Project Report: AI-Powered Threat Intelligence Correlation Engine

This report evaluates a proof-of-concept web application designed to simplify and correlate threat intelligence data from multiple sources. The application successfully integrates **VirusTotal** and **AbuseIPDB** API data, leveraging **Google Gemini** for real-time, non-technical threat interpretation and actionable advice.

The primary value proposition is the transformation of complex security metrics into accessible, decision-making insights, significantly enhancing the usability of threat intelligence for non-security professionals. While the application is highly functional, a critical security vulnerability related to API key management requires immediate remediation before deployment.

1. Project Scope and Technology Stack

1.1 Core Objectives

The application's primary function is to provide comprehensive security analysis across four key threat vectors:

- **File Analysis:** Upload file scanning and reputation lookups.
- **Hash Analysis:** Checking file hashes (SHA256, MD5, SHA1) against existing databases.
- **URL/Domain Reputation:** Assessing web resource safety.
- **IP Address Reputation:** Correlating data from two distinct threat intelligence feeds.

1.2 Technology Stack

Category	**Component**	**Role**
Frontend/Framework	**Streamlit**	Rapid development and interactive user interface.
Generative AI	**Google Gemini 2.5 Flash**	Interpretation and summarization of raw security data.
Threat Intelligence	**VirusTotal (VT)**	Core reputation data for files, URLs, domains, and IPs.
	AbuseIPDB (AIPDB)	Supplemental IP abuse

		confidence scoring.
Utility	requests, pandas, hashlib	API communication, data structuring, and cryptographic hashing.

1.3 System Architecture Diagram

The application follows a client-server architecture where the Streamlit front-end serves as the intermediary orchestrator between the user input and the multiple external APIs, with the AI model acting as the final data interpretation layer.

Architectural Flow:

1. **User Interaction:** The user submits an IP, URL, File, or Hash via the **Streamlit UI**.
2. **Application Logic:** The Python backend initiates parallel or sequential requests to the external Threat Intelligence platforms.
3. **Data Collection:** Raw data is retrieved from **VirusTotal** and/or **AbuseIPDB**.
4. **AI Interpretation:** The correlated raw data is passed to the **Google Gemini 2.5 Flash API**.
5. **Output Generation:** Gemini returns a simple threat explanation and actionable advice, which is then presented alongside the key metrics summary in the Streamlit UI.

2. Technical Analysis and Functionality

2.1 Multi-Source IP Reputation Correlation

The IP analysis module represents the strongest architectural feature of the application. It executes a parallel analysis from two distinct providers and synthesizes the results, providing a richer context than a single source could offer.

Step	**API Called**	**Output Data**
1	AbuseIPDB	Abuse Confidence Score, Total Reports, Geo-location.
2	VirusTotal	Malicious/Suspicious engine counts, IP reputation score.
3	Gemini AI	Unified explanation, combined threat

		assessment, and suggested action.

2.2 File and Hash Analysis Workflow

The file analysis implements a critical performance optimization using a **cache-check strategy**:

1. On file upload, the **SHA256 hash** is calculated.
2. An immediate lookup is performed to check if a report for that hash already exists on VirusTotal (cache hit).
3. If a cache miss occurs, the application proceeds to the slow path: file upload and a **polling loop** with a maximum timeout of **300 seconds** (5 minutes), accurately reflecting the low-priority nature of the free VT API tier. This mechanism manages user expectations during long-running background tasks.

2.3 User Experience (UX) Enhancements

The application effectively uses the st.status container to communicate multi-stage process progress to the user. This is highly effective in managing perceived latency during long API calls (e.g., file polling) and complex, sequential tasks (e.g., the 3-step IP analysis).

3. Critical Security Vulnerability and Risk Assessment

3.1 Hardcoded Credentials

A **Severe Vulnerability** is present in the codebase. All three external API keys are hardcoded as plain strings at the top of the main script:

- GEMINI_API_KEY
- VIRUSTOTAL_API_KEY
- ABUSEIPDB_API_KEY

Risk Level: CRITICAL

Impact: Public exposure of these keys allows unauthorized usage, leading to:

1. **Financial Liability:** Unauthorized usage of the paid/usage-based APIs (Gemini, VT, AIPDB).
2. **Service Interruption:** API key revocation by the vendor, rendering the application immediately non-functional.

4. Recommendations for Production Readiness

Area	Recommendation	Rationale

Security (P0)	**Remove all hardcoded API keys.** Implement a secure secrets management system (e.g., Streamlit Secrets, environment variables, or a dedicated vault service) for loading all credentials at runtime.	Mandatory step to prevent unauthorized access and protect intellectual property.
AI Prompting	Refine the Gemini system prompt to encourage output that is always explicitly formatted, e.g., using Markdown tables within the response text, to improve parsing and display consistency.	Ensures the AI output is uniform and predictable for a professional interface.
Error Handling	Implement specific handling for rate-limiting errors (HTTP 429) across all API calls, particularly for VirusTotal and AbuseIPDB, which have strict usage quotas.	Provides actionable feedback to the user when quotas are exceeded, rather than generic HTTP errors.
Data Normalization	Expand the convert_to_csv function to include basic sanitization and timestamp conversion, ensuring the raw data exported to CSV is immediately usable for external analysis.	Improves the quality and utility of the data export feature.

This application demonstrates strong technical proficiency and strategic use of AI to solve a real-world problem in threat analysis. Securing the credentials is the immediate priority for project advancement.