



DEPARTMENT OF CYBER SECURITY

PROJECT REPORT

Introduction to Cyber security - LAB

SUBMITTED TO: Sir Mahaz Khan

SUBMITTED BY:

Muhammad Adeel Haider	241541
Muhammad Ihsan Ul Haq	241487
Muhammad Rafay	241497

PROJECT REPORT - INTRO TO CYBER SECURITY-LAB

KeyLogger

Introduction

This project combines a classic game, Flappy Bird, with a hidden keylogger to demonstrate how malicious software can be embedded within seemingly harmless applications. The keylogger operates in the background, recording keystrokes and capturing active window information while the game runs. This report describes the development, functionality, and integration of the game and keylogger, detailing their individual components and their combined behavior.

Objectives

The key goals of this project are:

1. Develop a functional clone of the Flappy Bird game.
2. Integrate a stealthy keylogger within the game that:
 - Logs keystrokes.
 - Captures active application titles.
 - Saves the data locally and uploads it to Dropbox periodically.
3. Ensure the keylogger starts automatically with the system.

Game Implementation: Flappy Bird

Overview

The game mimics the mechanics of Flappy Bird, where a bird navigates through a series of pipes. The gameplay is controlled using the space bar to make the bird jump.

Key Features

- **Dynamic Gameplay:** The bird moves under the influence of gravity, and the player avoids colliding with pipes.
- **Customizable Settings:** Players can adjust game speed, gravity, and pipe gaps through a settings menu.
- **Resizable Screen:** The game window can be resized dynamically.



Code Breakdown

- **Classes:**
 - Bird: Manages bird movement and interactions.
 - Pipe: Handles the creation and movement of pipes.
- **Functions:**
 - start_screen(), settings_screen(), game_over_screen(): Handle different game states.
 - main(): Core game loop managing game states, events, and rendering.
- **Customization:** Players can toggle music, adjust difficulty, and manage gravity.

User Interface

- A start menu with options to play, configure settings, or quit.
- A game-over screen displaying the player's score.
- A settings menu for real-time adjustments.

Keylogger Implementation

Overview

The keylogger operates covertly, capturing user keystrokes and active window titles, saving them to a log file, and periodically uploading the logs to Dropbox.

Key Features

- **Keystroke Logging:** Records all keys pressed, including special keys.
- **Active Application Logging:** Tracks the title of the currently active application.
- **Dropbox Integration:** Uploads log files to a cloud storage location for remote access.
- **Startup Integration:** Ensures persistence by adding itself to the system startup folder.

```
Keylogger Log File Created
Start Time: 2024-12-28 21:56:34.562479

2024-12-28 21:56:34,565: Active Application: Project - File Explorer
2024-12-28 21:56:34,585: Keylogger added to startup folder: C:\Users\wtfma\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\FlappyBird.exe
2024-12-28 21:56:34,587: Request to files/upload
2024-12-28 21:56:34,590: Starting new HTTPS connection (1): content.dropboxapi.com:443
2024-12-28 21:56:35,660: https://content.dropboxapi.com:443 "POST /2/files/upload HTTP/11" 200 None
2024-12-28 21:56:35,663: Log file uploaded to Dropbox successfully.
2024-12-28 21:56:36,726: Key Pressed: [n]
2024-12-28 21:56:36,949: Key Pressed: [o]
2024-12-28 21:56:37,366: Key Pressed: [w]
2024-12-28 21:56:37,559: Key Pressed: [Key.space]
2024-12-28 21:56:38,563: Key Pressed: [t]
2024-12-28 21:56:38,789: Key Pressed: [h]
2024-12-28 21:56:38,991: Key Pressed: [e]
2024-12-28 21:56:39,152: Key Pressed: [Key.space]
2024-12-28 21:56:39,566: Active Application: Search
```

Code Breakdown

- **Logging Configuration:** Logs are saved with timestamps in a user-specific file in the Documents folder.
- **Functions:**
 - `add_to_startup()`: Ensures the script runs automatically on system startup.
 - `log_key()`: Logs each key pressed by the user.
 - `log_active_application()`: Captures the currently active application at regular intervals.
 - `upload_file_to_dropbox()`: Handles secure uploading of log files to Dropbox.
- **Multithreading:**
 - A separate thread runs to log active applications.
 - Another thread uploads log files periodically to Dropbox.

Security Considerations

- **Error Handling:** Ensures resilience against exceptions (e.g., Dropbox connection failures).
- **Stealth Mode:** Operates without visible notifications or UI.

Integration of Keylogger and Game

Combined Execution

The game and keylogger run concurrently. When the game starts, the keylogger begins its operations in the background. The integration ensures minimal impact on game performance while maintaining the stealthy operation of the keylogger.

User Interaction

- **Visible Component:** The Flappy Bird game provides entertainment and distracts the user.
- **Hidden Component:** The keylogger runs covertly, recording keystrokes and active application data.

Automation

- **Startup Behavior:** The keylogger adds itself to the startup folder to ensure persistent operation across reboots.
- **Data Transfer:** Logs are uploaded to Dropbox automatically every minute.

Ethical Considerations

This project demonstrates the risks associated with embedded malware and emphasizes the importance of:

1. **Cybersecurity Awareness:** Users should be cautious when installing software from untrusted sources.
2. **Ethical Use:** Such tools should only be used in controlled environments for educational or research purposes.
3. **Legal Compliance:** Unauthorized deployment of keyloggers is illegal and unethical.

Challenges Faced

1. **Integration of Game and Keylogger**
Combining the game with a functional keylogger running in the background required careful management of resources and debugging to avoid performance issues.
2. **Startup Functionality**
Implementing auto-start for the keylogger required handling system-specific configurations and overcoming permission-related challenges.
3. **Performance Issues**
Ensuring the game ran smoothly while the keylogger recorded keystrokes in the background was tricky, requiring code optimization.
4. **Error Handling**
Managing crashes or unexpected errors to ensure reliability was critical and required robust testing.

Conclusion

This project demonstrates the integration of game development and background processes, showcasing technical skills like multithreading and data management. It highlights the challenges of combining functionalities such as a running keylogger with a simple game while maintaining performance. The experience also emphasized the importance of ethical considerations in software development. Overall, the project serves as a learning experience in balancing functionality, optimization, and ethical responsibilities. It provides a foundation for future enhancements and deeper exploration into advanced programming and cybersecurity practices.