Interactive Threat Intelligence Dashboard

BY- SREYA SHYJASH

BTECH CSE CYBER SECURITY

3RD YEAR

INDEX

AIM

The primary aim of this project is to design and develop a standalone data analytics platform for visualizing and exploring cybersecurity threat intelligence generated by an external analysis engine.

To achieve this aim, the following specific objectives were established:

1. **To develop a robust data ingestion pipeline (ingest.py)** capable of processing a large dataset of messages by making API calls to the external "Vector-Threat" microservice.
2. **To establish a data persistence layer** using a structured SQLite database to store the structured JSON analysis results received from the API.
3. **To build an interactive dashboard using Streamlit** as the primary user interface for data exploration.
4. **To implement multiple data visualization components**, including time-series charts for trend analysis, bar charts for categorical data (e.g., top threat types), and key performance indicator (KPI) metrics.
5. **To integrate interactive filtering capabilities**, allowing users to dynamically filter the entire dashboard by date range and threat score, thereby facilitating deep-dive analysis.
6. **To leverage the Pandas library** for all data manipulation, aggregation, and preparation tasks prior to visualization.

INTRODUCTION

In the field of cybersecurity, the detection of individual threats is only the first step in a mature defensive strategy. Security Operations Centers (SOCs) are often inundated with thousands of individual alerts daily, making it nearly impossible to discern larger patterns or coordinated attack campaigns through case-by-case analysis. This "alert fatigue" means that security teams are often data-rich but insight-poor, stuck in a reactive posture rather than being able to proactively identify and counter emerging threats.

This project, the **"Interactive Threat Intelligence Dashboard,"** was developed to address this critical gap. It serves as a strategic analysis layer built upon the tactical capabilities of the "Vector-Threat" detection engine. By systematically ingesting, storing, and visualizing the data from individual threat analyses, this platform transforms a stream of raw data into a rich, explorable intelligence dashboard. It empowers a security analyst to move beyond single alerts and instead discover the trends, tactics, and timelines of malicious campaigns, enabling a more proactive and data-driven security posture.

METHOLODOGY

The project was executed using a decoupled, two-part architecture that simulates a real-world microservices environment. This approach clearly separates the data analysis service from the data visualization platform.

A. System Architecture:

- **Analysis Microservice:** The pre-existing "Vector-Threat" Flask application acts as a dedicated microservice. Its sole responsibility is to receive a message via an API call and return a JSON object with the analysis results.
- **Data Pipeline (ingest.py):** A standalone Python script serves as the ETL (Extract, Transform, Load) component. It extracts data from a sample list, sends it to the Flask API for transformation (analysis), and loads the structured result into the SQLite database. This process is designed to be run manually to populate the database.
- **Visualization Platform (dashboard.py):** The main Streamlit application functions as a pure data visualization tool. It is entirely decoupled from the analysis engine and operates solely on the data present in the SQLite database.

B. Data Flow and Processing:

1. **Ingestion:** The ingest.py script iterates through a list of sample messages. For each message, it makes an HTTP POST request to the running Flask API.
2. **Storage:** Upon receiving a successful 200 OK response from the API, the script parses the JSON and inserts the relevant fields (message body, threat score, findings) into the analysis_results table in the threat_data.db file.
3. **Loading and Visualization:** The dashboard.py application, when launched, connects to the threat_data.db file. It uses the Pandas read_sql_query function to load the entire dataset into a DataFrame. This data is cached using Streamlit's @st.cache_data decorator for performance. All charts and KPIs are generated from this in-memory DataFrame, which is filtered in real-time based on user interactions with the sidebar widgets.

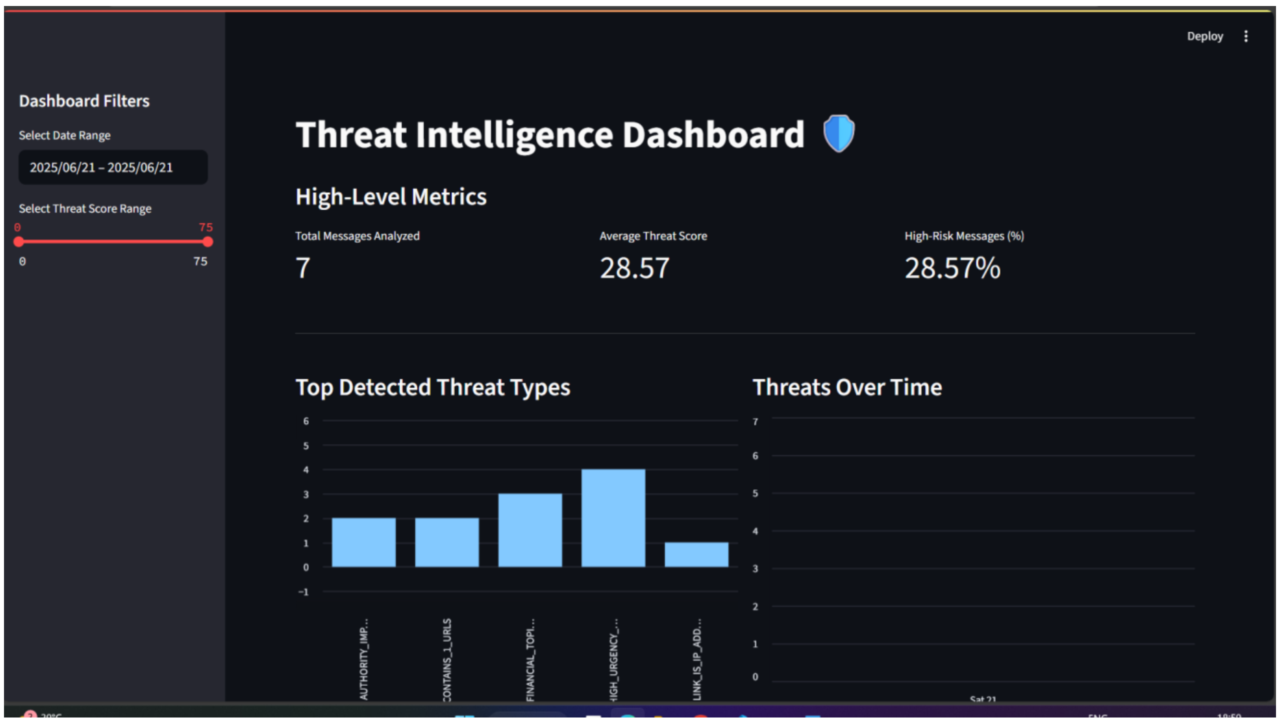
CODE

The codes are provided in the GitHub.

RESULT AND TESTING

The project successfully resulted in a fully functional, two-part data analytics platform that met all defined objectives.

- The **data ingestion pipeline** (ingest.py) was able to successfully communicate with the separately running "Vector-Threat" Flask API, process all sample messages, and populate the SQLite database without errors.
- The **"Threat-Intel Dashboard"** (dashboard.py) successfully loaded the data from the database and rendered a comprehensive, interactive user interface.
- All visualization components were implemented correctly: the **KPI cards** displayed accurate summary statistics, the **bar chart** correctly identified the most frequent threat types, and the **line chart** visualized threat volume over time.
- The **interactive filters** in the sidebar for date range and threat score worked as expected, allowing users to dynamically slice the dataset and have all on-screen visualizations update instantly in response. The final platform provided a seamless and powerful user experience for data exploration.


Dashboard Filters

Select Date Range: 2025/06/21 – 2025/06/21

Select Threat Score Range: 0 – 75

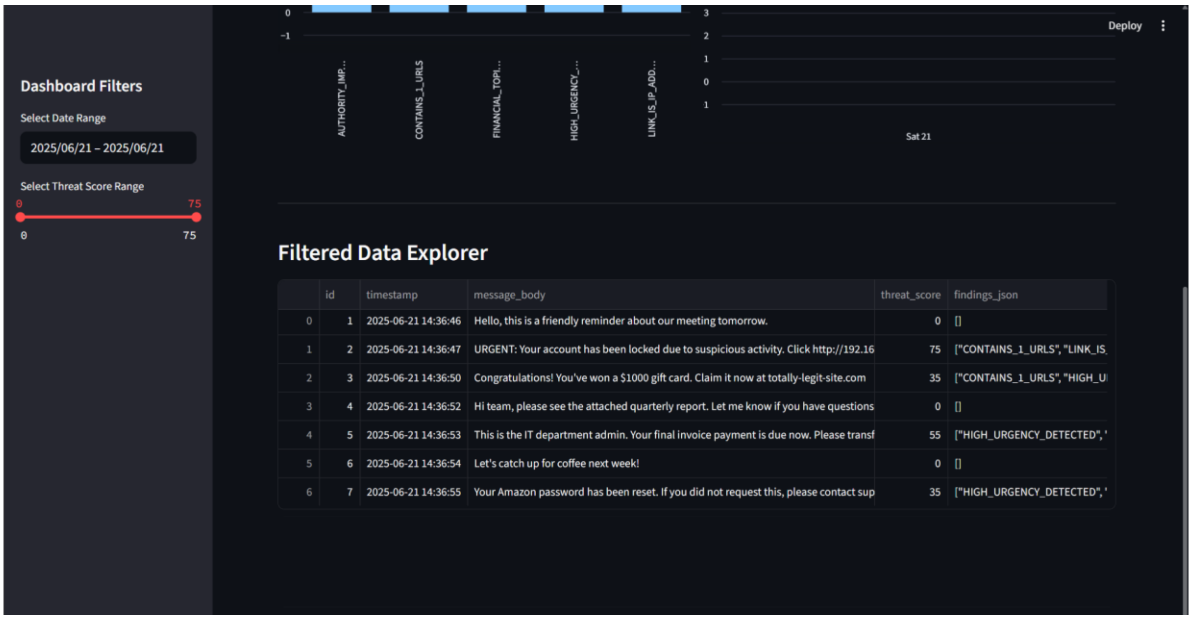
Threat Intelligence Dashboard

High-Level Metrics

Total Messages Analyzed: 7

Average Threat Score: 28.57

High-Risk Messages (%): 28.57%



Dashboard Filters

Select Date Range

2025/06/21 – 2025/06/21

Select Threat Score Range

0 75

0 75

Filtered Data Explorer

	id	timestamp	message_body	threat_score	findings_json
0	1	2025-06-21 14:36:46	Hello, this is a friendly reminder about our meeting tomorrow.	0	[]
1	2	2025-06-21 14:36:47	URGENT: Your account has been locked due to suspicious activity. Click http://192.16	75	["CONTAINS_1_URLS", "LINK_IS_
2	3	2025-06-21 14:36:50	Congratulations! You've won a $1000 gift card. Claim it now at totally-legit-site.com	35	["CONTAINS_1_URLS", "HIGH_U
3	4	2025-06-21 14:36:52	Hi team, please see the attached quarterly report. Let me know if you have questions	0	[]
4	5	2025-06-21 14:36:53	This is the IT department admin. Your final invoice payment is due now. Please transf	55	["HIGH_URGENCY_DETECTED", '
5	6	2025-06-21 14:36:54	Let's catch up for coffee next week!	0	[]
6	7	2025-06-21 14:36:55	Your Amazon password has been reset. If you did not request this, please contact sup	35	["HIGH_URGENCY_DETECTED", '

CONCLUSION

The "Interactive Threat Intelligence Dashboard" project successfully demonstrates a complete, end-to-end data pipeline within a realistic cybersecurity context. By decoupling the data analysis engine from the visualization platform, the project mirrors a professional microservices architecture and highlights a clear separation of concerns. The final dashboard is a powerful proof-of-concept that effectively transforms raw, individual threat data into strategic, actionable intelligence.

This project provided invaluable experience in data engineering, API consumption, database management, and advanced data visualization with tools like Pandas and Streamlit. While the current implementation relies on manual data ingestion, a future iteration could evolve this into a real-time system by having the analysis API write directly to the database. As it stands, the platform is a robust and impressive tool that showcases the critical process of turning data into insight.