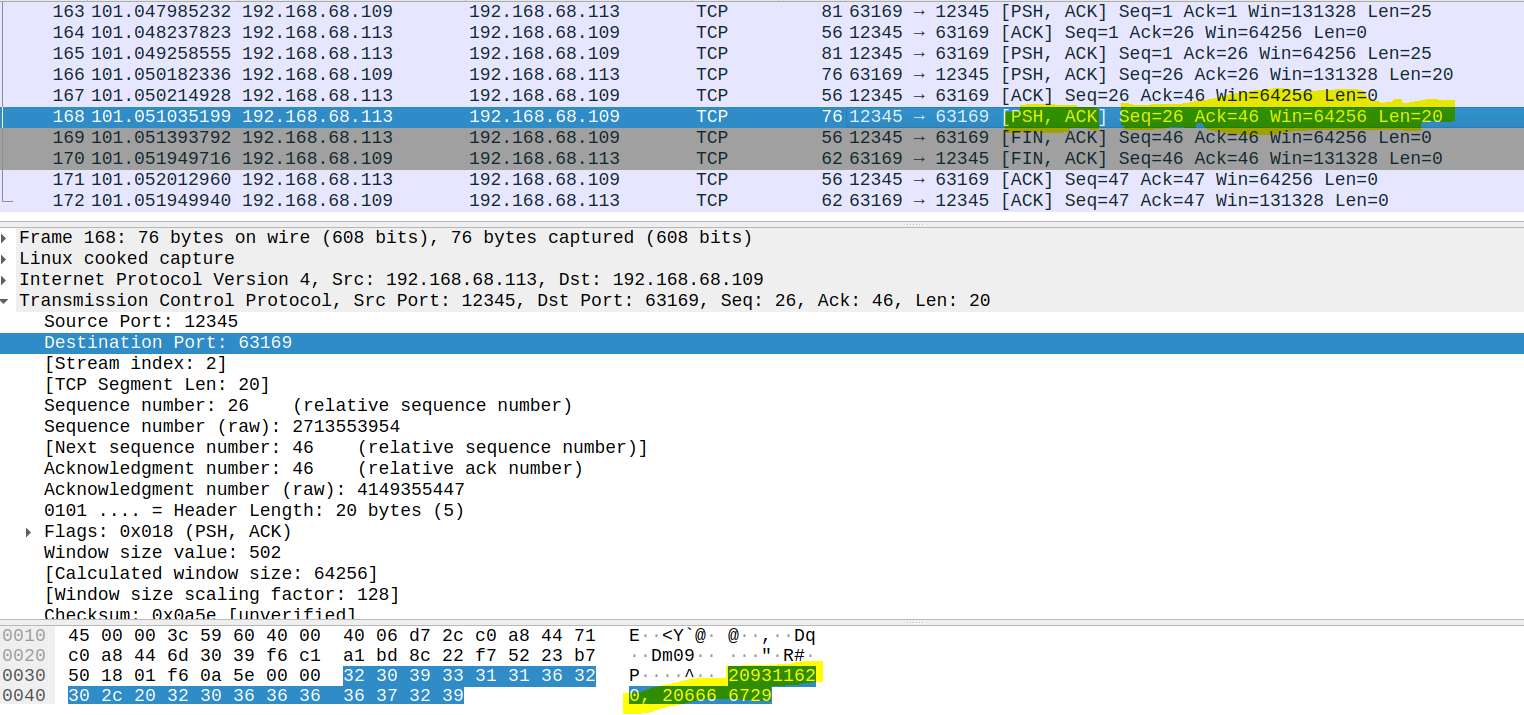
מספר ה ACK האמיתי הוא: 4149355447



עכשיו השרת מחזיר הודעת ACK שהוא קיבל את תעודות הזהות, כמו שהן נשלחו אליו. לכן דגל ה PUSH דולק, כמו כן, מספר הסגמנט הסידורי הוא 26, וה ACK הוא 46 כמו בהודעה שלפני כן.

מספר ה Sequence Number האמיתי הוא 2713553954

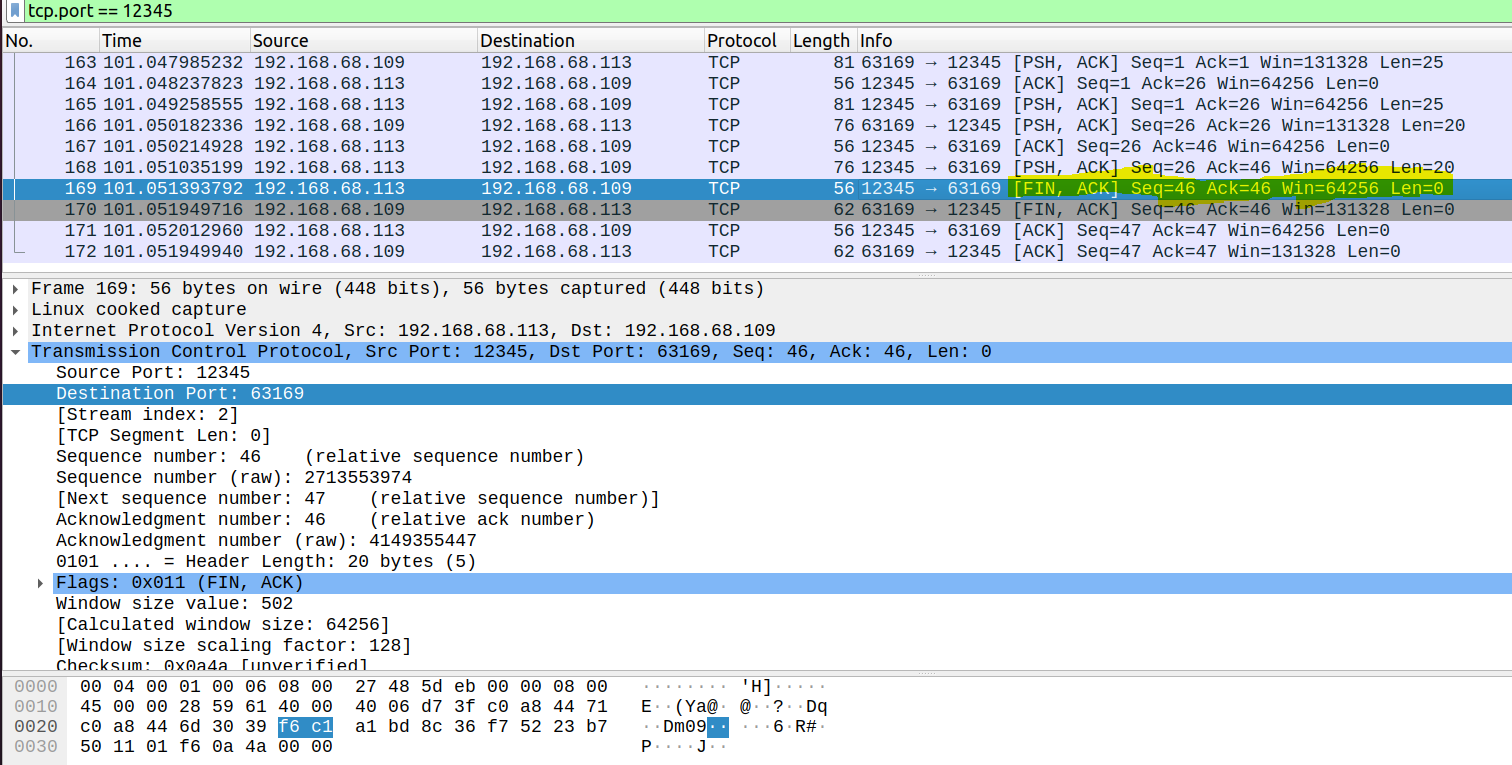
מספר ה ACK האמיתי הוא: 4149355447



השרת, מבצע client.close(), וזה בא לידי ביטוי בכך שהוא שולח הודעה ללקוח של FYN-ACK, המספר הסידורי הוא 46 (כי מקודם הוא שלח את תעודות הזהות באורך 20 – 26+20=46), כמו כן, גודל החלון נשאר 64256.

מספר ה Sequence number האמיתי הוא 2713553974

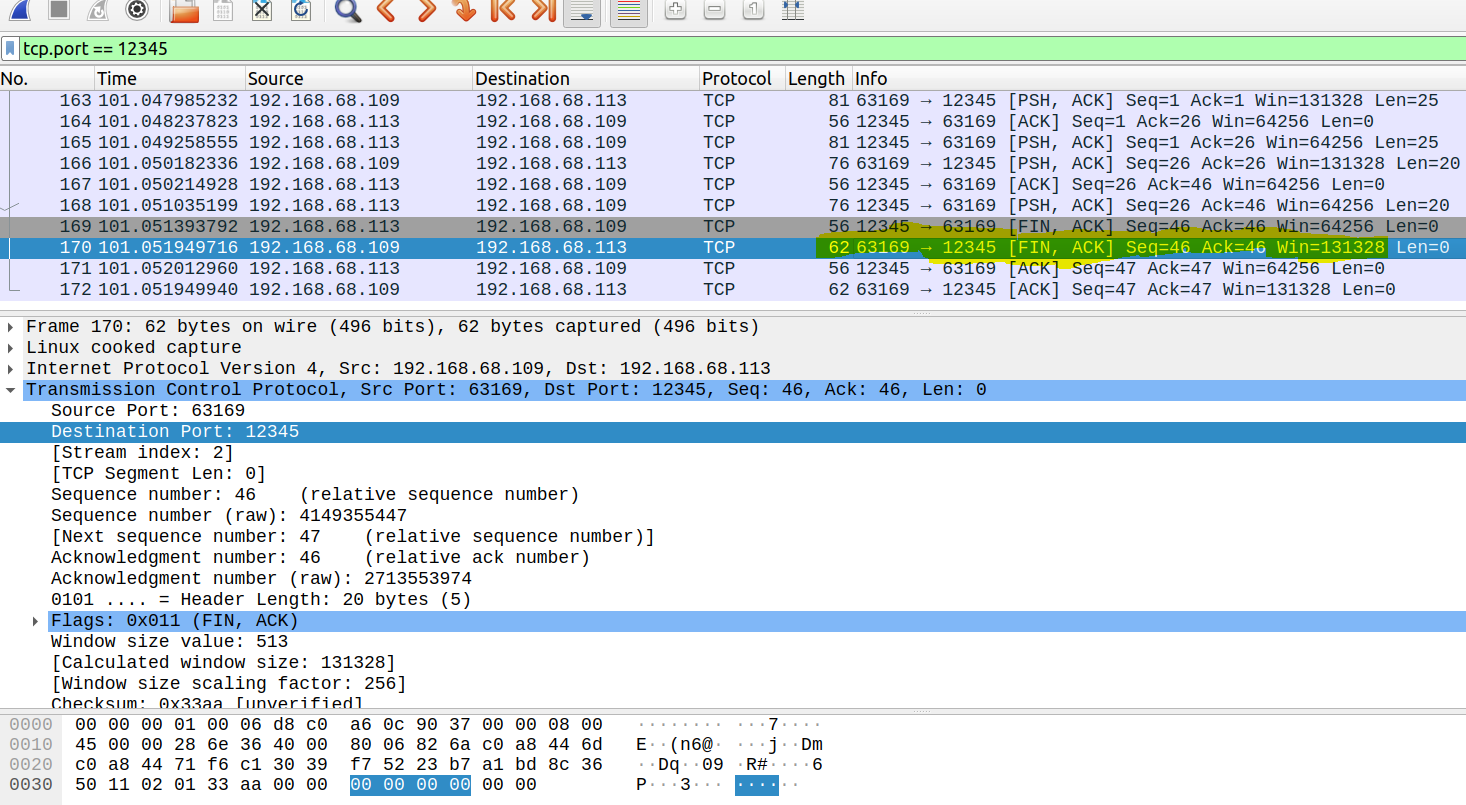
מספר ה ACK האמיתי הוא 4149355447



הלקוח מחזיר לשרת הודעת FYN-ACK עם מספר סגמנט 46, ומספר ACK של 46, אורך החלון נשאר 131328

מספר ה Sequence Number האמיתי הוא 4149355447

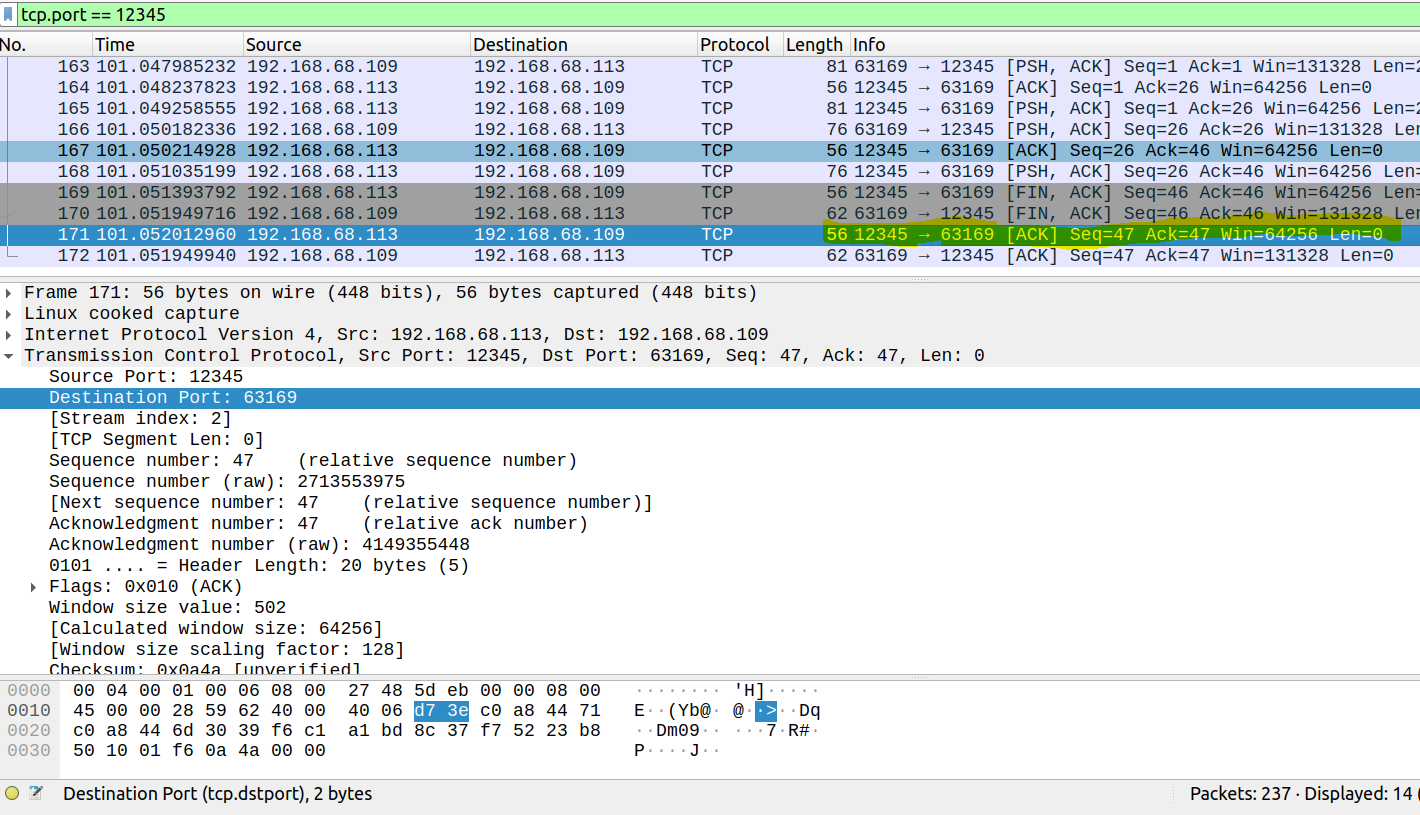
מספר ה ACK האמיתי הוא 2713553974



השרת שולח הודעת ACK ללקוח- כעת מספר ה ACK הוא 47 (הודעת ה FYN-ACK נספרת כאחת), וגם מספר הסגמנט הסידורי הוא 47, וגודל החלון הוא 64256

מספר ה Sequence Number האמיתי הוא 2713553975

מספר ה ACK האמיתי הוא 4149355448

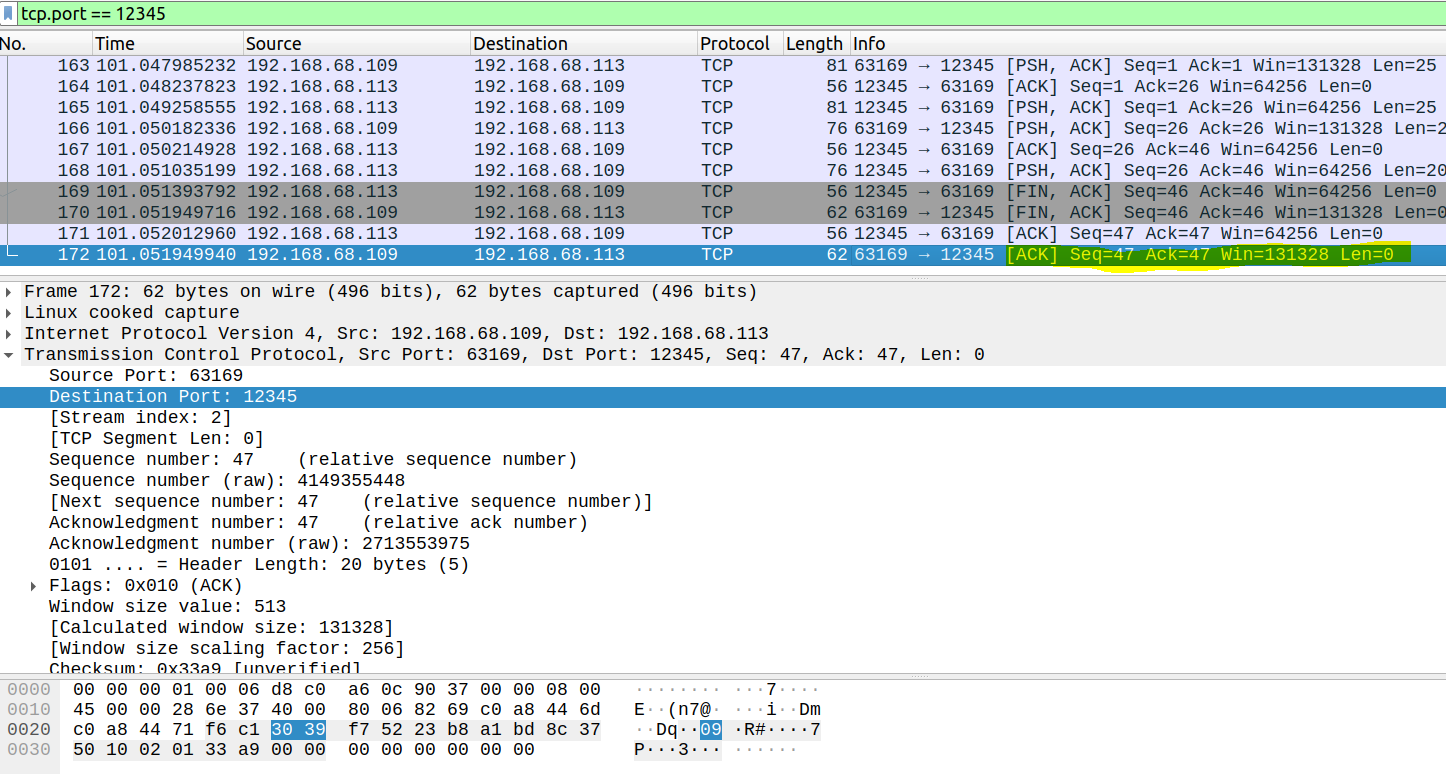


הלקוח שולח לשרת הודעת ACK כדי לסיים את החיבור.

כמו כן, עכשיו מספר הסגמנט הסידורי הוא 47, גודל החלון הוא 131328, וה ACK=47, כמו מקודם (הודעת ה FYN-ACK ממקודם נספרה כאחת).

מספר ה Sequence Number האמיתי הוא 414355448.

מספר ה ACK האמיתי הוא 2713553975



הרצת הלקוח:

python3 client.py 192.168.68.113 12345 /home/noam/Desktop 2

בשרת יש מיפוי בין לקוחות ל sockets שלהם. כי יכול להיות לקוח אחד ביותר ממחשב אחד.

פרוטוקול

* להירשם – אם לא הוזן פרמטר חמישי, הלקוח שולח לשרת בקשת REGS – והשרת יחזיר לו מחרוזת אקראית עם 128 תוים, שיש בה אותיות גדולות וקטנות וספרות.
* רישום בוצע בהצלחה- השרת שולח ללקוח את ה ID שלו שהוא באורך 128 תווים.

DONE

* עדכון מצד לקוח – הלקוח שולח לשרת הודעת עדכון UPDA – ההודעה מכילה את : סוג הקובץ, שם הקובץ, גודל הקובץ, ואת תוכנו.
* עדכון מצד שרת – SEUP – השרת שולח הודעה ללקוח עם קובץ חדש שהתעדכן: סוג הקובץ, שם הקובץ, גודל הקובץ ותוכנו.

(הלקוח שומר את המזהה שהוא נותן לו ובכל פניה עתידית, הוא צריך לשלוח את המזהה הזה).

אם לא הוזן מזהה כקלט, הלקוח מתייחס לתיקיה שהוזנה כקלט בתור תיקיה שלא קיימת במחשב, אבל המשתמש שהמזהה שלו הוכנס כקלט יש לו כבר תיקיה בשרת ולכן צריך לסנכרן את התיקיה מהשרת אל הלקוח.

בכל מקרה, הלקוח ממשיך לנטר את התיקייה ולהסתנכרן מול השרת.

להוסיף comments

מותר: sys, socket, time, watchdog, os, random, string

כשהלקוח פותח תיקיה התיקייה יכולה להישמר בשרת איפה שרוצים, הכי פשוט זה בתיקיה שבה נמצא השרת.

- ניתן  לשמור בשרת מה שתרצה איפה שתרצה. הכי פשוט להשתמש בתיקייה הנוכחית שבה נמצא השרת.

- ניטור ועדכון מלא לחלוטין. ברמת הקבצים, תיקיות, תתי קבצים ותתי תיקיות ובוודאי תוכנם. אם אני אשתמש בקובץ שנמצא על מחשב X או על מחשב Y ששניהם שייכים לאותו משתמש ושניהם נמצאים בתיקייה שמנוטרת - הם יהיו זהים לחלוטין, לא משנה איזה שינוי יתבצע במחשב X או Y, הם כל הזמן ימשיכו להתעדכן ולהיות מסונכרנים לפי הגרסא העדכנית ביותר.

python -m pip install -U watchdog

sudo apt install -y python3-watchdog