**פרויקט גמר - הנדסת תוכנה 5785**

# שם: משה בקריצקי

# ת"ז: 337609374

# תיכון: ישיבת בני עקיבא אורות יהודה

# מורה: שמעון סוויסה

# שם הפרויקט: TREETHIEF

**תוכן עניינים**

[**שם: משה בקריצקי 1**](#_osxkvjoebz6k)

[**ת"ז: 337609374 1**](#_lcuf7tudt6ot)

[**תיכון: ישיבת בני עקיבא אורות יהודה 1**](#_f6nsfgc7ke1g)

[**מורה: שמעון סוויסה 1**](#_mixx5lgsltv4)

[**שם הפרויקט: TREETHIEF 1**](#_88iaendzw9d)

[**מבוא 3**](#_yrm1nn3vb6tf)

[**הצפנת התוכנה 3**](#_8xrznb6hebs8)

[הצפנת XOR: 3](#_w9srs4xjcvwz)

[יישום בתוכנה 4](#_2b5frcfiav4j)

[שימוש בפייתון 4](#_vzqb4oaduup7)

[יישום בתוכנה שלי (קובץ ENCRYPTION.PY): 5](#_t1bbqm5zv9gs)

[תרשים זרימה 7](#_acppw5nfl5rk)

[**עד כמה התוכנה מובטחת? 7**](#_hjvdt2wrc74o)

[**היררכית העץ - מבנה הקבצים במערכת הפעלה 8**](#_d4tuz08w7sw2)

[ספריית OS: 9](#_79ymvwr7gmpb)

[יישום בקוד (CLIENT.PY, SERVER.PY): 9](#_1kprkkovt0je)

[פונקציית LISTFILES: 9](#_dswcd46s453t)

[פונקצית DOWNLOAD\_FILE: 9](#_xrl65exzyqds)

[פונקציית DELETE\_FILE: 9](#_65m5ak7bwkmg)

[פונקציית GO\_TO\_PARENT\_DIRECTORY: 10](#_i2fx2fj4kozp)

[פונקציית OPEN\_SELECTED\_DIRECTORY: 10](#_ps8zoai7nw99)

[**רשתות 10**](#_r6k86nkl65z2)

[הפרוטוקולים שלי: 10](#_cfje6z7d7go8)

[ארכיטקטורה: 10](#_86hfkxk3vhqo)

[ספריות: 11](#_16w6gll55h70)

[**THREADING 11**](#_pp6g2wnv5lhw)

[**GUI 13**](#_3dgpp1t9habv)

[TKINTER 14](#_wq9gywi50f4g)

[**הוראות הפעלת התוכנה 14**](#_a7bwu8po39q9)

[**מה הייתי משפר? 14**](#_u57lo7hoa11h)

# **מבוא**

בפרויקט הזה התמקדתי על מערכת הקבצים של המחשב. בעזרת התוכנה TREETHIEF אפשר לגשת דרך הרשת המקומית לכל מחשב עם התוכנה, ולשלוט על כל מערכת הקבצים מרחוק. התוכנה מובטחת סימטרית, ומעוצבת בעזרת TKINTER.

בעזרת התוכנה אפשר למחוק קבצים, ולהוריד קבצים מכל המחשב כולו, עד ל"דירקטורי הבסיס". אם התוכנה תגיע לידים לא מוסמכות, התוכנה יכולה ליצור נזק אמיתי ורציני מול השרת.

# 

# **הצפנת התוכנה**

### הצפנת XOR:

**מה זה הצפנת XOR? מה זה XOR בכלל?**

"אקס או אר" - "או אקסקלוסיבי" - פעולה לוגית, במקרה שלנו משתמש בביטים.

זה מחזיר אמת (1) רק אם ההשוואה בין שני ביטים (1 או 0) הם שונים. אחרת זה מחזיר שקר (0).

הפעולה של XOR מסומנת בסימן "^".

**איך זה מתקשר להצפנה?**

*Let C = A ^ B*

*Then: C ^ B = (A ^ B) ^ B = A*

*לפי הלוגיקה של XOR, אם C = A^B אזי A^B^B = A מכיוון ש B^B- שני היבטים שווים = מחזיר שקר.*

*לכן זה יוצא 0. בסוף יוצא 0 + A = A.*

*נחשוב כך: B הוא המפתח - C הוא הטקסט המוצפן בעזרת B. האות A הוא הטקסט המקורי, לפני ההצפנה. אפשר להשיג אותו בעזרת B.*

*יוצא שגם אפשר להצפין וגם להחזיר באותו משתנה B.*

*זה העיקרון שעומד מאחורי הצפנה זו.*

## יישום בתוכנה

בהצפנת מחשב אנו משתמשים בבייטים ולא בביט (פי 8). לכן, כשאנו מצפינים בעזרת XOR נשווה כל ביט בבייט אחד לשני.

**דוגמה:**

##### 

*a = 0b01101001 # 105 in decimal, ASCII for 'i'*

*b = 0b00110101 # 53 in decimal, ASCII for '5'*

*a ^ b = 0b01011100 # 92 in decimal, ASCII for '\'*

*משווים כל ביט בA בביט במיקום הדומה בB.*

*0b מקדם בינארי זה*

## שימוש בפייתון

**שימוש פשוט:**

*text = "hi"*

*key = "K"*

*# Convert to bytes*

*text\_bytes = b"hi" # [104, 105]*

*key\_bytes = b"K" # [75]*

*# XOR each byte*

*encrypted = bytes([t ^ key\_bytes[0] for t in text\_bytes])*

**התוכנה רצה בין כל המספרים הבינארים ומפעילה עליהם הצפנת XOR עם מפתח קטן**

## 

## יישום בתוכנה שלי (קובץ ENCRYPTION.PY):

*class Encryption:*

*def \_\_init\_\_(self, key):*

*self.key = key.encode() if isinstance(key, str) else key*

*def encrypt(self, data):*

*if isinstance(data, str):*

*data = data.encode()*

*key\_bytes = self.\_extend\_key(len(data))*

*return bytes(a ^ b for a, b in zip(data, key\_bytes))*

*def decrypt(self, data):*

*return self.encrypt(data) # XOR is its own inverse*

*def \_extend\_key(self, length):*

*return (self.key \* (length // len(self.key) + 1))[:length]*

**הסבר שורה בשורה:**

*self.key = key.encode() if isinstance(key, str) else key:*

*אם הטקסט הוא טקסט רגיל - התוכנה מהמרת אותה לבייטים בעזרת*

*פונקציית ENCODE*

*שורה זו בודקת אם המפתח שהוכנס הוא בבייטים, אם לא הוא מהמר לבייטים.*

*if isinstance(data, str): אם המידע שהוכנס הוא אובייקט מסוג מחרוזת*

*data = data.encode(): אם כן מהמרים אותו לבייט, אם לא הוא כבר יהיה בייט*

*key\_bytes = self.\_extend\_key(len(data)): פונה לפונקציית עזר כדי שהמפתח יהיה כאורך המידע*

*שהוכנס, כדי שהמפתח לא יחזור על עצמו ויהיה הרבה פחות עוצמתי*

*return bytes(a ^ b for a, b in zip(data, key\_bytes)): רץ על כל הביטים, משייך לקבוצות, ומקבל מהם את ערך האו אקס אר. בסוף הוא ממיר אותו חזרה מביט לבייט*

*return self.encrypt(data) # XOR is its own inverse: אם נעשה את אותם חישובים שוב - נקבל את המידע המקורי בחזרה. כמו שהדגמנו קודם - אותו ערך משומש גם להצפנה וגם חזור*

### תרשים זרימה

**

# **עד כמה התוכנה מובטחת?**

הצפנת XOR היא סימטרית - ומעניקה הבטחה בסיסית. היא חשופה לכל מיני התקפים, לדוגמה:

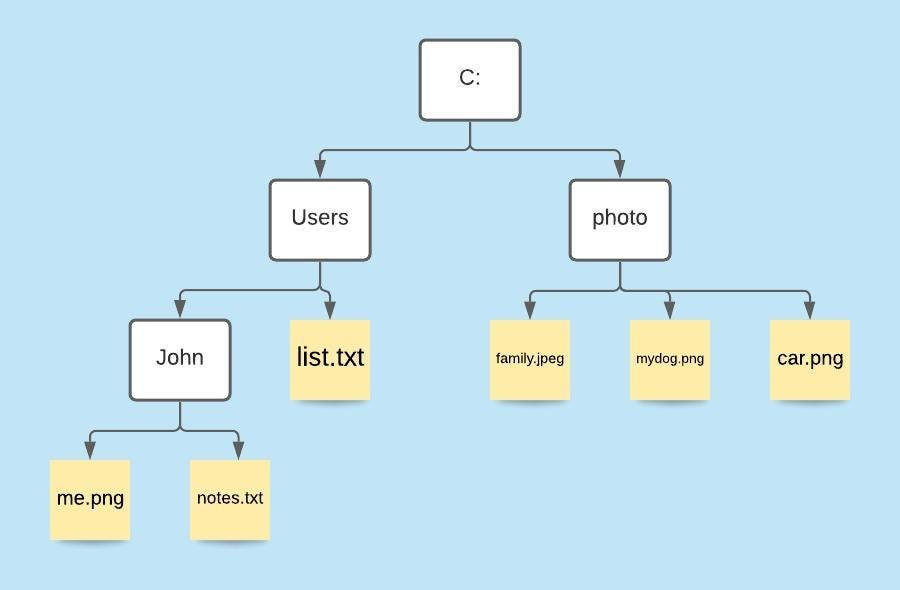
1. Known-Plaintext Attack - הפורץ יודע חלק מהטקסט ויכול מזה לגלות את הKEY
2. Chosen-Plaintext Attack - הפורץ מכשיל את השרת לשלוח לו טקסט בסיסי ומוצא את הKEY
3. DDoS (Distributed Denial-of-Service) Attacks - הפורץ יוצר המון משתמשים וקורס את התוכנה

למרות זאת היא חסינה נגד התקפים אחרים, לדוגמה:

* Passive Eavesdropping - גניבה ישירה של מידע.
* Basic Data Tampering - הפורץ לא יכול לגעת במידע ולשנות אותו באופן ישיר ואפקטיבי. הוא לא יכול להחליף את המידע של DOWNLOAD ל DELETE בלי המפתח.

# 

# **היררכית העץ - מבנה הקבצים במערכת הפעלה**

**

מבנה מערכת הקבצים במערכות הפעלה הן בנויות בצורת "עץ". יש "שורש", ולכל שורש יש ענפים (תיקיות) שבהם מוכלים קבצים ועוד תיקיות (שבהם גם מוכל תיקיות).

צורת ארכיטקטורה זו יוצרת בסיס, "שורש", ו"ענפים" רבים ההולכים וגדלים כמו ענפי עץ מתחתיו.

**WINDOWS**

במערכת הפעלה זו, כל כונן הוא "שורש" של העץ בפני עצמו. כל קובץ או תיקיה שקיים במחשב מתחיל בכונן כזה.

***LINUX | MAC OS***

במערכות הפעלה אלו יש "שורש" אחיד ובו מוכל כל הקבצים והתיקיות. גם כונן USB מוכל בתוך "שורש" זה.

**בפרויקט שלי השתמשתי במידע זו כדי לאפשר גישה לכל קבצי המחשב בשרת.**

## ספריית OS:

בתוכנה שלי השתמשתי בספריית OS כדי לגשת למפעל הקבצים של המחשב. הספרייה עוזרת בפעולות במערכת ההפעלה - כולל מערכת הקבצים (גם ב WINDOWS וגם ב LINUX).

## יישום בקוד (CLIENT.PY, SERVER.PY):

### פונקציית LISTFILES:

מאפשרת לCLIENT ליצור רשימה בDIR רצוי. רשימה זו היא רשימה של כל הקבצים ותקיות בDIR.

### פונקצית DOWNLOAD\_FILE:

מאפשר הורדה של מסמך מהשרת ללקוח.

### פונקציית DELETE\_FILE:

מאפשר מחיקה של מסמך מהשרת - ישירות בעזרת הלקוח.

### פונקציית GO\_TO\_PARENT\_DIRECTORY:

מאפשר גישה לאם של הDIR הנוכחי.

### פונקציית OPEN\_SELECTED\_DIRECTORY:

מאפשר פתיחה של הDIR הרצוי.

# **רשתות**

המושג רשת (Network) במדעי המחשב מתייחס למערכת של התקנים (NODES) המחוברים זה לזה באמצעות תקשורת, על מנת לשתף משאבים, להחליף נתונים ולתקשר. רשתות מאפשרות למחשבים לעבוד יחד, לגשת למידע משותף, להפעיל יישומים מרוחקים ולספק שירותים כמו גלישה באינטרנט, שליחת דוא"ל, שיתוף קבצים ועוד.

**לרשתות יש חוקים - פרוטוקולים.**

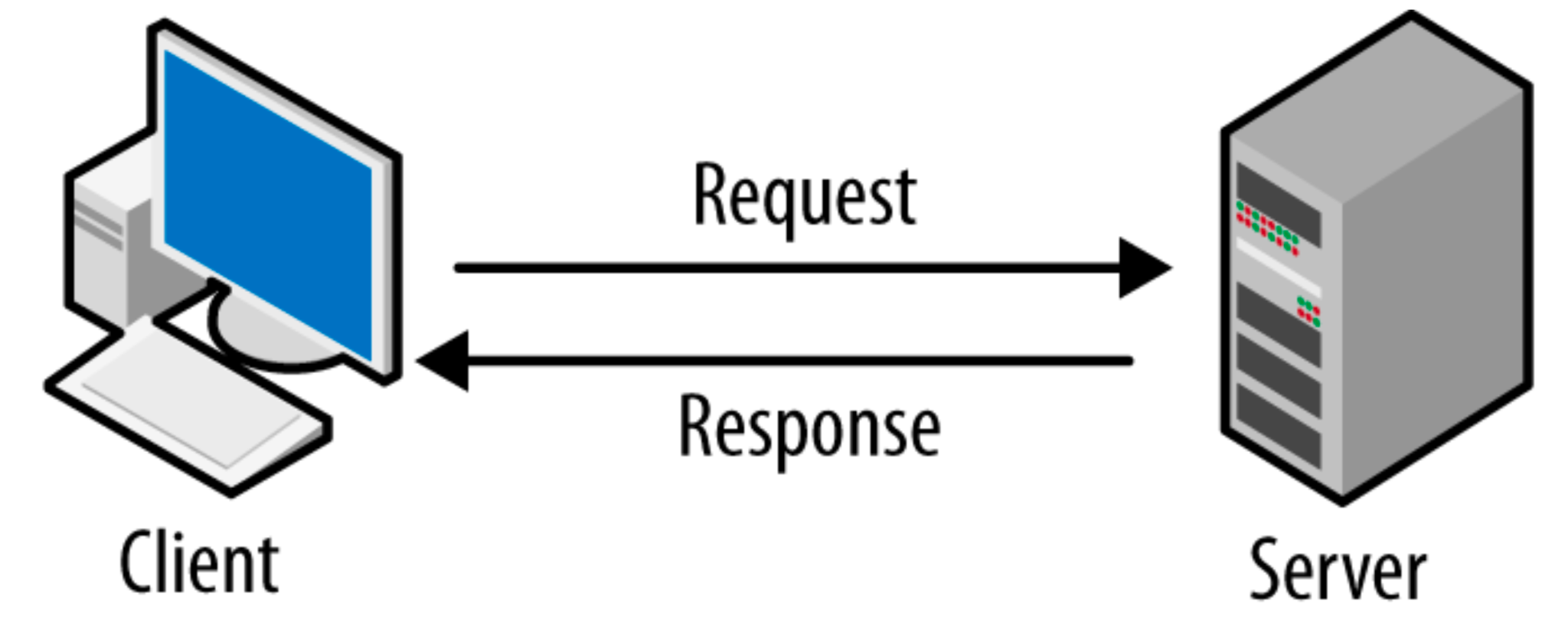
**יש כל מיני צורות רשת - כגון כוכב.**

## הפרוטוקולים שלי:

בפרוקייקט שלי השתמשתי בפרוטוקולים TCP וIP. הראשון מאפשר חיבור אמין עם לחיצת יד משולשת. השני מאפשר גישה לכתובת של מחשב זר.

## ארכיטקטורה:

בפרויקט שלי השתמשתי בטופולוגית כוכב. המבנה שלי הוא שרת - לקוח. שרת ראשי המנהל את העניינים, ולקוחות שמחוברים אליו.



## 

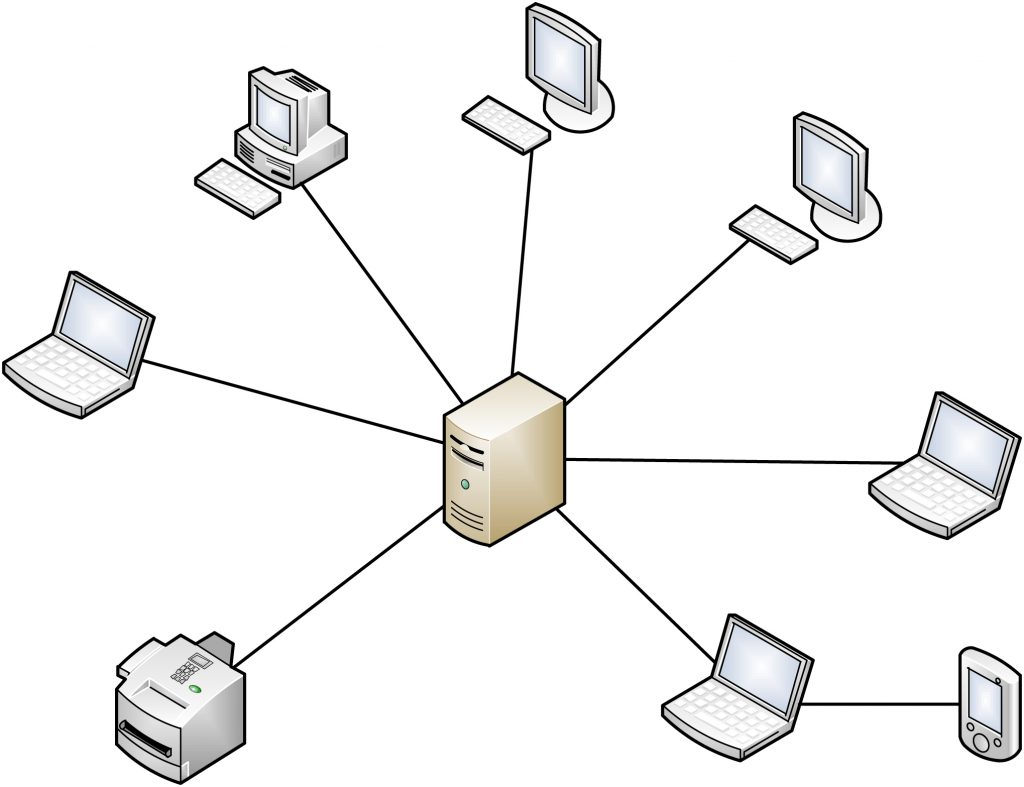
## ספריות:

כדי להשתלט על הרשת, השתמשתי בספריית SOCKET, ספריית JSON, ספריית STRUCT.

SOCKET - מאשר פתיחת וניהול צינור מידע

JSON, STRUCT - עוזר בתיאום קבצים יותר מסובכים לבייטים של הרשת

# **THREADING**



בפרויקט שלי השתמשתי בספריית THREADING כדי ליצור חיבורים בו זמניים בשרת. בעזרת THREADING - כל לקוח מקבל THREAD חדש משלו שעליו הוא רץ. ולכן השרת יכול לפעול בו זמנית בין לקוחות שונים.

conn, addr = self.socket.accept()

# Create a new thread for each client

client\_thread = threading.Thread(target=self.handle\_client, args=(conn, addr))

client\_thread.start()

השרת מקשיב למשתמשים - ואם התקבל משתמש חדש - הוא שם אותו בTHREAD של עצמו (כמו חדר שבו הוא מקבל יחס אישי).

# **GUI**

# 

## TKINTER

בפרויקט השתמשתי בספריית TKINTER כדי ליצור ממשק נוח לשימוש על ידי הלקוח. בעזרת הממשק אפשר להתחבר, למחוק, להוריד קבצים ועוד.

# 

# **הוראות הפעלת התוכנה**

1. הורדת פייטון
2. להכניס לקובץ CONFIG את הסיסמה שלך (חייב להיות דומה בכל מכשירי הקשר), ולשנות את כתובת הIP לכתובת הרצוי של השרת.
3. במקרה הצורך להסיר FIREWALL.
4. להריץ את השרת ואז את המשתמש
5. יפה! אתה יכול לשלוט במערכת הקבצים של השרת!

# 

# **מה הייתי משפר?**

* ממשק להכניס את הIP
* שינוי ססימה כל שימוש - שימוש בהצפנה חזקה יותר
* התחברות לרשתות מחוץ לLAN.

# **רפלקציה**

במהלך הפרויקט למדתי המון על הצפנה, תקשורת בין מחשבים, ומערכת הקבצים במערכות הפעלה. למדתי להתמודד עם קשיים - והיו הרבה.

אני ממש שמח מאיך שהפרויקט יצא. בעתיד אולי אכין עוד, ואכנס יותר לעולם

של הצפנה.

למדתי שעבודה קשה צורכת זמן רב - אבל בססוף זה שווה.

# **תודות**

אני רוצה להודות לשמעון סוויסה. הוא ממש עזר לי בהכנת הפרויקט - ועזר לי להתמודד עם הקשיים כאשר הם חסמו אותי.