

תרגיל מספר 1

להגשה עד ל 22.11 בשעה 22 : 00

1 חלק א

צרו במחשב שלכם קבצי קליינט וסרבר בעזרת הקוד שבמודל עבור פרוטוקול UDP והריצו אותם. ניתן להריץ את השרת והלקוח בכל קונפיגורציה שרוצים (אותו מחשב/מחשבים שונים/מכונות וירטואליות), העיקר שניתן יהיה לתפוס את התעבורה בWireshark.

1. ודאו שהשרת והלקוח מצליחים לתקשר ביניהם. כלומר, שלחו הודעה מהלקוח לשרת המכילה את השם והת"ז שלכם וודאו שהשרת מחזיר את התשובה המתאימה. יש לשנות את סקריפט הלקוח בהתאם. הפעילו ברקע Wireshark והסניפו את התעבורה.

2. סננו את החבילות שתפסתם כך שיוצגו רק החבילות שהלקוח שלח לשרת וחבילות שהשרת שלח ללקוח. הסבירו כיצד סיננתם את החבילות. שימרו את ה trace המסונן (ללא החבילות שאינן קשורות) והגישו אותו בשם trace1 והסיומת של Wireshark.

3. בקוד השרת והלקוח יש שימוש במספרי פורט. הסבירו איפה ומה השימוש, והדגימו בעזרת ה Wireshark כיצד הדבר בא לידי ביטוי בחבילות שנשלחו בתוכנית וציינו באיזה שכבה.

4. הדגימו בעזרת ה Wireshark מאיזה כתובת IP ולאיזה כתובת IP נשלחו החבילות שתפסתם. השוו כתובות אלו לכתובת ה IP של הכרטיס שלכם כפי שמופיעה בפלט הפקודה ipconfig/ifconfig.

2 חלק ב

כעת נממש מעין שירות 144 דיגיטלי. המטרה שלנו היא לבנות שרת שיודע לקבל שאלות ולהשיב עליהן. בשירות 144 הטלפוני, ניתן לשאול למשל: "מה מספר הטלפון של ישראל ישראלי?". שירות 144 יסתכל ברשומות שלו ויענה: "מספר הטלפון של ישראל ישראלי הוא 1234567 – 03". אנחנו רוצים לבנות משהו דומה, אבל במקום מספרי טלפון, עם כתובות IP, ובמקום שמות, כתובות אינטרנט.

עליכם לבנות שרת בפרוטוקול UDP. לפני שהשרת מתחיל להאזין לסוקט, עליו לטעון קובץ בשם ips.txt אשר מכיל את המיפויים שהוא מכיר (מיפויים בין כתובות אינטרנט לכתובות IP). למשל, להלן דוגמה לתוכן אפשרי של קובץ ips.txt:

biu.ac.il,1.2.3.4,180

google.co.il,1.2.3.5,240

שימו לב שבכל שורה ישנו רק מיפוי אחד, ובכל שורה שלושה "שדות": כתובת אינטרנט, כתובת IP ו-TTL, אשר מופרדים ביניהם בפסיק.

השרת מצפה לקבל הודעות אשר יכילו את כתובת האינטרנט, ועל כל הודעה כזאת, הוא מחזיר את שורת המידע המתאימה, כלומר, כתובת האינטרנט, כתובת ה IP וה TTL.

במידה והשרת מקבל כתובת אינטרנט שהוא אינו מכיר את ה IP שמשייך לה, הוא מפנה את השאלה לשרת ה"אב" שלו. כתובת ה IP של שרת האב תועבר כפרמטר לתכנית. השרת מפנה את השאלה בדיוק באותו הפורמט לשרת האב, מקבל את התשובה ממנו, "לומד" אותה, ועונה ללקוח.

המשמעות של "למידה" היא, שמעתה והלאה, לקוחות שישאלו את השרת את אותה השאלה יקבלו מיד את התשובה - כי השרת כבר יודע אותה. אין לשרת צורך שוב ללכת לשאול את שרת ה"אב". בכל פעם שהשרת לומד מיפוי חדש, השרת שומר את כל המיפויים שהוא מכיר לקובץ ips.txt כלומר, גם המיפויים החדשים שלמד ישמרו לקובץ, ובכך, אם השרת ייסגר ויופעל מחדש, יהיה לו את כל ה"זיכרון" שהיה לו בפעם הקודמת, כולל הדברים החדשים שלמד.

עם זאת, יש לקחת בחשבון את ה TTL. כלומר, נניח ומיפוי הגיע עם TTL של 180. אזי, אם השרת יישאל על הדומיין הזה ב180 השניות הקרובות, הוא משיב את התשובה שלמד. אחרת, עם עברו יותר מ 180 שניות, עליו לשכוח את המיפוי הזה וללמוד אותו מחדש.

בנוסף, כתבו לקוח אשר מקבל בלולאה קלט מהמשתמש (כתובת אינטרנט), ומציג ללקוח את כתובת ה IP המתאימה בעזרת השרתים הנ"ל. השרת מקבל 4 ארגומנטים כקלט לתוכנית:

[myPort] [parentIP] [parentPort] [ipsFileName]

הלקוח מקבל 2 ארגומנטים כקלט לתוכנית:

[serverIP] [serverPort]

דוגמת קלט פלט: תוכן קובץ מיפויים לשרת האב parent.txt:

www.google.co.il,8.8.8.8,300

mail.google.co.il,9.9.9.9,240

תוכן קובץ מיפויים לשרת הרגיל ips.txt:

www.biu.ac.il,1.2.3.4,180

mail.biu.ac.il,1.2.3.5,240

biu.ac.il,1.2.3.4,180

דוגמא להרצת שרת אב: (שימו לב שבגלל שהוא שרת אב, ואין לו אב מעליו, כתובת ה IP והפורט של האבא שלו אינם רלוונטיים ולכן אני מעביר את הפרמטרים -1)

```
server.py 55555 -1 -1 parent.txt
```

דוגמא להרצת שרת רגיל:

```
server.py 12345 127.0.0.1 55555 ips.txt
```

דוגמא להרצת לקוח (אם השרת לא רץ על אותה מכונה כמו הלקוח, כתובת ה IP צריכה להשתנות בהתאם)

:

```
client.py 127.0.0.1 12345
```

דוגמא למסך הפלט של הלקוח (השרת אינו מדפיס פלט): מה שמופיע אחרי ה// הוא הערה רק פה להסבר ואינו צריך להופיע בפלט שלכם

```
mail.biu.ac.il // user input
```

```
1.2.3.5
```

```
www.biu.ac.il // user input
```

```
1.2.3.4
```

```
mail.google.co.il // user input
```

```
9.9.9.9
```

הריצו את התכנית באופן הבא: הריצו שרת אב, שרת רגיל ולקוח. עליכם להריץ את התוכנית בשתי מכונות שונות: אחת עבור השרתים ואחת הלקוחות. אפשר להריץ בעזרת מכונה וירטואלית אחת, אבל אז השרתים צריכים לרוץ על המכונה הוירטואלית והלקוח על מ"ה המארחת או להיפך. הראו את הפלט המתקבל, אבל בעיקר - תפסו את התעבורה ב Wireshark, הראו והסבירו את החבילות שהתכנית שולחת והציגו את החבילות והשכבות השונות והסבירו בפירוט בעזרת צילומי מסך מתוך Wireshark כיצד חלקי התכנית שלכם מתבטאים בתעבורה שתפסתם ב Wireshark. כלומר, להראות את ההודעות שהלקוח שלח כיצד הן נבנו בכל השכבות, כנ"ל ההודעות שהשרתים שלחו וכו'. שימו לב: המרכיב העיקרי בציון על שאלה זו הוא החלק של Wireshark. שמרו את החבילות שתפסתם לקובץ pcap בשם trace2 והגישו גם אותו.

3 מה יש להגיש?

1. דו"ח המכיל תשובות לכל השאלות ולכל הסעיפים. דו"ח זה יהיה בקובץ PDF בלבד. כל פורמט אחר לא יתקבל (גם לא word). דוח שיוגש בפורמט שאינו PDF ציונו יהיה 0.

2. קבצי ה pcap ששמרתם.

3. קבצי השרת והלקוח שכתבתם בחלק ב, עם תיעוד בסיסי. על הקבצים להיות בשמות client.py, server.py חובה על הקבצים לרוץ על שרתי האוניברסיטה.

4. קובץ טקסט בשם details.txt עם שמות ות.ז. של המגישים ואיזה פייתון השתמשתם (2 או 3). שימו לב, על הקובץ חובה להיות בפורמט הבא:

```
Israel Israeli 123456789
```

```
Israela Israeli 012345678
```

- בלי רווחים נוספים, בלי שורות נוספות, ובשפה האנגלית בלבד. השורה השלישית מכילה רק 2 או 3 בלי שום דבר אחר. אי הגשה של קובץ ה details.txt הנ"ל, תגרור הורדה של 20 נקודות בציון התרגיל
5. ניתן להגיש בזוגות אך לא חובה. אם מגישים בזוג, שרק אחד מבני הזוג יגיש את התרגיל (כי יש את סעיף 4 הנ"ל)
6. יש להגיש את כל הנ"ל למודל וגם לסאבמיט
7. במידה ומגישים זיפ, מותר רק זיפ רגיל ולא פורמטים אחרים, למשל, אסור rar
8. על כל הקבצים להיות באותה תיקייה וללא תיקיות משנה
9. חובה להגיש את הקבצים עצמם בלבד ולא קישור לקבצים. קחו בחשבון שבמודל יש מגבלת העלאה לקבצים, ולכן היערכו לזה מראש ולא ברגע האחרון.

בהצלחה