

## מבוא לרשותת תקשורת

### תרגיל מס' 2

#### חלק א:

ב חלק זה, יש להריץ מס' 2 Wireshark, ולנתח את התעבורה המתקבלת ב Wireshark. את כל הפעולות ב חלק זה יש להריץ כאשר הלקוח והשרת נמצאים על מחשבים שונים. זה יכול להיות שני מחשבים שונים פיזית וזה יכול להיות שני מחשבים וירטואליים שונים, או אחד מהם רץ על מכונה וירטואלית והשני על המחשב עצמו (זה שמריצ' את המכונה הווירטואלית). אבל חשוב שהם ירצו כל אחד על מחשב נפרד.

שימוש לב, יתכן מאוד שסטודנטים שונים יראו תעבורת שונה ב Wireshark והתנווגות שונה בהרצת הסקריפט, כתוצאה מהבדלי חומרה ותוכנה והאפן בו אתם מരיצים. זה תיקן לחנותין. כל סטודנט או זוג צרייכים לנתח את התעבורה בהתאם למה שם קיבלו.

בעזרת קוד השרת והלקוח הפשטיטים ממצגת התרגול בпрוטוקול TCP, אשר גם יופיעו במודל לצד תרגיל זה, הריצו את השרת בעזרת פורט לבחירתכם והריצו לקוח אחד בהתאם.

התאמו את הקוד כך שהלקוח ישלח את שמו כהודעה.

לאחר שהשרת השיב את תשובתו, התאמו את הקוד כך שהלקוח ישלח הודעה נוספת עם מספר תעודת זהות שלכם. על ההודעות הללו להישלח על גבי אותו חיבור, ולא חיבורים נפרדים. כמובן, אין לסגור ולפתח מחדש סוקט בין הודעות אלו.

תפסו את התעבורה ב Wireshark, והשתמשו בתעבורה שתפסתם כדי לתאר ולהסביר ולהציג את תהליך החיבור של TCP ( מבחינת הודעות שנשלחו והתקבלו ותפקידן, וכן כיצד המידע בא לידי ביתוי בתחילת TCP).

בנוסף, הסבירו מה היו המספרים הסידוריים לאורץ החיבור הן מבחינת sequence number והן מבחינת acknowledgement number.

כלומר, הראו את המספרים בעת הקמת החיבור וסיוומו לאורץ החיבור עצמו כאשר נשלחות הודעות ומתקבלים ack-ים. התייחסו למספריים האמיטיים בכל חיבור ולא ליחסים.

כעת, עליים להריץ את הקבצים המצורפים לתרגיל. כלומר, במודל, בנוסף למסמך זה, תמצאו קבצי שרת ולקוח בוarieatzot שונות. ישן 4 גרסאות: V1, V2, V3, V4. עברו כל גרסאות שרת ולקוח, עליים להסתכל על הקוד ולהסבירו בקצרה באופן מילולי מה הוא עושה. יש להריץ את הקוד ולתפוס את התעבורה בwireshark ולהסביר אותה. **אין צורך** להיכנס לפירוט כמו בסעיף הקודם מבחינות מספריים סיידוריים וכדו. עם זאת, עליים להראות ולהסביר מה רואים בתעבורה ומדוע אנחנו רואים זאת. **התמקדו בחבילות או בהתנהגות שונה ממה שראיתם בהרצאת הגראסאות שקדמו לגרסה הנוכחית.** למשל, כאשר אתם מרכיבים את גרסה 2 V אתם צפויים לראות קוד ותעבורה מעט שונה مما שראיתם כאשר הריצתם את הקוד שופיע בV1.

**התמקדו בשינויים אלו** ובעיקר הסבירו כיצד השינויים בקוד משתקפים בתעבורה. הריצו את הקוד מספר פעמים ובמידה והתוצאה שונה בין הריצות הסבירו מדוע.

יש להסביר הכל בדוח' כולץירוף של צילומי מסך מתאימים מwireshark.  
**יש לצרף את קבצי ה pcap (או pcapng) של ההרצות להגשה.**

### חלק ב:

עליכם למשרף TCP המתפקיד באופן הבא:

הلكוח שולח לשרת שם של קובץ שהוא מעוניין להוריד ממנו (כלומר, שהשרת ישלח לו בחזרה). הקבצים יושבים בתוך תיקייה בשם files אשר נמצאת באותו תיקייה שבה נמצא השרת. שם הקובץ יכול לכלול גם נתיב. כלומר, אם הלקוח שלח רק את שם הקובץ, אז הקובץ צריך להיות בתוך התיקייה files בرمאה העליונה. במידה ושם הקובץ מכיל גם נתיב תיקייה, השרת מחפש את הקובץ בהתאם לנתיב בתוך התיקייה files.

הפורמט שבו הלקוח שולח לשרת הוא הפורמט הבא:  
בשורה הראשונה כתוב:

GET [file] HTTP/1.1

כאשר במקום [file] יהיה כתוב שם הקובץ. שמו לב, כאשר כתוב שורה - הכוונה היא שורה ממש, ולכן מיד בסופה מופיע הסימן \

נגידר שבמקרה ושם הקובץ הוא התו הבודד / (סלאש), אזי הכוונה לקובץ בשם index.html  
הלקוח שולח سورות נוספות בהודעה, אך על השרת שלכם להתעלם מרובן, כפי שיווגדר בהמשך.  
הלקוח סיים לשלוח את ההודעה כאשר הוא ישלח פעמיים שורה חדשה, ככלומר \r\n\r\n

אם הקובץ קיים, השרת יחזיר:

HTTP/1.1 200 OK

Connection: [conn]

Content-Length: [length]

ואז שורה ריקה, ואז את תוכן הקובץ.

כאשר, במקומות [conn] יהיה רשום הערך של השדה connection שהופיע בבקשת מהלקוח, ובמקומות [length] יופיע גודל הקובץ הנשלח. שימוש לב, length הוא כמוות הבטים שנשלחים בפועל.  
למשל, הלוקוח שלח:

GET / HTTP/1.1

...

Connection: close

...

(הלקוח שלח מידע נוסף - מסומן על ידי שלוש נק', אך הוא לא רלוונטי כי השרת שלנו מתעלם ממנו).

והשרת שלח בחזרה את התשובה הבאה ביחד עם התוכן של הקובץ :index.html

HTTP/1.1 200 OK

Connection: close

Content-Length: 11

hello world

(שימוש לב, כאן מדובר על קובץ index.html שהכיל רק את התוכן hello world (בתיקיית files במודול זה יהיה קובץ index.html עם תוכן אחר ובודיקה כמובן קבצים אחרים)

במידה והערך של השדה connection יהיה close (כמו בדוגמה לעיל), סגורים את החיבור לאחר שליחת הקובץ ומטפלים בחיבור הבא. לעומת זאת, אם הערך הוא keep-alive, יש להשאיר את החיבור פתוח - ולקראת את בקשת הקובץ הבאה של הלוקוח חלק מאותו חיבור.

**שים לב - זה דרישת מהותית בתרגילים. חובה למשמש אותה וחובה להציג את זה בפעולה ב wireshark. אי מימוש חלק זה יגרור הורדה של 50 נק'.**

麥יוון שאין לנו שליטה על הלוקוח בתרגיל הזה (פרטים בהמשך) - אין לנו אפשרות להגביל את כמות החיבורים שהוא יפתח מול השירות. אבל, מכיוון שהשרות שלנו אינו משתמש במקביליות **ואסוו** לו להשתמש במקביליות, נגידר לsocket פרק זמן מקסימלי שבו הוא נתקע על recv (חפשו בגוגל על timeout בהקשר של סוקט TCP בפייתו).

במידה והה\_recv לא מקבל תשובה אחרי 1 שניות, יש לסגור את החיבור הנוכחי בשרת ולטפל בלוקוח הבא (חיבור חדש). לחילופין, אם השירות מקבל "בקשת ריקות" מהлокוח (כלומר אפס בתים חררו מה\_recv), גם במקרה זה יש לסגור את החיבור הנוכחי בשרת ולטפל בלוקוח הבא (חיבור חדש).

**זאת ההתנהגות לכל סוגי הקבצים, למעט קבצים עם סיומת jpg או ico. במקרה שכזה, יש לקרוא את תוכן הקובץ בשרת בצורה בינארית ואז לשלוח אותו. למשל:**

HTTP/1.1 200 OK  
Connection: keep-alive  
Content-Length: [length]

[binary image data]

אם הקובץ לא קיים, השירות מחזיר:

HTTP/1.1 404 Not Found  
Connection: close

אם הלוקוח בקש קובץ בשם GET /redirect HTTP/1.1

השרת מחזיר בחזרה:

HTTP/1.1 301 Moved Permanently

Connection: close

Location: /result.html

(יש להקפיד להחזיר בתשובה שורה ריקה אחרי השורה של location. ככלומר שורה ריקה ממש ולא רק שורה חדשה, כנ"ל במקרה של 404).

שים לב, במקרה של 404 ו 301 יש תמיד לחזיר:

Connection: close

לא משנה מה הופיע בשדה connection שהגיע מהקליינט, ולסגור את החיבור ולהמשיך אל החיבור הבא.

- בנוספ', על השרת להציג למסך את הבקשות שהוא קיבל מהלקוח. כל הבקשה יכולה **אבל ללא שום תוספת**.

כעת, פתחו דפדפן, והקלידו בשורת הכתובת של הדפדפן את הדבר הבא:

http://[Server IP]:[Server port][Path]

כלומר, כתובות ה IP של השרת, נקודותיים, ואז הפורט שהשרת שלכם מازין לו ואז הנטייב של הקובץ. למשל:

http://1.2.3.4:80/

שורה זו פונה לשרת שנמצא בכתובת 1.2.3.4 ומאזין לפורט 80 ובקשת את הנטייב /  
(כפי שהוגדר לעיל)

השרת שלכם מקבל **ארגון** **לmain** רק ארגומנט אחד - הפורט אליו הוא מازין (כפי  
שהיה בתרגילים קודמים)

אם אתם מרים את השרת על המחשב שלכם, אפשר בדף במקומות כתובות IP  
הLocale (127.0.0.1) לכתוב localhost.

במודל תמצאו תיקייה files לדוגמא.  
בבדיקה, התרגיל ייבדק מול תיקייה אחרת.

לבסוף, עליוכם למשך לקוח ב 3 Python, אשר מקבל מה console נתיב כלשהו (בלולאה אינסופית), פונה אל השרת שלכם וմבקש בקשת GET עבור הנתיב זהה. עליוכם לחוץ מהתשובה של השרת את **המimetype** ולשמור אותו באותה תיקייה עם אותו שם (השם של הקובץ בנתיב, לא כל הנתיב כולם). למשל, אם יוזן:

/images/foo.jpg

עלויים יהיה לבקש בקשת Get מהשרת עבור נתיב זהה, וליצור קובץ בשם foo.jpg באותה התיקייה אשר יוכל את התמונה שהשרת שלח. **את התמונה בלבד** ולא את שאר המידע שהשרת שלח (למשל, בלי ה headers). הקובץ צריך להיות תקין. ככלומר, פתיחה של הקובץ (תמונה במקרה זהה) תפתח את התמונה תקין ותציג אותה. לדוגמה עלויים להדפיס **רק** את השורה הראשונה של response שהגיע מהשרת ולקЛОוט את הנתיב הבא. הלקוח יקבל ארגומנטים ל함ן main את הקוד של השרת ולאחר מכן את הפורט שהשרת מציין בו.

אסור להשתמש בספריות **בכל** בתרגיל (כל ספרייה או מודול אסור), למעט:

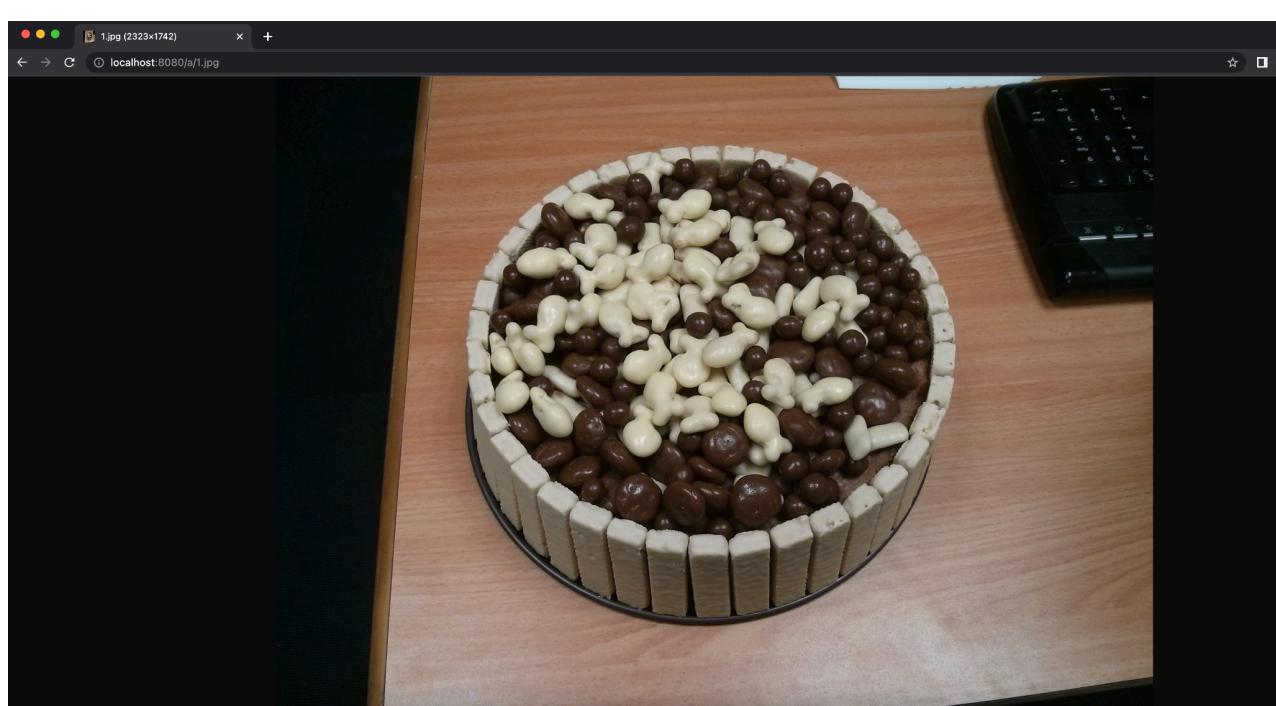
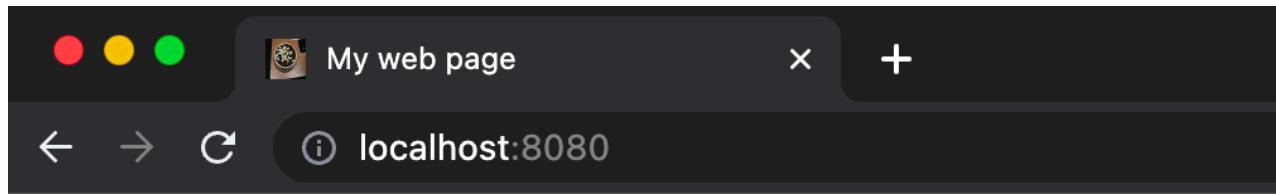
- הספרייה socket.
- הספרייה sys, רק בשbill לחלץ את ארגומנט הפלט.
- הספרייה os.

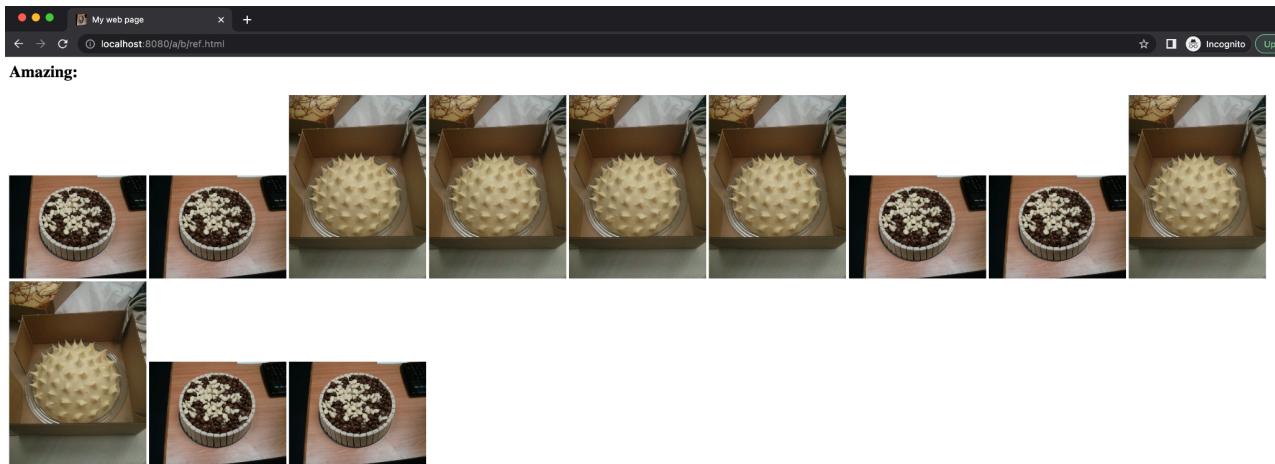
בנוסף לכך, יש להציגם בדוח' המוגש את הרצת התרגיל תוך שימוש ב wireshark והסביר של התעבורה המתרחשת. שימו לב, אין צורך להיכנס לפרטים ולהסביר כל אחת מהחבילות בנפרד או את השדות של כל אחת מהחבילות. הדגש העיקרי הוא לנתח ולהסביר באופן כללי את התעבורה, בדgesch על:

- כמה חיבורים היו?
- מה הועבר בכל חיבור?
- באילו חיבורים נשלחו יותר מבקשת אחת מהלקוח?
- מה קרה כשהזון הקישור לדפדף (מה נשלח לשרת, מה השרת החזיר)
- כאשר הדפדף קיבל תשובה מהשרת - האם הוא שלח בקשה נוספת? איזה?

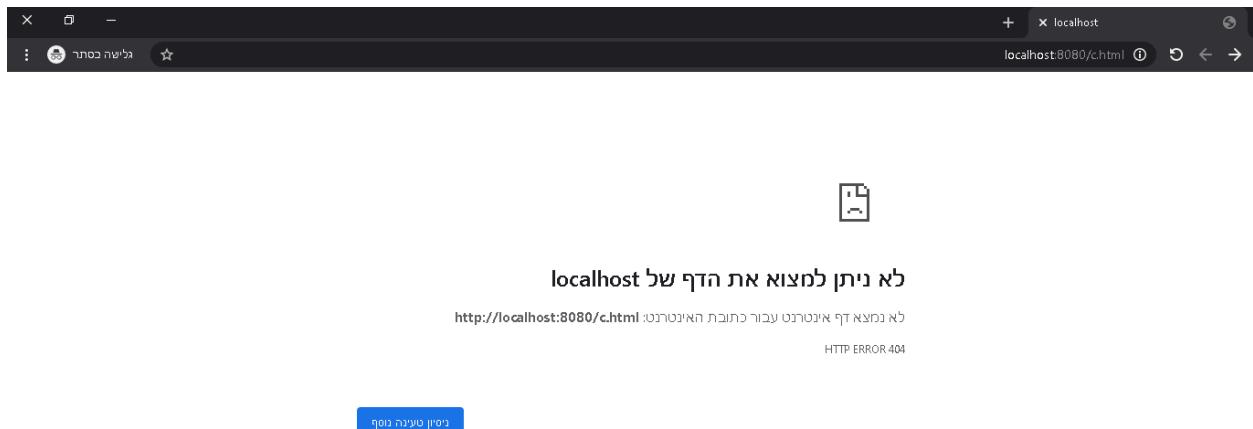
**אבל**, חובה להסביר ממש עם צילומי מסך מתאימים - ולא רק לענות בצורה קצרה ויבשה.

להלן מספר דוגמאות להרצת התרגיל בעזרת תיקייה files אשר במודול:





דוגמא לקובץ שלא קיים:



דוגמא לגלישה redirect:/ (הדף שינה את הכתובת בהתאם לוגיקה שהוגדרה לעיל):



### מה יש להגיש?

1. דוח המכיל תשובות לכל החקקים ולכל הסעיפים. דוח זה יהיה בקובץ PDF בלבד. כל פורמט אחר לא יתקבל (גס לא word). **דוח שיוגש בפורמט שאינו PDF ציינו יהיה 0.**
2. קבצי pcapng ששמरתם. למשל, קבצי pcapng של הרצת התרחישים המופיעים בחלק א וכן של הרצת והדגם חלק ב. **יש לסנן ולהציג את התעבורה בהתאם לכך שתכיל רק את החבילות של התרגילים.**
3. קבצי השרת והלקוח שכתבتم בחלק ב, עם תיעוד בסיסי. על השרת להיות בשם server.py בלבד ועל הלוקוח להיות בשם client.py ולא שום קבצי Python נוספים. יש לכתוב בפייתון גירסה 3 בלבד.
4. קובץ טקסט בשם details.txt עם שמות ות.ז. שימו לב, חובה על הקובץ להיות בפורמט הבא:

Israel Israeli 123456789

Israela Israeli 012345678

בלי רווחים נוספים, בלי שורות נוספות, ובשפה האנגלית בלבד.

**אי הגשה של קובץ ה txt הנ"ל, או הגשתו באופן שונה מאייך שהוגדר פה,**  
**תגרור הורדה של 20 נקודות בצדון 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 להגשים את כל הנייר למודל

7. במידה ומגישיםZIP, חובה שהקובץ יהיה ZIP ולא שום פורמט אחר. למשל אסור RAR.  
בנוסף חובה לוודא לאחר ההגשה שאתם מצליחים להוריד את הקובץ שהונחתם מהמודל  
ולעשות לו ZIPWHEN וshall מה שהונחתם תקין.

8. על כל הקבצים להיות באותה תקינה ולא תיקיות משנה

9. חובה להגשים את הקבצים עצם בלבד ולא קישור לקבצים. קחו בחשבון שבמודל יש  
מגבלה העלה לקבצים, ולכן היררכיו זהה מראש ולא ברגע האחרון.

10. שאלות יש לשלוח בפורום בלבד. חובה לבדוק שהשאלה שלכם לא עונתה כבר טרם  
שליחת השאלה.

### **בהצלחה**