Threat Intelligence API

LIGHTWEIGHT, SCALABLE, AND DEVELOPER-FRIENDLY

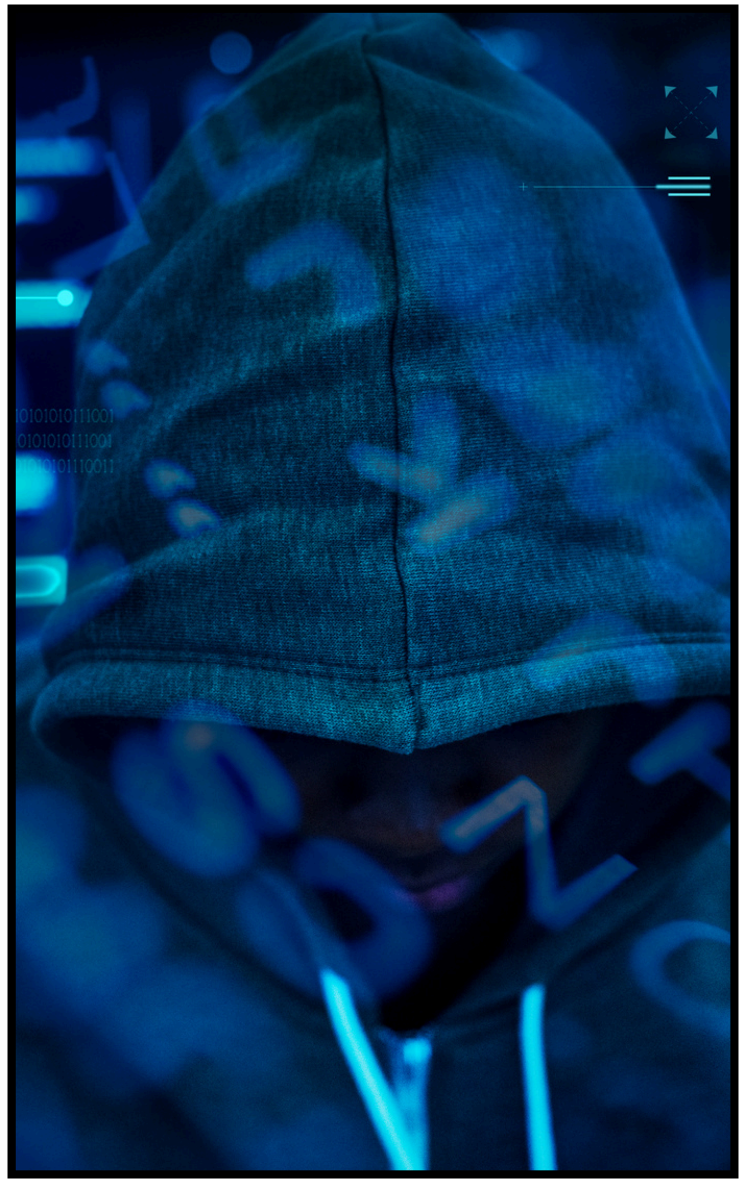
The Problem

- **Challenge**: Security analysts require swift access to accurate threat data.

- **Current Issues**: Existing tools are often expensive, difficult to customise, and without a scalable API design, you're stuck battling outages, slow response times and security risks.

GitHub: sirenc0de

The Solution



GitHub: sirenc0de

Introducing: A streamlined **Threat Intelligence API** (with a functional design).

Key Features

- Built with Flask (Python) and MySQL.
- Core Endpoints:
  - GET /threats: Retrieve all threats.
  - GET /threats/<id>: Fetch specific threat details.
  - POST /threats: Submit new threats.
- Robust error handling and data validation.

GitHub: sirenc0de

Future Enhancements

- **Planned Improvements:**
  - Implementing *JSON Web Token* (JWT) based authentication
  - Introducing pagination and advanced search filters
  - Developing a user-friendly frontend dashboard
  - Integrating logging and monitoring tools

From Functional to Object-Oriented Design

- **Initial Version:**
  - Simple, functional Flask routes
  - Direct DB queries and response logic
- **Refactored Version** (*OOP*):
  - *Threat* class models the domain
  - *Database* class handles all DB ops
  - *APIError* class for clean error handling
  - Modular file structure for scalability

Why Both?

- Functional: great for quick prototypes (i.e. CFG assignment)
- OOP: better structure, easier to test, scalable for teams.