**תיעוד צד לקוח**

**קבצי Header**

pch.h

הקובץ מכיל את כל משתני ההצהרות שכל המערכת משתמשת בהם, כמו כן הוא מכיל בתוכו קריאה לכל הספריות שהמערכת משתמשת בהם, ובגלל שהוא נקרא פעם אחת מכל קובץ ולכן יכול להקרא המון פעמים, מוצהר גם ערך PCH\_H שנועד לוודא שהקריאה נעשית רק פעם אחת.

bufSize: מצהיר את גודל הפקודה המקסימלי שיכול להקרא, hostSize: מצהיר את אורך הכתובת IP שיכול להקרא וportSize מצהיר את אורך הפורט שיכול להקרא.

connections.h

מצהיר על מבנה המחלקה MyRequest שנועד לנהל את הבקשות לשרת. הפעולה MyRequest יוצרת מחלקה חדשה. הפעולה compile שולחת לשרת פקודת Python כטקסט בתוך JSON ומחזירה את התשובה מן השרת. הפעולה myHTTPSrequets יוצרת את הבקשות עצמם לשרת עם הצפה באלגוריתם ssl. הפעולה get\_my\_url מחזירה כסטרינג כתובת url מובנה. הפועלה הסטטית test בודקת יכולת חיבור לשרת עם תקשורת מוצפנת.

instream.h

מצהיר על מבנה המחלקה Inputs שנועדה לצרכי קריאת קלט מן המשתמש תוך שימוש בכל ההגבלות אבטחה על המערכת. הפעולה Inputs יוצרת מחלקה. הפעולה get\_command קוראת פקודת python מן הקלט הסטנדרטי (תוך שימוש בכל הבדיקות הדרושות) ומחזירה ערך בוליאני אם הקריאה הצליחה. הפעולה is\_more בודקת אם ישנו קלט נוסף מן הקלט הסטנדרטי. הפעולה get\_host מחזירה כתובת טקסטואלית לשרת, תוך קריאה מן הקלט הסטנדרטי (אם ישנו) או החזרת הטקסט localhost. הפעולה get\_port קוראת בהתאם מן הקלט הסטנדרטי ומחזירה בהתאם את הפורט הרצוי (כאשר 443 הינו הדיפולטיבי)

**קבצי Source**

pcf.cpp קובץ C++ שקורא לקובץ הheaser שמכיל את כל משתני וספריות המערכת הגלובלים.

instream.cpp

יוצרת מימוש בC++ למחלקה Inputs

connections.cpp

יוצר מימוש למחלקה MyRequest

ClientSide.cpp

מכילה את פקודת ה main שמריצה את המערכת:

ראשית מסבירה למשתמש שהמערכת נועדה לקימפול פקודות Python3 מן הקלט הסטנדרטי והדפסת פלט לכל פקודה בהתאם לתוצאת הפקודה לפלט הסטנדרטי (ובעצם מייצרת הדמיה לשימוש ב Python Shell))

לאחר מכן מבקשת מן המשתמש להכניס כתובת host ו port באמצעות שימוש במחקלת Inputs. (אם אין קלט, המחלקה תחזיר ערכים דיפולטיבים שהם localhost:443 בהתאם) ובודקת יכולת התחברות באמצעות פתיחת socket מסוג tcp עם הצפנה באלגוריתם SSL ובאם התאפשר וחזרו תוצאות מתאימות מן השרת ההרצה ממשיכה לשלב הבא

בשלב הבא כל עוד ישנו קלט נוסף מן הקלט הסטנדרטי, הקלט נקרא בצורה מאובטחת באמצעות המחלקה Inputs ואז נשלח לקימפול לשרת, באמצעות שליחה כפקודת json והתקשרות באלגוריתם SSL. לאחר מכן מוחזרת מן השרת התוצאה ומודפסת לפלט הסטנדרטי.

**הרצה**:

בקובץ install.txt מוגדר כיצד להתקין את כל הספריות הדרושות כך שיתממשקו עם VisualStudio.

ניתן לדאבג (debugging) באמצעות VisualStudio 16+ מן הקובץ ClientSide/ClientSide.sln

ניתן להריץ את האפליקציה ClientSide.exe במערכת הפעלה Windows אשר נמצאת בנתיב:

ClientSide/Debug/ClientSide.exe ואפליקצית זו היא פריה של הקימפול האחרון שנעשה בפרויקט ב VisualStudio