

**שם העבודה: anonymizer**

**שם החלופה:** הגנת סייבר ומערכות הפעלה

**שם בית הספר:** ישיבת חורב בנים

**שם התלמיד:** יאיר יצחק דרעי

**ת.ז. התלמיד:** 325928877

**שם המנחה:** אבינעם חצרנוב

**תאריך הגשה:** 21.06.2022

**תוכן עניינים**

**מסמך יזום**

**מבוא ומטרות הפרויקט....................................................................................... 3**

**מצב השוק כיום................................................................................................. 3**

**פעולות עיקריות................................................................................................. 4**

**מסמך אפיון**

**פונקציונליות המערכת לפי נושאים......................................................................5**

**פירוט פונקציונליות המערכת...............................................................................5**

**טכנולוגיות המחייבות חקר ולמידה.......................................................................5**

**תרשים כללי לארכיטקטורת התקשורת בין שרת לקוח.............................................6**

**מסמך עיצוב**

**סביבת פיתוח.............. ....................................................................................7**

**ארכיטקטורת המערכת......................................................................................8**

**פרוטוקול התקשורת בין השרת ללקוח.................................................................9**

**תרחישים עיקריים לפרויקט...............................................................................9**

**הנושאים שנדרשתי ללימוד עצמי וחקר ..............................................................10**

**תהליך החקר...................................................................................................11**

**קטעי קוד עיקריים והוראות הפעלה....................................................................12**

**רפלקציה...................................................................................................... 16**

**ביבליוגרפיה.................................................................................................17**

**נספחים**

**נספח א' – תיעוד שלבי החקר..........................................................................19**

**מסמך ייזום**

**הגדרת נושא הפרויקט**

יצירת שרת proxy אשר נותן שירותי אבטחה כמו Browser Tor ודרכו יהיה ניתן לבצע גלישה באינטרנט מבלי לגלות את הip- האמיתי של המחשב הגולש ובכך תתאפשר גלישה אנונימית ומאובטחת .

**תקציר**

תפותח מערכת המיועדת לכלל האנשים אשר מעוניינים לבצע גלישה "פרטית" באינטרנט , מערכת זו תאפשר למשתמשים גלישה "פרטית" בכך שהתעבורה שתעשה על ידי המחשב תיורט על ידי השרת בכך שכל הזמן הוא יאזין ברקע לתקשורת הנובעת מהמחשב ובמקרה בו המחשב ינסה ליזום תקשורת אותה תקשורת תיורט ותעבור דרך שאת ביניים ובכך תתאפשר גלישה לא ישירה של המחשב באינטרנט (גלישה אנונימית) דבר הנותן פרטיות וביטחון לגולש.

**מטרות הפרויקט**

האינטרנט כיום הוא מקום לא בטוח אשר מאפשר להאקרים למצוא את המקומות בהם אנשים גולשים דרך המחשב שלהם. אני מעוניין ליצור ממשק בטוח דרכו אנשים יוכלו לגלוש ללא חשש שפרטי הגלישה שלהם ייחשפו.

**מוצרים דומים בשוק**

* **I2P- Invisible Internet Project**
* **TAILS**
* **Subgraph OS**
* **Whonix**
* **Tor Browser**

כל אלו הם תוכנות אשר תפקידם לאפשר אנונימיות לגולשים בעת גלישתם ברשת, לעומתם הפרויקט שלי נותן פחות אנונימיות וביטחון מכיון שתוכנות אלו כוללות הצפנות מתקדמות, ומעבר של התעבורה דרך תחנות ביניים רבות.

**סקירת טכנולוגיות הפרוייקט**

הפרוייקט שלי מורכב מטכנולגיות אשר כבר קיימות בשוק (tor היא דוגמא בולטת לכך) אך הפרוייקט הזה יעזור לי להבין יותר לעומק מה קורה מאחורי הקלעים של תוכנות אלו. הפרוקייט שלי הוא שונה מטכנולוגיות אלו בכך שהוא בעצם השלד של אותם תוכנות אך מעבר לשלד באותם תוכנות יש הצפנה (דבר אשר לא עשיתי), וכן מעבר של התקשורת בין תחנות רבות

**קהל היעד של הפרויקט:**

תוכנה זו מיועדת לכל האנשים אשר מעוניינים כי הגלישה שלהם תהיה פרטית, ומאובטחת.

**פעולות עיקריות**

תוכנה זאת עובדת ברקע ותפקידה הוא שבמקרה בו מחשב מנסה לגשת למידע מסוים או שאתר מנסה לשלוח מידע למחשב של הגולש אותו מידע עובר דרך שרתי proxy אותם הקמתי באמצעות aws והמידע עובר הצפנה ובכך הגולש נשאר אנונימי בעת גלישתו.

**קצת יותר על הפרויקט**

במהלך הפרויקט אתכנת בשפת הפייטון דרך **pycharm** ואשתמש בספריות רבות כגון:

scapy, socket, traceback, requests,threading וכו'. אחקור יותר לעומק את העניין של תעבורה ברשת ואבטחה, ומחקר לתוך נושא ה- vpn .

**מסמך אפיון**

**דרישות המערכת**

**פונקציונאליות המערכת**

* התחברות המחשב של המשתמש לשרת הproxy –
* שינוי מסלול התעבורה של מחשב המשתמש
* החזרה של המידע בחזרה למחשב המשתמש

**תכנון וניהול לו"ז זמנים לפיתוח המערכת**

* הקדשת זמן להבנת הפרויקט, המרכיבים שלו ומה שעומד מאחוריהם – שבוע
* הבנה יותר מעמיקה של תכנות רשתות ופיתוח הקוד של הפרוייקט – חודש(חלק זה היה ארוך מכיון שבתקופה זו של השנה היה עומס)
* חיפוש סביבת עבודה מתאימה בשביל הקמת שרתים – שבוע של חקר
* עבודה בסביבת העבודה (aws) הקמת השרתים ויישום הקוד בתוכם – חודש
* הכנת המסמך של הפרוייקט ושיפורים בקוד, ותיעוד של הקוד – שלושה שבועות

בנספחים יש תיעוד מפורט של מהלך העבודה ובעיות שצצו.

**פונקציונליות המערכת - הסברים ופירוט**

* **התחברות המחשב של המשתמש לשרת ה – proxy** כאשר משתמש רוצה להתחבר למערכת הוא צריך להתחבר לשרת אשר דרכו תעבור תעבורת המחשב דבר אשר יאפשר גלישה המאובטחת.
* **שינוי מסלול התעבורה של מחשב המשתמש –** השרת proxy ישנה את תעבורת המחשב מ- תעבורה ישירה מהמחשב ליעד למסלול אשר נמצאות בו כמה תחנות אשר יוכלו לאפשר ביטחון כי לא יהיה ניתן לדעת את כתובת ה- ip האמיתי של המחשב הגולש.
* **הצפנת המידע בעת המעבר בין התחנות –** בעת העברת המידע של המחשב בין התחנות השרת יצפין את המידע בכדי לאפשר יותר ביטחון כי המידע לא יימצא על ידי הצפנה סימטרית/ אסימטרית.

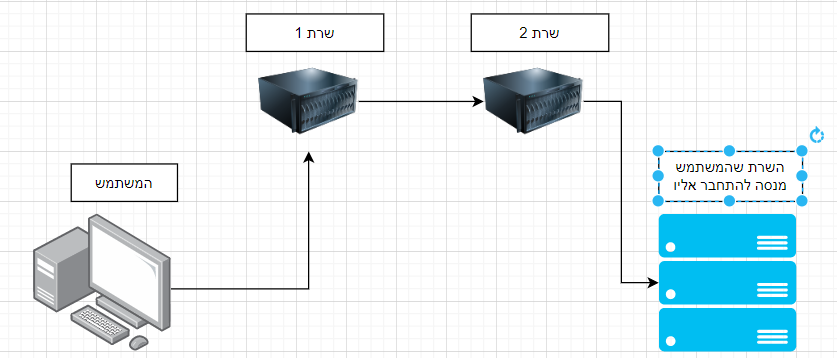
**טכנולוגיות המחייבות חקר ולמידה:**

* למידה על סביבת העבודה של aws
* למידה על תעבורה וכן על proxy
* למידה על פתיחת שרת והתעסקות איתו
* למידה מעמיקה על תכנות רשתות

**רשימת נושאים בעייתיים בפרויקט:**

* שינוי התעבורה – אצטרך ללמוד בצורה מעמיקה על התעבורה של המידע ועל איך לשנות אותה ולמצוא איך להעביר את המידע של המחשב דרך תחנות מסוימות.
* למידה יותר מעמיקה על הרשת (שרתי proxy, הצפנת מידע) - מעבר למה שנלמד בכיתה.
* למידה מעמיקה על סביבת עבודה חדשה (aws) – השימושים בה, מציאת הדברים הרלוונטים לפרוייקט שלי ושימוש בהם, וכו'.

**תרשים כללי לארכיטקטורת התקשורת בין שרת לקוח:**

****

**תיאור תחום הידע**

במהלך הפרוייקט הייתי צריל לפתח ולחקור דבר הבא לידי ביטוי ביכולות שהוצגו במהלך הפרוייקט, בפרק זה אני אפרט יותר על היכולות הללו.

יכולות שהוצגו:

1. שם היכולת: הקמת שרת ב-aws

מהות היכולת: זהו חלק מרכזי מהפרויקט . דרך השרת תעבור התקשורת מהמשתמש לשרת המבוקש

אוסף יכולות נדרשות:

* לימוד מקדים של aws
* קביעת פורט
* הגדרת השרת (לקבל בקשות, להיות פומבי, וכו')
* בדיקת החיבור בינו לבין המשתמש
* קליטת נתונים
* קבלת תשובה מהשרת
* העברת הנתונים למשתמש
* סוקט האזנה

אובייקטים נחוצים: תקשורת.

1. שם היכולת: חיבור המשתמש לשרת

מהות היכולת: המשתמש צריך להתחבר לשרת ולהגדיר את עצמו בצורה שהתקשורת שלו תעבור דרך השרת

אוסף יכולות נדרשות:

* קביעת פורט התואם את פורט השרת
* בדיקת החיבור בינו לבין השרת (הסוקט עובד)
* קבלת תשובה מהשרת aws

אובייקטים נחוצים: תקשורת.

**ארכיטקטורת המערכת**

ארכיטקטורת המערכת של התוכנית מורכבת מתהליכים ותרחישים.

**תהליכים**

**תהליך של הקמת הלקוח-** זהו המשתמש של המערכת. הלקוח מתחבר לסרבר (הוא צריך לקבוע את הפורט התואם לפורט הסרבר - 3128) ואז מתחבר לממשק המשתמש שלו שבמקרה הזה הוא דפדפן מכל סוג שהוא ואז גולש לאתרים (במקרה של הפרוייקט שלי יהיה ניתן לגלוש רק לאתרי http ) שהתקשורת איתם מיורטת למחשבי צד ג' .

**תהליך של הקמת הסרבר –** חלק זה יתממש על ידי הפעלת המחשבים אשר ההקמתי בaws והתחברות אליהם דרך הגדרות הפרוקסי של המחשב של המשתמש. לאחר מכן המשתמש יוכל לגלוש בזמן שהסרבר יעבוד ברקע – יירט את התקשורת ויעזור למשתמש לגלוש בצורה אנונימית באינטרנט. הסרבר הוא רב משתמשים, כך שכמה אנשים יוכלו לגלוש דרכו במקביל. נשתמש גם בפקטות של הפייטון (כגון הטרדים וtraceback) שיעזרו לנו בעת כתיבת הקוד ויאפשרו לנו לתפעל דברים בצורה יעילה.

**סביבת פיתוח**

פרויקט זה נכתב בסביבת עבודה לפיתוח של PyCharm .

שפת התכנות בה השתמשתי היא Python . וכן השתמשתי בסביבה של aws.

**סקירת חולשות ואיומים**

סקירת האיומים שיתכנו למערכת:

* בפרויקט יש סכנה כי צד ג' יאזין לתעבורה של המחשב של המשתמש, ומכיון שאין הצפנה הוא יוכל לראות אותה.
* ישנה סכנה של מתקפת ddos בה מחשבים רבים ישלחו בקשות לשרת שהקמתי ואז הוא יקרוס.
* ניתן לגלוש רק לשרתי http שאינם מאובטחים לכן יהיה ניתן דרך השרת Http להגיע לתקשורת של המחשב – אך מכיון שהתעבורה עוברת דרך שרת צד ג' (שניים) יהיה מאוד קשה להגיע למקור של התעבורה.

**פרוטוקולי התקשורת בין הclient ל-server**

|  |  |  |
| --- | --- | --- |
| **client server web** | | |
| המשתמש מתחבר לשרת |  |  |
| המשתמש מנסה לגלוש לאתר מסוים | השרת מיירט את התקשורת היוצאת מהמחשב  ↓  השרת שולח את אותה תקשורת לשרת המבוקש ברשת | השרת מקבל את התקשורת מה ה-server |
|  | ה-server מקבל את התקשורת ושולח אותה למשתמש | השרת ברשת שולח תקשורת בחזרה ל-server |
| המשתמש קיבל את המידע המבוקש (ובתהליך המשתמש תקשר עם השרת אך לא ישירות ובכך גלש אנונימית) |  |  |
| במקביל לתוכנית: | הסרבר יאזין תמיד לתקשורת של המחשבים ויתן מענה לכל משתמש בנפרד |  |

**תרחישים עיקריים בפרויקט**

* **התחברות לserver –** התחברות לסרבר - זהו התהליך הראשוני של המערכת, בו יתחבר המשתמש לסרבר וכך המערכת תוכל להתחיל לפעול.
* **האזנה של ה-server –** לאחר שהשרת פועל הוא מתחיל להאזין לתקשורת שהמחשב של המשתמש מנסה לעשות עם האינטרנט ומיירט את אותה תקשורת
* **התקשורת של השרת עם האינטרנט ועם המשתמש –** לאחר שהשרת מזהה ניסיון לתקשר עם הרשת (world wide web) הוא מעביר את אותה תקשורת לשרת המבוקש לדוגמא אם המשתמש מנסה לגלוש לפייסבוק אז אותה בקשה תועבר לאחד השרתים של פייסבוק דרך השרת שהתחברנו אליו (שרת ביניים) וכאשר השרת יקבל תשובה בחזרה מהשרת של פייסבוק שקיבל את השאילתא הוא יעביר אאת המידע חזרה אל המשתמש.

**הנושאים שנדרשתי ללימוד עצמי וחקר**

**בפרויקט זה נדרשתי ללימוד עצמי וחקר במספר נושאים:**

ראשית, הייתי צריכה ללמוד כיצד לעבוד עם ספריות לא מוכרות כמו ספריית \_thread, לבנות שרת רב משתמשים , לימוד תיאורטי רב על הנושא של proxy ומתוך כך היישום של החומר התיאורטי דרך תכנות רשתות, וכן במהלך הפרויקט הייתי צריך להשתמש בסביבת עבודה חדשה אשר צצו לי איתה בעיות רבות אשר הייתי צריך לשבת זמן רב בכדי לפתור ואף לקבל עזרה מהמורה ומאנשים אשר מתמצאים בנושא . שנית, הייתי צריכה ללמוד כיצד למצוא באינטנט, באתרים שונים, מענה לבעיות ושאלות שצצו במהלך הפרויקט ולדעת כיצד להשתמש בהם בפרויקט בצורה הטובה ביותר. נעזרתי בהרבה שאלות שנשאלו על ידי אנשים אחרים בעולם שהייתה להם את אותה בעיה כמוני, וקראתי את פתרונותיהם ויישמתי אותם בפרויקט שלי. שילבתי את הידע שלמדתי בחקר כאשר ממשתי אותם בשלבי הפרויקט להלן.

הינה מספר נקודות אותם חקרתי בפרויקט זה:

1 .בניית הסרבר – בכדי לבנות את הסרבר הייתי צריך ללמוד מאפס סביבה חדשה (aws) דבר אשר הציב בפניי אתגרים רבים ומסובכים בינהם היה להבין איך להשתמש בסביבה, להבין מה אני צריך לממש דרך אותה סביבה בכדי שהפרויקט שלי (קרה שעקבתי אחרי כל מני חוטי מחשבה ונעזרתי בסרטונים מיוטיוב בכדי לממש אותם אך הם הוכיחו את עצמם כלא יעילים דבר אשר גרם ל"בזבוז" זמן רב), ולדעת להתמודד עם בעיות רבות אשר צצו בדרך .

2. הייתי צריך להעמיק במחקר תיאורטי על הנושא של proxy ולאחר מכן להביא את המחקר לידי מימוש דרך תכנות רשתות

3. הייתי צריך להבין איך ליצור חיבור סוקט מהשרת אשר הקמתי למחשב האישי שלי

4 .יצירת טרדים ליעול הקוד, ועבודה באותו מקביל על דברים שונים.

5. להצליח בעזרת תכנות רשתות לנתח ולמשש את הרצון של המשתמש לגלוש לאתרים מסוימים (לדוגמא, שאם המשתמש רוצה לגלוש לפייסבוק אז הסרבר מקבל את כל הurl וצריך לפרק את זה לחתיכות של פורט, domain, וכו')

**תהליך החקר**

כפי שהסברתי בעמוד הקודם, לא מעט דברים הייתי צריך ללמוד בשביל הפרוייקט. את הפרוייקט שלי התחלתי בחקירה על הנושא של proxy נעזרתי בסרטונים מיוטיוב ומחומרים שלמדנו שנה שעברה, לאחר שהרגשתי בטוח ברקע התיאורטי התחלתי לעבור לעניין של היישום בעזרת תכנות רשתות, בשלב זה נתקלתי בהרבה קשיים (כמו באיזה ספריות להתשמש, איך להתשתמש בהם, איך לייעל את התוכנית, איך לכתוב בצורה ברורה ומסודרת, וכו'), דבר שמאוד עזר לי להתמודד איתם היה אתרי שאלות באינטרנט כמו stackoverflow"" ו-GeeksForGeeks""' לאחר שראיתי כי הפרוקסי עובד כאשר הקמתי אותו על המחשב שלי עברתי לשלב הבא שהוא הקמת שרת (מחשב צד ג' שדרכו תעבור התקשורת(. שלב זה הווה את האתגר הגדול ביותר בפרויקט, וכן הוא השלב שהשקעתי בו את מרבית הזמן. בהתחלה אני חשבתי להשתמש בשרתים הזמינים לכל אחד באינטרנט אך לאחר התייעצות עם המורה שלי החלטתי להשתמש בסביבה של aws בכדי להקים את השרת מכיון שדבר זה יהיה יותר מאתגר ויהווה היכרות לסביבת עבודה אשר די נפוצה בימינו. לאחר מכן התחלתי במחקר לתוך הסביבת עבודה של aws וכפי שידוע זוהי סביבה גדולה עם מלא אפשריות וכיוונים דבר שבעקבות כך שעשיתי את המחקר לבד עליה הביאו אותי להרבה מאוד מבואות סתומים דבר אשר הוביל לבזבוז של הרבה מאמץ וכן לתסכול רב, לאחר בסביבות כשבועיים של מחקר וניסיונות הצלחתי להקים את השרת אך למרות שהצלחתי להקים את השרת עדיין היו איתו כל מני בעיות שהייתי צריך לפתור ובשלב זה כבר נעזרתי קצת בסביבה (חברים, משפחה) לאחר שבוע הצלחתי כבר לפתור את הבעיות והשרת עובד!

**מימוש הפרויקט**

**סקירת המודולים**

**הכלים וספריות עיקריים בהם השתמשתי הם (המודולים שייבאתי)**

1 .socket – Socket הוא מודול לממשק עם תכנות רשת ברמת-בסיס. הוא משמש

לבניית תקשורת בין מחשבים שונים.

2 .\_thread -שבעזרתו נוכל להשתמש בתהליכונים ולהפעיל דברים במקביל.

3 . sys – בכדי לסגור את התוכנית במקרה מסוים

4 . traceback – ספריה זו שימשה אותי במקרים יוצאי דופן של קלט המתקבל על ידי התוכנית.

**מודולים שפיתחתי**

1.שם המחלקה: main

תפקיד המחלקה: האזנה למשתמשים (באצעות סוקט ההאזנה) וטיפול בבקשות

תכונות המחלקה:

1. my\_socket- יצירת סוקט tcp
2. LISTEN\_PORT – מספר הפורט שהשרת יאזין עליו
3. BUFFER\_SIZE – גודל החבילות של המידע שנקבל
4. MAX\_CONNECTIONS – כמות החיבורים המקסימלית של השרת

פעולות במחלקה:

* ניתן להפסיק את התוכנית בעזרת לחיצה על ctrl+c

2. שם המחלקה: conn\_string

תפקיד המחלקה: הפונקציה זו אחראית על מעבר לסרבר המבוקש על ידי המשתמש.

תכונות המחלקה:

1. First\_line – לקיחת המידע והפענוח שלו
2. url
3. http\_pos – מציאת המיקום של ://
4. web\_server\_pos – מציאת שם הדומיין – בכדי לדעת לאן לגשת
5. port\_pos – מציאת המיקום של הפורט – בכדי לדעת באיזה פורט לגשת לשרת
6. Port – הפורט איתו ניגשים לשרת (הדיפולט הוא 80 – http)

פעולות במחלקה:

* conn,addr אלו הם הנתונים של המשתמש, data זה הפרטים של הגלישה שהמשתמש מנסה לבצע

3. שם המחלקה: proxy\_server

תפקיד המחלקה: קבלה ושליחת מידע על ידי המשתמש ויצירת סוקט מהמשתמש לשרת.

תכונות המחלקה:

1. user\_socket – סוקט מהצד של המשתמש לשרת בכדי לקבל ולשלוח מידע
2. reply – מידע שהמשתמש מקבל מהשרת

פעולות במחלקה:

פרמטרים שהפונקציה מקבלת:

* Web\_server
* Conn
* Port
* Data
* addr

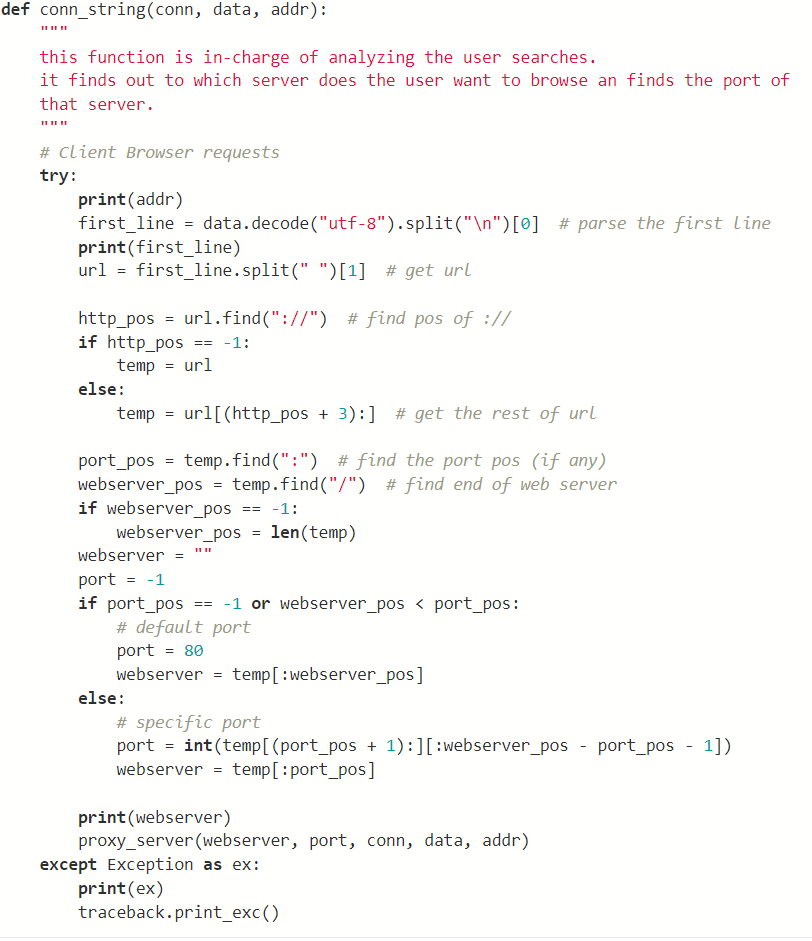
אלו הם משתנים שהסברתי את משמעותם בפונקציות הקודמות.

במקרה של תקלה אז הפונקציה תפסיק לעבוד והתוכנית תסגר.

**קטעי קוד עיקריים**

**להלן קטעי קוד עיקריים בפרוייקט:**

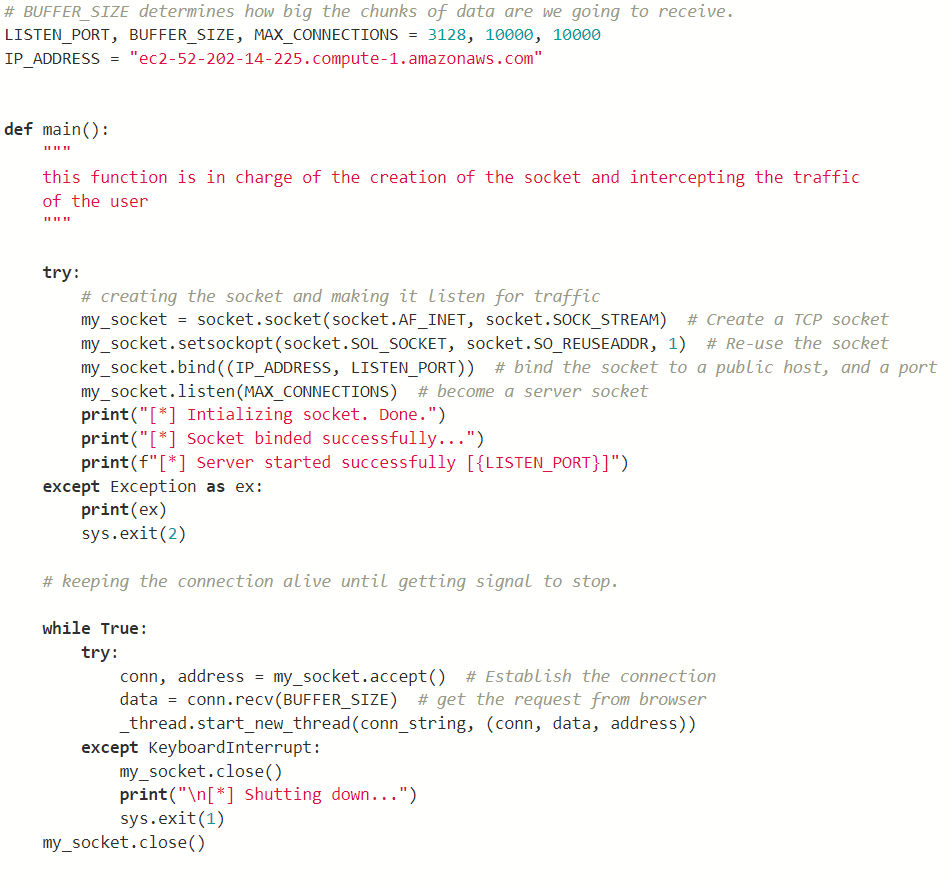
1. **קטע קוד זה אחראי על פירוק של הurl בכדי שהשרת יבין לאיזה domain לגלוש באיזה פרוטוקול ובאיזה פורט (אחראי על כך שהמשתמש יקבל מזרת את המידע שהוא רצה)**

****

ראשית פונקציה זו כאשר היא נקראת היא מקבלת מספר ערכים בינהם הIp שממנו יצא הבקשה לגלישה לאתר מסוים וכן את המידע (האתר אליו המשתמש מנסה לגלוש – URL) .

התוכנית מתחילה מההדפסה של המקור של המידע (האם השרת שולח את המידע או המשתמש) לאחר מכן לוקחים את המידע שהוא כרע בbytes והופכים אותו לstring לאחר מכן מסדרים את אותו string ומתוכו בוררים את שם האתר (google לדוגמא) את מספר הפורט וכן את הפרוטוקול איתו נגשים לאתר (http לדוגמא) לאחר מכן בעזרת כל הנותנים שבררנו אנחנו קוראים לפונקציית "proxy\_server" בכדי שמהסרבר שלנו אנחנו ניגש לאתר שהמשתמש מנסה לגשת אליו.

1. **קטע קוד זה אחראי על יצירת סוקט ההאזנה.**

****

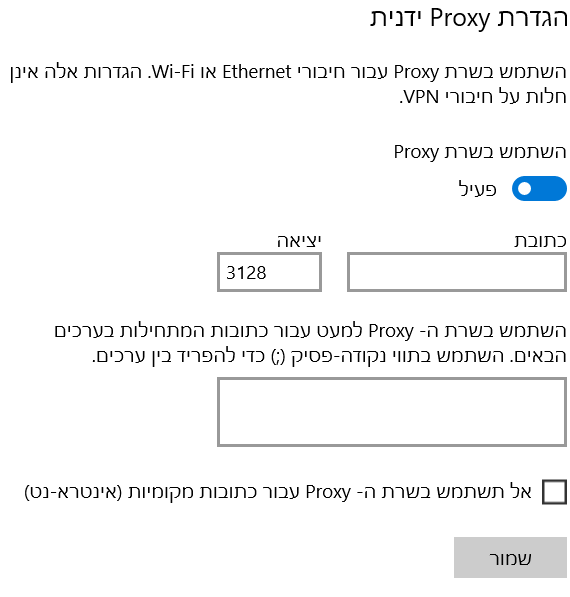
בעזרת socket ההאזנה שיצרנו יהיה ניתן להקשיב לתקשורת של המחשב של המשתמש בתוכו יש משתנים כמו הפורט של ההאזנה מה גודל המידע שניתן לקבל מהמשתמש וכן את מספר החיבורים המקסימלי שניתן לעשות. יש פה גם שימוש ב-multi threding בכך שכל פעם שמקבלים בקשה מהמשתמש פותחים thread חדש לאותה בקשה וכך ניתן להתייחס לכל הבקשות במקביל דבר הגורם ליעילות ומהירות מצד הproxy שלנו. בקוד הוספתי גם סעיף בו במקרה של תקלה מסוימת אז היא תודפס על ידי הproxy לסך של ה-cmd ולאחר מכן תתבצע יציאה מהמערכת. וכן תינתן יציאה מהמערכת על ידי המשתמש אם הוא ירצה (אם הוא ילחץ על ctrl + c).

**ממשק משתמש והוראות הפעלה**

תפעול המערכת מתחיל **בצד הלקוח**:

פרויקט זה בנוי בצורה המיועדת למערכת ההפעלה של windows!.

בכדי להפעיל את המערכת ראשית יש ללכת להגדרות פרוקסי (להכנס להגדרות של המחשב ובשורת החיפוש לכתוב הגדרות proxy) ולהזין שם את כתובת ה-ip של השרת (הכתובת לא קבועה ולכן צריך על פי זה להתאים ולשנות את הכתובת). הפורט הוא קבוע לכן ניתן חד פעמית להזין את מספר הפורט ולהשאיר אותו קבוע – מספר הפורט הוא 3128.



בתמונה זו ניתן לראות בבירור את מה שהזכרנו – בשדה של הכתובת כותבים את הIp של הproxy בשה היציאה (הפורט) כותבים את המספר 3128

חשוב לזכור ללחוץ אחרי זה על שמור בכדי להוציא את השנויים לפועל

לאחר מכן יש ללכת **לשרת הproxy** ולהיכנס לשורת פקודה של anaconda הנמצאת במסך הראשי (צריך להכנסת ספציפית לanaconda מכיון שבשורת פקודה בproxy אין פייתון)ם יש ללכת לכתוב את השורת קוד הבאה:

Cd Documents

(אפשר במקום לעשות זאת פשוט להריץ ישר את הפקודה python ואז המיקום המדויק של הקובץ בתוך גרשיים)

יש לכתוב שורה זו מכיון שבתקיה זו נמצא הקובץ הרצה של פייתון. לאחר מכן יש לכתוב את הדבר הבא:

Python proxy2.py

לאחר מכן הקובץ אמור לתת אינדיקציה שהוא מתחיל.

במידה והוא מתחיל בכדי לראות אם באמת אנחנו גולשים בצורה אנונימית יש לגלוש לשרת http (בפרוייקט שלי אני עשיתי פרוקסי לhttp והתעסקתי רק מבחינה תיאורטית בhttps אך בסוף לא שילבתי בפרוקסי את שינהם) – אתר שאני השתמשתי בו הוא:

http://www.rashim.co.il/

לאחר שנכנסנו לאתר יש ללכת בחזרה לproxy ולראות שבcmd כתוב שהייתה תקשורת בינו לבין השרת של ראשים במידה שתקשורת זו קיימת משמע שזה עבד וגלשנו בצורה אנונימית לראשים.

**רפלקציה**

הפרויקט בסייבר היה בשבילי אתגר קשה וחוויתי אשר למדתי ממנו הרבה מאוד הן על כל הנושא הקשור לסייבר והן כלים לחיים הקשורים למחקר וכן עצם העלאת רעיון שאתה חושב עליו שזה בפני עצמו לא פשוט ולאחר מכן לחקור על הנושא ולהקים אותו מאפס. זו מיומנות אשר חשובה מאוד לחיים – לקחת רעיון אשר יש לך בראש ופשוט לעוף עליו ולהתחיל לחקור והפיג ברעיון חיים דבר אשר אנו רואים שנהיה תדיר יותר ויותר בחיינו בעולם המתפתח של ההייטק. לכן אני שמח להגיד שזה היה אחד החוויות הלימודיות היותר מוצלחות שחוויתי מכיון שעצם זה שזורקים אותך למים ונותנים לך לנסות לשחות ולהתמודד לבד זה משנה את הלימוד ומתמיה יותר את הנלמד בך.

במהלך הפרויקט נתקלתי בקשיים רבים אשר לחלקם אף לא מצאתי פתרון במהלך המחקר והפקת התוצר הסופי לדוגמא כל מני בעיות שנתקלתי בהם במהלך הקמת הproxy דרך aws אך היו כל מני בעיות שמצאתי את עצמי לאחר מחקר מעמיק מצליח לפתור דבר שהיה מאוד מספק עבורי.

אני חושב שהפרויקט היה מאוד חוותי עבורי עם זאת למדתי ממנו כמה דברים שאם הייתי עכשיו מוצא את עצמי מתחיל פרויקט חדש אדע לשנות ולהשים עליהם יותר דגש אחד הדברים המרכזיים מבניהם הוא העובדה שמצאתי את עצמי כל הזמן מתפזר לכל מני כיוונים דבר אשר לקח ממנו הרבה זמן ומאמץ אשר לא היו קשורים לפרויקט שלי לכן כיום אני רוצה לעשות מחקר יותר מעמיק לתוך נושא מסוים לפני שאני זורק את עצמי לתוך המים דבר שכפי שחוויתי הוכיח את עצמו כלא יעיל.

אני רואה את הפרויקט שלי כמעין צוצר בסיס אשר אפשר לשפר ולשדרג אותו לכן בעזרת ה' בעתיד אני חושב להרחיב ולשדרג את הפרויקט בנקודות הבאות:

1. להוסיף הצפנות מתקדמות לתעבורה
2. לעשות רשת של שרתי proxy מהם כל הזמן באופן רנדומלי ייבחר מספר ועל פי אותו מספר ייבחרו השרתים בצורה רנדומלית.
3. להוסיף פיצ'רים כמו לדוגמא לעשות שהמשתמש עצמו יוכל לבחור דרך איזה שרתי proxy תעבור התעבורה שלו.

**ביבליוגרפיה – רשימת מקורות מידע בהם נעזרתי**

**מידע על aws וכיוצא מזה ec2:**

1. <https://www.youtube.com/watch?v=kvHwh1o3kCY&ab_channel=CKAlgos>
2. <https://www.youtube.com/watch?v=FCEHqB3wTcQ&t=276s&ab_channel=Programacion>
3. <https://www.youtube.com/watch?v=9Jndrp2ieBU&ab_channel=ramitgirdhar>
4. <https://www.youtube.com/watch?v=qUHQuZjTOFA&ab_channel=DataScienceGarage>

**מידע על הנושא של פרוקסי:**

1. <https://www.youtube.com/watch?v=5cPIukqXe5w&ab_channel=PowerCertAnimatedVideos>
2. <https://docs.google.com/presentation/d/1ar8pQDVHjVMBy8elqVXdPENuTykd0Nqs/edit#slide=id.p1>

**מידע על הנושא של תכנות רשתות**

1. <https://docs.google.com/presentation/d/1iVYfRzA8gDWE2Q7h6Yq8g693YkpWRcWz/edit#slide=id.p1>
2. <https://docs.google.com/presentation/d/1XQTt9c7DWldmW0C84pOmUvBH7jeVfpa8/edit#slide=id.p1>
3. <https://docs.python.org/3/library/socket.html>

**בעיות שנעזרתי ב"stack overflow":**

1. <https://stackoverflow.com/questions/16703936/proxy-connection-with-python>
2. <https://stackoverflow.com/questions/23857942/winerror-10049-the-requested-address-is-not-valid-in-its-context>
3. <https://stackoverflow.com/questions/62436205/connecting-aws-ec2-instance-using-python-socket>

**איך לכתוב קוד נקי ומסודר:**

1. <https://dev.to/alexomeyer/10-must-know-patterns-for-writing-clean-code-with-python-56bf>
2. <https://www.geeksforgeeks.org/pylint-module-in-python/>

**נספחים**

**נספח – תיעוד שלבי החקר:**

|  |  |
| --- | --- |
| תאריך |  |
| תיאור הבעיה למחקר | איך tor עובד?איך פרוקסי עובד? |
| מקור מידע רלוונטי | השתמשתי במצגת, בסרטוני youtube ובאנשים בכדי להעמיק לנושא הזה  מקור מידע מרכזי הוא מקור מספר 5 בביליוגרפיה |
| מסקנות שלב המחקר | מידע על הרקע של הפרויקט והבנה של בערך מה צריך להראות התוצר הסופי |
| היישום בפרויקט | כרגע אין יישום, רק למידה. |

|  |  |
| --- | --- |
| תאריך |  |
| תיאור הבעיה למחקר | איך ליישם פרוקסי בתכנות רשתות? |
| מקור מידע רלוונטי | מקורות המידע המרכזיים בהם השתמשתי הם 7-9 בביליוגרפיה |
| מסקנות שלב המחקר | בעזרת המחקר שעשיתי ובעזרת הלמידה שלי של תכנות רשתות התחלתי לתכנת את הhttp פרוקסי שישמש אותי בפרויקט |
| היישום בפרויקט | זהו חלק מרכזי מן הקוד בפרויקט שלי |

|  |  |
| --- | --- |
| תאריך |  |
| תיאור הבעיה למחקר | למה משמש aws ? איך להקים בaws מחשבים אשר הפקטות של המידע שלי יעברו דרכם? |
| מקור מידע רלוונטי | בחלק זה השתמשתי במקורות מידע 1-4 אשר נתנו לי רק רקע אך לאחר מכן נאלצתי להתמודד עם הרבה בעיות לבד |
| מסקנות שלב המחקר | למידה על ממשק חדש זה וניסוי וטעיה לגבי ההשתמשות בו |
| היישום בפרויקט | בסופו של דבר לאחר הרבה ניסיונות כושלים הצלחתי להקים את השרתים אשר יהיו הפרוקסי שלי |

|  |  |
| --- | --- |
| תאריך |  |
| תיאור הבעיה למחקר | איך לכתוב קוד בצורה נקייה ומסודרת? |
| מקור מידע רלוונטי | חלק זה נלמד דרך שיעור בכיתה שהמורה עשה בשביל הפרוקיט וכן נעזרתי במקורות המידע: 13-14 |
| מסקנות שלב המחקר | זה נתן לי כל מיני אנקדוטות לשיפור הקוד שלי |
| היישום בפרויקט | שיפרתי את הקוד וסידרתי אותו |

**תיעוד הקוד**

"""

the purpose of the following code is to help a third side party be a middle man

between my computer and the internet

"""

import socket

import sys

import \_thread

import traceback

# BUFFER\_SIZE determines how big the chunks of data are we going to receive.

LISTEN\_PORT, BUFFER\_SIZE, MAX\_CONNECTIONS = 3128, 10000, 10000

IP\_ADDRESS = "ec2-52-202-14-225.compute-1.amazonaws.com"

def main():

"""

this function is in charge of the creation of the socket and intercepting the traffic

of the user

"""

try:

# creating the socket and making it listen for traffic

my\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) # Create a TCP socket

my\_socket.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1) # Re-use the socket

my\_socket.bind((IP\_ADDRESS, LISTEN\_PORT)) # bind the socket to a public host, and a port

my\_socket.listen(MAX\_CONNECTIONS) # become a server socket

print("[\*] Intializing socket. Done.")

print("[\*] Socket binded successfully...")

print(f"[\*] Server started successfully [{LISTEN\_PORT}]")

except Exception as ex:

print(ex)

sys.exit(2)

# keeping the connection alive until getting signal to stop.

while True:

try:

conn, address = my\_socket.accept() # Establish the connection

data = conn.recv(BUFFER\_SIZE) # get the request from browser

\_thread.start\_new\_thread(conn\_string, (conn, data, address))

except KeyboardInterrupt:

my\_socket.close()

print("\n[\*] Shutting down...")

sys.exit(1)

my\_socket.close()

def conn\_string(conn, data, addr):

"""

this function is in-charge of analyzing the user searches.

it finds out to which server does the user want to browse an finds the port of

that server.

"""

# Client Browser requests

try:

print(addr)

first\_line = data.decode("utf-8").split("\n")[0] # parse the first line

print(first\_line)

url = first\_line.split(" ")[1] # get url

http\_pos = url.find("://") # find pos of ://

if http\_pos == -1:

temp = url

else:

temp = url[(http\_pos + 3):] # get the rest of url

port\_pos = temp.find(":") # find the port pos (if any)

webserver\_pos = temp.find("/") # find end of web server

if webserver\_pos == -1:

webserver\_pos = len(temp)

webserver = ""

port = -1

if port\_pos == -1 or webserver\_pos < port\_pos:

# default port

port = 80

webserver = temp[:webserver\_pos]

else:

# specific port

port = int(temp[(port\_pos + 1):][:webserver\_pos - port\_pos - 1])

webserver = temp[:port\_pos]

print(webserver)

proxy\_server(webserver, port, conn, data, addr)

except Exception as ex:

print(ex)

traceback.print\_exc()

def proxy\_server(web\_server: str, port:int, conn:int, data:str, addr:str):

"""

this is the function that is incharge of what is happening in the user side.

creating a socket to the proxy, sending and receiving data

"""

print(f"{web\_server} {port} {conn} {addr}")

try:

user\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

user\_socket.connect((web\_server, port))

user\_socket.send(data)

while True:

# receive data from web server

reply = user\_socket.recv(BUFFER\_SIZE)

if len(reply) > 0:

# send to browser/client

conn.sendall(reply)

print(f"[\*] Request sent: {addr[0]} > {web\_server}")

else:

break

user\_socket.close()

conn.close()

except Exception as ex:

print(ex)

traceback.print\_exc()

user\_socket.close()

conn.close()

sys.exit(1)

if \_\_name\_\_ == "\_\_main\_\_":

try:

main()

except KeyboardInterrupt:

sys.exit(0)