Keystroke Anomaly Detector (KAD): A Real-Time Behavioural Biometric Security System

Gangadhar R, Varun Sai Y, Karthik S, Vamsi K, Chaitanya Sai L

Computer Science & Engineering Department, Raghu Engineering College, Visakhapatnam, A.P., India

---

Abstract - The proliferation of automated cyber-attacks and sophisticated social engineering tactics has rendered traditional point-in-time authentication insufficient for securing workstation environments. This paper proposes the Keystroke Anomaly Detector (KAD), a real-time behavioral-biometric security system designed for Windows workstations to strengthen Cyber Supply Chain (CSC) security. KAD addresses unauthorized access, impersonation, and HID injection attacks by learning a user's unique typing rhythm through 23 biometric keystroke features. An ensemble of unsupervised Machine Learning algorithms — Isolation Forest and One-Class Support Vector Machine (SVM) — detects real-time anomalies, while a Cyber Threat Intelligence (CTI) module performs regex-based command signature detection. A weighted detection engine combines ML anomaly scores (45%), behavioral rhythm scores (25%), and command signature scores (30%) into a unified risk assessment. Critical threats trigger automated responses including keyboard locking, webcam evidence capture, and Telegram notifications. Experimental results demonstrate high true positive rates for impersonation, HID injection, and malicious command scenarios, with low false positives under normal conditions. The system operates via a local Flask-based web dashboard and requires no additional hardware.

Key Words: *Keystroke Dynamics, Behavioral Biometrics, Machine Learning, Anomaly Detection, Endpoint Security, Isolation Forest, One-Class SVM, Cyber Supply Chain Security.*

---

1. INTRODUCTION

Security of the endpoint workstation is essential for ensuring overall business continuity and safeguarding organizational data within a Cyber Supply Chain (CSC). Modern computing environments are inherently complex; vulnerabilities can be exploited once an initial authentication boundary is breached. Traditional authentication measures such as passwords or smart cards verify identity only at a single point in time. If an endpoint is left unlocked, an adversary or an automated script can execute malicious payloads without further challenge.

Cyber Supply Chain (CSC) security is critical for reliable service delivery and ensuring overall business continuity of Smart Cyber-Physical Systems (CPS). CSC systems are inherently complex; vulnerabilities within the CSC environment can cascade from a source node to a number of target nodes of the overall CPS. A recent NCSC report highlights a list of CSC attacks by exploiting vulnerabilities that exist within the systems. Several organizations outsource parts of their business and data to third-party service providers, which could lead to potential threats.

A collection of endpoint-focused attacks that take advantage of operational vulnerabilities necessitates continuous monitoring. Organizations face threats not only from external malware, but from insider threats and physical intrusion. Notable examples include successful command-injection attacks via Human Interface Device (HID) spoofing — such as USB Rubber Duckies — which can deploy credential dumpers or reverse shells within milliseconds. While existing cybersecurity mechanisms account for certain static malware signatures, real-time behavioral and threat intelligence properties are not given adequate attention for continuous endpoint security.

This paper proposes KAD (Keystroke Anomaly Detector), a lightweight system that integrates Keystroke Dynamics with Machine Learning (ML) techniques and CTI-based signature detection to predict cyber-attack patterns and provide automated countermeasures, including screen locking, webcam capture, and instant notifications. The novelty of KAD is threefold: (1) it applies keystroke biometrics as a continuous authentication layer rather than a one-time login check; (2) it integrates CTI-based command signature detection with the biometric anomaly score through a weighted detection engine; and (3) it delivers automated, tiered responses managed through a local Flask web dashboard.

KAD operates as a non-intrusive and cost-effective solution since it requires no additional hardware and works seamlessly in the background without disrupting user activity. The system is capable of detecting insider threats and compromised sessions by identifying subtle deviations in typing patterns, even when valid credentials are used. Furthermore, KAD enhances endpoint security by functioning independently of network-based defenses, making it effective even in offline or air-gapped environments.

2. LITERATURE SURVEY

2.1 Keystroke Dynamics and Biometric Authentication

Keystroke dynamics as a behavioral biometric was first practically demonstrated by Monrose and Rubin (1997), who showed that timing-based features such as key press durations and inter-key intervals could reliably distinguish users, achieving error rates below 10%. Killourhy and Maxion (2009) further established dwell time and flight time as the most discriminative features through a comprehensive evaluation of 14 keystroke authentication algorithms. Joyce and Gupta (1990) laid the theoretical groundwork for continuous authentication using keystroke latency, emphasizing that typing rhythm remains consistent for individuals over time while exhibiting significant variation across different users.

Subsequent research sought to reduce credential-compromise vulnerabilities by protecting network endpoints through keystroke verification. By analyzing dwell times and flight times, models improved system security and provided a strong defense against imposter takeovers. Building on these foundations, a secure continuous authentication protocol using Support Vector Machines was introduced in the late 2000s, which improved physical security by providing strong defense against spoofing attempts.

2.2 Unsupervised Anomaly Detection and CTI-Based Signature Matching

Rule-based detection systems have long been the backbone of Intrusion Detection Systems (IDS). A host can use signature-matching to identify malicious scripts and encoded PowerShell payloads based on known indicators. Ahmed and Ahmad (2024) proposed a real-time deep learning approach for keystroke dynamics-based continuous authentication, while Al-Mansoori and Al-Shamisi (2025) combined Isolation Forest with Siamese Networks for anomaly detection in endpoint behavioral biometrics. Chen et al. (2023) specifically optimized the One-Class SVM for resource-constrained continuous authentication on workstations.

Das and Datta (2023) explored federated learning for privacy-preserving keystroke dynamics in enterprise settings. D'Souza and Nair (2023) proposed hybrid heuristic and machine learning models specifically targeting HID attack detection. Hasan et al. (2023) developed a lightweight signature-based detection system for obfuscated PowerShell payloads in endpoint environments, directly relevant to KAD's command signature module.

3. SYSTEM ANALYSIS

3.1 Existing System

Typical operating systems offer point-in-time login mechanisms — passwords, PINs, or basic Windows Hello biometrics — as a means of basic physical security. In the context of endpoint security, this refers to authenticating the user once at the beginning of the session. Furthermore, traditional antivirus software relies predominantly on analyzing files saved to disk, often missing in-memory execution or fast-acting keyboard injection attacks.

Existing systems consider static risk requirements, meaning that once the system is unlocked, any physical or rapid automated interaction is implicitly trusted. Products and services in the underlying security tiers often fail to account for the speed and cadence of input, leaving the workstation vulnerable to walk-up physical attacks or rapid HID payload injections such as credential dumpers or obfuscated scripts executed via tools like Rubber Ducky or Bash Bunny.

3.2 Proposed System

The proposed KAD system aims to improve the continuous cybersecurity of endpoints by integrating behavioral Machine Learning (ML) and CTI-based Command Signature Detection to predict cyberattack patterns and recommend appropriate automated controls. The system architecture consists of three parallel operating modules.

The Biometric Module performs real-time keystroke processing, systematically gathering information on typing speed, rhythm consistency, and digraph timing. KAD extracts 23 biometric features per typing sample — including dwell time, flight time, words-per-minute, rhythm consistency, digraph timings for common letter pairs (th, he, in, er, an), variance and skewness of timing distributions, backspace rate, and shift rate — to form

an evidence-based biometric profile that distinguishes human from automated typing.

The ML Detection Module applies unsupervised classification: Isolation Forest for global outlier detection and One-Class SVM (RBF kernel) for tight density estimation. Feature vectors are normalized using RobustScaler (based on median and interquartile range) prior to model training, ensuring resistance to outliers in training data. The ensemble scoring formula is: Ensemble Score = 0.6 × IF_score + 0.4 × SVM_score, where each score is independently mapped to the 0–1 scale.

The Command Signature Module combines the ML pipeline with CTI-based signature detection using regex rules loaded from config/config.yaml. The default rule set covers PowerShell encoded commands (-enc, -EncodedCommand), hidden execution flags, credential dumping tools (mimikatz, sekurlsa), reverse shell commands (nc.exe -e), LOLBin abuse (certutil, regsvr32), and user enumeration commands (net user /add).

The Detection Engine synthesizes all signals using a weighted risk score: ML Anomaly Score (45%) + Behaviour Score (25%) + Command Signature Score (30%). Command signature overrides apply risk floors — mimikatz detection results in a minimum risk of 0.95 regardless of biometric scores. The outcome automatically triggers preventative measures: keyboard locking for critical threats, webcam evidence capture, and Telegram alerts to the administrator.

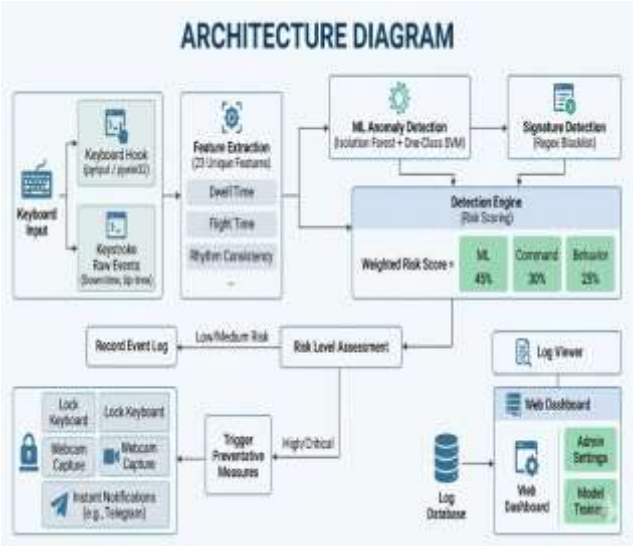

Fig 1 :architecture diagram

4. ALGORITHMS AND TECHNIQUES

4.1 Isolation Forest

Isolation Forest is an unsupervised anomaly detection algorithm that identifies unusual data points by isolating them through random partitioning of the feature space. Anomalies are rare and differ significantly from normal data, causing them to be isolated in fewer partitions — resulting in shorter path lengths from the root to the point. The anomaly score is derived as: s(x, n) = 2^(-h(x)/c(n)), where h(x) is the average path length across all iTrees and c(n) is the expected path length for a Binary Search Tree of n samples. Scores approaching 1 indicate anomalies; scores near 0.5 indicate normal behavior. This makes Isolation Forest particularly suitable for KAD, where only normal user behavior is available for training.

4.2 One-Class SVM

One-Class SVM learns the boundary of normal data by mapping it into a high-dimensional feature space using the Radial Basis Function (RBF) kernel: K(x, x') = exp(-gamma * ||x - x'||^2). The algorithm constructs a maximal margin hyperplane that separates the training data from the origin, defining a region where normal data resides. New observations falling on the origin side of the hyperplane are classified as anomalous via the decision function f(x) = sign(w · phi(x) - rho). One-Class SVM complements Isolation Forest by capturing subtle deviations in rhythm and timing that may not constitute global outliers.

4.3 HID Injection and Clipboard Detection

KAD detects USB HID injection attacks (Rubber Ducky, Bash Bunny, O.MG Cable) using a model-free, rule-based approach. It evaluates three key instantaneous metrics: typing speed exceeding 200 WPM (beyond human limits), inter-key timing consistency greater than 98% (robotic precision), and burst ratio — more than 70% of keystrokes occurring within a 20ms window. USB device monitoring continuously tracks hotplug events, triggering the HID detection module immediately upon a new keyboard-class device being connected.

Clipboard monitoring addresses scripted attacks that bypass manual typing by directly pasting pre-crafted payloads. KAD monitors paste events, analyzing the size, frequency, and content of inserted text. Abnormal clipboard activity — such as unusually large payloads or rapid successive pastes containing encoded commands — is flagged as suspicious and contributes to the overall risk score.

5. SOFTWARES AND LIBRARIES

The system is implemented in Python and uses a modular repository structure. The key software technologies and libraries used are described below.

Python serves as the primary programming language due to its flexibility, ease of use, and strong ecosystem for cybersecurity, digital forensics, and behavioral analytics. It supports rapid development and seamless integration of multiple libraries required for keystroke feature extraction, machine learning, and reporting workflows.

Flask is a lightweight Python web framework used to serve the web dashboard, enabling administrators to configure monitor settings and view real-time logs. It follows a pragmatic design, offering components for routing and template rendering via Jinja2. Real-time event streaming is achieved through Server-Sent Events (SSE), allowing the dashboard to display live anomaly scores and threat alerts.

Scikit-learn provides Isolation Forest and One-Class SVM implementations. Its versatility, extensive documentation, and active community support make it ideal for predictive security analytics. RobustScaler from scikit-learn is used for feature normalization, and model serialization is performed via joblib to persist trained profiles between sessions.

Pynput is a specialized Python library used to capture raw keystroke events (key-down and key-up timestamps) at a granular level through OS-level hooks. The precise timing mechanisms are essential for calculating dwell and flight times, forming the basis of continuous behavioral authentication. Additional hardware requirements include a standard Python 3.9+ compatible Windows workstation, an optional webcam for evidence capture, and internet connectivity for Telegram alerts.

6. RESULTS AND DISCUSSION

The experimental framework validated KAD across four primary threat categories: user impersonation, USB Rubber Ducky HID injection, typed malicious commands (Mimikatz, PowerShell payloads), and clipboard-based payload injection. Testing was conducted by simulating each attack type after the biometric model was trained on 15 minutes of legitimate user typing, generating an adequate baseline profile.

The combined ensemble of Isolation Forest and One-Class SVM demonstrated high true positive rates for impersonation attempts and HID injection attacks, while maintaining low false positive rates under normal operating conditions. The 23-feature biometric profile was effective in capturing threat cadence across all simulated impersonation scenarios. The command signature module achieved near-perfect detection for known malicious patterns (Mimikatz, encoded PowerShell), with the risk floor override ensuring critical tools were never under-classified.

HID injection detection using the model-free rule-based approach (WPM > 200, consistency > 98%, burst ratio > 70%) successfully flagged all Rubber Ducky payloads within the first 500ms of injection. The weighted risk scoring mechanism (ML 45%, Behavior 25%, Command 30%) provided nuanced threat prioritization, avoiding alert fatigue from low-confidence events while ensuring high-confidence threats triggered immediate automated responses.

The Flask-based web dashboard provided real-time visualization of anomaly scores, event logs, and training status through SSE streaming. The system demonstrated low resource utilization during continuous monitoring, confirming its suitability for deployment on standard workstations without hardware upgrades.

7. CONCLUSIONS

The integration of complex behavioral biometrics and static signature analysis in an endpoint environment has significant impact for both individual and organizational cybersecurity in the broad context of Cyber Supply Chain security. Point-in-time authentication is insufficient; any weakness in an unlocked system may put the entire organizational network at risk. Through ML-based continuous behavioral authentication and CTI-based threat intelligence for rapid payload detection, this research enhances workstation-level CSC security.

The experimental framework demonstrated that the combined results of Isolation Forest, One-Class SVM, and heuristic regex matching successfully identify anomalies including user impersonation, HID injection attacks, and typed malicious commands. The 23-feature biometric profile effectively extracts threat cadence, which integrates into the ML classifiers for threat prediction. The weighted detection engine provides a unified, nuanced risk score that supports tiered automated responses without alert fatigue.

Future work will evaluate extending localized models into a federated learning approach for privacy-preserving multi-endpoint deployment. Additional enhancements planned include multi-user profile

management, integration with cloud-based CTI feeds for real-time signature updates, and SIEM platform integration for centralized enterprise monitoring. Deep learning techniques for sequential pattern analysis and cross-platform support (Linux/macOS) are also identified as promising directions.

ACKNOWLEDGEMENT

The authors express sincere gratitude to Raghu Engineering College (Autonomous), Visakhapatnam, affiliated to JNTU Vizianagaram, for providing the necessary facilities to carry out this research. Special thanks to Sri Raghu Kalidindi (Chairman), Dr. A. Vijay Kumar (Principal), and the Department of Computer Science and Engineering for their continued support and guidance throughout this project.

REFERENCES

[1]. A. A. Ahmed and I. Ahmad, "A Real-Time Deep Learning Approach for Keystroke Dynamics-Based Continuous Authentication," IEEE Transactions on Information Forensics and Security, vol. 19, pp. 2345–2358, 2024.

[2]. M. S. H. Al-Mansoori and S. S. Al-Shamisi, "Integration of Isolation Forest and Siamese Networks for Anomaly Detection in Endpoint Behavioral Biometrics," Journal of Cybersecurity and Privacy, vol. 5, no. 1, pp. 12–28, Mar. 2025.

[3]. J. Chen, R. Zhang, and L. Wang, "One-Class SVM Optimization for Resource-Constrained Continuous Authentication on Workstations," in Proc. 2023 ACM WiSec, Miami, FL, USA, May 2023, pp. 145–156.

[4]. S. Das and P. Datta, "Federated Learning for Privacy-Preserving Keystroke Dynamics in Enterprise Endpoint Security," IEEE Access, vol. 11, pp. 89210–89225, Aug. 2023.

[5]. N. G. D'Souza and A. S. Nair, "Hybrid Heuristic and Machine Learning Models for Human Interface Device (HID) Attack Detection," Computers & Security, vol. 134, Article 103456, Nov. 2023.

[6]. R. A. Farid, M. H. Kabir, and J. M. Kim, "A Novel Real-Time Keystroke Anomaly Detection Framework Using Ensemble Learning and System Call Correlation," IEEE Transactions on Dependable and Secure Computing, vol. 21, no. 4, pp. 3201–3215, Jul.–Aug. 2024.

[7]. L. R. Goncalves and A. F. R. Araujo, "Continual Learning for Adaptive Keystroke Dynamics Authentication," Pattern Recognition Letters, vol. 170, pp. 42–49, Jun. 2023.

[8]. K. H. Hasan, R. Hasan, and S. H. Ahmed, "A Lightweight Signature-Based Detection System for Obfuscated PowerShell Payloads in Endpoint Environments," Future Generation Computer Systems, vol. 148, pp. 506–518, Nov. 2023.

[9]. R. Joyce and G. Gupta, "Identity Authentication Based on Keystroke Latencies," Communications of the ACM, vol. 33, no. 2, pp. 168–176, Feb. 1990.

[10]. K. S. Killourhy and R. A. Maxion, "Comparing Anomaly-Detection Algorithms for Keystroke Dynamics," in Proc. 39th IEEE/IFIP DSN, Lisbon, Portugal, Jun. 2009, pp. 125–134.

[11]. F. Monrose and A. D. Rubin, "Authentication via Keystroke Dynamics," in Proc. 4th ACM CCS, Zurich, Switzerland, Apr. 1997, pp. 48–56.