Project: Agentic SOC
Author: Sunil Tiwari
Date: 12/27/2025

Part 1: The Core Infrastructure (The Foundation)

Below are the permanent components already running in my system (NVIDIA DGX Spark).

NVIDIA DGX (AARCH64/ARM64 Architecture): The high-performance compute platform unlike standard PCs (x86).

Wazuh (SIEM/XDR): It collects logs from the Linux endpoints, analyzes them against rules, and generates alerts. It provides the telemetry the AI agents will analyze.

Ollama: The inference engine. Allows me to run Large Language Models (LLMs) like Llama 3.2 locally. In my SOC environment, using Ollama ensures data privacy. Hence, the security alerts never leave the network.

Part 2: The Agentic Framework (The Logic):

CrewAI: This is the orchestration framework. It allowed me to define “Agents” with specific roles, goals, and backstories. It manages the Process, ensuring “Agent A” passes its findings to “Agent B” in a structured way.

LangChain (Community): This is a library that acts as a bridge between the AI Agents and the local Ollama models. It handles the communication protocol so the agents can talk to the LLM.

Part 3: The Python Ecosystem (The Glue)

Following are the fundamental concepts I used:

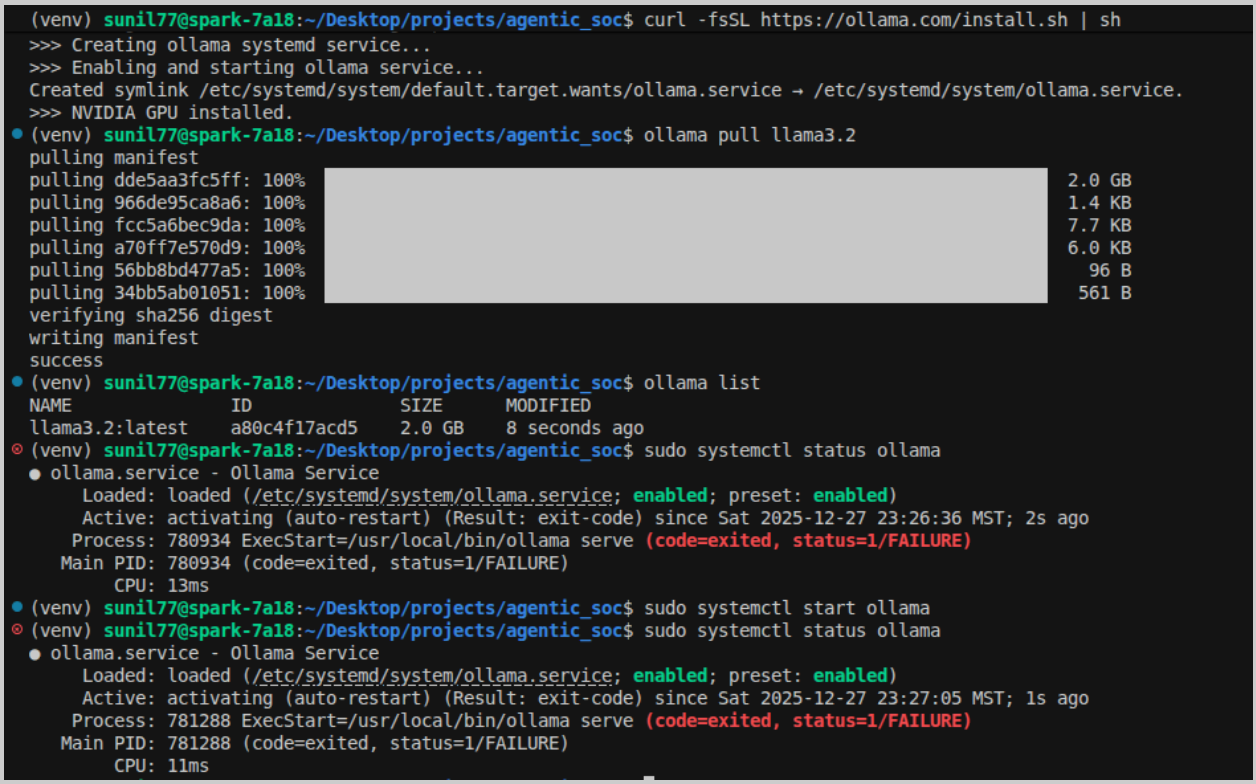
Virtual Environments (`venv`): An isolated “container” for Python which ensures that the libraries installed for this project don’t break other projects in my system.

JSON: This is the language of logs. Wazuh outputs alerts in `.json` format.

Tail -f Logic: It is a method of reading a file as it grows.

Part 3: The Tools

Llama 3.2: It plays the role of a “Brain”. I used it because it provides optimal reasoning capabilities for security triage.



CrewAI: It plays the role of the “Manager”. I chose this because it coordinates multiple agents to prevent “single-point-failure” reasoning.
Pydantic: It plays the role of a “Validator”. It is used inside the CrewAI because it ensures the data passed between agents is formatted correctly.

Git: This is a “Historian” as it tracks every change in the code for version control and recovery.

Decision Matrix: OODA Loop (Observe, Orient, Decide, Act)

Observation: Wazuh detects a brute force attack.
-> Orientation: Triage Agent extracts the IP.
-> Decision: Researcher Agent checks if this IP is a known threat.
-> Action: Commander Agent that triggers a block.

The Project Life Cycle - Phases (Maturity Model)

Phase 1 – Static (Foundations)
Establishing core infrastructure by setting up Linux, Wazuh, and baseline log collection.

Phase 2 – Observable (Telemetry)
Enrolling endpoints and generating meaningful security telemetry and real-world security events.

Phase 3 – AI-Ready (Local Intelligence)
Deploying local LLMs (Ollama / Llama 3.2) on DGX systems to enable on-premise AI processing.

Phase 4 – Automated (Response Automation)
Implementing Python-based automation that detects alerts and triggers predefined response actions.

Phase 5 – Autonomous (Agentic SOC) (Current)
Operating a multi-agent SOC where AI agents collaborate, reason, and coordinate incident response.

Phase 6 – Intelligent (RAG & Memory)
Enhancing agents with retrieval-augmented generation and memory, enabling them to learn from past attacks and ingest external intelligence such as PDF threat reports.

```
triage_specialist:
  role: >
    Wazuh Alert Triage Specialist
  goal: >
    Extract key security entities (IPs, Rule IDs, Usernames) from raw Wazuh JSON alerts.
  backstory: >
    You are a meticulous SOC analyst. Your job is to take complex JSON data and
    simplify it into a structured format for the rest of the team. You never guess;
    you only report the facts found in the logs.

threat_researcher:
  role: >
    Cyber Threat Researcher
  goal: >
    Analyze security entities to identify MITRE ATT&CK patterns and threat severity.
  backstory: >
    You are an expert in threat intelligence. You take the IPs and Rule IDs from
    triage and determine if they represent a known attack pattern or a false positive.
```

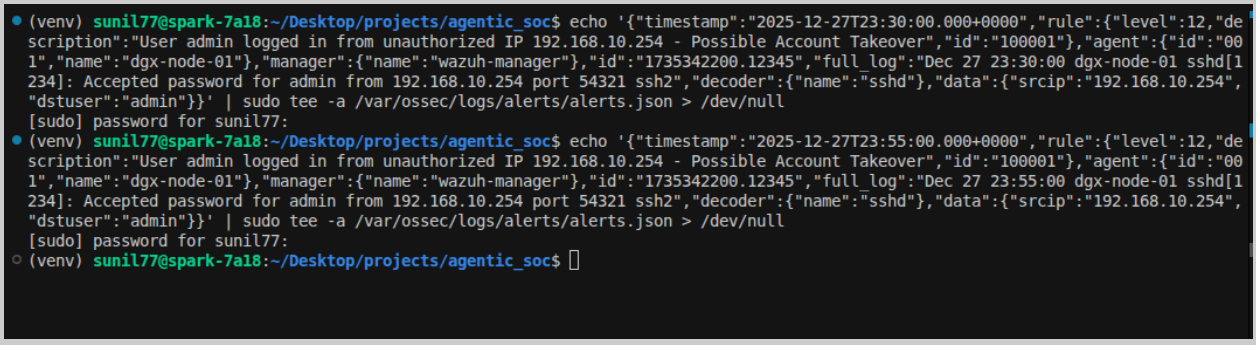
```
incident_commander:
  role: >
    SOC Incident Commander
  goal: >
    Make the final decision to Block, Watch, or Ignore an alert based on the team's findings.
  backstory: >
    You are the final authority. You balance security needs with business uptime.
    You provide clear, authoritative instructions on what action the system must take.
```

TESTING

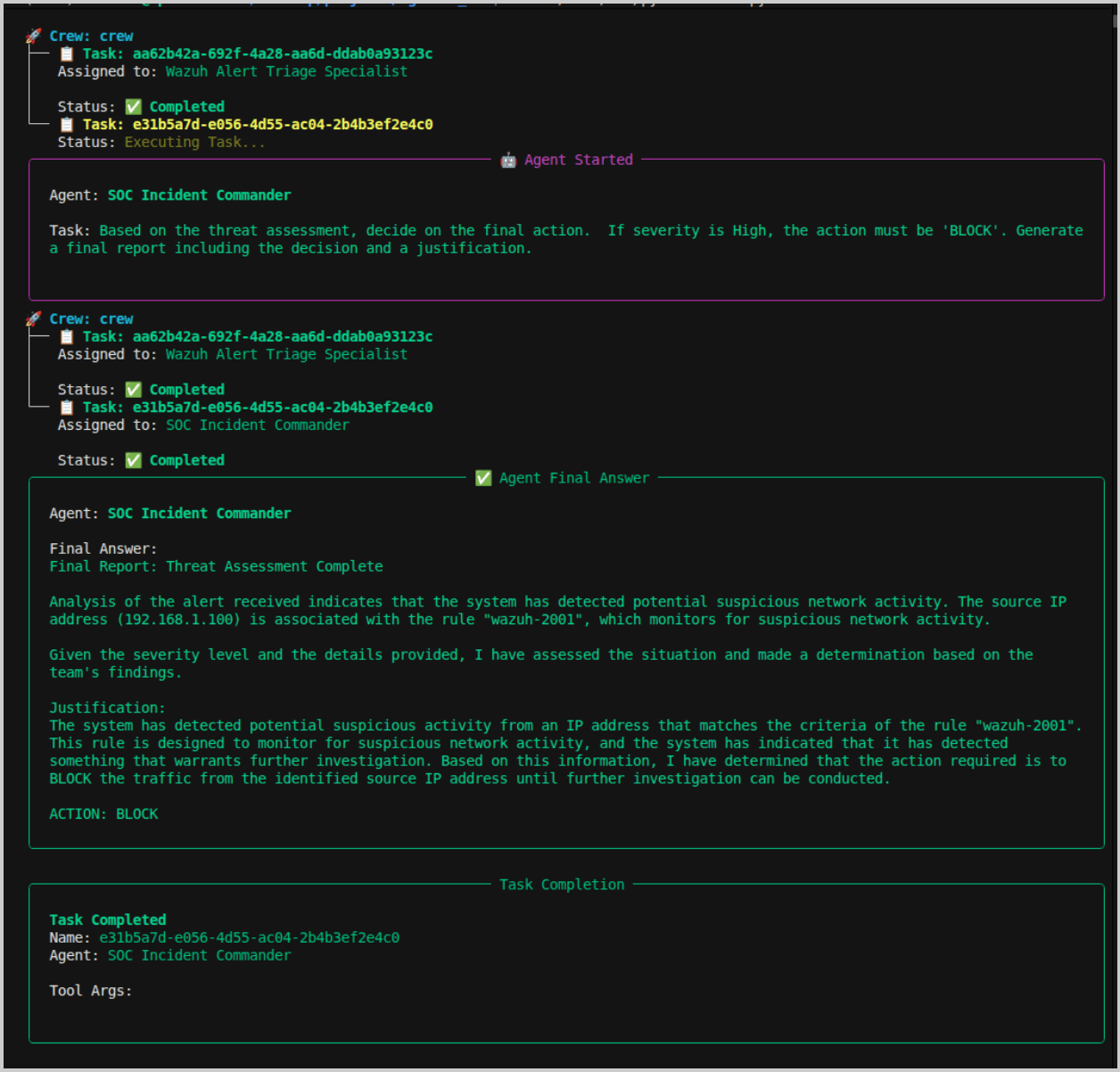
1. The "Stress Test" Logic

We will manually append a Level 12 (High Severity) alert into your Wazuh log file. This bypasses the need for a real attacker while forcing your Python "File Watcher" to trigger the Agentic SOC loop.

```
echo '{
"timestamp":"2025-12-27T23:30:00.000+0000",
"Rule":{
        "level":12,
        "description":"User "admin" logged in from unauthorized IP 192.168.10.254 - Possible Account Takeover",
        "Id":"100001"
},
"Agent":{
        "id":"001",
        "Name":"dgx-node-01"
},
"Manager":{"name":"wazuh-manager"},
"Id":"1735342200.12345",
"full_log":"Dec 27 23:30:00 dgx-node-01 sshd[1234]: Accepted password for admin from 192.168.10.254 port 54321 ssh2",
"Decoder":{"name":"sshd"},
"Data":{"srcip":"192.168.10.254",
"Dstuser":"admin"}
}' | sudo tee -a /var/ossec/logs/alerts/alerts.json > /dev/null
```



Detailed execution traces:


Crew: crew
Task: aa62b42a-692f-4a28-aa6d-ddab0a93123c
Assigned to: Wazuh Alert Triage Specialist
Status: Completed
Task: e31b5a7d-e056-4d55-ac04-2b4b3ef2e4c0
Status: Executing Task...
Agent Started
Agent: SOC Incident Commander
Task: Based on the threat assessment, decide on the final action. If severity is High, the action must be 'BLOCK'. Generate a final report including the decision and a justification.
Crew: crew
Task: aa62b42a-692f-4a28-aa6d-ddab0a93123c
Assigned to: Wazuh Alert Triage Specialist
Status: Completed
Task: e31b5a7d-e056-4d55-ac04-2b4b3ef2e4c0
Assigned to: SOC Incident Commander
Status: Completed
Agent Final Answer
Agent: SOC Incident Commander
Final Answer:
Final Report: Threat Assessment Complete
Analysis of the alert received indicates that the system has detected potential suspicious network activity. The source IP address (192.168.1.100) is associated with the rule "wazuh-2001", which monitors for suspicious network activity.
Given the severity level and the details provided, I have assessed the situation and made a determination based on the team's findings.
Justification:
The system has detected potential suspicious activity from an IP address that matches the criteria of the rule "wazuh-2001". This rule is designed to monitor for suspicious network activity, and the system has indicated that it has detected something that warrants further investigation. Based on this information, I have determined that the action required is to BLOCK the traffic from the identified source IP address until further investigation can be conducted.
ACTION: BLOCK
Task Completion
Task Completed
Name: e31b5a7d-e056-4d55-ac04-2b4b3ef2e4c0
Agent: SOC Incident Commander
Tool Args:

```
--- AGENTIC RESPONSE REPORT ---
Final Report: Threat Assessment Complete

Analysis of the alert received indicates that the system has detected potential suspicious network activity. The source IP address
 (192.168.1.100) is associated with the rule "wazuh-2001", which monitors for suspicious network activity.

Given the severity level and the details provided, I have assessed the situation and made a determination based on the team's find
ings.

Justification:
The system has detected potential suspicious activity from an IP address that matches the criteria of the rule "wazuh-2001". This
rule is designed to monitor for suspicious network activity, and the system has indicated that it has detected something that warr
ants further investigation. Based on this information, I have determined that the action required is to BLOCK the traffic from the
 identified source IP address until further investigation can be conducted.

ACTION: BLOCK
------------------------------
```

Crew Completion

Crew Execution Completed
Name: crew
ID: 057d7fa0-7986-4696-8891-7c922ec6261d
Tool Args:
Final Output: Final Report: Threat Assessment Complete

Analysis of the alert received indicates that the system has detected potential suspicious network activity. The source IP address (192.168.1.100) is associated with the rule "wazuh-2001", which monitors for suspicious network activity.

Given the severity level and the details provided, I have assessed the situation and made a determination based on the team's findings.

Justification:
The system has detected potential suspicious activity from an IP address that matches the criteria of the rule "wazuh-2001". This rule is designed to monitor for suspicious network activity, and the system has indicated that it has detected something that warrants further investigation. Based on this information, I have determined that the action required is to BLOCK the traffic from the identified source IP address until further investigation can be conducted.

ACTION: BLOCK

Tracing Status

Info: Tracing is disabled.

To enable tracing, do any one of these:
• Set tracing=True in your Crew/Flow code
• Set CREWAI_TRACING_ENABLED=true in your project's .env file
• Run: crewai traces enable

Execution Traces

🔍 **Detailed execution traces are available!**

View insights including:
- Agent decision-making process
- Task execution flow and timing
- Tool usage details

Would you like to view your execution traces? [y/N] (20s timeout):

Tracing Preference Saved

Info: Tracing has been disabled.

Your preference has been saved. Future Crew/Flow executions will not collect traces.

To enable tracing later, do any one of these:
• Set tracing=True in your Crew/Flow code
• Set CREWAI_TRACING_ENABLED=true in your project's .env file
• Run: crewai traces enable

Achievement:

- Environment Isolation: A stable Python virtual environment (venv) that handles complex dependency conflicts between crewai, litellm, and openai.
- Permission Bridging: A secure method for your AI agents to read protected SIEM logs without compromising root security.
- Local Inference: A low-latency connection to a local Llama 3.2 instance running on the DGX GPUs via Ollama.
- Orchestration Logic: A verified YAML-based configuration system that allows for modular agent and task management.