

**עבודת גמר 5 יח"ל**

**נושא העבודה :**  רשת VPN למשתמשים פרטיים

**שם תלמיד :** פימה פנחס מוצמכר

**ת.ז תלמיד :** 325118693

**שם בית ספר ועיר :** קריית החינוך ע"ש עמוס דה-שליט, רחובות

**שם המנחה :** ערן בינט

**מועד הגשה :** 20.4.2020

תוכן עניינים

[1. מבוא 3](#_Toc509432052)

[2. תיאוריה 4](#_Toc509432053)

[3. תוצר סופי 5](#_Toc509432054)

[4. תהליך כתיבת הפרויקט 11](#_Toc509432055)

[5. מרכיבי פתרון 11](#_Toc509432056)

[6. תסריטי בדיקה 13](#_Toc509432057)

[7. רפלקציה 13](#_Toc509432058)

[8. הוראות התקנה ותפעול 14](#_Toc509432059)

[9. ביבליוגרפיה 14](#_Toc509432060)

[10. נספחים 14](#_Toc509432061)

1. מבוא
   1. נושא העבודה

נושא העבודה הוא שירות Enterprise VPN למשתמשים פרטיים. השירות יכלול חיבור לשרת VPN אשר יצפין את התעבורה העוברת דרך המכונה של המשתמש. בנוסף השירות יציע גם שימוש בשרת FTP לשם העברת קבצים ובשימוש באזור מנוטר ברשת (DMZ). השירות יכלול האמתה אשר תאמת משתמשים לפי שם משתמש וסיסמא ותעניק JWT לכל אחד מהם.

המערכת תבצע שימוש נרחב בכלים Sockets המאפשרים ניתוב מידע בין מחשבים שונים על ידי המשתמש כרצונו.

* 1. מטרות מרכזיות

המטרות המרכזיות של העבודה הן:

* הקמת שירות שרת – לקוח אשר יאפשר ניהול תקשורת מאובטחת ואנונימית של הלקוח עם העולם.
* לאפשר ללקוח לשתף קבצים עם משתמשים שונים בצורה מאובטחת.
* דיווח על המתרחש ברשת בעזרת שרת פיקוח
* נגישות רחבה ממקומות מרוחקים
  1. רציונל

המוטיבציה שלי לפיתוח הרעיון הינה ליצור ממשק, פשוט נח ושמיש אשר יאפשר למשתמש לבצע גלישה בטוחה ברחבי הרשת הכלל עולמית. לדעתי, שירות שכזה יכול להבטיח את בטיחותם של כלל המשתמשים ואכן לתת תועלת רבה. באמצעות הפרויקט אני מקווה להבין יותר את כלל נושא אבטחת התקשורת ברמות הגבוהות של ה – OSI ובנוסף להצליח לשפר את יכולותי בשפת Python במגוון האפשרויות השונות שהיא מציעה בתחום תקשורת הנתונים. במהלך שיעורי הסייבר הבנתי עד כמה נושא האנונימיות ואבטחת המידע הוא נושא קריטי בימינו. ולכן, היית שמח להרחיב ולחזק את שליטתי בו ככל יכולתי גם בפרוייקט שלי. בנוסף, ברצוני להעמיק בהבנת התקשורת בין מחשבים, ולהרחיב את הידע והשליטה שלי בדקויות בפרוטוקולים השונים באמצעות הפרויקט. כמוכן ארצה לשפר את הדרך בה אני כותב קוד, ולהתמקצע בשימוש ב-source control.

* 1. קישור לחומר הנלמד

העבודה מתקשרת לחומר הנלמד במספר תחומים שונים.

ראשית, בתחום התקשורת. הפרוייקט מדבר על חומר אשר נלמד בכיתה בתחום התקשורת לרבות מודל השכבות של ה OSI, פרוטוקולי רשת בכלל השכבות הנפוצות. הפרוטוקולים HTTP/S, TCP, FTP מהווים בסיס עז של הפרוייקט.

1. תיאוריה
   1. תיאוריה

VPN היא שיטה להעברת מידע פרטי על גבי תשתית שעיקרה או כולה בבעלות ציבורית או בבעלות פרטית עם גישה לכלל הציבור. מטרתו של ה-VPN היא להעביר מידע מהרשת הארגונית לאדם כלשהו, שנמצא במקום שבו לא קיימת תשתית תקשורת פרטית של הארגון, אבל קיימת תשתית ציבורית - אינטרנט או טלפוניה.

שירותי ה VPN מגנים על המשתמש מהאקרים, תוכנות זדוניות, ושאר דברים חשודים ברשת. בנוסף, הם יכולים לעזור לעקוף צנזורה בכדי להנות מהאינטרנט ללא הגבלות.

על אף ששימוש בשירות VPN עשוי להישמע מסובך, הם בעצם מאוד ידידותיים למשתמש. ממש לא צריך להיות מומחה כדי להשתמש בהם!

ראשית, בכדי להבין את תפעולתו של ה VPN, יש עלינו להבין כיצד מכשירים אלקטרוניים

(מחשבים, פלאפונים וכו') מתקשרים זה עם זה. מחשבים ומכשירים ברשת שמהווים את המקור או היעד לנתונים, או מנתבים את הנתונים נקראים צומתי רשת (network nodes). צומת ברשת יכול להיות מחשב, טלפון, שרת או רכיב חומרה ייעודי לתקשורת מחשבים, כגון נתב או מתג. ניתן לומר שכל שני מכשירים כאלה מחוברים לאותה רשת כאשר מכשיר אחד מסוגל להחליף מידע עם המכשיר השני, בין אם קיים ביניהם חיבור ישיר ובין אם לאו.

מודל 7 השכבות (OSI) הוא מודל המתאר את הפעולות הנצרכות על מנת להעביר מידע על גבי רשת תקשורת.

כאשר יש שיחה בין מחשב לקוח למחשב שרת, הם מעבירים ביניהם מסר – הם מעבירים את המידע שצריך לעבור.

כדי שהמידע הזה יעבור הוא עובר בשלבים (בדיוק כמו שכדי להעביר הודעה בדואר – את המסר עצמו, צריך להכניס את ההודעה למעטפה, לרשום כתובת, בול ועוד – דברים נוספים שנלווים אך לא העיקר).

המידע זה ה- DATA, וכל השאר הם זנבות שמתווספים על מנת שיהיה אפשר לנוע ולהגיע ליעד בבטחה.

כל שכבה אחראית לתחום אחר הקשור למידע העובר ברשת. ניתן לומר שהתהליך הזה מבטא שלבי מעבר של המידע ברשת. כך שהמידע נוצר במחשב כלשהו בשכבה ה-7 ואז יורד במורד השכבות עד שמגיע לשכבה ה- 1 (פיזית), ונוסע כביטים (סיביות) מידע על גבי המדיה (כבלים או גלים).

כאשר המידע מגיע למחשב בצד אחר (או שרת), אז התהליך קורה הפוך, הוא מתחיל מהשכבה ה-1 ועולה במעלה השכבות עד שמגיע לשכבה ה- 7 והמידע נקרא ע"י התוכנה שבמחשב השני.

השכבות העיקריות בהן עוסק הפרוייקט הן:

**שכבת הרשת ((Network - שכבה 3**

מנתבת את ה"חבילות" (Packets) בין הרשתות. תפקיד השכבה למצוא נתיב בין המקור והיעד ולהעביר דרכו את המידע. השכבה מנתבת באמצעות ראוטר (נתב), עפ"י הכתובת הלוגית.

כתובת IP היא כתובת לוגית אשר מזהה את המארח. כדי לתכנן רשת היררכית יש צורך בכתובת לוגית בנוסף לכתובת הפיזית. הכתובת הלוגית קובעת לאיזה רשת מקומית המארח שייך.

כתובת IP היא כתובת ייחודית ברשת.

לכל "חבילה "(Packet) הנשלחת באינטרנט יש כתובת IP למקור וכתובת IP ליעד וזאת, כדי שה- "חבילה" תגיע ליעד וגם נקבל על כך תשובה.

כתובת IP כוללת 32 ביטים של: "1", "0". לדוגמא: 11000000101010000000000100000101.

**שכבת התעבורה (Transport) – שכבה 4**

מנהלת את העברת המידע ברשת מקצה לקצה. השכבה מוודאת שהנתונים שהתקבלו תקינים מבחינת המבנה והתבנית (בקרת איכות). השכבה מפצלת את ההודעה למנות בצד השולח ובונה אותן מחדש בצד המקבל.

הפרוטוקולים הנפוצים לתעבורה הם:

TCP – Transmission Control Protocol

UDP – User Datagram Protocol

שכבת התעבורה מפרק את הודעת HTTP למקטעים (Segments) אשר ישלחו באינטרנט. סגמנטים אלו ממוספרים לפי הסדר ומועברים לפרוטוקול IP כדי להרכיב מהם "חבילות".

פורט: מספר פורט הוא מספר המוכנס לכל סיגמנט הנשלח באמצעות TCP או UDP ומאפשר בקשת שירות מסוים. לכל הודעה נשלחת מוסיפים: פורט מקור ופורט יעד.

באמצעות מספר הפורט יודע השרת אשר מקבל את ההודעה איזה שירות נדרש ממנו לספק.

פורט היעד (Destination Port) מתייחס לשירות המבוקש על ידי המשתמש.

פורט מקור (Source Port), הוא מספר אקראי, המיוצר ע"י המערכת השולחת, כדי לזהות את השיחה בין שתי מערכות. דבר זה מאפשר לבצע שיחות רבות בו-זמנית.

**שכבת האפליקציה (Application) – שכבה 7**

בשכבה זו שהיא השכבה הגבוהה ביותר במודל, קיימים מגוון פרוטוקלים שונים. הפרוטוקול המרכזי שממומש ומשומש בפרוייקט שלי הוא פרוטוקול ה HTTP.

HTTPהינו פרוטוקול שרת-לקוח מבוסס טקסט. במילים פשוטות, HTTP מגדיר שפת טקסט פשוטה המאפשרת לצרכן שירות כלשהו – הלקוח – לבקש משאבים מנותן שירות כלשהו – השרת.

לקוח ה-HTTP הנפוץ ביותר הוא הדפדפן, המבצע בקשות אל מול שרת אינטרנט להורדת דפי אינטרנט, תמונות וכדומה. אך זהו לא השימוש היחיד בפרוטוקול: HTTP משמש גם תוכנות אחרות מבוססות שרת לקוח כגון אפליקציות מובייל, מערכות החושפות REST API, שרתי source control, תקשורת בין רכיבים במערכות מורכבות ועוד אינספור דוגמאות אחרות.

כפי שהזכרתי קודם, פרוטוקול HTTP הינו מבוסס טקסט, כלומר כל המידע העובר בפרוטוקול הינו טקסט קריא בלבד, ואין בו מידע "בינארי". תכונה זאת של HTTP, גם אם מוסיפה מעט לנפח המידע הנשלח, מקלה מאוד על תהליכי הפיתוח, הניתוח ואיתור השגיאות, פשוט כי תמיד ניתן להסתכל בעיניים על המידע ולהבין מה הולך.

במודל 4 השכבות (וגם בזה המכיל 7), HTTP שייך לשכבה הגבוהה ביותר – שכבת היישום (האפליקציה). כלומר כדי להשתמש בפרוטוקול HTTP אנחנו צריכים לדאוג קודם כל לחיבור אמין בין השרת ללקוח, שיאפשר לנו העברת טקסט בין שתי הישויות. אחרי שייצרנו חיבור כזה, שכמעט תמיד יהיה מסוג TCP, פרוטוקול HTTP למעשה מגדיר איך יראה המידע שיעבור בין השרת ללקוח, כלומר איך תיראה בקשה למידע, ואיך תיראה תגובה אליה.

הפרוייקט שלי כולל אלמנט משמעותי של הצפנה ולכן חשוב להכיר גם את פרוטוקל HTTPS.

HTTPS אינו פרוטוקול תקשורת בפני עצמו, אלא יישום של פרוטוקול HTTP. הוא זהה לו מבחינת המבנה, אך מורה לדפדפן להוסיף שכבת הצפנה כדי להגן על תעבורת המידע. היישום של מודל SSL על פני HTTP מתאים במיוחד מפני שניתן לאבטח את המידע אפילו אם רק זהותו של צד אחד בתקשורת מאומתת. בתקשורת HTTP על גבי האינטרנט, זהותו של השרת (אתר) לרוב מאומתת על ידי השוואה בין Public key certificate של האתר לזה השמור בדפדפן.

SSL/TLS הוא פרוטוקול ורסטילי שמטרתו אבטחת שיחת שרת/לקוח בשיטות קריפטוגרפיות חזקות והוא אמור למנוע ציתות, זיוף, או חבלה (שינוי זדוני) של המידע העובר בין השרת והלקוח. מאפשר חיבור אנונימי, אימות שרת (חד-צדדי) או אימות דו-צדדי, תוך שמירה על דיסקרטיות ושלמות המסרים. שלוש נקודות עיקריות שהפרוטוקול אמור לתת להן מענה הן:

פרטיות - המושגת באמצעות הצפנה סימטרית.

אימות - המושג באמצעות תעודת מפתח ציבורי.

אותנטיות - המושגת באמצעות קוד אימות מסרים.

בפרוטוקול תקשורת שתומך במצב SSL כאופציה, על הלקוח ליידע את השרת על רצונו לעבור לתקשורת מאובטחת, דרך אחת היא להשתמש בפורט ייעודי (מקובל להוסיף את האות s) שהוקצה על ידי ICANN כמו פורט 443 של HTTPS או פורטים 989/990 של FTPS.

בתקשורת נתונים, מִנְהוּר, הוא השימוש בפרוטוקול תקשורת המאפשר הכמסה של פרוטוקול אחר בתוכו, באופן בו שדה המטען (payload) של פרוטוקול המעטפת, מכיל למעשה הודעות המקודדות על פי פרוטוקול הליבה. המונחים תיעול ומנהור נובעים מכך שלמעשה נוצרת תעלה או מנהרה מבודדת שבה עוברות הודעות של פרוטוקול הליבה. שימוש בפרוטוקול תיעול מאפשר לדוגמה חצייה של רשתות שאחרת לא היו מאפשרות מעבר של הפרוטוקול לו מבוצעת הכמסה. דוגמת שימוש נפוצה בפרוטוקול תיעול היא הכמסה של פרוטוקול שאיננו תומך בהצפנה באחר שכן, דבר המאפשר את אבטחת ערוץ מעבר ההודעות ללא כל שינוי בהגדרת הפרוטוקול הגלוי.

פרוטוקולי תיעול לרוב אינם מקיימים את מודל השכבות של ה-OSI במובן שפרוטוקול המעטפת לרוב (אך לא תמיד) שייך לשכבה גבוהה או זהה לזו של פרוטוקול הליבה.

בפרוייקט שלי הנני משתמש בפרוטוקל HTTPS, במתודת CONNECT שמאפשרת למעשה יצירה

תקשורת מוצפנת עם שרת באמצע המשמש שרת מנהרה.

* 1. מוצרים קיימים - ExpressVPN

ExpressVPN הוא שירות VPN בתשלום שמציע את שירותיו ברחבי העולם. ExpressVPN מגנים על החיבור של משתמשיו עם **הצפנת AES-256-ביט ברמה צבאית**, אותה רמת הצפנה הנמצאת בשימוש על ידי סוכנויות ממשלתיות רבות, כדי לסבר את האוזן בנוגע לכמה אלגוריתם ההצפנה הזה מאובטח. ExpressVPN מציעים גם מנהור מפוצל. משמעותה של האפשרות הזו היא שאתם יכולים לבחור אתרים שלא יהיו מוצפנים, וישתמשו באינטרנט "הרגיל", במקום להעביר את כל התעבורה שלכם דרך ה-VPN. תוכלו גם להגדיר את המכשיר שלכם כדי שיתמוך במצב של מנהור מפוצל הפוך, ולבחור לשלוח תעבורה מאתרים מסוימים רק דרך ה-VPN. מנהור מפוצל זמין רק עבור התוכנות של ExpressVPN לווינדוס, מק OS, אנדרואיד, ונתבים.

לאחרונה ExpressVPN עלו לרמה אפילו גבוהה יותר עם טכנולוגיית TrustedServer החדשנית. במהותו, המשמעות של TrustedServer היא ש-ExpressVPN מפעילים את השרתים שלהם רק באמצעות RAM. זה מפחית את הסיכון של נקודות תורפה אבטחתיות בשרתי ExpressVPN בהשוואה לרשתות שרתי VPN אחרות.

אמנם האפליקציה שלי לא תהיה כה משוכללת ומתוחכמת, אך היא תהיה חינמית ותאפשר גישה למגוון תכנים נוספים מלבד גלישה בטוחה באינטרנט, כגוון שיתוף קבצים ו – DMZ.

1. תוצר סופי
   1. תיאור הפרויקט

המשתמש פותח את קליינט הVPN במחשבו האישי אשר מריץ חלונות. חובה על שרת הVPN ושרת ההתחברות להיות פעילים לשם פעילות התכנית. הקליינט מתחבר לשירות ומקבל גישה לכל המוזכר לעיל, לרבות התחברות מאובטחת לאינטרנט, התחברות לשרת FTP, התחברות לשרת DMZ. כמו כן, כל פרק זמן מוגדר יצטרך המשתמש לחדש את

המערכת מאפשרת מספר פעולות:

* הצפנה מוחלטת של המידע המועבר בין המשתמש לממשק בעזרת חיבור SSL
* אנונימיות במרחב בעקבות החיבור לשרת ה VPN.
  1. אלגוריתמים עיקריים

אלגוריתמים מרכזיים במערכת:

1. JWT AUTH (ראשי תיבות באנגלית של: JSON Web Token) הוא תקן פתוח מבוסס JSON ליצירת מפתח גישה (Access Token) המשמש לולידציה של "טענות" (לדוגמה שם משתמש, הרשאות, סיסמה). המפתחות חתומים על ידי מפתח ייחודי של השרת, כך שהלקוח והשרת מסוגלים כל אחד בנפרד לוודא שהמפתח לגיטימי. האלגוריתם אחראי על התחברות לקוחות באופן בטוח ומוודא את היותם מקבלי שירות מורשים.
2. "מנהור HTTP" הוא התהליך בו מתקבלות תקשורת באמצעות פרוטוקול HTTP. מנהור HTTP מיועד בעיקר לרתיעה מפני חומת אש. מנהור HTTP מבצע אנקפסולציה של פרוטוקול, על ידי סגירת מנות נתונים של פרוטוקול אחד (SOAP, JRMP וכו ') בתוך מנות HTTP. לאחר מכן נשלחים מנות ה- HTTP דרך חומת האש כתעבורת אינטרנט רגילה. במקרה שלי, המנהור מאפשר לקיים אנונימיות והצפנה של המידע המועבר לשרת ה-VPN.
3. שירות FTP נגיש למשתמש. FTP (ראשי תיבות: File Transfer Protocol) הוא פרוטוקול תקשורת מבוסס TCP להעברת קבצים בין מחשבים. באמצעות פרוטוקול זה, תוכנת לקוח FTP מתקשרת עם תוכנת שרת FTP, לשם לקיחת קובץ מהשרת או הוספת קובץ אליו.
4. שימוש בDMZ לשם הגנה על הנכסים של המערכת. שאלת על DMZ.

ראשי התיבות של DMZ הן: DeMilitarized Zone - כלומר אזור מפורז.

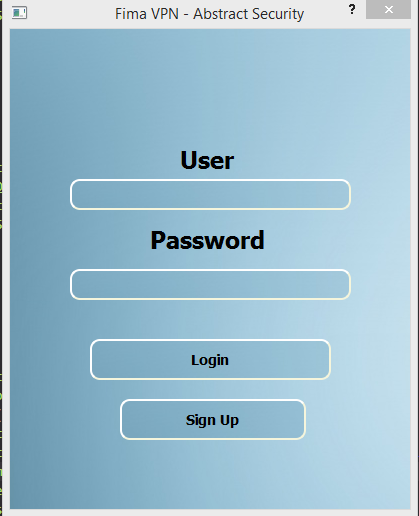
המשמעות היא אזור שהוא מפורז כלומר פתוח. אזור זה משמש כהגנה על רשת ה- LAN של ארגון מסוים, ונמצאת בין ה- LAN לבין היציאה ל- WAN או אינטרנט.

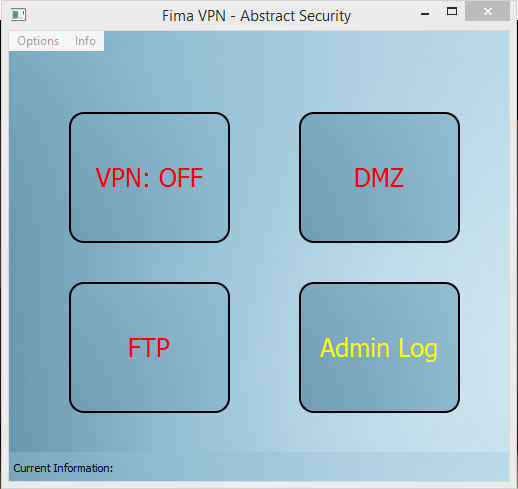
באזור ה- DMZ נמצאים רכיבים ומחשבים שאינם מסווגים ואינם חוששים מפני פריצות וכו'.

1. שינוי של ערכים שונים המתייחסים להגדרות ה PROXY ב Registry. Registry הוא מאגר מרכזי של הגדרות/מערכת רישום, המשמשת את מערכת ההפעלה "חלונות" לגרסאותיה השונות (מחלונות 95 ואילך), לשם רישום מאפיינים של מערכת ההפעלה עצמה ושל תוכנה וחומרה המותקנות בה, העדפות משתמש וכדומה
   1. ממשק משתמש

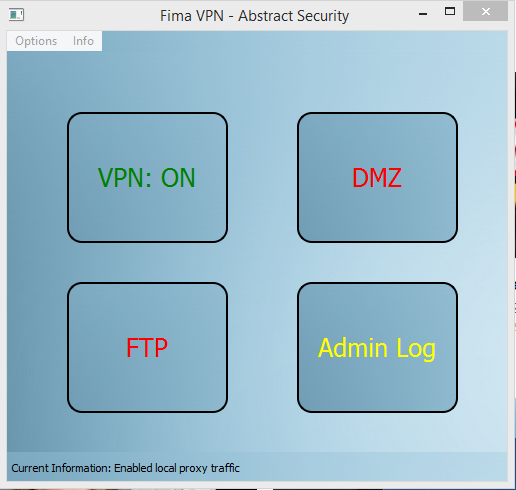
GUI אשר נוצר בעזרת PyQt4

מסך התחברות:

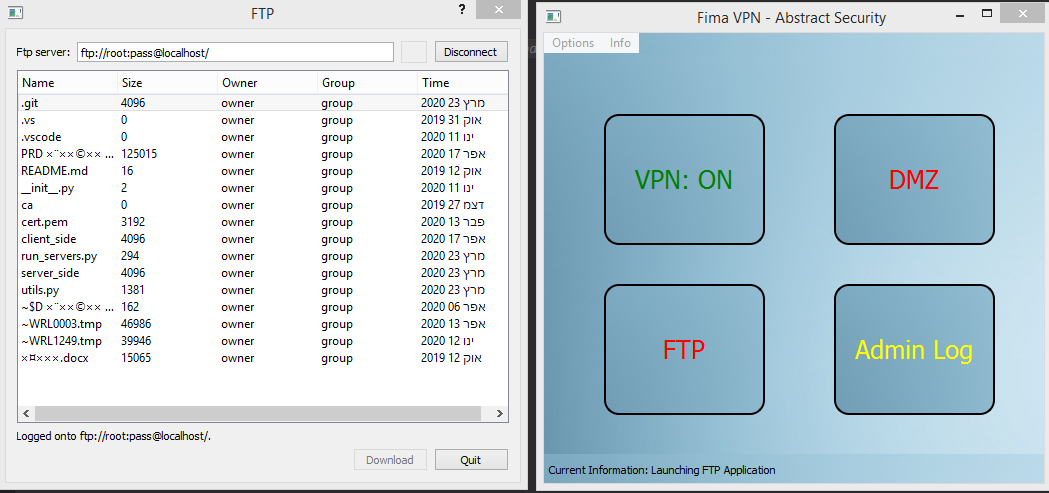


מסך המשתמש לאחר התחברות ראשונית

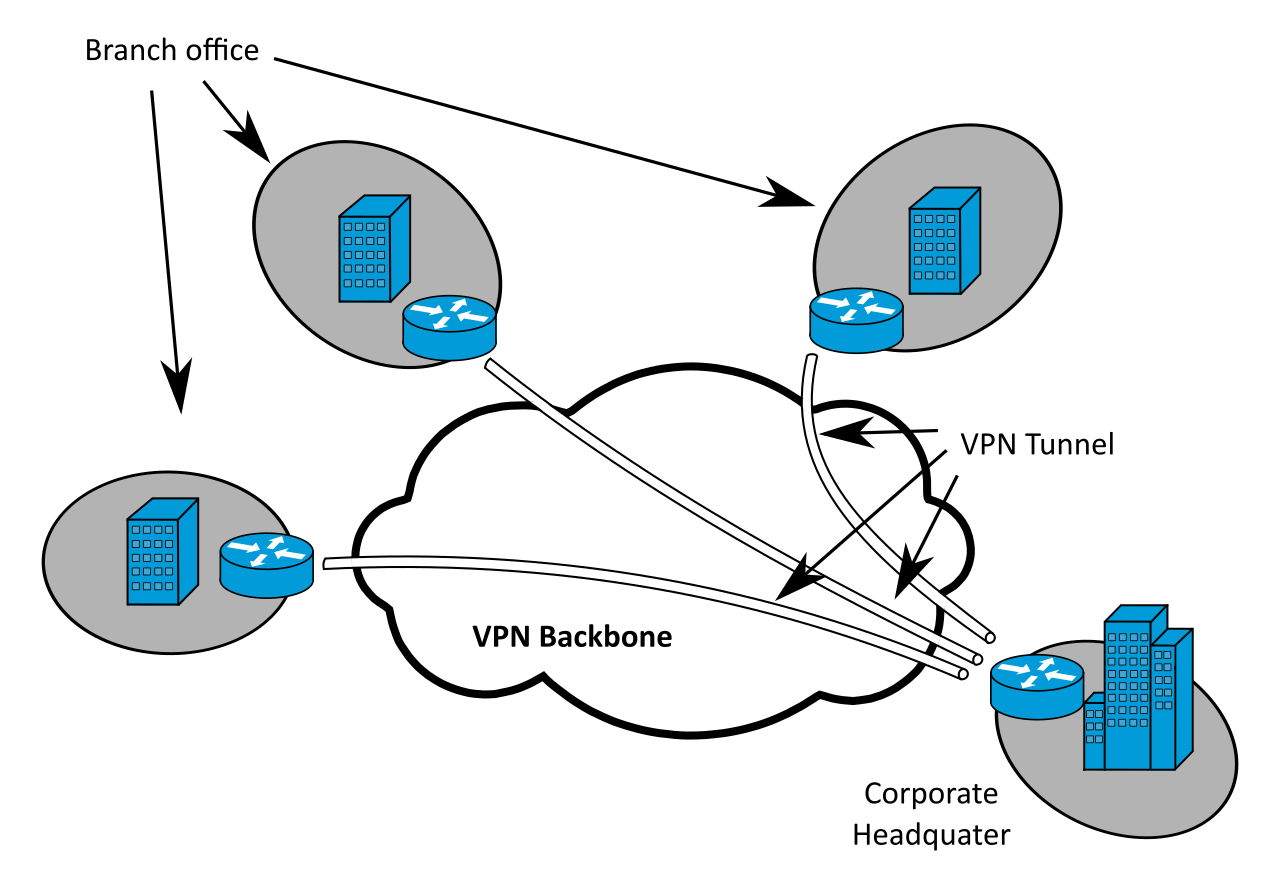
מסך המשתמש לאחר ההתחברות ל VPN



ממשק ה FTP



טיפולוגיית ה VPN באופן מופשט:



* 1. תרחישים עיקריים

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| תרחיש | משתמש | שרת VPN | שרת התחברות | שרת FTP | DMZ |
| הרשמה או  התחברות | המשתמש בוחר להתחבר לשירות. מקבל בתמורה משרת ההתחברות jwt token |  | בודק את בקשת המשתמש ומחלק token זמני בהתאם. |  |  |
| התחברות לגלישה בטוחה דרך  שרת ה VPN | מדליק את כפתור ה VPN ובאופן אוטומטי משנה את ההגדרות המתאימות במכונה שלו. | כעת כל ה Traffic של המשתמש מועבר דרך השרת שיוצר Tunnel בהתאם. בכל הודעה בודק האם הToken של המשתמש בתוקף. אם לא, מפסיק את השירות ומעביר אותו חזרה לקבל token חדש. |  |  |  |
| הורדת תוכן דרך FTP | משתמש מוריד למכונה שלו את הקובץ הרצוי |  |  | מעביר את המידע דרך Firewall הבוחן את המידע | בדיקה של התוכן המועבר |
| הדפסקת Log ע"י משתמש מנהל | הדפסת Log | שולח למשתמש רלוונטי את המידע הרצוי על תעבורת השרת. |  |  |  |

* 1. דרישות ומגבלות המערכת

מכיוון שהאוטומציה של הגדרות הניתוב בעת התחברות למערכת, מתבצעות בעזרת שינויים בטבלת

הRegistry חובה על המכונה להיות מכונה של ווינדוס (7 ומעלה).

בנוסף לכך, דרושה גרסה של Python 2.7.9.

בכדי להתקין את המודולים והתוספים הנדרשים לפעולת הפרוייקט, יש להריץ את הפקודה הבאה בכדי להתקינם: 

מגבלות המערכת הינם חוסר התאמה לפרוטוקלים שונים, אשר יכולים להגביל את יעילות המערכת.

* 1. ממשקים למערכות חיצוניות

אין בפרוייקט ממשקים למערכות חיצוניות

* 1. התייחסות לנושא האבטחה

מיטב מהות הפרוייקט היא לעסק באבטחה ואנונימיזציה של המשתמש ברשת.

* הפרוייקט מאפשר גלישה אנונימית ברשת, שכן כתובת הIP של המתשמש "הופכת להיות זו של שרת ה VPN.
* הפרוייקט מאפשר גלישה מוצפנת שכן הוא משתמש בפרוטוקולי הצפנה שונים.
* הפרוייקט מציג סנריו של אבטחת מידע יוצא משרת FTP.

1. תהליך כתיבת הפרויקט
   1. תהליך הפרויקט

בתחילת כתיבת הפרוייקט, התרכזתי בלמידה על תכונות הvpn ומימושיו השונים בעולם. בתור

POC, כתבתי קוד של שרת המבצע HTTP TUNNEL, שיאפשר יצירת PIPE בין המשתמש לשרת. לאחר מכן, התחלתי לעבד על שרת ההתחברות. השתמשתי ב FLASK בכדי ליצור שרת BACK END שיקבל בקשת התחברות ממשתמשים שונים, יאמת אותם בעזרת DATABASE ויעניק להם

JWT TOKEN לשימוש בשירותים שונים של המערכת.

בכדי ליצור אוטומציה מצד המשתמש, הייתי צריך ליצור קוד שישנה את הגדרות ה PROXY של המשתמש בהתאם לכתובות שהוא מקבל מהשרת. לאחר מכן כל הבקשות מנותבות לעוד שרת לוקאלי שנפתח על מחשבו של המשתמש, המבצע עטיפה של המידע שהמשתמש שולח בעוד שכבת הצפנה נוספת, SSL. שרת לוקאלי זה מתחבר באופן ישיר לשרתי ה VPN.

בהמשך גם הוספתי לשירות אפשרות של חיבור לשרת FTP.

* 1. אתגרים ואופציות שונות למימוש

האתגר המשמעותי בפרוייקט היה לממש מנהרה מוצפנת של המשתמש ליעדיו. לפני המימוש שקיים כרגע, הכוונה הייתה ליצור Interface וירטואלי אמיתי על מכונת המשתמש. אמנם מפאת מגבלות מערכת ההפעלה חלונות וכך שדרך מימוש שכזו תכלול שיטות הצפנה מרחיקות לכת (שכן כל המידע

שיוצא מה NIC של המחשב יעבר הצפנה), המימוש עבר לעטיפה של פטקות HTTP/S בלבד.

אולם אפשרי גם להוסיף פרוטוקלים נוספים בהתאם.

בנוסף לכך, אתגר נוסף שאליו נחשפתי במהלך הפרוייקט היה הגדרה ברורה של פעולות ב DMZ.

באופן עקרוני, DMZ הוא למעשה מקום שבו ממוקמים כל השרתים ה"חיצוניים" של המערכת.

בין ה DMZ לבין האינטרנט ישנה חומת אש וכן גם בין ה DMZ לשרתים הפרטיים של המערכת קיימת חומת אש. הגדרת "חומות האש" היא לא פשוטה. למעשה, היא קובעת את רמת תפקודו של מערך זה. בסופו של דבר החלטתי להכווין את הסינון כנגד תקשורת מהשרת לקליינט של פקטות FTP.

1. מרכיבי פתרון
   1. תיחום הפרויקט

* תקשורת – ביצוע של "מנהור" מידע בין המשתמש לVPN ברמות מעל IP/TCP
* אבטחת מידע - SSL/TLS הוא פרוטוקול ורסטילי שמטרתו אבטחת שיחת שרת/לקוח בשיטות קריפטוגרפיות חזקות והוא אמור למנוע ציתות, זיוף, או חבלה (שינוי זדוני) של המידע העובר בין השרת והלקוח. מאפשר חיבור אנונימי, אימות שרת (חד-צדדי) או אימות דו-צדדי, תוך שמירה על דיסקרטיות ושלמות המסרים.
* תצוגה – תצוגת משתמש מובנת בעזרת PyQt4
* מערכת הפעלה וחקירה low-level - עבודה עם Registry, עם \_winreg בשפת פייתון.
* מבנה נתונים – שימוש במבנה נתונים SQL-י עבור שמירת מידע של שרת ההתחברות
* מבנה קוד – שימוש במחלקות וקבצים מותאמים. הערות ותיעוד.
  1. סביבת העבודה (טכנולוגיה)
* שפות התכנות:

האפליקציה עצמה כתובה בפייתון 2.7.9 – הן צד המשתמש והן צד השרת. שפת פייתון מספקת כלים נוחים לעבודה עם תעבורה ברשת וכן גם כלים נהדרים ליצירת back-end, כגון מודול Flask שבו הנני משתמש. צד הקליינט כתוב עזרת מודול PyQt4.

* סביבות פיתוח:

Visual Studio Code –עובדת נהדר עם שפת פייתון, מאפשרת הרצה של מספר

Process-ים שונים במקביל. דרך מאד נוחה לבדק את הקוד.

* 1. מבט טופולוגי

**DMZ**

**VPN SERVER**

**FTP SERVER**

**INTERNET**

**AUTH SERVER**

**"firewall"**

**PRIVATE SERVERS**

**DA**

**LOCAL HTTPS ENCAPSULATIONWITH SOCKETS.**

**CLIENT**

* 1. מבט מודולרי

חלקי הפרוייקט שלי הינם:

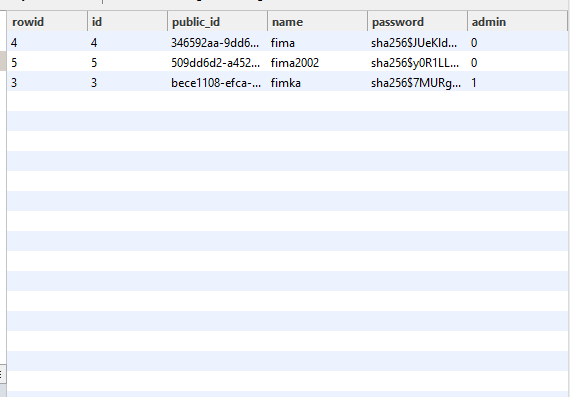
**צד הקליינט:**

* **Loginscreen.py**: קובץ המרים את ריצת המערכת, לאחר התחברות מעביר אותנו לmainwindow.
* **Mw.py:** כולל עמו את מסך האפליקציה המרכזי, כפתורים והפניות לפונקציונליות השונה של הפרוייקט.
* **Localproxy.py:** הפרוקסי הלוקאלי שמצפין את תעבורת ה TCP שעוברת בין שרת ה VPN לבין המשתמש.
* **Autoproxy.py:** הקוד היוצר אוטומציה של הגדרות Windows, בעת הפעלת התקשורת עם שרת הVPN.
* **Ftpclient.py:** קוד הקליינט של שירות הftp, מתממששק ביחד עם המסך הראשי בעת לחיצה.

**צד השרת:**

* **Api.py:** שרת ההתחברות.
* **Httpproxy.py:** שרת המנהור.
* **Ftpserver.py:** שרת הftp הידוע לשמצה
* **Ftpscanner.py:** בודק תוכן השליחה בftp (מימוש עתידי של הdmz הרצוי של ערן).
  1. מסד נתונים

ישנו מסד נתונים אחד ויחיד השומר טבלה (SQL) שבה נשמרות רשימות המשתמשים הקיימים בשירות. הטבלה מתעדכנת בעת הרשמה. הסיסמאות כמובן שמורות בHash בשיטת sha256.



* 1. פירוט מודלים עיקריים

Api.py

|  |  |  |
| --- | --- | --- |
| **Function** | **I/O** | **Explanation** |
| token\_required | **Input: HTTP Request**  **Output:** | A wrapping function that checks if the token sent via the request is valid for its user and if it has not expired. |
| create\_user | **Input:** HTTP Request  **Output:** new user | Gets a json encoded request and transforms it to a row in the user database. |
| get\_proxy | **Input:** HTTP Request  **Output:** vpn address | Sends the address of the vpn server to the user |
| login | **Input:** HTTP Request  **Output:** JWT Token | Logins users to the framework. |

Httpproxy.py

|  |  |  |
| --- | --- | --- |
| **Function** | **I/O** | **Explanation** |
| run | **Input:** client socket  **Output:** | The function creates a pipe between the client socket and the destined server socket. Gets 'CONNECT' and 'GET' requests and then proceeds with the piping |
| get\_domain\_port | **Input:** HTTP request  **Output:** | Parses CONNECT/GET requests to their url and port of destiny. |
| main | **Input:**  **Output:** | Starts the program. |

|  |  |  |
| --- | --- | --- |
| **Function** | **I/O** | **Explanation** |
| SetupUi | **Input:**  **Output:** |  |
| login\_callback | **Input:**  **Output:** |  |
| handle\_ftp | **Input:**  **Output:** |  |
| handle\_vpn | **Input:**  **Output:** |  |

Loginscreen.py, mw.py

Autoproxy.py, localproxy.py

|  |  |  |
| --- | --- | --- |
| **Function** | **I/O** | **Explanation** |
| on | **Input:** IP and port  **Output:** | Manipulates the proxy settings of the registry to specific ip and port |
| off | **Input:**  **Output:** | Returns default proxy settings of registry |
| run | **Input:**  **Output:** | The function intercepts all traffic from the machine to the vpn server and encapsulates it inside an HTTPS conncetion, with jwt token appended. |
| post\_sync | **Input:**  **Output:** | Just the tcp pingpong function between machine and the vpn server |

1. תסריטי בדיקה
   1. דגשים בבדיקה

* חיבור וניתוק ללא קריסות
* גלישה עובדת עם ביצועים טובים
* JWT tokens עם ניהול זמן ומשתמשים
* סינון נכון של הdmz
  1. תסריטי בדיקה עיקריים
* התחברות משתמש ללא בעיה
* מניפולציה נכונה של ההגדרות הלוקאליות בעת הדלקה/כיבוי של ה VPN
* בדיקה של מבני הנתונים
* בדיקה של תעבורת הרשת עם Wireshark, האם אכן הכל מוצפן?
* התנתקות משתמש
* בדיקת הסינון של הdmz

1. רפלקציה
   1. לוח זמנים מוערך לניהול הפרויקט:

|  |  |
| --- | --- |
| נובמבר | יישום הפרוקסי |
| דצמבר | יישום שרת התחברויות |
| ינואר | יישום אוטומיזציה בצד המשתמש |
| פברואר | יישום FTP |
| מרץ | יישום של ספר הפרוייקט |
| אפריל | יישום DMZ |

* 1. אתגרים ותרומה אישית

הפרוייקט מהווה עבורי משוכה לא פשוטה השנה. אין זו הפעם הראשונה שהנני מיישם פרוייקט לבית הספר. אולם, זו כן הפעם הראשונה שהוא כה אינדיבדואלי – כל אחד לעצמו. ברב הזמן אני רגיל לעבד עם עוד בן אדם – שיטת עבודה זו בהכרח תורמת לי ובמיוחד לתוצר שלנו. אני מרגיש יותר מרוכז, חד ומדוייק כאשר אני עובד עם עמית. עבודה אינדיבידואלית הרבה יותר קשה עבורי. וזה למה הפרוייקט מהווה חשיבות רבה עבורי, הצחלתי ללמד ולהתשפר בנושא הסייבר והתקשורת בעצמי, כל הכבוד לי.

אתגר מרכזי עבורי היה להתפקס ולהבין את הנושא. VPN זהו נושא מטורף וענק, שמאות ואף אלפי חברות רבות בעולם מתפרנסות ומתקיימות בזכותו. מה אני, תלמיד י"ב יכול לממש כך שיספק אותי ואת המסתכלים? הדבר אילץ אותי לחקר רבות על הנושא, על הצפנות, תעבורה ברשת וכו'.

היום אני מרגיש שאני מתמצא הרבה יותר בתחום, למרות שאין ספק כי הוא מתפתח ואף אין סופי.

מעבר לאתגר המקצועי והאתגר האישי, אתגר נוסף איתו נפגשתי במהלך הפרוייקט הוא האתגר הניהולי – אישי. הפרוייקט מצריך עבודה ודיקדוק בנושא ניהול הזמן, דבר שהוא חשוב מאד.

אני חושב שהעובדה שהתקדמתי מאד בהתחלה, דווקא האטה אותי לאורך הזמן, שכן פרוייקט מסדר גודל כזה הוא אינו ספרינט אלא מרתון, צריך לעבד באופן עקבי ולעקב אחר לוח זמנים צמוד.

אין ספק שהפרוייקט הטיל עלי לא מעט לחץ אך אני חושב שהסתדרתי איתו רגשית באופן מצויין, הוא לא השפיע על חיי החברה שלי או על שאר הישגי הלימודיים.

* 1. תובנות

אני מרגיש שאכן העמקתי רבות בנושא ה VPN ונושא ה DMZ, ומיציתי את ציפיותי. למדתי, חקרתי והתנסתי בפועל בכל הקשור לתחום. הבנתי את חשיבות GitHub בפרוייקטים מאסיביים ואני מבטיח לעצמי שמפעם לפעם אשתפר עם הכלי הזה. הבנתי גם שבתחומינו, אין נושא קיים שאין לו הסבר או דוקומנטציה, אפשר ללמד כל דבר אם רק נתאמץ ונרצה.

עם זאת אני גם מרגיש תחושה מעטה של פספוס, אני חושב שהוא לווא דווקא בהכרך בגללי. בחירת הנושא שלי לא הפיקה את מה שציפיתי, אני חושב בדיעבד שאם הייתי בוחר בנושא אחר אולי הייתי יכול לתרום לעצמי עוד יותר, במובן המקצועי. ולכן אני תמיד מרגיש שאני רוצה להתשפר ולהוכיח את עצמי יותר ויותר וכך יהיה.

1. הוראות התקנה ותפעול
   1. תצורה ודרישות קדם

* שימוש במערכת הפעלה Windows
* Python 2.7.9 Installed
* Flask, jwt, pyftpdlib, PyQt4, requests
* הפעלת צד הקליינט כ Admin

1. ביבליוגרפיה

במהלך כתיבת הפרויקט הסתמכתי על מספר מקורות מידע:

* ויקיפדיה : <https://en.wikipedia.org/wiki/HTTP_tunnel>
* GitHub
* YouTube (PyQt4 Learning)
* StackOverflow
* Use of DMZ

<https://security.stackexchange.com/questions/3667/what-is-the-real-function-and-use-of-a-dmz-on-a-network>

* Flask Tutorial https://coreyms.com/category/development/python

1. נספחים