בס"ד

מעבדת סייבר - הגנה
מטלת תכנות Service Monitor

מגישה:

רז אלבז ת"ז: 207276775

בס"ד

Monitor

מטלת תכנות Service Monitor

שפת כתיבה:

Python 3.8/3.9

מבוא:

במטלה זו נפתח כלי שיינטר לנו את השירותים($services$) הרצים במערכת, וידווח על שינויים שיכולים להיות קריטיים עבורנו כאנשי SOC בדומה לכלי $Zenoss$ שהכרנו - המנטר לנו שירותים. את הכלי נפתח בשפת Python.

פירוט של המחלקות:

main

במחלקה זו מתבצעת הקליטה מהמשתמש של מצב המוניטור שירצה ידני/לא, כמו כן ניתנת אפשרות ללחוץ על '0' ולצאת מהתוכנית.

Manual

במחלקה זו מומש המצב הידני של המוניטור; במצב זה נרצה להשתמש בקובץ $serviceList$ על מנת לטעון 2 דגימות מטווחי זמן שונים ולבצע השוואה. התכנית תקבל תאריך ושעה ל2 אירועים, תטען מהקובץ את 2 הדגימות, ותציג שינויים בדומה למצב ה$monitor$.

Monitor

במחלקה זו מומש מצב $Monitor$; עבור X זמן שהמשתמש קובע, התוכנית דוגמת כל X זמן את כל השירותים הרצים במחשב, ומציגה האם נצפה שינוי מהדגימה הקודמת. כלומר האם יש $service$ שכבר אינו רץ, או האם יש $service$ חדש שרץ במערכת. על כל שינוי שהתקיים יש להתריע למשתמש בממשק.

Services

במחלקה זו בוצעו הדפסות למסך התפריטים בטרמינל, נקראו הפונקציות לכתיבת קבצים ועודכנה רשימת ה$processes$ הרצים והלא.

Process

במחלקה זו בדקתי את מערכת ההפעלה של אותו מחשב שמריץ. המוניטור שלנו

תומך במערכות ההפעלה Linux / Windows ולכן ביצענו את המימוש בהתאם.
בנוסף, במחלקה זו נוצרו ה$dictionaries$ לתהליכים בפורמט הבא לכל תהליך:
process pid: process name

Write_to_files

במחלקה זו התבצעה הכתיבה לשני הקבצים כפי שהתבקשנו:

$serviceList.txt$– לקובץ זה נדפיס את דגימות הסרביסים שכרגע רצים (בכל פעם את הדגימה האחרונה). בכל פעם קובץ זה יתמלא בכל דגימה שלנו, וישמור את כל הדגימות שעשינו במהלך מצב המוניטור לפי תאריך ושעה.
$StatusLog.txt$ – קובץ לוג זה הוא למטרת מעקב. נדפיס לקובץ כל שינוי שהוצג לנו במצב המוניטור. לדוגמה $service$ חדש שנוצר, $service$ שהפסיק לעבוד וכו'.
בנוסף, התבצעה במחלקה זו ההגנה מפני ההאקרים והצפנת רשימת התהליכים לאבטחה באמצעות השימוש במבנה נתונים HASH וב :

```
#The MD5 is a hash algorithm to turn inputs into a fixed 128-bit (16
bytes) length of the hash value
hash=hashlib.md5()
```

דרכים להתגונן מפני האקרים:

השימוש בפונקציה hashlib נועדה בכדי להצפין את רשימת ה-Processes , כמו כן התבצעה בדיקה שלא השתנה דבר בקבצים הקיימים, במידה וכן תישלח הודעה מתאימה והתוכנית תיסגר. תמונות מהתוכנית מהקובץ Write_to_files.py :

```
def write_serviceList(Add_to_file):
    """
    This function is designed to check that no change has been made by a stranger, in case the user receives an error message and the program ends.
    If everything is correct, we will write the changes to the file
    ***The way to protect against hackers***
    """
    file_path = 'TXT_files/serviceList.txt'
    try:
        get_hash_file(file_path, serviceList)
        write_Status_Log(file_path, Add_to_file)
    except ValueError:
        print("A hacker tried to sabotage this file and modify it")
        exit(0)


def write_statusLog(Add_to_file):
    """
    This function is designed to check that no change has been made by a stranger, in case the user receives an error message and the program ends.
    If everything is correct, we will write the changes to the file
    ***The way to protect against hackers***
    """
    file_path = 'TXT_files/statusLog.txt'
    try:
        get_hash_file(file_path, statusLog)
        write_Status_Log(file_path, Add_to_file)
    except ValueError:
        print("A hacker tried to sabotage this file and modify it")
        exit(0)
```

בס"ד

שימוש בספריות חשובת במטלה:

os

מודול מערכת ההפעלה ב-Python מספק פונקציות ליצירה והסרה של ספרייה (תיקיה), שליפת תוכנה, שינוי וזיהוי הספרייה הנוכחית וכו'.

מטרת שימוש: בתוכנית היה עדכון בכל פעם שיש שינויים בProcesses לכן התבצעה כתיבה לקובץ זמני שאליו נוספו השינויים על המידע הקיים בקובץ file . לכן, היה צורך בשימוש בOS בכדי לאפשר את החלופה של השם בין השם החלופי למקורי ובכך קיבלנו קובץ מעודכן בתוספת השיניים. בנוסף השימוש ב-OS אפשר לנו למחוק את הקובץ file שממנו העתקנו את המידע.

path ; os

מודול path הוא מודול בשימוש נרחב מאוד, שימושי בעת עיבוד קבצים ממקומות שונים במערכת. הוא משמש למטרות שונות כגון למיזוג, נרמול ושליפה של שמות נתיבים ב-python.

מטרת שימוש: בכדי לבדוק שהנתיב של ה-file קיים במערכת.

hashlib

מודול Python hashlib הוא ממשק לגיבוש הודעות בקלות. זה מכיל שיטות רבות שיטפלו ב-hashing של כל הודעה גולמית בפורמט מוצפן. מטרת הליבה של מודול זה היא להשתמש בפונקציית hash על מחרוזת, ולהצפין אותה כך שקשה מאוד לפענח אותה.

מטרת שימוש: להצפין את רשימת ה-Processes .

json

הצורה המלאה של JSON היא סימון אובייקט JavaScript. המשמעות היא שקובץ סקריפט (ניתן להרצה) שעשוי מטקסט בשפת תכנות, משמש לאחסון והעברת הנתונים. Python תומך ב-JSON דרך חבילה מובנית בשם json.

מטרת שימוש: לקרוא ולכתוב מהקבצים שהתבקשנו לכתוב אליהם: serviceList.txt ו- statusLog.txt.

datetime, timedelta ;datetime

אובייקט **Timedelta** מייצג משך זמן, ההבדל בין שני תאריכים או זמנים. כל הארגומנטים הם אופציונליים וברירת המחדל היא 0. טיעונים עשויים להיות מספרים שלמים או צפים, ויכולים להיות חיוביים או שליליים.

מודול **Python Datetime** מספק classes לעבודה עם תאריך ושעה. שיעורים אלה מספקים מספר פונקציות להתמודדות עם תאריכים, שעות ומרווחי זמן.

מטרת שימוש: לעדכן את הרשימות של הקבצים באמצעות השעות וחיסור הרשימות, ובכך לדעת אילו Processes רצים כעת במערכת ואילו כבר לא.

threading

Threading ב-python משמש להפעלת שרשורים מרובים (משימות, קריאות לפונקציות) בו-זמנית. שימו לב שזה לא אומר שהם מבוצעים על מעבדים שונים.
מטרת שימוש: להפעיל את הפונקציה שרצה תמיד ומעדכנת את הקבצים שהתבקשנו, ובנוסף כותבת במקביל לשני הקבצים (serviceList.txt ו-statusLog.txt)את העדכונים המתאימים להם.

time

מודול הזמן של Python מספק דרכים רבות לייצוג זמן בקוד, כגון אובייקטים, מספרים ומחרוזות.

מטרת שימוש: הוצאת הזמנים של המערכת בכדי להדפיס למסך הטרמינל ולקבצים (serviceList.txt ו- statusLog.txt) את השעות המדויקות של ה-Processes .

system ;platform

Python מגדירה פלטפורמת מודול מובנית המספקת מידע מערכת. מודול הפלטפורמה משמש כדי לאחזר מידע רב אפשרי על הפלטפורמה שבה התוכנית מבוצעת כעת.

מטרת שימוש: בכדי לבדוק מהי מערכת ההפעלה של המחשב אשר יפעיל את התוכנית ובהתאם לשלוח לתוכנית את רשימת ה- Processes היות ובכל מערכת הפעלה אופן השליפה שונה .

subprocess

Subprocess ב- Python הוא מודול המשמש להפעלת קודים ויישומים חדשים על ידי יצירת תהליכים חדשים. זה מאפשר לנו להפעיל יישומים חדשים ישירות מתוכנית Python שאנחנו כותבים כעת.

מטרת שימוש: באמצעות ספריה זו שלפנו את רשימת ה- Processes במערת ההפעלה Linux.

psutil

psutil (כלי עזר לתהליכים ומערכת) היא ספרייה חוצת פלטפורמות לאחזור מידע על תהליכים רצים וניצול מערכת (CPU, זיכרון, דיסקים, רשת, חיישנים) ב-Python. זה שימושי בעיקר לניטור מערכת, פרופיל והגבלת משאבי תהליכים וניהול תהליכים רצים.

מטרת שימוש: באמצעות ספריה זו שלפנו את רשימת ה- Processes במערת ההפעלה Windows.

הרצה:

נכנסים לתוכנית main.py ומריצים אותה. לאחר מכן, בחלון הטרמינל ייפתח התפריט של המוניטור עם האפשרויות הבאות:

- Press 1 for Monitor mode
- Press 2 for Manual mode You can click 0 whenever you want to finish the program

הכנס את הבחירה שלך בהתאם למצב המוניטור שתרצה לבחור.
שימו ♥ ניתן לעבור ממצב Monitor למצב ידני Manual בכל זמן שתרצה על ידי הכנסת התו '2' .

שימו ♥ מפאת הרצון לממש מוניטור מדויק ביותר במצב הידני הכניסו את השעות המדויקות, במידה ולא יודפס לכם שאין שינוי חדש.

בס"ד

תמונות מההרצה:

הפעלת התוכנית main.py ופתיחה של תפריט המוניטור:

```
C:\Users\97252\AppData\Local\Programs\Python\Python39\python.exe C:/Users/97252/PycharmProjects/ServiceMonitor/main.py
Service Monitor
****************************************
*********** Monitor menu: ************
  * Press 1 for Monitor mode
  * Press 2 for Manual mode
****************************************
****************************************
You can click 0 whenever you want to finish the program
```

בחירה של מצב Monitor:

```
main
C:\Users\97252\AppData\Local\Programs\Python\Python39\python.exe C:/Users/97252/PycharmProjects/ServiceMonitor/main.py
Service Monitor
****************************************
*********** Monitor menu: ************
  * Press 1 for Monitor mode
  * Press 2 for Manual mode
****************************************
****************************************
You can click 0 whenever you want to finish the program
1
Please insert the interval that we will update you in a new process
1
You can click 2 whenever you want to change the monitor mode to the Manual mode

2022-03-27 19:54:05 {'NetSetupSvc': 'running'} running
2022-03-27 19:54:26 {'BcastDVRUserService_2c5af3a': 'start_pending', 'ClipSVC': 'running'} running
2022-03-27 19:54:28 {'XblAuthManager': 'running'} running
2022-03-27 19:55:04 {'CaptureService_2c5af3a': 'running'} running
2022-03-27 19:55:06 {'NetSetupSvc': 'running'} has stopped
```

בחירה של מצב Manual(ידני):

הקלדת השעות והתאריכים שנרצה לדגימה לפי הפורמט:

```
main
C:\Users\97252\AppData\Local\Programs\Python\Python39\python.exe C:/Users/97252/PycharmProjects/ServiceMonitor/main.py
Service Monitor
****************************************
*********** Monitor menu: ************
  * Press 1 for Monitor mode
  * Press 2 for Manual mode
****************************************
****************************************
You can click 0 whenever you want to finish the program
2
Type the time when you want to start checking the processes in this way: year-month-day hour:minutes:seconds
2022-03-27 19:37:44
Type the time when you want to finish reviewing the processes in this way: year-month-day hour:minutes:seconds
2022-03-27 19:59:49
The nonexistent process:
{'2022-03-27 19:59:49': {'Appinfo': 'running', 'AppXSvc': 'running', 'AudioEndpointBuilder': 'running', 'Audiosrv': 'running', 'BFE': 'running', 'BrokerInfrastructure': 'running'
The new process:
{'2022-03-27 19:37:44': {'Appinfo': 'running', 'AppXSvc': 'running', 'AudioEndpointBuilder': 'running', 'Audiosrv': 'running', 'BFE': 'running', 'BrokerInfrastructure': 'running'

Process finished with exit code 0
```

בס"ד

עדכון של הקובץ serviceList.txt:

serviceList.txt

```
{"2022-03-27 19:59:49": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running",
{"2022-03-27 19:59:48": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:47": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:46": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:45": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:44": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:43": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:42": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:40": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:39": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:38": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:37": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:36": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:35": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:34": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:33": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:31": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:30": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:29": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:28": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:27": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:26": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:25": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:24": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:23": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
{"2022-03-27 19:59:22": {"Appinfo": "running", "AppXSvc": "running", "AudioEndpointBuilder": "running", "Audiosrv": "running", "BFE": "running", "BrokerInfra
```

עדכון של הקובץ statusLog.txt:

statusLog.txt

```
2022-03-27 19:59:30 {'XblAuthManager': 'running'} has stopped
2022-03-27 19:58:03 {'wlidsvc': 'running'} running
2022-03-27 19:56:00 {'wlidsvc': 'running'} has stopped
2022-03-27 19:55:06 {'NetSetupSvc': 'running'} has stopped
2022-03-27 19:55:04 {'CaptureService_2c5af3a': 'running'} running
2022-03-27 19:54:28 {'XblAuthManager': 'running'} running
2022-03-27 19:54:26 {'BcastDVRUserService_2c5af3a': 'start_pending', 'ClipSVC': 'running'} running
2022-03-27 19:54:05 {'NetSetupSvc': 'running'} running
2022-03-27 19:36:14 {'wlidsvc': 'running'} has stopped
2022-03-27 19:35:11 {'WerSvc': 'running'} has stopped
2022-03-27 19:35:10 {'NetSetupSvc': 'running'} has stopped
2022-03-27 19:33:48 {'NetSetupSvc': 'running'} running
2022-03-27 19:33:11 {'wlidsvc': 'running'} running
2022-03-27 19:33:10 {'WerSvc': 'running', 'WdiSystemHost': 'running'} running
2022-03-27 19:32:58 {'AppXSvc': 'running', 'XblAuthManager': 'running'} running
2022-03-27 19:32:56 {'ClipSVC': 'running', 'BcastDVRUserService_2c5af3a': 'start_pending'} running
2022-03-27 19:32:50 {'AppXSvc': 'running'} has stopped
2022-03-27 19:32:27 {'NetSetupSvc': 'running'} running
```

בס"ד

כיבוי של המוניטור:

```
Service Monitor
**************************************
*********** Monitor menu: ************
  * Press 1 for Monitor mode
  * Press 2 for Manual mode
**************************************
**************************************
You can click 0 whenever you want to finish the program
0
********* Shutdown of monitor ************

Process finished with exit code 0
```

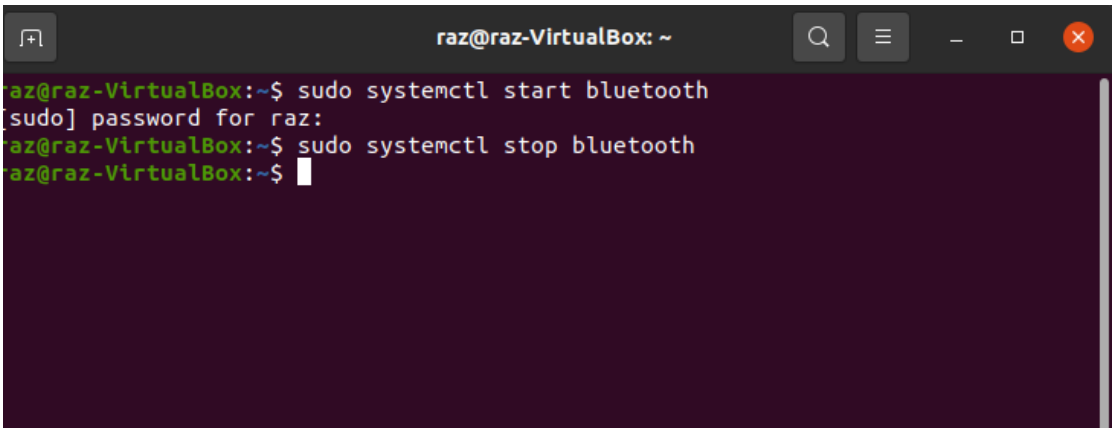
מעבר יזום על ידי לחיצת '2' ממצב מוניטור למצב ידני והמשך מצב מוניטור:

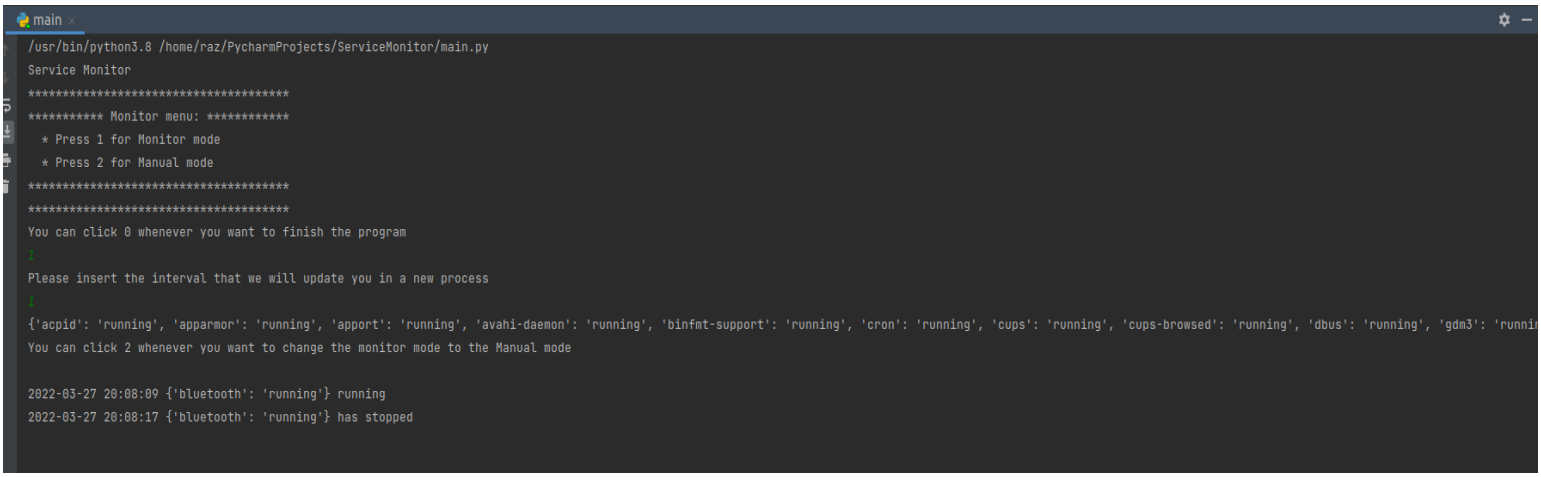
```
main
C:\Users\97252\AppData\Local\Programs\Python\Python39\python.exe C:/Users/97252/PycharmProjects/ServiceMonitor/main.py
Service Monitor
**************************************
*********** Monitor menu: ************
  * Press 1 for Monitor mode
  * Press 2 for Manual mode
**************************************
**************************************
You can click 0 whenever you want to finish the program
1
Please insert the interval that we will update you in a new process
1
You can click 2 whenever you want to change the monitor mode to the Manual mode

2
Type the time when you want to start checking the processes in this way: year-month-day hour:minutes:seconds
2022-03-27 19:55:46
Type the time when you want to finish reviewing the processes in this way: year-month-day hour:minutes:seconds
2022-03-27 19:57:23
The nonexistent process:
{'2022-03-27 19:57:23': {'Appinfo': 'running', 'AppXSvc': 'running', 'AudioEndpointBuilder': 'running', 'Audiosrv': 'running', 'BFE': 'running', 'BrokerInfrastructure': 'running
The new process:
{'2022-03-27 19:55:46': {'Appinfo': 'running', 'AppXSvc': 'running', 'AudioEndpointBuilder': 'running', 'Audiosrv': 'running', 'BFE': 'running', 'BrokerInfrastructure': 'running

2022-03-27 19:58:03 {'wlidsvc': 'running'} running
```

בס"ד

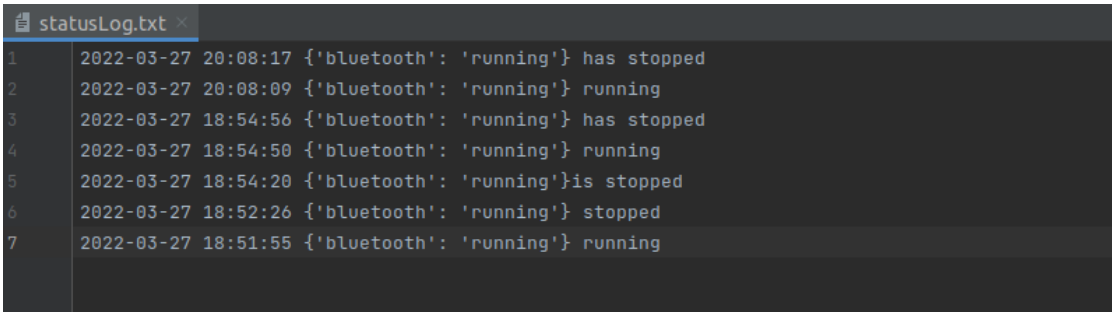
מוניטור במערכת ההפעלה Linux:

כל ההדפסות למסך וההפעלה זהה למעט המצב מוניטור אותו הייתי צריכה להפעיל ולכבות פרוססים באמצעות הטרמינל כך:





העדכון בקובץ statusLog.txt:



בס"ד

דוגמה נוספת:

```
main
Service Monitor
**************************************
*********** Monitor menu: ************
  * Press 1 for Monitor mode
  * Press 2 for Manual mode
**************************************
**************************************
You can click 0 whenever you want to finish the program

Please insert the interval that we will update you in a new process

You can click 2 whenever you want to change the monitor mode to the Manual mode

2022-03-30 10:02:40 {'uuidd': 'running'} running
2022-03-30 10:02:43 {'uuidd': 'running'} has stopped
2022-03-30 10:03:49 {'uuidd': 'running'} running
2022-03-30 10:03:53 {'uuidd': 'running'} has stopped
```

```
raz@raz-VirtualBox: ~
raz@raz-VirtualBox:~$ sudo systemctl start uuidd
raz@raz-VirtualBox:~$ sudo systemctl stop uuidd
Warning: Stopping uuidd.service, but it can still be activated by:
  uuidd.socket
raz@raz-VirtualBox:~$
```

```
statusLog.txt   main.py   Services.py
1  2022-03-30 10:03:53 {'uuidd': 'running'} has stopped
2  2022-03-30 10:03:49 {'uuidd': 'running'} running
```