**מטלת תכנות 1 -Service Monitor**

**מגיש: זוהר זריהן**

**ת"ז: 316405505**

**מבנה התוכנית:**

התוכנית שלי בנויה בשני קבצים, קובץ אחד הוא קובץ שמתעסק באבטחת המוניטור(הסבר למטה), והקובץ השני הוא המוניטור עצמו.

**המוניטור:** בקובץ יש קודם כל פונקציות אשר מטפלות בדברים הבאים:

1. פונקציה initfiles – אשר דואגת לאתחל את הקבצים statuslog,servicelist. במידה וקבצים אלה קיימים כבר משימוש קודם הם ימחקו ויוצרו מחדש, במידה ולא קיימים ניצור אותם.
2. פונקציה checking date - מקבלת תאריך, תפקידה לבדוק את פורמט התאריך שמתקבל ע"י המשתמש במצב ידני ומחזירה הודעת שגיאה במידה והפורמט אינו טוב.
3. פונקציה pullbydate – תפקידה לקבל 2 תאריכים ולהחזיר רשימה של כל הארועים שהתרחשו בין התאריכים האלה, לצורך השוואה. (נעשה שימוש במצב ידני)
4. פונקציה win\_services – כותבת לתוך הקובץ service list את התאריך הנוכחי בתחילת כל פעולה ולאחר מכן את השירותים הקיימים ומצבם. מותאמת למערכת הפעלה windows (בעזרת ספריית psutil )
5. פונקציה linux\_services- פועלת כמו הפונקציה win\_services רק מותאמת למערכת הפעלה לינוקס, ההתאמה נעשית בעזרת ספריית psutil.
6. פונקציה difrence – מקבלת 2 מילונים המייצגים את כל השירותים שפעלו בתאריך מסויים כל אחד. הפונקציה מבצעת השוואה בין 2 מילונים אלה ובמידה ויש שינוי מעדכנת את המשתמש וכותבת זאת לקובץ statuslog.

**התפריט וההסבר כיצד להשתמש בפונקציה מוצג למשתמש כאשר הוא מריץ את הקובץ ע"י python MainMonitor.py .**

**ספריות שנעשה בהן שימוש:**

1. Time – על מנת לכתוב את התאריך והשעה שבו נכתב תיעוד השירותים הפועלים ובנוסף על מנת להשוואות בין תאריכים שמתקבלים ע"י המשתמש ולבדוק את הפורמט שלהם.
2. Psutil- הספרייה העיקרית והשימושית ביותר בתוכנית- ספרייה זו מאפשרת לקבל את רשימת התהליכים שרצים, לעבור עליהם באיטרציות, ולקבל פרטים על התהליכים כמו שם התהליך ומצבו.
3. Os – ספרייה זו משמשת בתוכנית על מנת לבדוק האם קבצי הלוג קיימים ולאתחל אותם.
4. Platform- נעשה שימוש בספרייה זו על מנת לזהות את מערכת ההפעלה של המשתמש ולשלוח לפונקציות המתאימות בהתאם.
5. Subprocess – נעשה שימוש בפונקציה של הספרייה הנקראת checkoutput פונקציה זו מריצה לנו במערכת הפעלה לינוקס את הפקודה בטרמינל שמחזירה את רשימת כל התהליכים הפועלים ומצבם, ומחזירה לנו בפייתון את הפלט כמחרוזת של בייטים.
6. Sys – ספרייה זו משמשת אותנו לקבל קלט מהמשתמש בהרצת המוניטור ועל פי קלט זה להפעיל את התוכנית בהתאם (מצב מוניטור או מצב ידני, תאריכים להשוואה וכו').

**מבני נתונים שנעשה בהם שימוש:**

מילון - במבנה זה נעשה שימוש בכדי לאחסן רשומות של התהליכים שרצים כאשר הkey הוא שם התהליך והvalue הוא מצב התהליך. שימוש נוסף במילון בקוד הוא בכדי לבצע השוואה בין רשומות של 2 תאריכים שונים אשר מאוחסנות כל רשומה במילון אחר (השוואה בין הkeys ).

**אבטחת המוניטור:**

סכנת האבטחה הגדולה היא כאשר תוקף ירצה לפגוע בקבצי הלוג שלנו או לנסות לעצור\להפעיל תהליכים מסוימים. פתרון ראשוני לכך הוא הפעלת סיסמה (רצוי חזקה) שתידרש מכל משתמש על מנת לבצע תהליכים אלו.  
פתרון נוסף בו השתמשתי בקוד הוא הספרייה "watchdog" של פייתון. השימוש שעשיתי בספרייה הוא להריץ במקביל למוניטור קובץ נוסף אשר מצורף "monitorsaver.py" כאשר שני הקבצים נמצאים באותו המיקום יש להפעיל את "monitorsaver.py" ולציין את שם הקבצים עליהם אנו רוצים להשגיח למשל באופן הבא: "python monitorsaver.py serviceList.log" וכעת הקובץ "serviceList.log" יהיה מוגן, לא ניתן יהיה למחוק אותו או לשנות אותו ולמשתמש תינתן התראה אם יהיה ניסיון לעשות כך.