**אבטחת רשתות תקשורת**

**פרויקט סוף**

**אוטומציה ליצירת אתרי פישינג**



**מגישים:**

פבל שוורצוב, 319270583

יקיר יעקב, 305213787

**תכן עניינים**

מטרת הפרויקט - 2

מימוש - 2

בדיקת נכונות - 2

הוראות הפעלה - 3

דוגמת הרצה - 6

הגבלות - 9

**מטרת הפרויקט**

יצירת מערכת שיכולה ע"י קבלת אתר התחברות מסוים ליצור העתק לוקאלי שלו, ולשנות את פונקציונליות כפתור ההתחברות שלו לכך שישלך לנו את פרטי ההתחברות.

**מימוש**

המימוש התבצע בשפת Python.

נשתמש בספריית beautiful soup בכדי לנתח את האתר המבוקש. במהלך ניתוח האתר נוריד את כל הקבצים הרלוונטיים לתצוגה שלו, כמו תמונות, אייקונים, קבצי CSS וכו'. במקביל נשנה את מסלול הגישה לקבצים שיהיה למיקום אליו הורדנו אותם ולא למיקום המקורי שלהם.

בנוסף ניצור שרת לאתר אשר יעלה את האתר שהעתקנו, יקבל ממנו את הפרטים שמילאו בו וישלח אותם במייל אלינו.

נשתול לתוך העתק של האתר סקריפט של JS שיצרנו, אשר יופעל כאשר לוחצים על כפתור ההתחברות. הסקריפט ישלח לשרת שלנו את הפרטים שמילאו בכל השדות הקיימים באתר הפישינג שלנו. השרת ייקח את השדות יעביר אותם לפורמט טקסט וישלח מייל מאובטח למשתמש שהוגדר מראש.

**בדיקת נכונות**

בדקנו את הקוד שלנו על מערכת ההתחברות של אתר המשחקים Steam. תחילה הרצנו את הקוד ויצרנו העתק לוקאלי של עמוד ההתחברות שלהם. כעת כדי לבדוק שאכן כל הפונקציונליות הרצויה עובדת הרצנו את השרת שלנו בצורה לוקאלית.

ראינו שאכן האתר הועתק כמו שצריך וניראה זהה לאתר ההתחברות של Steam. כמו כן יצרנו משתמש חדש ב-Gmail, והכנסנו אותו לשרת כדי לבדוק שבאמת נשלח מייל עם הפרטים הנכונים. הכנסנו פרטי התחברות ולחצנו על התחברות וקיבלנו למייל את כל המידע שהכנסנו.

**הוראות הפעלה**

הקוד רשום בשפת Python ורץ בגרסה 3.7.2.

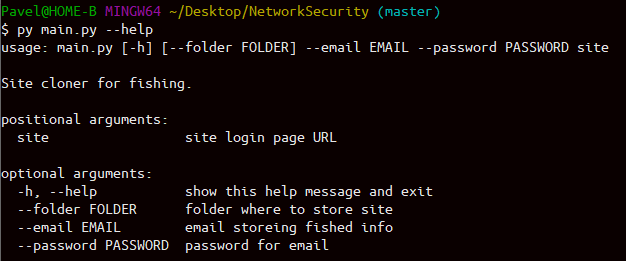
צריך להתקין את הספריות הבאות:

* beautifulsoup4
* requests
* flask
* python-dotenv

זאת ניתן לעשות ע"י הפקודה הבאה:

py -m pip install beautifulsoup4 requests flask python-dotenv

קובץ התוכנית הוא קובץ main.py שנמצא בתיקייה הראשית הפרויקט, ניתן לבדוק את אופן הפעולה עם התוכנית ע"י:

py main.py --help  
  


יש 4 משתנים שצריך לשלוח:

* Site
* Folder
* Email
* Password

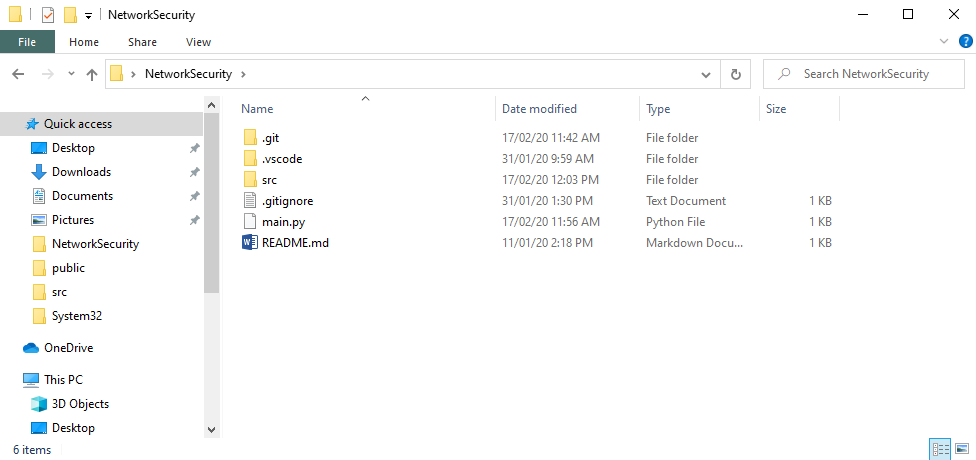
דוגמה לשורת הרצה נכונה:

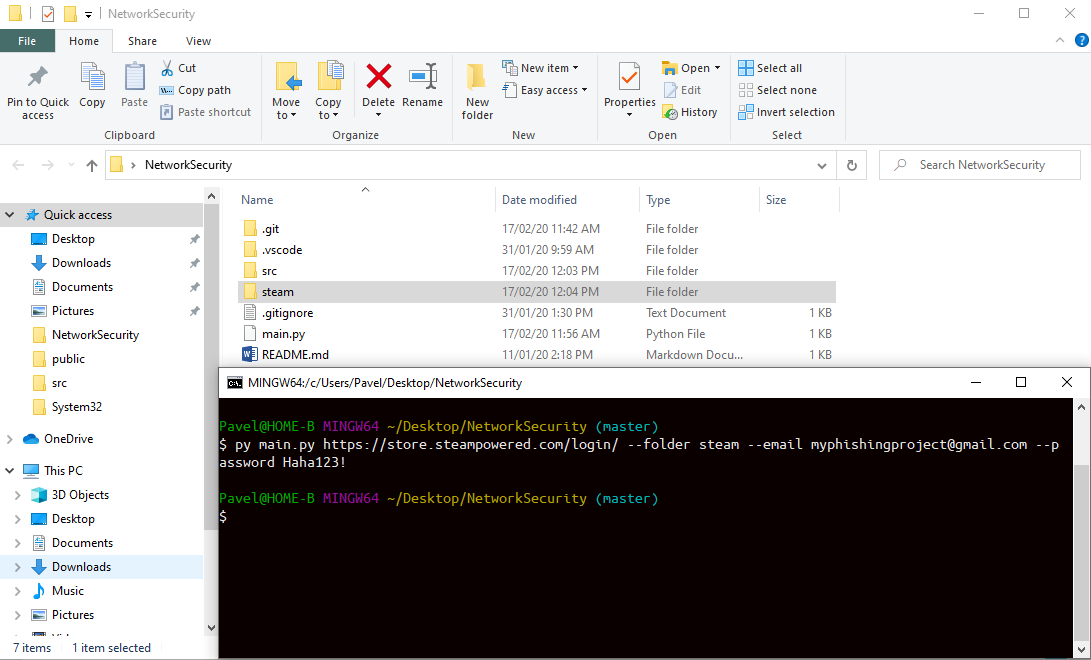
py main.py https://store.steampowered.com/login/ --folder steam --email myphishingproject@gmail.com --password Haha123!

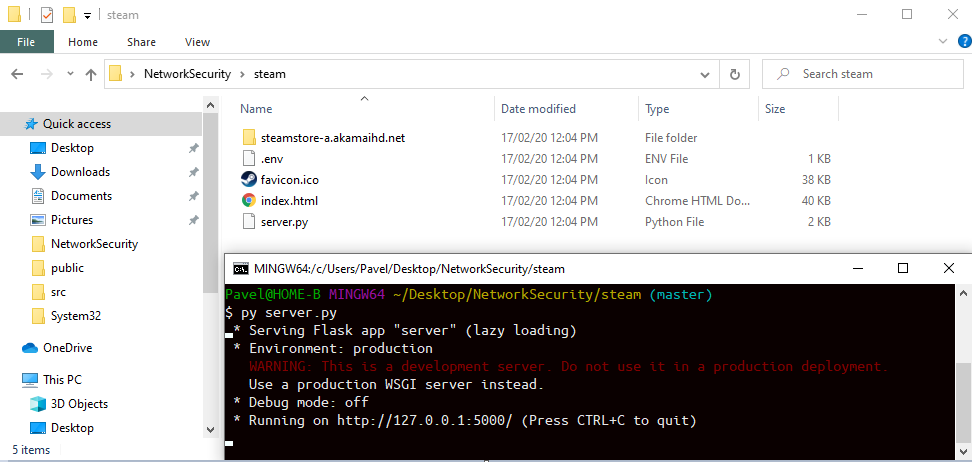
הפקודה הזאת תיצור תיקייה בשם steam ולתוכה תעתיק את כל תוכן האתר, כמו כן תיצור קובץ server.py שניתן להריץ אשר עושה שרת מקומי לאתר שהעתקנו. ניתן להריץ את server.py כדי ליצור את השרת. התוכנה תייצר קובץ .env שישמור את פרמטרי ההתחברות למייל.

יצרנו את המייל והסיסמא [myphishingproject@gmail.com](mailto:myphishingproject@gmail.com), Haha123! עבור הפרויקט, ניתן להשתמש בהם בבדיקות הרצה.

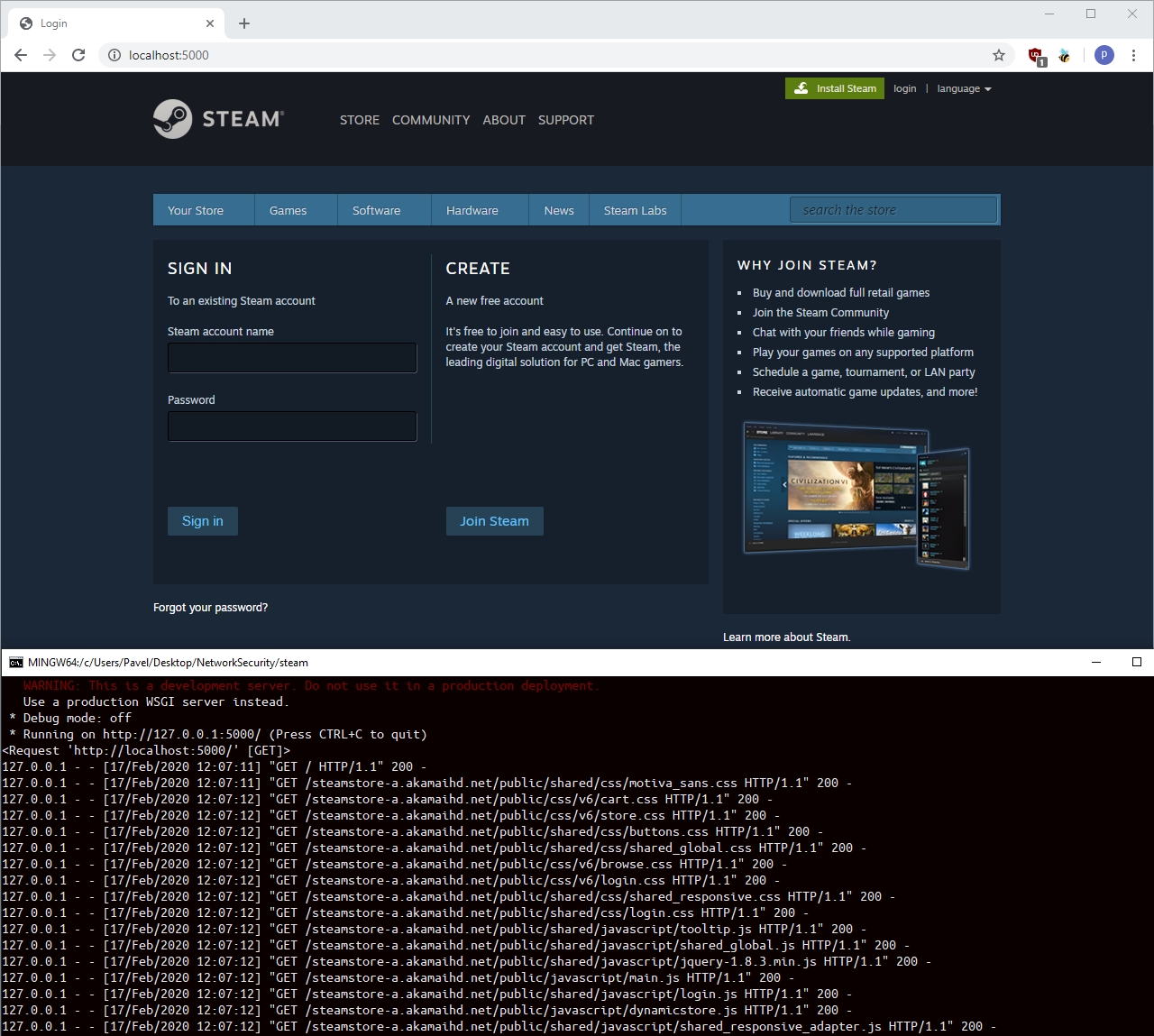
**דוגמא:**

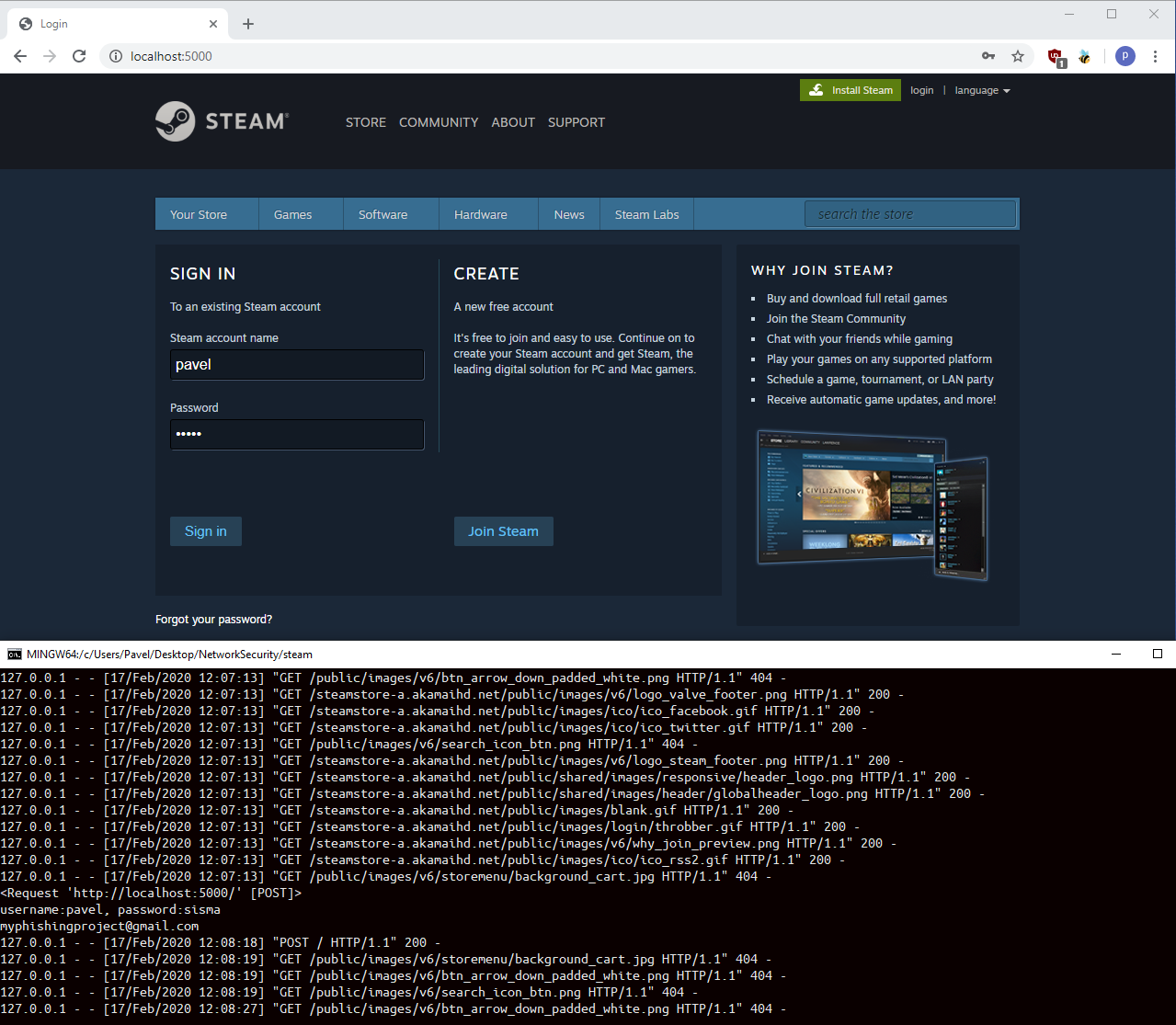
**תיקיית הפרויקט:  
**

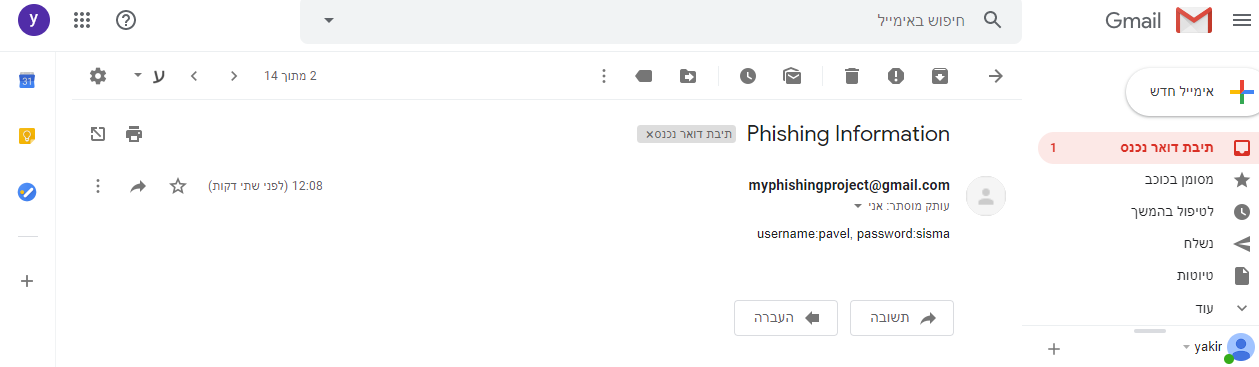
**אחרי הרצה:  
**

**הרצת השרת:  
**

**בקשת get לשרת:**

****

**ניסיון התחברות:  
**

**קבלת מייל מהשרת:  
**

**הגבלות**

ניסינו להריץ את הקוד על מספר אתרי התחברות שונים, בחלקם אכן הוא מצליח להעתיק ובחלקם לא. דוגמה בולטת היא אתר PayPal שההתחברות שם היא ב-2 שלבים, ולכן ניתן לקבל רק את שם המשתמש או את הסיסמה, לא את שניהם ביחד.

כמו כן ביצוע מספר רב של ניסיונות העתקה על אותו האתר עלול לגרום לכך שיהיה הגנת Captcha וההעתקה לא תעבוד.

עם זאת הקוד כתוב כך שאם הוא הצליח להעתיק את אתר ההתחברות, אין לו תלות בצורה שהדברים כתובים באתר. עשינו זאת ע"י כך שאנו לוקחים את כל המידע שהמשתמש מכניס בלי תלות בשם השדה, ושולחים עם המידע את שם או ID של השדה שמילאו. כמו כן אנו הופכים כל כפתור מכל סוג לכפתור רגיל, ומכניסים אליו שיריץ את פונקציית ה-JS שלנו.

בכך למעשה מנענו את הצורך ביצירת מאגר שמות ענק של שמות שדות וכפתורים, ומצב שאכן יהיה שם שרשום בצורה שאינה במאגר באתרים מסוימים ולא יעבוד אצלנו בתוכנית.