myMyKeylogger Documentation

This documentation provides a comprehensive guide on how to use the myKeylogger script, including using the default file location, specifying a custom file path, encrypting/decrypting log files using the password "test", and creating a Windows executable.

Prerequisites

- Python 3.6 or higher
- Required Python libraries: `pynput` and `cryptography`

Install the required libraries using pip:

```
> pip install pynput cryptography
```

Using the Default File Location

1. **Running the MyKeylogger**: To run the myKeylogger using the default file location, execute the following command:

   ```
   > python myKeylogger.py --password test
   ```

   This will start the myKeylogger and save the keystroke logs to a default location based on your operating system:

   - **Windows**: `C:\Users\<YourUsername>\AppData\Roaming\processmanager2.txt`
   - **Linux**: `/root/processmanager2.txt`

2. **Stopping the MyKeylogger**: ctrl-c does not kill the myKeylogger. Either close the window or kill the process.
3. **Decrypting the Log File**: To decrypt and view the log file, use the following command:

   ```
   python myKeylogger.py --password test --decrypt
   ```

   This will print the decrypted contents of the log file to the terminal.

Using a Custom File Path

1. **Running the MyKeylogger with a Custom File Path**: To specify a custom file path for saving the keystroke logs, use the `--path` argument. For example:

   ```
   > python myKeylogger.py --password test --path /path/to/custom_log.txt
   ```

   Replace `/path/to/custom_log.txt` with your desired file path.

2. **Stopping the MyKeylogger**: Press `Ctrl+C` in the terminal to stop the myKeylogger. The log file will be automatically encrypted using the provided password ("test").
3. **Decrypting the Custom Log File**: To decrypt and view the log file saved at a custom location, use the following command:

   ```
   > python myKeylogger.py --password test --path /path/to/custom_log.txt --decrypt
   ```

   This will print the decrypted contents of the log file to the terminal.

File Structure

The myKeylogger script uses the following file structure:

- **Temporary Log File**: A temporary file (`.tmp` extension) is used to store keystrokes before encryption. This file is located in the same directory as the final log file.)
- **Encrypted Log File**: The encrypted log file is saved at the specified path (or default path if not specified). This file is created after the myKeylogger stops and the temporary file is encrypted.

Example Usage

1. **Using the Default File Location**:

   ```
   python myKeylogger.py --password test
   ```

2. **Using a Custom File Path**:

   ```
   python myKeylogger.py --password test --path /path/to/custom_log.txt
   ```

3. **Decrypting the Log File**: For default location:

   ```
   python myKeylogger.py --password test --decrypt
   ```

   For custom location:

   ```
   python myKeylogger.py --password test --path /path/to/custom_log.txt --decrypt
   ```

Creating an Executable for Windows

To convert the Python script into a standalone executable for Windows, follow these steps:

1. **Install PyInstaller**: Ensure you have PyInstaller installed. If not, install it using pip:

   ```
   pip install pyinstaller
   ```

2. **Generate the Executable**: Navigate to the directory containing you're the Python script and run the following command:

   Use the following command to create a secure, console-less executable that runs in the background

   ```
   pyinstaller --onefile --noconsole --hidden-import=pynput --hidden-import=cryptography myKeylogger.py
   ```

   This will create a single executable file. PyInstaller will generate several directories and files. The main directories are:

   - **`build/`**: Contains files used during the build process.
   - **`dist/`**: Contains your final executable.

   The executable file will be located in the `dist/` directory and will be named `myKeylogger.exe`.

3. **Running the Executable**: To run the myKeylogger executable, open a command prompt, navigate to the `dist/` directory, and execute:

   ```
   myKeylogger.exe --password test
   ```

   This will start the myKeylogger and save the logs to the default location:

   - **Windows**: `C:\Users\<YourUsername>\AppData\Roaming\processmanager2.txt`
4. **Stopping the MyKeylogger**: kill the process or close the python windows.
5. **Decrypting the Log File**: To decrypt and view the log file using the executable, use the following command:

   ```
   myKeylogger.exe --password test –decrypt
   ```

Additional Steps for Enhanced Security

1. **Obfuscate Your Code** (Optional): Consider using code obfuscation tools to make reverse engineering more difficult. Tools like `pyarmor` can help with this:

   ```
   > pip install pyarmor
   ```

   ```
   > pyarmor gen myKeylogger.py
   ```
   (version 8+) this will put the new scrypt in a \dist folder

2. **Set File Attributes to Hidden** (Optional): You can make the executable hidden by setting its file attributes. This can be done using the `attrib` command in Windows:

   ```
   > attrib +h path\to\your\executable\myKeylogger.exe
   ```

Current as of 05 Jun 24