# THREATFUSE SENTINEL

# Threat Intelligence Aggregation & Correlation System

**Domain:** Cybersecurity / Blue Team
**Type:** Practical Security Project
**Submitted by:** *Chirayu Paliwal*
**Duration:** Internship Practical Assignment
**Tools & Technologies:** Python, OSINT Threat Feeds, JSON, CSV, REST APIs

## 1. INTRODUCTION

Modern cyber defense does not begin at the endpoint—it begins with context. Security events, logs, and alerts are only meaningful when enriched with external intelligence that explains who, what, and why behind observed activity. This is the role of Threat Intelligence.

Threat intelligence feeds provide indicators such as malicious IP addresses, domains, URLs, file hashes, and campaign metadata collected from open-source intelligence (OSINT), research groups, and security vendors. However, raw feeds are noisy, inconsistent, duplicated, and often overwhelming when consumed directly.

THREATFUSE SENTINEL is a lightweight threat intelligence aggregation and correlation engine designed to ingest multiple threat intelligence sources, normalize indicators into a unified schema, remove duplicates, enrich context, and generate analyst-readable outputs. The project simulates how a SOC or Threat Intelligence Platform (TIP) consolidates external intelligence to support detection, alert enrichment, and threat hunting.

## 2. OBJECTIVE OF THE PROJECT

- To aggregate threat intelligence indicators from multiple sources
- To normalize heterogeneous feed formats into a unified structure
- To deduplicate overlapping indicators across feeds
- To classify indicators by type (IP, domain, hash, URL, etc.)
- To enrich indicators with contextual metadata
- To generate structured outputs usable for SOC analysis and enrichment

### 2.1. PROBLEM STATEMENT

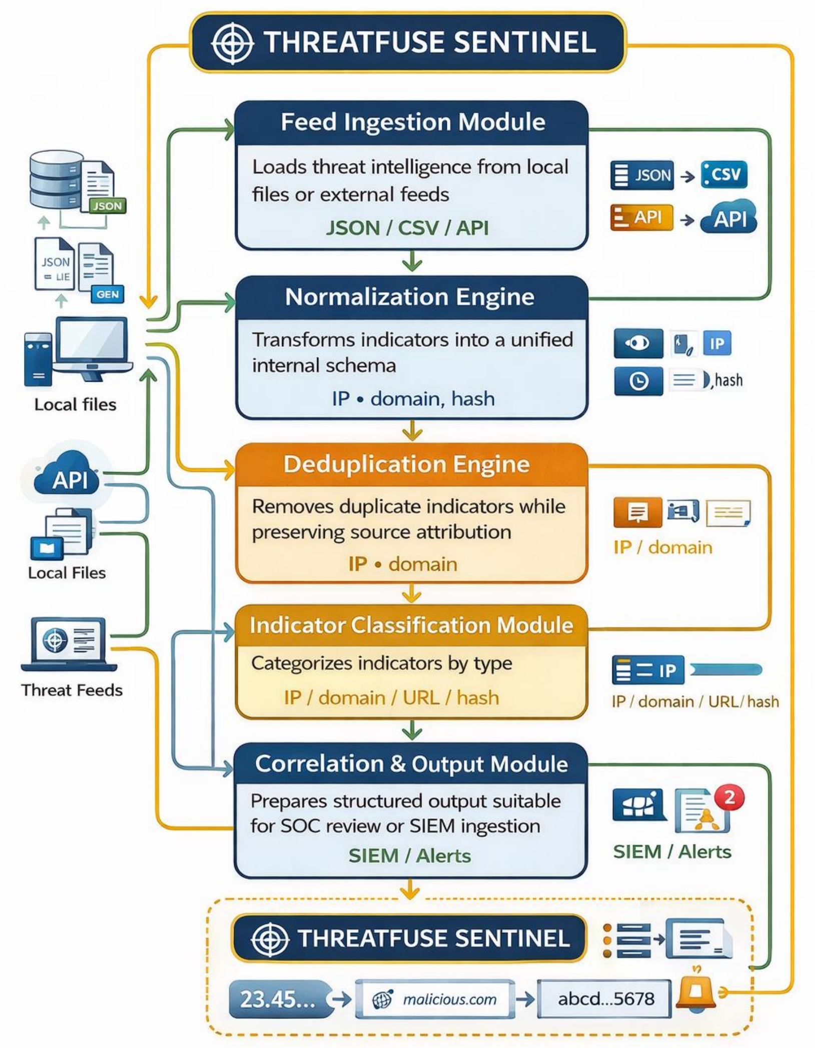
Threat intelligence feeds are not designed for direct consumption by analysts or SIEM tools. Each feed uses different formats, confidence levels, naming conventions, and update cycles. Without aggregation and normalization:

1. Analysts are overwhelmed by duplicate and low-value indicators
2. Correlation across feeds becomes manual and error-prone
3. Threat context is fragmented across sources
4. Intelligence cannot be reliably integrated into detection workflows

Many beginner tools simply download feeds without processing them, producing large lists with little operational value. This project addresses that gap by focusing on **quality, structure, and correlation**, not raw volume.

## 3. ARCHITECTURE

System Architecture of Threat Intelligence Aggregation & Correlation Engine


THREATFUSE SENTINEL
Feed Ingestion Module
Loads threat intelligence from local files or external feeds
JSON / CSV / API
JSON
CSV
API
API
JSON
JSON
GEN
Local files
Normalization Engine
Transforms indicators into a unified internal schema
IP • domain, hash
IP
hash
API
Local Files
Deduplication Engine
Removes duplicate indicators while preserving source attribution
IP • domain
IP / domain
Threat Feeds
Indicator Classification Module
Categorizes indicators by type
IP / domain / URL / hash
IP
IP / domain / URL / hash
Correlation & Output Module
Prepares structured output suitable for SOC review or SIEM ingestion
SIEM / Alerts
2
SIEM / Alerts
THREATFUSE SENTINEL
23.45...
malicious.com
abcd...5678

THREATFUSE SENTINEL aggregates, normalizes, deduplicates, classifies, enriches threat intelligence feeds, and prepares structured outputs suitable for SOC and SIEM integration

THREATFUSE SENTINEL follows a modular pipeline inspired by real-world Threat Intelligence Platforms and SOC enrichment engines.

## 3.1 ARCHITECTURE COMPONENTS

• **Feed Ingestion Module**
Loads threat intelligence from local files or external feeds (JSON / CSV / API-ready structure).

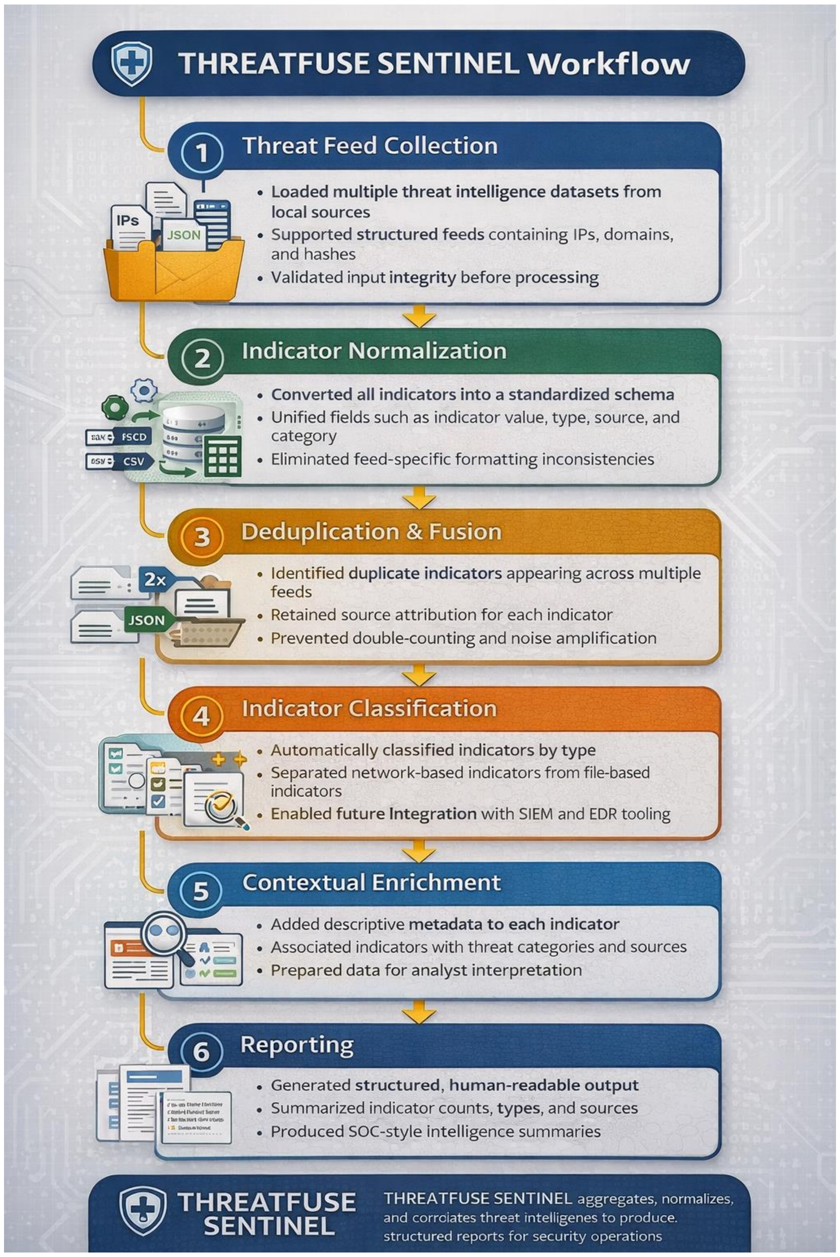
• **Normalization Engine**
Transforms indicators into a unified internal schema regardless of source format.

• **Deduplication Engine**
Removes duplicate indicators across multiple feeds while preserving source attribution.

• **Indicator Classification Module**
Categorizes indicators by type (IP address, domain, URL, file hash).

• **Context Enrichment Module**
Adds metadata such as source, first-seen context, indicator category, and threat labels.

• **Correlation & Output Module**
Prepares structured output suitable for SOC review or SIEM ingestion.



# THREATFUSE SENTINEL Workflow

## 1 Threat Feed Collection

- Loaded multiple threat intelligence datasets from local sources
- Supported **structured feeds** containing IPs, domains, and hashes
- Validated input **integrity** before processing

## 2 Indicator Normalization

- Converted all indicators into a standardized schema
- Unified fields such as indicator value, type, source, and category
- Eliminated feed-specific formatting inconsistencies

## 3 Deduplication & Fusion

- Identified **duplicate indicators** appearing across multiple feeds
- Retained source attribution for each indicator
- Prevented double-counting and noise amplification

## 4 Indicator Classification

- Automatically classified indicators by type
- Separated network-based indicators from file-based indicators
- **Enabled future Integration** with SIEM and EDR tooling

## 5 Contextual Enrichment

- Added descriptive **metadata to each indicator**
- Associated indicators with threat categories and sources
- Prepared data for analyst interpretation

## 6 Reporting

- Generated **structured**, **human-readable output**
- Summarized indicator counts, **types**, and sources
- Produced SOC-style intelligence summaries

THREATFUSE SENTINEL

THREATFUSE SENTINEL aggregates, normalizes, and corrciates threat intelligenes to produce. structured reports for security operations

## 4. METHODOLOGY

### *Step 1: Threat Feed Collection*

- Loaded multiple threat intelligence datasets from local sources
- Supported structured feeds containing IPs, domains, and hashes
- Validated input integrity before processing

### *Step 2: Indicator Normalization*

- Converted all indicators into a standardized schema
- Unified fields such as indicator value, type, source, and category
- Eliminated feed-specific formatting inconsistencies

### Step 3: Deduplication & Fusion

- Identified duplicate indicators appearing across multiple feeds
- Retained source attribution for each indicator
- Prevented double-counting and noise amplification

### *Step 4: Indicator Classification*

- Automatically classified indicators by type
- Separated network-based indicators from file-based indicators
- Enabled future integration with SIEM and EDR tooling

### *Step 5: Contextual Enrichment*

- Added descriptive metadata to each indicator
- Associated indicators with threat categories and sources
- Prepared data for analyst interpretation

### *Step 6: Reporting*

- Generated structured, human-readable output
- Summarized indicator counts, types, and sources
- Produced SOC-style intelligence summaries

## 5. TOOLS & TECHNOLOGIES USED

| Tool / Technology | Purpose |
|---|---|
| **Python 3.14** | Core aggregation and processing logic |
| **OSINT Threat Feeds** | External intelligence sources |
| **JSON / CSV** | Feed storage and parsing |
| **Visual Studio Code & POWERSHELL** | Development environment<br>Testing and validation |

## 6. Threat Model and Assumptions

### Attacker Assumptions

- Attacker infrastructure is reused across campaigns
- Malicious IPs, domains, and hashes appear in public intelligence feeds
- Indicators may appear across multiple sources

### Defender Assumptions

- No real-time blocking or prevention
- Intelligence used for enrichment and hunting
- Focus on visibility, correlation, and context

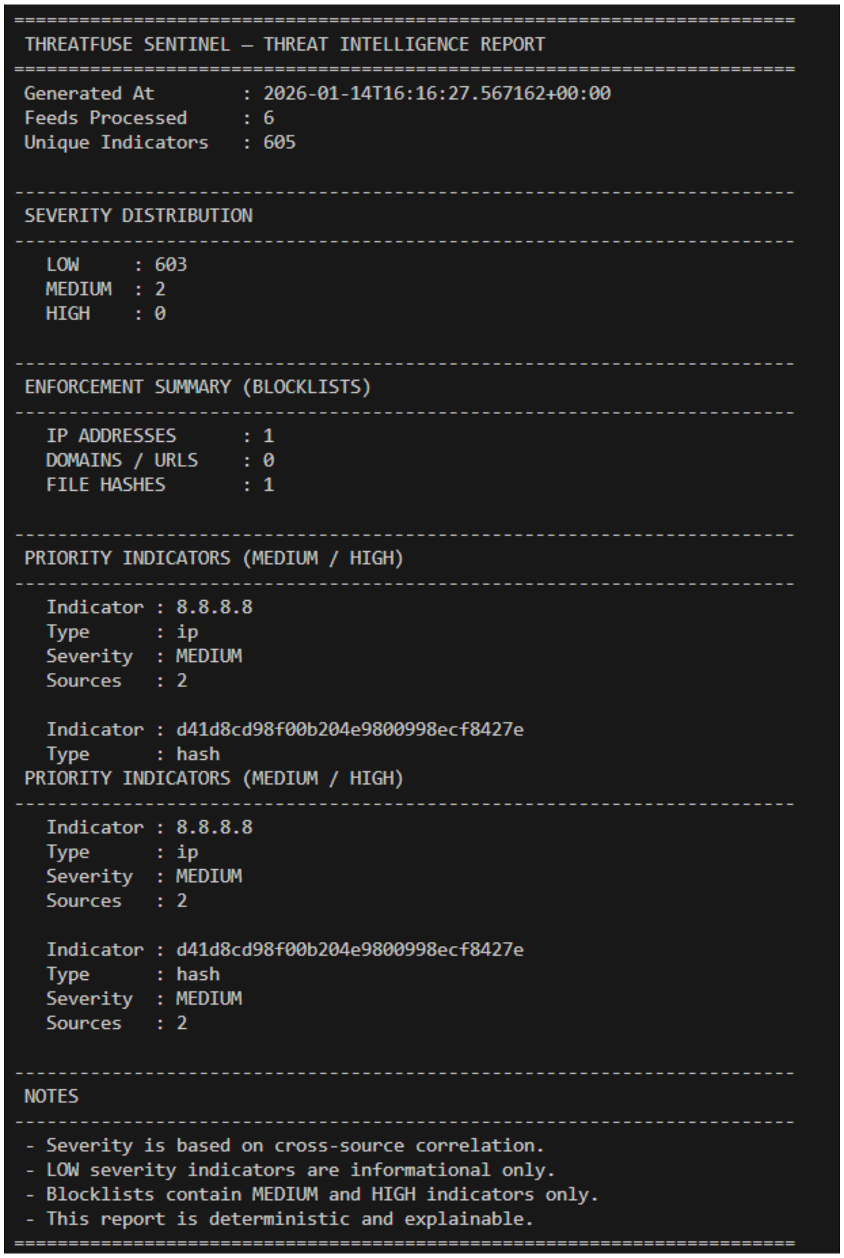
## 7. DETECTION & CORRELATION PHILOSOPHY

THREATFUSE SENTINEL does not treat threat intelligence as binary truth. Instead:

- An indicator's value increases when confirmed by multiple sources
- Context is prioritized over raw indicator count
- Deduplication reduces alert fatigue
- Attribution and explainability are preserved

This mirrors how enterprise SOCs operationalize threat intelligence without blindly trusting feeds.

## 8. Observations



```
========================================================================
 THREATFUSE SENTINEL – THREAT INTELLIGENCE REPORT
========================================================================
 Generated At        : 2026-01-14T16:16:27.567162+00:00
 Feeds Processed     : 6
 Unique Indicators   : 605

------------------------------------------------------------------------
 SEVERITY DISTRIBUTION
------------------------------------------------------------------------
   LOW     : 603
   MEDIUM  : 2
   HIGH    : 0

------------------------------------------------------------------------
 ENFORCEMENT SUMMARY (BLOCKLISTS)
------------------------------------------------------------------------
   IP ADDRESSES      : 1
   DOMAINS / URLS    : 0
   FILE HASHES       : 1

------------------------------------------------------------------------
 PRIORITY INDICATORS (MEDIUM / HIGH)
------------------------------------------------------------------------
   Indicator : 8.8.8.8
   Type      : ip
   Severity  : MEDIUM
   Sources   : 2

   Indicator : d41d8cd98f00b204e9800998ecf8427e
   Type      : hash
 PRIORITY INDICATORS (MEDIUM / HIGH)
------------------------------------------------------------------------
   Indicator : 8.8.8.8
   Type      : ip
   Severity  : MEDIUM
   Sources   : 2

   Indicator : d41d8cd98f00b204e9800998ecf8427e
   Type      : hash
   Severity  : MEDIUM
   Sources   : 2

------------------------------------------------------------------------
 NOTES
------------------------------------------------------------------------
 - Severity is based on cross-source correlation.
 - LOW severity indicators are informational only.
 - Blocklists contain MEDIUM and HIGH indicators only.
 - This report is deterministic and explainable.
========================================================================
```

During testing, THREATFUSE SENTINEL successfully demonstrated:

- Consolidation of multiple threat feeds into a single dataset
- Removal of duplicate indicators across sources

- Clear classification of indicators by type
- Improved readability compared to raw feed consumption

## 9. RESULTS

The system:

- Aggregated threat intelligence from multiple sources
- Normalized heterogeneous data formats
- Reduced noise through deduplication
- Generated structured, analyst-ready output
- Simulated a real-world threat intelligence enrichment pipeline

## 10. CONCLUSION

Through the development of THREATFUSE SENTINEL, I gained a practical understanding of how threat intelligence is operationalized in real-world SOC environments.

This project demonstrated that threat intelligence is not about collecting the most data—it is about reducing chaos into clarity. Normalizing feeds, deduplicating indicators, and preserving context are essential for making intelligence actionable.

By building this system, I learned how SOC teams transform raw OSINT into a usable security context, how intelligence feeds complement detection systems, and why correlation matters more than volume.

THREATFUSE SENTINEL strengthened my blue team skillset by bridging the gap between external intelligence and internal security operations. The project closely mirrors how modern SOCs enrich alerts, support threat hunting, and make informed defensive decisions.