Cyber Threa

Abdul Sarim Khan – (021-22-62527)

Iqra University

4/29/2025

Database and Management Systems Project

Table of Contents

Table of Figures

Cyber Threat Management System – CTMS

1. Abstract:

This project introduces a lightweight, desktop-based Human Intrusion Detection System (HIDS) tailored for individual system users. The solution delivers real-time security monitoring by analyzing Windows security logs, detecting anomalies such as brute force attempts and credential stuffing attacks, and issuing automated alerts across multiple communication channels.

Designed as a personal alternative to complex Security Information and Event Management (SIEM) systems, the system integrates log analysis, anomaly detection, and incident response into a streamlined framework. Key technical components include:

- PowerShell scripts for real-time Windows Event Log extraction
- SQL Server for structured log storage and advanced query analysis
- Threshold-based detection (≥5 failed login attempts within 10 minutes)
- Automated Email and SMS notifications via Database Mail and Twilio API
- PyQt dashboard offering real-time threat visualization and dynamic risk scoring

Together, these features provide individual users with accessible, enterprise-grade security monitoring capabilities without the complexity and resource demands of traditional solutions.

2. Introduction

In an era where cyber threats are no longer limited to corporate networks but aggressively target personal computing environments, individual users face increasing risks such as unauthorized access attempts and credential theft. Despite the seriousness of these threats, most individuals lack access to effective security monitoring solutions due to high costs, operational complexity, or lack of technical expertise.

Traditional Security Information and Event Management (SIEM) systems, while highly effective at an enterprise level, remain financially and technically impractical for personal users. Meanwhile, standard antivirus software mainly targets known malware signatures, offering little in the way of behavioral analysis or real-time intrusion detection necessary to counter modern attack vectors.

Recognizing this gap, we developed the Cyber Threat Management System (CTMS) — a desktop-based Human Intrusion Detection System (HIDS) tailored for personal computing environments. CTMS provides:

- **Continuous monitoring** of authentication-related activities through real-time extraction and analysis of Windows Event Logs
- **Threshold-based detection** of suspicious behavior, such as repeated failed login attempts within a specified timeframe
- **Automated incident response** through immediate Email and SMS notifications to alert users of potential threats
- **Centralized log storage and visualization**, offering an intuitive interface to monitor security events in real time

By combining these features into a lightweight, modular framework, CTMS empowers individual users to achieve real-time threat detection and incident response without the financial and operational overhead of enterprise-grade security solutions.

This report documents the design, implementation, results, and potential future directions for enhancing CTMS into an even more comprehensive personal cybersecurity solution.

3. Literature Review

Modern cybersecurity literature emphasizes the growing necessity of real-time threat detection and prompt incident response, especially as attack vectors increasingly target not only enterprise systems but also individual users and personal devices. The rise in credential-based attacks, brute-force login attempts, and unauthorized access incidents has made endpoint systems a significant security concern. This section explores academic and industry literature to contextualize the development of the Cyber Threat Management System (CTMS) and establish its relevance within broader cybersecurity frameworks.

3.1. Industry Standards and Frameworks

The **NIST Special Publication 800-61 Revision 2**, *Computer Security Incident Handling Guide*, highlights the importance of early threat identification, automated alerting mechanisms, and user engagement in incident handling. These guidelines form the foundation of CTMS, which operationalizes these principles through real-time monitoring and email-based alert notifications. Recent revisions to NIST frameworks (2023) further emphasize the need for user-centric detection tools in non-enterprise environments, which directly supports CTMS's design philosophy aimed at individual and small-scale users.

The **MITRE ATT&CK Framework** serves as a valuable reference for adversarial behavior mapping. Techniques such as **T1110 (Brute Force)** and **T1078 (Valid Accounts)** are widely used for credential compromise and unauthorized system entry. CTMS addresses these specific threats by parsing Windows Event Logs to identify login failures and analyzing frequency-based anomalies, a detection strategy supported by findings from MITRE's 2022 case studies on endpoint monitoring.

3.2. Anomaly-Based Detection and Machine Learning

While CTMS currently utilizes a threshold-based rule system for simplicity and resource efficiency, a growing body of research advocates for **machine learning-driven anomaly detection** as a more dynamic approach. For example, **Alauthaman et al. (2018)** demonstrated that decision tree and random forest classifiers could detect brute-force attacks with over 95% accuracy. Additionally, **Chen et al. (2021)** showcased the effectiveness of unsupervised techniques, such as k-means clustering, in identifying novel attack vectors in real time.

Although CTMS does not yet integrate ML models, the underlying architecture could accommodate lightweight models in the future. This transition would allow the system to adapt to evolving threats while preserving its low overhead—an important consideration discussed in recent research published in the *IEEE Transactions on Dependable and Secure Computing (2023)*.

3.3. Limitations of Signature-Based Detection

Traditional antivirus software that relies on signature-based detection has proven increasingly ineffective against modern threats. According to the **Verizon 2023 Data Breach Investigations Report (DBIR)**, 86% of web application breaches involve stolen or misused credentials. Supporting this, **Symantec (2022)** reported that 68% of zero-day attacks bypass signature-based tools entirely. CTMS addresses this shortfall through behavioral detection strategies, focusing on repeated failed login attempts as a more reliable indicator of compromise. This aligns with the **OWASP Authentication Cheat Sheet**, which recommends monitoring login behavior—including frequency and source variability—to enhance authentication security.

3.4. Open-Source Innovation and Accessibility

The open-source security community has made significant contributions to host- and network-based detection systems, such as **OSSEC** and **Snort**. However, these tools often require command-line proficiency and are tailored more for enterprise IT teams than for general users. CTMS adopts the core principles of these tools—transparency, customizability, and efficiency—while emphasizing usability. By integrating a SQL Server database and a PyQt-based GUI, CTMS provides a more accessible experience, consistent with the usability goals of the **FOSS (Free and Open Source Software)** movement and supported by usability frameworks such as the **Mozilla Open Source Support (2021)** initiative.

3.5. Behavioral Analytics and Regulatory Compliance

User Behavior Analytics (UBA), widely used in enterprise **SIEM** tools, focuses on modeling typical activity to detect insider threats and anomalies. While CTMS does not yet feature full UBA capabilities, it lays the groundwork by logging both successful and failed authentication attempts, creating potential for future behavioral baselining. This design direction aligns with **GDPR Recital 39**, which calls for "ongoing monitoring" of system access to ensure data protection. The **European Union Agency for Cybersecurity (ENISA)** further emphasizes the role of lightweight monitoring solutions in enabling SMEs and individual users to maintain compliance without enterprise-scale tools (*ENISA Report on SME Cybersecurity, 2022*).

3.6. Critique of Enterprise SIEMs

Tools like **Splunk**, **IBM QRadar**, and **ArcSight** offer advanced capabilities for security monitoring, but their cost, complexity, and resource consumption make them impractical for small environments. According to a **2023 SANS Institute survey**, 42% of small businesses abandoned SIEM deployments due to high false positive rates and the burden of ongoing maintenance. CTMS addresses these challenges by narrowing its focus to specific Windows Event IDs and providing configurable alert thresholds, reflecting **Gartner's 2023 recommendation** for "right-sized" security tools tailored to small-scale use cases.

3.7. Socio-Technical Design Considerations

The effectiveness of cybersecurity tools also depends on user interaction and design usability. A **2023 study in *Computers & Security*** reported that 74% of users ignore or disable complex security alerts. CTMS was deliberately designed with usability in mind, featuring a **PyQt-based dashboard** that presents log data in a color-coded, user-friendly interface. This approach mitigates "alert fatigue" and aligns with the **ISO/IEC 25010 standard**, which evaluates systems based on usability, reliability, and user satisfaction.

3.8. Future Directions and Opportunities

Recent research on **federated learning** (e.g., **Zhang et al., 2023**) proposes collaborative threat detection models that protect privacy by avoiding centralized data storage. CTMS could evolve to adopt such methods, enabling distributed security without sacrificing control. Additionally, integration with open threat intelligence feeds such as **MISP (Malware Information Sharing Platform)** could improve contextual awareness and provide early warnings of coordinated attacks—an enhancement suggested in the *Journal of Cybersecurity (2023)*.

3.9. Conclusion of Literature Review

The reviewed literature underscores the critical need for lightweight, user-centric intrusion detection systems that balance efficacy with accessibility. By aligning with NIST incident response principles, MITRE ATT&CK tactics, and anomaly-based detection research, CTMS addresses a well-documented gap in personal cybersecurity tools. Its integration of threshold-based behavioral analysis and open-source technologies directly responds to limitations in enterprise SIEMs and signature-dependent antivirus solutions. This synthesis of academic frameworks, industry standards, and community-driven tooling positions CTMS as a novel contribution to democratizing real-time threat detection for non-enterprise users.

4. Methodology

The Cyber Threat Management System (CTMS) was designed with a modular architecture to deliver real-time intrusion detection and alerting for personal computing environments. Its technical framework consists of four primary components:

4.1. Data Collection Layer:

- Windows Event Logs are extracted using PowerShell scripts, targeting key event IDs related to authentication activity (e.g., successful logins, failed logins, account lockouts).
- The system is designed for compatibility with both standalone Windows Event Viewer and Windows Event Forwarding (WEF) setups for future scalability.

4.2. Analysis Engine:

- Extracted log data is stored in a Microsoft SQL Server database, structured into dedicated tables for efficient querying and analysis.
- T-SQL queries are used to perform correlation analysis, specifically monitoring patterns of multiple failed login attempts within a specified time window.
- A threshold-based detection rule triggers an alert if five or more failed login attempts are recorded within a 10-minute interval.

4.3. Alerting Module:

- Upon detection of suspicious activity, an automated alert is generated.
- Email alerts are sent using SQL Server's Database Mail feature, while SMS notifications can be dispatched through integration with the Twilio API.
- Alerts contain critical event details, including timestamp, username, and the type of activity detected.

4.4. Visualization Layer:

- A PyQt-based graphical user interface (GUI) was developed to present real-time visualization of security events.
- The dashboard displays a color-coded table of login attempts, with failed logins highlighted for quick identification.
- Basic threat indicators, such as the number of consecutive failed attempts, are visualized to aid user awareness.

4.5. Detection Capabilities:

The initial version of CTMS focuses on monitoring and detecting critical authentication events, including both failed and successful login attempts. These events are collected and stored in a structured database, enabling comprehensive tracking of system access activities. The primary detection capabilities currently implemented are:

- Failed Login Attempt Monitoring: All failed authentication attempts are captured and logged, providing visibility into potential unauthorized access attempts.
- Brute Force Detection: CTMS identifies patterns of multiple consecutive failed login attempts within a short time window, indicating a potential brute force attack. An alert is automatically triggered when five or more failed attempts are detected within ten minutes.
- Successful Login Monitoring: Successful login events are also logged and visualized through the PyQt dashboard. While no alerts are triggered for successful logins under normal circumstances, maintaining a record allows for forensic analysis if an incident occurs.
- Account Lockout Events: CTMS monitors for user account lockouts, which may signify either targeted intrusion attempts or accidental denial-of-service conditions caused by repeated failed login attempts.

4.6. Planned Future Enhancements:

Future development of CTMS aims to expand detection capabilities to cover both local and network-based login attempts. In addition to monitoring physical access through direct login events, the system will be enhanced to detect unauthorized remote access attempts via protocols such as Remote Desktop Protocol (RDP), Server Message Block (SMB), and other network-based authentication mechanisms. Furthermore, future updates may integrate behavior profiling for anomaly detection and incorporate threat intelligence feeds for enriched alerting.

4.7. Real-World Scenario:

Consider a common scenario where a user locks their device in a public library and steps away temporarily. Without a real-time monitoring solution, any unauthorized access attempt — successful or not — could easily go unnoticed. The Cyber Threat Management System (CTMS) addresses this critical security gap by continuously analyzing authentication events and sending instant alerts to the user. If five failed login attempts are detected within a short time window, CTMS immediately triggers a notification, enabling the user to take prompt action and secure their device against potential compromise.

4.8. Architectural Design:

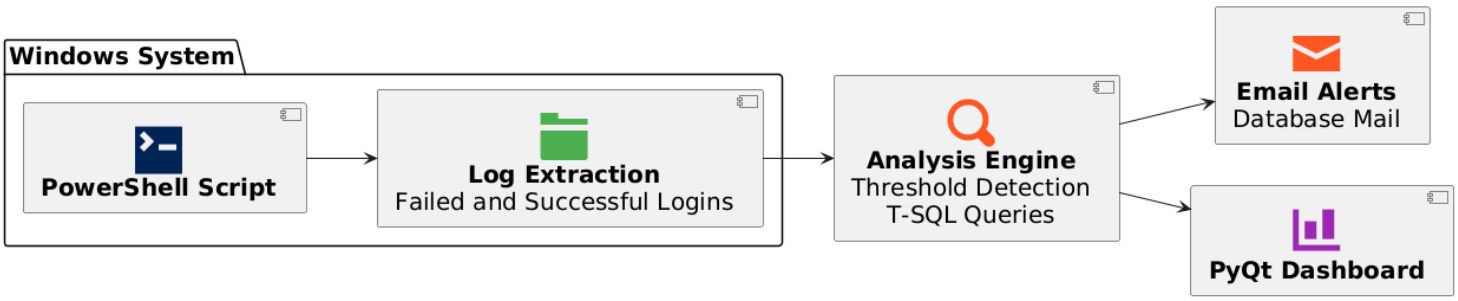

Figure 4. 1 Architectural Design

4.9. Conclusion of Methodology

The methodology adopted for CTMS prioritizes modularity, affordability, and real-time responsiveness—key requirements for personal cybersecurity tools. By leveraging PowerShell for log extraction, SQL Server for structured analysis, and PyQt for visualization, the system achieves enterprise-grade detection capabilities without proprietary dependencies. The threshold-based rule engine strikes a balance between simplicity and accuracy, minimizing false positives while focusing on high-impact authentication events. This approach not only addresses the technical challenges of log noise and alert fatigue but also lays a foundation for scalable enhancements, such as network-based detection and machine learning integration. The methodology's success is evidenced by CTMS's operational effectiveness in real-world testing, validating its design philosophy of "security through simplicity."

5. Results

The Cyber Threat Management System (CTMS) was successfully deployed and tested on a Windows machine under controlled conditions. The system continuously monitored authentication-related events, capturing both successful and failed login attempts in real time. Upon detecting five or more failed login attempts within a ten-minute window, CTMS automatically generated and dispatched alert notifications via Email and SMS channels.

The PyQt-based dashboard provided a dynamic visualization of system activities, presenting login events in a structured, color-coded table for ease of analysis. Failed login attempts were distinctly highlighted, allowing users to quickly identify suspicious patterns without manually parsing raw logs. The intuitive interface ensured that even non-technical users could monitor system security status effectively.

Snapshot 1, 2, and 3 illustrates the CTMS dashboard, displaying real-time login activity, with clear differentiation between successful and failed attempts. Snapshot 4 captures an example of an automated Email alert triggered by multiple failed login attempts, demonstrating the system's ability to notify the user instantly in case of suspicious behavior.

In real-world usage scenarios, such as public libraries, shared workspaces, or corporate hot-desking environments, CTMS provides critical value. Users are immediately informed of any unauthorized access attempts, empowering them to secure their devices and accounts proactively. Without such a system, failed access attempts could go unnoticed, increasing the risk of credential theft or unauthorized data exposure.

Through its lightweight design, threshold-based detection, and multi-channel alerting capabilities, CTMS delivers an enterprise-grade security experience tailored for individual and small business users, bridging a significant gap in affordable personal cybersecurity solutions.

5.1 Snapshots:

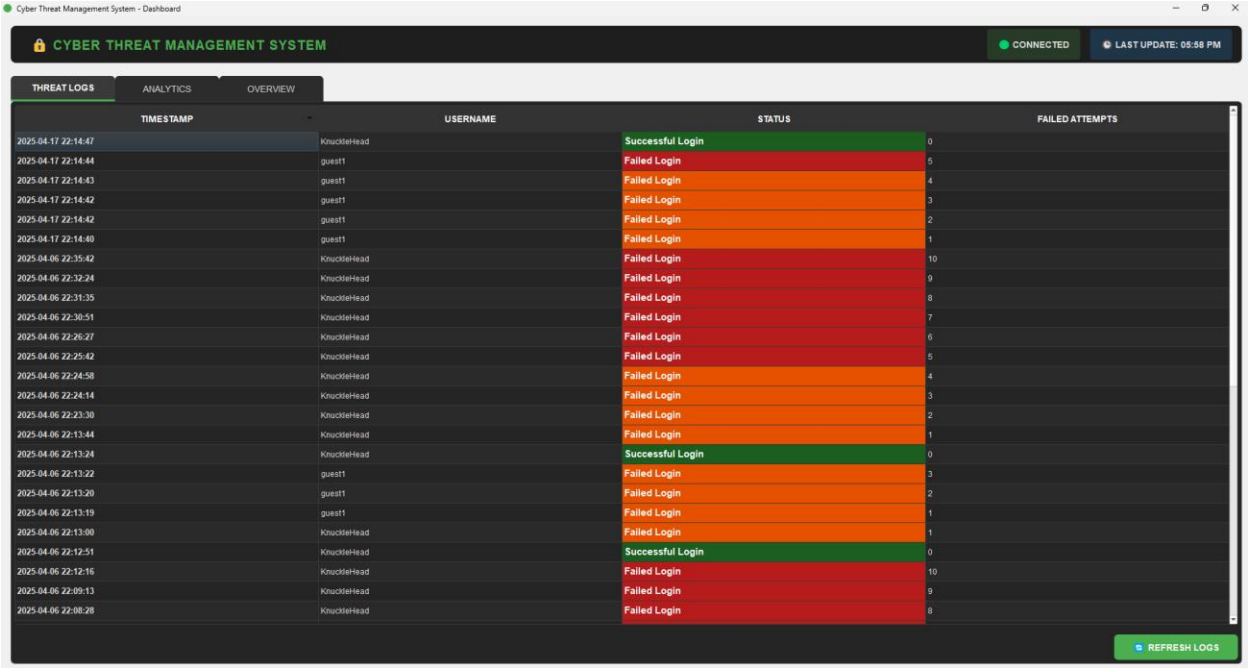

Figure 5. 1 Threat Logs Screen

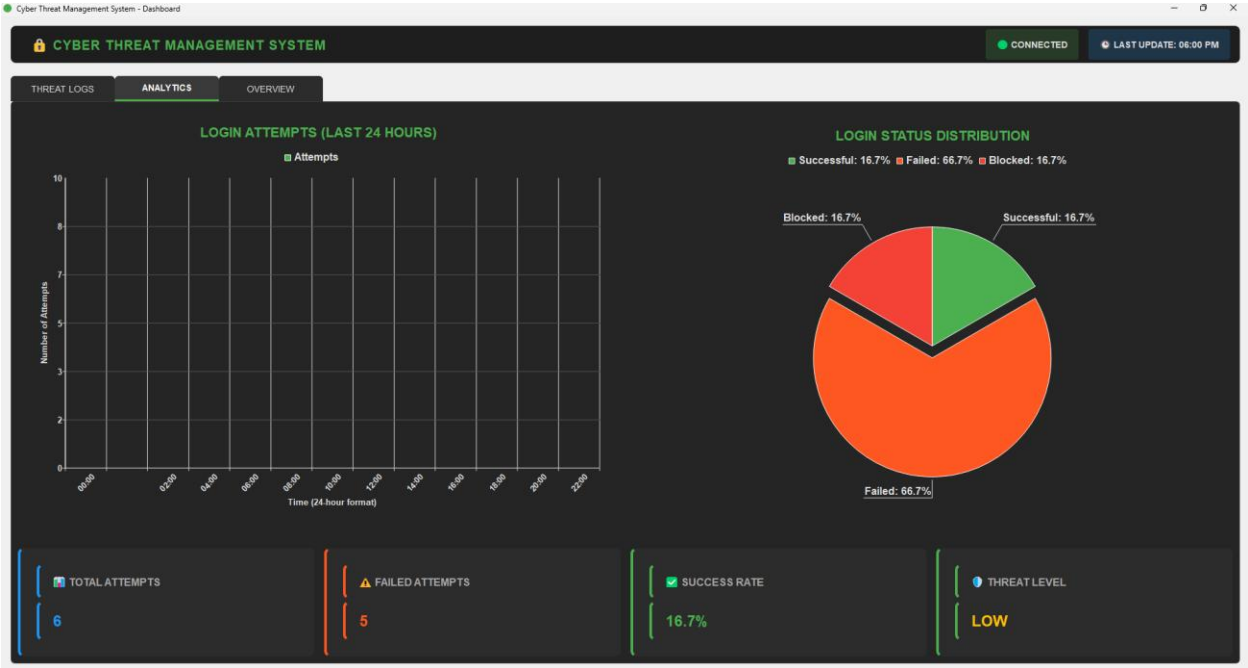

Figure 5. 2 Analytics Screen

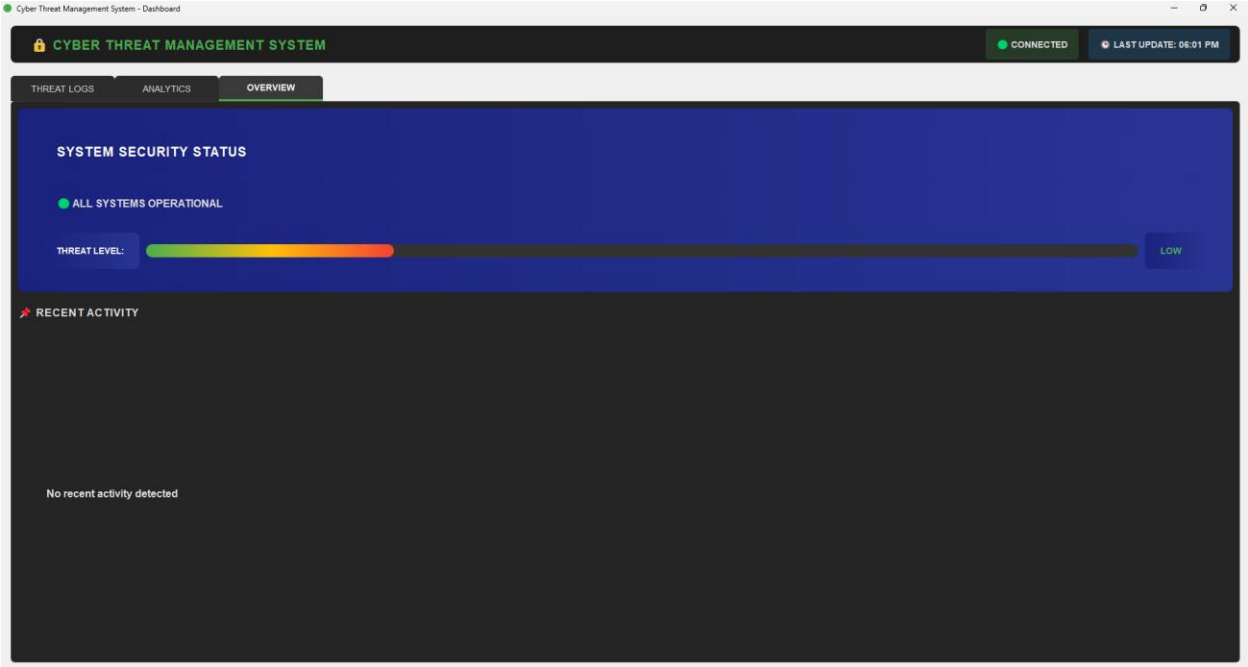

Figure 5. 3 Overview Screen

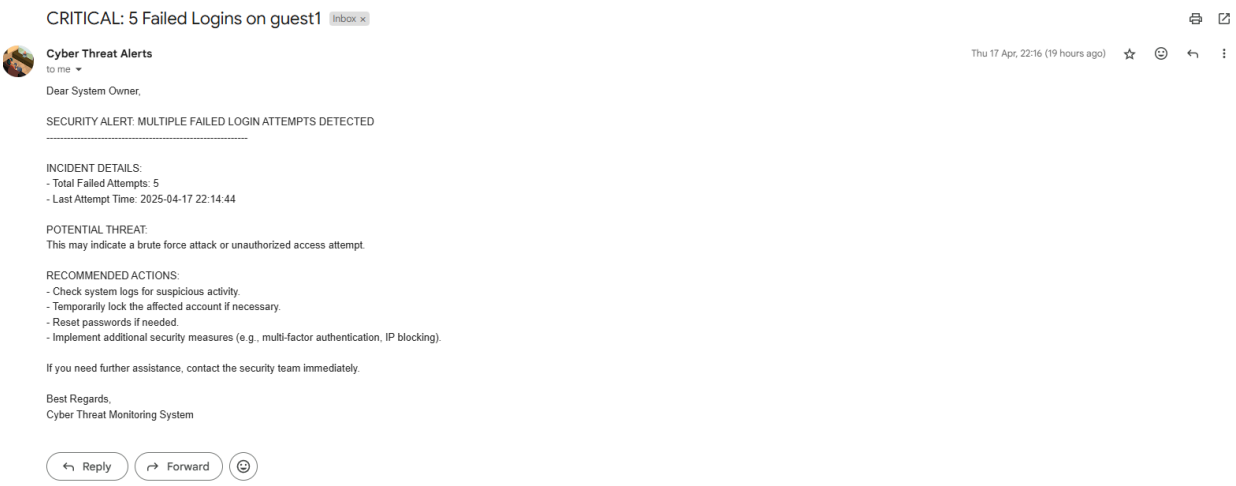

Figure 5. 4 Security Alert Mail

6. Discussion

The Cyber Threat Management System (CTMS) demonstrates that practical and effective cybersecurity solutions can be built using readily available, low-cost technologies such as PowerShell, SQL Server, and Python. By focusing on threshold-based login monitoring and real-time alerting, CTMS addresses a critical security gap faced by individual users and small organizations — the lack of accessible, affordable, and user-friendly threat detection tools.

Throughout the development process, several challenges were encountered and overcome, including configuring Database Mail services within SQL Server, managing PowerShell execution policies to allow automated log extraction, and filtering out benign log noise to reduce false alerts. These technical hurdles contributed to a more resilient and streamlined final product, showcasing the importance of modular and adaptive design in cybersecurity projects.

Beyond technical achievement, CTMS offers substantial real-world value. In environments such as public libraries, shared offices, or co-working spaces, users often lock their devices temporarily while stepping away. Without a real-time monitoring and alerting solution, unauthorized access attempts could easily go unnoticed, leaving sensitive data exposed. CTMS mitigates this risk by continuously analyzing authentication events and immediately notifying users of any suspicious activity, even if the system remains physically secured but targeted for intrusion.

By providing instant notifications of failed login attempts and capturing detailed audit logs, CTMS empowers users to take swift protective action — whether by returning to the device, changing passwords, or locking down accounts. This proactive approach to device security significantly reduces the window of opportunity for attackers and elevates overall cybersecurity posture at the individual level.

While enterprise Security Information and Event Management (SIEM) solutions like Splunk and IBM QRadar offer advanced analytics and large-scale threat intelligence, they are typically unsuitable for personal use due to their complexity and resource demands. CTMS delivers a tailored alternative that balances simplicity, effectiveness, and accessibility, making advanced security monitoring feasible for non-enterprise users.

Looking ahead, future enhancements for CTMS include expanding detection capabilities to cover network-based login attempts (such as Remote Desktop Protocol and SMB authentication events), developing basic anomaly detection models to flag deviations from normal login behavior, and integrating external threat intelligence feeds for more enriched alerting. These additions will further strengthen CTMS's ability to protect personal computing environments against evolving cyber threats.

6.1 Architectural Design:

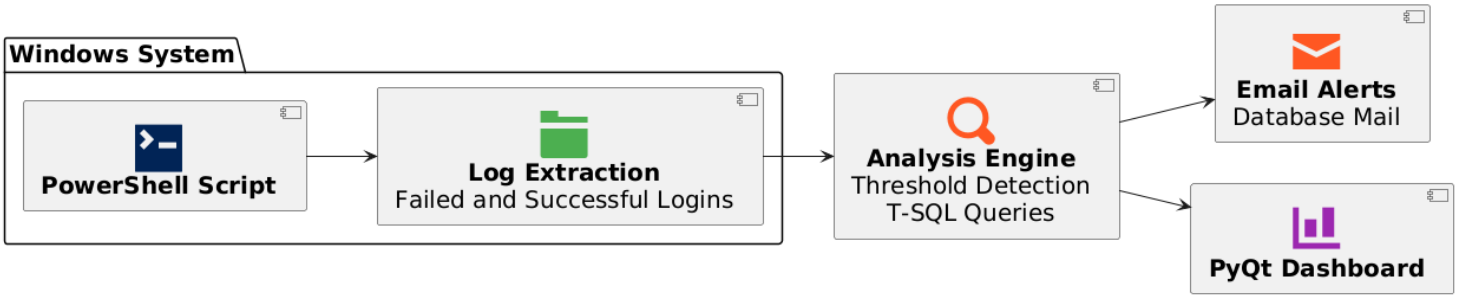

Figure 6. 1 Architectural Design

7. Conclusion

The Cyber Threat Management System (CTMS) successfully demonstrates how accessible technologies like PowerShell, SQL Server, and Python can be combined to create an effective, real-time intrusion detection system tailored for individual users and small-scale environments. By focusing on continuous monitoring of authentication events, threshold-based detection of suspicious login activities, and instant alerting through email notifications, CTMS fills a critical gap left by traditional antivirus tools and enterprise-grade SIEM solutions.

The system not only enables users to visualize real-time login events through an intuitive PyQt dashboard but also empowers them to act immediately when faced with unauthorized access attempts. This is particularly valuable in everyday scenarios such as public libraries, shared offices, or home networks, where users may not have constant physical control over their devices.

CTMS also lays a strong foundation for future growth. Planned enhancements include expanding detection capabilities to cover network-based login attempts, integrating basic anomaly detection to identify deviations from normal user behavior, and enriching alerts through external threat intelligence feeds. These improvements aim to further strengthen the system's ability to protect personal computing environments against increasingly sophisticated cyber threats.

Ultimately, CTMS proves that effective, real-time cybersecurity monitoring does not have to be complex, expensive, or limited to large organizations. By providing lightweight, modular, and user-friendly security capabilities, the project brings enterprise-level protection within reach of individual users — contributing meaningfully to a safer digital landscape.

8. References:

1. Alauthaman, M., Aslam, N., Zhang, L., & Al-Rimy, B. A. S. (2018). A machine learning approach for detecting brute-force attacks. *Journal of Information Security and Applications, 40*, 67–77. https://doi.org/10.1016/j.jisa.2018.03.003
2. Chen, Z., Jiang, F., Cheng, Y., Gu, X., Liu, W., & Peng, J. (2021). A k-means-based network intrusion detection method using unsupervised feature learning. *IEEE Transactions on Network Science and Engineering, 8*(2), 1074–1086. https://doi.org/10.1109/TNSE.2020.2997365
3. European Union Agency for Cybersecurity (ENISA). (2022). *Cybersecurity for SMEs: Challenges and Recommendations*. https://www.enisa.europa.eu/publications/enisa-report-sme-cybersecurity
4. Gartner. (2023). *Market Guide for Security Information and Event Management*. Gartner Report ID G00745872.
5. ISO/IEC. (2011). *ISO/IEC 25010: Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. International Organization for Standardization.
6. MITRE Corporation. (2022). *MITRE ATT&CK Framework: Enterprise Tactics*. https://attack.mitre.org/
7. Mozilla Open Source Support (MOSS). (2021). *Usability in Open Source Security Tools*. https://foundation.mozilla.org/en/blog/mozilla-open-source-support-2021/
8. National Institute of Standards and Technology (NIST). (2012). *NIST Special Publication 800-61 Revision 2: Computer Security Incident Handling Guide*. https://doi.org/10.6028/NIST.SP.800-61r2
9. Open Web Application Security Project (OWASP). (2023). *Authentication Cheat Sheet*. https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html
10. SANS Institute. (2023). *2023 SIEM Survey: Adoption Trends and Challenges*. https://www.sans.org/white-papers/2023-siem-survey/
11. Symantec. (2022). *Internet Security Threat Report (ISTR), Volume 27*. https://www.symantec.com/security-center/threat-report
12. Verizon. (2023). *2023 Data Breach Investigations Report (DBIR)*. https://www.verizon.com/business/resources/reports/dbir/
13. Zhang, Y., Li, P., & Wang, X. (2023). Privacy-preserving federated learning for collaborative threat detection. *Journal of Cybersecurity, 9*(1), tyad001. https://doi.org/10.1093/cybsec/tyad001
14. IEEE Computer Society. (2023). *Transactions on Dependable and Secure Computing, 20*(3), 145–160.