**KEYLOGGER**

Submitted in partial fulfillment of the requirements of the degree

**B.Tech. (Computer Engineering)**

By

**Shreyash Deo 22CC1064**

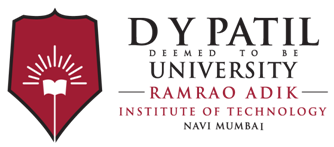
**Sarthak Patil 22CC1061**

**Hardik Dhote 22CC1049**

**Varad Parte 22CC1050**

Supervisor

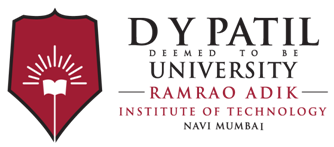
**(Ms. Ashwini Chavan)**



Department of Computer Engineering Ramrao Adik Institute of Technology,

Sector 7, Nerul, Navi Mumbai

(Under the ambit of D. Y. Patil Deemed to be University) **November 2023**



**Ramrao Adik Institute of Technology**

(Under the ambit of D. Y. Patil Deemed to be University)

Dr. D. Y. Patil Vidyanagar, Sector 7, Nerul, Navi Mumbai 400 706.

# Certificate

This is to certify that, the Mini Project – II entitled

**“Keylogger”**

is a bonafide work done by

**Shreyash Deo 22CC1064**

**Sarthak Patil 22CC1061**

**Hardik Dhote 22CC1049**

**Varad Parte 22CC1050**

and is submitted in the partial fulfillment of the requirement for the degree of

**B. Tech. in Computer Engineering**

to the

**D. Y. Patil Deemed to be University**

### 

Supervisor

**(Ms. Ashwini Chavan)**

Dr. Shivangi Agarwal Dr. A. V. Vidhate Dr. Mukesh Patil

Mini Project Coordinator Head of Department Principal

# Mini Project - II Approval

This Mini Project - I entitled **“Keylogger”** by

**Shreyash Deo (22CC1064), Sarthak Patil (22CC1061),**

**Hardik Dhote (22CC1049), Varad Parte (22CC1050)**

is approved in the partial fulfillment of the requirement for the degree of **B. Tech. in Computer Engineering**

**Examiners**

**1………………………………………**

(Internal Examiner Name & Sign)

### 2…………………………………………

(External Examiner name & Sign)

Date:

Place:

# Abstract

# Keyloggers, the clandestine agents of digital espionage, pose a profound threat to the security and privacy of individuals and organizations in today's interconnected world. Operating surreptitiously, these insidious tools covertly monitor and record keystrokes entered by users on computers and mobile devices, harvesting sensitive information such as passwords, credit card numbers, and personal communications. This report aims to dissect the anatomy of keyloggers, exploring their mechanisms, classifications, detection methods, and broader implications for cybersecurity. Keyloggers manifest in two primary forms: software-based and hardware-based. Software variants infiltrate systems through malware, phishing attacks, or compromised websites, embedding themselves within the operating system to evade detection by traditional security measures. Conversely, hardware keyloggers, physical devices inserted between keyboards and computers, intercept keystrokes directly, presenting a formidable threat in scenarios where physical access to the target device is feasible. Detecting keyloggers poses a significant challenge, as they operate stealthily, often bypassing antivirus software and firewalls. Specialized detection techniques such as behavioral analysis, anomaly detection, and signature-based scanning are crucial in identifying keylogger activity. Moreover, the use of anti-keylogging software and regular system audits can aid in mitigating the risk posed by these covert threats. The implications of keyloggers extend beyond individual privacy breaches to encompass broader cybersecurity concerns. In corporate environments, keyloggers jeopardize sensitive information, leading to financial losses, reputational damage, and legal ramifications. State-sponsored actors utilize keyloggers for espionage, infiltrating government agencies, defense contractors, and critical infrastructure systems. By understanding the intricacies of keyloggers and implementing proactive security measures, individuals and organizations can strive to safeguard digital ecosystems from the pervasive threat posed by these invisible adversaries.

# Contents

# Introduction

# Overview

# Motivation

# Problem Statement and Objectives

# Literature Survey

# 2.1 Survey of Existing System

# 2.2 Limitations of Existing System

# Proposed System

# 3.1 Problem Statement

# 3.2 Proposed Methodology / Techniques

# 3.3 System Design

# 3.4 Details of Hardware and Software Requirements

1. Results and Discussion

4.1 Implementation Details

4.2 Project Outcomes

1. Conclusion and Future Work

References

Appendices

* Weekly Progress Report

Acknowledgement

# Chapter 1

# Introduction

* 1. **Overview**

A keylogger is a type of software or hardware device designed to covertly record keystrokes entered by users on a computer or mobile device. It operates stealthily, capturing sensitive information such as usernames, passwords, and credit card numbers, which can be exploited by malicious actors for various purposes including identity theft and financial fraud.

* 1. **Motivation**

The primary motivation behind creating keyloggers is to surreptitiously gather sensitive information such as passwords, financial credentials, and personal data. Malicious actors use this information for nefarious purposes including identity theft, financial fraud, espionage, and sabotage, highlighting the lucrative potential and malicious intent driving the development of keylogging software.

* 1. **Objectives**

While keyloggers are primarily associated with malicious intent, there are some legitimate use cases where they can be employed for beneficial purposes:

* Parental Control
* Employee Monitoring
* Data Recovery
* Password Recovery
* Accessibility Assistance

# Chapter 2

# Literature Survey

# 2.1 Survey of Existing System

**ICCEMME 2021 -**Keyloggers are kind of a rootkit malware that catch composed keystroke occasions of the console and save into log record, hence, it can capture delicate data, for example, usernames, PINs, and passwords, in this manner communicates into vindictive assailant without pulling in the consideration of clients.

**Research Gate-** Cybercriminals have devised many methods to obtain sensitive information from your endpoint devices. However, few of them are as effective as keystroke logging. Keystroke logging, also known as keylogging, is the capture of typed characters. The data captured can include document content, passwords, user ID’s, and other potentially sensitive bits of information

**JETIR(Journal of Emerging Technologies and Innovative Research) -** Keyloggers are becoming more prevalent. They operate in stealth mode, making them difficult for antivirus software to identify. Keyloggers can be avoided, nevertheless, by taking certain precautions. Applications for prevention, such as firewalls and anti-malware, must be downloaded. Updating security fixes on a regular basis

**International Journal for Research in Applied Science & Engineering Technology (IJRASET)** This paper has surveyed the working of keyloggers and the different types of keyloggers. We have also examined the categories of software keyloggers. though keylogger developers market their products as legitimate software.

# Joint Workshop of KO2PI 2017 & ICMSTEA 2016- Keylogger applications designed by implementing the Exact String Matching algorithm can record all user activities related to the keyboard, and the results are stored automatically in a dedicated database that can only be accessed by the keylogger owner, the next development of the keylogger application can record the activity on the virtual keyboard or remote activity on the user's computer.

# 2.2 Limitations of Existing System

Addressing these limitations requires ongoing innovation in keylogger development, focusing on enhancing stealth capabilities, improving compatibility, and ensuring compliance with ethical and legal standards. Additionally, user education and awareness initiatives can help mitigate the risks associated with keyloggers and promote responsible usage practices.

* **False Positives and inaccurate capture:** Keylogger may inadvertently capture irrelevant

Keystrokes or produce false positives, leading to inaccurate data collection and analysis, especially in high-traffic environment or system with multiple users.

* **Not applicable on all applications:** Some keyloggers may not work on web browsers due to security measures of that browser, leading to not being able to collect valuable data from the target.
* **Network Dependency:** Some keyloggers rely on network connectivity to transmit captured data to remote servers or control centers, posing a risk of data loss or interception if network communication is disrupted or intercepted.
* **Limited Compatibility:** Keyloggers may not be compatible with all operating systems, applications, or input devices, restricting their functionality and usability across diverse computing environments.

# Chapter 3

# Proposed System

A proposed keylogger system would focus on stealth, compatibility, and data security. It would incorporate advanced encryption methods to protect captured data, operate covertly to evade detection by antivirus software, and ensure compatibility with diverse computing environments to maximize usability and effectiveness.

# Problem Statement

# The prevalent use of keyloggers in cybercrime poses a significant threat to individuals' privacy, organizational security, and data integrity. However, existing detection methods and countermeasures often fall short in effectively identifying and mitigating keylogger attacks. This highlights the urgent need for robust solutions that can accurately detect and neutralize keyloggers across various platforms and applications. Additionally, ethical and legal concerns surrounding the use of keyloggers further compound the challenge, necessitating comprehensive approaches that balance security needs with privacy rights and regulatory compliance. Addressing these issues is imperative to safeguarding digital ecosystems and protecting against covert data exfiltration and exploitation.

# Proposed methodology / Techniques

# Proposed methodologies and technologies for keyloggers should prioritize stealth, accuracy, and compatibility while addressing ethical and legal considerations. We made this entire keylogger project on python using following libraries- keystroke, keyboard time ,date time , threading. By integrating these methodologies and technologies into keylogger detection and prevention solutions, organizations can enhance their cybersecurity posture, mitigate the risks posed by keyloggers, and safeguard sensitive information from unauthorized access and exploitation.

# 5. Python is a versatile and widely-loved programming language celebrated for its readability and simplicity. Python is a general-purpose language, serving various domains, from web development to simple applications like making a keylogger.

# Python libraries used in the keylogger are:

# Keyboard: This library provides functions for capturing keyboard events in Python. It allows the program to listen for keystrokes and execute callback functions when keys are pressed or released. In the code, the `keyboard.on\_release()` function is used to register a callback function `callback()` that gets executed whenever a key is released.

# Timer: This class from the `threading` module is used to schedule the execution of a method (`report()`) at regular intervals (`SEND\_REPORT\_EVERY`). It allows the program to periodically send email notifications or save keylogs to a file.

# Threading: The `threading` library in Python provides a high-level interface for creating and managing threads within a Python program. Threads are lightweight subprocesses that run concurrently within a single process, allowing multiple tasks to be executed simultaneously. Overall, the `threading` library simplifies the process of working with threads in Python, allowing developers to write concurrent and parallel programs efficiently.

# These libraries work together to create a simple keylogger that captures keystrokes, saves them to a file or sends them via email at regular intervals, and runs indefinitely until terminated by the user.

# 3.3 System Design

# Creating a flowchart for Keylogger would involve visually representing the sequence of steps and decision points within the system. Flowcharts typically use standard symbols to represent processes, decisions, inputs, and outputs.

# 

# Fig. 1 : Design of keylogger

# The Fig 1 depicts a simplified computer keyboard interface. It consists of the following functional blocks:

# Keyboard Matrix: This is the physical layout of the keys on the keyboard. Each key is connected to a row and a column in a matrix. When a key is pressed, it completes a circuit between the corresponding row and column.

# Keyboard Controller: This is an integrated circuit (IC) that scans the keyboard matrix to detect which key has been pressed. It does this by systematically applying a voltage to each row one at a time and then reading the columns to see if any connection is made.

# Keyboard Translator: This block translates the scan code generated by the keyboard controller into a keycode that the operating system can understand. The scan code is a unique identifier for the key that was pressed based on its position in the matrix. The keycode represents the character or function associated with the key.

# Keyboard Buffer: This is a temporary storage location that holds the keycodes of the keys that have been pressed. The keyboard buffer is necessary because the keyboard controller can scan the keyboard matrix and generate scan codes faster than the operating system can process them.

# Operating System: The operating system is the software that manages the computer's resources and provides an interface for users to interact with the computer. The operating system receives the keycodes from the keyboard buffer and interprets them as keystrokes.

# Keylogger (optional): A keylogger is a hardware device or software program that records all the keystrokes that are typed on a keyboard. It is important to note that keyloggers can be used for malicious purposes, such as stealing passwords or other sensitive information.

# 3.4 Flowchart of Keylogger

# 

# Fig 2 : Working of Keylogger

The Fig 2 is a flowchart illustrating the process of how a **keylogger** works. Here are the key steps outlined in the flowchart:

1. **Initialize Keylogger**:
   * The process begins by initializing the keylogger.
   * This step sets up the necessary components for capturing keystrokes.
2. **Check if Keylogger is Active**:
   * The flowchart includes a decision point to determine whether the keylogger is active or not.
   * If it is active, the process proceeds to the next steps; otherwise, it terminates.
3. **Capture Keystrokes**:
   * If the keylogger is active, it captures keystrokes made by the user.
   * These keystrokes can include letters, numbers, special characters, and function keys.
4. **Store Keystrokes**:
   * The captured keystrokes are stored in memory or a log file.
   * This step accumulates the user’s input over time.
5. **Check for Special Commands**:
   * The keylogger checks for any special commands or sequences.
   * If a specific command is detected (e.g., a predefined combination of keys), it triggers an action.
6. **Execute Command**:
   * If a special command is detected, the keylogger executes the corresponding action.
7. **Check for Transmission Interval**:
   * The flowchart includes another decision point to determine if it’s time to transmit the captured keystrokes.
   * If the interval condition is met, the process proceeds to the next step; otherwise, it waits.
8. **Transmit the Keystrokes to Remote Server**:
   * If the transmission interval is reached, the keylogger sends the stored keystrokes to a remote server.
   * This step allows an attacker to retrieve the logged data remotely.
9. **End**:
   * The flowchart concludes with an “End” symbol, indicating the completion of the keylogging process.

The flowchart provides an overview of how a keylogger operates, from initialization to data transmission. Keyloggers can be used for legitimate purposes (e.g., monitoring employee activity) or maliciously (e.g., stealing sensitive information).

# 3.5 Algorithm of Keylogger

**Import necessary libraries:**

* keyboard for capturing keystrokes.
* Timer from threading for scheduling tasks.
* datetime for handling date and time operations.

**Set constants:**

* SEND\_REPORT\_EVERY: Interval in seconds for reporting keystrokes.

**Define the Keylogger class:**

* Initialize instance variables including interval, report method, log, start and end datetimes.
* Define callback method to capture keystrokes and update log.
* Define update\_filename method to construct a filename based on start and end datetimes.
* Define report\_to\_file method to save keystrokes to a local log file.
* Define report method to send logs or save to file based on the report method.
* Define start method to start capturing keystrokes and scheduling reporting.

**Implement the Keylogger class methods:**

* callback: Capture keystrokes and update log with key names.
* update\_filename: Generate a filename based on start and end datetimes.
* report\_to\_file: Save keystrokes to a local log file.
* start: Start capturing keystrokes and scheduling reporting.

**Instantiate a Keylogger object and start the keylogging process.**

# Details of Hardware and Software Requirements

# A Keylogger is designed to work on any hardware that meets the minimum requirements for a computer with a keyboard.

# Hardware Requirements:

# Computer: A computer system where the keylogger will be installed.

# Storage: Sufficient storage space to store the logged keystrokes and any additional data.

# Input Device: A keyboard connected to the computer where the keylogger will capture keystrokes.

# Optional: Internet connectivity if remote monitoring or reporting features are desired.

# Chapter 4

# 

# Results and Discussion

# This chapter presents the results generated by Keylogger and all the key strokers captured.

# Main File

# Keylogger file is stored

# Fig. 3: Keylogger file

# Key strokes can be captured on any website or an application on the device.

# 

# Fig 4: Capturing key strokes

# 

# Fig. 5: Capturing key strokes on Instagram

# Keyloggers can be used to monitor employee activity on company-owned computers to ensure compliance with company policies and procedures. They can help track email-id or any insider threat.

# 

# Fig 6: Capturing key strokes of gmail-id

# Parents may feel it's necessary to monitor their child's online behaviour for safety reasons, it's crucial to approach this with care and consideration for the child's privacy and trust.

# 

# Fig. 7: Capturing key strokes on You-tube

# All the stored Files

# Fig 8 show all the key strokes that are captured in the file. The name of the file has the date and time when the key strokes were captured. New file gets created per minute.

# 

# Fig. 8: All captured key strokes

# Chapter 5

# Conclusion

# In conclusion, the keylogger mini-project provided valuable insights into the functioning and limitations of keylogging software. Through the development and testing process, it became evident that keyloggers can be powerful tools for capturing text input but are subject to various constraints. These include the potential for detection by antivirus software, limitations in capturing non-textual input, such as mouse clicks, and challenges in encrypted environments. Additionally, the need for physical access to install certain keyloggers and concerns regarding accuracy and legal implications further underscored the complexities involved in their use.

# Despite these limitations, the project highlighted keyloggers' potential applications in cybersecurity, parental monitoring, and employee supervision when used ethically and within legal boundaries. Furthermore, the experience gained from this mini-project serves as a foundation for understanding broader concepts in digital security, such as privacy concerns, data protection, and ethical considerations surrounding surveillance technologies. Moving forward, further exploration into the evolving landscape of cybersecurity tools and practices can deepen our understanding and proficiency in safeguarding digital assets and maintaining ethical standards in the digital realm.

# Ethical Considerations:

# Unlike other types of malicious program, keyloggers present no threat to the system itself. Nevertheless, they can pose a serious threat to users, as they can be used to intercept passwords and other confidential information entered via the keyboard. Unfortunately access to confidential data can sometimes have consequences which are far more serious than an individual’s loss of a few dollars. Though keylogger developers develop their products as legitimate software, but most of the keylogger are used to steal user data.

# References

# Tom Olzak (2008). Keystroke Logging (Keylogging)

# Arjun Singh, Pushpa Choudhary, Akhilesh Kumar Singh, Dheerendra Kumar Tyagi (2021). Keylogger Detection and Prevention.

# Kavya .C , Suganya. R.(2021) Survey on Keystrokes Logging attacks.

# Viraj Prajapati, Rahul Kalsariya, Amitesh Dubey, Kewal Mehta, Prof. Mahendra Patil (2020). Analysis of Keyloggers in Cybersecurity.

# Appendices

# Weekly Progress Report

**ACKNOWLEDGEMENT**

We take this opportunity to express my profound gratitude and deep regards to my guide **Ms. Ashwini Chavan** for her exemplary guidance, monitoring and constant encouragement throughout the completion of this report. We are truly grateful to her efforts to improve my understanding towards various concepts and technical skills required in our project. The blessing, help and guidance given by her time to time shall carry us a long way in the journey of life on which we are about to embark.

We take this privilege to express my sincere thanks to **Dr. Mukesh D. Patil, Principal, RAIT, D. Y. Patil deemed to be University** for providing the much necessary facilities. We are also thankful to **Dr. A. V. Vidhate,** Head of Department of Computer Engineering, **Mrs. Smita Bharne,** Mini Project Co-ordinator, for their generous support.

Last but not the least we would also like to thank all those who have directly or indirectly helped us in completion of this project report.

Shreyash Deo 22CC1064

Sarthak Patil 22CC1061

Hardik Dhote 22CC1049

Varad Parte 22CC1050