Cyber Threat Intelligence (CTI) Dashboard

1. Introduction

Cyber threats are rapidly evolving, and organizations need tools to collect, analyze,
and visualize Indicators of Compromise (IOCs) in real-time. To address this, we developed a Cyber Threat
Intelligence Dashboard that aggregates live OSINT threat feeds, integrates with public threat intelligence APIs,
and provides visual analytics for better situational awareness.

2. Objectives

- Build a real-time dashboard that collects and stores Indicators of Compromise (IOCs).
- Integrate VirusTotal and AbuseIPDB APIs for IOC enrichment and threat intelligence lookups.
- Enable users to search IPs/domains/URLs, view reputation data, and apply tags for categorization.
- Automate the ingestion of OSINT feeds to ensure continuous data updates.
- Provide visualizations of threat trends and metrics using interactive charts.
- Allow users to manage lookup history and export IOCs/lookup records in CSV format.

3. Tools & Technologies

Python, Flask, MongoDB, PyMongo, APScheduler, VirusTotal API, AbuseIPDB API,
OSINT Feeds, Bootstrap 5, Chart.js, Git & GitHub, .env & .gitignore

4. System Architecture

Workflow:
1. Ingestion → Collect live IOCs from OSINT feeds.
2. Storage → Save IOCs, lookups, and metrics in MongoDB.
3. Enrichment → Query VirusTotal/AbuseIPDB APIs for additional intelligence.
4. Dashboard → Present data with Flask templates, Bootstrap UI, and Chart.js visualizations.
5. User Interaction → Lookup queries, tagging, deletion, and CSV export.
6. Automation → APScheduler updates feeds and metrics periodically.

5. Implementation

Backend (Flask + MongoDB):
- Routes for dashboard, lookup, history, charts, and APIs.
- MongoDB collections for IOCS, LOOKUPS, and METRICS.

API Integrations:
- VirusTotal for IP/domain/URL reports.
- AbuseIPDB for malicious IP reputation.
- Error handling with graceful fallbacks.

Automation:
- APScheduler runs ingestion job every 10 minutes.
- Collects OSINT data and updates metrics.

Frontend:
- Templates: base.html, dashboard.html, history.html, result.html, charts.html.
- Visualizations: ingestion trend line chart and analytics boxplot.

6. Features

- Real-time IOC ingestion from OSINT feeds.
- Lookup IPs/domains/URLs against VirusTotal and AbuseIPDB.
- Automatic classification and tagging of IOCs.
- Lookup history with delete & export options.
- Interactive visualizations with Chart.js.
- Secure handling of API keys via .env.
- Modern, responsive design with Bootstrap 5.

7. Results & Output

- Functioning CTI dashboard that aggregates threat intelligence in real-time.
- Users can perform lookups and see enriched threat data.
- IOC tagging and storage improves long-term tracking.
- Charts provide meaningful insights into ingestion patterns.
- Export functionality allows further offline analysis.

8. Challenges Faced

- Managing API key security → solved with .env and .gitignore.
- Ensuring ingestion updates charts dynamically → Flask JSON API endpoints.
- Making the UI professional → Bootstrap & Chart.js styling.
- Handling multiple IOC types reliably with regex + normalization.

9. Conclusion

The Cyber Threat Intelligence Dashboard provides a lightweight, extensible platform for
real-time threat intelligence collection, analysis, and visualization. It demonstrates how cybersecurity knowledge,
APIs, and modern web technologies can be combined to deliver a practical and interactive tool.
Future enhancements:
- User authentication & role-based access.
- Machine learning for IOC risk scoring.
- Integration with more CTI APIs and feeds.
- Deployment with Docker/Kubernetes for scalability.

10. References

- VirusTotal API Documentation: https://developers.virustotal.com/
- AbuseIPDB API Documentation: https://docs.abuseipdb.com/
- MongoDB Documentation: https://www.mongodb.com/docs/
- Flask Documentation: https://flask.palletsprojects.com/
- Chart.js: https://www.chartjs.org/