

מבוא לרשתות תקשורת

תרגיל מספר 2

להגשה עד ל 13.12 בשעה 22:00

חלק א:

בחלק זה, יש להריץ מספר קטעי קוד, ולנתח את התעבורה המתקבלת ב wireshark. את כל הסעיפים בחלק זה יש להריץ כאשר הלקוח והשרת נמצאים על מחשבים שונים. זה יכול להיות שני מחשבים שונים פיזית וזה יכול להיות שני מחשבים וירטואלים שונים, או שאחד מהם רץ על מכונה וירטואלית והשני על המחשב עצמו (זה שמריץ את המכונה הוירטואלית). אבל חובה שהם ירוצו כל אחד על מחשב נפרד. שימו לב, ייתכן מאוד שסטודנטים שונים יראו תעבורה שונה ב wireshark והתנהגות שונה בהרצת הסקריפט, כתוצאה מהבדלי חומרה ותוכנה והאופן בו אתם מריצים. זה תקין לחלוטין. כל סטודנט או זוג צריכים לנתח את התעבורה בהתאם למה שהם קיבלו.

בעזרת קוד השרת והלקוח בפרוטוקול TCP מהמודל, הריצו את השרת בעזרת פורט לבחירתכם והריצו לקוח אחד בהתאם. התאימו את הקוד כך שהלקוח ישלח את שמכם כהודעה. לאחר שהשרת השיב את תשובתו, התאימו את הקוד כך שהלקוח ישלח הודעה נוספת עם מספר תעודת הזהות שלכם. תפסו את התעבורה ב Wireshark, והשתמשו בתעבורה שתפסתם כדי לתאר ולהסביר ולהדגים את תהליך החיבור של TCP (מבחינת הודעות שנשלחו והתקבלו ותפקידן, וכיצד המידע בא לידי ביטוי בתחילת TCP). בנוסף, הסבירו מה היו המספרים הסידוריים לאורך החיבור הן מבחינת sequence number והן מבחינת acknowledgement number. כלומר, הראו את המספרים בעת הקמת החיבור וסיומו ולאורך החיבור עצמו כאשר נשלחות הודעות ומתקבלים ack-ים. התייחסו למספרים האמיתיים בכל חיבור ולא ליחסיים.

כעת, עליכם להריץ את הקבצים המצורפים לתרגיל. כלומר, במודל, בנוסף למסמך זה, תמצאו קבצי שרת ולקוח בואריאציות שונות.

ישנן 4 גרסאות: V1, V2, V3, V4.

עבור כל גרסאת שרת ולקוח, עליכם להסתכל על הקוד ולהסבירו בקצרה באופן מילולי מה הוא עושה. יש להריץ את הקוד ולתפוס את התעבורה ב wireshark ולהסביר אותה. אין צורך להיכנס לפירוט כמו בסעיף הקודם מבחינת מספריים סידוריים וכדו'. עם זאת, עליכם להראות ולהסביר מה רואים בתעבורה ומדוע אנחנו רואים זאת. התמקדו בחבילות ו/או בהתנהגות שונה ממה שראיתם בהרצת הגרסאות שקדמו לגירסה הנוכחית. למשל, כאשר אתם מריצים את גרסה V2 אתם צפויים לראות קוד ותעבורה מעט שונה ממה שראיתם כאשר הרצתם את הקוד שמופיע ב V1. התמקדו בשינויים אלו ובעיקר הסבירו כיצד השינויים בקוד משתקפים בתעבורה.

יש להסביר הכל בדו"ח כולל צירוף של צילומי מסך מתאימים מ wireshark. יש לצרף את קבצי ה pcap של ההרצות להגשה.

חלק ב:

עליכם לממש שרת TCP המתפקד באופן הבא:

הלקוח שולח לשרת שם של קובץ שהוא מעוניין להוריד ממנו (כלומר, שהשרת ישלח לו בחזרה).

הקבצים יושבים בתוך תיקייה בשם files אשר נמצאת באותה תיקייה שבה נמצא השרת. שם הקובץ יכול לכלול גם נתיב. כלומר, אם הלקוח שלח רק את שם הקובץ, אזי הקובץ צריך להיות בתוך התיקייה files ברמה 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.

שימו לב, כאשר כתוב שורה - הכוונה היא שורה ממש, ולכן מיד בסופה מופיע הסימן \n נגדיר שבמידה ושם הקובץ הוא התו הבודד / (סלאש), אזי הכוונה לקובץ בשם

index.html

הלקוח שולח שורות נוספות בהודעה, אך על השרת שלכם להתעלם מרובן, כפי שיוגדר בהמשך.

הלקוח סיים לשלוח את ההודעה כאשר הוא ישלח פעמים שורה חדשה, כלומר \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, במקום [conn] יהיה רשום הערך של השדה connection שהופיע בבקשה מהלקוח, ובמקום [length] יופיע גודל הקובץ הנשלח. שימו לב, length הוא כמות הבתים שנשלחים בפועל. למשל, הלקוח 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

במידה והערך של השדה connection יהיה close (כמו בדוגמא לעיל), סוגרים את החיבור לאחר שליחת הקובץ ומטפלים בחיבור הבא. לעומת זאת, אם הערך הוא keep-alive, יש להשאיר את החיבור פתוח - ולקרוא את בקשת הקובץ הבאה של הלקוח כחלק מאותו חיבור.

שימו לב - זוהי דרישה מהותית בתרגיל. חובה לממש אותה וחובה להדגים את זה בפעולה ב wireshark. אי מימוש חלק זה יגרור הורדה של 50 נק'.

מכיוון שאין לנו שליטה על הלקוח בתרגיל הזה (פרטים בהמשך) - אין לנו אפשרות להגביל את כמות החיבורים שהוא יפתח מול השרת. אבל, מכיוון שהשרת שלנו אינו משתמש במקביליות ואסור לו להשתמש במקביליות, נגדיר ל socket פרק זמן מקסימלי שבו הוא נתקע על recv (חפשו בגוגל על timeout בהקשר של סוקט TCP בפייתון). במידה וה recv לא מקבל תשובה אחרי 1 שניות, יש לסגור את החיבור הנוכחי בשרת ולטפל בלקוח הבא (חיבור חדש). לחילופין, אם השרת מקבל בקשות ריקות מהלקוח, גם במקרה זה יש לסגור את החיבור הנוכחי בשרת ולטפל בלקוח 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, השרת מחזיר:

```
HTTP/1.1 404 Not Found
Connection: close
```

אם הלקוח ביקש 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. כלומר שורה ריקה ממש ולא רק שורה חדשה, כנ"ל במקרה של 404).
שימו לב, במקרה של 404 ו 301 יש תמיד להחזיר:

Connection: close

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

בנוסף, על השרת להדפיס למסך את הבקשות שהוא קיבל מהלקוח. כל הבקשה כולה - אבל ללא שום תוספת.

בתרגיל זה, אינכם כותבים לקוח. בתרגיל זה תשתמשו בלקוח קיים, והוא - הדפדפן שלכם (chrome). עליכם להקליד בשורת הכתובת של הדפדפן את הדבר הבא:

[Server IP]:[Server port][Path]

כלומר, כתובת ה IP של השרת, נקודותיים, ואז הפורט שהשרת שלכם מאזין לו ואז הנתבי של הקובץ. למשל:

1.2.3.4:80/

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

השרת שלכם מקבל כארגומנט לmain רק ארגומנט אחד - הפורט אליו הוא מאזין (כפי שהיה בתרגילים קודמים)

אם אתם מריצים את השרת על המחשב שלכם, אפשר במקום כתובת ה IP המקומית (127.0.0.1) לכתוב localhost.

במודל תמצאו תיקיית files לדוגמא.

בבדיקה, התרגיל ייבדק מול תיקייה אחרת.

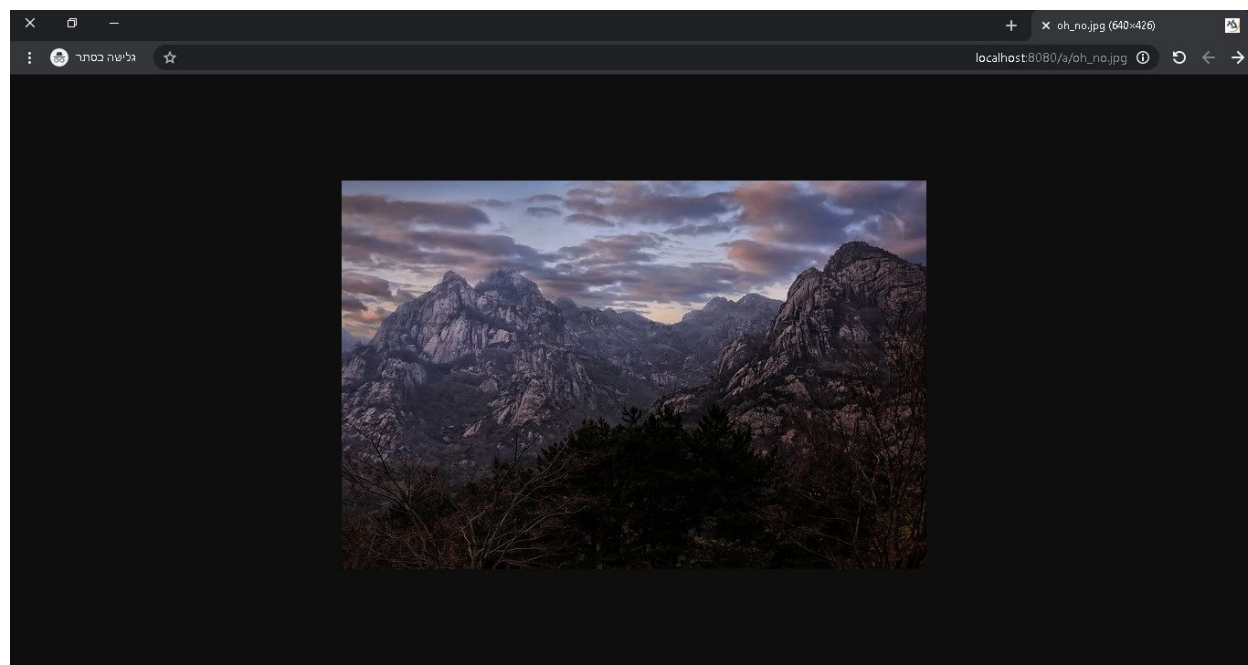
אסור להשתמש בספריות **בכלל** בתרגיל (כל ספרייה או מודול אסור), למעט:

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

בנוסף לקוד, יש להדגים בדו"ח המוגש את הרצת התרגיל תוך שימוש ב wireshark והסבר 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, בדגש על:

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

להלן מספר דוגמאות להרצת התרגיל בעזרת תיקיית ה files אשר במודל:

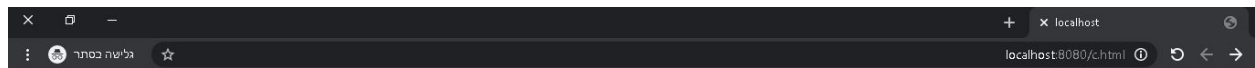




Amazing:



דוגמא לקובץ שלא קיים:



לא ניתן למצוא את הדף של localhost

לא נמצא דף אינטרנט עבור כתובת האינטרנט: http://localhost:8080/c.html

HTTP ERROR 404

ניסיון טעינה נוסף

דוגמא לגלישה /redirect (הדפדפן שינה את הכתובת בהתאם ללוגיקה שהוגדרה לעיל):



מה יש להגיש?

1. דו"ח המכיל תשובות לכל החלקים 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. כלומר, קבצי pcap של הרצת התרחישים המופיעים בחלק א וכן של הרצה והדגמת חלק ב. יש לסנן ולייצא את התעבורה בהתאם כך שתכיל רק את החבילות של התרגיל.
3. קובץ השרת שכתבתם בחלק ב, עם תיעוד בסיסי. על הקובץ להיות בשם server.py בלבד וללא קבצים נוספים.
4. קובץ טקסט בשם details.txt עם שמות ות.ז. של המגישים ואיזה פייתון השתמשו (2 או 3). שימו לב, על הקובץ חובה להיות בפורמט הבא:

Israel Israeli 123456789

Israela Israeli 012345678

2

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.

10. להזכירכם, אין לשאול שאלות שקשורות למימוש, כמו למשל, "איך לממש X" או "האם מימוש Y הוא נכון".

11. שאלות יש לשלוח אלי במייל. במודל יהיה קובץ שאלות ותשובות שיעודכן מדי פעם. באחריותכם להתעדכן בקובץ, שכן כל הנכתב בו מחייב את כולם. חובה לבדוק שהשאלה שלכם לא נענתה כבר בקובץ טרם שליחת השאלה.

בהצלחה