

**ספר פרויקט - SecureBox**

שם המנחה: רועי זימון

שמות המגישים: נאור מליחי - 300685518

אמיר מגן דוד - 301401626

שנת לימודים: 2018

פקולטה: מדעים

סימול הקורס: 66001



חתימת הסטודנטים:

נאור מליחי

אמיר מגן דוד

חתימת המרצה:

רועי זימון

תוכן עניינים

[מבוא 2](#_Toc515620145)

[רקע תיאורטי 4](#_Toc515620146)

[פירוט הפרויקט 6](#_Toc515620147)

[איורים ישימים 13](#_Toc515620148)

[ביבליוגרפיה 18](#_Toc515620149)

# 

# מבוא

הצורך להגן על מידע רגיש בתוך ומחוץ לארגון הינו צורך הולך וגובר לאור איומי הסייבר וגניבת המידע ברשת.

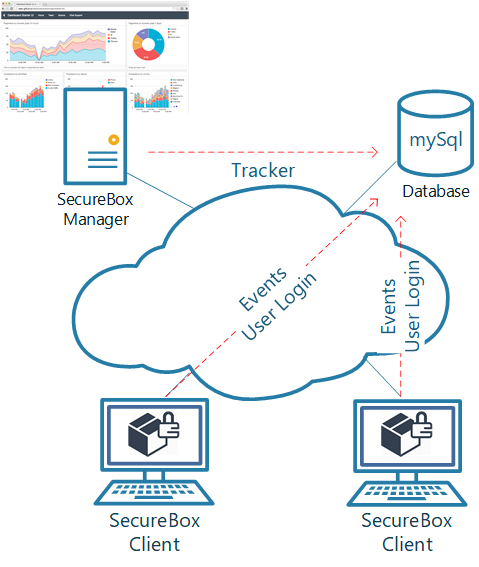
ההרשאות על קבצים ניתנים לשליטה כאשר הם נמצאים באותה מערכת הפעלה אך כאשר הם נשלחים למשתמש אחר ויוצאים מתחומי מערכת ההפעלה המקומית ונשלחים למערכת הפעלה אחרת השליטה על הקובץ אובדת.

התכונה אותה נפתח בפרויקט נקראת **SecureBox** ומטרתה לספק הגנה ובקרה על מסמכים מסווגים באמצעות תיוג הקבצים בהתאם לתוכן. המוצר עוקב באופן שוטף אחר המסמכים ומאפשר אבטחת מידע בכל נקודת זמן ושימוש בהם בלי קשר למקום המצאתם.

התוכנה תפותח בשפת Java Application ותורכב משלושה חלקים:

* צד Client - תוכנה צד לקוח, אשר תספק תהליך Login מאובטח, יכולת לטעון קבצים, לתייג את רמת ההרשאות הרצויה לכל משתמש במערכת , להצפין ולפענח את הקבצים, צפיה Read Only, עדכון פרטי משתמש.
* צד Server -תוכנה צד שרת, תספק שירותי ניהול למערכת, בקרה ומעקב אחר הקבצים, יכולת הוספה והסרה של משתמשים במערכת.
* שרת Database - מבוסס mySql.

להלן ארכיטקטורת המערכת:



# רקע תיאורטי

**שרת-לקוח**  
היא ארכיטקטורת תוכנה לחישוב מבוזר, אשר מגדירה את היחס בין תוכנות משתפות פעולה. המודל מחלק את המשימות או עומס העבודה בין ספק השירות או המשאבים - השרת, לבין מבקש השירות-הלקוח.

**צופן**   
צופן הוא אלגוריתם שמטרתו היא קידוד המידע /ו או פענוח .הצפנים מתבססים על מפתחות ומתחלקים לשתי קבוצות בהתאם לסוג המפתח שבשימוש:

**צופן סימטרי** ( סודי, פרטי) שני הצדדים בהתקשרות משתמשים במפתח זהה .  
 **צופן -א סימטרי** (פומבי) לכל צד בהתקשרות מפתח שונה.

**תכונות של צפנים**

**סודיות** ( Confidentiality ) – יכולת של הצדדים להעביר הודעה אחד לשני ללא יכולת פענוח של צד שלישי "המאזין" לתקשורת בינם .

**שלמות** ( Integrity )– יכולת לאתר שינויים שנעשו על הודעה שעוברת מצד אחד לצד שני. (כלומר – אם גורם שלישי יכול לשנות את ההודעה שאחד הצדדים העביר ללא אבחנה של הצופן, אזי הצופן הוא לא שלם

**אימות** ( Authentication )– יכולת לאמת שהמידע שהגיע לצד המקבל בהכרח הגיע מהצד השולח (ולא מגורם שלישי ).

**תהליך החלפת מפתחות**  
 האופן שבו מחליפים הצדדים את המפתחות ביניהם לטובת פענוח, כך שגורם שלישי לא יוכל להאזין לתקשורת ולהבין את המפתח (ובכך "לשבור" את ההצפנה) בעצמו . במסגרת התכונות הנ"ל ניתן לבצע חלוקה של השיטות והצפנים הנפוצים על סמך סוג המפתחות שבהם האלגוריתם משתמש

**SSL – Secure Socket Layer**

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

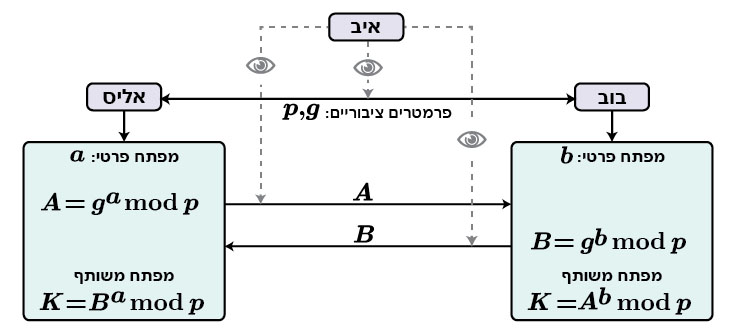
**SHA - Secure Hashing Algorithm**

משפחה של פונקציות גיבוב קריפטוגרפיות שהן חלק מתקן פדרלי של ממשלת ארצות הברית [FIPS](https://he.wikipedia.org/w/index.php?title=Federal_Information_Processing_Standard&action=edit&redlink=1)(תקן עיבוד מידע פדרלי). התקן נועד לשימוש כחלק ממנגנון אימות והבטחת שלמות מידע וחלק מתקן חתימה דיגיטלית. הפונקציות ממשפחה זו משמשות ביישומים ופרוטוקולים מאובטחים רבים.

**AES – Advanced Encryption Standard**

אלגוריתם הצפנה סמטרי שאומץ ע"י המכון הלאומי לתקנים וטכנולוגיה (NIST) של ארצות הברית כתקן הצפנה רשמי להצפנה מאסיבית. AES הוא צופן בלוקים סימטרי אלגוריתם AES נמצא בשימוש מעשי נרחב בכל העולם הן בתוכנה והן בחומרה וידוע כאלגוריתם בטוח

**Diffie-Hellman**   
פרוטוקול שיתוף מפתח, פרוטוקול מאפשר לשני משתתפים שלא נפגשו מעולם ואינם חולקים ביניהם סוד משותף כלשהו מראש, להעביר אחד לשני מעל גבי ערוץ פתוח (שאינו מאובטח) סוד כלשהו כך שאיש מלבדם אינו יודע מהו. הסוד יכול להיות בהקשר של פרוטוקול תקשורת מפתח הצפנה. פרוטוקול דיפי-הלמן מתמודד עם בעיה זו בשיטה אסימטרית על ידי פונקציה חד-כיוונית. היתרון שלו שהוא פוטר את המשתתפים בתקשורת מוצפנת לשתף מפתחות הצפנה סודיים מראש, תחת זאת המצפין יכול להכין מפתח שיחה ארעי ולהעבירו באמצעות הפרוטוקול לצד השני ואז התקשורת ביניהם יכולה להיות מוצפנת באמצעות מפתח זה עם צופן סימטרי לפי בחירה כמו AES ובגמר השימוש בו המפתח מושמד.



**RSA**

RSA היא מערכת הצפנת מפתח ציבורי דטרמיניסטית מעשית הראשונה שהומצאה והיא עדיין בשימוש נרחב במערכות אבטחות מידע מודרניות, תקשורת מחשבים ומסחר אלקטרוני.

ב RSA כבכל מפתח ציבורי, מפתח ההצפנה אינו סודי והוא שונה ממפתח הפענוח שנשמר בסוד ולכן היא נקראת אסימטרית.

הרעיון של RSA הוא בכך שהשולח משתמש במפתח ההצפנה הציבורי של הנמען כדי להצפין עבורו מסר שרק הנמען מסוגל לפענחו באמצעות המפתח הפרטי שברשותו.

# פירוט הפרויקט

בפרק זה אנו נתאר לכם באופן מפורט את היכולות אותם ביצענו במהלך הפרויקט על מנת לתת פתרון לבעיית שליחת הקובץ בצורה מאובטחת.  
נתאר שלב אחרי שלב את היכולות השונות במערכת:

1. ***תהליך ה Login***

בערכת ניתן להבחין בין שני סוגי Login.  
 האחד תהליך ה-Login של האפליקציה ל-DB והשני הוא תהליך ה-Login שמבצע המשתמש למערכת.

באיור 1.1 ניתן לראות תרשים עקרובי של התחברות למערכת.

איור 1.1 – תרשים עקרוני להתחברות למערכת



* נסביר את אופן פעולת 2 תהליכי ההתחברות:

1. **חיבור מאובטח בין ה- Agent ל-Database**
   1. עבור תהליך זה יפותח Class ייעודי אשר ייקרא MySqlCoonect
   2. כפי שצוין בפרק המבוא SecureBox תמומש באמצעות שימוש בDB מסוג MySql Server
   3. על מנת לאשר חיבור מאובטח נבצע שימוש ב-SSL
   4. המפתחות אותם ננפיק בצד הMySql Server הינם self-Signed באמצעות תוכנת open source ששמה openSSL
   5. מאפייני ה-SSL
      1. DHA - מנגנון החלפת מפתחות
      2. RSA - עבור אלגוריתם לצופן מפתח ציבורי
      3. AES256 - הצפנת המידע באופן סימטרי
      4. SHA - חתימה דיגיטלית
   6. ה-Class יאפשר לבצע פעולות query ,delete, update לשימושים נוספים שיתוארו בהמשך.
2. **תהליך Login של משתמשי המערכת**
   1. עבור תהליך זה יפותח Class ייעודי אשר יקרא LoginGUI
   2. בעת double click על אייקון האפליקציה יפתח למשתמש דף Login אשר יכיל את השדות הבאים:
      1. Username
      2. Password
   3. כמו כן, יופיע כפתור Login שיאפשר ניסיון התחברות של המשתמש באמצעות הUsername וה-Password אשר הזין המשתמש בשדות הנ"ל
   4. בעת לחיצה על כפתור ה-Login יבוצעו תהליכי הרקע הבאים:
      1. התחברות ל-DB באופן מאובטח- באמצעות SSL
      2. ביצוע תהליך Hash ל-Password באמצעות אלגוריתם הצפנה אסימטרי SHA256
      3. חיפוש ב-DB בטבלה client נתוני Username ו-Hash Password התואמים לשדות שהוזנו ע"י המשתמש.
      4. במידה ותמצא רשומה זהה ב-DB החיבור יצליח, יסגר חלון ה-Login ויפתח החלון הראשי של ה-Agent אחרת תוצג שגיאת מתאימה:
         1. Empty fields detected - כאשר יימצא שדה username או password ריק
         2. Incorrect Login Credentials - כאשר אין ערך תואם ב-DB לערכי username ו-passsword שהוזנו
         3. Database Connection Failed - כאשר תהליך החיבור ל-DB נכשל.
3. ***Encrypt/Decrypt Files***

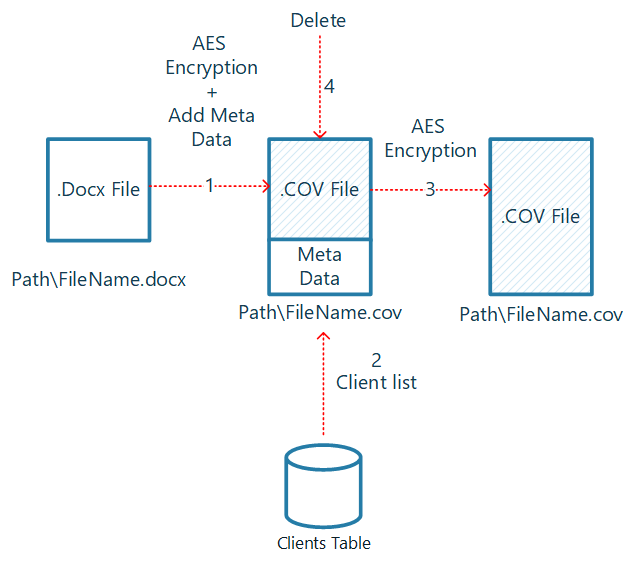
במהלך פיתוח האפליקציה נדרשנו לפתח יכולת הצפנה עבור הקובץ אותו אנו רוצים לשלוחה וכמו כן "לתייג" את שמות המשתמשים הספציפיים אותם אנו היינו רוצים להעביר אליהם את הקובץ ואת רמת ההרשאה.

1. ***הסבר תהליך ה- Encrypt:***

כאשר אנו נכנסים למסך הראשי של ה-Agent App יתאפשר עבורנו היכולת להצפין קבצים מפורמט DOCX . תהליך ההצפנה יבוצע באופן הבא:

1. באמצעות לחצן Browse יתאפשר למשתמש גישה למערכת הקבצים של מערכת ההפעלה והיכולת לבחור קובץ מסוג docx.
2. במידה ואף קובץ לא ייבחר והמשתמש ילחץ על לחצן ה-Encrypt תוצג הודעת שגיאה למשתמש שתציין שלא נבחר שום קובץ.
3. לאחר בחירת הקובץ תתאפשר היכולת להתחיל בתהליך הצפנה באמצעות לחיצה על Encrypt.
4. לאחר בחירה ב-Encryt יפתח חלון חדש אשר יאפשר להזין את המשתמשים להם אנחנו מעוניינים לאפשר הרשאות RO, תהליכי הרקע שיתבצעו:
   1. ה-Agent יתחבר ל-DB לטבלה ששמה Client
   2. שליפת כל משתמשי המערכת והצגתם למשתמש באמצעות טבלה
   3. לצד הטבלה יתווסף לחצן Checkbox אשר יאפשר לבחור את המשתמשים להם נרצה לספק הרשאות RW לקובץ. כל משתמש אחר יקבל הרשאות RO
   4. בסיום בחירת המשתמשים, לחיצה על לחצן OK תבצע את שלושת הפעולות הבאות:
      1. הצפנת המידע באמצעות אלגוריתם הצפנה סימטרי AES
      2. הוספת meta data לסוף הקובץ
      3. הצפנת המידע המכיל את ה-meta data פעם נוספת באמצעות אלגוריתם הצפנה סימטרי AES
   5. על מנת לבצע את תהליך ההצפנה הסיטמרי AES נעזר ב-Class ייעודי שיקרא ChangeUserPasswordGUI
5. הקובץ המוצפן ישמר ב-file path של הקובץ המקורי עם סיומת COV

איור 2.1 – אופן פעולת תהליך ה Encrypt



1. ***הסבר תהליך ה- Decrypt:***

כאשר הקובץ מוצפן בתהליך שהוזכר קודם הוא מגיע עם סיומת .cov והקובץ אינו ניתן לקריאה.  
על מנת שמשתמש המערכת אשר נמצא במקום מסוים על פני הגלובוס יוכל לערוך את הקובץ חייבים להתקיים התנאים הבאים:

1. ראשית צריך שה agent יהיה מותקן על המערכת
2. המשתמש צריך שיהיה לו user + password על מנת להיכנס למערכת
3. המשתמש צריך שהצד ששולח יציין אותו בתהליך ההצפנה כאשר מופיע לו ה users list.

לאחר שכל התנאים שהוזכרו קודם מתקיימים, נעבור לתהליך ה Decrypt.

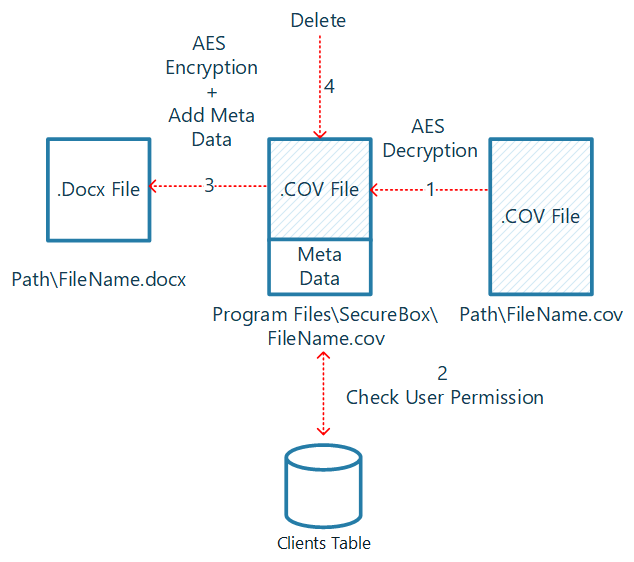
* כאשר אנו מבצעים Browse לקובץ מבוצע "קילוף" להצפנה העליונה בעזרת הצפנת AES, הצפנה אשר מאפשר לנו את היכולת לשחזר את הקובץ המוצפן.
* לאחר מכן בשלב השני המערכת בודקת האם המשתמש נמצא ב Meta Data, אותו מידע למעשה שמכיל את שם המשתמש המורשה לגשת מהשלב הראשון.

בשלב 2 למעשה קוראים מאחורי הקלעים 2 תהליכים:

1. במידה ולמשתמש שרוצה לפענח את הקובץ אין הרשאות אז המערכת מוחקת את הקובץ ושולחת שגיאה למשתמש שאין לו הרשאה.
2. במידה ולמשתמש יש הרשאות למערכת מבוצעת ההצפנה לאחור השנייה בעזרת פרוטוקול ה AES, נוצר קובץ .docx טהור, ונמחק הקובץ שעבר את ההצפנה הראשונית.

ובכך למעשה המשתמש בצד השני מקבל בסוף ההליך את הקובץ שעבר את ההצפנה המקורית הראשונה: DOCX Viewer

הערה: לא תתאפשר העתקת טקסט בהצגת קובץ מוצפן!

איור 2.2 – אופן פעולת תהליך ה Decrypt .

1. ***אופן בניית מסד הנתונים***

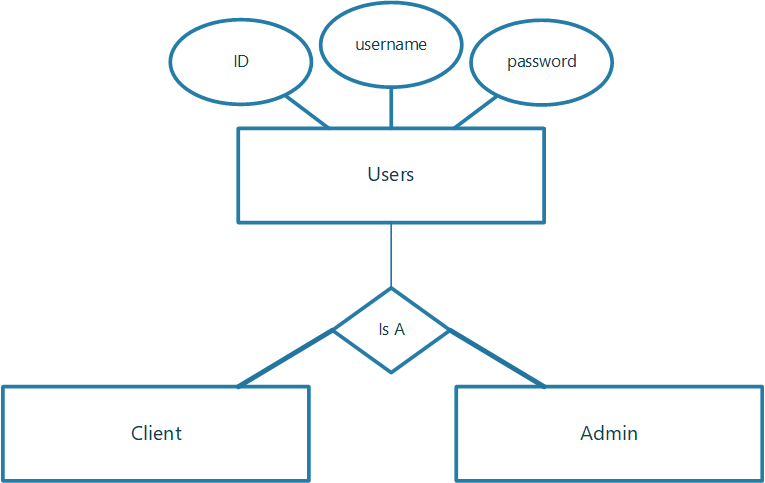
*סוגי משתמשים*

מסד הנתונים בפרויקט הוא דיי פשוט ולא נבנה בצורה מורכבת (אין דרישה אפליקטיבית ל db מורכב)  
נוכל לראות כי אנו מגדירים אובייקט עיקרי, שמתחלק ל 2 אובייקטים אשר מהווים את משתמשי המערכת:

1. תחת האובייקט users אנו מגדירים את ההגדרות הבסיסיות של משתמש המערכת שלנו.

תחתיו יהיה לנו קשר של is a לצורך הגדרת 2 סוגי משתמשים:

1. מנהל המערכת, אשר שם יהיו לו הגדרות admin לצורך עריכת קבצים, הגדרת משתמשים וכו
2. משתמש המערכת, אשר לו יהיו הגדרות צפייה בקובץ בלבד ללא הגדרות אדמיניסטרטיביות

איור 3.1 – דיאגרמת ERD לבניית מסד הנתונים

*התראות אפליקטיביות לצורך תחקור*

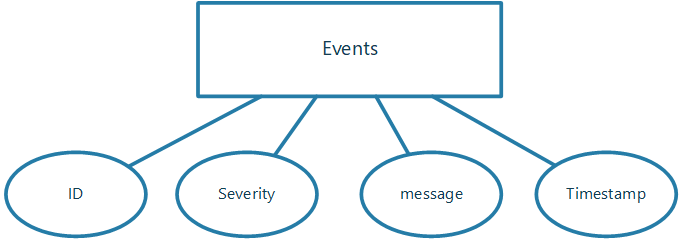
על מנת לתחקר את אופן פעולת המערכת כגון: משתמשים שלא הצליחו להיכנס למערכת, משתמשים שכן הצליחו להיכנס למערכת, קובץ שהגיע ליעד בהצלחה, קובץ שנכשל וכו, אנו שומרים את ההתראות הללו במערכת שלנו.

לשם כך הוגדר feature נוסף במערכת על מנת לשמור ולתחקר את ההתראות הללו.

לכן הגדרנו אובייקט בטבלה נפרדת בשם EVENTS ושם יהיו הערכים הבאים:

* ID לצורך מניה
* Severity לצורך הגדרת חומרת האירוע
* Message אשר שם יהיה תוכן ההודעה
* Timestamp - הזמן בו האירוע התרחש

איור 3.2 – אופן טבלת ה events במערכת



# איורים ישימים

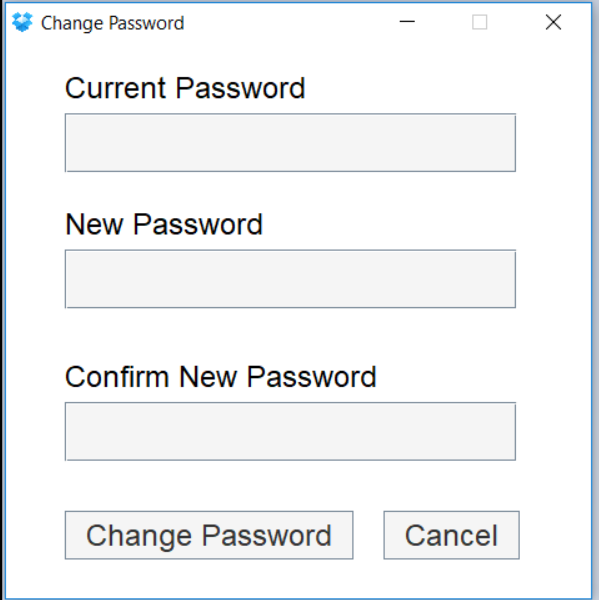
בעמודים הבאים נציג את מרכיבי האפליקציה ונסביר על כל אחד מהם.

**איור 4.1 – התחברות למערכת**



* באיור זה אנו רואים את אופן ההתחברות לאפליקציה בעזרת שם משתמש וסיסמא.

**איור 4.2 – שינוי סיסמא**



* באיור זה ניתן לראות את אופן שינוי הסיסמא בעזרת הקלדת הנתונים החדשים בשדות המתאימים.

**איור 4.3 – הסוכן**

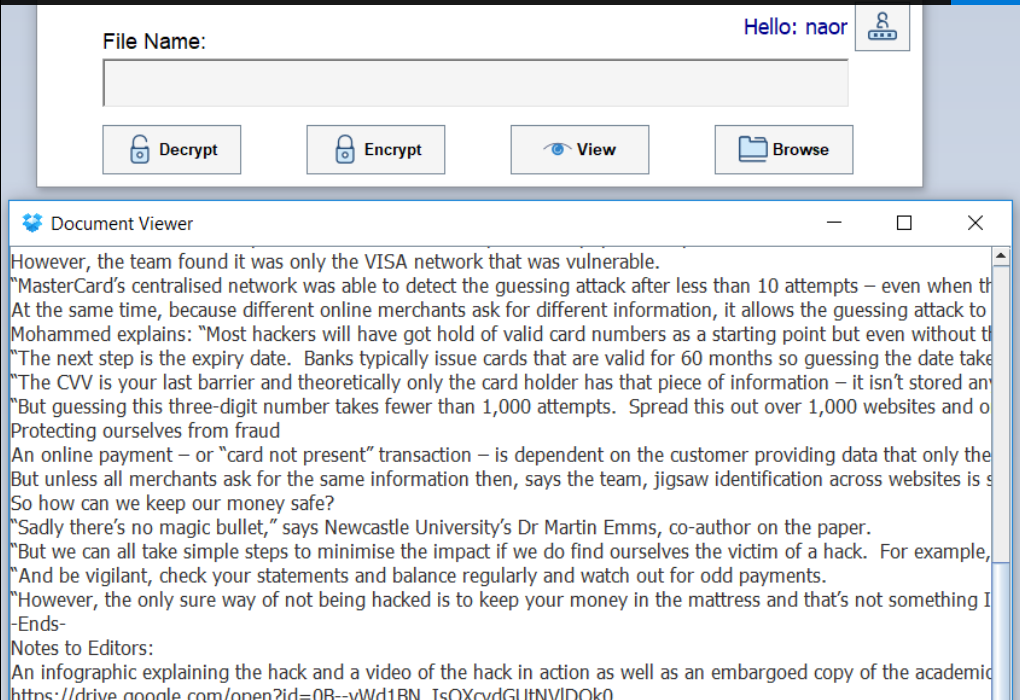
****

באיור זה אנו יכולים לראות איך בנוי הסוכן אשר מותקן על תחנות הקצה.

נסביר את האופציות של המשתמש לפעולה:

* **Browse** – יכולת העלאת קובץ לטובת צפיה/פענוח.
* **View** – במידה והמשתמש רוצה לצפות בקובץ ובמידה ויש לו אישור לכך לחצן זה מקנה את היכולת מבלי לשנות/לערוך את הקובץ אלא צפייה בלבד(לא תתאפשר העתקה מתוך הפלט של הקובץ).
* **Decrypt** – במידה וביצענו העלאה של קובץ ואנו מעוניינים לשנות אותו, ובמידה ויש האת ההרשאות המתאימות לעריכת הקובץ, כפתור זה יפענח את הקובץ ונוכל לערוך אותו ולקודד אותו מחדש במידה ונרצה.
* **Encrypt** – יכולת הצפנת קובץ עבור משתמשים ספציפיים.

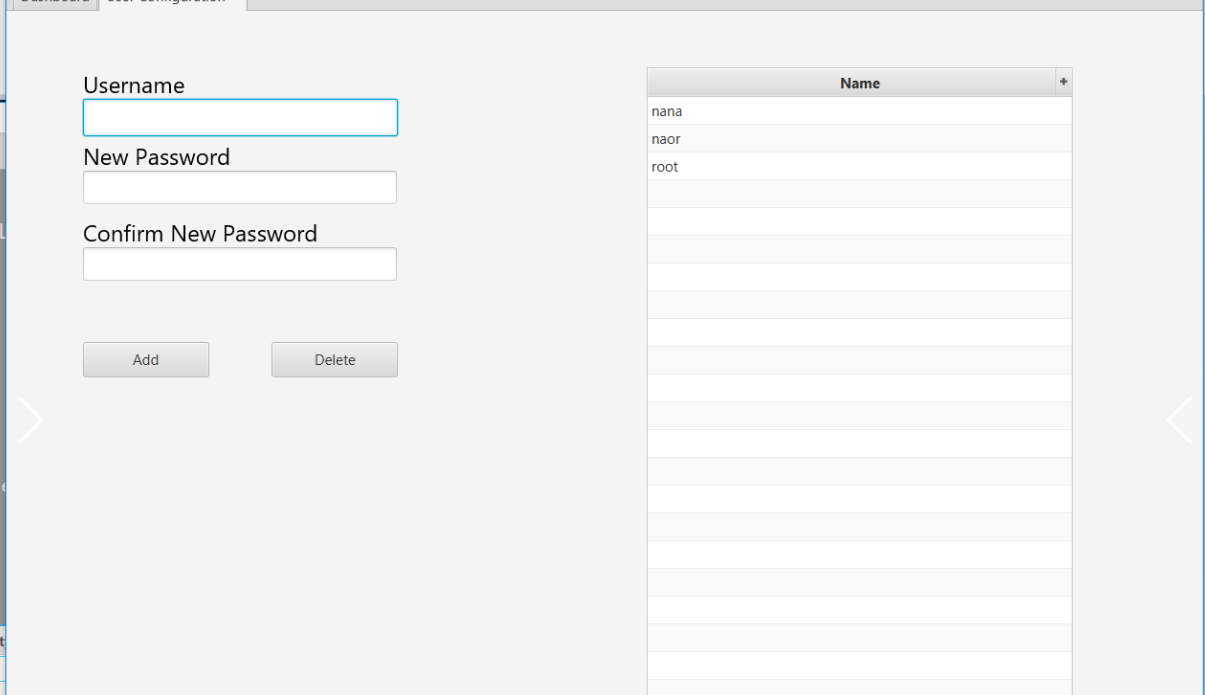
**איור 4.4 – יכולת צפייה בקובץ מבלי לפענחו**

****

במידה ואני מעוניינים לצפות בקובץ מסוים ובמידה ויש לנו את ההרשאות המתאימות עבור צפייה בקובץ זה, לאפליקציה יש יכולת צפייה מבלי לבצע Encrypt לקובץ.

באיור זה אנו רואים את הפלט שיוצג למשתמש, לא ניתן לבצע copy לפלט זה בעזרת העכבר/מקלדת.

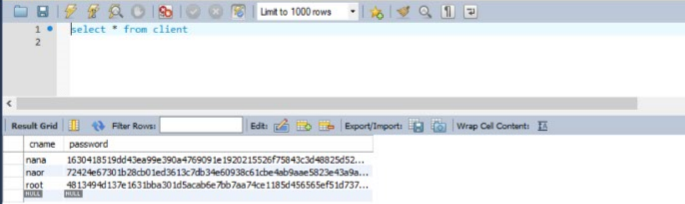
**איור 4.5 – הגדרות משתמש (מנהלי מערכת בלבד)**



למנהלי המערכת יש את האפשרות בלבד להוסיף משתמשים חדשים ל db.

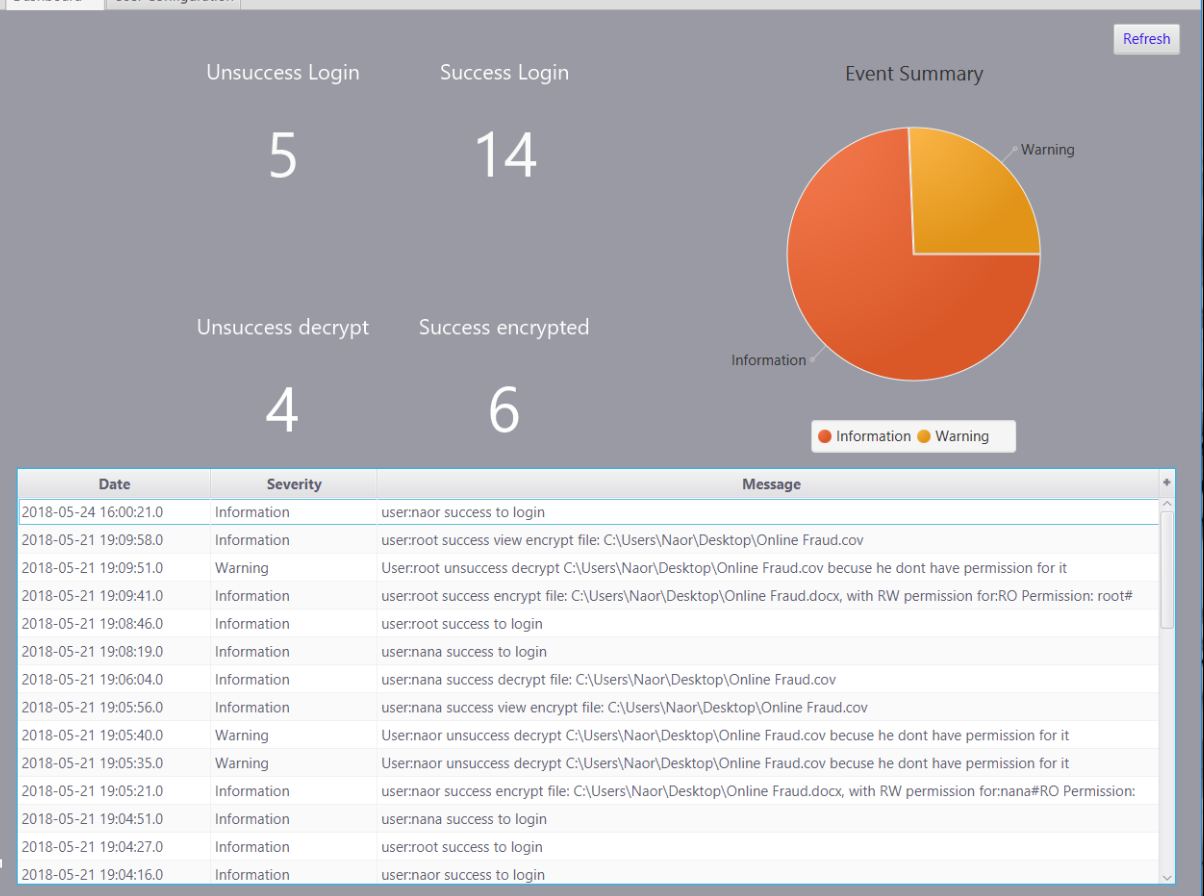
במסך זה אנו רואים את אופן הגדרת המשתמשים החדשים של המערכת, מיד לאחר מכן תיווסף שורה חדשה עם המשתמש הרלוונטי שנוצר.

**איור 4.6 – אופן הכנסת המשתמשים ל DB**



הטבלה מורכת מ 2 עמודות של המשתמש והסיסמא המוצפנת ב db.

**איור 4.7 – מנהל האירועים**

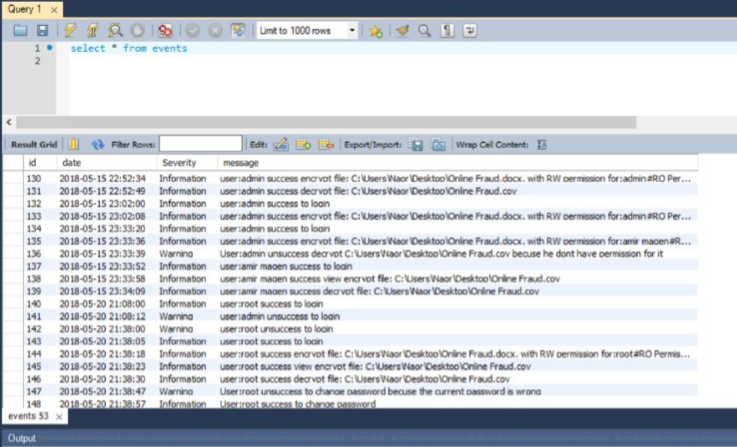
****

באיור זה ניתן לראות את היכולת החשובה והמשמעותית עבור מנהל המערכת.  
כאחראים על האפליקציה ועם ריבוי פעולות הסייבר בעולם, חשוב למנהל המערכת לדעת מה קורה מבחינת האפליקציה שלו והאם ישנם ניסיונות חדירה לתוך המערכת.

מנהל האירועים בא לתת מענה עבור איומים אלו, בוצע אפיון אפליקטיבי עבור כמה סוגי התראות חשובות שיועברו לכיוון ה DB , ומשם בעזרת שליפות רלוונטיות ניתן להציג סטטיסטיקות למנהלי המערכת כגון:

* Pie chart אשר מכיר בתוכו את סוגי ההתראות שאופיינו עבור המערכת.
* נתונים יבשים על פעולות שהצליחו/נכשלו כגון:
  + אימותים שצלחו
  + אימותים שנכשלו
  + הצפנות שצלחו
  + הצפנות שנכשלו
* ההתראות עצמן שניתן לראות בחלק התחתון של האיור, אשר יכילו בתוכן את התאריך שקפצה ההתראה, החשיבות של ההתראה ותוכן ההודעה עצמה.

**איור 4.8 – אופן אחסון הנתונים ב DB**



באיור זה אנו יכולים לראות איך אוחסנו הנתונים בתוך ה DB , מנתונים אלו למעשה אנו מציגים את העמוד של ניטור המערכת.

# ביבליוגרפיה

1. <https://he.wikipedia.org>
2. מצגות מקורסים שנלמדו: נאור זה סבבה לרשום את הסעיף הזה?
   1. אבטחת מחשבים
   2. אבטחת רשתות מרשתת
   3. קורס sql
   4. קורס תקשורת נתונים
   5. קורס java
3. https://stackoverflow.com/