תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטיתרגיל רשתות 2:

הרצנו משני מחשבים מרוחקים פיזית.

כמו שניתן לראות ip של לקוח ושרת שונים.

נשים לב שאם פותחים הודעה מסוימת מקבלים שגם ה "מאק" שונה כי המחשבים שונים.

הסברים:

בשלב הראשון הלקוח שולח הודעת syn לשרת (הוא מנסה להתחבר אליו)

השרת שולח accept להודעה שזה הack ואז גם syn וזה אומר שמערכת ההפעלה מאשרת ומוכנה לתקשורת מצד השרת (מופיע בשלוש הודעות ראשונות בwireshark בין השרת והלקוח).

הלקוח מאשר בack את בקשת השרת להתחלת התקשורת ביניהם.

מאז, על כל הודעה מתבצע תהליך של push וacknowledge,

שמסמלים שצד אחד דחף הודעה והשני קיבל.

בסיום התקשורת שרואים אותו תחת FIN, מתבצע תהליך דומה להתחלה.

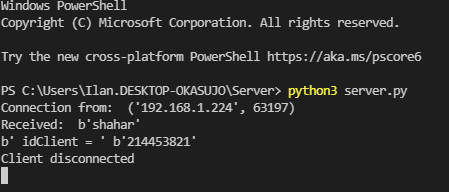
צד אחד שולח בקשה לסיום, השני שולח בקשה ומאשר את הבקשה של הצד הראשון, ואז הצד השני מקבל והקשר מתנתק (הודעות 7,8,10).

צד לקוח:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

צד שרת:



פירוט :ack & seq

נפרט לפי סדר ההודעות בwire

בהודעה הראשונה (מהלקוח) דגל הack מכובה כי עדיין אין על מה לאשר חבילות.

בהודעה השנייה שהשרת מאשר חיבור ומבקש גם חיבור, ה ack שלו משתנה לאחד, כלומר, בחיבור הראשון למדנו שמעלים את מספר הack למרות שלא התקבל מידע לשכבת האפליקציה (כדי לחסוך במקרים מסוימים את הצורך לשלוח מחדש הודעת בקשת התחברות).

לאחר מכן הseq של הלקוח מתעדכן ל1 בגלל הack שקיבל מהשרת,

בהמשך התקשורת לפני סגירת החיבור (הודעות בwire: 4,5,6,9), הack של כל צד עולה בהתאם

לכמות הבתים שהוא מקבל (אורך ההודעה) והseq של הצד השולח מתעדכן בהתאם לack שהוא מקבל על כל חבילה.

בסוף החיבור מתנהל תהליך דומה להתחלה.

בצד השרת:

ערך seq: 8 בגלל שבהקמת החיבור קיבלנו ack שהחיבור זמין מהלקוח, במהלך החיבור שלחנו הודעה בגודל 6 (אורך השם שלי) וקיבלנו אישור על זה, ובסוף החיבור קיבלנו מהלקוח אות שהשיחה הסתיימה.

ערך ack: 17 שלחנו ללקוח ack בהתחלה, כדי לאותת לו שאנחנו מוכנים לחיבור, קיבלנו ממנו שם באורך שש בתים, ותעודת זהות באורך 9 בתים, ולכן שלחנו לו עליהן ack לפי הסדר, ובסוף החיבור שלחנו ack לפי הסדר, שאנחנו מוכנים לסיים את השיחה.

נובע מכך כי, בצד הלקוח הערכים יהיו הפוכים כמו שרואים בwire.