

תרגיל מספר 2

להגשה עד ל 11.12 בשעה 00 : 22 דרך המודל בלבד

1 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. הרצה על אותו מחשב מטרמינלים שונים כמו בתרגיל 1 לא תתקבל. שימו לב, ייתכן מאוד שסטודנטים שונים יראו תעבורה שונה ב Wireshark, כתוצאה מהבדלי חומרה ותוכנה והאופן בו אתם מריצים. זה תקין לחלוטין. כל סטודנט או זוג צריכים לנתח את התעבורה בהתאם למה שהם קיבלו.

1. בעזרת קוד השרת והלקוח בפרוטוקול TCP מהמצגת, הריצו את השרת בעזרת פורט לבחירתכם והריצו לקוח אחד בהתאם. שילחו את שמכם כהודעה. לאחר שהשרת השיב את תשובתו שילחו הודעה נוספת עם מספר תעודת הזהות שלכם. תפסו את התעבורה ב Wireshark, והשתמשו בתעבורה שתפסתם כדי לתאר ולהסביר ולהדגים את תהליך החיבור (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 ackים. התייחסו למספרים האמיתיים בכל חיבור ולא ליחסיים.

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

2 חלק ב

בחלק זה נרצה לממש רשת פשוטה לשיתוף קבצים. לשם כך, נממש שני חלקים: שרת (Server) ולקוח (Client) בעזרת TCP בלבד. השרת שלנו פותח סוקט ומאזין על פורט אותו יקבל כארגומנט לתוכנית (ארגומנט ל"main"). שימו לב, ברשת זו השרת רק נועד בכדי לחבר בין הלקוחות. העברת הקבצים בפועל תהיה ישירות בין הלקוחות (כפי שיובהר בהמשך).

השרת יכול לקבל את ההודעות הבאות:

1. התחברות: לקוח השולח הודעה זאת, מעוניין להצטרף. ההודעה תהיה בפורמט הבא:

1 [Port] [Files]

כאשר:

- Port הוא הפורט בו יאזין הלקוח לבקשות העברת קבצים.
- Files היא רשימת קבצים מופרדת בפסיקים אשר הלקוח מעוניין לשתף.

השרת אינו שולח בחזרה כלום ללקוח.

2. חיפוש: לקוח מעוניין לחפש לקוחות שיש להם קובץ מסויים. ההודעה תהיה בפורמט הבא:

2 [Search]

כאשר:

- Search הוא ביטוי שיש לחפש. זה יכול להיות שם הקובץ, הסימנים שלו, גם וגם או רק חלק מהם.

השרת שולח בחזרה ללקוח רשימה של כל הקבצים ופרטי הלקוחות שאפשר למצוא אצלם את הקבצים.

הרשימה תהיה בפורמט הבא ותסתיים בתו ירידת שורה: [Name] [IP] [Port]\n, ..., [Name] [IP] [Port]

השרת רץ בלולאה אינסופית וממתין ללקוחות. כל לקוח שמתחבר יכול לבקש בקשה אחת בלבד. כל בקשה

תסתיים בתו ירידת שורה \n.

במידה והלקוח שלח הודעה לשרת שאינה לפי מה שהוגדר לעיל, על השרת להתעלם מההודעה. שימו לב: ישנם

מקרי קצה שונים, אנא ודאו שהשרת והלקוח ערוכים אליהם.

יש להקפיד על הפלט כפי שמוצג פה, כולל אותיות גדולות וקטנות ורווחים.

השרת שתואר לעיל מטפל בלקוח אחד בלבד בו זמנית (אין מקביליות).

אנחנו רוצים שהלקוח שלנו יוכל מצד אחד לאפשר למשתמש להוריד קבצים ומצד שני ללקוחות אחרים להוריד

קבצים מאיתנו. מכיוון שאני מניח שלא כל הסטודנטים עדיין למדו מקביליות בקורסים האחרים, אנחנו נממש

זאת באופן הבא. תוכנית הלקוח תופעל פעמיים. כאשר הפרמטר הראשון שיוזן לתוכנית יהיה הערך 0, התוכנית

תהיה במצב האזנה. כאשר הפרמטר הראשון שיוזן לתוכנית יהיה הערך 1, התוכנית תהיה במצב משתמש. בנוסף,

הלקוח מקבל עוד שני פרמטרים: את כתובת ה IP ואת הפורט של השרת. התוכנית במצב האזנה תקבל גם

פרמטר רביעי - שהוא מספר הפורט להאזנה.

בעת ההרצה, הלקוח במצב האזנה:

1. מודיע לשרת שהוא מצטרף לרשת, ומודיע לו את רשימת הקבצים שהוא מוכן לשתף, שהם כל הקבצים

שנמצאים בתיקיה בו נמצאת התוכנית.

2. ממתין לבקשות להעברת קבצים על פורט האזנה

בעת ההרצה, הלקוח במצב משתמש מחכה לקלט מהמשתמש לחיפוש. כאשר הלקוח מזין מחרוזת לחיפוש, הלקוח פונה לשרת ומציג את תוצאות החיפוש למשתמש. הרשימה ממוינת לפי סדר א"ב עולה של שם הקובץ. הלקוח בוחר קובץ ומוריד את הקובץ מהיכן שנמצא הקובץ. כלומר, הוא פותח חיבור מתאים ושולח את שם הקובץ שהוא רוצה להוריד ומסיים בתו ירידת שורה. ואז, הלקוח שולח לו את הקובץ. כאשר הקובץ סיים לעבור הם סוגרים ביניהם את החיבור.

הקבצים שניתן להעביר יכולים להיות מכל סוג שהוא. ודאו שלאחר שהעברתם את הקובץ הוא אכן תקין וזהה לקובץ המקורי.

שימו לב, התוכנית בנויה כך שאין צורך להשתמש במקביליות בקוד. ולכן, אסור להשתמש בקוד של multi-processing או multi-threading.

לדוגמאת הרצה של התוכנית, ראו נספח א.

שאלות אודות התרגיל יש לשאול בפורום המתאים במודל.

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).

2. קבצי ה pcap של חלק א (של שני הסעיפים). ודאו שאתם מסננים החוצה תעבורה שאינה קשורה לתקשורת של התרגיל עצמו - ושהקובץ שאתם מגישים אינו מכיל תעבורה שאינה קשורה.

3. קבצי השרת והלקוח שכתבתם בחלק 2, 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). למשל:

Israel Israeli 123456789

Israela Israeli 012345678

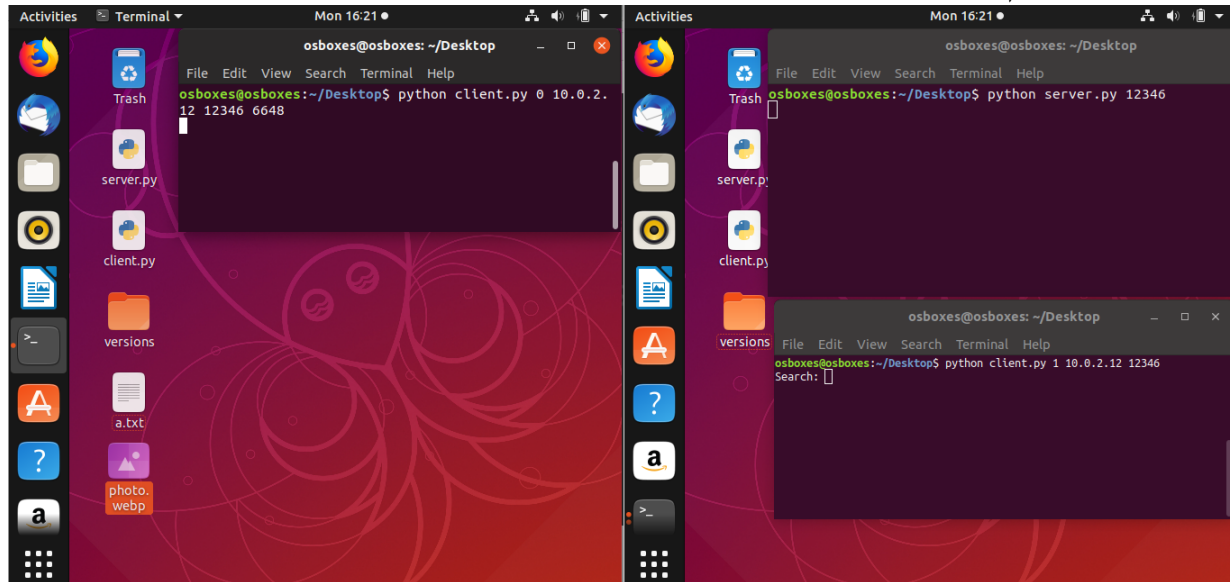
2

5. ניתן להגיש בזוגות אך לא חובה. אם מגישים בזוג, שרק אחד מבני הזוג יגיש את התרגיל (כי יש את סעיף 4 הנ"ל)

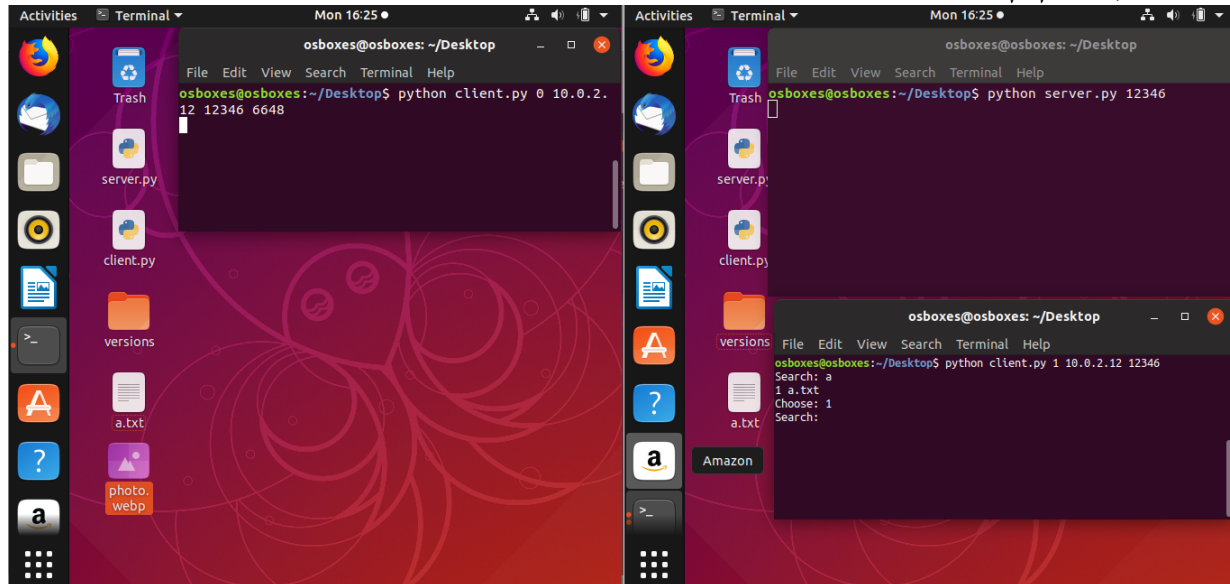
בהצלחה

א' נספח

1. הרצת שרת ושני לקוחות



2. חיפוש והעברת קובץ



3. חיפוש והעברת קובץ נוסף

