Prodigy InfoTech Cyber Security Report

Week 4: Simple Keylogger

Full Name: Sairaj Turalkar

Date: 20/07/2024

1. Introduction

In the fourth week of my cybersecurity internship at Prodify Infotech, I was assigned the task of creating a basic keylogger program. The keylogger's purpose is to record and log keystrokes, focusing on capturing keypresses and saving them to a file. This project emphasized the importance of ethical considerations and obtaining necessary permissions due to the nature of keyloggers.

2. Objective

The primary objective of this task was to develop a basic keylogger tool that captures and logs keystrokes. The keylogger should:

- Record all keystrokes, including both standard and special keys.
- Save the recorded keystrokes to a log file for analysis.
- Be implemented with a strong focus on ethical considerations, ensuring that the use of the keylogger is transparent and legal.

3. Methodology

3.1. Keylogger Implementation

The keylogger was implemented in Python using the pynput library. This library allows for the monitoring and controlling of input devices, making it suitable for capturing keyboard events.

3.2. Python Implementation

The implementation of the keylogger involves several key components:

- **Recording Keystrokes**: Capturing keypress events and writing them to a log file.
- **Handling Special Keys**: Differentiating between standard characters and special keys.
- **User Interface**: Providing a simple command-line interface for starting and stopping the keylogger.

```
from pynput import keyboard
log_file = "keylog.txt"

def write_to_file(key):
    with open(log_file, "a") as f:
        f.write(str(key) + "\n")

def on_press(key):
    try:
        if key == keyboard.Key.esc:
            return False
        else:
            write_to_file(key.char)
    except AttributeError:
        write_to_file(key)

def start_keylogger():
    with keyboard.Listener(on_press=on_press) as listener:
        listener.join()

if __name__ == "__main__":
    start_keylogger()
```

4. Implementation Details

4.1. Setting Up the Environment

To implement the keylogger, I set up a Python environment and installed the pynput library. This library facilitates the monitoring of keyboard events in a cross-platform manner, making it ideal for our needs.

```
C:\Users\saira>pip install pynput
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pynput in c:\users\saira\appdata\roaming\python\python310\site-packages (1.7.6)
Requirement already satisfied: six in c:\users\saira\appdata\roaming\python\python310\site-packages (from pynput) (1.16.
0)
```

4.2. Writing the Keylogger

The keylogger was designed to:

- **Capture Keystrokes**: The on_press function is a callback that is invoked every time a key is pressed. It handles both standard and special keys.
- **Log Keystrokes**: The write_to_file function opens the log file in append mode and writes each keystroke to it. This ensures that all key presses are recorded without overwriting previous entries.
- **Start the Listener**: The start_keylogger function initializes and starts the keylogger listener, continuously monitoring for keypress events until the Escape key is pressed.

4.3. Testing the Keylogger

To ensure the keylogger functioned correctly, I tested it with various inputs, including:

- Standard text (e.g., letters, numbers).
- Special characters (e.g., !, @, #).
- Special keys (e.g., Enter, Shift, Escape).

The keylogger successfully recorded all inputs and saved them to the log file, keylog.txt.

5. Ethical Considerations

Creating and using keyloggers involves significant ethical and legal considerations:

- **Permission**: It is crucial to obtain explicit permission from the owner of the computer where the keylogger will be used. Unauthorized use of keyloggers is illegal and unethical.
- **Legality**: Ensure that the deployment and use of the keylogger comply with local laws and regulations. Keylogging without consent is considered a breach of privacy and can lead to legal consequences.
- **Transparency**: Inform users about the keylogger and its purpose. Transparency helps maintain trust and ensures that the tool is used responsibly.

6. Conclusion

This task provided valuable experience in developing tools for monitoring user input while emphasizing the importance of ethical considerations. The keylogger successfully captured and logged keystrokes, demonstrating the ability to create such tools responsibly.

7. References

- pynput Documentation: pynput.readthedocs.io

8. Future Work

Future enhancements to the keylogger could include:

- **Encryption**: Encrypting the log file to ensure that the recorded keystrokes are secure and only accessible by authorized personnel.
- **Filtering**: Implementing filters to exclude specific applications or types of input (e.g., passwords) to enhance privacy and focus on relevant data.
- **Real-time Monitoring**: Developing a real-time monitoring system to alert administrators of specific keywords or activities.