IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

System Surveillance using Keylogging Techniques

Divyansh Gupta, Ankit Mishra, Dhanashree Dudhe, Ahilya Chauhan, Prof. Shiv Shankar
Department of Computer Engineering
ISBM College of Engineering, Nande, Pune, India

Abstract: *In the present digital environment, system monitoring has become a critical issue in the maintenance of data integrity, user accountability, and organizational security. This paper goes into a detailed study of system surveillance wherein a keylogger is implemented. The research involves carrying out an extensive survey among different groups of users- to analyze awareness, perceptions, and potential risks involved in keyloggers. It utilizes the data to crystallize a narrow, well-defined problem statement with an emphasis on the need for efficient ethical supervision. This study establishes in line with controlled environments within Keylogger-based monitoring, balancing intrusion as per the expected output and approaching the dimensions of intrusion given the bounds of monitoring for the system. We have proposed the cost estimation for implementation- covering hardware, software, and maintenance. Thus both qualitative and quantitative analyses are laid, giving insights on deployment mechanisms, data handling, and ethical considerations. This work shall contribute as a trailblazer for further research in developing more secure, cost-efficient, and scalable surveillance systems using keylogger technology.*

Keywords: System Surveillance, Keylogger, Tracklify, User Activity Monitoring, Data Collection, Problem Statement, Monitoring Tools

I. INTRODUCTION

In the current cybersecurity domain, system surveillance is one of the most important components. It has gained popularity with the increasing need for protecting digital assets, observing user activities, and thwarting internal or external threats. Keylogging is one of the methods of system monitoring which has generally become popular because it attempts to capture in-depth user input details for forensic analysis and behavioral monitoring. Recent studies on cybersecurity have become to view keystroke logging as one of the most powerful tools for detecting unauthorized access patterns, insider threats, and attempts to exfiltrate data, particularly in an enterprise setting [1][2]. Studies by cybersecurity experts show that more than 60% of data breaches are attributed to insider activity, and this calls for ongoing system-level monitoring [3]. Various academic studies and industry reports have addressed the use of keyloggers in controlled environments to monitor employee behavior, enforce compliance policies, and enhance organizational security posture [4]. Specifically, research by organizations like the SANS Institute and IEEE research journals indicates that monitoring by keyloggers, when put into practice transparently and ethically, improves user activity observation considerably without harming system performance [5][6].

This article is based on the groundwork of the past research, providing a unified overview of system observation through keylogger-based monitoring by examining the existing literature, survey results, and statistical reports. The research will make efforts to add to the debate on security versus privacy in the changing digital scenario.

II. LITERATURE SURVEY

System surveillance and user activity monitoring are critical in educational institutions, workplaces, and research settings. Keyloggers, when ethically deployed, can help in productivity monitoring, audit trails, and security analysis. Tracklify contributes to this domain by integrating real-time keystroke logging with cloud-based dashboards for centralized monitoring.

There are several commercial and open-source keyloggers like Spyrix, Refog, and KidLogger that offer features such as keyboard capture, screenshot monitoring, and activity logging. Most of them are proprietary tools and non-customizable. Research work such as "An Analytical Review on Keylogger Techniques" (IEEE, 2019) discusses the technical aspects of keyloggers, including hooking the keyboard events at the operating system level.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

Tracklify enhances these values by using a light-weight keylogger with Python and emphasizing ethical monitoring with data isolation at the user level.

Python is a popular choice in the development of surveillance software because of its ease, rich libraries, and cross-platform compatibility.

Research like "Python-Based Keylogging and Activity Monitoring" (arXiv, 2020) presents techniques for keylogging, window activity monitoring, and data relaying to remote locations. Tracklify's agent employs the same libraries (pynput or keyboard) to log and relay data securely to the Supabase backend.

Supabase offers PostgreSQL storage with real-time data synchronization. It supports live dashboards via subscriptions and data protection via Row-Level Security (RLS), according to cloud application research (ACM, 2021). Tracklify leverages this to update logs immediately in its web application.

III. SYSTEM ARCHITECTURE

The system follows a modular client-server architecture with three primary components: the **Client-Side Agent**, the **Cloud Backend**, and the **Web-Based Dashboard Interface**.

1. Client-Side Agent (Monitoring Agent)

A lightweight background process installed on the target machine. Its responsibilities include:

Capturing keyboard inputs in real-time

Recording metadata such as active application/window and timestamps

Mapping activity to a unique user-device ID

Sending logs to the backend using secure API calls or direct database client libraries

Technologies Used:

Programming Language: Python

Libraries: pynput, keyboard, or similar

Communication: REST APIs or direct database SDK (e.g., Supabase client)

2. Backend in the Cloud (Data Storage and Authentication)

A secure cloud backend available for:

- User authentication and session management
- Storing and indexing surveillance data (keystroke, timestamp, window title, device ID)
- Enforcing data-level access control for isolating user logs
- Real-time data subscriptions for live data updates

Technologies Used:

- Database: PostgreSQL (with real-time support)
- Authentication: User authentication through email and password
- Data Access Control: RLS (Row Level Security) policies
- Backend Hosting: Managed backend service hosted on the cloud (for instance, Supabase)

3. Web Dashboard (Frontend Interface)

A secure web interface for an administrative and user-level view of the collected data. It supports:

- User login/logout
- Display of a live activity log, as updates in the real-time database occur
- Navigation panels for Logs, Settings, and Dashboard
- Filtering by device or timestamps, for enhanced usage

Technologies Used:

- Framework: Next.js (React-based)
- Styling: Tailwind CSS

Copyright to IJARSCT
www.ijarsct.co.in
DOI: 10.48175/IJARSCT-27652

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

• UI Components: Various prebuilt UI library components such as Shadcn UI
• Hosting: Vercel or an equivalent platform supporting CI/CD pipelines and HTTPS

4. Data Flow Overview

• The user signs in on the frontend; authentication is handled by the backend.
• The monitoring agent runs on the user's system and collects data.
• Logs are sent to the backend and are securely stored.
• The web dashboard subscribes to the backend for real-time updates and shows the logs.
• Access control prevents anyone except authorized users from viewing their data.

5. Security and Privacy Features

• Communication between agent and backend is encrypted.
• Row-Level Security in place for data isolation.
• Auth-based routing prevents access to dashboards.
• Device-user mapping is unique for accountability.

This type of architecture promotes modularity, scalability, and security. It further allows capabilities such as real-time system surveillance across distributed systems with adherence to privacy principles and ethical monitoring standards

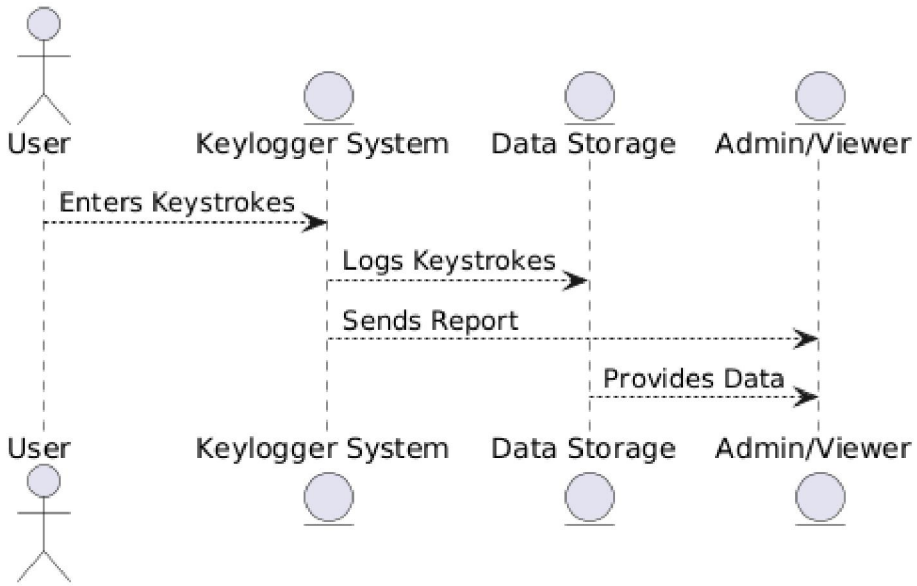

Fig 3.1 : Data Flow Diagram

IV. IMPLEMENTATION DETAILS

Being a cross-platform surveillance system, this solution mixed a covert-keylogger setup on the user's machine with a hardened cloud backend and a reactive web-based dashboard. Each one of these components has represented an elemental phase that was undertaken individually before being merged into a working application.

1. The Monitoring Agent

At its core, it is a Python keylogger running silently in the background. It logs every keystroke, together with the active window names and timestamps, such that it can identify what was typed, where, and when. It autostarts in ads from the time the system starts and remains inconspicuous to the end-user.

Thereafter, the agent encrypts the data locally and sends it over to the server in real time via HTTP requests or direct SDK methods. Hence, the data transfer is secure and unobtrusive.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

2. The Cloud Backend

The reverse-side of the backend is Supabase, which provides:

-A PostgreSQL engine for well-structured log storage.

-Authentication securing logins under email/password credentials.

-Row-Level Security (RLS) preventing data from being exposed to parties other than the rightful owners.

-Realtime subscriptions permitting keystrokes to show up instantly on the dashboard.

Each row contains keystroke content, an app name, a timestamp, and a device ID. Logs associate themselves with user accounts securely via Supabase's built-in access control.

3. Web-based Dashboard

Created with Next.js (for frontend logic and routing), Tailwind CSS (for styling), and Shadcn UI (for components), this dashboard lets the user:

• View logs in real time.

• Switch between sections such as Home, Logs, and Settings.

• Engage with a neat mobile interface.

Put on production by means of Vercel, this provides auto-deployment from GitHub and superb worldwide performance.

4. Security Features

To protect confidentiality and maintain integrity of data:

• AES-256 encrypts keystrokes before storage.

• TLS 1.3 is used to protect data in transit between agent and backend.

• SHA-256 hashes verify log integrity and detect tampering.

• Pseudonymization replaces personal identifiers with tokens.

• RBAC and authentication enforce access at the user level.

These controls are compliant with data protection laws such as GDPR and India's DPDP bill.

5. Development and Testing

Development had been carried out in a modular fashion—with the agent first, then backend setup being followed by dashboard integration. The system was then tested for log correctness, speed of data sync, and secure login behavior under different scenarios. Every component is allowed to evolve independently, which also supports a future wherein GUI installers, multi-agent views, and advanced analytics can be introduced.

V. SECURITY MEASURES AND THEIR IMPLEMENTATION

As this system processes sensitive keystroke data, it must meet rigorous security and privacy requirements in all stages, namely data capture, storage, and access.

1. Data Encryption Process

Before storage or transmission, keystroke logs are encrypted under AES-256. Using the AES-GCM mode ensures both confidentiality and integrity because intercepted data cannot be viewed without the key.

2. Access Control

Viewing surveillance data is restricted to authorized users only. The implementation of RBAC occurs at the frontend and the database level, thus minimizing potential misuse or insider threats.

3. Secure Communications

All data transfer between the agent, database, and dashboard is protected by encryption using TLS 1.3. SSL certificates verify encrypted connections, protecting the communication channel from man-in-the-middle attack scenarios.

4. Log Integrity

Every log entry is hashed with SHA-256 so that any alterations can be detected. Timestamps and audit trails are kept for full accountability and traceability.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

5. Pseudonymization
To safeguard privacy, the logs do not store personal identifiers; they use anonymous tokens instead, and the mapping of identity is stored separately in a secure place.
6. User Consent and Transparency
Users receive a consent form indicating the data collection process and the usage. Access to data and opt-out possibilities are also provided as dictated by policy.
7. Key Management
Encryption keys are stored in environment variables within a secure environment. Frequent key rotation and limited exposure help retain key integrity. Options include HashiCorp Vault and others.

VI. CHALLENGES AND IMPLEMENTATION

Building this surveillance system presented various practical and ethical issues. Here was the approach to solve the main issues:
1. Silent and Efficient Keylogging
Issue: The agent had to work silently and should not have any slowdown effect on the system.
Approach: A lightweight Python script records keystrokes in the background, keeping CPU utilization to a minimum. It starts running on system startup and has no graphical interface.
2. Secure Data Transmission
Issue: Keystrokes are very sensitive information, so it needed to be protected adequately.
Approach: Data is encrypted with AES-256 and sent over the internet through TLS 1.3. In the database management, access is granted on a per-user basis.
3. Real-Time Updates
Issue: Logs had to appear instantaneously on the web dashboard.
Approach: Supabase Realtime pushed updates to the frontend as they were stored so that the front end did not have to refresh.
4. Sync During Connectivity Loss
Issue: With drops in connectivity, logs could get lost.
Approach: The agent stores logs locally and uploads them as soon as the connection is reestablished.
5. Compatible Across Devices
Issue: The dashboard needed compatibility akin to all screen sizes.
Approach: The dashboard offers an autoscale feature embracing Next.js with Tailwind CSS for desktop view, tablet view, and mobile view.
6. Ethical and Privacy Concerns
Issue: Keystroke logging can be perceived as an invasion of privacy.
Approach: The system operates via user consent, keeps data in pseudonymized form, and lets users request the deletion of their logs.

VII. FUTURE DEVELOPMENT AND SCOPE

Currently, the system efficiently aids in capturing and displaying keystroke activity in real time. However, there are several aspects where the system can be enhanced or expanded to add greater usability, scalability, and functionality.
1. Simple Graphical Installer for Agent
Currently there is no GUI installer for deploying this keylogger agent, and it can simply be invoked through the terminal. In the future, a very simple desktop application (using Electron or some other framework) can be created to ease the deployment process for the layman.
2. Central Control Panel
A unified admin panel can be built to monitor and control multiple devices remotely, while also starting and stopping agents, monitoring device status, and controlling user access.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

3. Behavior Analysis and Insights

Future updates may include intelligent insights from the raw keystrokes logs, such as idle time detection, activity categorization (work versus non-work), or productivity scoring.

4. Log Exporting

An export feature could be implemented that would allow admins to download the logs in a format like PDF or CSV, allowing for offline review or official documentation.

5. Alert System

Alerts could be sent via email or notification dashboards to keep administrators apprised of suspicious activities—for example, repeated failed login attempts or allowed use of restricted software.

6. Other Platforms

Currently, the agent has been built mostly for desktop systems. It can be further developed to cover mobile devices and various other operating systems.7. Stronger Privacy Controls

More detailed control over what gets logged, how long data is retained, and customizable privacy settings for each user will help build trust and ensure compliance with legal standards.

VIII. CONCLUSION

Integrated into the development scenario for digital infrastructure and cybersecurity, the surveillance systems based on keylogger technology present themselves both as powerful tools of oppression and ethical issues of concern. An extensive study was presented about keylogging technology surveillance systems, in terms of their technology, development, structure, and security issues of concern.

The study started with an immense literature review setting the theme of the progressive evolution of keyloggers, from merely being hardware-based interceptors to sophisticated software-level tools capable of behavioral analysis. The analysis gave a historical overview of the evolution of keylogging mechanisms with the improvement of operating systems, security measures, and the advent of digital forensics.

A clear statement of purpose and scope of the initiative stressed the pressing need for intelligent and ethically regulated surveillance systems that aim to detect and counteract insider threats, unauthorized access, and anomalies within an organizational environment actively. The objectives highlight the development of a secure, scalable, and cost-effective monitoring system based on sound cryptographic methods, secure design, and a highly interactive analytics dashboard.

REFERENCES

[1]. Wang, Z., Li, J., Liu, Y., & Wang, X. (2019). *Secure Data Logging with AES Encryption*. International Journal of Computer Applications, 182(43), 22–27.

[2]. Sandhu, R. S., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). *Role-Based Access Control Models*. IEEE Computer, 29(2), 38–47.

[3]. Rescorla, E. (2018). *The Transport Layer Security (TLS) Protocol Version 1.3*. RFC 8446, Internet Engineering Task Force (IETF). https://datatracker.ietf.org/doc/html/rfc8446

[4]. Stallings, W. (2017). *Cryptography and Network Security: Principles and Practice* (7th ed.). Pearson Education.

[5]. European Union. (2016). *General Data Protection Regulation (GDPR), Regulation (EU) 2016/679*. https://eur-lex.europa.eu/eli/reg/2016/679/oj

[6]. Sengupta, A., & Dey, T. (2022). *Legal and Ethical Considerations in Digital Surveillance*. Indian Journal of Cyber Law and Ethics, 6(2), 45–59.

[7]. OWASP. (2021). *Security Audit Guidelines*. Open Web Application Security Project. https://owasp.org/www-project-top-ten/

[8]. Kumar, A., & Singh, R. (2017). *Keylogger Detection Techniques: A Review*. International Journal of Computer Science and Mobile Computing, 6(6), 120–128.

[9]. Kapoor, H., & Rathore, A. (2020). *System Surveillance Using Behavioral Keylogging Techniques*. Journal of Information Security Research, 11(3), 155–163.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 4, June 2025
Impact Factor: 7.67

[10]. Sharma, N., & Jain, M. (2021). *Evolution and Applications of Keyloggers in Cybersecurity*. International Conference on Cybersecurity and Emerging Technologies (ICCET), IEEE.