Threat Analyzer Platform

BY- SREYA SHYJASH

BTECH CSE CYBER SECURITY

3RD YEAR

INDEX

AIM

The primary aim of this project is to design and build a self-contained, multi-page web application that provides both real-time message analysis and historical threat intelligence visualization.

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

1. **To consolidate threat analysis logic** by integrating structural, link-based, and NLP analysis modules into a single, cohesive Python backend.
2. **To develop a unified, multi-page application using Streamlit**, featuring a "Real-Time Detector" page for individual message analysis and an "Intelligence Dashboard" page for data visualization.
3. **To implement a persistent data layer using SQLite**, where the application automatically initializes the database and saves every analysis result from the real-time detector.
4. **To create an interactive dashboard** that loads data from the database and uses Pandas and Streamlit to display KPIs, dynamic charts showing threat trends and types, and a filterable data explorer.
5. **To ensure seamless data flow**, whereby data entered on the detector page is immediately available for analysis and visualization on the dashboard page.

INTRODUCTION

Modern cybersecurity defense requires a dual capability: the ability to react to immediate, individual threats and the strategic insight to understand overarching attack campaigns. Traditional security tools often operate in silos, forcing a disconnect between tactical response and strategic analysis. An end-user might analyze a suspicious email, but that single data point is often lost, failing to contribute to the organization's collective intelligence. Security analysts, in turn, are left with large volumes of disconnected alerts, making it difficult to identify coordinated attack patterns and emerging trends.

This project addresses this fundamental gap by developing the **"Threat Analyzer Platform,"** a single, integrated application that seamlessly combines real-time threat detection with strategic intelligence dashboarding. By creating a unified ecosystem where every tactical analysis automatically enriches a central data repository, the platform empowers users to not only neutralize immediate threats but also to visualize and understand the broader threat landscape. This transforms a series of isolated events into actionable, data-driven security intelligence.

METHOLODOGY

The project was developed as a unified, single-file Streamlit application (app.py) to maximize code cohesion and simplify deployment. This design choice eliminated the need for a separate backend API and inter-process communication, allowing for a more elegant and efficient architecture.

A. System Architecture:
The application follows a modular, two-page structure managed by Streamlit's session state and sidebar navigation:

- **Page 1: Real-Time Detector:** This page presents a text area for user input. Upon submission, it calls a core run_full_analysis() function, which sequentially executes all the imported analysis modules (structural, link, nlp, and scoring). The result is then both displayed to the user and saved to the database via a save_analysis() function.
- **Page 2: Intelligence Dashboard:** This page's primary function is data visualization. It uses a load_data() function, accelerated by Streamlit's @st.cache_data decorator, to read the entire analysis history from the SQLite database into a Pandas DataFrame. The dashboard's interactive widgets (sliders, date pickers) dynamically filter this DataFrame, and the results are rendered in various charts and tables.

B. Data Persistence and Flow:
A key feature of the methodology was creating a closed-loop data system.

1. **Database Initialization:** The application first calls an initialize_database() function to ensure the SQLite database and its table exist.
2. **Data Writing:** The save_analysis() function is triggered after a successful real-time analysis. It connects to the SQLite database and inserts a new row containing the message body, threat score, and a JSON string of the findings.
3. **Cache Invalidation:** Critically, after saving new data, the save_analysis() function calls st.cache_data.clear(). This tells Streamlit to discard the old, cached DataFrame. The next time the user visits the dashboard page, the load_data() function is forced to re-run, fetching the fresh data from the database and ensuring the visualizations are always up-to-date.

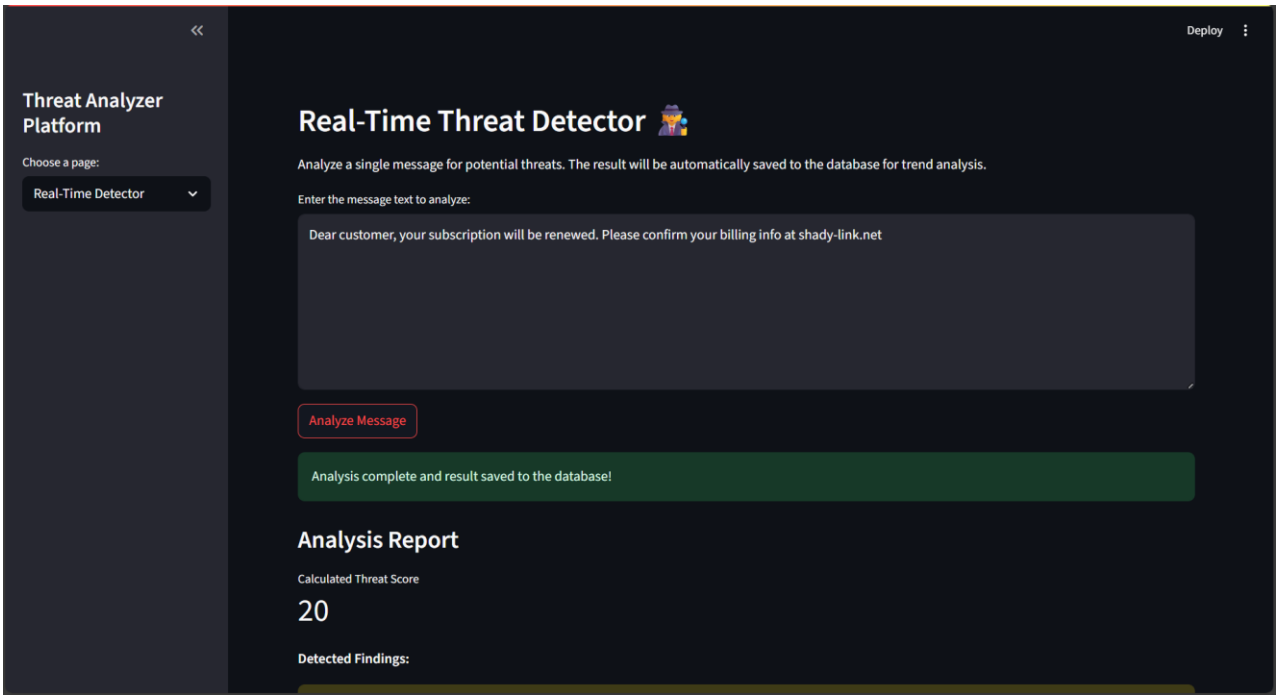
CODE

The codes are provided in the GitHub.

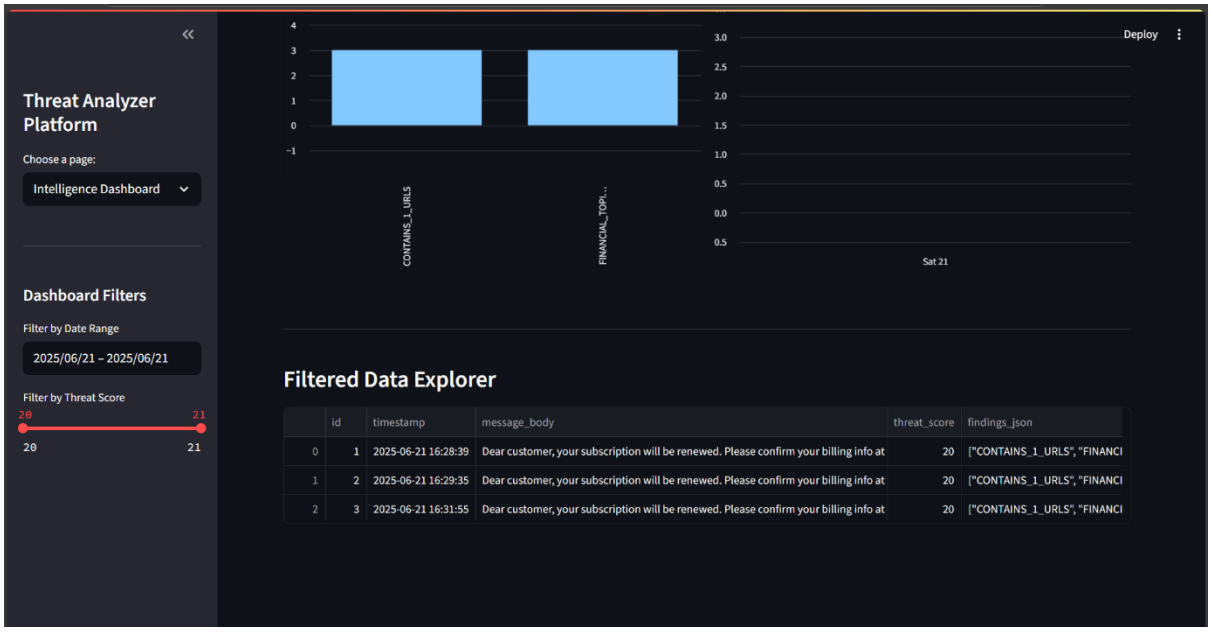
RESULT AND TESTING

The project successfully resulted in a single, fully functional, multi-page "Threat Analyzer Platform" that met all its objectives.

- The application launches via a single command (streamlit run app.py) and presents a clean user interface with clear navigation between the "Real-Time Detector" and "Intelligence Dashboard" pages.
- The **Real-Time Detector** page effectively analyzes user-submitted messages, correctly identifying various threat indicators and calculating an appropriate threat score. Upon analysis, it successfully saves the result to the threat_analysis_data.db file.
- The **Intelligence Dashboard** page successfully loads all historical data from the database. The interactive filters for date range and threat score function correctly, updating all KPIs and charts in real-time in response to user input.
- The data flow between the two pages is seamless. An analysis performed on the detector page is immediately reflected in the dashboard's visualizations upon navigating to it, confirming the success of the cache invalidation strategy.




Deploy
Threat Analyzer Platform
Choose a page:
Intelligence Dashboard
Dashboard Filters
Filter by Date Range
2025/06/21 – 2025/06/21
Filter by Threat Score
20
21
20
21
Threat Intelligence Dashboard
High-Level Metrics
Total Messages in View
3
Average Threat Score
20.00
High-Risk Messages (%)
0.00%
Top Detected Threat Types
5
4
3
2
1
0
−1
CONTAINS_1_URLS
FINANCIAL_TOPI...
Threats Over Time
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Sat 21


Deploy
Threat Analyzer Platform
Choose a page:
Intelligence Dashboard
Dashboard Filters
Filter by Date Range
2025/06/21 – 2025/06/21
Filter by Threat Score
20
21
20
21
4
3
2
1
0
−1
CONTAINS_1_URLS
FINANCIAL_TOPI...
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0.5
Sat 21
Filtered Data Explorer
id
timestamp
message_body
threat_score
findings_json
0
1
2025-06-21 16:28:39
Dear customer, your subscription will be renewed. Please confirm your billing info at
20
["CONTAINS_1_URLS", "FINANCI
1
2
2025-06-21 16:29:35
Dear customer, your subscription will be renewed. Please confirm your billing info at
20
["CONTAINS_1_URLS", "FINANCI
2
3
2025-06-21 16:31:55
Dear customer, your subscription will be renewed. Please confirm your billing info at
20
["CONTAINS_1_URLS", "FINANCI

CONCLUSION

The "Threat Analyzer Platform" project successfully demonstrates the power of integrating tactical tools with strategic platforms. By merging a real-time analysis engine and a historical data dashboard into a single application, it provides a holistic solution for threat management that is both user-friendly and powerful. This unified architecture proved to be highly efficient, eliminating the complexities of API communication and creating a seamless user experience.

The project provides a strong foundation for future work, such as implementing a secure user login system or expanding the analysis engine with more advanced forensic capabilities like email header validation. As a completed proof-of-concept, it is a powerful testament to the practical application of skills across software engineering, data science, and cybersecurity, resulting in a tool that is not only technically impressive but also genuinely useful.