ISSN: **3107-6513**

# Real Time Soc Dashboard for Threat Monitoring

Kanishk Hariharan.S, Thamizharasan.N
Department of Computer Science, Rathinam College of Arts and Science (Autonomous), Coimbatore, Tamil Nadu, India

***Abstract***

*The rapid growth of digital technologies has significantly increased the complexity and frequency of cybersecurity threats faced by modern organizations. With the widespread use of cloud platforms, web applications, and interconnected systems, the risk of unauthorized access, brute-force attacks, malware infections, and distributed denial-of-service (DDoS) attacks has grown substantially. Security Operations Centers (SOCs) play a vital role in continuously monitoring and responding to such threats; however, enterprise-level SOC and Security Information and Event Management (SIEM) solutions are often expensive and resource-intensive, making them unsuitable for small organizations and academic environments.*

*This paper presents the design and development of a Real-Time SOC Dashboard for Threat Monitoring, a lightweight and cost-effective system that enables continuous log analysis, real-time threat detection, and centralized visualization. The system utilizes rule-based detection techniques to identify suspicious activities such as repeated failed login attempts and abnormal system resource usage. It also provides an interactive dashboard that displays critical security information including IP activity, CPU usage, and memory utilization. Alerts are generated when predefined thresholds are exceeded, enabling proactive threat identification and response. The proposed solution demonstrates how essential SOC functionalities can be implemented efficiently using open-source tools without requiring complex infrastructure, thereby improving situational awareness and reducing manual monitoring efforts.*

***Keywords***

*Real-Time Monitoring, Security Operations Center (SOC), Threat Detection, Log Analysis, Cybersecurity, Intrusion Detection, Dashboard Visualization, SIEM, Network Security, Anomaly Detection*

## 1.INTRODUCTION

In the modern digital era, organizations depend heavily on information systems to manage operations, communication, and data storage. As digital dependency increases, cyber threats have become more sophisticated and frequent, targeting vulnerabilities in authentication systems, network

configurations, and software applications. Attackers often use automated tools and advanced techniques to gain unauthorized access, steal data, or disrupt services. Traditional security mechanisms such as firewalls and antivirus software are no longer sufficient to handle these evolving threats, as they primarily focus on prevention rather than continuous monitoring.

A Security Operations Center (SOC) plays a critical role in modern cybersecurity by acting as a centralized unit responsible for monitoring, detecting, and responding to security incidents in real time. SOC dashboards provide a consolidated view of system logs, alerts, and performance metrics, enabling analysts to quickly identify anomalies and take appropriate actions. Despite their importance, implementing a full-scale SOC requires significant financial investment, advanced infrastructure, and

ISSN: 3107-6513

skilled personnel, which makes it impractical for small organizations and educational institutions.

The Real-Time SOC Dashboard for Threat Monitoring project addresses this limitation by providing a simplified and efficient monitoring system. It focuses on real-time log analysis, rule-based threat detection, and dynamic dashboard visualization. The system demonstrates core SOC functionalities such as log aggregation, anomaly detection, alert generation, and system health monitoring in an accessible and educational format.

## 2.PROBLEM STATEMENT

Many organizations generate large volumes of system logs from servers, authentication systems, and network devices, but these logs are often stored without active analysis. As a result, potential security threats may remain undetected until they cause significant damage. Manual log inspection is not only time-consuming but also prone to human error, making it an inefficient approach for modern cybersecurity requirements.

Additionally, most enterprise-level monitoring solutions are expensive and require complex configurations, which are beyond the capabilities of small organizations and academic environments. The lack of affordable and efficient monitoring systems creates a gap in cybersecurity preparedness. Therefore, there is a need for a simplified and cost-effective solution that can automatically analyze logs, detect anomalies, and provide real-time alerts in an easy-to-understand format. This project aims to address this gap by developing a lightweight SOC dashboard capable of continuous threat monitoring and efficient visualization.

## 3.OBJECTIVES OF THE PROJECT

The primary objective of this project is to design and implement a real-time security monitoring system that provides centralized visibility into system activities. The system aims to detect repeated failed login attempts, identify suspicious IP activity, and monitor system performance metrics such as CPU and memory utilization. By generating alerts when abnormal behavior is detected, the system enhances the ability to respond to potential threats proactively.

Another important objective is to demonstrate the architecture and functioning of a Security Operations Center in a simplified and educational manner. The project emphasizes modular design, allowing for easy scalability and future enhancements. It also focuses on minimizing resource consumption so that the system can operate efficiently on standard hardware without requiring advanced infrastructure. Through this approach, the project serves as both a learning tool and a practical solution for basic threat monitoring.

## 4.SYSTEM ARCHITECTURE

The architecture of the Real-Time SOC Dashboard is designed using a modular and layered approach to ensure flexibility, scalability, and maintainability. The system consists of multiple components that work together to achieve real-time monitoring and threat detection. The log collection module is responsible for gathering system logs and authentication records from the operating system.

The threat detection engine analyzes the structured logs using predefined rules and thresholds. It identifies suspicious activities such as repeated failed login attempts from a specific IP address within a certain time interval. The database management layer stores processed logs and alert records in an organized manner, allowing for efficient data retrieval and analysis.

ISSN: 3107-6513

The alert generation module is responsible for triggering notifications when suspicious activities are detected. These alerts help in identifying potential threats at an early stage. Finally, the dashboard visualization layer presents the collected data in a user-friendly format. It displays information such as IP addresses, login attempts, CPU usage, and memory utilization in real time, enabling users to monitor system activity effectively. This modular design ensures that each component functions independently while contributing to the overall performance of the system.

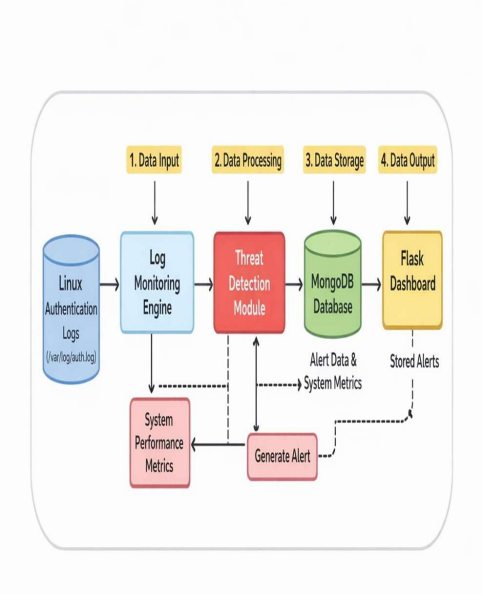


Fig 1. Data Flow Diagram

## 5.THREAT DETECTION METHODOLOGY

The threat detection methodology used in this project is based on a rule-based and threshold-driven approach. The system continuously monitors authentication logs to identify repeated failed login attempts. When the number of failed attempts from a particular IP address exceeds a predefined threshold within a specific time frame, the system flags the activity as suspicious and generates an alert. This method is effective in detecting brute-force attacks and unauthorized access attempts.

In addition to authentication monitoring, the system also tracks CPU and memory usage to identify unusual patterns. Sudden spikes in resource utilization may indicate the presence of malicious processes, automated attack scripts, or malware infections. By combining log-based analysis with system performance monitoring, the system provides a comprehensive view of potential threats. Although the approach is rule-based rather than machine learning-driven, it effectively demonstrates the fundamental principles of real-time anomaly detection used in professional SOC environments.

## 6. SYSTEM IMPLEMENTATION

The system is implemented using Python as the core programming language due to its simplicity, flexibility, and wide range of libraries. Flask is used to develop the web-based dashboard interface, allowing dynamic content rendering and user interaction. The backend continuously processes log data and stores structured information in a database, while the frontend retrieves this data and displays it in an organized format.

The dashboard presents information such as IP addresses, number of login attempts, CPU usage, and memory usage in a tabular format. The system is designed to run in a Linux environment and does not require external cloud infrastructure. Its lightweight nature ensures that it can operate efficiently on standard hardware configurations. The implementation focuses on simplicity and performance, making it suitable for both demonstration and practical use.

## 7. RESULTS AND DISCUSSION

The system was tested using simulated login attempts and controlled suspicious activities to evaluate its performance. The dashboard successfully displayed structured log data and generated alerts when predefined thresholds were exceeded. The

ISSN: 3107-6513

monitoring of CPU and memory usage provided additional insights into system performance and helped in identifying abnormal behavior.

The results indicate that the system effectively performs real-time monitoring, centralized visualization, and rule-based threat detection. It provides a clear representation of how a Security Operations Center operates, even though it is a simplified model. The system's low resource consumption and efficient performance make it suitable for small-scale environments and educational purposes.

In addition to authentication monitoring, system resource monitoring was evaluated under varying workload conditions. Artificial CPU-intensive tasks were executed to simulate abnormal resource usage. The dashboard accurately reflected increases in CPU and memory usage percentages in real time. This demonstrates that the monitoring module effectively captures system performance metrics and integrates them into the security analysis process. Monitoring system resources alongside login attempts provides a broader perspective on potential threats, as unusual resource spikes often accompany malicious scripts or automated attack tools.

## 8. FUTURE ENHANCEMENTS

The system can be further enhanced by integrating advanced technologies and features. Machine learning algorithms can be incorporated to improve anomaly detection and reduce false positives. The integration of geolocation APIs can provide accurate information about the origin of IP addresses, enhancing threat analysis. Automated response mechanisms such as IP blocking can be implemented to prevent further attacks.

Additional improvements may include advanced graphical visualizations using charts and dashboards, cloud integration for scalability.

## 9. CONCLUSION

The Real-Time SOC Dashboard for Threat Monitoring successfully demonstrates the fundamental concepts of continuous security monitoring, centralized log analysis, and real-time alert generation. By utilizing open-source tools and a modular design, the system provides an affordable and efficient alternative to complex enterprise SOC solutions.

The project highlights the importance of proactive monitoring in modern cybersecurity and shows how essential SOC functionalities can be implemented in a simplified manner. It serves as a strong foundation for further research and development in the field of cybersecurity, particularly in threat detection and security analytics. Despite certain limitations—such as reliance on rule-based detection and simulated geolocation data—the system performs reliably within its defined scope. It provides accurate alert detection, stable dashboard performance, and efficient log processing. These outcomes validate the practicality of the proposed solution.

In conclusion, the Real-Time SOC Dashboard for Threat Monitoring successfully fulfills its objectives by providing continuous monitoring, real-time alerting, and structured visualization of security events. The project serves as a strong foundation for further enhancement and demonstrates how cybersecurity monitoring concepts can be implemented using open-source tools and standard.

## 10. REFRENCES

[1] Scarfone, K., and Mell, P., "Guide to Intrusion Detection and Prevention Systems (IDPS)," National Institute of Standards and

ISSN: **3107-6513**

Technology (NIST), Special Publication 800-94, 2007.

[2] Bejtlich, R., The Practice of Network Security Monitoring: Understanding Incident Detection and Response, No Starch Press, 2013.

[3] Stallings, W., Network Security Essentials: Applications and Standards, 6th Edition, Pearson Education, 2017.

[4] Liao, H.J., Lin, C.H.R., Lin, Y.C., and Tung, K.Y., "Intrusion Detection System: A Comprehensive Review," Journal of Network and Computer Applications, Vol. 36, No. 1, 2013, pp. 16–24.

[5] Sommer, R., and Paxson, V., "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," IEEE Symposium on Security and Privacy, 2010.

[6] MongoDB Inc., "MongoDB Manual – Database Administration and Security," Official Documentation, 2023.

[7] Grinberg, M., Flask Web Development: Developing Web Applications with Python, 2nd Edition, O'Reilly Media, 2018.

[8] Ronacher, A., "Flask Framework Documentation," Pallets Projects, 2023.

[9] The Python Software Foundation, "Python 3 Documentation," Available at: https://docs.python.org.

[10] Psutil Documentation, "Process and System Utilities for Python," 2023.

[11] OWASP Foundation, "OWASP Top 10 – Web Application Security Risks," 2021.

[12] Chuvakin, A., Schmidt, K., and Phillips, C., Logging and Log Management: The Authoritative Guide to Understanding the Concepts Surrounding Logging and Log Management, Syngress, 2013.

[13] Kent, K., Chevalier, S., Grance, T., and Dang, H., "Guide to Integrating Forensic Techniques into Incident Response," NIST Special Publication 800-86.

[14] Northcutt, S., and Novak, J., Network Intrusion Detection, 3rd Edition, New Riders Publishing, 2002.

[15] MITRE Corporation, "MITRE ATT&CK Framework – Adversarial Tactics and Techniques," 2023.

## 11. ACKNOWLEDGEMENT

This article is the outcome of the research work carried out in the **Department of Computer Science.** The authors would like to express their sincere gratitude to the Department for providing the necessary support and resources to successfully complete this work. We also extend our thanks to the faculty members and mentors for their valuable guidance and encouragement throughout the research. Their continuous support has greatly contributed to the successful development of this project.