Autonomous Mini-SOC / Network Sensor Suite

Project Specification & Milestone Plan

1. Overview

This document outlines a full engineering specification for a distributed, multi-language Autonomous Mini-SOC / Network Sensor Suite. The system includes a C++ endpoint agent, a Python FastAPI backend, an asynchronous event pipeline, a rule-based detection engine, persistent alert storage, and a user-facing dashboard.

2. System Architecture Summary

The system is composed of multiple cooperating components designed to mimic real-world SOC/EDR pipelines:

- C++ Agents collect telemetry (processes, ports, resource usage)
- Events are securely transmitted to a Python backend
- FastAPI ingestion feeds an asynchronous event pipeline
- Normalization + enrichment layers prepare data for detection
- A YAML/JSON rule engine generates alerts
- Heartbeat tracking monitors agent availability
- A dashboard or CLI provides threat visibility

3. Component Breakdown

3.1 C++ Agent Responsibilities

- Telemetry collection: processes, ports, resource usage
- Threaded local event queue
- Batching + JSON/Protobuf serialization
- Secure communication with backend
- Optional: local rule evaluation

3.2 Python Backend Responsibilities

- FastAPI ingestion endpoints
- Async queue for event processing
- Normalization + enrichment
- YAML/JSON rule engine (multi-stage)

- SQLite/Postgres event storage
- Heartbeat tracking
- Dashboard or CLI interfaces for visibility

4. Milestones

Milestone 1 — C++ Agent Foundation (Week 1–2)

- Create agent skeleton (threading + event model)
- Implement process/port enumeration
- Create event batching + JSON/Protobuf serialization
- Implement basic network sending to backend

Milestone 2 — Python Backend Ingestion Layer (Week 2–3)

- Create FastAPI server with /ingest endpoint
- Implement async queue
- Store raw events in DB
- CLI tool to read stored events

Milestone 3 — Event Processing & Detection Engine (Week 3–4)

- Create decoder/normalizer pipeline
- Implement YAML rules
- Generate alerts based on rule matches
- Alert storage + retrieval

Milestone 4 — Heartbeat System (Week 4–5)

- Agent periodic heartbeats
- Backend tracking of agent status
- Offline detection

Milestone 5 — Secure Communication (Week 5–6)

- Implement TLS or libsodium encryption
- Agent enrollment + secret storage
- Authenticated ingestion

Milestone 6 — Dashboard (Week 6–7)

- Web dashboard or HTMX UI
- Agent list + heartbeat indicator
- Alert viewer
- Event filtering

Milestone 7 — Advanced Add-ons (Week 8+)

- Command dispatch to agents
- Sysmon XML parser integration
- GeoIP enrichment
- Custom binary protocol

5. Deliverables

- Cross-platform C++ endpoint agent
- FastAPI backend with ingestion + event pipeline
- Detection engine with YAML rules
- Persistent alerts + event storage
- Dashboard or CLI interface
- Documentation + architecture diagrams