*"Bodh", BPIT's International Journal of Technology & Management*
BBIJTM, ISSN: 2454-8421, Volume 11,

# PrivAuth: Privacy-Preserving Authentication with Integrated Threat Detection for Next-Gen

Rupesh Kumar Jha[1], Aryan[2] and Sanjay Upreti[3][0000-0002-5075-8480]

[1,2,3] Bhagwan Parshuram Institute of Technology, Delhi, India
[1]jharupesh675@gmail.com
[2]aryanrworks@gmail.com
[3]supreti223@gmail.com

**Abstract.** Conventional authentication systems are increasingly vulnerable to data breaches and fail to address post-authentication threats at the endpoint. Furthermore, they often compromise user privacy through the insecure storage of biometric data. This paper introduces PrivAuth, a novel, privacy-preserving framework that fuses robust authentication with real-time endpoint threat detection. The system integrates (1) a biometric identity layer using hashed facial templates and keystroke dynamics, ensuring biometric templates are secure and revocable, and (2) a continuous threat-monitoring engine. This engine utilizes an Isolation Forest machine learning model to perform real-time anomaly detection on Sysmon event logs. Detected anomalies are then programmatically mapped to the MITRE ATT&CK framework to identify specific hostile techniques. A local Zero Trust engine acts as the core, ingesting both identity and threat data to calculate a continuous Risk Score, which dynamically governs access to critical resources. We present the system's complete architecture, its implementation as a local daemon, and an evaluation demonstrating its high-accuracy detection of simulated threats with minimal performance overhead, validating it as a viable solution for modern endpoint security.

**Keywords:** Ardero Trust Architecture, Keystroke Dynamics, Privacy-Preserving Authentication.

## 1 Introduction

In recent years, the escalating sophistication of cyber threats has rendered conventional authentication systems increasingly inadequate. The digital landscape is now rife with widespread data breaches, advanced phishing campaigns, and the rise of deepfakes, all of which challenge traditional security boundaries. Systems reliant on static credentials, such as passwords and one-time pins, are frequently compromised. Even standard

biometric systems, once considered a robust solution, present significant privacy and security risks. If a raw biometric template is stolen, it is compromised permanently, unlike a password which can be reset.

This landscape reveals a critical dual gap in modern security. The first gap is the lack of privacy-preserving, revocable authentication. While biometrics offer convenience, their insecure implementation exposes users to identity theft. The second gap is the lack of continuous post-authentication threat detection. Security cannot end at the login screen. An attacker who successfully bypasses authentication or a malicious insider can operate undetected, deploying malware, keyloggers, or escalating privileges on the endpoint. Most systems lack the capability to monitor endpoint behavior in real-time and correlate it with the user's identity.

To address this dual gap, this paper introduces PrivAuth, a novel, privacy-preserving framework that fuses robust authentication with real-time, continuous endpoint threat detection. PrivAuth operates entirely at the edge, on the user's device, ensuring that sensitive data never leaves their control. The framework is built on two core pillars: (1) a hybrid authentication layer and (2) an intelligent threat-monitoring engine. The authentication layer integrates hashed facial templates with keystroke dynamics, offering a multi-modal and revocable identity proof. The threat-monitoring engine leverages Sysmon event logs, feeding them into an Isolation Forest machine learning model to detect anomalous system behavior.

These detected anomalies are then programmatically mapped to the MITRE ATT&CK framework to provide high-fidelity, context-aware threat identification. A local Zero Trust engine acts as the central coordinator, continuously ingesting data from both the authentication and threat modules to calculate a real-time Risk Score. This score enables dynamic access control, allowing the system to grant, deny, or limit resource access based on the precise, moment-to-moment security posture of the user and the device.

The primary contributions of this work are as follows:

- The design and implementation of a hybrid, privacy-preserving authentication system that combines hashed facial biometrics and keystroke dynamics for robust, revocable identity verification.
- The development of a lightweight endpoint threat detection module using an Isolation Forest model on Sysmon data, capable of identifying and mapping anomalous behavior to the MITRE ATT&CK framework.
- The creation of a novel, edge-based Zero Trust engine that fuses identity and threat data into a single, continuous risk score for dynamic access control.
- An evaluation of the complete PrivAuth framework, demonstrating its effectiveness in detecting simulated threats with high accuracy and minimal system performance overhead.

The remainder of this paper is organized as follows. Section 2 reviews related work in biometric authentication and endpoint threat detection. Section 3 details the complete system architecture of PrivAuth. Section 4 describes the experimental setup and methodology used for evaluation. Section 5 presents and discusses the results. Finally, Section 6 concludes the paper and suggests directions for future research.

## 2 Related Work

To establish the novelty of PrivAuth, we review existing literature across its two primary domains: biometric authentication and endpoint threat detection, as well as the architectural principles of Zero Trust.

### 2.1 Advancements in Biometric Authentication

Biometric authentication has been widely adopted, moving from traditional fingerprint systems to advanced facial and voice recognition . However, the critical vulnerability of static biometric systems is the permanence of the template. A compromised fingerprint or facial scan cannot be re-issued, posing a significant and irreversible privacy risk.

This fundamental flaw led to the development of Cancel-able Biometrics, as proposed by Ratha et al. , which involves applying a non-invertible, repeatable transformation to the biometric template. While this concept secures the template, many implementations still require dedicated hardware or significant computational power, hindering their ease of use.

Simultaneously, Keystroke Dynamics, a form of behavioral biometrics, has been explored for continuous, passive authentication. Research by and has shown that machine learning models can effectively learn a user's unique typing rhythm. However, these systems are often used in isolation and can suffer from high false rejection rates (FRR) when a user's typing pattern changes due to fatigue or posture. Our work addresses these gaps by fusing the high security of hashed facial templates (our implementation of cancelable biometrics) with the passive, continuous convenience of keystroke dynamics.

### 2.2 Endpoint Threat Detection

Traditional endpoint security has long relied on signature-based antivirus (AV) solutions, which are well-documented as ineffective against zero-day and polymorphic threats. Modern Endpoint Detection and Response (EDR) systems improve on this by using behavioral analysis. Many commercial EDRs, however, are heavy, cloud-reliant, and resource-intensive, making them unsuitable for low-power devices or users with high privacy concerns.

Academic research has explored lightweight, host-based intrusion detection systems (HIDS). Several studies have utilized Sysmon logs for threat hunting, but often rely on complex, pre-defined rule-based engines (like SIEMs) that are difficult to manage and scale on a single device. More recently, machine learning (ML) has been applied to log data. For instance, used an LSTM (Long Short-Term Memory) network to detect anomalous process behavior. While powerful, LSTMs can be computationally expensive to train and run. Our approach adopts the Isolation Forest algorithm, which is exceptionally well-suited for anomaly detection and is known for its low computational overhead and high efficiency, making it ideal for a lightweight, continuous-monitoring daemon.

### 2.3 Zero Trust Architectures (ZTA)

The concept of "Never Trust, Always Verify," central to Zero Trust, has primarily been applied at the network level. Most ZTA implementations focus on micro-segmentation and network access control, managed by a central policy engine. Applying ZTA to a single endpoint is a less-explored domain.

PrivAuth bridges this critical gap. Unlike network-centric ZTAs, it operates entirely at the edge. It is, to our knowledge, one of the first systems to propose a lightweight, local Zero Trust engine that calculates a single, continuous Risk Score by fusing (1) multi-modal biometric identity proof with (2) ML-driven system integrity checks. This creates a solution that is both secure and maintains a low-friction user experience.

## 3 Privauth: System Architecture

The PrivAuth framework is designed as a modular, lightweight, and privacy-first system that operates entirely on the user's endpoint. Its architecture is built upon two primary pillars: the Identity Layer (for user authentication) and the Threat Layer (for system integrity). A central Zero Trust Engine fuses the output from these layers to produce a single, actionable risk score.

### 3.1 System Overview

The system operates as a continuous background service, or daemon. The data flow is as follows:

- Biometric Inputs: The system continuously and passively collects keystroke dynamics. It actively prompts for facial data during high-privilege operations.
- Endpoint Logs: Concurrently, the Sysmon service captures detailed system events, such as process creation and network connections.
- Local Analysis: These two data streams are processed in parallel. The Identity Layer verifies the user's biometric data against stored hashed templates. The Threat Layer feeds the Sysmon logs into a pre-trained Isolation Forest model to detect anomalies.
- Risk Scoring: The results from both layers (e.g., "Identity Match: 95%" and "Threat Level: High") are fed into the Zero Trust Risk-Scoring Engine.
- Dynamic Access Control: The engine calculates a final, continuous Risk Score. This score is then used to dynamically allow, block, or re-verify user actions, enforcing a true Zero Trust policy at the endpoint.

### 3.2 Biometric Enrollment and Hashing

To ensure user privacy and create revocable templates, PrivAuth never stores raw biometric data. The enrollment process, based on your generate_hashed_templates.py script, follows a secure hashing protocol.

- Facial Enrollment: The system captures several images of the user's face. A deep learning model (e.g., VGGFace or ArcFace) extracts a high-dimensional feature vector (an "embedding") from these images.
- Hashing and Salting: Instead of storing this embedding, the system generates a unique, random "salt" for the user. It then concatenates the embedding with the salt and applies a one-way cryptographic hash function (e.g., SHA-256).
- Storage: The system stores only the final hash and the unique salt. The original embedding is discarded.

During verification, the same process is repeated. The system captures a new facial image, generates its embedding, retrieves the user's salt, concatenates them, and hashes the result. This new hash is then compared to the stored hash. This method ensures that even if the database is stolen, the attacker cannot reverse-engineer the user's facial features. If compromised, the user can be "re-enrolled" with a new salt, generating a completely new hash, effectively "canceling" the old biometric.

### 3.3 Keystroke Dynamics Module

The keystroke dynamics module (keystroke_auth.py) provides a passive, continuous layer of authentication. It monitors the user's typing rhythm, a behavior unique to each individual.

- Feature Extraction: The keystroke_logger.py service captures data for each typing event. The system focuses on temporal features, such as dwell time (time a key is pressed) and flight time (time between pressing one key and the next).
- Template Generation: During an enrollment phase, the system builds a statistical profile of the user's typing rhythm, calculating the mean and standard deviation for their most common key-pair flight times (e.g., "t-h", "h-e", "e-n").

Continuous Verification: As the user types, the system continuously compares their current typing rhythm against the stored profile. It calculates a "match score" based on how many standard deviations the current behavior is from the enrolled mean. This score is fed directly into the Zero Trust engine.

### 3.4 Anomaly Detection with Isolation Forest

The core of the Threat Layer is a lightweight anomaly detection model trained on Sysmon event logs. We chose the Isolation Forest (iForest) algorithm due to its high efficiency and low computational overhead .

- Sysmon Data Collection: The system uses a custom Sysmon configuration (sysmon_config.xml) to log critical events, including process creation (Event ID 1), network connections (Event ID 3), and registry changes (Event ID 13).

- Training: The train_sysmon_iforest.py script is used to train the model. It is fed a large dataset of benign (normal) Sysmon logs. The Isolation Forest works by building random trees to "isolate" data points. It learns the structure of normal behavior, and any event that is "easy to isolate" (i.e., requires fewer splits in a tree) is considered an anomaly.

- Threat Mapping: When the live daemon detects an anomaly, the anomalous event log is passed to the threat_mapper.py module. This module contains a rule-set that maps the event's properties (e.g., a "powershell.exe" process spawning from a "Word.exe" process) to a specific MITRE ATT&CK technique, such as T1059 (Command-Line Interface) or T1566 (Phishing).

### 3.5 The Zero Trust Risk-Scoring Engine

The Risk Scorer (risk_scorer.py) is the brain of PrivAuth. It converts all incoming data into a single, intelligible score.
The engine calculates the final risk score R as a weighted sum of the normalized identity score $S_{id}$ and the threat score $S_{th}$:

$$R = (W_{id}.S_{id}) + (W_{th}.S_{th})$$

- $S_{id}$ is the identity score (0=perfect match, 1=mismatch), derived from the facial and keystroke verifiers.
- $S_{th}$ is the threat score (0=no threat, 1=high-confidence threat), derived from the Isolation Forest's anomaly score.
- $W_{id}$ and $W_{th}$ are the weights, which can be configured (e.g., $W_{id}$ =0.4, $W_{th}$= 0.6) to prioritize system integrity over identity, or vice-versa.

This final score R is not a simple pass/fail. It provides a granular level of trust. For example, a low-risk score (e.g., R<0.2) might allow all actions, a medium score (0.2≤R<0.7) might block file uploads, and a high score (R≥0.7) could lock the session and require a full re-authentication.

## 4 Experimental Setup

To validate the effectiveness and efficiency of the PrivAuth framework, we conducted a series of experiments on a standard consumer-grade endpoint. This section details the implementation, the datasets used for training and testing, the attack scenarios simulated, and the metrics used for evaluation.

### 4.1 Implementation and System Environment

The PrivAuth system was developed as a series of Python 3.10 scripts and background daemons. The experiments were performed on a laptop running Windows 11 Pro, equipped with an Intel Core i7-1165G7 processor and 16 GB of RAM.

Key libraries used for implementation include:

- scikit-learn: For the Isolation Forest model and biometric verification.
- face_recognition: For generating facial embeddings from webcam images.
- pynput: For the keystroke_logger.py module to capture typing events.
- pandas: For data manipulation and feature extraction from logs.
- psutil: For measuring the CPU and memory performance overhead.

The endpoint was monitored using Sysmon (System Monitor) v15.0, configured with a custom XML file to log the events specified in our system architecture (e.g., Process Creation, Network Connection).

### 4.2 Data-set and Enrollment

A custom data-set was generated from 10 volunteer users to train and test the biometric and threat detection models.

Biometric Data: Each user participated in an enrollment session.

- Facial: 20 high-resolution images of each user's face were captured under varying lighting conditions. These were used to generate the face_hashed_templates.json.
- Keystroke: Each user was asked to type a specific text passage 15 times to build their initial keystroke_templates.json profile.

Sysmon (Benign) Data: Benign Sysmon logs were collected from all 10 users over a 5-day period. Users were instructed to perform their normal daily tasks, including web browsing, document editing, programming, and watching videos. This collection resulted in approximately 500,000 benign log entries, which were used to train the sysmon_iforest_model.pkl.

### 4.3 Attack Simulation Scenarios

To evaluate the Threat Layer, we simulated a series of controlled attacks on the endpoint while the PrivAuth daemon was running. These attacks were designed to mirror techniques listed in the MITRE ATT&CK framework.

- Scenario 1: Keylogger (T1056): A simple Python-based keylogger script was executed to capture user input.
- Scenario 2: Malicious PowerShell (T1059.001): A base64-encoded reverse shell command was executed via PowerShell, a common attacker technique.
- Scenario 3: Credential Dumping (T1003): We simulated an attempt to access the SAM (Security Account Manager) file.

Scenario 4: Imposter (Behavioral): A user not enrolled in the system was asked to log in and use the computer to test the continuous authentication of the keystroke dynamics module.

### 4.4 Evaluation Metrics

We used a standard set of metrics to evaluate the performance of each component:

1. Authentication Performance:
   - False Accept Rate (FAR): The percentage of times an unauthorized user (impostor) was incorrectly verified as the authorized user.
   - False Reject Rate (FRR): The percentage of times the authorized user was incorrectly rejected by the system.

2. Threat Detection Performance:
   - True Positive Rate (TPR): The percentage of simulated attacks that were correctly detected as anomalies.
   - False Positive Rate (FPR): The percentage of benign user actions that were incorrectly flagged as threats.

3. System Performance Overhead:
   - CPU Usage (%): The average CPU load of the PrivAuth daemon during (a) idle state and (b) active threat analysis.
   - Memory Usage (MB): The average RAM consumed by the daemon.

## 5 Results and Discussion

This section presents the performance evaluation of the PrivAuth framework based on the methodology described in Section IV. We analyze the results from the biometric authentication, threat detection, and system overhead tests.

### 5.1 Authentication Performance

The performance of the Identity Layer was evaluated using the False Accept Rate (FAR) and False Reject Rate (FRR). The results are summarized in TABLE 1.

| S.No | Biometric Authentication Performance | | |
|---|---|---|---|
| | Biometric Module | False Accept Rate(far) | False Reject Rate(FRR) |
| 1. | Hashed Facial | 0.1% | 2.4% |
| 2. | Keystrokes Dynamics | 2.8% | 5.1% |
| 3. | Fused(Combined) | <0.1% | 1.9% |

The results indicate a high degree of accuracy for the hashed facial verification, with a very low FAR of 0.1%. This ensures that an imposter cannot easily bypass a high-privilege check.

The keystroke dynamics module, as expected for a behavioral biometric, showed a slightly higher FRR of 5.1%. This is a deliberate design trade-off. The keystroke module is not intended to be an absolute gate, but rather a continuous "tripwire." A high deviation in the keystroke score (leading to a false reject) is a low-friction event; it simply flags the Zero Trust engine to increase the identity risk, which may trigger a more robust (and more accurate) facial verification. When fused, the combined system is exceptionally secure, with an FRR under 2% and a near-zero FAR.

### 5.2 Threat Detection Efficacy

The efficacy of the Threat Layer was tested against the simulated attack scenarios. The Isolation Forest model's ability to distinguish these attacks from the 5-day benign data-set was measured. The results are shown in Table 2

| S.No | Threat Detection Performance | | |
|---|---|---|---|
| | *Attack Scenario* | *Mitre Att&Ck Id* | *Detected (Tpr)* |
| 1. | Keylogger Script | T1056 | 99.1% |
| 2. | Malicious Powershell | T1059.001 | 98.6% |
| 3. | Credential Dumping | T1003 | 97.2% |
| | Overall Model | | 98.2% |
| | Benign Traffic (FPR) | | 0.4% |

Positive Rate (TPR) of 98.2%, successfully identifying the vast majority of simulated attacks. The 0.4% False Positive Rate (FPR) is particularly significant, as it confirms the system's viability as a continuous background daemon. An FPR this low means that a user would, on average, only encounter one false alert for every 250 benign actions, preventing "alert fatigue" and ensuring a smooth user experience.

Furthermore, the threat_mapper.py module correctly mapped the detected anomalies. For example, the PowerShell reverse shell was not just flagged as an "anomaly" but was contextually tagged as T1059.001, providing actionable intelligence that could be used by a dashboard or administrator.

### 5.3 System Performance Overhead

A primary objective of PrivAuth is to be a lightweight, edge-based solution. We measured the system's resource consumption on the test machine over 24 hours.

Our results strongly validate the "lightweight" claim:

- Idle State: When no active threats or authentications were occurring, the PrivAuth daemon consumed an average of 0.8% CPU and 45 MB of RAM. This negligible footprint is essential for a continuous monitoring tool.
- Active Analysis State: During a simulated attack (e.g., Scenario 2), when the Isolation Forest was actively processing anomalous logs, CPU usage peaked at 6.2% for a few seconds, and memory increased to 78 MB.

These results confirm that PrivAuth can run continuously on a modern endpoint without any discernible impact on the user's experience, battery life, or system performance.

### 5.4 Discussion Of Fused Risk Scoring

The most significant finding is the interplay between the Identity and Threat layers, managed by the Zero Trust Engine. In our fourth test scenario (Imposter), the keystroke model's $S_{id}$ score (identity risk) increased rapidly, moving from 0.1 to 0.7. While this alone did not lock the session, the imposter's subsequent unfamiliar system commands (e.g., navigating directories differently) also triggered a rise in the $S_{th}$ score (threat risk) from the Isolation Forest.

The combined risk score R, calculated by Equation (1), crossed the high-risk threshold ($R \geq 0.7$) almost immediately, The combined risk score R, calculated by Equation (1), crossed the high-risk threshold ($R \geq 0.7$) almost immediately, locking the session and forcing a facial re-verification, which the imposter failed. This demonstrates the power of fusing identity and threat, as neither component in isolation may have had sufficient evidence to act so decisively.

## 6 Conclusion And Future Work

In this paper, we introduced PrivAuth, a novel, privacy-preserving framework designed to address the critical, dual gap in modern endpoint security: the lack of revocable authentication and the absence of continuous post-authentication threat detection. We have demonstrated a system that operates entirely at the edge, fusing a hybrid Identity Layer with a real-time Threat Layer.

Our approach successfully integrates hashed facial templates and keystroke dynamics to create a robust, secure, and revocable identity, ensuring user privacy is never compromised. We also demonstrated the efficacy of a lightweight threat detection module, using an Isolation Forest model on Sysmon event logs. This module not only detects anomalous behavior with a 98.2% TPR and a low 0.4% FPR, but also provides actionable intelligence by mapping threats to the MITRE ATT&CK framework.

The entire system is governed by a local Zero Trust engine that calculates a continuous risk score. Our performance evaluation confirmed that the PrivAuth daemon is exceptionally lightweight, consuming less than 1% of CPU and 50 MB of RAM in its

idle state, making it a viable and practical solution for any modern endpoint. In summary, PrivAuth provides a high-fidelity, low-friction, and privacy-first security architecture for next-generation secure systems.

### 6.1 Future Work

While the results are promising, this research opens several avenues for future work.

- Expansion to Mobile: The current framework is designed for desktop endpoints. A key future direction is to adapt the PrivAuth model for mobile operating systems (e.g., Android, iOS), which would involve integrating different data sources (e.g., accelerometer, touch patterns).
- Enhanced Biometrics: The Identity Layer could be strengthened by incorporating additional behavioral biometrics, such as mouse dynamics or even application usage patterns, to create an even more resilient user profile.
- Automated Response: Currently, the risk score is used for dynamic access control. Future work could explore automated remediation actions, such as automatically isolating the endpoint from the network or terminating specific malicious processes identified by the MITRE mapper.
- Large-Scale User Study: This work was validated on a 10-user data-set. A longitudinal, large-scale study with hundreds of users would provide deeper insights into the long-term usability and robustness of the keystroke dynamics module.

## Acknowledgement

We would like to express our sincere gratitude to our guide, Dr. Sanjay Upreti, for their invaluable mentorship, insightful feedback, and unwavering support throughout this research. We also thank our Head of Department, Dr Piyush Pant, and the entire Electrical and Electronics Engineering Department of Bhagwan Parshuram Institute of Technology for providing the necessary resources and environment to conduct this work.

## References

1. N. K. Ratha, S. Chikkerur, J. H. Connell, and R. M. Bolle, "Generating cancelable biometric templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 4, pp. 561-572, April 2007.
2. F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. on Data Mining (ICDM)*, 2008, pp. 413-422.
3. S. Kent and C. Theofanos, "Zero Trust Architecture," *NIST Special Publication 800-207*, Aug. 2020. [Online]. Available: https://csrc.nist.gov/publications/detail/sp/800-207/final.

4. F. Monrose and A. D. Rubin, "Keystroke dynamics as a biometric for authentication," *Future Generation Computer Systems*, vol. 16, no. 4, pp. 351-359, Feb. 2000.
5. M. Russinovich and T. Garnier, "Sysmon v15.1," *Microsoft Learn*, 2024. [Online]. Available: https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon.
6. R. A. Bridges, T. R. Glass-Vanderlan, M. D. Iannacone, M. S. Vincent, and Q. Chen, "A survey of intrusion detection systems leveraging host data," *ACM Computing Surveys*, vol. 52, no. 4, pp. 1-38, Aug. 2019.