תרגיל מתקדם - תקשורת סודית מעל TCP

קודם לכן, בסעיף תרגיל - תקשורת סודית מעל מספרי פורט, סייעתם לשני תלמידים, יואב ומאור, לתקשר בצורה סודית מעל הרשת. להזכירכם, מטרת התלמידים הייתה להצליח להעביר מסרים מאחד לשני, מבלי שאף אדם יוכל לקרוא אותם, גם אם הוא יכול להסניף את התעבורה ביניהם. בתרגיל זה תסייעו לשתי תלמידות, זוהר ושיר, להשיג מטרה דומה, אך בדרך אחרת.

את הפתרון הקודם מימשתם באמצעות UDP, הפעם תשלחו את המסר הסודי מעל TCP.

תרגיל מכין- מימוש Three Way Handshake ע"י Scapy

בשלב הראשון תממשו Three Way Handshake בעזרת Scapy. מימוש הלקוח יהיה כמו בתרגיל מודרך 6.18, מימוש Three Way Handshake, אך בנוסף תממשו גם את צד השרת.

הלקוח:

- ישלח פקטות מסוג SYN לשרת, לפורט אקראי כלשהו שייבחר ע"י הלקוח (חשוב להמשך התרגיל). גם ה-source port צריך להיות אקראי כפי שמקובל בתקשורת TCP.
- יצפה לקבלת SYN-ACK מהשרת. אם לא מתקבלת פקטה זו, יישלח SYN בלולאה כל זמן מוגדר.
- עם קבלת ה-SYN-ACK, הלקוח ישלח את הודעת ה-ACK שחותמת את התהליך. יש להקפיד על ערכי seq ו-ack נכונים, בהתאם לערכים שנשלחו בהודעות קודמות.

השרת:

- יאזין לפקטות מסוג SYN. שימו לב שהפקטות עשויות להשלח לכל פורט (בניגוד לתקשורת TCP רגילה בה השרת מבצע listen לפורט ספציפי) כלומר השרת לא יכול להעזר במספר הפורט כדי לסנן את הפקטות. חישבו איך בכל זאת ניתן לסנן פקטות SYN.
- יענה עם פקטת SYN-ACK. יש להקפיד על ערכי seq ו-ack נכונים.

בסיום התהליך עליכם להיות מסוגלים למצוא את הפקטות ששלחתם הן בשרת והן בלקוח. לדוגמה:

No.	Time	Source	Destination	Protocol	Length	Info
14	4.96560800	10.0.0.4	10.0.0.8	TCP	54	2222→80 [SYN] Seq=0 Win=8192 Len=0
17	5.78265300	10.0.0.8	10.0.0.4	TCP	60	80→2222 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
18	5.96839100	10.0.0.4	10.0.0.8	TCP	54	2222→80 [ACK] Seq=1 Ack=1 Win=8192 Len=0

No.	Time	Source	Destination	Protocol	Length	Info
561	13.0079590	10.0.0.4	10.0.0.8	TCP	60	2222→80 [SYN] Seq=0 Win=8192 Len=0
597	13.8201310	10.0.0.8	10.0.0.4	TCP	54	80→2222 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0
607	14.0082450	10.0.0.4	10.0.0.8	TCP	60	2222→80 [ACK] Seq=1 Ack=1 Win=8192 Len=0

יש לציין שמימוש ה-Three Way Handshake באמצעות Scapy לא יוצר socket אמיתי של TCP אלא משמש רק לטובת תרגול. במילים אחרות, לא נוצר קשר אמין ברמת שכבת התעבורה וההודעות הבאות שיועברו בין השרת ללקוח לא יקבלו ACK (אלא אם כן אתם תממשו זאת בעצמכם...)

מימוש תקשורת סודית

כעת נעבור לחלק בו אנחנו מממשים תקשורת סודית. בכדי שהעברת המידע תתבצע כך שלא ניתן יהיה להבין אותה, תשלחו את הפקטות בסדר לא נכון, ובכל פעם לפורט אחר. הערך של ה-Sequence Number לא יהיה תקין ביחס לקישור שהוקם, אלא ביחס למסר הכללי אותו התלמיד מנסה להעביר.

נסביר באמצעות דוגמה: במידה ששיר מעוניינת לשלוח לזוהר את המסר: "TCP connections are awesome", היא תוכל, למשל, לעשות זאת כך:

- להרים קישור (באמצעות Three Way Handshake) לפורט 24601.
- לשלוח את המסר "TCP c". ערך ה-Sequence Number יהיה 0.
- לסגור את הקישור.
- להרים קישור (באמצעות Three Way Handshake) לפורט 1337.
- לשלוח את המסר "awesome". ערך ה-Sequence Number יהיה 20.
- לסגור את הקישור.
- להרים קישור (באמצעות Three Way Handshake) לפורט 555.
- לשלוח את המסר "onnections are ". ערך ה-Sequence Number יהיה 5.
- לסגור את הקישור.

שימו לב: התקשורת אותה תיארנו לעיל אינה תקשורת TCP תקינה. לדוגמה, כאשר נשלח המסר "TCP c", ערך ה-Sequence Number אמור להיות מחושב בהתאם לערך ה-ISN שהוגרל בתחילת הקישור, ולא להיות 0. הוא מקבל את הערך 0, שכן אלו הבתים הראשונים במסר ששיר מנסה לשלוח לזוהר. מעבר לכך, אין חשיבות למספר הפורט.

עבור המסתכל מן הצד, התקשורת נראית כמו תקשורת TCP שאינה תקינה ואינה הגיונית. עם זאת, מי שמכיר את דרך הפעולה של התקשורת הסודית, יכול להבין את המסר.

חלק א' - לקוח ששולח מסר סודי

כתבו סקריפט בשם tcp_secret_message_sender.py. על הסקריפט לבצע את התהליך הבא:

- לבקש מהמשתמש להקליד הודעה.
- לבקש מהמשתמש מספר שמשמעותו לכמה חלקים לחלק את ההודעה בהעברת המסר הסודי.
- לשלוח את ההודעה באופן סודי, כפי שתואר למעלה, ובהתאם למספר החלקים שביקש המשתמש.

דגשים

- עליכם להשתמש ב-Scapy בכדי להסניף, לקבל ולשלוח את החבילות.
- את כתובת ה-IP של השרת אתם יכולים לכלול בקוד שלכם באופן קבוע ולא לבקש אותה מהמשתמש.
- את הפורט עליכם לקבוע באופן אקראי (בכל מקרה השרת לא יכול להניח שידוע לו מספר הפורט).
- במידה שמספר החלקים שהמשתמש שלח גדול יותר ממספר התווים שבהודעה, הציגו לו הודעת שגיאה.
- על מנת לבדוק את תרגיל זה, עליכם להשתמש בשני מחשבים שונים – אחד ללקוח ואחד לשרת[1].

חלק ב' - שרת שקורא מסר סודי

כתבו סקריפט בשם tcp_secret_message_receiver.py. על הסקריפט לבצע את התהליך הבא:

- לחכות לפניות TCP (סגמנט עם הדגל SYN).
- להרים קישור (באמצעות Three Way Handshake) עם מי שפתח את חיבור ה-TCP.
- לקבל מעל חיבור ה-TCP את המידע של השולח, ולהבין מה המיקום שלו במסר הכולל.
- לבסוף, לחבר את המידע בסדר הנכון ולהדפיס את המסר הסודי שהתקבל.

דגשים

- עליכם להשתמש ב-Scapy בכדי להסניף, לקבל ולשלוח את החבילות.
- בכדי להדפיס את המסרים שהתקבלו מהלקוחות, על השרת להבין מתי מסר אחד נגמר, ומסר אחר מתחיל. חשבו על דרך לעשות זאת. תוכלו לשנות את קוד הלקוח לשם כך.
- בדקו את השרת באמצעות הלקוח שכתבתם בחלק א'. וודאו כי אתם מצליחים לקרוא נכונה את המסרים הנשלחים אליכם. בדקו הן מסרים שונים והן מספר שונה של חלקים אליהם המסר יחולק.

חלק ג' – תקשורת אמינה

הוסיפו אמינות לתקשורת בין השרת והלקוח שלכם. במילים אחרות, חישבו על מצב בו פקטה לא הגיעה וטפלו בו. אתם מוזמנים לממש פתרון כרצונכם, אפשר ללמוד על מנגנון ה-ACKים של TCP ולשאוב ממנו רעיונות. עם זאת עקב המורכבות שלו עדיף לממש מנגנון פשוט יותר.

חלק ד' - לקוח ושרת משולבים

כתבו סקריפט בשם tcp_secret_messenger.py. הסקריפט ישלב את הלקוח והשרת שכתבתם בחלקים הקודמים של התרגיל, ויאפשר לשני לקוחות לתקשר זה עם זה כמו בצ'אט, באמצעות העברת מסרים סודיים.

[1] באופן תאורטי, יכלנו לעשות זאת מעל loopback device – כלומר מעל הכתובת "127.0.0.1", המוכרת לנו מתרגילים קודמים. עם זאת, עקב Bug של Scapy בשליחה וקבלת מסגרות מעל loopback device ב-Windows, נשתמש בשני מחשבים.