



Keylogger Detection and Termination System

Presented By:

Name: Bittu Kumar

Reg. No.: 222025109271

Department: B.Tech(CSE)

Semester: 7th

Presented To:

Guide: Mrs. Shibya Swaroop

HOD: Dr. Naghma Khatoon

Date: 18 Nov 2025

Problem Statement

Challenge:

Keyloggers represent one of the most dangerous and stealthy types of malware, silently recording every keystroke and compromising sensitive data.

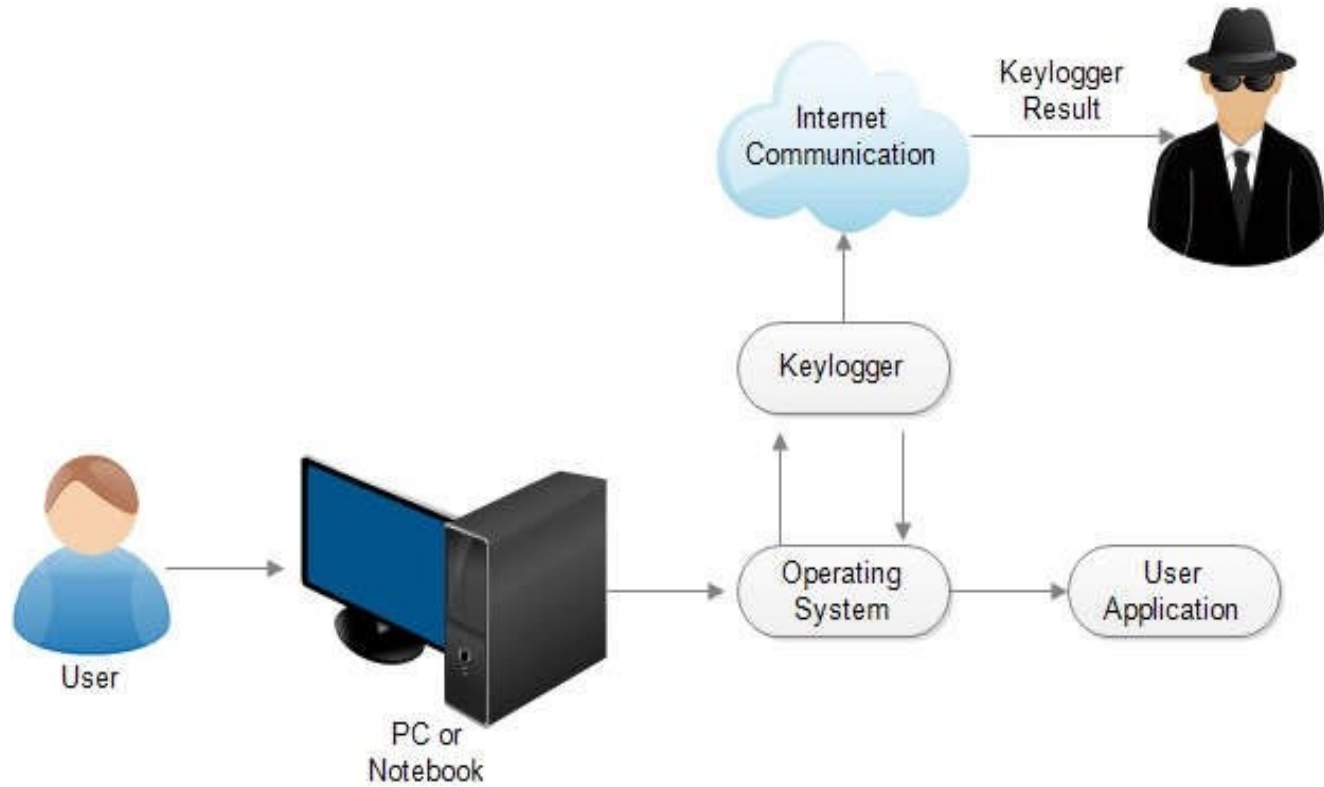
Key Issues:

- > Some keyloggers operate at kernel level, remaining undetected
- > Current systems Lack behavioral and proactive defense mechanisms
- > Resource-heavy detection systems discourage continuous monitoring

Impact

Financial theft, identity fraud, unauthorized access, and data breaches affecting individuals and enterprises

Working of Keyloggers



Project Objectives

1) Intelligent Detection Framework

Develop behavioral and heuristic-based analysis to identify keyloggers through actions rather than code signatures

2) Decoy Input Injection

Implement proactive defense by injecting virtual keystrokes to trap hidden keyloggers

3) Minimize False Positives

Use decision fusion models to optimize detection accuracy with minimal false alarms

4) User-Friendly Interface

Create intuitive dashboard for monitoring, analysis, and forensic review

System Architecture

1

Input Capture Layer

Low-level monitoring of keyboard events with minimal overhead

2

Detection Engine

Behavioral analysis for comprehensive threat identification

3

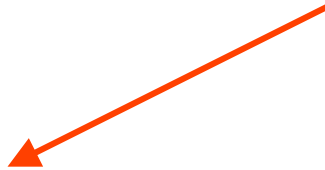
Process Monitoring

Constant surveillance of processes, system hooks, and API access patterns

4

Decision Fusion and Response

Intelligent aggregation and automatic threat neutralization



Detection Components

1) Suspicious Process Scanner

- > Multi-factor threat scoring based on naming, path, and resource usage
- > Categorization into **Medium**, **High**, and **Critical** levels

2) Startup & Persistence Monitor

- > Scans `~/.config/autostart/`, `systemd` services, and cron jobs
- > 80% detection rate of autostart configurations in testing

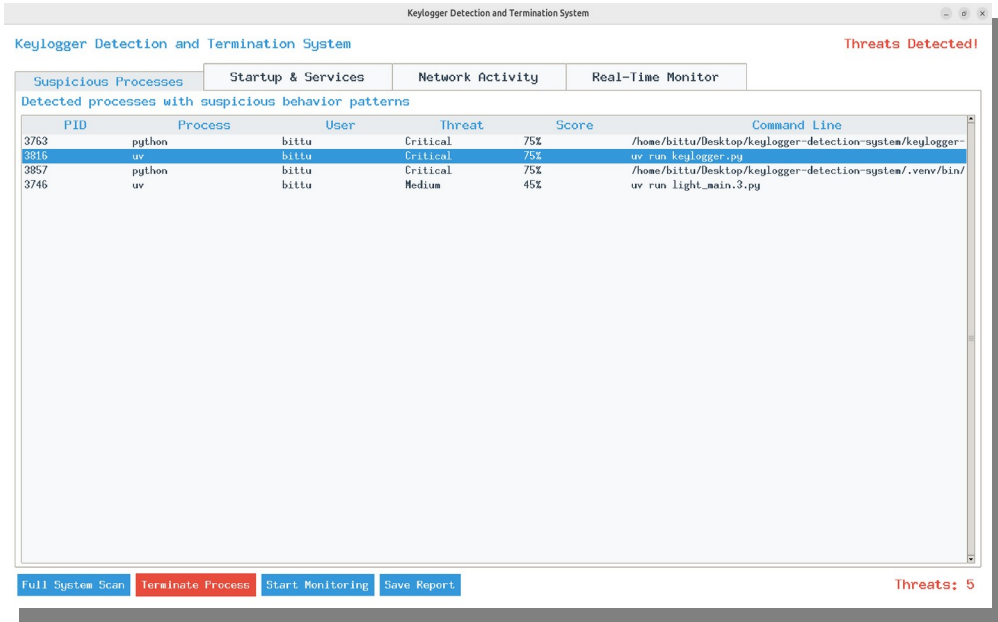
3) Network Activity Analyzer

- > Monitors external connections from suspicious processes
- > Flags potential data exfiltration attempts

4) Input Device Access Monitor

- > Tracks processes with open handles to `/dev/input/*` devices
- > Identifies both hardware and software-based keystroke capture

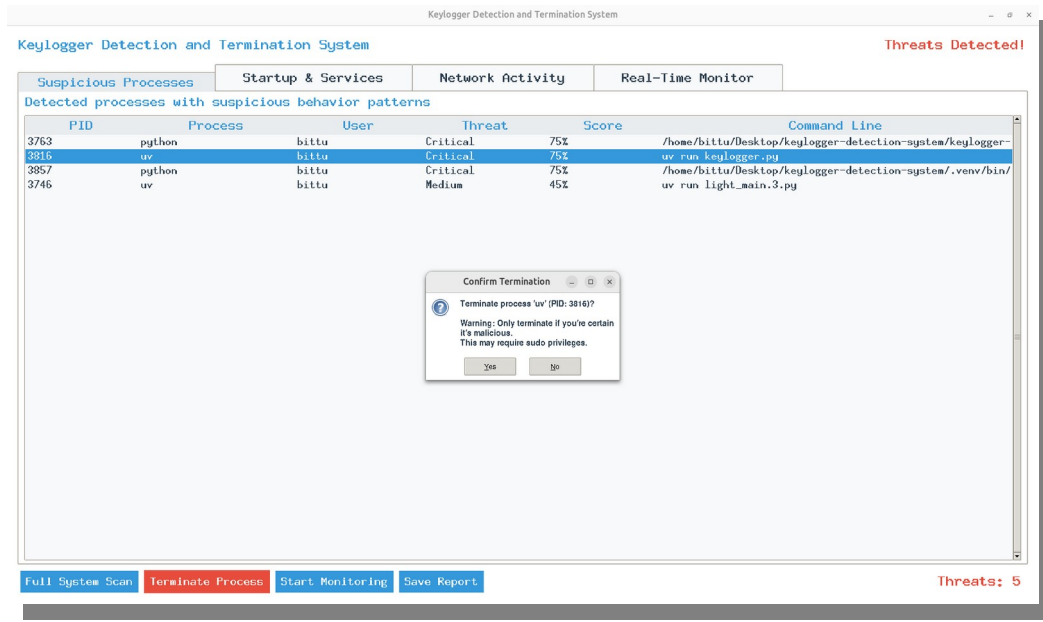
System Interface



The screenshot shows the 'Keylogger Detection and Termination System' window. The 'Suspicious Processes' tab is active, displaying a table of detected processes with suspicious behavior patterns. The table has columns for PID, Process, User, Threat, Score, and Command Line. The processes listed are python (PID 3763), uv (PID 3816), python (PID 3857), and uv (PID 3746). The uv processes are highlighted in blue. The bottom of the window shows a status bar with 'Threats: 5' and a row of buttons: 'Full System Scan', 'Terminate Process', 'Start Monitoring', and 'Save Report'.

PID	Process	User	Threat	Score	Command Line
3763	python	bittu	Critical	75%	/home/bittu/Desktop/keylogger-detection-system/keylogger-
3816	uv	bittu	Critical	75%	uv run keylogger.py
3857	python	bittu	Critical	75%	/home/bittu/Desktop/keylogger-detection-system/.venv/bin/
3746	uv	bittu	Medium	45%	uv run light_main.3.py

1) Suspicious Process Detection



The screenshot shows the same 'Keylogger Detection and Termination System' window, but with a 'Confirm Termination' dialog box open. The dialog box asks 'Terminate process 'uv' (PID: 3816)?' and includes a warning: 'Warning: Only terminate if you're certain it's malicious. This may require sudo privileges.' There are 'Yes' and 'No' buttons at the bottom of the dialog. The background window shows the same table of suspicious processes as the first screenshot, with the status bar indicating 'Threats: 5'.

2) Suspicious Process Termination

Limitations

Advanced Evasion

*Sophisticated malware
with polymorphic code
and anti-debugging
can bypass detection*

Kernel-Level Threats

*Rootkit-level
keyloggers can hide
from user-space
detection*

False Positives

*Legitimate tools may
trigger alerts*

Resource Impact

*May affect older
hardware during
continuous monitoring*

Conclusion

The Keylogger Detection and Termination System successfully demonstrates that effective security monitoring is achievable through open-source solutions combining behavioral intelligence, proactive defense, and user-centric design.

Key Achievements:

- 1) Integrates behavioral analysis, process monitoring, and heuristic rules*
- 2) 80%+ detection rate with 30% false positives in testing*
- 3) Minimal resource consumption suitable for continuous monitoring*
- 4) Easy threat neutralization with a single click*
- 5) Intuitive interface making advanced security accessible to all users*

Thank You!