**דו"ח – תרגיל 1**

**רשתות**

נפרט על פעולתו של הקוד בכל אחד מן השלבים[[1]](#footnote-1):

לאורך כל ההסבר שלושת התוכנות רצו על טרמינלים שונים כמתואר בטבלה הבאה:

|  |  |  |  |
| --- | --- | --- | --- |
| client | foo | server |  |
| רנדומלי, מוקצה ע"י מערכת ההפעלה | 12345 | 23456 | port |
| 127.0.0.1 | 127.0.0.1 | 127.0.0.1 | ip |

**שלב 1**

בשלב זה foo "מתנהג יפה" ומעביר את החבילות כמו שצריך בתנאי שאינן מכילות יותר מ-100 בתים.

נצרף דוגמת הרצאה של התכנית:

**server**

>> python3 server.py 23456

**foo**

**>>** python3 foo.py 12345 127.0.0.1 23456 1

**client**

>> python3 client.py 127.0.0.1 12345 input.txt

נסניף את התעבורה המתאימה ב wireshark המגיעה ויוצאת מן foo[[2]](#footnote-2):

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

ונתסכל על השכבות השונות בפקטה:

שכבת התעבורה – בשכבת התעבורה עבדנו עם פרוטוקול UDP, לכן ניתן לראות בשכבה זו פורט מקור ופורט יעד:



**תמונה 1.1** : פקטה מ client אל foo



**תמונה 1.2** : הפקטה מועברת ל server



**תמונה 1.3** : פקטת תגובה חוזרת מ server ל foo

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

**תמונה 1.4**: הפקטה חוזרת ל client

כמו שניתן לראות, פורט המקור ופורט היעד משתנים בהתאם למסלול הפקטה

שכבת הרשת – בשכבת הרשת נראה את כתובות ה-ip, מאחר והרצנו את התוכנות בטרמינלים שונים באותה המכונה נראה כי כולם יהיו 127.0.0.1



**תמונה 1.5** : כתובת המקור והיעד בפקטות

שכבת אפליקציה

בשכבת האפליקציה ניתן לראות את המידע שמועבר בפקטות, הכתוב בפרוטוקול שהגדרנו לעיל[[3]](#footnote-3). לדוגמה:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

**תמונה 1.6** : הפקטה הראשונה שנשלחה מ client ל foo

**שלב 2**

בשלב זה foo בנוסף לתנאי של גודל החבילות זורק באקראי אחוז מסויים של החבילות בשני כיוונים. על כן נראה תעבורה יותר מסיבית – הרבה יותר פקטות יישלחו.

נצרף דוגמת הרצאה של התכנית:

**server**

>> python3 server.py 23456

**foo**

**>>** python3 foo.py 12345 127.0.0.1 23456 2

**client**

>> python3 client.py 127.0.0.1 12345 input.txt

נסניף את התעבורה המתאימה ב wireshark המגיעה ויוצאת מן foo:

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

התעבורה אכן יותר גדולה – 114 פקטות לעומת 68 בשלב 1

נתסכל על השכבות השונות בפקטה:

שכבת התעבורה – בשכבת התעבורה לא השתנה הפרוטוקול. עבדנו עם פרוטוקול UDP, לכן ניתן לראות בשכבה זו פורט מקור ופורט יעד:



**תמונה 1.1** : פקטה מ client אל foo



**תמונה 1.2** : הפקטה מועברת ל server



**תמונה 1.3** : פקטת תגובה חוזרת מ server ל foo

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

**תמונה 1.4**: הפקטה חוזרת ל client

כמו שניתן לראות, פורט המקור ופורט היעד משתנים בהתאם למסלול הפקטה

שכבת הרשת – גם בשכבת הרשת לא השתנה הפרוטוקול. נראה את כתובות ה-ip, מאחר והרצנו את התוכנות בטרמינלים שונים באותה המכונה נראה כי כולם יהיו 127.0.0.1



**תמונה 1.5** : כתובת המקור והיעד בפקטות

שכבת אפליקציה

בשכבת האפליקציה ניתן לראות את המידע שמועבר בפקטות, הכתוב בפרוטוקול שהגדרנו לעיל – בשלב זה כבר מספרנו כל פקטה, נוכל לראות זאת בתחילת חלק המידע. לדוגמה:

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

**תמונה 1.6** : הפקטה הראשונה שנשלחה מ client ל foo

**שלב 3+4**

שלבים אלו עובדים באותו פרוטוקול של שלב 2 בשכבת האפליקציה, הרשת והתעבורה ולכן נראה מידע דומה בפקטות.

התעבורה תהיה גדולה ככל שפרמטרי העיכוב וזריקה יגדלו.

מצ"ב קבצי הסנפה לכל אחד מהשלבים

1. קובץ הטקסט מכיל חלק מן הסיפור "אליסה בארץ הפלאות" [↑](#footnote-ref-1)
2. הסינון בוצע ע"י הפילטר udp&&udp.port==12345 [↑](#footnote-ref-2)
3. בשלב זה עוד לא היה צריך למספר את הפקטות [↑](#footnote-ref-3)