פרק א – ייזום

# א. תקציר:

הרבה פעמים כאשר גולשים באינטרנט לא מרגישים מספיק בטוחים, הרבה פעמים יש את ההרגשה של החוסר פרטיות כאשר גולשים באינטרנט.

הפרויקט שלי פותר את הבעיה הזאת, הפרויקט שלי יאפשר לאנשים לגלוש באינטרנט באנונימיות מוחלטת, בלי שיידעו מי הם או מאיפה הם גולשים ובכך להעניק תחושת פרטיות ומוגנות כאשר הוא גולש באינטרנט לכל מי שמשתמש במוצר.

# ב. למי זה מיועד?

הפרויקט שלי פונה לכל בן אדם שמעוניין להרגיש יותר פרטיות וביטחון כאשר הוא גולש באינטרנט. לדוגמה אנשים שרוצים להגיב תגובות אנונימיות או אנשים שרוצים לעשות פעולות ברשת בלי להשאיר עקבות.

# ג. הגדרת יעדים/מטרות:

המטרות המרכזיות של המוצר הן להעניק ביטחון ופרטיות, שלא יהיה אפשר לגלות מי הבן אדם שגולש והיכן הוא נמצא.

# ד. בעיות, תועלות וחסכונות:

בעיות: כשלקוח גולש ברשת הIP שלו גלוי, כלומר אפשר לזהות מי הבן אדם שגולש והיכן הוא נמצא. אני מנסה להשיג שכאשר בן אדם יגלוש ברשת אף אחד לא יידע את הIP שלו.

תועלות שסביר לצפות מהמערכת: הפחתה בכמות הפריצות , כאשר לפורץ אין את הIP שלך יותר קשה לו לגלות מידע עלייך ובכך יותר קשה לו לבצע פריצה. בנוסף יאפשר ללקוחות לגלוש באנונימיות ובפרטיות או אפילו ממקום אחר בעולם (תלוי איפה מוצבות תחנות הביניים).

אילו שירותים המערכת מציעה? גלישה אנונימית ברשת, והצפנה בצורת הבצל וגלישה מאזורים שונים.

כעת בשוק יש שתי פתרונות דומים: )VPN כמו NordVPNאו (TunnelBearו TOR , שניהם מחביאים את הIP של המשתמש אך VPN מעביר את ההודעה דרך שרת אחד שמשנה את הIP , בעוד שבTOR יש מספר שרתים שבכל אחד מתבצע הצפנה של ההודעה עם מפתח של אחר ולבסוף גם שינוי הIP , כך שהTOR מבטיח בצורה יותר טובה אנונימיות ושאף אחד לא יוכל לגנוב את המידע של ההודעה.

<https://www.torproject.org> - Tor

<https://nordvpn.com/> - NordVPN

https://www.tunnelbear.com/ - TunnelBear

# ה. קשיים או מגבלות בהגדרת המערכת

האם מדובר בטכנולוגיה חדשה ובלתי מוכרת? לא, הטכנולוגיה הזאת מוכרת מאוד ונמצאת בשימוש רב כבר די הרבה זמן.

האם קיימים סייגים בהגדרת המערכת? המערכת תלויה במהירות של המחשבים של המעבדה איתם אני משתמש כתחנות מעבר , בנוסף המערכת מטפלת רק בהודעות GET

# ו. תיחום הפרויקט

הפרויקט עוסק ברשתות ובתחום האבטחה.

פרק ב' – אפיון

# פירוט המערכת:

כאשר גולשים ברשת נשלחת בקשה ברשת הכוללת את ה IP של המשתמש, הip עלול להימצא על ידי אנשים המאזינים לרשת ועם המידע הזה אפשר לדעת כמעט הכל על המשתמש בעזרת עוד קצת שימוש בip שלו , דברים כמו מהיכן הוא נכנס או באיזה עוד אתרים הוא נוטה לגלוש בתדירות גבוהה, מידע זה עלול לתת לאנשים מסוכנים יותר סיכוי לפרוץ לדברים הקשורים למשתמש. המערכת שאני בונה פותרת את הבעיה הזאת ומסתירה את הip , ההודעה תוצפן במחשב בעזרת שלוש מפתחות של שלוש תחנות ביניים שהמחשב יבחר. בכל אחת מהתחנות התחנה תוריד שכבה אחת של הצפנה בעזרת המפתח שלה ובכך תדע למי להעביר אותה הלאה. כך שבסופו של דבר ההודעה תגיע מהתחנה האחרונה כאשר הip שעובר ברשת הוא של הip של התחנה האחרונה. חשוב לציין כי אף אחת מהתחנות לא תוכל לדעת את תוכן ההודעה וגם כאשר ההודעה תעבור בין התחנות היא תהיה מוצפנת.

# יכולות המערכת:

משתמש קצה:

1. גלישה אנונימית ברשת – המשתמש יוכל לגלוש חופשי ברשת בלי שידעו את הip האמיתי שלו , כלומר לא יוכלו לזהות אותו.

מנהל מערכת:

1. ממשק גרפי **– מנהל המערכת יוכל לבחור לראות מי התחנות שנבחרו בעזרת ממשק גרפי מובנה.**
2. ניתוק תחנות לאלתר – **מנהל המערכת יוכל לבחור להוציא תחנות ביניים מהאפשרויות, עד שיחליט אחרת**
3. הוספת תחנות ביניים – **המנהל יוכל לבחור להוסיף תחנות ביניים.**

# בדיקות המערכת – 'קופסא שחורה':

| מספר | שם הבדיקה | מה אמורה לבדוק | איך מתכננים לבדוק |
| --- | --- | --- | --- |
| 1. | העברת החבילה לשרת | שהחבילה הגיעה לשרת לאחר ניסיון התחברות לאתר. | מתחברים לאתר ובודקים עם הסנפה שההודעה באמת מגיעה לשרת |
| 2. | זמני תגובה | בודקת שהזמן משליחת ההודעה ועד לקבלת התשובה לא גדול מדי. | שולחים הודעה , ומודדים את הזמן מתחילת ההודעה ועד לקבלת תשובה. |
| 3. | בדיקת IP | בודקת שהIP שנשלח לשרת הוא לא של הלקוח. | מסתכלים על הפלט של כל אחד מתחנות הביניים , כל אחת מהתחנות תדפיס את הip של החבילה שהיא מקבלת. |
| 4. | הצפנה | בדיקה שההצפנה עובדת | מנסים להסניף את החבילה , ובודקים האם ניתן לקרוא אותה. |
| 5. | התמודדות עם תחנה שקורסת | בודקת שהמערכת ממשיכה לעבוד למרות סגירת תחנה | פותחים לקוח, בודקים למי נשלח ההודעה בפעם הראשונה , וסוגרים את התחנה . אחר כך בודקים שההודעה נשלחת שוב לתחנה אחרת. |
| 6. | טיפול במספר לקוחות | בודקת שהשרת יכול לתפעל עם מספר לקוחות | מנסים להיכנס עם שני לקוחות בו זמנית ובודקים שהכל פועל כמתוכנן. |
| 7 | עומסים | בודקת האם המערכת מסוגלת לעמוד בעומס בקשות | נכנסים לשרת מעמדות שונות בו זמנית (יותר משתיים) ובודקים זמני תגובה בכל תחנה |
| 8. | תשובה מלאה | בדיקה האם כל התשובה חוזרת במלואה ללקוח | נכנסים עם הלקוח לאתר ובודקים שהאתר נטען במלואו |

# תכנון זמנים:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| פעילות | זמן התחלה מתוכנן | זמן סיום מתוכנן | זמן התחלה בפועל | זמן סיום בפועל | הערות |
| יזום | 15.10.21 | 22.10.21 | 15.10.21 | 18.10.21 |  |
| אפיון | 25.10.21 | 1.11.21 | 25.10.21 | 1.11.21 |  |
| ניתוח | 3.11.21 | 10.11.21 | 3.11.21 | 16.11.21 |  |
| עיצוב | 14.11.21 | 28.11.21 | 15.11.21 | 30.11.21 |  |
| שינוי IP | 28.11.21 | 15.12.21 | 2.12.21 | 14.12.21 |  |
| הצפנה בין התחנות | 16.12.21 | 25.12.21 | 14.12.21 | 14.12.21 |  |
| זמני תגובה | 25.12.21 | 31.12.21 | 16.12.21 | 20.12.21 |  |
| ממשק מנהל | 3.1.22 | 14.1.22 | 25.12.21 | 10.1.22 |  |
| גרסה ראשונית | 15.1.22 | 1.2.22 | 10.1.22 | 2.2.22 |  |
| מסמך בדיקות | 2.2.22 | 4.2.22 | 3.2.22 | 3.2.22 |  |
| מדריך למשתמש | 4.2.22 | 6.2.22 | 4.2.22 | 5.2.22 |  |
| גרסה סופית | 10.2.22 | 15.3.22 | 6.2.22 | 28.2.22 |  |
| סגירת תיק פרויקט | 18.3.22 | 24.3.22 | 2.3.22 |  |  |
| מצגת הסבר | 26.3.22 | 27.3.22 |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **הסיכון** | **פירוט הסיכון** | **רמת הסיכון**  **(קל/בינוני/קשה)** | **תיאור דרכים (לפחות 2 ) להתמודדות עם הסיכון ולהקטין אותו** | **מה בוצע בפועל** | **תאריך** |
| אי עמידה בזמנים | פרויקט לא יושלם | **קשה** | הקדמת לו"זים משימות.  ארגון הזמן בצורה יעילה.  להתחיל בדברים היותר קשים. | הקדמתי לוזים  ארגנתי את הזמן בצורה יעילה והספקתי להגיע לפרויקט מושלם בזמן | 2.3.22 |
| חוסר הצלחה בשינוי הIP | כאשר ההודעה תגיע לבסוף לשרת הוא יקבל את הIP של הלקוח | **קשה** | לעשות בדיקות לראות עם ההודעה באמת השתנתה לפני שיצאה מהשרת  לבדוק את התקשורת בין התחנות ולוודא שאין שינוי בהודעה. | בדקתי את התקשורת ווידאתי שאין שינויים בהודעה. | 27.2.22 |
| חוסר הצלחה בהצפנה | התקשורת עם תחנות הביניים לא תעבוד עם ההצפנה(התחנות לא ידעו לפענח את ההצפנה) | בינוני | לבדוק את התקשורת ביניהם ולבדוק שההצפנה היא נכונה ולפי הסדר.  לקרוא ולחקור עוד על הבעיה.  לבדוק שהמפתחות אכן תואמים בין כל תחנה לשרת. | בדקתי את ההצפנה בתקשורת והיא אכן נכונה  כל מפתח תאם בין כל תחנה לשרת | 27.2.22 |
| זמני תגובה | הטיפול בבקשה יארך זמן רב והמשתמשים יינטשו את המערכת | קל | -לתכנן חכם את המערכת כך שהתקשרות לא תהיה איטית.  - לחזור לקוד למצוא מקומות בהם אפשר לשפר את היעילות והמהירות של המערכת. | תכננתי חכם את המערכת, עברתי על קטעי קוד שעלולים לקחת זמן והפכתי אותם ליעילים יותר | 27.2.22 |

פרק ג – ניתוח

# פירוט יכולות המערכת:

**צד לקוח:**

בצד לקוח אין יכולות שאני מפתח – הגלישה היא באמצעות הדפדפן הרגיל שהשרת במערכת מוגדר כשרת ה proxy עבורו.

המידע שחוזר מהשרת אף הוא מוצג בדפדפן כמידע שהתקבל מהאינטרנט ללא טיפול שלי.

**צד שרת:**

* **יצירת מפתחות** – השרת ייצר לעצמו מפתח ציבורי ומפתח פרטי . רשימת אובייקטים: הצפנה א-סימטרית , מפתח ציבורי ומפתח פרטי.
* **פרסום מפתח ציבורי** – כל תחנה שהשרת יצרף תקבל את המפתח הציבורי של ההצפנה

רשימת אובייקטים: מפתח ציבורי , תקשורת.

* **החלפת מפתחות** – השרת יחליף מפתחות עם כל אחת מתחנות הביניים שהתחברו.

רשימת אובייקטים: הצפנה א-סימטרית((RSA , פענוח אסימטרי, תקשורת

* **בחירת תחנות הביניים** – השרת יבחר את 3 תחנות הביניים שאליהן ירצה להתחבר בצורה רנדומלית מתוך כלל תחנות הביניים שמקושרות אליו.

רשימת אובייקטים: בסיס נתונים(SQL), אלגוריתם בחירה.

* **עדכון תחנות הביניים שנבחרו** – השרת יעדכן את התחנות שנבחרו במספר port דרכו תעבור אליהן ומהן ההודעה
* רשימת אובייקטים: אלגוריתם בחירת port, תקשורת.
* **ניתוק תחנת ביניים** – אם תחנה לא מגיבה השרת ינתק אותה ויחליף אותה באחרת.

רשימת אובייקטים: תקשורת

* **הסנפת הודעה** – השרת יסניף את ההודעה מהלקוח.

רשימת אובייקטים: הודעה, סינון לפרוטוקול HTTP.

* **פירוק הודעה** – השרת יוציא מההודעה שהתקבלה את ה IP של מחשב הלקוח ויכניס במקומו את ה- IP של תחנת הביניים האחרונה בשרשרת.

רשימת אובייקטים: הודעה, IP לקוח, IP תחנת ביניים

* **בניית והצפנת ההודעה** – השרת יבנה הודעה ויצפין את ההודעה בעזרת המפתחות של השרתים ע"פ פרוטוקול הבצל.

רשימת אובייקטים: הצפנה סימטרית(AES), מפתחות תחנות הביניים, הודעה

* **שליחת הודעה לתחנת ביניים –** השרת יתחבר לתחנת הביניים הראשונה במסלול ויעביר אליה את ההודעה

רשימת אובייקטים: הודעה, תחנת ביניים ראשונה, תקשורת

* **פענוח ההודעה** – השרת יידע לפענח את ההודעה שחוזרת מתחנות הביניים.

רשימת אובייקטים: פענוח של הצפנה סימטרית., מפתחות תחנות הביניים

* **החזרת ההודעה** – השרת ישלח חזרה את התשובה למשתמש המקורי.

רשימת אובייקטים: תקשורת, בסיס נתונים.

* **בדיקת התחברות** – כאשר מחשב מנסה להתחבר לשרת הוא יבדוק את כתובת הMAC של המחשב מול שתי רשימות שברשותו, whitelist ו blacklist, השרת יאפשר את החיבור עם המחשב רק אם כתובת הMAC של המחשב יהיה בwhitelist בblacklist

רשימת אובייקטים: בסיס נתונים, תקשורת, כתובת MAC

* **קבלה של נתונים למנהל או שליחה של נתונים למנהל** – במידה והמנהל משתמש בממשק חייבת להיות תקשורת לטובת קבלת פקודות ושליחת מידע.

רשימת אובייקטים: תקשורת, ממשק מנהל, בסיס נתונים

**ממשק מנהל:**

* **אישור תחנת ביניים** – המנהל יוכל להכניס תחנת ביניים לwhitelist , השרת יתייחס למשתמש המתחבר אליו כתחנת ביניים רק אם הכתובת MAC שלו נמצאת בwhitelist.

רשימת אובייקטים: מסד נתונים , כתובת MAC, ממשק גרפי.

* **חסימת תחנת ביניים** – במידה והמנהל זוהה שמדובר במחשב המתחזה לתחנת ביניים הממשק יאפשר למנהל להכניס את המחשב המתחזה לblacklist וגם להוציא אותו מהwhitelist במידה ונמצא שם כלומר השרת יסגור את החיבור ויחסום את החיבור עם המחשב כאשר הוא ינסה להתחבר בפעם הבאה.

רשימת אובייקטים: מסד נתונים, ממשק גרפי.

* **התחברות לממשק מנהל** – המנהל יצטרך לשלוח לשרת הראשי שם משתמש וסיסמא על מנת להיכנס לממשק מנהל.

רשימת אובייקטים: תקשורת , שרת ראשי, בסיס נתונים, הצפנה

* **קביעת מספר תחנות ביניים** - המנהל יוכל לבחור את מספר התחנות שהודעה תצטרך לעבור עד שתגיע לשרת.

רשימת אובייקטים: ממשק גרפי, מסד נתונים, תקשורת

* **קבלה של נתונים מהשרת או שליחה של נתונים לשרת** – בממשק המנהל יוצגו למנהל הנתונים השמורים במערכת: רשימות whitelist ו- blacklist עם כתובות Mac וכן מספר המציין את מספר ההעברות שהודעה צריכה לעבור עד שיוצאת לאינטרנט. המנהל יוכל לבצע שינויים במערכת דרך הממשק

רשימת אובייקטים: תקשורת, שרת ראשי, ממשק גרפי, הצפנה

**צד תחנת ביניים:**

* **יצירת מפתחות** – התחנה תייצר לעצמה מפתח ציבורי ומפתח פרטי

רשימת אובייקטים: הצפנה א-סימטרית , מפתח ציבורי ומפתח פרטי

* **התחברות לשרת המרכזי** – התחנה תתחבר לשרת המרכזי.

רשימת אובייקטים: תקשורת

* **התחנה תפתח אצלה שרת -** שיחכה לחיבורים מתחנות ביניים אחרות בהתאם לport שהתקבל מהשרת

רשימת אובייקטים: תקשורת , שרת.

* **החלפת מפתחות** – תחנת הביניים תחליף מפתחות עם השרת כאשר השרת יפנה אליה.

רשימת אובייקטים: הצפנה ופענוח אסימטרי, מפתחות, תקשורת

* **פענוח הודעה** – התחנה תקבל הודעה מתחנת ביניים אחרת או מהשרת ותפענח שכבה אחת של הצפנה בעזרת המפתח הפרטי שלה.

רשימת אובייקטים: פיענוח של הצפנה סימטרית (AES) , מפתח פרטי

* **התחברות לתחנת ביניים** – התחנה תתחבר לתחנת ביניים בהתאם למידע בהודעה שהתקבלה.

רשימת אובייקטים: תקשורת.

* **העברת ההודעה** – התחנה תעביר את ההודעה לתחנת הביניים.

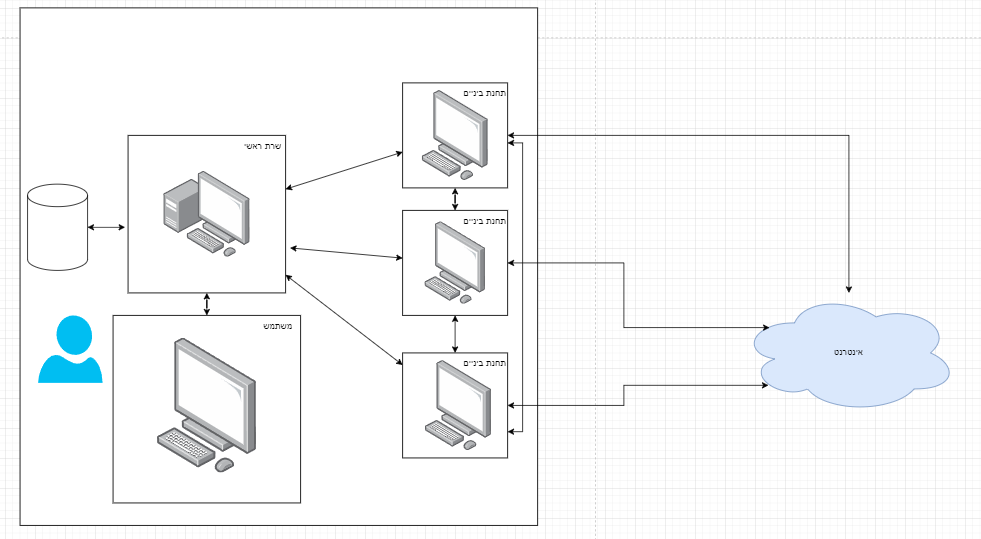
רשימת אובייקטים: תקשורת.

* **דיווח שגיאות** – התחנה תדע לדווח לשרת על אי יכולת להתחבר לתחנת ביניים נדרשת

רשימת אובייקטים: תקשורת, שרת ראשי, שגיאה

פרק ד' – עיצוב

# תיאור ארכיטקטורה של המערכת המוצעת:



בשרטוט מוצג שרת בעל קשר למשתמש, בנוסף יש עוד 3 תחנות ביניים הקשורות כל אחת מהן אחת לשנייה, לשרת ולאתר.

שרת , תחנות ביניים , מחשב לקוח – מחשבים מבוססים מ"ה windows.

בסיס נתונים – בסיס נתונים טבלאי SQL.

רשת מקומית – השרת תחנות הביניים והלקוח משותפות בה.

רשת אינטרנט .

דפדפן – נמצא בעמדת המשתמש.

# תיאור הטכנולוגיה הרלוונטית:

**שפת תכנות –** Python

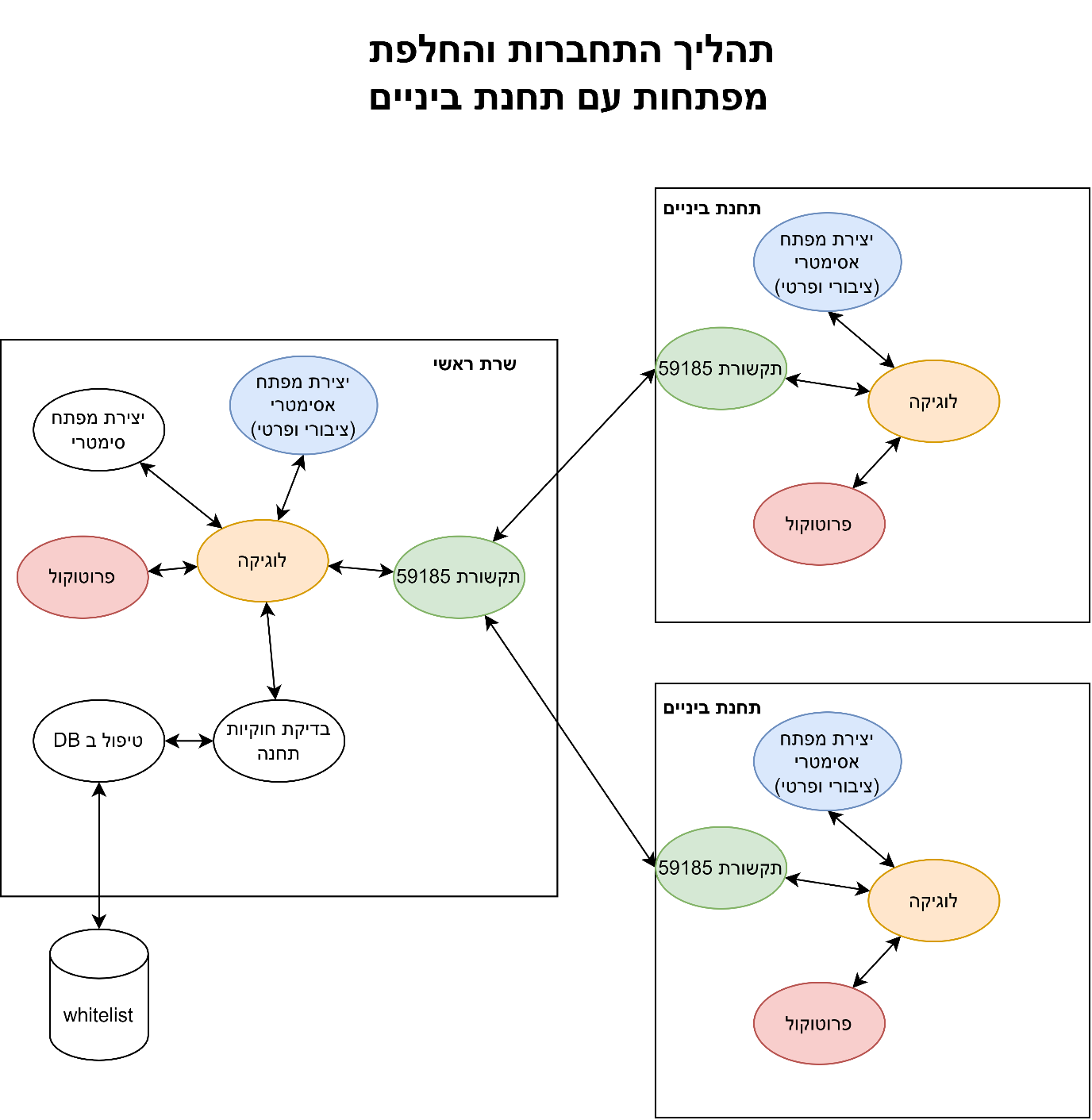
**מערכת הפעלה –** Windows

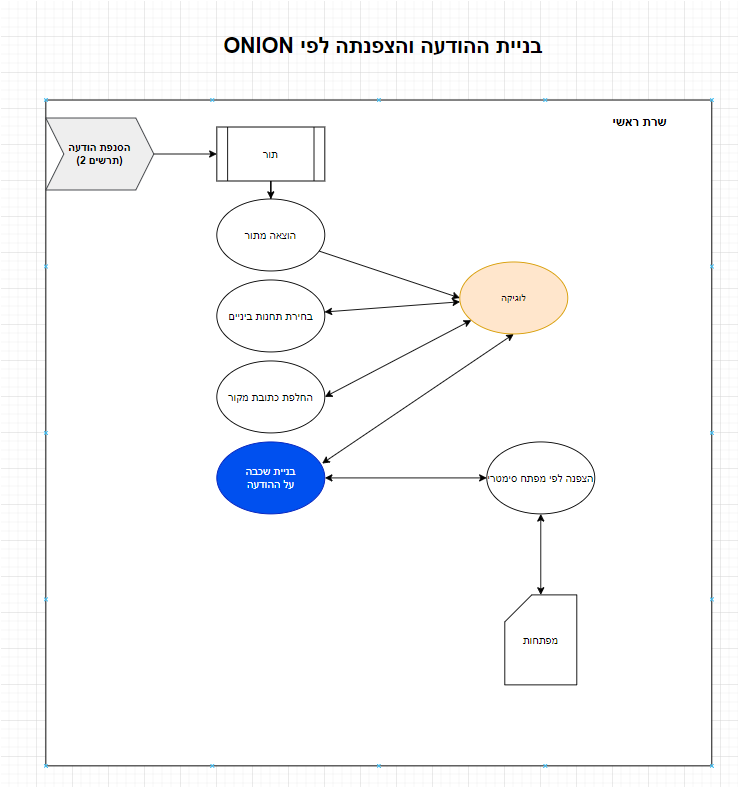
דפדפן – chrome עם יכולת להגדרת proxy.

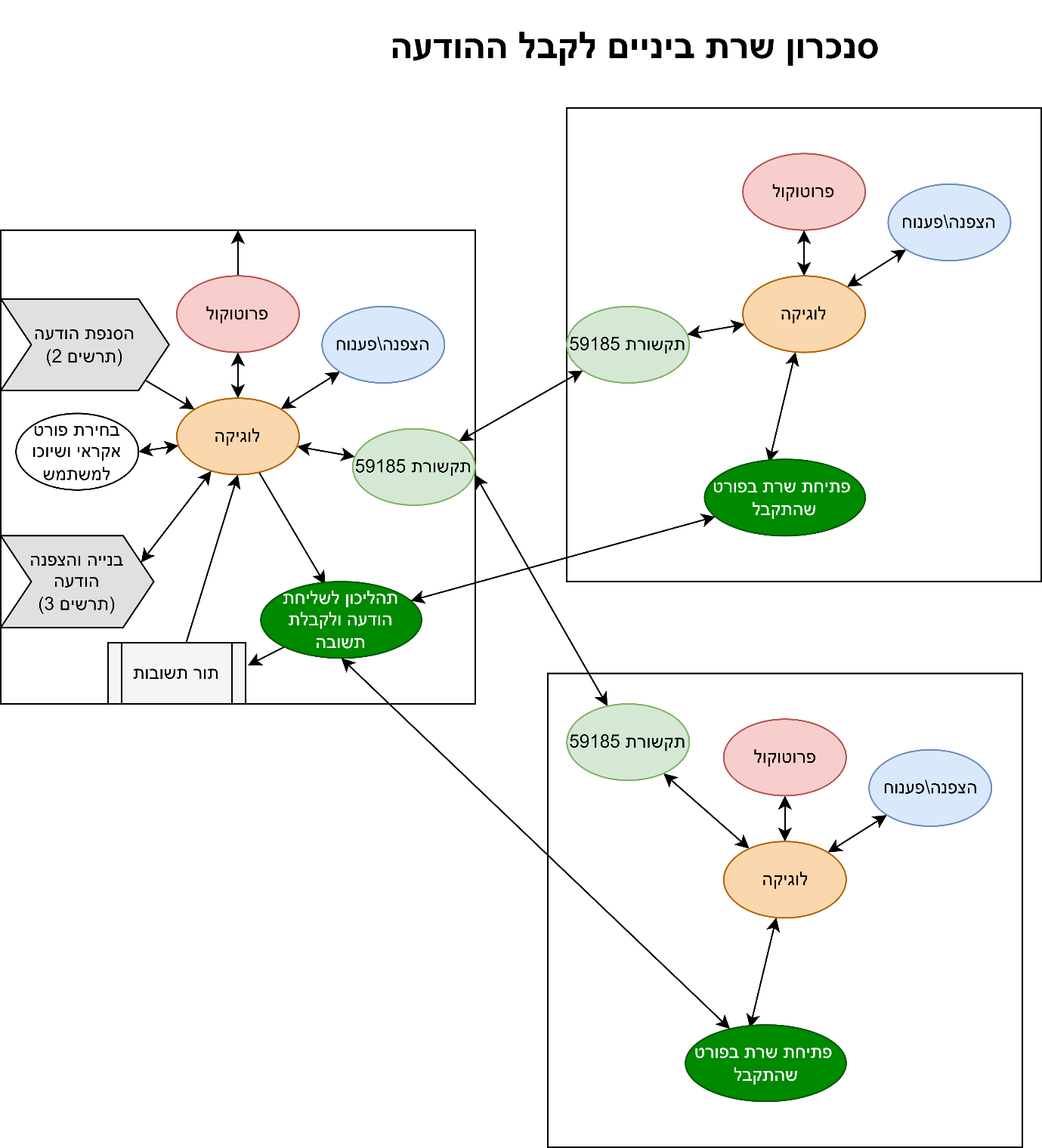
תקשורת מבוססת סוקטים.

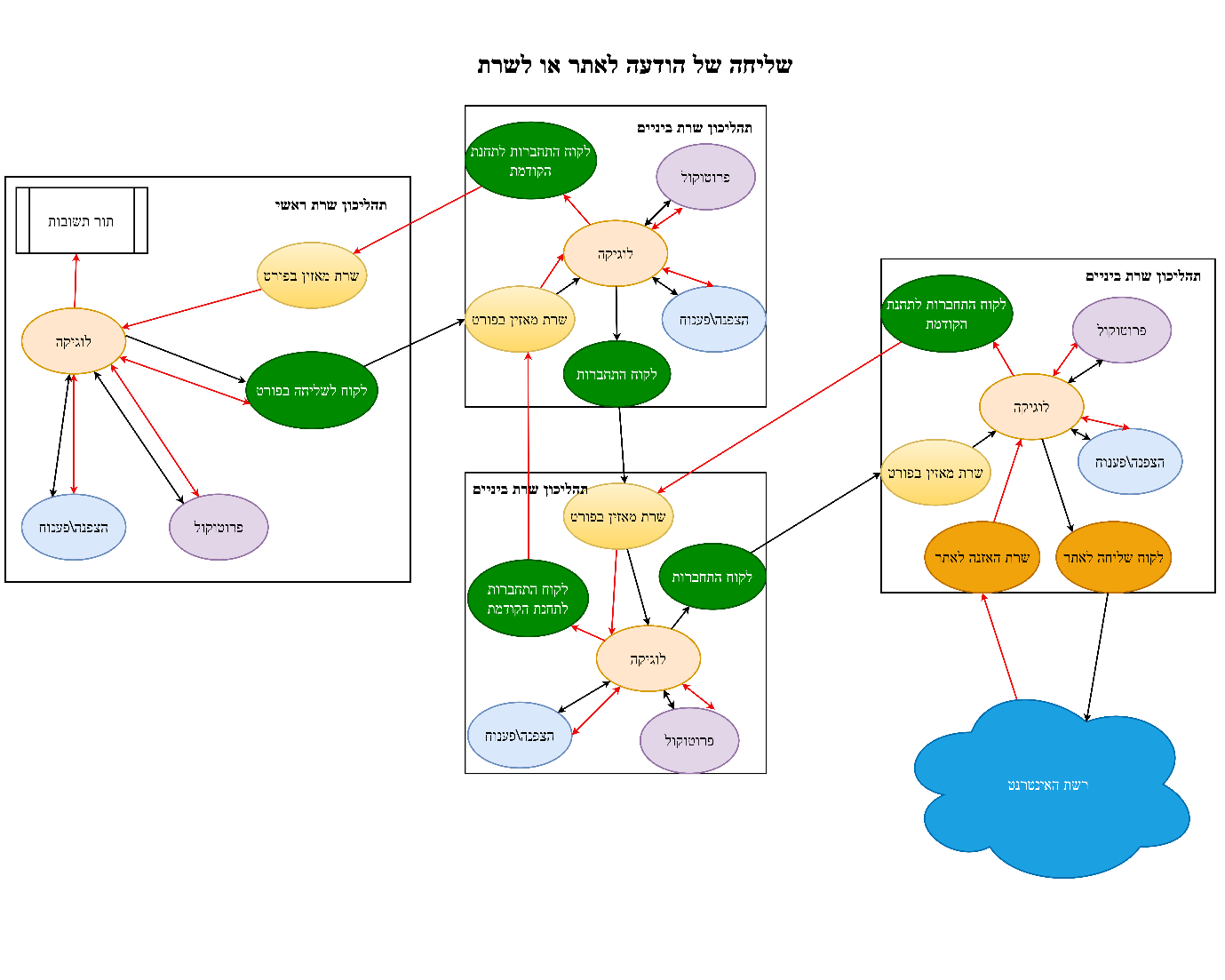
# תיאור מודלים בהם נעשה שימוש:

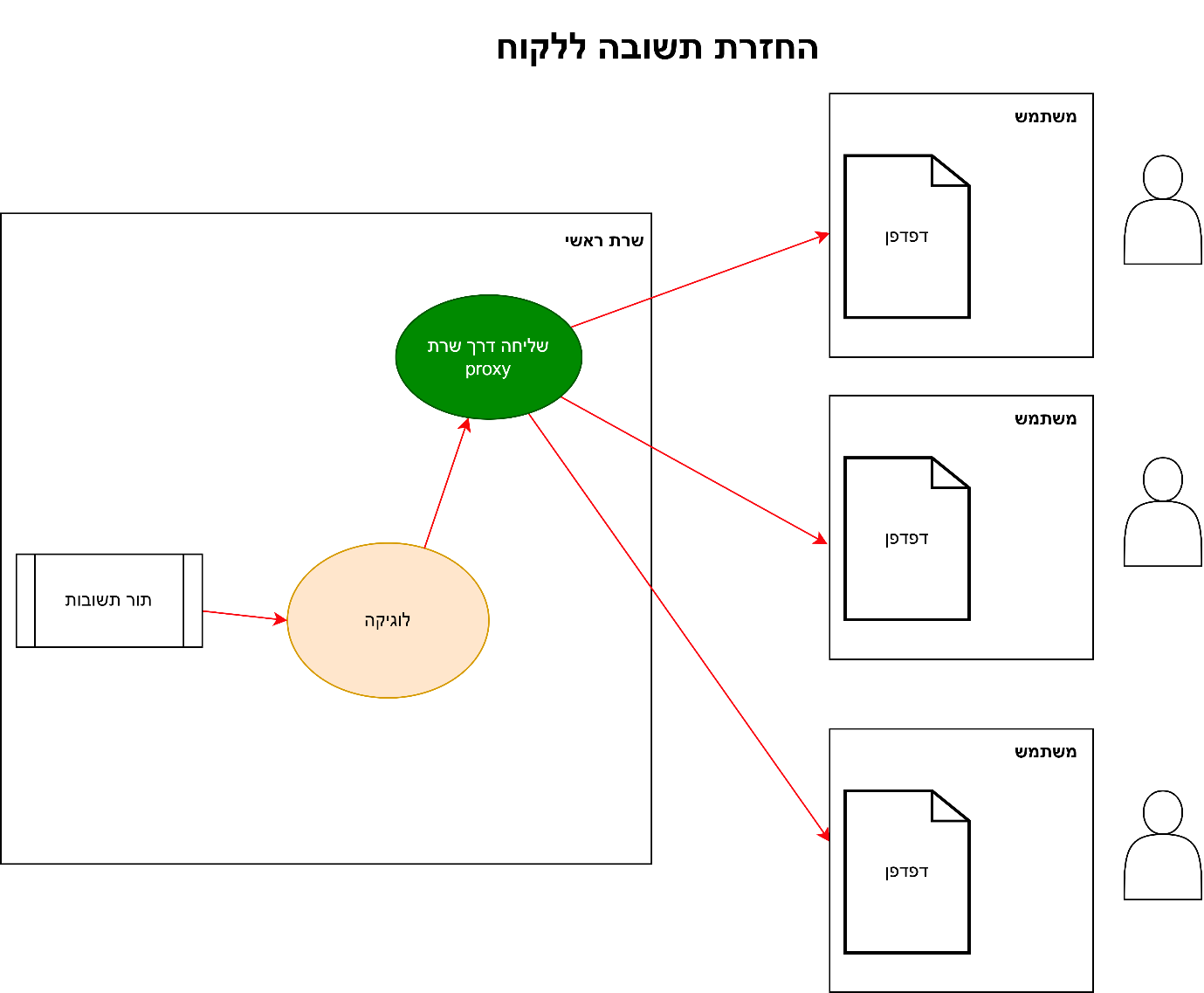
תרשימי הזרימה:



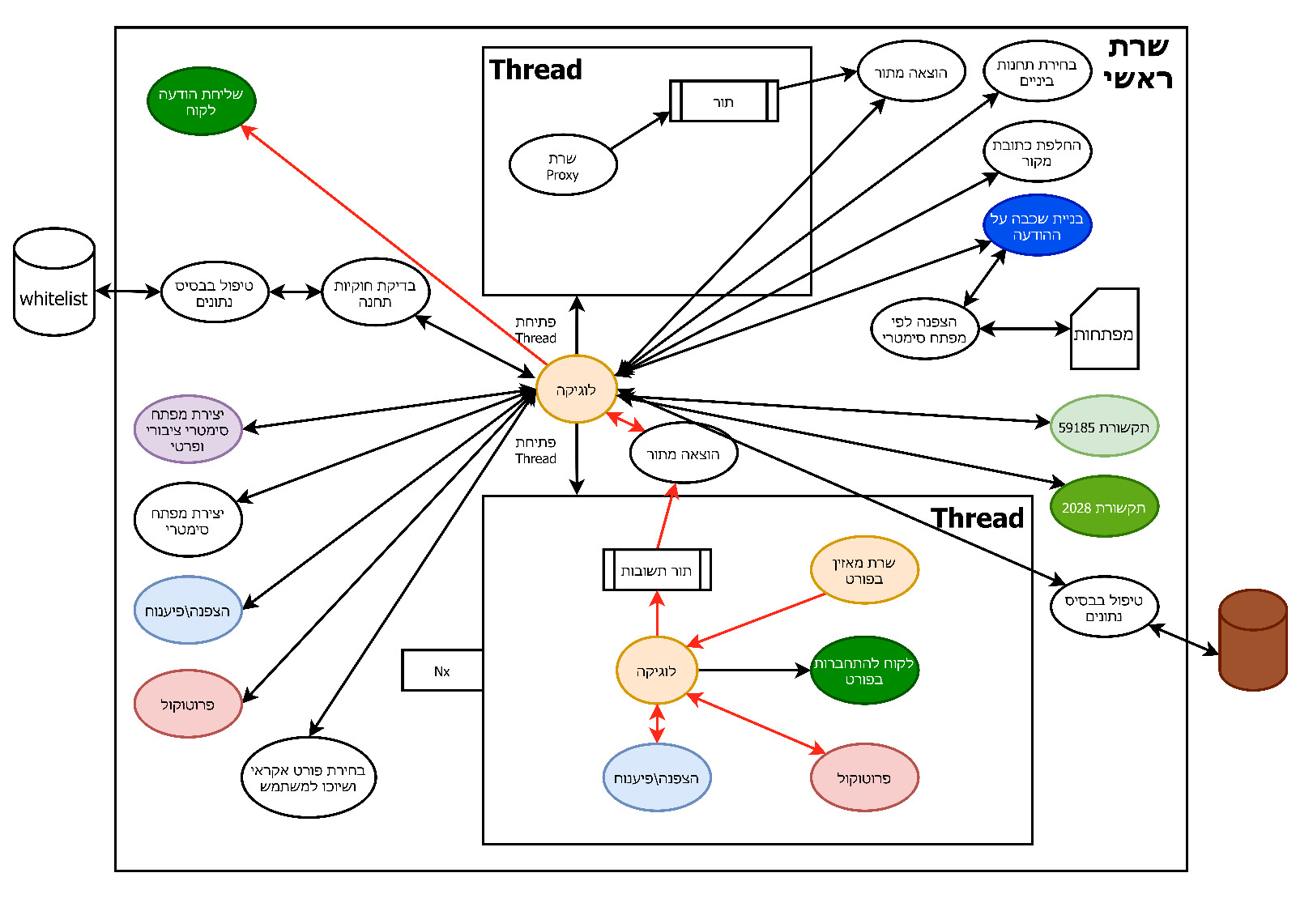




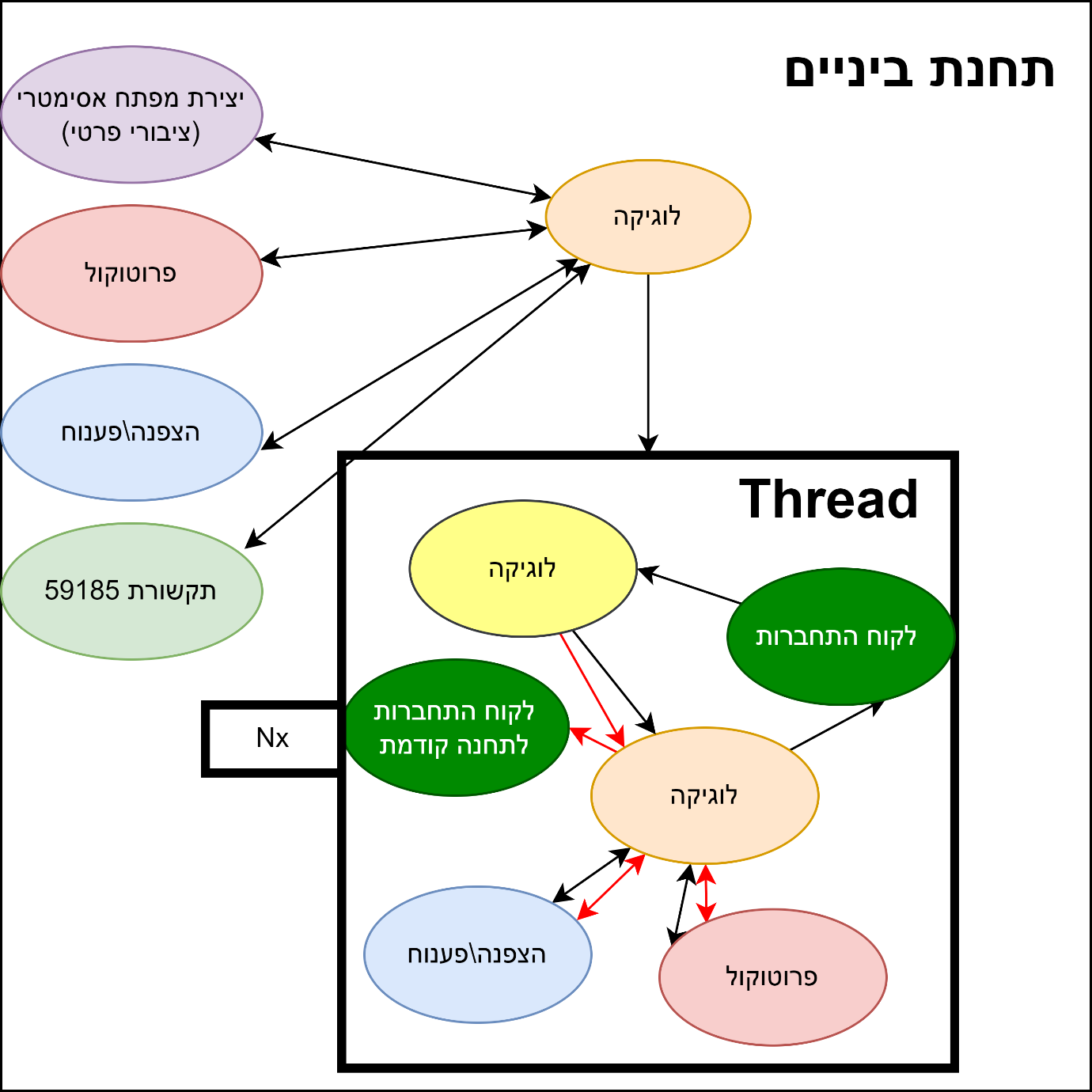


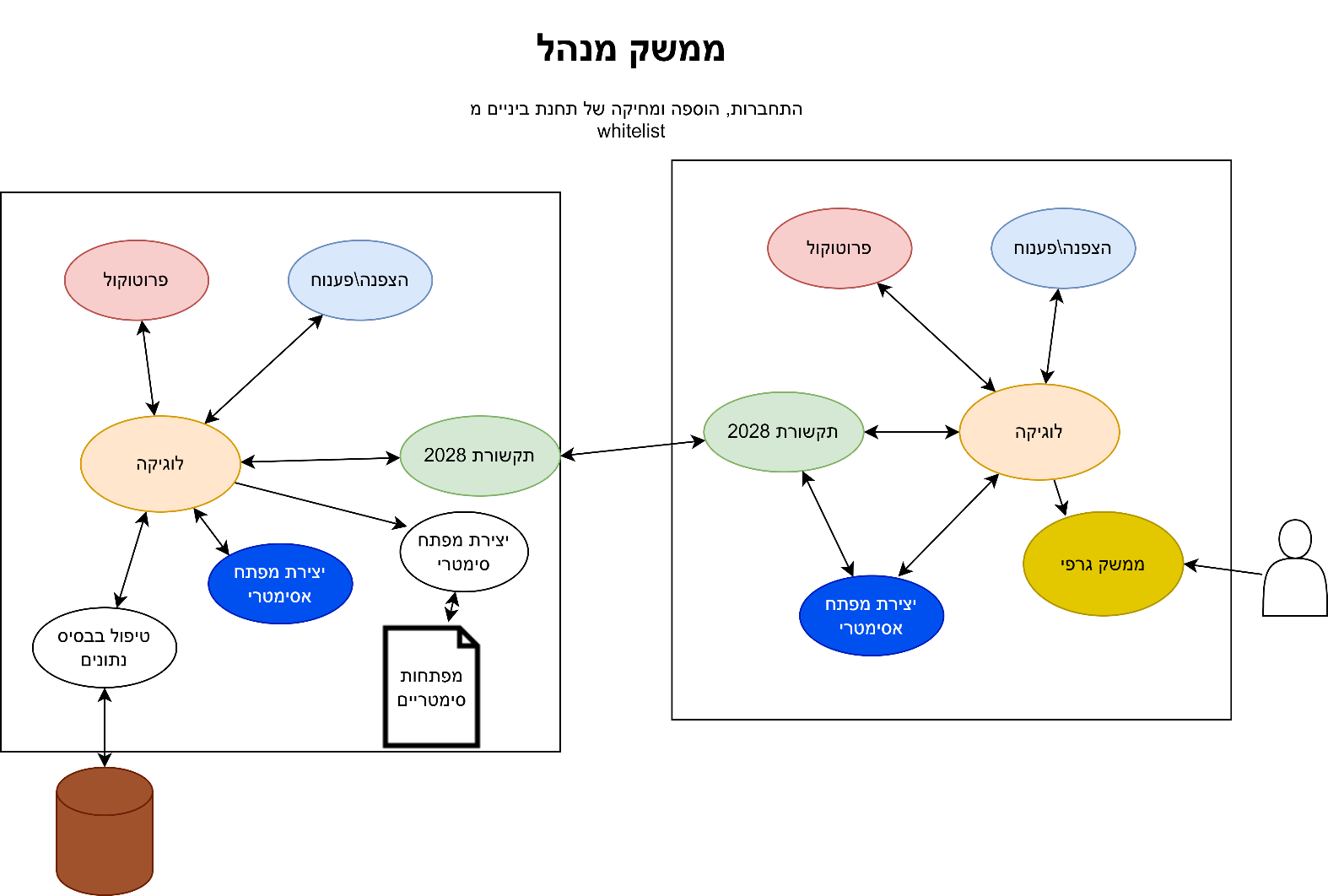


שרת ראשי מסוכם לתרשים אחד:



צד תחנת ביניים מסוכם בתרשים אחד:





מודלים שאני מייבא :

| שם | שימוש |
| --- | --- |
| Random | בחירה רנדומלית של פורט ובחירה רנדומלית של תחנות ביניים |
| AES | משמש להצפנה ופענוח – הצפנה סימטרית |
| RSA | משמש להצפנה ופענוח – הצפנה א-סימטרית |
| Hashlib | משמש להצפנה ופענוח – פונקציית ריבוב |
| Queue | העברת המידע מהthreadים לthread הראשי |
| Thread | פתיחה של מספר threading על מנת להקל על העומס וליצור מקביליות |
| Socket | על מנת ליצור תקשורת בין רכיבים |
| Select | לטובת מימוש שרת מרובה לקוחות |
| Wxpython | ממשק גרפי של המנהל |
| Pyodbc | בסיס נתונים SQL |
| sys | לסגירת הthread או התכנית |

מודלים שלי:

**AESCipher:**

| משמשת להצפנה ופענוח סימטרי לפי אלגוריתם AES | | | |
| --- | --- | --- | --- |
| שם התכונה | גישה | סוג | תיאור |
| key | private | Bytes | המפתח הסימטרי איתו משתמש המחלקה להצפנה והפענוח |
| bs | private | Integer | גודל הבלוק של ההודעה |

|  |  |  |
| --- | --- | --- |
| משמשת להצפנה ופענוח סימטרי לפי אלגוריתם AES | | |
| פונקציה | טענת כניסה | טענת יציאה |
| **\_\_init\_\_(self, key)** | מפתח סימטרי ((String | פעולה בונה |
| **encrypt(self, raw)** | הודעה להצפין (String) | ההודעה אחרי ההצפנה (String) |
| **decrypt(self, enc)** | הודעה מוצפנת (String) | ההודעה אחרי פענוח (String) |

**DB:**

| משמש לטיפול מול בסיס הנתונים SQL | | | |
| --- | --- | --- | --- |
| שם התכונה | גישה | סוג | תיאור |
| conn | private | handler | Handler של בסיס הנתונים |
| Cursor | private | handler | משמש לביצוע פקודות SQL בתוך בסיס הנתונים |
| dbName | private | String | שם בסיס הנתונים |
| adminTbl | private | String | שם טבלת המנהלים |
| stationTbl | private | String | שם טבלת תחנות הביניים |

| פונקציה | טענת כניסה | טענת יציאה |
| --- | --- | --- |
| **\_\_init\_\_(self, DB\_name)** | שם בסיס הנתונים(String) | פעולה בונה לבסיס הנתונים  מפעילה את createTableAdmin  וגם את \_createTableStation |
| **\_createTableAdmin(self)** | כלום | יוצר טבלה לנתוני המנהל |
| **\_createTableStation(self)** | כלום | יוצר טבלה whitelist |
| **addUser(self,username, password** | שם משתמש (String)  סיסמה(String) | מוסיף עוד משתמש לטבלת נתוני המנהל עם הסיסמה והשם הנתון |
| **addStation(self, mac)** | כתובת (String) MAC | מוסיף את ה MAC לwhitelist |
| **checkUser(self, username)** | שם משתמש (String)  סיסמה(String) | בודק האם הסיסמה תואמת לשם  המשתמש |
| **CheckStation(self, mac)** | כתובת (String) MAC | בודק האם הכתובת נמצאת בטבלת הwhitelist |
| **deleteUser(self, username)** | שם משתמש(String) | מוחק את המשתמש, מחזיר אמת או שקר האם הצליח |
| **deleteStation(self, mac)** | כתובת (String) MAC | מוחק את ה MAC מהwhitelist ומחזיר אמת או שקר האם הצליח |
| **update\_password(self, username, new\_password)** | שם המשתמש (String)  הסיסמה החדשה (String) | מעדכן את הסיסמה עבור המשתמש הנתון , מחזיר אמת או שקר האם הצליח  שומר בטבלה את תוצאת ה hash על הסיסמה |

**RSAClass**:

| משמש להצפנה ופענוח בעזרת מפתח אסימטרי לפי אלגוריתם RSA | | | |
| --- | --- | --- | --- |
| שם התכונה | גישה | סוג | תיאור |
| keypair | private | RSAKey | מחזיק במפתח הציבורי והמפתח הפרטי |

| פונקציה | טענת כניסה | טענת יציאה |
| --- | --- | --- |
| **\_\_init\_\_(self)** | כלום | פעולה בונה  מפעיל את \_create\_keys |
| **\_create\_keys(self)** | כלום | יוצר מפתח ציבורי ופרטי |
| **get\_public\_key\_pem(self)** | כלום | ומחזיר את הציבורי |
| **decrypt\_msg(self, encdata)** | ההודעה המוצפנת(String) | מפענח את ההודעה ומחזיר אותה מפוענחת |
| **encrypt\_msg(data, pubkey)** | ההודעה להצפין(String)  והמפתח הציבורי (String) | מצפין את ההודעה עם המפתח הציבורי |

**StationComs:**

| משמש לתקשורת תחנה – **שרת** מצד התחנה | | | |
| --- | --- | --- | --- |
| שם התכונה | גישה | סוג | תיאור |
| sock | private | socket | הסוקט שבעזרתו נוצר החיבור |
| port | private | integer | הפורט אליו יתחבר הסוקט |
| ip | private | String | כתובת ה IP אליו יתחבר הסוקט |
| stationQueue | private | Queue | תור המכיל את כל ההודעות המתקבלות בתקשורת |
| running | private | boolean | דגל הבודק אם עדיין יש צריך להאזין לתקשורת |
| bufferSize | private | integer | גודל הbuffer, בעזרתו יודעים כמה ביטים לקלוט מהתקשורת בפעם אחת |

| פונקציה | טענת כניסה | טענת יציאה |
| --- | --- | --- |
| **\_\_init\_\_(self,port,ip, q)** | פורט (Integer)  IP (String) | פעולה בונה + מפעיל את הפונקציה receive בthread |
| **sendMsg(self,msg)** | הודעה (String) | שולח את ההודעה |
| **receive(self)\_** | כלום | מתחבר לIP בפורט בלולאה אינסופית מחכה להודעה מהסוקט ואז מכניס לתור |

**:ServerComs**

| משמש לתקשורת שרת – תחנת ביניים\לקוח מנהל מצד **השרת** | | | |
| --- | --- | --- | --- |
| שם התכונה | גישה | סוג | תיאור |
| Users\_dict | private | Dictionary | מילון המכיל כמפתח את הsocket ובערך יהיה tuple המכיל פורט ו IP |
| Open\_clients | private | Dictionary | מילון המכיל כמפתח את הIP ובערך את הSocket |
| port | private | Integer | Port בו השרת יאזין |
| serverSock | private | Socket | Socket השרת |
| serverQueue | private | Queue | תור השרת , הערכים הנכנסים אליו יהיו tupleים המכילים IP ואת ההודעה. |
| running | private | boolean | דגל הבודק אם עדיין יש צריך להאזין לתקשורת |
| bufferSize | private | integer | גודל הbuffer, בעזרתו יודעים כמה ביטים לקלוט מהתקשורת בפעם אחת |

| פונקציה | טענת כניסה | טענת יציאה |
| --- | --- | --- |
| **\_\_init\_\_(self,port, serverQueue)** | פורט (Integer)  תור (Queue) | פעולה בונה לשרת  + פותח בthread את \_recieve. |
| **\_receive(self)** | כלום | יוצר את השרת ומאזין להתחברויות לשרת + מאזין להודעות של תחנה\לקוח מנהל ומכניס לתור השרת . |
| **sendMsg(self,ip,msg)** | IP (String)  Msg – ההודעה (String) | שולח את ההודעה לip הנתון באמצעות הסוקט השמור במלון . |

**ServerProtocol:**

| משמש לבניית הודעות לפי הפרוטוקול המוגדר. | | |
| --- | --- | --- |
| פונקציה | טענת כניסה | טענת יציאה |
| **buildPublishPKeySE(Pkey)** | מפתח אסימטרי(String) | בונה הודעה לפי הפרוטוקול עבור פרסום המפתח הציבורי של השרת. |
| **buildSendSymetricKey(sym\_key)** | מפתח סימטרי (AESCipher) | בונה הודעה לפי הפרוטוקול עבור שליחה המפתח הסימטרי לתקשורת בין השרת לתחנה. |
| **buildSendPort(port)** | פורט (Integer) | בונה הודעה לפי הפרוטוקול עבור שליחת הפורט מהשרת לתחנה. |
| **buildSendMsg(msg,passto,lastip)** | ההודעה(String)  Passto- למי להעביר - כתובת IP (String)  Lastip- הIP אליו ההודעה מיועדת (האתר)(String) | בונה הודעה לפי הפרוטוקול עבור העברת ההודעה לתחנה הבאה. |
| **buildLoginMsg(isTrue)** | האם השם משתמש והסיסמה נכונים(True/False) | בונה הודעה לפי הפרוטוקול עבור שליחת תשובה למנהל לגבי האם השם משתמש והסיסמה נכונים. |
| **buildStationInfo(stations\_per\_msg, stations)** | Stations\_per\_msg- מספר התחנות להודעה יחידה  Stations - רשימה של כל התחנות הנמצאות בwhitelist | בונה הודעה לפי הפרוטוקול עבור שליחת כל התחנות ומספר התחנות להודעה למנהל |
| **buildChangeOK** | Current\_num - מספר התחנות להודעה יחידה | בונה הודעה לפי הפרוטוקול עבור אישור על שינוי של מספר התחנות להודעה |
| **buildAddOK** | Mac – כתובת mac שהוספה | בונה הודעה לפי הפרוטוקול עבור אישור על הוספת תחנה |
| **buildChangeOK** | Mac – כתובת mac שנמחקה | בונה הודעה לפי הפרוטוקול עבור אישור על מחיקת תחנה |
| **buildConnectMsg(passto, browserIP, browserPort, msg)** | Passto – כתובת ip של תחנה הבאה  browserIP – IP של האתר  browserPort – port של האתר  msg – הודעת ההתחברות | בונה הודעה על פי הפרוטוקול עבור שליחה הודעת CONNECT |
| **buildSendMsgHTTPS(msg)** | Msg – ההודעה מהלקוח | בונה הודעה על פי הפרוטוקול עבור העברת הודעה של לקוח לאתר דרך התחנות |
| **unpack(msg)** | הודעה (String) | מפרק את ההודעה לפי הפרוטוקול ומחזיר את המידע ואת סוג ההודעה. |

**StationProtocol:**

| משמש לבניית הודעות לפי הפרוטוקול המוגדר. | | |
| --- | --- | --- |
| פונקציה | טענת כניסה | טענת יציאה |
| **buildSendMacAdr(mac)** | Mac - כתובת mac (String) | בונה הודעה לפי הפרוטוקול עבור שליחת כתובת הmac של התחנה. |
| **buildPublishPKeyST(Pkey)** | Pkey - מפתח אסימטרי (String) | בונה הודעה לפי הפרוטוקול עבור פרסום המפתח הציבורי של התחנה. |
| **buildOKmessege()** | כלום | בונה הודעה לפי הפרוטוקול עבור החזרת OK לשרת מהתחנה. |
| **buildSendMsgRet(msg)** | Msg- ההודעה(String) | בונה הודעה לפי הפרוטוקול עבור העברת ההודעה בדרך חזרה ללקוח. |
| **buildSendEstablished(browserIP, browserPort, msg)** | browserIP – כתובת IP של האתר  browserPort – הפורט של האתר  msg – ההודעה עליה בונים את ההודעה החדשה | בונה הודעה לפי הפרוטוקול עבור העברת ההודעת אישור על התחברות לאתר(Connect) |
| **buildSendHTTPS(msg)** | Msg – ההודעה מהאתר | בונה הודעה לפי הפרוטוקול עבור העברת ההודעה מהאתר לשרת דרך התחנות |
| **unpack(msg)** | Msg - הודעה (String) | מפרק את ההודעה לפי הפרוטוקול ומחזיר את המידע ואת סוג ההודעה. |

**ManagerProtocol:**

| משמש לבניית הודעות לפי הפרוטוקול המוגדר. | | |
| --- | --- | --- |
| פונקציה | טענת כניסה | טענת יציאה |
| **buildPublishPKeyMA(Pkey)** | מפתח אסימטרי (String) | בונה הודעה לפי הפרוטוקול עבור פרסום המפתח הציבורי של המנהל. |
| **buildSendUserAndPassword(username,password)** | שם משתמש (String) | בונה הודעה לפי הפרוטוקול עבור שליחת שם המשתמש והסיסמה. |
| **buildAddStationMsg(mac)** | כתובת MAC של התחנה(String) | בונה הודעה לפי הפרוטוקול עבור שליחת כתובת ה MAC של התחנה לשרת אותה המנהל רוצה להוסיף. |
| **buildDeleteStationMsg(mac)** | כתובת MAC של התחנה(String) | בונה הודעה לפי הפרוטוקול עבור שליחת כתובת ה MAC של התחנה לשרת אותה המנהל רוצה להסיר. |
| **buildChNumOfStations(num)** | מספר כמות התחנות הישמשו להעברת הודעה אחת(Integer) | בונה הודעה לפי הפרוטוקול עבור שליחת מספר התחנות הישמשו להעברת הודעה אחת. |
| **unpack(msg)** | הודעה (String) | מפרק את ההודעה לפי הפרוטוקול ומחזיר את המידע ואת סוג ההודעה. |

**OnionServer:**

| משמש לבניית הודעות לפי פרוטוקול Onion לשרת | | |
| --- | --- | --- |
| פונקציה | טענת כניסה | טענת יציאה |
| **buildLayer(msg, ip, lastIP, key)** | Msg - הודעה(String)  Ip - כתובת IP של תחנה (String)  lastIP - כתובת IP של האתר (String)  Key – מפתח סימטרי(AESCipher) | בונה שכבה מעל ההודעה עם הIP הנתון לפי הפרוטוקול של העברת http ומצפין עם המפתח הסימטרי הנתון, מחזיר את ההודעה המוצפנת. |
| **buildLayerConnect(msg, ip, lastIP, browserPort, key)** | Msg - הודעה(String)  Ip - כתובת IP של תחנה (String)  lastIP - כתובת IP של האתר (String)  browserPort – פורט של האתר  Key – מפתח סימטרי(AESCipher) | בונה שכבה מעל ההודעה עם הIP הנתון לפי הפרוטוקול של הודעת connect ומצפין עם המפתח הסימטרי הנתון, מחזיר את ההודעה המוצפנת. |
| **buildLayerHTTPS(msg, key)** | Msg – הודעה(String)  Key – מפתח סימטרי (AESCipher) | בונה שכבה מעל ההודעה עם הIP הנתון לפי הפרוטוקול של העברת https ומצפין עם המפתח הסימטרי הנתון, מחזיר את ההודעה המוצפנת. |
| **removeLayer(msg, key)** | Msg - הודעה (String)  Key - מפתח סימטרי (AESCipher) | מוריד שכבה מההודעה, מפענח עם המפתח הסימטרי ומחזיר את הIP של האתר ושל התחנה ואת ההודעה. |
| **buildLayerAll(msg, ip\_key\_list, lastIP)** | Msg - הודעה (String)  Ip\_key\_list - רשימה של tuple שבכל tuple שמור כתובת הIP של תחנה והמפתח הסימטרי של התחנה  lastIP - IP של האתר (String) | משתמשת בbuildLayer בשביל לבנות את כל השכבות להודעה. מחזיר את ההודעה אחרי הבנייה. |
| **buildLayerAllConnect(msg, ip\_key\_list, lastIP, broswerPort)** | Msg - הודעה (String)  Ip\_key\_list - רשימה של tuple שבכל tuple שמור כתובת הIP של תחנה והמפתח הסימטרי של התחנה  lastIP - IP של האתר (String)  browserPort – פורט של האתר | משתמשת בbuildLayerConnect בשביל לבנות את כל השכבות להודעה. מחזיר את ההודעה אחרי הבנייה. |
| **buildLayerAllHTTPS(msg, ip\_key\_list)** | Msg - הודעה (String)  Ip\_key\_list - רשימה של tuple שבכל tuple שמור כתובת הIP של תחנה והמפתח הסימטרי של התחנה | משתמשת בbuildLayerHTTPS בשביל לבנות את כל השכבות להודעה. מחזיר את ההודעה אחרי הבנייה. |
| **removeLayerAll(msg, key\_list)** | Msg - הודעה (String)  Key\_list - רשימה של מפתחות סימטריים (AESCipher) | משתמש בremoveLayer בשביל להוריד את כל השכבות מההודעה. מחזיר את ההודעה לאחר הורדת השכבות. |

**OnionStation:**

| משמש לבניית הודעות לפי פרוטוקול Onion לתחנה | | |
| --- | --- | --- |
| פונקציה | טענת כניסה | טענת יציאה |
| buildLayer(msg, key) | הודעה(String)  מפתח סימטרי(AESCipher) | בונה שכבה מעל ההודעה עם ההודעה לפי הפרוטוקול ומצפין עם המפתח הסימטרי הנתון, מחזיר את ההודעה הבנויה המוצפנת. |
| removeLayer(msg, key) | הודעה (String)  מפתח סימטרי (AESCipher) | מוריד שכבה מההודעה, מפענח עם המפתח הסימטרי ומחזיר את הIP של האתר ושל התחנה ואת ההודעה בtuple. |

# תיאור סביבת הפיתוח:

**שפת התכנות –** Python, wxPython, SQL

**כלי הפיתוח הנדרשים לפיתוח –** Scapy , pycharm

**כלים הנדרשים לבדיקות:** Wireshark, pycharm-debugger , procmonitor

# תיאור האלגוריתמים המרכזיים בפרויקט:

**בעיה 1** : איך להצפין את המידע בין השרת הראשי לתחנות הביניים. הצפנת המידע הכרחית כחלק מהאבטחה בפרויקט

| חלופה | יתרונות | חסרונות |
| --- | --- | --- |
| שימוש במפתח אסימטרי להצפנת כל המידע לדוגמא RSA | חסין כנגד פריצות  ייחודי לתחנה (לכל תחנה יש מפתח פרטי ומפתח ציבורי ייחודי) | איטי מאוד, תהליך יצירת המפתח איטי וגם תהליך ההצפנה והפענוח איטי |
| שימוש במפתח סימטרי לדוגמא AES | מהיר | יש צורך בהעברת המפתח הסימטרי בתחילת התהליך בתקשורת  יש בעיה בלאמת את אמינות המידע |
| שימוש במפתח סימטרי ע"י דיפי-הלמן | תהליך היצירה מהיר  לא דורש העברת המפתח בתקשורת | לא עמיד בפני MITM |
| משולבת – שימוש במפתח סימטרי להצפנת המידע + שימוש במפתח אסימטרי להעברת המפתח הסימטרי | חסין  תהליך ההצפנה והפענוח מהיר  העברת המפתח הסימטרי סמויה וחסינה באמצעות המפתח האסימטרי  שימוש במפתח סימטרי לטובת הבטחת אמינות ושלמות מידע בעזרת אלגוריתם AES מסוג OCB | יצירת המפתח האסימטרי עדיין איטית |

קישור להצפנת :RSA https://he.wikipedia.org/wiki/RSA

קישור להחלפת מפתח דיפי-הלמן: https://he.wikipedia.org/wiki/%D7%A4%D7%A8%D7%95%D7%98%D7%95%D7%A7%D7%95%D7%9C\_%D7%93%D7%99%D7%A4%D7%99-%D7%94%D7%9C%D7%9E%D7%9F

קישור להצפנה סימטרית: <https://he.wikipedia.org/wiki/AES>

קישור להסבר על OCB - https://en.wikipedia.org/wiki/OCB\_mode

**החלופה בה בחרתי: החלופה המשולבת** – שימוש במפתח סימטרי להצפנת המידע + שימוש במפתח אסימטרי להעברת המפתח הסימטרי

**בעיה 2**: טיפול בבקשות המשתמש, המגיעות אל השרת, השרת מקבל בקשות (get) ממשתמשים שונים בו זמנית  
השרת יצטרך לשלוח את הבקשות שהוא מקבל לתחנות הביניים וגם לחכות לתשובה מהן

| חלופה | יתרונות | חסרונות |
| --- | --- | --- |
| הבקשות מטופלות לפי סדר הגעתן | פשוט לטיפול | במידה וטיפול בבקשה אחת מתארך, כל שאר הטיפול מתעכב, יכול להיווצר תוך ארוך של בקשות  יעמיס על השרת , עומס יגרום לאיטיות של שאר התהליכים בשרת |
| Thread אחד יטפל בכל הבקשות | השרת לא יהיה עמוס, ויוכל לטפל בדברים אחרים במקביל לטיפול בבקשות | הthread יהיה עמוס , הבקשות יעברו באיטיות ועדיין עיכוב בבקשה אחת יעכב את שאר הבקשות ויוצר תור ארוך |
| Thread לכל בקשה | גם השרת וגם כל thread לא יהיו עמוסים, כך שכל הבקשות יעברו במהירות במקביל לדברים נוספים שהשרת יטפל בהם | יוצרו הרבה thread-ים במערכת |

**החלופה בה בחרתי: Thread לכל בקשה –** מאוד חשובה המהירות בפרויקט שלי , Thread לכל בקשה עונה על הבקשה ואין בו חסרונות.

**בעיה 3:** העברת תוצאות הבקשה (מהאינטרנט) בחזרה

| חלופה | יתרונות | חסרונות |
| --- | --- | --- |
| העברת ההודעה ישירות מהאתר ללקוח בעזרת שינוי ה source | העברה מהירה של ההודעה ללקוח | הip של הלקוח יהיה חשוף לאתר  (מה שהמערכת מנסה למנוע) |
| העברת ההודעה ישירות מאתר לשרת המרכזי | העברה מהירה של ההודעה ללקוח  הip של הלקוח חסוי לאתר | ההעברה תחשוף מי הוא השרת המרכזי והוא יכול להפוך ליעד לתקיפות |
| העברת ההודעה דרך תחנות הביניים באותה הדרך בה הגיעה הבקשה | ההודעה תהיה מוצפנת ברשת  ה IP של הלקוח והשרת יהיו חסויים לאתר | דורש תקשורת בין מספר תחנות ביניים ובכך מאט את החזרת ההודעה ללקוח. |

**החלופה בה בחרתי : העברת ההודעה דרך תחנות הביניים-** אחת מהמטרות העיקריות של הפרויקט הוא שהמידע יעבור ברשת בצורה בטוחה ואמינה מבלי לחשוף את הip של הלקוח .

**בעיה 4**: הפורט בו תעבור ההודעה

| חלופה | יתרונות | חסרונות |
| --- | --- | --- |
| לשלוח ולקבל על אותו הפורט בשביל כל ההודעות | פשוט | יעמיס מאוד על התקשורת ובכך יאט את קבלת ושליחת ההודעה  מחייב לכתוב קוד לשיוך פורט להודעה |
| לבחור פורט אקראי לכל הודעה | לתוקף יהיה יותר קשה לעקוב אחרי הודעה ברשת  לא מעמיס על הרשת, ובכך תורם להעברה מהירה של הודעות  עוזר לשייך פורט להודעה בצורה אוטומטית | דורש אלגוריתם בחירה ושליחת הפורט בתקשורת. |

**החלופה בה בחרתי: לבחור פורט אקראי לכל הודעה –** בפרויקט מאוד חשובה המהירות של החזרת התשובה מהשרת, ככל שהמשימות יהיו אוטומטיות המהירות תגבר ולכן דרוש פורט אקראי לכל הודעה.

**בעיה 5:** שמירת הנתונים על תחנות הביניים והמנהל

| חלופה | יתרונות | חסרונות |
| --- | --- | --- |
| שמירת הנתונים מקודדים בקוד השרת | קל מאוד לכתיבה | כל המידע שרשום בקוד יכול להיות משוחזר על ידי IDA ולהיחשף לכולם  אם השרת ייפול כל המידע ייהרס  ולא ניתן יהיה לשמור מידע על תחנות מאושרות |
| בבסיס נתונים SQL | מסודר  פשוט לתפעול  לא חשוף לאף משתמש  אם השרת ייפול המידע לא ייהרס | מצריך טיפול מיוחד עבור שמירת הנתונים בבסיס הנתונים – קוד ייעודי |

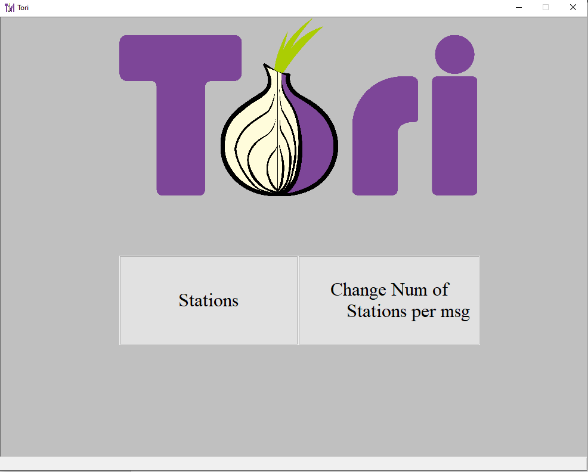
**החלופה בה בחרתי: בסיס נתונים SQL –** שמירה בקוד השרת היא מסוכנת ואין צורך בה , אסור שהמידע על התחנות או המנהל יוכל להיחשף.

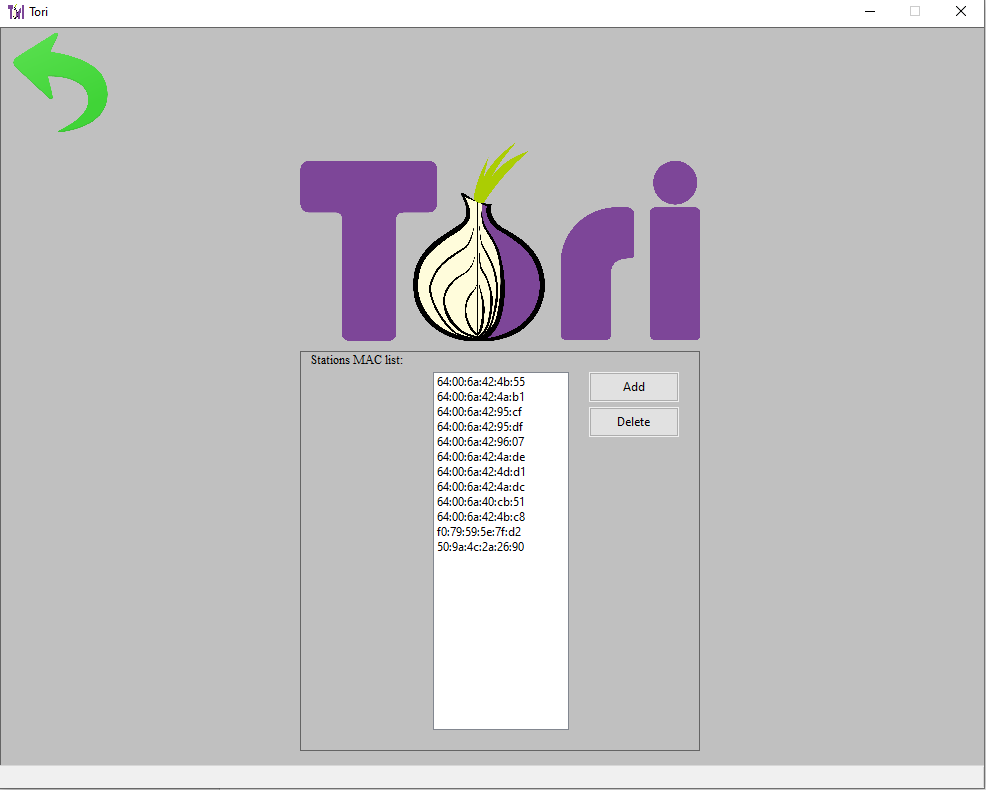
קישור ל SQL - <https://he.wikipedia.org/wiki/SQL>

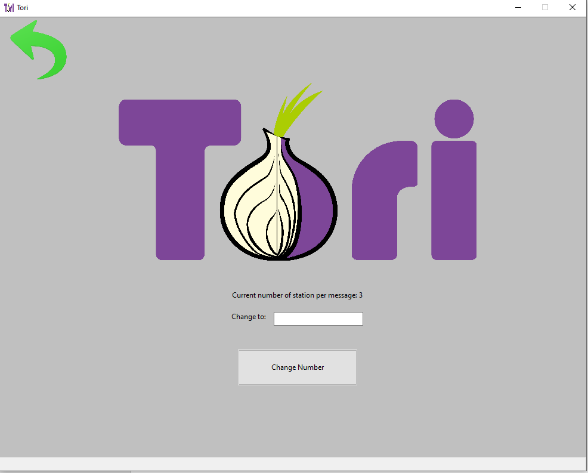
# תיאור מסכי הפרויקט:

מסכי הפרויקט נמצאים רק אצל ממשק המנהל

התרשים כאן מראה את מסכי הממשק:







# תיאור פרוטוקול התקשורת:

כל הודעה בפרוטוקול תיהיה בפורמט הבא:

Code – 2 bytes DATA

ההודעה תתחיל בקוד (מספר בעל שתי ספרות) בעזרתו יהיה אפשר לדעת באיזה סוג הודעה מדובר

ולאחר מכן headers נוספים בהתאם להודעה

ולבסוף המידע

| שם ההודעה | מ- | אל | קוד ההודעה | מבנה | דוגמא |
| --- | --- | --- | --- | --- | --- |
| שליחת כתובת הMAC | תחנה | שרת | 00 | קוד ההודעה – 00  כתובת mac – 17 תווים | 00ff:4a:49:f9:a6:b2 |
| פרסום מפתח ציבורי של השרת | שרת | תחנה | 01 | קוד הודעה - 01  אורך – 8 בתים  מפתח ציבורי - ? | 0100000008##$$;;12 |
| שליחת מפתח ציבורי של התחנה\מנהל | תחנה | שרת | 02 | קוד הודעה - 02  אורך – 8 בתים  מפתח ציבורי - ? | 0200000008##\*\*;;12 |
| שליחת מפתח סימטרי | שרת | תחנה\מנהל | 03 | קוד הודעה - 03  אורך – 2 בתים  מפתח סימטרי – עד 18 תווים | 0318example12345678901 |
| שליחת פורט | שרת | תחנה | 04 | קוד הודעה - 04  הפורט – 5 תווים | 0459185 |
| אישור | תחנה | שרת | 05 | קוד הודעה - 05 | 05 |

| שם ההודעה | מ- | אל | קוד ההודעה | מבנה | דוגמא |
| --- | --- | --- | --- | --- | --- |
| העברת ההודעה  HTTP | Thread שרת  \Thread תחנה | Thread תחנה | 06 | קוד הודעה - 06  אורך IP ראשון – 2 תווים  למי להעביר – 15 תווים  אורך IP אחרון – 2 תווים  מי הip האחרון – 15 תווים  אורך החבילה המוצפנת – 8 תווים  החבילה המוצפנת | 0609192.1.1.8009193.2.2.304911200045123//xxz//wda/dasw##? |
| החזרת ההודעה | Thread תחנה | Thread תחנה\  Threadשרת | 07 | קוד ההודעה- 07  אורך החבילה – 8 תווים  החבילה המוצפנת | 0700054321#swasdwsg… |
| שליחת שם משתמש וסיסמה | מנהל | שרת | 08 | קוד ההודעה – 08  אורך שם – 2 תווים  שם משתמש – 16 תווים  אורך סיסמה – 2 תווים  סיסמה – 16 תווים | 0805admin05admin |
| שליחת אישור או דחייה על שם המשתמש והסיסמה | שרת | מנהל | 09 | קוד ההודעה – 09  אורך – תו 1  אמת\שקר – 4\5 תווים | 094true  או  095false |
| שליחת מידע על הודעה | שרת | מנהל | 10 | קוד ההודעה – 10  מספר תחנות שבו עוברת הודעה – תו אחד  כתובות MAC של תחנות הביניים – 17\*מספר התחנות שבו עברה ההודעה | 10364:00:6a:42:4b:55 |
| הודעת הוספת תחנה | מנהל | שרת | 11 | קוד ההודעה – 11  כתובת MAC של התחנה – 17 תווים | 11ff:4a:49:f9:a6:b2 |
| הודעת מחיקת תחנה | מנהל | שרת | 12 | קוד ההודעה – 12  כתובת MAC של התחנה – 17 תווים | 12ff:4a:49:f9:a6:b2 |
| הודעת שינוי מספר התחנות להודעה | מנהל | שרת | 13 | קוד ההודעה – 13  מספר תחנות להודעה – תו אחד | 138 |
| הודעת אישור על החלפת מספר התחנות להודעה | שרת | מנהל | 14 | קוד הודעה- 14  מספר תחנות להודעה חדש – תו אחד | 145 |
| הודעת אישור על הוספת תחנה | שרת | מנהל | 15 | קוד הודעה – 15  כתובת התחנה – 17 תווים | 15ff:4a:49:f9:a6:b2 |
| הודעת אישור על מחיקת תחנה | שרת | מנהל | 16 | קוד הודעה -16  כתובת התחנה – 17 תווים | 16ff:4a:49:f9:a6:b2 |
| העברת הודעת connect | Thread שרת  \Thread תחנה | Thread תחנה | 17 | קוד הודעה - 17  אורך IP ראשון – 2 תווים  למי להעביר – 15 תווים  אורך IP אחרון – 2 תווים  מי הip האחרון – 15 תווים  אורך החבילה המוצפנת – 8 תווים  החבילה המוצפנת | 1709192.1.1.8009193.2.2.304911200045123//xxz//wda/dasw##? |
| החזרת הודעה שהתחברות עברה בהצלחה | תחנה | שרת | 18 | קוד הודעה – 18  אורך IP אתר – 2  IP אתר- עד 15 תווים  אורך פורט אתר- תו אחד  פורט אתר – 5 תווים | 1809192.1.1.804312409193.2.2.304911223 |
| הודעת HTTPS מהלקוח לאתר | שרת | תחנה | 19 | קוד הודעה – 19  אורך החבילה המוצפנת – 8 תווים  החבילה המוצפנת | 1900045123//xxz//wda/dasw##? |
| הודעת HTTPS מהאתר ללקוח | תחנה | שרת | 20 | קוד הודעה – 20  אורך ההודעה – 8 תווים  ההודעה המוצפנת | 2000058976\bb\\xaz\wad… |

# תיאור מבני הנתונים:

**שם בסיס הנתונים** : **ToriDB**

בסיס הנתונים הוא מסוג SQL

בסיס הנתונים יכיל שתי טבלאות:

טבלה 1: טבלה המשמשת למידע על תחנות הביניים

שם טבלה: **whitelist**

|  |  |
| --- | --- |
| שם השדה | MacAdr |
| תיאור | MAC – כתובת הmac של התחנה (משמש כkey) , String באורך 17 תווים |
| דוגמה | ff:4a:49:f9:a6:b2 |

טבלה 2 : טבלה של נתוני המנהל

שם הטבלה: **users**

|  |  |  |
| --- | --- | --- |
| תיאור | סיסמה, String באורך של בין 8 ל 16 תווים (נשמר רק הhash של הסיסמה) | שם משתמש (משמש כ key) , String באורך של בין 8 ל 16 תווים |
| שם השדה | password | username |
| דוגמה | Dax12daw97bkwgdyaw | admin |

**שם מבנה הנתונים**: **ip\_key\_dict**

בסיס הנתונים הוא מסוג dictionary

|  | תיאור | דוגמה |
| --- | --- | --- |
| key | IP של התחנה | 27.231.154.8 |
| value | מפתח סימטרי של התחנה | Example12345678901 |

**שם מבנה הנתונים: ports\_stations**

בסיס הנתונים הוא מסוג dictionary

|  | תיאור | סוג | דוגמה |
| --- | --- | --- | --- |
| key | port | integer | 54871 |
| value | IP של התחנות | String | (16.135.214.87, 16.135.214.88, 16.135.214.89) |
| value | לקוח שולח של תחנה ראשונה | StationComs |  |
| value | כתובת IP ופורט של לקוח | integer | 27.231.154.82, 9982)) |

**שם מבנה הנתונים: ip\_mac\_dict**

בסיס הנתונים הוא מסוג dictionary

|  | תיאור | סוג | דוגמה |
| --- | --- | --- | --- |
| key | IP של תחנה | String | 16.135.214.87 |
| value | mac | String | aa:bb:cc:dd:ee:ff |

**שם מבנה הנתונים: symkey\_list**

בסיס הנתונים הוא מסוג list

|  | תיאור | סוג | דוגמה |
| --- | --- | --- | --- |
| value | mac | String | aa:bb:cc:dd:ee:ff |

# ניתוח החולשות והאיומים:

אפליקציה:

* **MITM** - יש סיכוי שמישהו מסניף את ההודעות המועברות לתחנות הביניים, ה  
  פתרון: מצפינים את ההודעות המועברות בין התחנות בעזרת מפתחות סימטריים כך שלאותו הבן אדם שמסניף לא יהיה מה לעשות עם המידע. MITM  
   לא יכול גם לחבל בתקשורת ולחתוך חלקים מההודעה כי בהצפנת AES יש אימות שהחבילה הגיעה בצורה נכונה.
* תהליך **הlogin** לממשק מנהל – יש חשש שבן אדם שהוא לא המנהל יכנס לממשק המנהל, הפתרון הוא אימות, כלומר נבקש מהבן אדם שרוצה להיכנס לממשק מנהל להכניס שם משתמש וסיסמה, רק אם הם נכונים תתאפשר לו הגישה לממשק.
* **SQL injection** - תמיד יש חשש לSQL injection כשמשתמשים בSQL , הפתרון לבעיה היא הכנסה בטוחה של פרמטרים בפקודת SQL, python פותר לנו את הבעיה עם נכתוב f לפני הגרש של הString.
* **פריצה לבסיס הנתונים** וגניבת זהות – במידה ואדם הצליח לפרוץ לבסיס הנתונים יש חשש שהוא יוכל לקבל את הסיסמה של המנהל, לכן הפתרון הוא להכניס לבסיס הנתונים רק את הhash של סיסמת המנהל , כך שגם אם הפורץ יקבל גישה לבסיס הנתונים הוא לא יוכל לגלות את הסיסמה.
* **Brute force** לסיסמת המנהל , נשתמש בcaptcha בlogin לממשק המנהל כך שלא יתאפשר brute force לסיסמה והשם.
* **DDOS** - כמות גדולה של לקוחות בו זמנית עלולה להרוס את מהירות התקשורת, לכן עבור כל לקוח יש port שונה , כך שתקשורת עם לקוח אחד לא תשפיע על תקשורת עם לקוח אחר.

שכבת התעבורה:

* חשוב מאוד לפרויקט שכל חבילה תגיע ליעד שלה בסדר הנכון, לכן החיבור שנצטרך לבצע הוא חיבור מסוג TCP שידאג לכך שכל החבילות יגיעו ליעד שלהן בסדר הנכון. וכיום מתקפת לחיצת יד-משולשת כבר לא ישימה בפרוטוקול זה

פרק ו – בדיקות

| מספר | שם הבדיקה | מה אמורה לבדוק | איך מתכננים לבדוק | איך התבצעה הבדיקה | תאריך | איך התבצעה הבדיקה אחרי https | תאריך  אחרי  https |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1. | העברת החבילה לשרת | שהחבילה הגיעה לשרת לאחר ניסיון התחברות לאתר. | מתחברים לאתר ובודקים עם הסנפה שההודעה באמת מגיעה לשרת | נכנסתי לאתר בזמן שהרצתי wireshark וראיתי שאכן ההודעה נשלחת לשרת | 8.2.22 | נכנסתי לאתר בזמן שהרצתי wireshark ואריתי שאכן ההודעה נשלחה לשרת | 28.2.22 |
| 2. | זמני תגובה | בודקת שהזמן משליחת ההודעה ועד לקבלת התשובה לא גדול מדי. | שולחים הודעה , ומודדים את הזמן מתחילת ההודעה ועד לקבלת תשובה. | נכנסתי לאתר בזמן שהמערכת רצה ומדדתי עם שעון וראיתי שזמן התגובה הוא 0.22 שניות | 8.2.22 | נכנסתי לאתר בזמן שהמערכת רצה ומדדתי עם השעון בטלפון וראיתי שהזמן עד שהעמוד נטען הוא 6 שניות | 28.2.22 |
| 3. | בדיקת IP | בודקת שהIP שנשלח לשרת הוא לא של הלקוח. | מסתכלים על הפלט של כל אחד מתחנות הביניים , כל אחת מהתחנות תדפיס את הip של החבילה שהיא מקבלת. | אף אחת מהתחנות לא יכלה להדפיס את הip של המשתמש , רק את התחנה שהיא מעבירה אליו ואת התחנה ממנה מקבלת | 8.2.22 | אף אחת מהתחנות לא יכלה להדפיס את הip של המשתמש , רק את התחנה שהיא מעבירה אליו ואת התחנה ממנה מקבלת | 28.2.22 |
| 4. | הצפנה | בדיקה שההצפנה עובדת | מנסים להסניף את החבילה , ובודקים האם ניתן לקרוא אותה. | הסנפתי את החבילה וראיתי שהיא מוצפנת כאשר היא עוברת מתחנה לתחנה. | 8.2.22 | הסנפתי את החבילה וראיתי שהיא מוצפנת כאשר היא עוברת מתחנה לתחנה. | 28.2.22 |
| 5. | התמודדות עם תחנה שקורסת | בודקת שהמערכת ממשיכה לעבוד למרות סגירת תחנה | פותחים לקוח, בודקים למי נשלח ההודעה בפעם הראשונה , וסוגרים את התחנה . אחר כך בודקים שההודעה נשלחת שוב לתחנה אחרת. | הרצתי 4 תחנות ואז ניתקתי אחת כך שעדיין יהיה 3 עובדות, והמערכת לא נפלה | 10.2.22 | הרצתי 3 תחנות , ניתקתי אחת תוך כדי שהמערכת עובדת ואחר כך הרצתי שוב והמערכת המשיכה לעבוד כרגיל | 28.2.22 |
| 6. | טיפול במספר לקוחות | בודקת שהשרת יכול לתפעל עם מספר לקוחות | מנסים להיכנס עם שני לקוחות בו זמנית ובודקים שהכל פועל כמתוכנן. | הרצתי 3 לקוחות בו זמנית וכולם קיבלו את התשובה שלהם | 10.2.22 | הרצתי עם 3 לקוחות בו זמנית  אחד נכנס לwalla השני נכנס לstackoverflow והשלישי נכנס לyoutube ולכולם עלה העמוד | 28.2.22 |
| 7 | עומסים | בודקת האם המערכת מסוגלת לעמוד בעומס בקשות | נכנסים לשרת מעמדות שונות בו זמנית (יותר משתיים) ובודקים זמני תגובה בכל תחנה | כל שלושת הלקוחות (אותה בדיקה כמו הקודמת) קיבלו את התשובה שלהם בזמן טוב | 10.2.22 | כאשר כל שלושת הלקוחות היו מחוברים זמן התגובה לא היה שונה בהרבה | 28.2.22 |
| 8. | תשובה מלאה | בדיקה האם כל התשובה חוזרת במלואה ללקוח | נכנסים עם הלקוח לאתר ובודקים שהאתר נטען במלואו | כל אחד הלקוחות קיבל את התשובה המלאה , בדקתי את האתר בלי המערכת ופעם אחת עם המערכת וראיתי שאכן הכל אותו דבר | 10.2.22 | כל הלקוחות קיבלו תשובה מלאה , בדקתי את האתר עם ובלי המערכת ואין הבדלים | 28.2.22 |

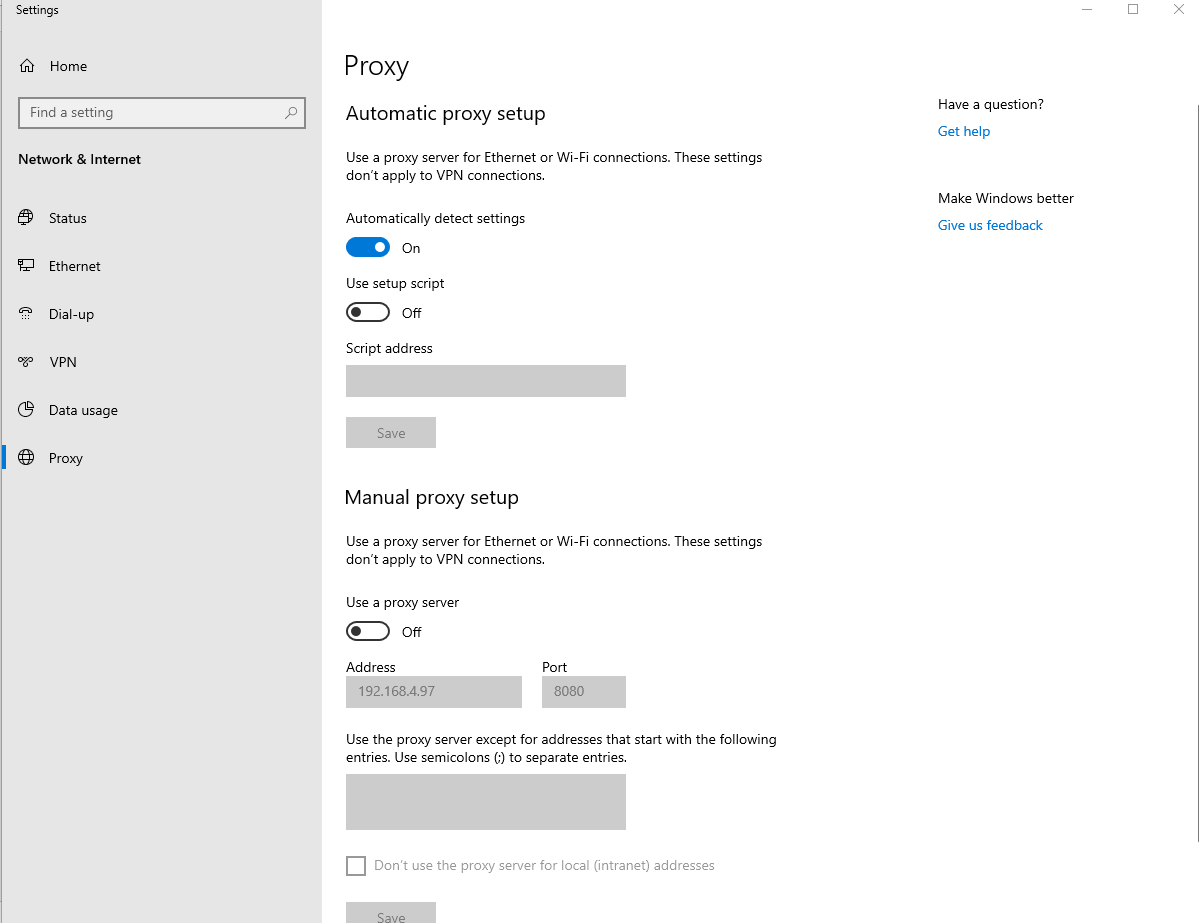
פרק ז – מדריך למשתמש

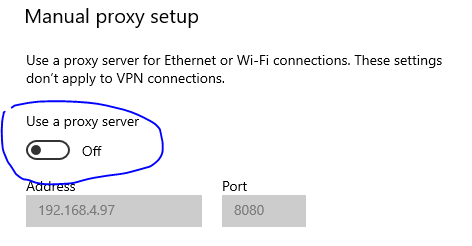
# איך מריצים סרבר ותחנות:

קודם כל צריך לבדוק שכל התחנות מוגדרות כך שהם יתחברו לip של הסרבר (אפשר לשנות בשורה מספר 107 כרגע) , למחשב אחד צריך להביא את כל הקבצים בתיקייה “server” ויש להתחיל את הקובץ server.py , הוא מתחיל את הסרבר, לאחר מכן יש להוריד ל3 מחשבים לפחות את הקבצים בתיקייה “station” ולהריץ עליהם את הקובץ station.py . צריך להסתכל בפלט של הסרבר ולראות שאכן התחברות והחלפת מפתחות הצליחה, במקרה שסיים להחליף מפתחות יהיה בפלט “SYMKEY” ואת המפתח הסימטרי. במידה ולא הצליח להתחבר בגלל שהתחנות לא נמצאו בwhitelist זאת אומרת שהMAC של המחשבים לא נמצא ברשימה של הmac המותר ויש להוסיף אותו. דרך הmanager.py או שהולכים ישירות לקובץ הsql ומוסיפים ידנית את הmac לפי הפורמט הנמצא שם.

לאחר שנפתחו 3 תחנות והתבצע החלפת מפתחות הסרבר יוכל לקבל הודעות מהפרוקסי כלומר אפשר להתחיל עם הגלישה באינטרנט.

במחשב שבו רוצים לגלוש צריך קודם כל להגדיר את הproxy:

1. נכנסים להגדרות פרוקסי
2. מדליקים את המתג של “Use a proxy server”



1. מכניסים את פרטי הפרוקסי, כרגע כתובת הפרוקסי היא כתובת השרת והפורט הוא 8080.

לוחצים על save ואפשר להתחיל לגלוש.

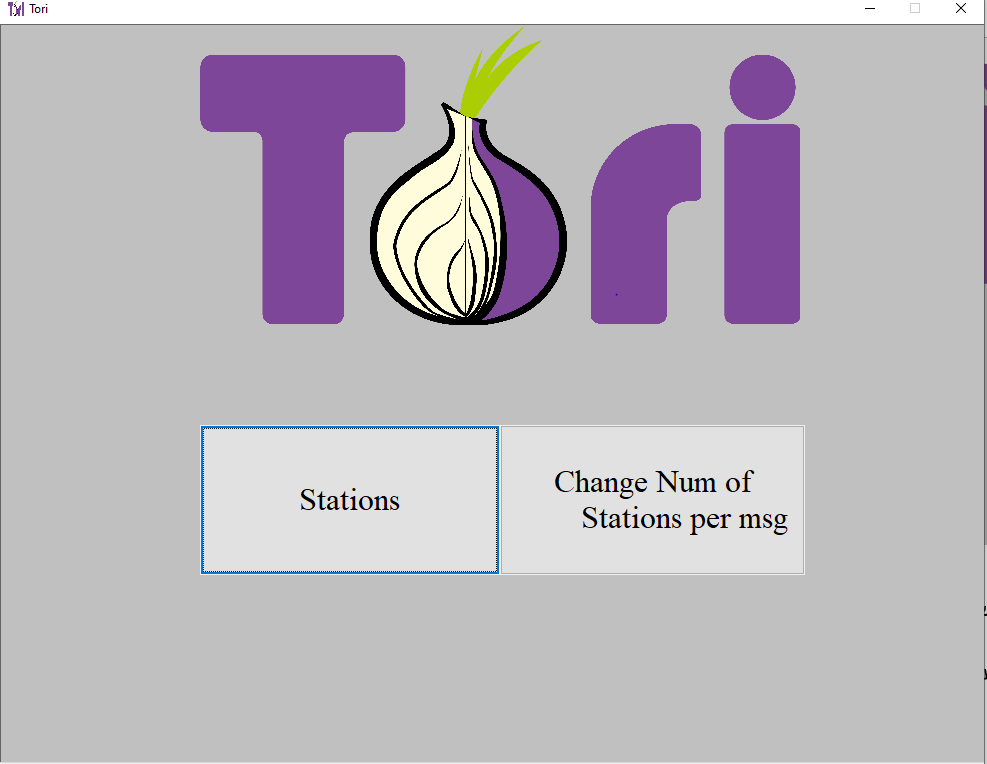
# איך מריצים את המנהל:

מורידים למחשב מסוים את הקבצים המצויים בתיקייה “manager” ומריצים את manager.py , יש לוודא שהIP שאליו מכוון המנהל הוא באמת הIP של השרת (שורה 671 בינתיים).

לאפליקציה ייקח כמה שניות להתחיל בגלל החלפת המפתחות.

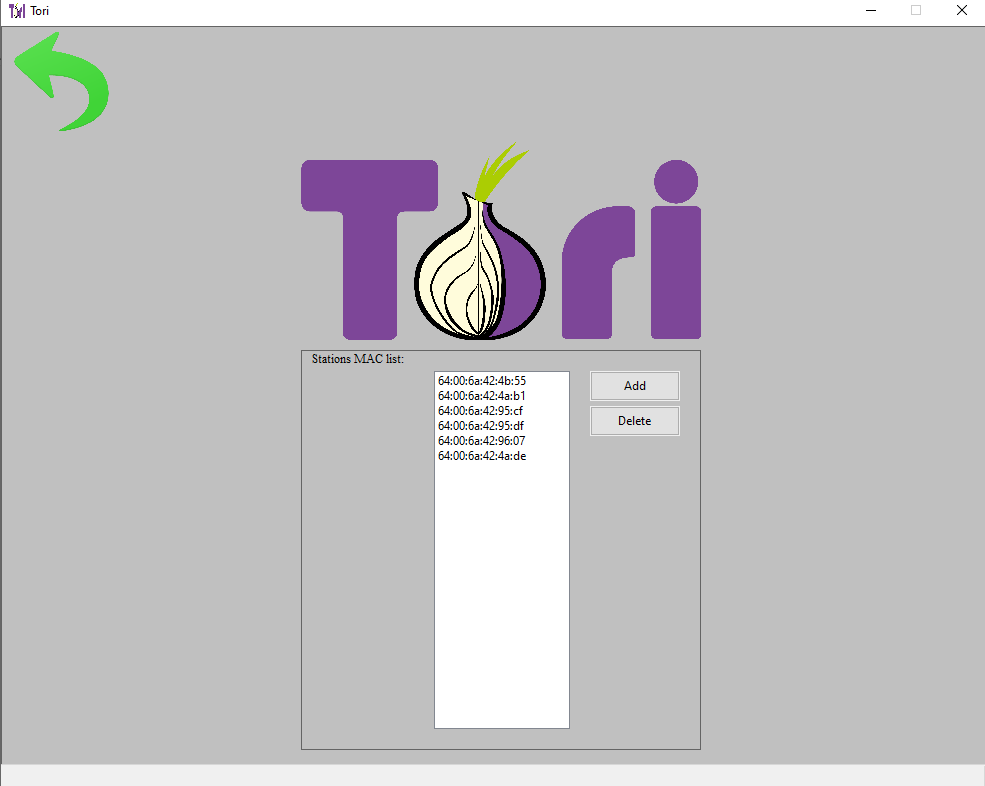
לאחר שהאפליקציה תחל היא תראה את מסך הכניסה: 

כרגע האפליקציה מבקשת שם משתמש וסיסמא על מנת שיהיה אפשר להשתמש באפליקציה, כרגע השם והסיסמא הם admin.

לאחר שמתחברים מגיעים למסך השני: 

אם לוחצים על מקש הStations מגיעים למסך התחנות, ואם לוחצים על מקש הChange Num of Stations per msg מגיעים למסך המשנה את מספר התחנות הדרושות להודעה.

מסך התחנות:



מסך התחנות מציע שתי יכולות: הוספת תחנה ומחיקת תחנה, בשביל להוסיף תחנה יש ללחוץ על כפתור הAdd

שיפתח חלון קטן שיאפשר כתיבת mac , לתוכו יש לכתוב את הmac של התחנה לפי הפורמט כלומר בין כל שני תווים צריך להיות ":" .

בשביל למחוק תחנה צריך פשוט ללחוץ על הכתובת שרוצים למחוק ולוחצים על Delete.

מסך החלפת מספר תחנות להודעה:



במסך זה אפשר להחליף את מספר התחנות הדרושות להודעה, רושמים את מספר התחנות הדרושות להודעה בשדה הריק ולוחצים על change number.

מעל השדה אפשר לראות את המספר הנוכחי של תחנות הדרושות להעברת ההודעה.

גם במסך הזה וגם במסך הקודם יש כפתור "חזור" שמחזיר למסך של התפריט.

פרק ח – רפלקציה

* במהלך העבודה היו לי אתגרים שרובם הם באגים בפרויקט, אבל גם היה לי במשך זמן העבודה על הפרויקט מערכת זמנים די צפופה שהקשתה על עבודה רציפה על הפרויקט. בנוסף גם אתגרתי את עצמי עוד והוספתי טיפול בhttps.
* האירועים המעניינים שקרו לי במהלך הפרויקט זה היה שלב החקירה על https, לא למדנו לעומק על הפרוטוקול הזה והיה לי מאוד מעניין לחקור על זה ולהבין איך זה עובד.
* במהלך הפרויקט היה לי לוח זמנים מאוד צפוף שהיווה קושי למהלך הפרויקט, בשביל להתמודד עם הקושי ניהלתי את מערכת הזמנים שלי נכון וחישבתי כמה זמן להקדיש לכל דבר. בנוסף קושי נוסף זה היה להבין איך https עובד, לא ידעתי איך הפרוטוקול עובד והייתי צריך לחקור ולקרוא לבד איך לעבוד עם זה וזה מה שעשיתי
* הפתרונות שהיו בפרויקט לבעיות היו בדיוק אותם פתרונות שחשבתי עליהם בתכנון. ולכן אין לי המלצות לשיפור
* אני רוצה להודות לכל החברים שלי שעזרו לי מאוד במהלך הפרויקט, הם הפכו את סביבת העבודה לכיפית ונעימה והיה לי ממש כיף לעבוד על הפרויקט בסביבתם.

אני רוצה גם להגיד תודה למורה שלי מרי גבע שלימדה אותי את כל החומר שתרם וגם את החומר שלא תרם לפרויקט ושלא היססה לעזור כשהיה צריך.

פרק ט – ביבליוגרפיה

1. רקע תיאורטי –
2. ספרות מקצועית ספציפית לנושא העבודה:

* HTTPS - <https://www.cloudflare.com/learning/ssl/what-is-https/>
* HTTP Tunneling - <https://en.wikipedia.org/wiki/HTTP_tunnel>
* Proxy - <https://www.fortinet.com/resources/cyberglossary/proxy-server>

1. קישורים לאתרים באינטרנט:

* מציאת כתובת mac של מחשב ע"י קוד בפייתון : <https://stackoverflow.com/questions/159137/getting-mac-address>
* איך ליצור http tunnel - <https://stackoverflow.com/questions/24218058/python-https-proxy-tunnelling>
* איך ליצור חיבור לproxy בwindows - <https://www.ibm.com/docs/tr/odmoc?topic=services-connecting-proxy-server>