Suspicious Login Detection Using Python

Prachi Ghoghari
Email: prachipatel0848@gmail.com
GitHub: https://github.com/prachi01645
Date: 2026

1.Abstract

This project implements a Python-based system to detect suspicious login attempts based on login time, location, and device type. It flags logins that deviate from normal behavior and provides a summary of potential security risks. The project demonstrates practical skills in Python scripting, anomaly detection, and independent problem-solving.

2.Introduction

With the increasing reliance on digital platforms, monitoring login activity for unusual or suspicious patterns is crucial for security. This project creates a lightweight, interpretable system to identify such logins. The system is fully self-contained and designed to demonstrate **practical cybersecurity and ML skills**.

3.Methodology

1. **Dataset:** A simulated set of login records, including time, location, and device type.
2. **Detection Logic:**
   - Logins outside standard hours (6 AM – 10 PM) are flagged.
   - Logins from unusual locations or devices are also considered suspicious.
3. **Implementation:** Pure Python, no external libraries required.
4. **Summary Metrics:** Calculates total logins flagged and percentage of suspicious logins.
5. **Optional ML Extension:** Can be enhanced with anomaly detection models such as Isolation Forest for larger datasets.

4.Example Output

Suspicious Login Detection Output:

User 1: Normal (Login at 08:15 from India)

User 2: Suspicious (Login at 02:30 from USA)

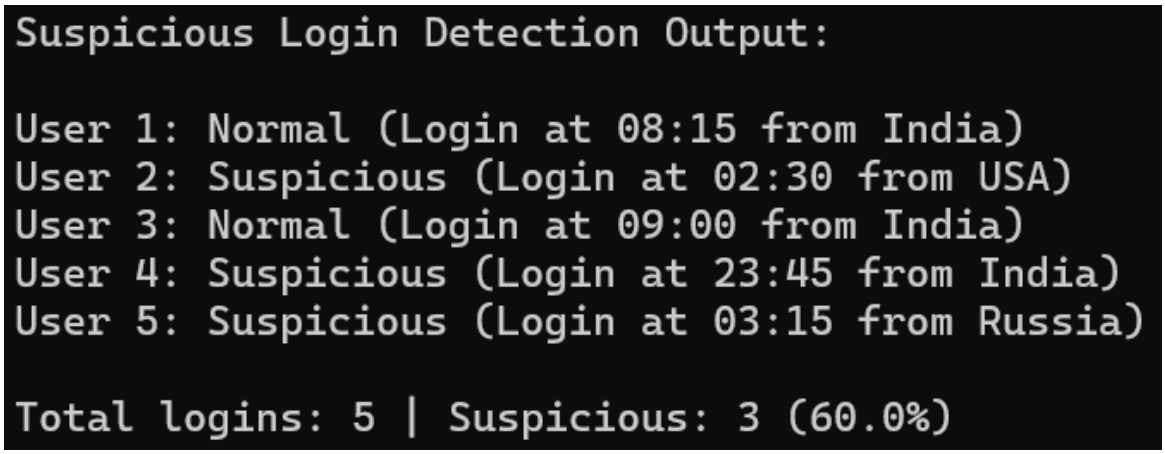
User 3: Normal (Login at 09:00 from India)

User 4: Suspicious (Login at 23:45 from India)

User 5: Suspicious (Login at 03:15 from Russia)

Total logins: 5 | Suspicious: 3 (60.0%)

- “Normal” = Login follows expected behavior
- “Suspicious” = Login deviates from typical patterns



5.Observations

- The system effectively flags anomalous logins using simple rules.
- Demonstrates the **trade-off between simplicity and accuracy** — while effective for small datasets, rule-based detection may miss context in real-world scenarios.
- Shows how Python can be used for **ethical cybersecurity prototyping**.

6.Applications

- Educational: Teaches Python scripting, anomaly detection, and cybersecurity concepts.
- Research: Acts as a prototype for developing more advanced login monitoring systems.
- Industry: Can serve as a foundation for real-world monitoring tools for online platforms and internal systems.

7.Insights

- This project highlights **independent problem-solving**, creating a functional system without external libraries.
- Encourages future expansion to include **streaming data, multi-factor analysis, and anomaly detection models**.
- Can be adapted for **larger datasets** or integrated into web dashboards for real-time monitoring.

8.Conclusion

The project demonstrates an independent implementation of a suspicious login detection system in Python. It is self-contained, easy to run, and provides a foundation for learning, experimentation, and further development in cybersecurity and anomaly detection.

9.Further Observations

- The rule-based detection is **simple, fast, and interpretable**, making it ideal for small datasets and educational purposes.
- Flagging based on login time, location, and device highlights **common patterns in suspicious behavior**, even without advanced ML.
- Shows **limitations of static rules** — unusual logins may not always indicate malicious activity, emphasizing the need for context-aware detection.
- Demonstrates the **importance of preprocessing and standardizing input data** (e.g., time format, location names) for accurate results.
- Provides a **foundation for scaling up**: larger datasets, anomaly detection models, or real-time monitoring systems.
- Can be used as a **teaching tool** for Python, cybersecurity basics, and anomaly detection concepts.
- Highlights **independent problem-solving** and the ability to build functional prototypes **without external libraries**.