PROJECT

Threat Intelligence Aggregator

AYYAVARI VAMSHI KRISHNA

22 DEC 2025

1. Abstract

Cybersecurity systems rely on accurate and timely threat intelligence to defend against malicious activities. Threat data such as malicious IP addresses and domains are often distributed across multiple sources in different formats, making manual analysis difficult and time-consuming.

This project, **Threat Intelligence Aggregator**, is a Python-based, non-AI system designed to collect, parse, normalize, and correlate threat intelligence indicators from multiple input feeds. The system identifies high-risk indicators based on frequency of occurrence and generates actionable blocklists and summary reports. The project simulates real-world SOC (Security Operations Center) workflows and improves understanding of blue-team defensive security operations.

2. Objectives

The main objectives of this project are:

1. To collect threat intelligence indicators from multiple local feeds
2. To parse Indicators of Compromise (IOCs) such as IP addresses and domains
3. To normalize and remove duplicate indicators
4. To correlate repeated indicators across multiple feeds
5. To identify high-risk indicators based on occurrence count
6. To generate blocklists for security enforcement
7. To generate a final threat intelligence report

3. Problem Statement

Organizations receive threat intelligence data from multiple sources such as open-source feeds, logs, and reports. These data sources often use different formats and contain duplicate or inconsistent information. Manual correlation of such data is inefficient and error-prone.

This project addresses the problem by creating an automated system that consolidates threat data, identifies repeated indicators, and produces structured outputs suitable for defensive security operations.

4. System Architecture

The system follows a modular architecture consisting of the following components:

1. **Feed Loader** – Reads threat data from multiple input files
2. **IOC Parser** – Extracts IP addresses and domain indicators
3. **Normalization Engine** – Cleans and removes duplicate indicators
4. **Correlation Engine** – Counts frequency of indicators
5. **Blocklist Generator** – Generates IP and domain blocklists
6. **Reporting Module** – Generates a final threat intelligence report

Architecture Flow:

Threat Feeds → Parsing → Normalization → Correlation → Blocklists → Report

5. Tools & Technologies Used

- **Operating System:** Kali Linux
- **Programming Language:** Python 3
- **Libraries Used:**
  - os
  - re
  - collections.Counter
- **Editor:** GNU Nano
- **Documentation:** Word / PDF

6. Algorithm

1. Initialize empty list to store all indicators
2. Read all files from the feeds directory
3. Extract indicators line by line
4. Convert indicators to lowercase and remove empty entries
5. Separate IP addresses and domains using regular expressions
6. Remove duplicate indicators
7. Count frequency of each indicator
8. Identify high-risk indicators based on occurrence count
9. Generate IP and domain blocklist files
10. Generate final report summarizing results

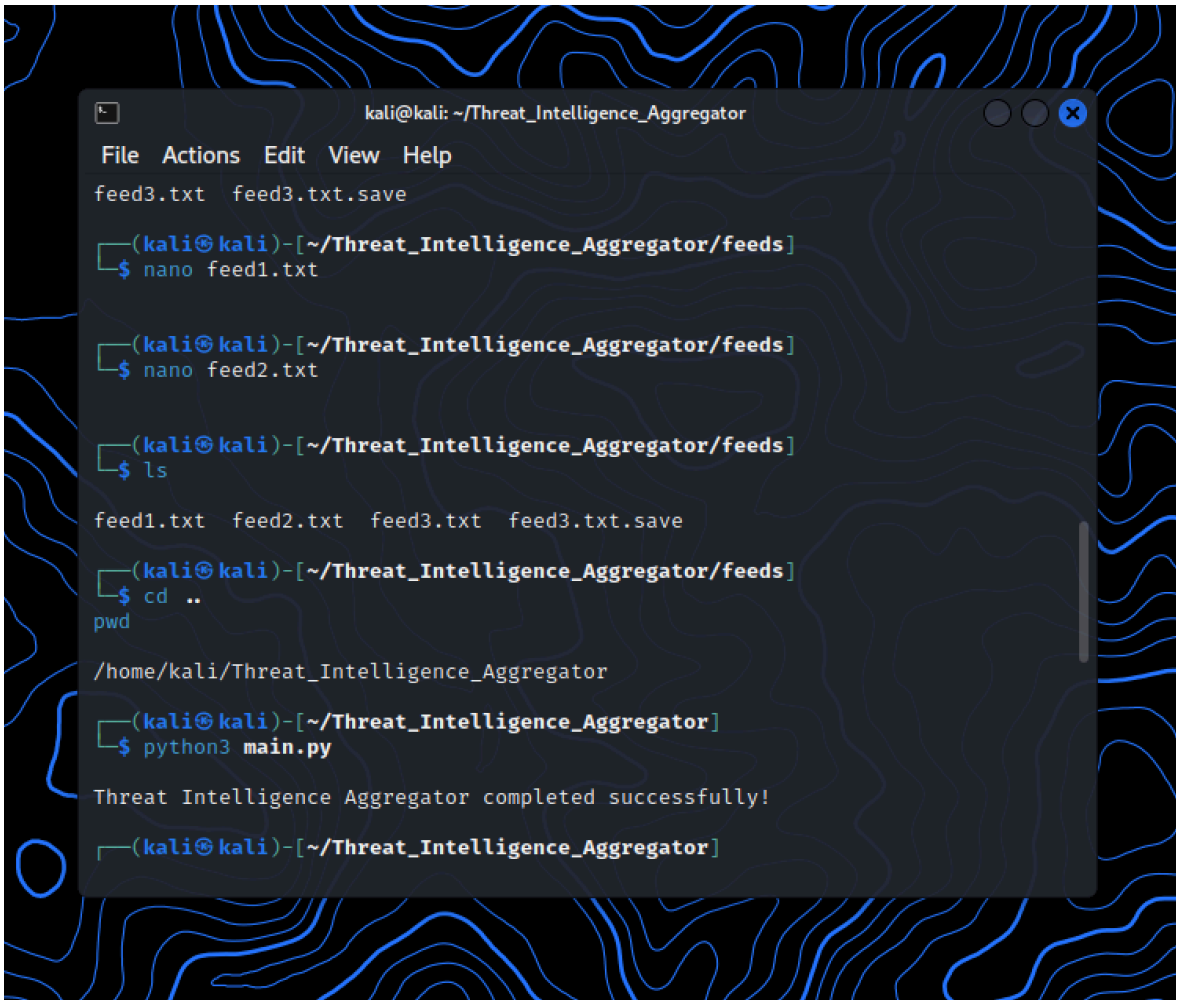
7. Implementation Details

The project is implemented in Python using file handling and regular expression techniques. The system processes multiple text-based threat feeds and aggregates indicators into a unified structure. Frequency-based correlation is used to identify high-risk indicators. The final output includes categorized blocklists and a detailed report.

8. Output

1: Project Execution

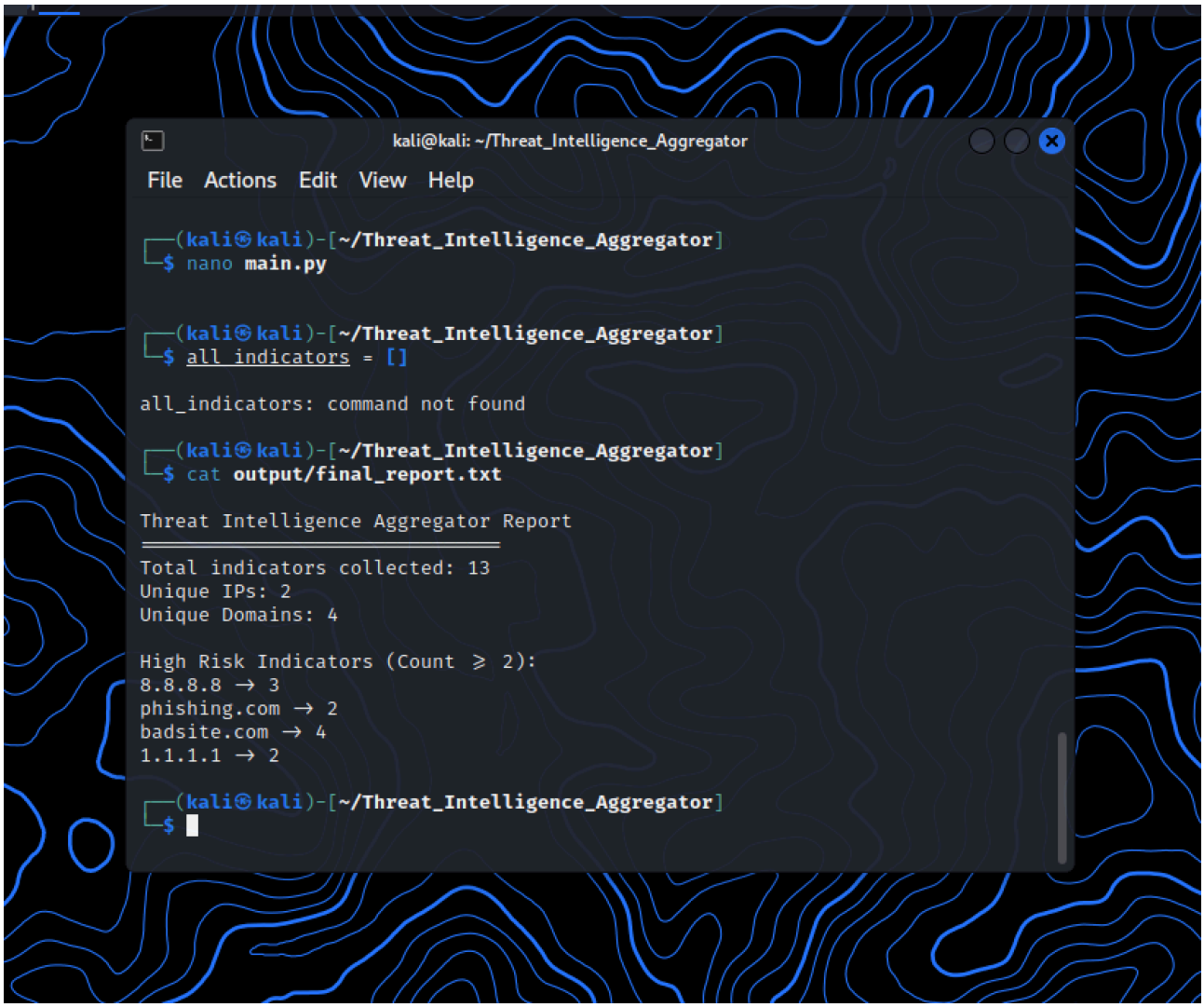
Execution of Threat Intelligence Aggregator



This screenshot shows the successful execution of the Threat Intelligence Aggregator using Python on Kali Linux. The program reads threat intelligence feeds and completes aggregation without errors.

2: Final Threat Intelligence Report

Generated Threat Intelligence Report



This screenshot displays the final threat intelligence report generated by the system. The report includes total indicators collected, unique IPs and domains, and high-risk indicators identified through correlation.

9. Results & Analysis

- Successfully aggregated threat indicators from multiple feeds
- Identified high-risk indicators based on frequency
- Generated structured blocklists for IPs and domains
- Produced a clear and readable threat intelligence report
- Simulated real-world SOC threat intelligence workflow

10. Conclusion

The Threat Intelligence Aggregator project demonstrates an effective approach to consolidating and analyzing threat intelligence data without using artificial intelligence or machine learning. By automating indicator collection and correlation, the system improves efficiency and accuracy in identifying high-risk threats. This project provides strong practical exposure to SOC operations and defensive cybersecurity techniques.

THANKYOU