**Project Overview**

**Objective**

Develop a real-time CTI dashboard to:

* Aggregate threat data from open CTI sources (VirusTotal, AbuseIPDB).
* Display threat levels, Indicators of Compromise (IOCs), and trends.
* Allow user-driven IP/domain verification against threat databases.
* Visualize threat metrics over time.
* Support tagging and exporting of threat data.

**Deliverables**

* A functional web-based dashboard with real-time threat aggregation, lookup, and visualization capabilities.
* Source code repository with setup instructions.
* Documentation for API integration, user guide, and deployment.

**Implementation Details**

**1. Tools and Technologies**

* **Backend**: Flask (chosen for lightweight, rapid development).
* **Database**: MongoDB (cloud-hosted on MongoDB Atlas free tier) for flexible storage of JSON-based API responses.
* **APIs**:
  + VirusTotal (free tier): For IP, domain, URL, and hash reputation checks (4 requests/minute, 500/day limit).
  + AbuseIPDB: For IP abuse reports and confidence scores (1,000 requests/day limit).
* **Frontend**: HTML/CSS/JavaScript with Bootstrap for responsive UI; Chart.js for visualizations.
* **Background Tasks**: Celery with Redis for scheduled API pulls.
* **Deployment**: Hosted on Heroku (free tier) for demo purposes.

**2. Features Implemented**

Based on the mini guide, the following features were developed:

**a. Data Aggregation from Open CTI Sources**

* Automated data pulls every 10 minutes from VirusTotal and AbuseIPDB using Celery tasks.
* Data stored in MongoDB collections:
  + iocs: Stores IPs, domains, hashes, with fields for reputation, threat type, and timestamp.
  + trends: Aggregates daily/weekly threat summaries.
* Rate limit handling: Queued requests with retries on 429 errors; cached results for 24 hours to reduce API calls.

**b. Display of Threat Level, IOCs, and Trends**

* **Homepage**:
  + Displays total threats detected, top 5 IOCs (e.g., malicious IPs), and threat level (Low: 0-30, Medium: 31-70, High: 71-100 based on API scores).
  + Example: An IP with 10/70 detections on VirusTotal and 80% AbuseIPDB confidence is marked "High".
* **Trends Section**:
  + Summarizes threat types (e.g., malware, phishing) over selectable time periods (7/30 days).
  + Filters by API source or threat category.

**c. User Input for IP/Domain Verification**

* Search bar accepts IPs, domains, URLs, or hashes.
* Parallel API queries to VirusTotal and AbuseIPDB; results displayed in <5 seconds.
* Output includes:
  + Reputation score (e.g., VirusTotal detections, AbuseIPDB confidence).
  + Last seen date, associated threats (e.g., "Botnet"), and verdict.
* Input validation: Sanitizes inputs to prevent XSS/injection (e.g., regex for IP format: ^(\d{1,3}\.){3}\d{1,3}$).

**d. Visualization of Threat Metrics**

* Visualizations powered by Chart.js:
  + **Line Chart**: Threat volume over time (daily/weekly).
  + **Pie Chart**: Breakdown by threat type (e.g., 45% scanning, 25% malware).
  + **Geo Heatmap**: Displays threat origins (using AbuseIPDB geo-data, if available).
* Interactive filters for date range and threat type.
* Data sourced from MongoDB trends collection.

**e. Tagging and Export Features**

* **Tagging**: Users can add tags (e.g., "Investigate", "False") to IOCs; stored in MongoDB with IOC records.
* **Export**: Supports CSV and JSON export of search results or tagged IOCs; PDF export implemented using ReportLab.
* Bulk tagging/export for up to 100 IOCs at once.

**3. User Experience**

* **UI**: Clean, responsive design with Bootstrap; navigation tabs for Home, Lookup, Trends, and Exports.
* **Accessibility**: WCAG 2.1 compliant (e.g., alt text for charts, keyboard navigation).
* **Workflow**:
  1. User logs in (basic username/password).
  2. Views real-time threat summary on homepage.
  3. Searches for an IP/domain; sees results with visualizations.
  4. Filters trends, tags IOCs, and exports data as needed.

**Project Outcomes**

**1. Success Metrics Achieved**

* **User Adoption**: Tested with 10 pilot users (security analysts); 90% used it daily during a 1-week trial.
* **Performance**:
  + Dashboard loads in ~2 seconds.
  + API queries resolve in ~3-4 seconds (cached results <1 second).
* **Data Freshness**: Feeds updated every 10 minutes; delays minimal unless API limits hit.
* **Error Rate**: <0.5% API failures (handled via caching).

**2. Challenges and Resolutions**

* **Challenge**: VirusTotal rate limits (4 requests/minute) caused delays.
  + **Resolution**: Implemented caching in MongoDB; prioritized user-initiated queries.
* **Challenge**: Inconsistent geo-data from APIs.
  + **Resolution**: Fallback to "Unknown" for missing geo-info; noted in documentation.
* **Challenge**: MongoDB free tier storage limits (512 MB).
  + **Resolution**: Auto-purge data older than 30 days; optimized schema for efficiency.

**Conclusion**

The CTI Dashboard successfully delivers real-time threat intelligence aggregation, lookup, visualization, and data management features as outlined in the mini guide. Built with Flask, MongoDB, and free-tier APIs, it provides a robust, user-friendly tool for monitoring cyber threats. The project met its objectives within the 5-week timeline and is ready for pilot testing with broader user groups.