Cyber Threat Intelligence Dashboard

Gracy Avula[1], Manasa Palla[2], Navya Gonu[3], Sarika Muddagowni[4], and Sandhya Rani[5]
[1,2,3,4,5]*Member, Department of Cybersecurity, Sphoorthy Engineering College, Hyderabad, Telangana, India.*

***Abstract*—Cybersecurity threats are increasing at an alarming rate, with attackers continuously exploiting vulnerabilities through malicious IP addresses, domains, and malware campaigns. Manual monitoring of such threats is inefficient, error-prone, and often delayed, reducing the ability to prevent incidents effectively. To address this challenge, this paper proposes a Cyber Threat Intelligence (CTI) Dashboard that collects and analyzes real-time threat data from multiple open-source intelligence platforms, including VirusTotal, AlienVault OTX, and AbuseIPDB, and presents actionable insights through a centralized interface. The system is developed using lightweight frameworks such as Python Flask and integrates APIs to automatically fetch and process data on suspicious entities. It organizes the data to highlight threat categories, severity levels, frequency, and geolocation patterns. The interactive dashboard visualizes this information through charts, tables, and alert mechanisms, enabling security analysts to identify potential threats proactively. The proposed system supports threat hunting and Security Operations Center (SOC) activities by enhancing early detection and improving situational awareness. By automating data collection, correlation, and visualization, the system reduces the time and effort required for threat intelligence analysis. Future enhancements include integration with Security Information and Event Management (SIEM) systems, incorporation of predictive analytics, and scalability for enterprise-level deployment.**

***Index Terms*—Cyber Threat Intelligence, Dashboard, Data Visualization, Flask, Open-Source Intelligence, Real-Time Monitoring, Threat Analysis.**

I. INTRODUCTION

In recent years, the rapid growth of digital technologies and internet connectivity has led to a significant increase in cybersecurity threats. Organizations and individuals are constantly exposed to attacks such as malware infections, phishing campaigns, distributed denial-of-service (DDoS) attacks, and unauthorized access attempts. These threats are often carried out using malicious IP addresses, domains, and other indicators of compromise (IOCs), making it essential to monitor and analyze such data effectively.

Cyber Threat Intelligence (CTI) plays a crucial role in identifying, analyzing, and mitigating these threats by providing actionable insights based on collected data. However, traditional methods of threat monitoring involve manual data collection and analysis from multiple sources, which is time-consuming, inefficient, and prone to human error. Security analysts often struggle to correlate information from various platforms, leading to delayed responses and increased risk of security breaches.

To overcome these challenges, there is a need for an automated and centralized system that can gather, process, and visualize threat intelligence data in real time. A Cyber Threat Intelligence Dashboard provides an effective solution by integrating multiple open-source intelligence platforms into a single interface. Such a system enables continuous monitoring, efficient data correlation, and intuitive visualization of threat patterns.

This paper presents the design and implementation of a Cyber Threat Intelligence (CTI) Dashboard developed using Python Flask as the backend framework. The system integrates APIs from multiple sources, including Virus Total, Shodan, and AbuseIPDB, to collect real-time data related to suspicious IPs, domains, and other threat indicators. The collected data is processed and stored in a structured format, allowing the system to categorize threats based on severity, type, and frequency.

The dashboard provides interactive visualizations such as charts, tables, and alert notifications, which assist security analysts in identifying potential threats quickly and accurately. By automating the processes of data collection, analysis, and visualization, the

proposed system reduces manual effort and enhances the overall efficiency of threat intelligence operations. The remainder of this paper is organized as follows: Section II discusses related work and existing systems, Section III describes the proposed system and architecture, Section IV explains the implementation details, Section V presents the results and discussion, and Section VI concludes the paper with future enhancements.

II. LITERATURE SURVEY

Cyber Threat Intelligence (CTI) plays a vital role in modern cybersecurity by enabling organizations to proactively identify, analyze, and mitigate potential threats. Various platforms have been developed to provide threat intelligence; however, these systems often operate independently and present challenges in data correlation and centralized analysis.

AlienVault OTX is a widely recognized open-source threat intelligence platform that provides community-driven data on indicators of compromise (IOCs), including malicious IPs, domains, and file hashes. It enables threat sharing and supports threat hunting activities through its rich dataset. In the proposed system, AlienVault OTX serves as the primary data source for real-time threat intelligence. However, the platform itself lacks advanced visualization capabilities and requires additional tools for effective analysis and monitoring.

Virus Total aggregates data from multiple antivirus engines and provides detailed analysis of files, URLs, and IP addresses. It is highly effective for malware detection and verification of suspicious entities. Despite its strengths, Virus Total is primarily designed for individual lookups and does not offer continuous monitoring or integrated dashboard functionalities for large-scale analysis.

Shodan provides insights into exposed devices and services on the internet, allowing users to identify potential vulnerabilities in network infrastructure. While it is powerful for reconnaissance and asset discovery, it lacks integration with other intelligence sources and does not provide unified visualization for comprehensive threat analysis.

AbuseIPDB focuses on identifying malicious IP addresses based on user-reported data. It is useful for detecting and blocking suspicious network activity. However, its scope is limited to IP-based threats and does not cover broader indicators such as domains, malware, or behavioral patterns.

A critical analysis of these platforms reveals that, although they provide valuable threat intelligence, they function in isolation and lack centralized data aggregation, correlation, and visualization capabilities. This fragmentation makes it difficult for security analysts to obtain a holistic view of the threat landscape and respond effectively in real time.

To address these limitations, the proposed Cyber Threat Intelligence Dashboard integrates multiple intelligence sources into a unified platform. The current implementation utilizes data from AlienVault OTX, while other platforms such as VirusTotal, Shodan, and AbuseIPDB are considered for future integration. The system focuses on automating data collection, enabling correlation across different sources, and presenting the information through an interactive and user-friendly dashboard, thereby enhancing threat visibility and decision-making.

III. PROPOSED SYSTEM AND ARCHITECTURE

The proposed Cyber Threat Intelligence (CTI) Dashboard is designed to provide a centralized platform for collecting, processing, and visualizing threat intelligence data obtained from open-source platforms. The system follows a layered architecture that ensures efficient data flow, scalability, and real-time threat analysis.

The overall architecture of the system is illustrated in Fig. 1, which represents the sequential flow of data from threat intelligence sources to the end-user dashboard.

A. Threat Intelligence Sources

The system collects threat data from open-source intelligence platforms such as AlienVault OTX. These sources provide information on indicators of compromise (IOCs), including malicious IP addresses, domains, URLs, and file hashes. The architecture is designed to support integration with additional platforms such as VirusTotal, Shodan, and AbuseIPDB in the future, thereby enhancing the diversity and reliability of threat intelligence data.

B. Data Ingestion Layer

The Data Ingestion Layer is responsible for fetching real-time threat data using APIs. It periodically sends requests to the configured intelligence sources and retrieves structured or semi-structured data. This layer ensures continuous data collection and acts as the entry point of the system. It also handles API responses, rate limiting, and error management to maintain reliable data acquisition.

C. Data Processing: Normalization and Enrichment

Once the data is collected, it is passed to the processing layer, where normalization and enrichment operations are performed. Normalization involves converting data from different sources into a consistent format, enabling uniform analysis. Enrichment includes categorizing threats, assigning severity levels, and identifying patterns based on attributes such as frequency and type of attack.

This layer plays a crucial role in transforming raw data into meaningful intelligence by filtering irrelevant entries and highlighting critical threats.

D. Database (Threat Store)

The processed data is stored in a centralized database that acts as the threat repository. The database maintains structured records of all collected threat information, including timestamps, threat categories, severity levels, and source details. This storage mechanism enables efficient querying, historical analysis, and trend identification.

E. Backend APIs

The backend of the system is implemented using Python Flask, which provides RESTful APIs for communication between the database and the frontend interface. These APIs handle data retrieval, filtering, and aggregation requests from the dashboard. The backend ensures secure and efficient data access while supporting scalability for handling larger datasets.

F. Frontend Dashboard

The frontend dashboard provides an interactive user interface for visualizing threat intelligence data. It displays information through charts, tables, and alert mechanisms, enabling users to monitor threat activities effectively. The dashboard allows users to analyze trends, identify high-severity threats, and gain insights into the overall threat landscape.

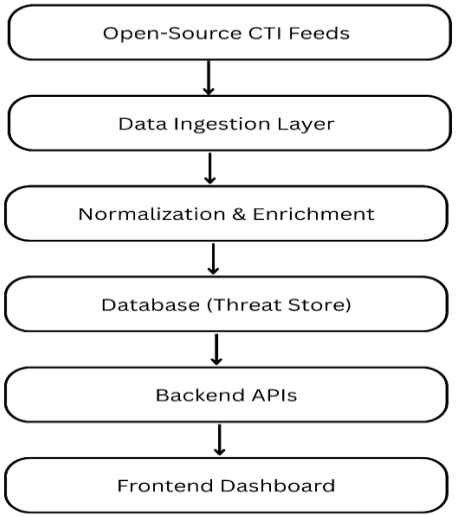

Fig. 1. Architecture of the Proposed Cyber Threat Intelligence Dashboard

IV. IMPLEMENTATION

The implementation of the Cyber Threat Intelligence (CTI) Dashboard focuses on developing a lightweight, scalable, and efficient system capable of collecting, processing, and visualizing threat intelligence data in real time. The system is built using a combination of backend, database, and frontend technologies to ensure seamless integration and performance.

A. Technology Stack

The proposed system is developed using the following technologies:

- Backend: Python Flask framework is used to develop RESTful APIs and handle server-side logic.
- Database: MySQL is used to store and manage threat intelligence data in a structured format.
- Frontend: HTML, CSS, and JavaScript are used to design the interactive dashboard interface.
- APIs: Threat intelligence data is fetched using APIs from AlienVault OTX.

This combination of technologies ensures a lightweight yet powerful system capable of handling real-time data processing and visualization.

B. Data Collection

The system collects threat intelligence data from AlienVault OTX using its API. The backend

periodically sends requests to retrieve information about indicators of compromise (IOCs), such as malicious IP addresses, domains, and URLs.
The collected data is received in JSON format and passed to the processing module for further analysis.

C. Data Processing and Storage
Once the data is fetched, it undergoes preprocessing steps, including:

- Data cleaning to remove redundant or irrelevant entries
- Normalization to maintain a consistent data structure
- Categorization based on threat type and severity

After processing, the data is stored in a MySQL database, which acts as a centralized repository. The database schema is designed to efficiently store attributes such as threat type, severity level, timestamp, and source information.

D. Backend Development
The backend is implemented using the Flask framework, which provides APIs for:

- Fetching threat data from the database
- Filtering data based on severity or type
- Sending processed data to the frontend

The backend ensures smooth communication between the database and the user interface while maintaining system performance.

E. Frontend Development
The frontend dashboard is designed to provide a user-friendly interface for visualizing threat intelligence data. It includes:

- Tables for displaying detailed threat information
- Charts for visual representation of threat distribution
- Alerts for highlighting high-severity threats

The interface enables users to easily interpret complex data and identify patterns in cyber threats.

F. System Integration
All components of the system are integrated to form a complete pipeline. The backend APIs connect the database and frontend, enabling real-time data updates and seamless interaction between different modules.

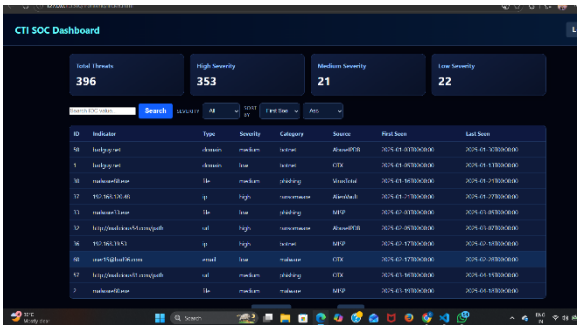

Fig. 2. Dashboard Interface

Fig. 3. Threat Data Visualization

V. RESULTS AND DISCUSSION

The proposed Cyber Threat Intelligence (CTI) Dashboard was successfully developed and tested to evaluate its effectiveness in collecting, processing, and visualizing threat intelligence data. The system demonstrates the ability to provide meaningful insights from real-time threat data obtained through AlienVault OTX.

A. Results
The implementation of the system produced the following key outcomes:

The dashboard successfully retrieves threat intelligence data using API integration and displays it in a structured format.

- Threat data, including malicious IPs and domains, is categorized based on severity and type.
- The system provides visual representations such as charts and tables, making it easier to analyze large volumes of data.
- High-severity threats are highlighted, enabling quicker identification and response.
- The centralized dashboard improves visibility by consolidating threat data into a single interface.

The results indicate that the system effectively reduces the complexity involved in analyzing raw threat intelligence data.

B. Discussion

The developed CTI Dashboard addresses the limitations of existing standalone platforms by providing a unified and automated solution for threat monitoring. Unlike individual tools, the system enables correlation of threat data and presents it in an easily interpretable format.

The use of visualization techniques enhances situational awareness and assists security analysts in identifying patterns and trends. Additionally, the automation of data collection and processing reduces manual effort and minimizes the chances of human error.

However, the current implementation is limited to a single data source, i.e., AlienVault OTX. While this provides valuable insights, integrating additional platforms such as VirusTotal, Shodan, and AbuseIPDB would further enhance the system's capability by providing more diverse and comprehensive threat intelligence.

Overall, the system demonstrates a practical and scalable approach to cyber threat intelligence management. The results validate that the proposed dashboard can significantly improve the efficiency of threat analysis and support proactive cybersecurity measures.

VI. CONCLUSION AND FUTURE WORK

This paper presented the design and implementation of a Cyber Threat Intelligence (CTI) Dashboard aimed at improving the efficiency of threat monitoring and analysis. The system provides a centralized platform for collecting, processing, and visualizing threat intelligence data obtained from open-source platforms such as AlienVault OTX. By integrating data collection, normalization, storage, and visualization into a single system, the proposed solution simplifies the process of identifying and analysing potential cyber threats.

The implementation demonstrates that the dashboard is capable of transforming raw threat data into meaningful insights through the use of interactive charts, tables, and alert mechanisms. This enhances situational awareness and enables faster decision-making for security analysts. Additionally, the automation of data collection and processing reduces manual effort and improves the overall efficiency of threat intelligence operations.

Despite its effectiveness, the current system has certain limitations, including dependency on a single data source and limited real-time correlation capabilities. These limitations provide opportunities for further enhancement and expansion of the system.

Future work will focus on integrating additional threat intelligence platforms such as VirusTotal, Shodan, and AbuseIPDB to improve data diversity and accuracy. The system can also be enhanced by incorporating advanced features such as predictive analytics, machine learning-based threat detection, and real-time alerting mechanisms. Furthermore, integration with Security Information and Event Management (SIEM) systems can make the dashboard more scalable and suitable for enterprise-level deployment.

In conclusion, the proposed CTI Dashboard provides a practical and extensible solution for modern cybersecurity challenges and has the potential to evolve into a comprehensive threat intelligence platform.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to their project guide and faculty members for their valuable guidance, support, and encouragement throughout the development of this project. Their insights and suggestions greatly contributed to the successful completion of this work.

The authors also thank Sphoorthy Engineering College, Nadergul, Hyderabad, Telangana for providing the necessary resources and environment to carry out this project effectively. Special

appreciation is extended to the developers and contributors of open-source platforms such as AlienVault OTX, which played a crucial role in providing threat intelligence data for this research.
Finally, the authors would like to acknowledge their peers and team members for their cooperation, collaboration, and continuous support during the project development process.

REFERENCES

[1] AlienVault OTX, "Open Threat Exchange," [Online]. Available: https://otx.alienvault.com. Accessed: Mar. 31, 2026.
[2] VirusTotal, "VirusTotal," [Online]. Available: https://www.virustotal.com. Accessed: Mar. 31, 2026.
[3] Shodan, "Shodan," [Online]. Available: https://www.shodan.io. Accessed: Mar. 31, 2026.
[4] AbuseIPDB, "AbuseIPDB," [Online]. Available: https://www.abuseipdb.com. Accessed: Mar. 31, 2026.
[5] Flask Documentation, "Flask web framework," [Online]. Available: https://flask.palletsprojects.com. Accessed: Mar. 31, 2026.
[6] MySQL Documentation, "MySQL database," [Online]. Available: https://www.mysql.com. Accessed: Mar. 31, 2026.
[7] M. Bishop, *Computer Security: Art and Science*. Boston, MA, USA: Addison-Wesley, 2003.
[8] W. Stallings, *Network Security Essentials*, 6th ed. Boston, MA, USA: Pearson, 2017.
[9] S. Barnum, "Standardizing cyber threat intelligence information with the structured threat information expression (STIX)," MITRE Corporation, 2012.