**עמוד השער**

תמונה

**יכלול:**

1. לוגו של בית הספר
2. לוגו של האפליקציה/יישום/מערכת
3. נושא העבודה
4. שם התלמיד
5. ת"ז של התלמיד
6. שם בית הספר ועיר
7. שמות המנחים + המורים המלווים
8. מועד הגשת המסמך (חודש ושנה)

**תוכן עניינים וראשי פרקים (אוטומטי לא ידני)**

# Phoenix U.E.S

**פרק א' - 'שם המערכת' - ייזום**

## תיאור ראשוני של המערכת

המערכת היא מערכת להצפנת דיסק־און־קי ( DOK).   
ה דיסק־און־קי – אינו מאובטח וברגע שהוא נופל לידיים זרות, כל המידע שבו גלוי לעיני כל.  
המערכת שלי תדאג לאבטחת כלל המידע ב דיסק־און־קי בצורה שתאפשר אך ורק לבעל ה דיסק־און־קי לפענח אותו. במקרה שה דיסק־און־קי יפול לידיים זרות והמערכת תזהה זאת, היא תדאג להשמיד את כל תוכנו.  
במקביל לעבודה עם ה דיסק־און־קי הוא יגובה לשרת מרוחק, מה שיאפשר לבעל ה דיסק־און־קי לשחזר מידע לאחר אובדן ה דיסק־און־קי ו/או לאחר השמדת ה דיסק־און־קי.

**המוצר המוגמר:**

מערכת דיסק־און־קי מאובטחת הכוללת הצפנה מלאה של כל המידע ב דיסק־און־קי , בדיקת משתמשים, שמירה עותק של כל הקבצים בשרת מרוחק למען שיחזור, מחיקה מלאה של הדיסק־און־קי בכישלון של כניסה אליו, אפשרות שיחזור דרך השרת, הפעלת ההצפנה בכל פעם מחדש שה דיסק־און־קי מופעל, אחרי הכנסה של הסיסמא ושם משתמש נכונים להחזיר את כל הקבצים למצב הראשוני שלהם.

**אז למה בחרתי בפרוייקט הזה?**

בחרתי בפרוייקט הזה כי אני מאוד מתעניין באבטחת מידע ואיך כל זה פועל גם שהשרת אינו מחובר ועובד כל הזמן האתגר בפרוייקט עניין אותי. בנוסף לזה כל שהותי במגמה הייתי עם דיסק־און־קי והוא באמת אף פעם לא היה מוצפן וכל בן אדם היה יכול לגשת לי לפרטים ולדברים שלי מבלי שום בעיה והמחשבה הזאת מטרידה אותי ומונעת ממני פרטיות והפרוייקט הזה יעזור לי לפרטיות שלי ולהגנה שלי.

**אתגרים שאני צופה להם בפרוייקט:**

יצירת קובץ מערכת פנימי שמסוגל לאמת גישה גם ללא חיבור לשרת, הצפנה ברמת קובץ בודד ובמקביל גם ברמת כלל דיסק־און־קי, יצירת גיבוי מאובטח לכל הקבצים המוצפנים, התאמה בין השרת לדיסקו דיסק־און־קי נקי כך שהמערכת לא תישבר גם אם המשתמש מעתיק או מזיז קבצים.

## הגדרת הלקוח

הלקוח הוא כל בן אדם אשר נחזיק בבעלותו דיסק־און־קי ורוצה שהוא יהיה מאובטח ולא יוכלו לפרוץ לו פנימה + שמירה על הקבצים במקום אחר שיוכל לשחזר אותם במקרה הצורך.

## למי מיועדת המערכת ומי הולך להשתמש בה?

המערכת מיועדת לכל אדם שמחזיק דיסק־און־קי – תלמידים, אנשי מחשבים, בעלי עסקים, עובדים שמחזיקים קבצים מסווגים, ואפילו אנשים פרטיים שרוצים לשמור תמונות/מסמכים חשובים.

### מטרות לכל לקוח:

**אבטחת מידע מלאה:** הצפנת הקבצים כך שלא ייפתחו ללא הרשאות  
**גיבוי ה דיסק־און־קי**: שמירת כל המידע ב דיסק־און־קי במקום נוסף  
**נוחות שימוש:** כל ההצפנה מתבצעת אוטומטית בעת חיבור לשרת.  
**הגנת מידע מקסימלית:** מחיקה אוטומטית לאחר כישלון בהזדהות.  
**שחזור דרך השרת:** יכולת לשחזר מידע באמצעות הגיבוי המוצפן.

## בעיות, תועלות וחסכונות

### הבעיה

רוב ה דיסק־און־קי -ים אינם מוצפנים, ומי שיגנוב או ימצא אותם – יכול לקרוא את כל התוכן בקלות וגם אם איבדת את ה דיסק־און־קי איבדת את כל מה שהיה בתוכו

### מה אנחנו מנסים להשיג?

מניעת גישה לא מורשית.  
שמירה על פרטיות המשתמש.  
העלאת רמת האבטחה בלי שימוש בחומרה מיוחדת.  
גיבוי כלל המידע ב דיסק־און־קי

### תועלות צפויות:

רמת אבטחה גבוהה ל דיסק־און־קי.  
קושי משמעותי לפריצת ההצפנה.  
אמינות גבוהה בהגנה על נתונים. שמירה מוגנת על הנתונים.

### שירותים שהמערכת תיתן:

הצפנה  
אימות משתמש  
מחיקה אוטומטית למניעת גניבת מידע  
גיבוי מאובטח  
שיחזור קבצים שנמחקו\ נאבדו

## השוואת העבודה עם פתרונות קיימים:

### 1. VeraCrypt

VeraCrypt היא תוכנה חינמית וקוד פתוח להצפנת DOK מלאים או יצירת "מכולות" מוצפנות. המערכת משתמשת באלגוריתמי הצפנה כמו AES, Serpent ו‑Twofish, ולעיתים שילובים שלהם. הצפנת DOK מתבצעת באמצעות יצירת שכבת קידוד מעל הקבצים, והגישה אליהם אפשרית רק לאחר הזנת סיסמה נכונה.

VeraCrypt מבצעת הצפנה בזמן אמת (on‑the‑fly), כלומר כל קובץ שנפתח או נשמר עובר פענוח/הצפנה מיידית בלי שהמשתמש מרגיש בכך. כדי לפתוח את הכונן המוצפן נדרש להזין סיסמה או להשתמש בקובץ‑מפתח מיוחד. אם הסיסמה שגויה – פשוט לא ניתן לפתוח את הכונן, אך התוכנה **אינה מוחקת את הקבצים** לאחר ניסיונות כושלים, אלא רק חוסמת גישה עד להזנת סיסמה נכונה.

### 2. BitLocker (Windows)

כלי מובנה ב-Windows להצפנת התקנים. מספק הצפנה, אך אינו כולל מערכת ניסיונות כושלים ומחיקה.

שניהם נותנים הצפנה, אבל המערכת שלי מוסיפה מנגנון הגנה ייחודי – מחיקה מלאה לאחר כישלון + שיחזור הקבצים לבעל ה דיסק־און־קי בעזרת השרת.

## האם צפויים קשיים או מגבלות?

המנגנון חייב להיות אמין גם בלי חיבור לשרת.  
שמירה על כל הקבצים של כל המשתמשים במקום מאובטח ואמין. שיחזור הקבצים במקרה הצורך.

## האם מדובר בטכנולוגיה חדשה?

הצפנה עצמה אינה חדשה, אך בניית מנגנון מחיקה אוטומטי ומערכת אימות מקומית ללא שרת – זה יחסית מתקדם. גם שיחזור הקבצים אינם מתקדמים מאוד אך הפרוייקט יוצר שילוב מעניין בין השנים.

## מגבלות בהגדרת המערכת

ייתכן שמערכות הפעלה מסוימות יחסמו הפעלה אוטומטית מה דיסק־און־קי.  
ייתכן צורך בדרייברים או הרשאות מיוחדות.  
לטובת הגיבוי ה דיסק־און־קי ושרת הגיבוי צריכים להימצא על אותה הרשת (מגבלות בה"ס)

## 

## 

## 

## תיחום הפרויקט

### תחומים שבהם הפרויקט עוסק:

**רשתות:** תקשורת בין השרת ל דיסק־און־קי.  
**מערכות הפעלה:** גישה לקבצים, הרשאות מערכת, הצפנה.  
**אבטחת מידע:** הצפנה, אימות, מחיקה יזומה.

### תחומים שבהם הפרויקט לא מטפל:

לא מטפל בניהול משתמשים מתקדם.  
לא מגן מפני שיבוש פיזי או גניבה של החומרה עצמה.

**פרק ב' - 'שם המערכת' - אפיון**

1. פרוט המערכת:

המערכת היא מערכת הצפנה של דיסק און קי (DOK)

המערכת עצמה לוקחת DOK שהוא ל מאובטח בכלל, כל אחד יכול להיכנס לדיסק און קי ולקחת מה שהוא רוצה ממנו

המערכת שלי מצפינה ומגבה את המידע שקיים ב DOK, (שמירה על נתונים במקרה וצריך שחזור). המערכת נותנת הרשאת גישה ופיענוח של ההצפנה רק לבעל ה DOK מה שיוצר אבטחה והגנה מפני פורצים שרוצים את המידע.

כאשר אתה מתחבר ל- DOK הוא יבקש ממך סיסמא וקוד ואם אתה טועה שלוש פעמים ימחק לך כל המידע מה DOK ובעל ה DOK יקבל הודעה (מייל) האומרת לו שנמחק לו הכל ואם הוא רוצה לשחזר הוא צריך לפנות לשרת שישחזר לו הכל מכיוון שהכל מגובה אז קיימת האפשרות.

* + מה היכולות שהיא תעניק למשתמש:
* הצפנת DOK – הצפנה מלאה של ה DOK
* גיבוי – גיבוי הקבצים במקום מאובטח
* שמירה על המידע – לאחר שלושה ניסיונות כניסה לא מורשים ל DOK תהיה מחיקה מלאה של ה DOK
* שיחזור – אם אתה מגיע למצב שנמחק לך ה DOK תהיה יכולת לשחזר את ה DOK
* התרעות – במקרה של מחיקת ה DOK המשתמש מקבל התרעות מהמערכת ומוזמן לשחזור

1. פירוט הבדיקות ('קופסא שחורה')

| **מספר** | **שם הבדיקה (שם שיעיד על תוכן הבדיקה)** | **מה אמורה לבדוק** | **איך מתכננים לבדוק**  **(לתאר בפירוט את שלבי הבדיקה)** |
| --- | --- | --- | --- |
| 1. | DOK מוצפן | האם כל ה DOK מוצפן בכל פתיחה שלו | פותחים את ה DOK ומנסים להיכנס לקובץ ורואים אם הוא מוצפן ועושים זאת שוב אחרי שפתחת את ההצפנה ורואים שניתן לגשת לקובץ |
| 2. | שגיאת סיסמא | אם אחרי שלוש שגיאות ה DOK נמחק | מכניסים שלושה שגיאות ל DOK ורואים אם אחרי שלושה שגיאות נמחק ה DOK |
| 3. | התראת מחיקה | לבדוק שבעל ה DOK מקבל התראה כשה DOK נמחק | לוודא שבסיום הבדיקה הקודמת נשלח מייל לבעל ה DOK |
| 4. | שיחזור | אם אחרי מחיקה מצליח לשחזר את הכל | אחרי קבלת הודעה מהשרת לעשות את מה שהוא מבקש ממך ולראות שהכל חוזר לקדמותו |
| 5. | עבודה ללא שרת | שניתן לעבוד עם ה DOK גם כשהשרת לא מחובר | מנסים להיכנס ל DOK ומוודאים שאחרי הכנסת סיסמא נכונה הכל זמין |
| 6. | גיבוי שלם | שהקבצים שאיתם עבדנו כשלא היה שרת, ברגע שיש שרת מגובים | לעבוד ללא שרת על אחד הקבצים, לבצע בו שינוי. לסגור את המערכת. להפעיל את השרת, להפעיל את המערכת. בסיום לגשת לשרת ולראות שהקובץ מהפעם הקודמת גובה |

1. תכנון לוח זמנים לפרויקט

| **פעילות** | **זמן התחלה מתוכנן** | **זמן סיום מתוכנן** | **זמן התחלה בפועל** | **זמן סיום בפועל** | **הערות** |
| --- | --- | --- | --- | --- | --- |
| יזום | 13.11.25 | 16.11.25 | 15.11.25 | 16.11.25 |  |
| אפיון | 16.11.25 | 30.11.25 | 26.11.25 | 30.11.25 |  |
| ניתוח | 30.11.25 | 21.12.25 | 30.11.25 | 4.12.25 |  |
| עיצוב | 21.12.25 | 22.2.26 |  |  |  |
| הצפנה ופענוח של DOK (בלי קשר לשרת) | 22.2.26 | 24.2.26 |  |  |  |
| מימוש תקשורת לשרת | 24.2.26 | 26.2.26 |  |  |  |
| מימוש מערכת גיבויים | 26.2.26 | 28.2.26 |  |  |  |
| שרת יוצר משתמשים ומשתיל שרתים קטנים שלו באחרים | 28.2.26 | 1.3.26 |  |  |  |
| תפעול המיני שרתים | 1.3.26 | 4.3.26 |  |  |  |
| שמירת נתונים "בענן" | 4.6.26 | 6.3.26 |  |  |  |
| שיחזור DOK | 6.3.26 | 13.3.26 |  |  |  |
| פיצירים נוספים | 13.3.26 | 15.3.26 |  |  |  |
| גרסה ראשונית | 22.02.26 | 15.03.26 |  |  |  |
| מסמך בדיקות | 15.03.26 | 1.4.26 |  |  |  |
| מדריך למשתמש | 15.03.26 | 1.4.26 |  |  |  |
| גרסה סופית | 1.4.26 | 1.5.26 |  |  |  |
| סגירת תיק פרויקט | 1.4.26 | 1.5.26 |  |  |  |

* **לוח הזמנים – הינו מסמך חי ויש לעדכנו לאורך כל הפרויקט**

1. ניהול סיכונים בפרויקט

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **הסיכון** | **פירוט הסיכון** | **רמת הסיכון**  **(קל/בינוני/קשה)** | **תיאור דרכים (לפחות 2 ) להתמודדות עם הסיכון ולהקטין אותו** | **מה בוצע בפועל** | **תאריך** |
| אי עמידה בזמנים | פרויקט לא יושלם | קשה | * הקדמת לו"זים משימות * ארגון הזמן בצורה יעילה * להתחיל בדברים היותר קשים |  |  |
| הצפנה לא עובדת | DOK אינו מאובטח | קשה | * לחקור על הצפנות שונות * מעבר על ההצפנה ולראות איך אפשר לשדרג אותה |  |  |
| "שרת" שתול לא עובד | DOK אינו יפענח את עצמו + יצפין | קשה | * החלפת דרך ההתקנה של השרת * ניטור רציף כדי לזהות תקלות בזמן אמת |  |  |
| שיחזור קבצים | אין שמירת נתונים ל-DOK ויכול לאבד הכל | קשה | * שימוש בשיטת שיחזור שונה * בדיקה אוטומטית מקיפה שבודקת האם הכל שוחזר |  |  |
| שמירת משתמש + נתונים | אי סדר בשיחזור ואיבוד נתונים | קל | * שמירה במסמך צד שהשרת יצור * שמירה בתוך DOK של השרת |  |  |
| מחיקת הפרויקט | איבוד כל הנתונים וכל השרת | קל | * שמירה בDRIVE * שמירה בGITHUB * שמירה בתוך DOK של השרת |  |  |

* **ניהול סיכונים – הינו מסמך חי ויש לעדכנו לאורך כל הפרויקט**

**פרק ג' - 'שם המערכת' - מסמך ניתוח**

**יכולות בצד שרת:**

שם היכולת: הרשמה למערכת

מהות היכולת: רישום משתמש חדש במערכת

אוסף יכולות:

* קבלת נתונים מהלקוח
* פענוח
* בדיקה מול בסיס הנתונים
* גיבוב סיסמא
* הוספה לבסיס נתונים
* החזרת תשובה מוצפנת

רשימת אובייקטים: הצפנה/פענוח, תקשורת, בסיס נתונים

שם היכולת: התחברות למערכת

מהות היכולת: התחברות למערכת למשתמש קיים

אוסף יכולות:

* קבלת נתונים מהלקוח
* פענוח
* בדיקה מול בסיס הנתונים
* בדיקת בקשה
* החזרת תשובה מוצפנת

רשימת אובייקטים: הצפנה/פענוח, תקשורת, בסיס נתונים

שם היכולת: יצירת קבצי מערכת

מהות היכולת: יצירת קבצי מערכת על מנת לטעון ל DOK של המשתמש

אוסף יכולות:

* קבלת נתונים מהלקוח
* קבלת נתוני הלקוח מבסיס הנתונים
* יצירה של קבצי המערכת
* הצפנה
* שליחה

רשימת אובייקטים: בסיס נתונים, ניהול קבצים, הצפנה/פענוח, תקשורת,

שם היכולת: גיבויים

מהות היכולת: מגבה את הDOK על מנת מתן שיחזור

אוסף יכולות:

* קבלת נתונים מהלקוח
* פענוח שם הקובץ
* שמירה ב"ענן" של השרת שם – קובץ לפי משתמש

רשימת אובייקטים: הצפנה/פענוח, תקשורת, בסיס נתונים, ענן

שם היכולת: שיחזור DOK

מהות היכולת: משחזר את הDOK במקרי הצורך

אוסף יכולות:

* קבלת נתונים מהלקוח
* שליפת קבצי המשתמש מה 'ענן'
* שליחת קבצים

רשימת אובייקטים: הצפנה/פענוח, תקשורת, בסיס נתונים, ענן

שם היכולת: שליחת הודעות למשתמש

מהות היכולת: שולח התרעה למשתמש במקרה הצורך

אוסף יכולות:

* קבלת נתוני משתמש (התראה)
* שליפת נתונים מבסיס נתונים
* שליחת הודעת מייל למשתמש

רשימת אובייקטים: תקשורת, בסיס נתונים, מייל

**יכולות בצד לקוח – עמדת התקנה/שחזור:**

שם היכולת: הרשמה

מהות היכולת: רישום משתמש חדש

אוסף יכולות:

* ממשק משתמש - הכנסת שם משתמש, סיסמא, מייל
* שליחה מוצפנת לשרת
* קבלת תשובה של הרשמה
* הצגתה למשתמש

רשימת אובייקטים: ממשק משתמש, הצפנה/פענוח, תקשורת

שם היכולת: התחברות

מהות היכולת: מתחבר שרת

אוסף יכולות:

* ממשק משתמש - הכנסת שם משתמש וסיסמא
* שליחה מוצפנת לשרת
* קבלת תשובה שהתחברת
* הצגת התשובה למשתמש

רשימת אובייקטים: ממשק משתמש, הצפנה/פענוח, תקשורת

שם היכולת: התקנת DOK

מהות היכולת: מתקין את קבצי המערכת לתוך הDOK

אוסף יכולות:

* שליחת בקשה מוצפנת של התקנה לשרת
* קבלת קבצי מערכת
* הטענה קבצי המערכת ל DOK

רשימת אובייקטים: ממשק משתמש, הצפנה/פענוח, תקשורת, DOK

שם היכולת: שחזור DOK

מהות היכולת: משחזר את הDOK בעת הצורך

אוסף יכולות:

* התחברות
* שליחת בקשת שיחזור DOK מוצפן
* קבלת קבצי המשתמש
* העתקת קבצים המשתמש ל DOK

רשימת אובייקטים: ממשק משתמש, הצפנה/פענוח, תקשורת, DOK

**יכולות בצד לקוח:**

שם היכולת: כניסה לDOK

מהות היכולת: כניסה לתוך הDOK

אוסף יכולות:

* חיבור הDOK למחשב
* ממשק משתמש - הכנסת סיסמא
* פענוח רשימת הקבצים והצגתה

רשימת אובייקטים: DOK**,** ממשק משתמש, סיסמא, רשימת קבצים, פענוח

שם היכולת: יציאה DOK

מהות היכולת: יציאה מהDOK

אוסף יכולות:

* כניסה לDOK
* ממשק משתמש - בקשת יציאה
* הצפנת כל הקבצים הפתוחים
* שליחה לגיבוי של כל הקבצים שטרם נשלחו
* קבלת אישור ו הוצאת הDOK

רשימת אובייקטים: DOK, ממשק משתמש, הצפנה, גיבוי, תקשורת

שם היכולת: עבודה עם הDOK

מהות היכולת: עבודה עם הDOK עצמו

אוסף יכולות:

* כניסה לDOK
* ממשק משתמש – הצגת רשימת הקבצים
* פעולות לפי הצורך שלך
* ניטור קבצים פתוחים ושליחת שינויים לגיבוי לשרת אם קיים או הוספת השם ל'רשימת גיבויים'

רשימת אובייקטים: DOK, ממשק משתמש, גיבויים, ניטור קבצים פתוחים, תקשורת

שם היכולת: פיענוח קובץ DOK

מהות היכולת: מפענח קובץ DOK ספציפי

אוסף יכולות:

* כניסה לDOK
* פענוח רשימת קבצי DOK
* ממשק משתמש - בחירת קובץ
* פענוח הקובץ הספציפי
* ניטור שינויים בקובץ הפתוח
* שמירה של הקובץ שפענח
* שליחת שינוים לשרת אם קיים / לקובץ 'רשימת גיבויים' את שם הקובץ

רשימת אובייקטים: DOK, פענוח, ממשק משתמש, קובץ, תקשורת, ניטור

שם היכולת: הצפנת קובץ DOK

מהות היכולת: מצפין את הקובץ DOK

אוסף יכולות:

* פיענוח קובץ DOK
* ניטור הקובץ
* זיהוי 'סגירת קובץ'
* הצפנה
* שליחת גיבוי אם שרת קיים / לקובץ 'גיבוי' את שם הקובץ

רשימת אובייקטים: DOK, הצפנה, ניטור, גיבוי, תקשורת

שם היכולת: שליחת קבצים לגיבוי בשרת

מהות היכולת: שולח קבצים על מנת לגבות אותם בשרת

אוסף יכולות:

* כניסה לDOK
* זיהוי תקשורת עם השרת – שליחת קבצים שטרם גובו ע"פ הרשימה

רשימת אובייקטים: DOK, הצפנה, תקשורת, FILE XEPLORER, ניטור, רשימת קבצים לגיבוי

שם היכולת: מחיקת הDOK

מהות היכולת: מחיקת כל הDOK

אוסף יכולות:

* כניסה לDOK
* כשל בכניסה
* מחיקה של הDOK
* שליחת הודעת מחיקה לשרת

רשימת אובייקטים: DOK, מחיקה, תקשורת

**פרק ד' - 'שם המערכת' – העיצוב**

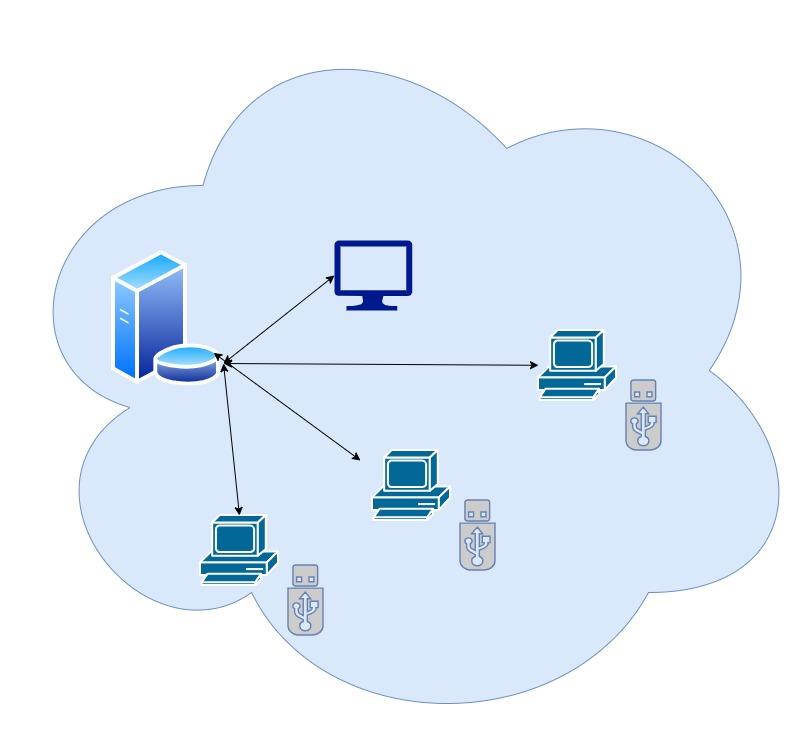
תיאור הארכיטקטורה של המערכת המוצעת

המערכת תהיה שרת שיהיה מחשב בעל מערכת הפעלה של WINDOWS

השרת מתקשר עם עמדת ההתקנה שגם הוא צריך להיות בעל מערכת הפעלה של WINDOWS שם הלקוח מגיע ויכול להתקין את עצמו בפעם הראשונה אך הלקוח צריך להיות גם הוא בתקשורת עם השרת וגם הוא חייב להיות מחשב בעל מערכת הפעלה של WINDOWS והUSB בוא הוא משתמש חייב להיות FAT32

השרת מחשב מתקשר עם עמדת ההתקנה שהיא גם כן מחשב

השרת מתקשר עם לקוח שהוא מחשב עם USB



הטכנולוגיה הרלוונטית

* פירוט ההחלטות שנלקחו בחשבון בעת בחירת הטכנולוגיה (שפת תכנות, מ"ה, תקשורת, תחומי עיניין, ועוד...)

שפת התכנות שאני אשתמש בה היא פייטון וSQL מכיוון שבהם אני הכי מתורגל ואני צריך בסיס נתונים בשביל הפרוייקט.

התקשורת שתהיה לי תהיה תקשורת מבוססת SOKETS המעבירים מידע בעזרת פרוטוקול TCP מכיוון שכל הרעיון של הפרוייקט הוא הגנה ואמינות ואני רוצה שכל המידע יעזור כמו שצריך

אני משתמש בפרויקט בהרבה הצפנות וההצפנות שאני אשתמש בהם יהיו AES לשמות קבצים ולקבצים עצמם ולשמות משתמש אני אשתמש בHASH כדי להגן על המשתמש – המפתחות שלי יהיו אי סימטריות (RSA) אז ההצפנה תקרא הצפנה היברידית.

שימוש במערכת ההפעלה של הDOK בעזרת FAT32

תיאור מודולים בהם נעשה שימוש

* **שרטוט המראה את הזרימה בין המודולים המרכזיים (להקפיד בשרטוט על מקרא אחיד תוך ביצוע אבחנה בין מה נמצא בלקוח ומה בשרת, שימוש ב draw.io) – החלק בו אתם קובעים את תצורת המערכת – חשוב לשבת איתי ולאשר את התצורה לפי שצוללים פנימה**
* סקירת כל המודולים המרכיבים את המערכת וקשרי הגומלין ביניהם:
  1. מודולים שאתם מייבאים – מספיקה שורה אחת המסבירה למה מיועד המודול.
  2. מודולים שלכם - יש לפרט ברמת אובייקטים (שם האובייקט והתכונות שבו) וכן כלל הפעולות שבמודל (כותרות וטענת כניסה ויציאה לכל פעולה – באנגלית כפי שיקראו בתוכנה)

תיאור סביבת הפיתוח

**שפות תכנות:**

PYTHON – שפת הבסיס של הפרויקט היא תשמש לרוב הפרויקט ותהיה אחרית על השרת עמדת התקנה והלקוחות עצמם

SQL – קיים רק בצד שרת ויהיה אחראי על בסיס הנתונים בקוד עצמו

HTML - רק אם ישאר זמן אני ארצה ליצור אתר הרשמה במקום עמדת התקנה שתעשה בדיוק מה שעמדת ההתקנה אמורה לעשות(מסמן את זה כי לא בטוח יקרה)

**כלי פיתוח וכלים הנדרשים לבדיקות:**

PYCHARM – כלי בוא יכתב כל הקוד של המערכת צד שרת צד לקוח

USB – שם מתקיים כל ההצפנה והפיענוח עצמו ועל זה מבוסס כל הפרויקט הרוצה

WIRSHARK – ישמש לבדיקות תקשורת

FILE EXPLORER – ישמש לבדיקה של של מידע מוצפן בטוח הDOK

DB BROWSER – צפיה בבסיס נתונים

תיאור האלגוריתמים המרכזיים בפרויקט

סוגיה - פענוח הקבצים  
תיאור – האם בעת הזדהות נכונה של המשתמש מפענחים לו את כל הקבצים

| מספר | האופציה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | פענוח מידי של כלל הקבצים | כל הקבצים זמינים ללקוח | תהליך ארוך – יכול להתיש את המשתמש וגם לעכב את יציאתו  תהליך מיותר – יתכן ובכלל לא ייגש לקבצים  מסוכן – כל הקבצים חשופים ופגיעים |
| 2 | פענוח שמות הקבצים בלבד ופענוח קובץ רק לאחר בחירתו | תהליך קצר לא חושף קבצים שלא משתמשים בהם | פתיחת קובץ יכולה להתעכב |
|  |  |  |  |

החלופה שנבחרה היא **פענוח שמות הקבצים בלבד ופענוח קובץ רק לאחר בחירתו** מכיוון שבאופציה הזאת יש הכי פחות חסרונות והיא חסכונית ולא מעמיסה על מערכת

סוגיה - שמירת שמם משתמש וסיסמה  
תיאור – איך שומרים את המידע של זיהוי המשתמש?

| מספר | האופציה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | שמירה בתוך הקוד | קל לשימוש | פריץ מאוד למשתמשים בIDA – REVERS ENGENIRING |
| 2 | שמירת הקוד בתוך קובץ מלא מידע | קל לשימוש | חשוף מאוד  גל בן אדם הנמצא ליד הUSB יכול לגשת לזה בקלות  אפשר להצפין את זה בקלות ואז לא נוכל לדעת איך לפענח פעם הבאה |
| 3. | שמירה בתוך הקוד של הHASH של המשתמש | מוגן מאנשים  בן אדם שעושה REVERS ENGENIRING לא יוכל לעשות כלום עם המידע הזה | יותר עבודה לממשק משתמש – צריך לעשות HASH בעצמו כל פעם על מנת לבדוק אם הקוד תקין |

החלופה שנבחרה היא **שמירה בתוך הקוד של הHASH של המשתמש** מכיוון שזה הכי בטוח והחשיבות הכי גדולה של הפרויקט זה האבטחה שלו ואיני רוצה שבן אדם מהצד פשוט יוכל לבוא ולהיכנס לקוד שלי

סוגיה - מעבר המידע לגיבוי בשרת  
תיאור – איך המידע עובר לשרת וכל כמה זמן?

| מספר | האופציה | יתרונות | חסרונות |
| --- | --- | --- | --- |
| 1 | מעבר בסגירת הDOK | מעביר את כל המידע ביחד | בעת סגירת הDOK יקח לך הרבה זמן לצאת  אם מוציא לפני הזמן המידע פשוט לא יעבור ושום דבר לא ישמר בשרת |
| 2 | שמירת שמות הקבצים ומעבר בסוף | מעביר את כל המידע ביחד  מגן מהוצא סתמית של הDOK וישלח פעם הבאה שינסה כי השם שמור | בעת סגירת הDOK יקח הרבה זמן עד יציאה  תופס מקום בDOK |
| 3. | ניטור הקבצים ושליחה לשרת, אם אין שרת שמירת השמות בקובץ מסודר ומוחבא ושליחה בפעם הבאה | אין פחד של הוצאה לא בזמן הנכון  אפשרות עבודה בזמן השליחה | תופס מקום בDOK |

החלופה שנבחרה היא **ניטור הקבצים ושליחה לשרת, אם אין שרת שמירת השמות בקובץ מסודר ומוחבא ושליחה בפעם הבאה** מכיוון שככה הסיכוי לאבד קובץ ואי גיבוי שלו הוא קטן מאוד והוא גם מוגן ככה מאנשים מבחוץ ואני מוכן לאבד קצת מקום למען גיבוי

תיאור מסכי הפרויקט:

* לכל מסך מיועד בפרויקט (שרטוט ידני, בהמשך יצורפו מסכי המערכת):
  1. מה מטרת המסך
  2. שרטוט של המסך (מה כולל)
  3. ולהיכן הוא מוביל (ברמת מסכים)

תיאור פרוטוקול התקשורת

| שם | נשלח מ | מתקבל אצל | פקודה | שדות בהודעה | גודל byte |
| --- | --- | --- | --- | --- | --- |
| רישום | עמדת התקנה | רשת | 00 | 1. שם משתמש  2. סיסמה  3. מייל  4. קוד פקודה | 1. 10  2. 10  3. 30  4. 2 |
| רישום - תגובה | שרת | עמדת התקנה | 00 | 1. קוד פקודה  2. סטטוס | 1. 2  2. 1 |
| התחברות | עמדת לקוח | שרת | 01 | 1. שם משתמש  2. סיסמה  3. קוד פקודה | 1. 10  2. 10  3. 2 |
| התחברות – תגובה | שרת | עמדת לקוח | 01 | 1. קוד פקודה  2. סטטוס | 1. 2  2. 1 |
| עדכון פרטים | עמדת לקוח | שרת | 02 | 1. שם משתמש  2. סיסמה  3. מייל  4. קוד פקודה | 1. 10  2. 10  3. 30  4. 2 |
| עדכון פרטים תגובה | שרת | עמדת התקנה | 02 | 1. קוד פקודה  2. סטטוס | 1. 2  2. 1 |
| גיבויי | DOK | שרת | 03 | 1. |  |

תיאור מבני הנתונים

* פירוט מבני הנתונים (מסד נתונים, קבצים, מקומיים וכו')
* פירוט מאגרי המידע של המערכת (בדומה למסדי נתונים) ברמת שדות, טיפוסים, אורכים וכו'
* מסד נתונים: שם המסד, שם הטבלאות, לכל שדה: שם, טיפוס. לתת דוגמא לערכים אפשריים. להגדיר לכל טבלה במסד – מי המפתח הראשי בה.

סקירת חולשות והאיומים   
למערכת ואת הפתרונות שלכם לאיומים, יש לסקור ע"פ מאפייני המערכת השונים:

שכבת האפליקציה:

* עבודה עם בסיס נתונים sql injection ...
* עבודה עם אתרי web ...
* תהליך ה login אימות ווידוא
* MITM : הצפנה, איזו סוג?
* DOS/DDOS : יכול להיות? איך מתמודדים
* העלאת קבצים : hash

שכבת התעבורה:

* פרוטוקול TCP , לחיצת יד משולשת??
* הצפנה?

הפעלת המערכת

* אילו חולשות קיימות ( 'הזרקת קוד' , sql injection וכו') ואיך טופלו

ועוד...

**פרק ה' - 'שם המערכת' - הקוד**

1. עבור (שעושים קטעי קוד מיוחדים משהו מיוחד, משהו מסובך, משהו בדרך שונה, משהו יפה בעיניכם ויש לכלול בתוכם את קטעי הקוד הרלוונטים לאלגוריתמים המרכזיים שציינתם בפרק הקודם):

* הסבר על היכולת
* הקוד עצמו (כתוב ע"פ כללי התכנות הנכון ומלווה בהערות כנדרש)
* (למצוא לפחות 7 קטעי קוד מעניינים ...)
* רצוי תמיד לבחור את אלא שהסברתם קודם לכן את האלגוריתם שלהם

1. בנוסף יש לצרף את **שאר קוד המערכת** , לא נדרש הסבר לצד כל קוד אבל כן את שם הקובץ.  
   לבצע 'שמירה' של הקוד לקבצי PDF אן העתקה לתוך קבצי word **ולא לצרף כתמונה**

**פרק ו' - 'שם המערכת' – בדיקות ('קופסא לבנה')**

1. פירוט הבדיקות

* פירוט הבדיקות שהופיעו בפרק האפיון ברמת הביטים – לתאר בפירוט מה נדרש לבצע ומה בוצע בפועל. וכמובן על בדיקה צריכה להתבצע יותר מפעם אחת ועד שהיא עוברת

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **שם הבדיקה** | **מטרת הבדיקה** | **מה נדרש לבצע** | **מתי** | **מה בוצע בפועל** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

* יש לתעד כל בדיקה שמבוצעת, חייבים לחזור לפחות פעמיים על כל בדיקה – כל חזרה תתועד בשורה נפרדת עם תוצאות הבדיקה

**פרק ז' - 'שם המערכת' – מדריך למשתמש**

1. מדריך למשתמש הכולל עבור כל תהליך/יכולת במערכת:

* הסבר על התהליך/יכולת
* תרשים זרימה (במקרה של תהליך)
* צילומי מסך מלווים בהסברים בשפה פשוטה

במידה ויש יותר ממשתמש אחד למערכת , לבנות פרקים שונים לדוגמא: מדריך למשתמש קצה, מדריך למנהל מערכת וכו'

**פרק ח' - 'שם המערכת' – רפלקציה**

1. מבט אישי על העבודה ועל תהליך פיתוחה:

* אתגרים שהיו לי בדרך
* אירועים מעניינים שקרו במהלך הפיתוח
* התמודדות עם קשיים, איך מה עשיתי
* הערכת הפתרון לעומת התכנון והמלצות לשיפורו
* תודות חשוב מאוד לחברים, למשפה, למורים...

יש למלא עד עמוד ולא להתבייש בכתיבה – כאן לא המקום לצניעות.

**פרק ט' - 'שם המערכת' – ביבליוגרפיה**

1. רקע תיאורטי
2. ספרות מקצועית ספציפית לנושא העבודה (רשימת ספרים, מאמרים והפנייה לכתובות אתרים המכילים חומר רקע ששימש לפיתוח העבודה)
3. הערה: לא מספיק להתייחס לספרים המיועדים ללימוד שפה או כלים יישומיים ומדריכים למיניהם.
4. קישורים לאתרים באינטנרט – קישורים ל stackOverflow עם הסבר על הנושא שנחקר (לא תחביר...)

**נספח א – החקר שבוצע בפרויקט**