**תיעוד עבודה - פרויקט בנושאים מערכות הגנה לרשת**

**רועי חפץ 207851395**

**יובל אופיר 207288853**

**תיאור כללי ומטרה**

המערכת שפותחה היא מערכת אבטחה לניטור וניהול ניסיונות התחברות לחשבונות ברשת.  
היא פועלת במספר רבדים המשלימים זה את זה כדי להגן על המשתמשים, למנוע פריצות, ולאפשר שליטה מלאה על חסימות:

* זיהוי ניסיונות התחברות כושלים - המערכת עוקבת אחר כל ניסיון התחברות ומבצעת רישום של:
  + כתובת הIP ממנה בוצעה ההתחברות.
  + זמן ותאריך הניסיון.
  + שם המשתמש שאליו ניסו להתחבר.
  + רמת הדמיון בין הסיסמה שהוזנה לבין הסיסמה הנוכחית או סיסמאות קודמות (באמצעות אלגוריתם Levenshtein Distance)
* חסימת כתובות IP חשודות  
  לאחר חריגה ממספר ניסיונות כושלים או זיהוי ניסיון פריצה עם סיסמה בעלת דמיון נמוך במיוחד, המערכת מוסיפה את הIP לרשימה שחורה (blacklist.json) ומונעת התחברויות נוספות ממנו.
* שליחת התראות למשתמשים בזמן אמת  
  כאשר IP נחסם, המערכת שולחת למשתמש הודעת דוא"ל הכוללת:
  + פרטי הIP שנחסם.
  + הסיבה לחסימה (כמות ניסיונות כושלים).
  + קישור חד־פעמי לביטול החסימה דרך שרת הFlask המקומי.
  + המלצות מפורטות לשיפור אבטחת החשבון.
* ניהול IPים מורשים  
  המערכת מאפשרת למשתמש לאשר כתובות IP חדשות שמזוהות כחריגות, כדי למנוע חסימות מיותרות למשתמשים הלגיטימיים עצמם.
* ביטול חסימות באופן מאובטח  
  קישור הביטול במייל מפנה לשרת (unblock\_server.py), שמסיר את הIP מהרשימה השחורה ומאפס את מונה ניסיונות הכניסה הכושלים, רק אם הטוקן החד פעמי שנשלח במייל תקף.

**האם הכלי עונה על המטרה?**  
בהתאם למטרות שהוגדרו בתוכנית העבודה, הכלי עומד ביעדים באופן מלא:

* מניעת חסימות מיותרות - בזכות ניתוח איכות הסיסמאות ושימוש ברמת דמיון ו"היסטוריה" של סיסמאות המשתמש כסינון נוסף.
* שקיפות ושליטה למשתמש - המשתמש מקבל התראה בזמן אמת, עם כל המידע הדרוש, במקרה ונעשה ניסיון פריצה לחשבון שלו, במקרה וזוהי טעות יש למשתמש אפשרות לבטל את החסימה על הIP המדובר.
* תיעוד מלא - כל ניסיון התחברות, מוצלח או כושל, נרשם בלוג עם פרטי הIP שלו, הזמן, שם המשתמש, רמת הדמיון של הסיסמה שהוזנה ועל פי רמה זו גם "רמת הסיכון" של ניסיון הפריצה. באמצעות הרשומות הללו נוכל לבצע ניתוחים והצעות ייעול ושיפור אבטחה למערכת שלנו.

**בעיות שנתקלנו בה :**

* בתחילה, ניסיונות שליחת קישור לביטול חסימה (unblock) ישירות מתוך קוד המערכת לא עבדו בצורה יציבה, בין היתר בגלל הצורך בכתובת יעד תקינה לטיפול בבקשה, מגבלות שירותי הדוא"ל, וחוסר מנגנון לקליטת בקשות שחרור מצד הלקוח.
* לאחר ניסיון התחברות כושל, התוכנית המשיכה לשלב הבא במקום להחזיר את המשתמש לתפריט הראשי.
* המשתמשים לא יכלו לצאת מהמערכת בכל נקודת זמן, אלא רק בשלבים מוגדרים מראש.
* ניסיונות פריצה זוהו, אך ההודעה שנשלחה במייל הייתה קצרה ולא מספקת.
* היה צורך לנהל ניסיונות כושלים לפי כתובת IP כדי למנוע חסימה גורפת לכלל המשתמשים.

**חוזקות המערכת -**

1. שילוב של מספר מנגנוני אבטחה  
   המערכת לא מסתמכת על הזנת סיסמה בלבד ללא הגבלה – היא כוללת חסימת IPבמקרה של מספר שגיאות, בדיקת דמיון סיסמאות, בדיקת סיסמאות ישנות.
2. ניהול משתמשים מובנה  
   יש ממשק לרישום משתמשים חדשים, שינוי סיסמאות, ושמירת היסטוריית סיסמאות.
3. מערכת חסימת IP חכמה  
   ספירה של ניסיונות כושלים להזנת סיסמה לפי IP וחסימה אוטומטית אחרי מספר ניסיונות – מונעת מתקפות brute force בסיסיות.
4. שליחת התראות במייל בזמן אמת  
   המערכת שולחת מיילים עם מידע מפורט כגון: כתובת , IP שם המשתמש, קישור לשחרור נעילה וכן טיפים לשיפור האבטחה – מאפשר למשתמש תגובה מיידית.
5. שרת ניהול (Flask) לביטול חסימות  
   ממשק אינטרנטי פשוט שמאפשר למשתמש לשחרר חסימה או לאשר/לדחות IP – מעלה את חוויית השימוש ומדמה פתרון אמיתי.
6. יכולת זיהוי ניסיונות התחברות “חכמים”  
   שימוש ב־Levenshtein distance לזיהוי סיסמאות שדומות מאוד לסיסמא האמיתית – מוסיף שכבת הגנה מפני חסימות לא נחוצות .
7. שקיפות וניטור  
   כל הפעולות נרשמות לקובץ לוג עם מידע מלא על הניסיון – מקל על debug , מעקב וחקירה.

**חולשות המערכת**

1. תלות בחיבור לאינטרנט ובשרת מייל חיצוני (Gmail) לצורך שליחת מייל.
2. ניהול נתונים בקבצי JSON   
   מתאים להדגמה, אבל לא למערכת אמיתית – אין נעילות כתיבה, אין תמיכה בעומס גבוה, ואין אינדוקס לחיפושים.
3. חוסר הצפנה בתקשורת פנימית  
   ה־Flask server והקישורים (Unblock) פועלים ב־HTTP רגיל, לא HTTPS – פגיע להאזנה ושינוי תעבורה.
4. חוסר הגנה מפני CSRF / Link Hijacking   
   הקישורים הנשלחים במייל אינם מוגנים באמצעים נוספים מעבר ל־token.

**תיאור עבודת הכלי –**

**שלבי השימוש במערכת**

* 1. **הפעלת שרת ביטול חסימות (Unblock Server)**

בחלון טרמינל ראשון, הפעל: python unblock\_server.py

A screenshot of a computer

AI-generated content may be incorrect.שרת זה ירוץ על http://localhost:8000 ויטפל בבקשות לביטול חסימת כתובת IP .

* 1. **הפעלת המערכת**

בחלון טרמינל שני, הפעל: יש להריץ את הקובץ python login\_system.py

עם הפעלת התוכנית, תוצג למשתמש הודעת פתיחה עם אפשרויות:

* + - 1 - כניסה לחשבון (Login)
    - 2- רישום משתמש חדש (Register)
    - - Exit בכל שלב ניתן לצאת מהמערכת

A screenshot of a computer

AI-generated content may be incorrect.

* 1. **רישום משתמש חדש (Register)**
* בחר באפשרות 2
* הזן שם משתמש ייחודי (אם כבר קיים, תתבקש לבחור אחר). לדוגמה: "testuser"
* הזן סיסמה לדוגמה: "Password!123"
* כתובת מייל: כתובת המייל שלך (תקבל אליה את ההודעות)
* המשתמש יתווסף למסד הנתונים (users\_db.json) .
* ניתנת אפשרות לצאת מהמערכת בכל שלב ע"י הקלדת המילה **exit**.

A screenshot of a computer

AI-generated content may be incorrect.

* 1. **כניסה למערכת (Login)**
* בחר באפשרות 1
* הזן את שם המשתמש שלך.
* המערכת תבדוק:
  + אם שם המשתמש קיים.
  + אם כתובת ה־IP שלך חסומה.
* הזן את הסיסמה:
  + A screenshot of a computer

    AI-generated content may be incorrect.במקרה של הצלחה – כניסה מאושרת ותינתן אפשרות לשינוי סיסמה בהקשה נוספת של סיסמה נכונה.
  + A screenshot of a computer

    AI-generated content may be incorrect.במקרה של סיסמה ישנה – תקבל הודעה לבחור סיסמה עדכנית אך לא ייספר לך ניסיון שגוי.
  + במקרה של סיסמה שגויה:
    - A screenshot of a computer program

      AI-generated content may be incorrect.המערכת בודקת דמיון לסיסמה הנוכחית ומסווגת את רמת הסיכון לפי שימוש ב Levenshtein distance . לדוגמה:

המרחק הלבנשטייני הוא מדד לכמה פעולות "עריכה" נדרשות כדי להפוך מחרוזת אחת לשנייה.  
פעולות עריכה כוללות:

הוספה (insertion) – להוסיף תו חדש.

מחיקה (deletion) – למחוק תו.

החלפה (substitution) – להחליף תו אחד בתו אחר.

הסיסמה המקורית הינה : Password!1234

ולכן החישוב של Levenshtein distance הינו מתבצע כך :

כדי להגיע מהסיסמה המקורית לסיסמה המוקשת:

1. הוספה של התו '5' – פעולה אחת.
2. הוספה של התו '6' – פעולה שנייה.

אין כאן מחיקות או החלפות, רק הוספות לכן Levenshtein distance = 2.

*והינו גדול מהסף 0.8 שהגדרנו למרחק תקין, לכן ניסיון זה לא ייספר.*

* + - לאחר מספר ניסיונות כושלים (MAX\_ATTEMPTS) כתובת ה־ IP תחסם אוטומטית ויישלח אימייל התראה.
    - הזן שלוש פעמים סיסמה שגויה כדי לגרום לחסימת IP .
  1. **חסימת IP ושליחת התראה**
* בעת חריגה ממספר הניסיונות המותר, כתובת ה־IP תתווסף ל blacklist.json.
* יישלח אימייל לבעל החשבון עם:
  + - * התראה על ניסיון חדירה.
      * קישור לבקשת הסרת החסימה (Unblock).
      * A screenshot of a computer program

        AI-generated content may be incorrect.המלצות לשיפור אבטחת החשבון.
  1. A white background with black text

     AI-generated content may be incorrect.**בקשת הסרת חסימה**
* הקישור באימייל מוביל לשרת ה־unblock\_server.py
* המשתמש יוכל לאשר שהניסיון הכושל היה שלו ולבקש הסרת חסימה.
* בקשות נשמרות ב־unblock\_requests.json .
* מנהל המערכת יכול לאשר או לדחות את הבקשה.
* לאחר לחיצה על "Unblock My IP" תוסר החסימה וייפתח חלון שיודיע על כך ובעת ניסיון התחברות חדש לא תתקבל דחייה אוטומטית.



**רכיבי המערכת:**

* + 1. ממשק ההתחברות login\_handler.py:
* אחראי על תהליך ההתחברות בפועל.
* בודק האם הIP חסום (blacklist.json).
* בודק האם המשתמש קיים במערכת.
* משווה את הסיסמה שהוזנה לסיסמה הנוכחית או הישנות.
* מחשב דמיון סיסמאות באמצעות password\_similarity כדי להעריך רמת סיכון

(low risk, suspicious, likely attack).

* מנהל ספירת ניסיונות כושלים (attempts.json).
* חוסם IP ים ומפעיל שליחת מייל התראה (send\_alert\_email) במקרה הצורך.
  + 1. ניהול משתמשים user\_manager.py:
* טוען ושומר את מסד המשתמשים (users\_db.json).
* מאפשר הרשמה עם שם משתמש, סיסמה ואימייל.
* מאפשר שינוי סיסמה - כולל שמירה של שתי הסיסמאות האחרונות ברשימת old\_passwords.
* מוודא שלא ניתן לרשום שם שכבר קיים.
  + 1. login\_system.py – נקודת הכניסה הראשית למערכת בממשק CLI:
* טוען את מסד הנתונים של המשתמשים ומציג תפריט עם שתי אפשרויות: התחברות או הרשמה.
* תהליך ההתחברות מתבצע דרך הlogin\_handler.py, במידה וההתחברות מצליחה, המשתמש יכול לבחור לשנות את סיסמתו (כולל שמירה של סיסמאות ישנות).
* תהליך ההרשמה כולל בדיקת זמינות שם משתמש, קבלת סיסמה ואימייל, ושמירת הנתונים ב־users\_db.json.
* המערכת כוללת תמיכה בפקודה exit בכל שלב, המאפשרת יציאה מיידית.
  1. שרת ביטול חסימה unblock\_server.py:
* שרת Flask מקומי עם מספר נקודות קצה (endpoints):
  + /unblock - ביטול חסימת IP לפי טוקן חד־פעמי (מנקה גם את ניסיונות הכניסה הכושלים).
  + /approve\_ip - הוספת IP חדש לרשימת ה IPים המורשים של המשתמש (authorized\_ips.json).
  + /reject\_ip - חסימת IP חשוד ושליחת הודעת אזהרה עם טיפים לאבטחה.
* מנהל את הקבצים:
  + blacklist.json
  + unblock\_requests.json
  + attempts.json
  + authorized\_ips.json
  1. פונקציות עזר ואבטחה Utils.py:
* ניהול רשימות:load\_blacklist, save\_blacklist, load\_authorized\_ips, save\_authorized\_ips.
* לוגים: log\_failed\_attempt - שמירת כל ניסיון כושל בlog.txt עם תאריך, IP, שם משתמש, סיסמה שנוסתה והסיבה.
* שליחת מיילים:
  + send\_alert\_email - מייל התראה עם קישור לביטול חסימה + המלצות אבטחה.
  + send\_ip\_verification\_email – בקשת אישור IP חדש (approve/ reject).
  + תמיכה בגרסת טקסט ו-HTML.
* זיהוי דמיון סיסמאות:
  + levhenstein\_distance - חישוב מרחק עריכה.
  + Password\_similarity - חישוב אחוז הדמיון בין סיסמאות.
* ניהול טוקנים:

generate\_unblock\_token - יצירת טוקן חד פעמי לביטול חסימה.