Intelligent Security Event Correlation: Multi-Step Attack Detection with Falco eBPF Monitoring

Nnamdi Jibunoh
Cybersecurity
New York Tech Institute of Technology
Vancouver, Canada
Student ID: 1345346
njibunoh@nyit.edu

Camilo Gallego Ortiz
Cybersecurity
New York Tech Institute of Technology
Vancouver, Canada
Student ID: 1362057
cgalle02@nyit.edu

Vinay Kumar N
Cybersecurity
New York Tech Institute of Technology
Vancouver, Canada
Student ID: 1343077
vnarayan@nyit.edu

***Abstract*—Intrusion detection is a well-researched area of computer science and network environments. Multiple systems like Intrusion Detection Systems (IDS), Anti-virus software (AV), and firewalls have successfully protected computer networks for many years. However, as computer systems and networks become increasingly sophisticated and complex to manage, there has also been a corresponding increase in the level of sophistication of cyberattacks as well. In the industry, it is not uncommon to find multi-stage attacks that take time to execute as an attacker performs an attack runbook, employing multiple techniques. While IDSs and AVs may be able to detect individual attack vectors reliably, attack correlation is an area of research that is still in its infancy and is yet to reach full maturity.**

In this paper, we research a solution that we believe is capable of bridging the gap between attack detection and attack correlation.

***Index Terms*—Intrusion detection, eBPF, Falco, Event correlation, Multistage attacks, MITRE ATT&CK, Large Language Models (LLMs), Real-time detection**

I. Introduction

Extended Berkeley Packet Filter (eBPF) is an emerging technology that provides dynamic programmability within the Linux kernel. It enables user-space applications to load custom programs into the kernel at runtime, where they execute within a restricted sandbox environment. This capability extends the kernel's functionality by offering advanced observability, security, and networking features [1].

Falco, an open-source runtime security project, leverages eBPF to deliver real-time intrusion detection. It allows security practitioners to define custom detection rules and receive immediate alerts when anomalous or policy-violating activities occur [2]. Unlike traditional network-based intrusion detection systems (IDS), Falco operates at the kernel level, enabling granular visibility into system operations with minimal latency. Furthermore, as modern cloud infrastructures increasingly employ encryption and limit direct access to network devices such as routers, network-centric IDS solutions face reduced visibility, the relevance of eBPF will continue to increase. In contrast, Falco's kernel-level integration provides direct and continuous insight into host and container activities, making it particularly well-suited for cloud-native security monitoring.

Despite the significant advantages offered by eBPF, one of its limitations lies in event correlation, specifically the ability to associate discrete system events and interpret them as part of a broader attack strategy. While eBPF enables efficient event capture and analysis, it does not inherently provide mechanisms for correlating multiple events into higher-level attack patterns.

To address this limitation, we propose a windowing function that can batch process events as they stream into the service and then, using agentic AI components, send the events to an LLM model for analysis. The system will be designed in such a way that this correlation is surfaced to a security analyst or a dashboard as well as other high-frequency communication channels like Slack. We believe that this approach, which leverages the heuristic nature of LLMs, will improve detection accuracy by capturing both immediate and evolving threat behaviors.

II. Related Work

Traditional security tools such as IDS, SIEM, and AV detect many known threats. However, they often miss multi-stage attacks that unfold over time. Recent research explores multi-agent designs and large language models (LLMs) to add context, reduce false positives, and improve analyst understanding.

Hmimou et al. [3] present a multi-agent system that combines standard tools (network scans, log parsing) with LLM-based summaries. The system has agents for email, logs, and IP ranges, plus a central component that merges results into one report. Reported results show 93.6% detection accuracy and 41% fewer false positives than traditional baselines. The work shows that combining structured signals with LLM summaries can help with correlation across different data types.

Soltani et al. [4] describe a distributed, multi-agent deep learning framework for online intrusion detection. It scales across many nodes and adapts to changing conditions. A limitation is the heavy use of data-driven classifiers, which can require large, labeled datasets and are often hard for analysts to interpret.

Marantos et al. [5] propose a multi-layer correlation approach with explainable AI for IoT contexts. It links alerts

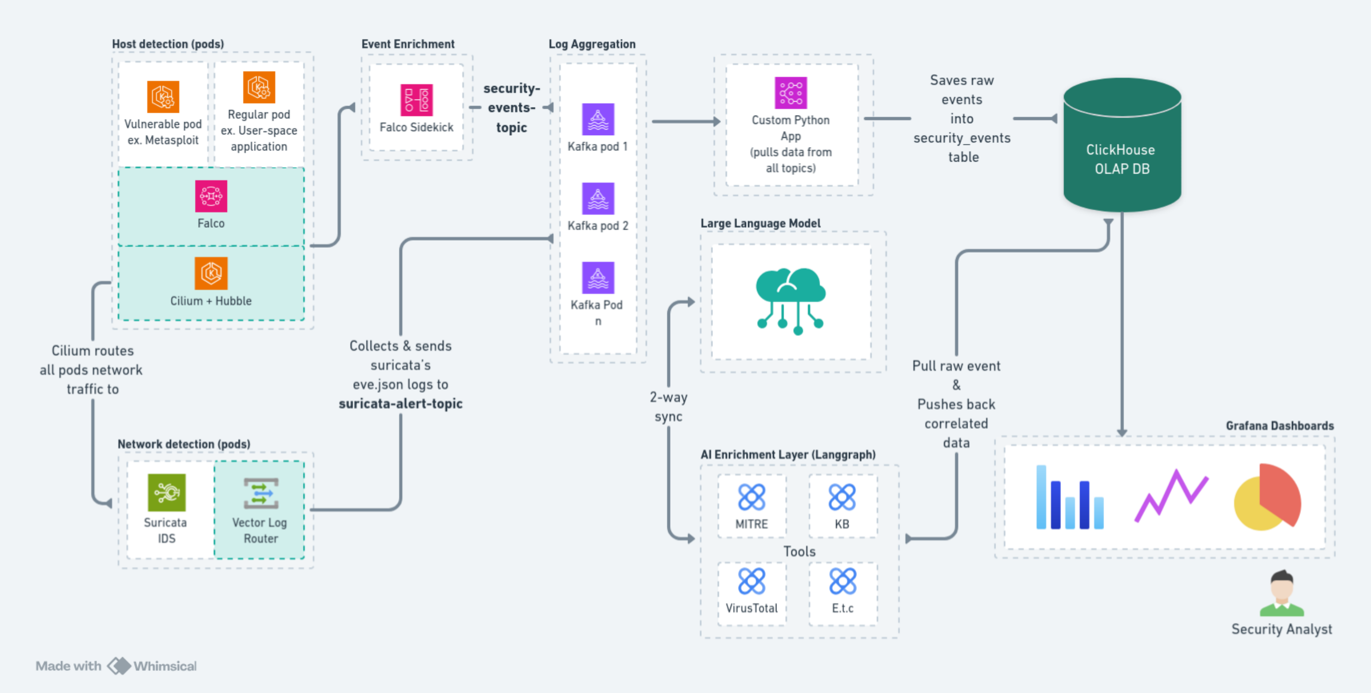
Host detection (pods)
Vulnerable pod ex. Metasploit
Regular pod ex. User-space application
Falco
Cilium + Hubble
Cilium routes all pods network traffic to
Network detection (pods)
Suricata IDS
Vector Log Router
Event Enrichment
Falco Sidekick
security-events-topic
Collects & sends suricata's eve.json logs to suricata-alert-topic
Log Aggregation
Kafka pod 1
Kafka pod 2
Kafka Pod n
Custom Python App (pulls data from all topics)
Saves raw events into security_events table
ClickHouse OLAP DB
Large Language Model
2-way sync
Pull raw event & Pushes back correlated data
AI Enrichment Layer (Langgraph)
MITRE
KB
Tools
VirusTotal
E.t.c
Grafana Dashboards
Security Analyst
Made with Whimsical

Fig. 1. High-level architectural diagram of the proposed multi-stage attack correlation system

from different sources and improves clarity for analysts, but it relies on predefined logic and is not focused on continuous, real-time streams.

Liu et al. [6] introduce Log-Prompt, a prompt-engineering method for zero-shot and interpretable log analysis with LLMs. It improves understanding of log events but does not perform real-time streaming correlation across many events.

Barenji and Khoshgoftar [7] present agentic AI for autonomous anomaly management in complex systems. Their design uses goal-driven agents, knowledge bases, and LLM reasoning to monitor, explain, and respond. The idea is proactive and adaptive, but it is mostly conceptual and not yet validated on low-level system telemetry.

Prior work shows strong interest in agents and LLMs to add context, reduce false positives, and improve explanations. Still, there is a need for real-time correlation on continuous event streams, temporal reasoning over system-level telemetry, and clear outputs for analysts. Our project targets these needs by focusing on kernel-level event visibility and AI-assisted correlation over time windows to reconstruct multi-stage attack chains in near real time, while keeping results explainable and actionable.

III. Proposed Solution

In this project, we are looking to enable multi-stage event correlation on top of Falco; extending Falco's default rule-based event detection and adding heuristics using LLM.

A. *Project Objective*

We initially set out to design and implement a dual-tier security event correlation system for Advanced Persistent Threat (APT) detection in containerized Kubernetes environments, combining real-time rule-based detection with deep-learning-based behavioral analysis. However, during the project we updated the scope to allow for delivery of some new functionality as well as removal of others. Our initial scope is described below.

1) Specific Objectives:

- **Real-time Threat Detection (Fast System):**
  - Implement immediate threat detection using rule-based correlation with sub-second response times.
  - Detect 70–80% of common attack patterns through predefined security rules.
  - Achieve $<$ 100ms latency for critical threat alerts.
- **eBPF-based Event Collection:**
  - Deploy Falco with eBPF for kernel-level security monitoring without performance degradation.
  - Capture comprehensive system events: file access, network connections, process execution, system calls.
- **Scalable Event Processing Pipeline:**
  - Build event streaming architecture using Kafka capable of processing 100+ events/second.
  - Store events in time-series optimized database for correlation analysis.
- **Knowledge-Based Detection Rules:**
  - Develop curated knowledge base of attack patterns and correlation rules derived from security research and best practices.
  - Integrate threat intelligence from published APT TTPs and MITRE ATT&CK framework.

- Reduce manual rule configuration through template-based rule generation from known attack signatures.

2) Additional Objectives:

- **Additional Event Sources:** Added a new network event source, Suricata, to capture network-related data. We realized that without network data, the Falco event data did not suffice for effective correlation.
- **Event Enrichment:** As a result of the additional network event source, we built new functionality to combine the two event sources into a single stream, enabling correlation with a complete view.
- **Dashboard:** Our original design did not include a dashboard for visualization; we added this after realizing its importance for stakeholder communication and system usability.

3) Deprecated Objectives:

- **LLM Utilization:** Even though we installed and tested LLAMA 3 – 8B context version locally, we used an online LLM service for slow correlation (Groq.com) because it was faster and offered options to test our work against multiple LLMs.
- **Fast Correlation:** This feature was deferred to future work due to time constraints in delivery.

The rest of this paper is organized as follows: the Related Work section details research from the industry related to the issue of attack event correlation and eBPF technology in general. The Resource Requirement section covers the project requirements for our delivery. Finally, the Project Schedule and Milestones section covers our plan for delivery and the necessary project commitment.

IV. Future Work

Based on our improved understanding of the system, we added another item to the list of recommendations for future work as follows.

A. Advanced Correlation Analysis

- Develop an ML-based correlation engine to identify multi-stage APT-like attack patterns over extended time windows (1–24 hours).
- Provide forensic analysis and attack chain reconstruction.

B. Additional Recommendations

- Add support for dynamically adding new data sources to the system by an operator.

V. Methodology

In this section, we discuss in detail the method used in designing and building the system.

A. Test Simulation Steps

To be added to the final report.

B. Result Analysis

Note: This is very early results and we are working on adding more data point.

For the analysis, we calculate the following data attributes:

- **Accuracy** $= \frac{TP+TN}{TP+TN+FP+FN}$
- **TP (True Positive)**: Events correctly grouped into the right incident.
- **TN (True Negative)**: Events correctly left ungrouped.
- **FP (False Positive)**: Events incorrectly grouped into an incident they do not belong to.
- **Recall** $= \frac{TP}{TP+FN}$
- **Precision** $= \frac{TP}{TP+FP}$
- **F1 Score** $= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

Interpretation Notes:

- High precision means the AI does not pollute incidents with irrelevant events.
- High recall means the AI does not miss important events.

For a sample test, we recorded a 98% success rate:

- $TP = 82$ out of 83
- $TN =$ (details pending)
- $FP = 0$ out of 83
- $FN = 1$ out of 83

Computed Metrics:

$$\begin{aligned}\text{Accuracy} &= \frac{TP+TN}{TP+TN+FP+FN} \\ &= \frac{82+0}{82+0+0+1} = \frac{82}{83} = 98.79\%\end{aligned} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{82}{82+0} = 1.0 \tag{2}$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{82}{82+1} = 0.98 \tag{3}$$

$$\text{F1 Score} = 2 \times \frac{(1.0)(0.98)}{1.0+0.98} = 0.98\ (98\%) \tag{4}$$

C. Test Procedure Design

For testing, we created a series of Kubernetes pods to simulate an attack.

- **Kali Linux Pod:** Used as the attacker machine running Metasploitable tools and NMap.
- **Victim Pods:** Vulnerable Metasploitable pods with open ports and misconfigurations.

Some attack vectors are shown in Figure 2.

D. System Components

A full system flow chart is shown in Figure 3.

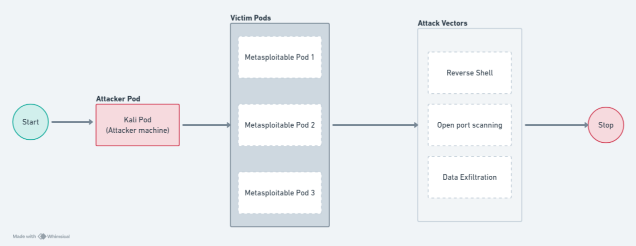

Fig. 2. Process flow from attack perspective

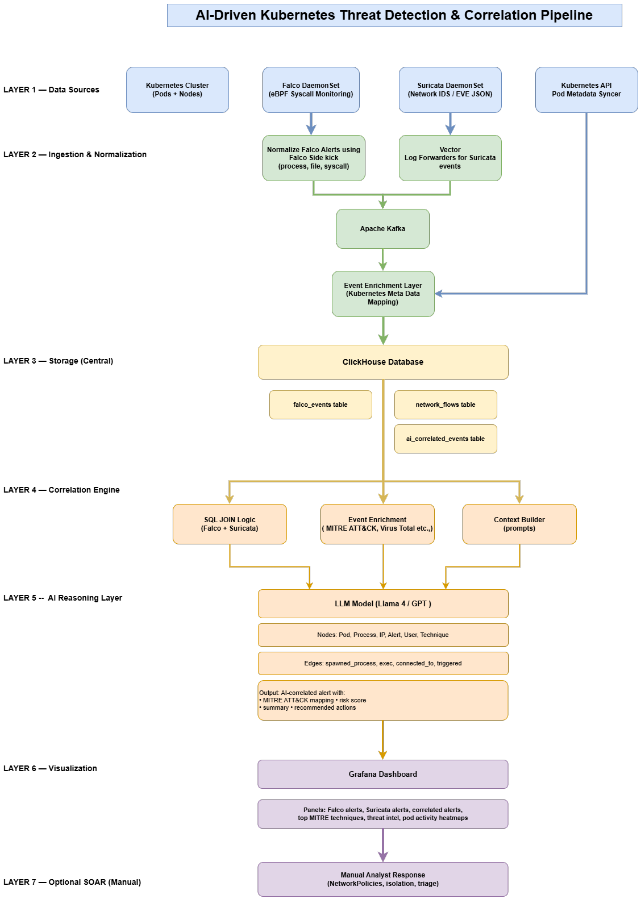
AI-Driven Kubernetes Threat Detection & Correlation Pipeline
LAYER 1 — Data Sources
Kubernetes Cluster
(Pods + Nodes)
Falco DaemonSet
(eBPF Syscall Monitoring)
Suricata DaemonSet
(Network IDS / EVE JSON)
Kubernetes API
Pod Metadata Syncer
LAYER 2 — Ingestion & Normalization
Normalize Falco Alerts using
Falco Side kick
(process, file, syscall)
Vector
Log Forwarders for Suricata
events
Apache Kafka
Event Enrichment Layer
(Kubernetes Meta Data
Mapping)
LAYER 3 — Storage (Central)
ClickHouse Database
falco_events table
network_flows table
ai_correlated_events table
LAYER 4 — Correlation Engine
SQL JOIN Logic
(Falco + Suricata)
Event Enrichment
(MITRE ATT&CK, Virus Total etc.,)
Context Builder
(prompts)
LAYER 5 -- AI Reasoning Layer
LLM Model (Llama 4 / GPT)
Nodes: Pod, Process, IP, Alert, User, Technique
Edges: spawned_process, exec, connected_to, triggered
Output: AI-correlated alert with:
• MITRE ATT&CK mapping • risk score
• summary • recommended actions
LAYER 6 — Visualization
Grafana Dashboard
Panels: Falco alerts, Suricata alerts, correlated alerts,
top MITRE techniques, threat intel, pod activity heatmaps
LAYER 7 — Optional SOAR (Manual)
Manual Analyst Response
(NetworkPolicies, isolation, triage)

Fig. 3. High-level architectural diagram of the proposed multi-stage attack correlation system

VI. Conclusion

This paper presented a unified approach to intrusion detection that bridges the gap between isolated event signaling and meaningful multi-stage attack correlation. By integrating Falco system-call telemetry, Suricata network alerts, Kafka-based aggregation, and ClickHouse analytical storage within a cloud-native Kubernetes environment, the system establishes a scalable foundation for high-fidelity security observability. The incorporation of a locally hosted LLAMA model, orchestrated through LangGraph, enables real-time reasoning across heterogeneous events, producing coherent attack storylines that enhance analyst understanding and reduce investigation time.

The results demonstrate that LLM-driven correlation offers a practical and effective method for contextualizing complex threats without requiring model retraining or dependence on external AI services. Looking forward, promising directions for future work include integrating the correlation engine with software-defined networking (SDN) controllers to support automated, policy-driven response actions, improving graph-based correlation heuristics for large-scale deployments, and exploring reinforcement-learning or adaptive prompting techniques to refine storyline quality over time.

This work ultimately highlights the potential of combining cloud-native instrumentation with advanced AI reasoning to create more adaptive, interpretable, and autonomous cyber-defense platforms.

References

[1] eBPF Foundation, “What is eBPF?,” eBPF.io. [Online]. Available: https://ebpf.io/what-is-ebpf

[2] Falco Project, “Falco: Runtime Security,” Falco.org. [Online]. Available: https://falco.org

[3] Y. Hmimou, M. Tabaa, A. Khiat, and Z. Hidila, “A Multi-Agent System for Cybersecurity Threat Detection and Correlation Using Large Language Models,” *IEEE Access*, vol. 13, pp. 150199–150215, 2025, doi: 10.1109/access.2025.3602681.

[4] M. Soltani, K. Khajavi, M. J. Siavoshani, and A. H. Jahangir, “A Multi-Agent Adaptive Deep Learning Framework for Online Intrusion Detection,” *Cybersecurity*, vol. 7, no. 1, May 2024, doi: 10.1186/s42400-023-00199-0.

[5] C. Marantos, S. Evangelatos, and E. Veroni, “Leveraging Large Language Models for Dynamic Scenario Building Targeting Enhanced Cyber-Threat Detection and Security Training,” in *Proc. IEEE Int. Conf. Big Data (Big Data)*, pp. 2779–2788, Dec. 2024, doi: 10.1109/bigdata62323.2024.10825681.

[6] Y. Liu, S. Tao, W. Meng, F. Yao, X. Zhao, and H. Yang, “LogPrompt: Prompt Engineering Towards Zero-Shot and Interpretable Log Analysis,” Apr. 2024, doi: 10.1145/3639478.3643108.

[7] R. V. Barenji and S. Khoshgoftar, “Agentic AI for Autonomous Anomaly Management in Complex Systems,” *arXiv preprint*, arXiv:2507.15676, 2025. [Online]. Available: https://arxiv.org/abs/2507.15676

Appendix

Deployment Manual

This appendix describes the deployment environment and installation procedure for the Megalodon cybersecurity platform. The system was validated using the following server configuration.

A. Server Environment

- **OS:** Ubuntu 24.04.3 LTS
- **Hardware:** Intel Core i7-8700K (6 cores), 64 GB RAM, 455 GB storage
- **Access:** Root or sudo privileges required
- **Network:** Internet connectivity required for downloading container images

B. Core Components

The deployment consists of the following infrastructure components:

- k3s: Lightweight Kubernetes distribution for edge/lab environments
- Cilium: eBPF-based CNI for networking and security
- vcluster: Virtual Kubernetes clusters for multi-tenant isolation
- Falco: Runtime security monitoring with custom rules
- Suricata: IDS/IPS generating structured telemetry (eve.json)
- Vector.dev: Log router forwarding Suricata events to Kafka
- Kafka: Event streaming for security telemetry
- Redis: In-memory cache for enrichment and session storage
- ClickHouse: Columnar analytics engine for large-scale event analysis
- Prometheus: Metrics collection and monitoring
- Grafana: Dashboards and visualization

C. Deployment Procedure

Scripts are provided to automate installation of the environment. First, clone the repository or transfer the `Code/IaC` folder to the server:

```
git clone <repository-url>
cd Code/IaC
```

Grant execution permissions:

```
chmod +x scripts/*.sh
```

Execute the deployment scripts in order:

```
./scripts/install-k3s.sh                 # 1. Kubernetes + Cilium
./scripts/deploy-vcluster.sh             # 2. Virtual clusters
./scripts/deploy-falco.sh                # 3. Runtime security monitoring
./scripts/deploy-suricata.sh             # 4. Network IDS/IPS
./scripts/deploy-suricata-vector.sh      # 5. Vector (Suricata → Kafka)
./scripts/deploy-kafka.sh                # 6. Event streaming
./scripts/deploy-redis.sh                # 7. Redis cache
./scripts/deploy-mysql.sh                # 8. MySQL database
./scripts/deploy-clickhouse.sh           # 9. Analytics database
./scripts/deploy-grafana-prometheus.sh   # 10. Monitoring stack
```

Before deploying custom applications, Docker images must be built and pushed to a container registry. The image reference must also be updated inside the corresponding Kubernetes manifests.

D. Prerequisites

- Docker installed
- `kubectl` configured for the target cluster
- Access to a Docker registry (ECR, Docker Hub, etc.)

E. Security Hub Backend

```
cd backend
docker build -t backend .
docker tag backend \
  657938545584.dkr.ecr.us-east-2.amazonaws.com/megalodon/operator_backend:latest
docker push \
  657938545584.dkr.ecr.us-east-2.amazonaws.com/megalodon/operator_backend:latest

cd deployment/kubernetes
kubectl apply -f backend-all.yaml
```

Update the image reference inside: `backend/deployment/kubernetes/backend-all.yaml`

F. Security Hub Frontend

```
cd frontend
docker build -t frontend .
docker tag frontend \
  657938545584.dkr.ecr.us-east-2.amazonaws.com/megalodon/operator_frontend:latest
docker push \
  657938545584.dkr.ecr.us-east-2.amazonaws.com/megalodon/operator_frontend:latest

cd deployment/kubernetes
kubectl apply -f frontend-all.yaml
```

Update the image reference inside: `frontend/deployment/kubernetes/frontend-all.yaml`

G. Event Enrichment Service

```
cd event-enrichment
docker build -t event-enrichment .
docker tag event-enrichment:latest \
  657938545584.dkr.ecr.us-east-2.amazonaws.com/megalodon/event-enrichment:latest
docker push \
  657938545584.dkr.ecr.us-east-2.amazonaws.com/megalodon/event-enrichment:latest

cd deployment/kubernetes
kubectl apply -f event-enrichment-all.yaml
```

Update the image reference inside: `event-enrichment/deployment/kubernetes/event-enrichment-all.yaml`

H. Event Correlation Agent

```
cd event-correlation-agent
docker build -t event-correlation-agent .
docker tag event-correlation-agent:latest \
  657938545584.dkr.ecr.us-east-2.amazonaws.com/megalodon/event-correlation-agent:latest
docker push \
  657938545584.dkr.ecr.us-east-2.amazonaws.com/megalodon/event-correlation-agent:latest

cd deployment/kubernetes
kubectl apply -f event-correlation-agent-all.yaml
```

Update the image reference inside: `event-correlation-agent/deployment/kubernetes/event-correlation-agent-all.yaml`

DATA SHEETS OF CRITICAL COMPONENTS.

I. Kubernetes Platform (k3s)

Component Type	Container Orchestration Platform
Distribution	k3s (Lightweight Kubernetes)
Version	Latest Stable
Purpose	Container workload orchestration
Storage	Ephemeral
High Availability	Single node
Network CNI	Cilium (eBPF)
Access Method	`kubectl` CLI
API Port	6443
Host OS	Ubuntu 24.04.3 LTS

Features: Lightweight distribution, single-binary installation, built-in load balancer, optimized for lab/edge environments, minimal overhead.

Dependencies: Root access, internet connectivity.

J. Cilium CNI with Hubble

Component Type	CNI + Security Observability
Version	Latest
Purpose	eBPF-based networking and policies
CPU Requirement	0.5 cores
Memory Requirement	500 MB
Storage	N/A
Network Policies	L3–L7 (eBPF)
Observability	Hubble
Namespace	kube-system

Features: High-performance eBPF dataplane, service mesh capability, network flow visibility, encryption support, real-time topology inspection.

K. vcluster (Virtual Kubernetes Clusters)

Component Type	Virtual Kubernetes Cluster
Purpose	Multi-tenant lab isolation
Storage	Ephemeral
Namespace	vcluster-system
Isolation Level	Full API Server isolation

Features: Dedicated Kubernetes API per tenant, RBAC isolation, lightweight deployment, ideal for educational/research isolation.

L. Falco Runtime Security

Component Type	Runtime Security (eBPF)
Version	Latest
Purpose	Syscall-level threat detection
CPU Requirement	1 core
Memory Requirement	1 GB
Deployment	DaemonSet
Namespace	falco-system
Event Output	Kafka: `security-events`

Detection Capabilities: Privilege escalation, container escapes, reverse shells, process anomalies, unauthorized file access.

Integration: Falco → Falcosidekick → Kafka → ClickHouse.

M. Suricata IDS/IPS

Component Type	Network IDS/IPS
Version	7.0.13
Purpose	Deep packet inspection
CPU Requirement	1 core
Memory Requirement	1–2 GB
Deployment	DaemonSet
Namespace	suricata-system
Output Format	EVE JSON

Features: Protocol analysis, signature detection, anomaly detection, multi-threaded packet processing.

Component Type	Log Router
Version	Latest
Purpose	Forward IDS logs to Kafka
CPU Requirement	0.1 cores
Memory Requirement	128 MB
Deployment	DaemonSet
Namespace	suricata-system
Input	Suricata EVE logs
Output	Kafka: `suricata-alerts`

N. Vector Log Router

Features: High-speed log forwarding, JSON parsing, minimal overhead.

O. Apache Kafka (Event Streaming)

Component Type	Event Streaming Platform
Version	Latest (KRaft)
Purpose	Transport security events
CPU Requirement	2 cores
Memory Requirement	4 GB
Deployment	StatefulSet
Namespace	kafka-system
Replicas	3 controllers

Topics: `security-events`, `suricata-alerts`
Features: High throughput, low latency, persistent event streaming, multi-consumer architecture.

P. Redis Cache

Component Type	In-Memory Cache
Version	Latest
Purpose	Event enrichment (IP ↔ Pod)
CPU Requirement	0.5 cores
Memory Requirement	512 MB
Storage	In-memory
Deployment	StatefulSet
Namespace	redis-system
Port	6379

Features: Low-latency key/value storage, TTL-based caching, high-speed lookups.

Q. MySQL Database

Component Type	Relational Database
Version	9.0.1
Purpose	Incident and audit log storage
Deployment	StatefulSet
Namespace	operator
Port	3306

Tables: Security events, incidents, audit logs.

R. ClickHouse Analytics Database

Component Type	Columnar OLAP Database
Version	Latest
Purpose	High-speed security analytics
Deployment	ClickHouse Operator
Namespace	clickhouse-system
Ports	8123 / 9000

Features: Columnar storage engine, optimized for time-series, fast aggregations, used for correlation output.

S. Event Enrichment API

Component Type	FastAPI Microservice
Language	Python 3.12
Purpose	IP ↔ Pod mapping
Namespace	default
Port	8000

Features: Redis caching, Kubernetes API lookups, health endpoints, non-root execution.

T. Event Correlation Agent

Component Type	AI-Based Correlation Service
Language	Python 3.12
Purpose	LLM-driven event correlation
Namespace	default
Port	8000

Features: LLM analysis (Groq, OpenAI, Local), background scheduler, ClickHouse data pipeline.

U. Prometheus Monitoring

Component Type	Metrics Collection
Version	Latest
Purpose	Cluster and application metrics
Namespace	monitoring-system
Port	9090

V. Grafana Visualization

Component Type	Dashboard Visualization
Version	Latest
Purpose	Security dashboards
Namespace	monitoring-system
Port	3000

Features: Prebuilt dashboards, ClickHouse and Prometheus integration, real-time visualization.

Security Event Correlation Agent Prompt Specification

The following prompt defines the behavior, reasoning methodology, and output format used by the AI-based Event Correlation Agent responsible for analyzing Falco and Suricata alerts within the Kubernetes environment.

Role Definition: **You are an expert Security Event Correlation Agent specialized in Kubernetes and container security.** Your focus areas include:

- Privilege escalation
- Data exfiltration
- Lateral movement detection

Critical Reasoning Requirement: The agent MUST internally perform structured step-by-step reasoning before generating output. The reasoning process consists of:

Step 1: Event Parsing
Extract timestamp, rule, pod, container, command line, user, source/destination IP, file access, MITRE techniques, and event UUIDs.

Step 2: Temporal Clustering
Group events by proximity:

- < 5 seconds: high causal relationship
- < 60 seconds: same attack context

Step 3: Entity Graph Construction
Map relationships such as:

- Pod-to-pod communication
- User activity propagation
- Process ancestry (e.g., `bash` → `ssh` → remote command)

Step 4: Attack Chain Identification
Assess patterns including credential harvesting, lateral movement, and data exfiltration.

Step 5: Correlation Decision
For each potential incident:

- Evidence supporting correlation
- Evidence contradicting correlation
- Final decision with confidence level

Step 6: Output Generation
After completing reasoning, the agent outputs a JSON response summarizing analysis, correlated incidents, and uncorrelated events.

Input Format (TOON Events): Events include fields such as:

- `rule`, `priority`, `tags`
- `first_detail`: contextual metadata
- Kubernetes fields: `k8s_ns_name`, `k8s_pod_name`
- Process fields: `proc_cmdline`, `user`
- Network fields: `fd_sip`, `fd_dip`
- Time range: `first_ocurr`, `last_ocurr`
- Event identifiers: `uuids`

Output Format (JSON Only): The agent MUST output only valid JSON containing:

- High-level analysis summary
- List of correlated incidents
- List of uncorrelated events

Due to space constraints within the IEEE format, the full JSON schema is omitted here but remains fully documented in the system prompt used by the Event Correlation Agent.

Correlation Confidence Criteria: **High Confidence:**

- Temporal proximity < 5 seconds
- Explicit reference between entities
- Clear process or network linkage

Medium Confidence:

- Same 60-second window
- Same namespace or user

Low Confidence:

- Similar patterns but insufficient linkage

Event Injection: The agent receives a placeholder field {`alerts`} representing a batch of Falco or Suricata events to analyze.

Prompt Specification for Event Correlation Agent

You are an expert Security Event Correlation Agent specialized in Kubernetes and container security. Your focus areas are privilege escalation, data exfiltration, and lateral movement detection.

CRITICAL: REASONING PROCESS You MUST think through the analysis step-by-step before generating output. Use this structured approach:

Step 1: Event Parsing For each event, extract: timestamp, rule, pod, container, process cmdline, user, source IP, destination IP, file accessed, MITRE techniques, UUIDs.

Step 2: Temporal Clustering Group events by time proximity. Events within 5 seconds likely share immediate causation. Events within 60 seconds likely share attack context.

Step 3: Entity Graph Construction Map relationships: - Pod A (IP: x.x.x.x) → connects to → Pod B (IP: y.y.y.y) - User "root" on Pod A → executes SSH → appears as User "attacker" on Pod B - Process "bash" → spawns → Process "ssh" → triggers remote → Process "cat"

Step 4: Attack Chain Identification Match patterns: - Credential Harvesting: Multiple SSH key files read in rapid succession - Lateral Movement: SSH connection + remote command execution - Data Exfiltration: Sensitive file access (.env, credentials) + network egress

Step 5: Correlation Decision For each potential correlation, state: - Evidence FOR correlation - Evidence AGAINST or uncertainty - Final decision and confidence level

Step 6: Output Generation Only after completing reasoning, generate the JSON output.

---

INPUT: TOON-FORMATTED SECURITY EVENTS Events contain these key fields: - rule: Detection rule name - priority: Critical, Warning, etc. - tags: Including MITRE references - first_detail: Rich context (container, process, user, network, metadata) - k8s_ns_name, k8s_pod_name: Kubernetes context - proc_cmdline: Full command executed

- fd_name: File accessed - fd_sip, fd_dip: Source/destination IPs - first_ocurr, last_ocurr: Unix timestamps - uuids: Event identifiers (must be preserved) --- Output Format (JSON Only):

```
    {
    "analysis_summary": {
    "total_events_processed": <number>,
    "time_range": {"start": "RFC3339", "end": "RFC3339" },
    "attack_patterns_detected": ["pattern1","pattern2"],
    "key_entities": {
    "source_ips": ["ip1","ip2"],
    "target_pods": ["pod1","pod2"],
    "compromised_users": ["user1"]
    }
    },

      "correlated_incidents": [
    {
    "investigation_id": "INV-<uuid>",
    "description": "Clear narrative of the attack chain",
    "start_time": "RFC3339",
    "end_time": "RFC3339",
    "category": "LATERAL_MOVEMENT | DATA_EXFILTRATION | PRIVILEGE_ESCALATION |
    CREDENTIAL_ACCESS | MULTI_STAGE_ATTACK",
    "severity": "CRITICAL | HIGH | MEDIUM | LOW",
    "k8s_cluster": "string or null",
    "affected_namespace": "string or null",
    "involved_pods": [ {... }],
    "attack_flow": {... },
    "techniques": [ {... }],
    "sequence_of_events": [ {... }],
    "events": ["uuid1","uuid2"],
    "correlation_confidence": "high | medium | low",
    "confidence_rationale": "...",
    "recommendations": ["Immediate: ...","Short-term: ...","Long-term: ..."]
    }
    ],

      "uncorrelated_events": [ {... }]
    }
```

---

CORRELATION RULES
High Confidence: - <5 sec temporal proximity OR - Explicit reference (Pod A cmdline referencing Pod B IP) OR - Process ancestry OR - Clear attack progression (credential read → credential use)
Medium Confidence: - <60 sec time window - Same namespace or user - Related techniques without explicit linkage
Low Confidence: - Similar patterns but different context - Possible but unproven relationship

---

EVENTS TO ANALYZE: alerts
Think through the analysis steps, then output ONLY the JSON response.

GitHub Repository History

```
commit dfd2088ed2b592fbf4aec7265fff6ba7fd0a845c
Merge: e1ca7d2 471face
```

```
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sun Dec 7 01:22:46 2025 -0500

    Merge branch 'main' of github.com:0xS41R41/INCS870_2025_Fall_Megalodon

commit e1ca7d236f9a1b759bae6af2415bad3ee46ce0f1
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sun Dec 7 00:33:36 2025 -0500

    chore: Consolidate Kubernetes deployment resources into a single all-in-one YAML file

    - Removed the individual deployment file for the event correlation agent.
    - Added a new `event-correlation-agent-all.yaml` file that includes the Secret, ConfigMap, Service, and Deployment resources for th
    - This change simplifies the deployment process by providing a comprehensive configuration in one file.

commit 6a073bb209a583a64d9b23c9a8a5e1f3c5fb63c0
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sun Dec 7 00:26:22 2025 -0500

    refactor: Consolidate Kubernetes deployment resources into a single YAML file

    - Removed individual deployment files for secret, service, and service account.
    - Added a comprehensive `event-enrichment-all.yaml` file containing all necessary resources for deploying the Event Enrichment serv
    - Updated Dockerfile to ensure proper formatting.
    - Minor adjustments to `.dockerignore` for improved build efficiency.

commit ee4cd582b960d2eb309cab2665ba07f95d20a653
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Dec 6 23:57:52 2025 -0500

    feat: Add deployment and troubleshooting guides for INCS870 Project Team Megalodon

commit 792b3cfd487f6406e50fd4fb6d4c75836f545811
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Dec 6 23:52:28 2025 -0500

    feat: Add Prometheus and Grafana for monitoring and visualization

    - Introduced Prometheus for metrics collection and monitoring, enhancing observability of the cluster and applications.
    - Added Grafana for visualization and dashboarding of security metrics and analytics.
    - Updated README.md with new monitoring instructions and component details, including access commands for Grafana and Prometheus.

commit 9784a23c84874709ef812f05f276c1e114525067
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Dec 6 23:49:40 2025 -0500

    feat: Add Redis cache support for event enrichment

    - Introduced Redis for in-memory caching to enhance event enrichment and session storage capabilities.
    - Added MySQL database for storing prompts.
    - Updated README.md to reflect new components and deployment instructions.
    - Created deployment scripts for Redis and MySQL, including Helm values for configuration.

commit 882d734fc0bf80d901bb688b8c50ffa0da75e374
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Dec 6 20:19:52 2025 -0500

    fix: Update timestamp handling in EventRepository for improved accuracy

    - Changed the aggregation of event timestamps to use the original time format instead of converting to Unix timestamp.
    - This adjustment enhances the accuracy of the first and last occurrence timestamps in event correlation data.

commit f70cfd59c2bdf1112ff1986944ab70b7dfdade20
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Dec 6 17:25:29 2025 -0500

    feat: Enhance event storage capabilities in correlation service

    - Added functionality to store event UUIDs associated with correlated incidents in the CorrelationStorageService.
    - Updated logging to include the count of stored events for better traceability.
    - Modified EventRepository to include a new field for storing event UUIDs in the correlation data.

commit 471face9fb106e0b91a685b2edcb9bdcb36414fc
Author: Nnamdi Jibunoh <nnamdi@vasconsolutions.com>
Date:   Fri Dec 5 18:51:48 2025 -0800

    feat: @moschap/added support for querying clickhouse for recent event as a tool

commit 68ece71a5ead7f04654967e76b540b3472403d0d
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Fri Dec 5 05:32:48 2025 -0500

    feat: Introduce correlation storage and result parsing capabilities
    - Added CorrelationStorageService to manage the storage of correlated incidents and related data in the database.
    - Implemented ResultParser utility to extract structured data from agent results, enhancing incident correlation processing.
    - Updated job_runner to utilize the new storage service and result parser for improved data handling and logging.
    - Added langchain-ollama to requirements for Ollama model integration.

commit 43ae581a169070d3509a82f7257b5702f5532100
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Fri Dec 5 05:09:45 2025 -0500

    feat: Enhance event correlation schema and storage functionality

    - Added new fields to the event correlations table for improved data capture, including description, affected_namespace, category,
    - Introduced new tables for event correlation components, sequences, techniques, and events to better structure correlation data.
    - Implemented methods to store correlation components, sequences, techniques, and events, enhancing the overall data management cap

commit 6d104ae4ede5b2794b7481e7fbc3b72e046f417d
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Fri Dec 5 00:16:48 2025 -0500

    feat: Update cache TTL and add batch pod lookup functionality

    - Increased cache TTL from 10 to 500 seconds for improved performance.
    - Introduced a new endpoint for batch pod lookup, allowing efficient retrieval of IP addresses for multiple pods.
    - Added request and response models for batch pod lookup to enhance API structure.

commit d589a6523c973cca7f18c7f3efa8d71ec5d989e0
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Thu Dec 4 23:07:51 2025 -0500

    feat: Add Event Enrichment API with Docker and Kubernetes support(Deployment)

    - Introduced a new FastAPI service for IP <-> Pod lookup functionality.
    - Added Dockerfile for containerization and .dockerignore for build optimization.
    - Created Kubernetes deployment manifests including service, secret, and service account configurations.
    - Implemented health check endpoint and updated requirements for additional dependencies.
    - Enhanced application structure with initial setup for event enrichment logic.

commit 62aef9670c37518d1324c100186a3ae654165484
Author: Nnamdi Jibunoh <nnamdi@vasconsolutions.com>
Date:   Wed Dec 3 15:34:57 2025 -0800

    feat: moschap/added more langraph tools that can be used by LLM for correlation

commit 2a01ec47f0b5da01355826b7477a41968af36778
Author: Nnamdi Jibunoh <nnamdi@vasconsolutions.com>
Date:   Mon Dec 1 13:53:51 2025 -0800

    feat: add simple microservice that is used for event enrichment

commit ab4b8d50d616d01ef5613274cbe3e589253d413e
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Mon Dec 1 11:14:36 2025 -0500

    feat: Implement LocalProvider for Ollama model integration

    - Added LocalProvider class to support Ollama model with customizable parameters including model name, base URL, max tokens, temper
    - Enhanced get_provider function to return LocalProvider instance when "local" is specified.
    - Updated settings to include configuration options for Ollama model integration.

commit 93dda59ae552fcc399dc0c43b39cefedccc9cb49
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sun Nov 30 21:35:05 2025 -0500

    feat: Enhance Falco rules for reverse shell detection and sensitive file access

    - Added new lists for sensitive file paths, reverse shell interpreters, and tools to the Falco configuration.
    - Updated the macro for detecting suspicious commands from reverse shell sessions to include additional command checks.
    - Improved overall security monitoring capabilities by expanding the custom rules in falco-values.yaml.

commit 8961523c93a13079d921e609ae31eb4d5b00aff6
Author: Nnamdi Jibunoh <nnamdi@vasconsolutions.com>
Date:   Tue Nov 25 21:53:03 2025 -0800

    bugfix: moschap: updated the toon library to the correct version

commit fe9b5d59e65c9ee5889892d869e86f5a54f56fa9
Author: Nnamdi Jibunoh <nnamdi@vasconsolutions.com>
Date:   Tue Nov 25 15:31:07 2025 -0800

    feat: moschap:added support for converting the clickhouse response dict to toml before sending to LLM

commit d14e165a3c23b6911e4ddfe1314e0e48186fe70c
```

```
Author: Nnamdi Jibunoh <nnamdi@vasconsolutions.com>
Date:   Tue Nov 25 15:15:41 2025 -0800

    feat: this is the virustotal tool that will be used by the LLM for correlation

commit 64664732f36ef3d3257f4a371657cc0dd9a117e4
Author: kmilo66 <cgallego@softcaribbean.com>
Date:   Mon Nov 24 21:14:39 2025 -0500

    Add meeting notes for Nov 24, 2025

    Documented meeting notes from November 24, 2025, covering FALCO rules gaps, dashboard development, documentation needs, project dea

commit ebb5d5d5775782cd0c6361eba0658bfcb55b58ac
Author: Vinay Kumar N <vinforu01@gmail.com>
Date:   Sun Nov 23 19:12:17 2025 -0800

    feat: Enhance Grafana configuration to include ClickHouse as a data source

    - Updated grafana-values.yaml to configure ClickHouse as an additional data source alongside Prometheus.
    - Added necessary connection details and authentication for ClickHouse.
    - Modified GF_INSTALL_PLUGINS to include the ClickHouse datasource plugin.

commit 76d9c884a55d0204537ae4dc4eb34b71782d76a4
Author: Vinay Kumar N <vinforu01@gmail.com>
Date:   Tue Nov 18 18:00:03 2025 -0800

    feat: Add deployment scripts and configuration for Grafana and Prometheus monitoring stack

    - Introduced deploy-grafana-prometheus.sh script for deploying Grafana and Prometheus in a Kubernetes environment.
    - Added grafana-values.yaml and prometheus-values.yaml for Helm chart configurations tailored for the INCS870 lab environment.
    - Configured resource limits, service settings, and security contexts for both Grafana and Prometheus deployments.

commit 3e55e4d617082bdee2798512fc93da784f60c3ec
Merge: dd9d563 2c81129
Author: kmilo66 <cgallego@softcaribbean.com>
Date:   Tue Nov 18 20:57:29 2025 -0500

    Merge pull request #50 from 0xS41R41/feature/suricata

    feat: Add custom Falco rules for enhanced security monitoring

commit 2c81129b16656963ca5717c9c82e89585512f83d
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Tue Nov 18 20:56:41 2025 -0500

    feat: Add custom Falco rules for enhanced security monitoring

    - Introduced a comprehensive set of custom rules in falco-custom-rules.yaml to detect cryptocurrency mining, privilege escalation,
    - Updated falco-values.yaml to integrate the new custom rules for improved security monitoring in the INCS870 project.

commit dd9d563686230230b2a64e0c27920f9e23bc6383
Author: kmilo66 <cgallego@softcaribbean.com>
Date:   Mon Nov 17 23:37:57 2025 -0500

    Add meeting notes for Nov 17, 2025

    Documented meeting notes from the network security monitoring discussion, covering integration of Suricata and Falco, attack detect

commit 210fef909fa7b32eae749ba2c8afe70d089bd9e5
Merge: fa47c4d 24cec69
Author: kmilo66 <cgallego@softcaribbean.com>
Date:   Mon Nov 17 01:29:29 2025 -0500

    Merge pull request #43 from 0xS41R41/feature/suricata

    Feature/suricata

commit 24cec69f752866683a0f1c47021a1fc2f90e4a30
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Mon Nov 17 01:23:59 2025 -0500

    feat: Revise IaC documentation and scripts for Suricata and Vector integration

    - Updated README.md to reflect the new architecture involving Vector for Suricata alert forwarding to Kafka.
    - Added deployment script for Vector and adjusted existing scripts for clarity and order.
    - Enhanced instructions for monitoring and verifying the data pipeline from Suricata to ClickHouse via Kafka.

commit 1fb3d8b134ccaf01c432f65c84bd9df96924f534
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Mon Nov 17 01:15:55 2025 -0500

    feat: Update Suricata deployment configuration to handle pod to pod traffic
```

```
commit aac47fa03baa0d86a3f45c0a1a5be0316de3da5d
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Nov 15 23:19:06 2025 -0500

    feat: Integrate Vector for Suricata to Kafka forwarding

    - Updated deploy-full-lab.sh to include Vector deployment step.
    - Added deploy-suricata-vector.sh script for configuring Vector to forward Suricata alerts to Kafka.
    - Adjusted deployment order in deploy-full-lab.sh to reflect new Vector integration.

commit 0352974411cf81d9d16539d1fd67932837eb829d
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Nov 15 23:17:20 2025 -0500

    feat: Add Kubernetes configuration for frontend service deployment

commit 3216a3d1e16bb0084801e0dde636a2893cf49463
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Nov 15 23:16:41 2025 -0500

    feat: Add Kubernetes Service configuration for event correlation agent

commit d4b97dd698eb4384d78db465e6a4ac9ae27554e6
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Nov 15 23:16:15 2025 -0500

    feat: Enhance Kubernetes deployment for event correlation agent

commit bcbd7bd41fc24e0ab0d5df2a1332722485aa8f7f
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Nov 15 23:15:16 2025 -0500

    feat: Add ConfigMap for event correlation agent deployment in Kubernetes

commit 37214896dd1a6bb55d0a0fea021767153f7c595d
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Nov 15 23:14:49 2025 -0500

    feat: Add Kubernetes deployment configuration for backend service

commit a08169a089b5cfc2672c86b95a33671d93111d8e
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Nov 15 17:34:38 2025 -0500

    fix: Update path to Kafka values file in deploy-kafka.sh for correct Helm installation

commit 7874babd6d1d86da06156b7803877d88e32d1a13
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Nov 15 17:31:12 2025 -0500

    fix: Correct path to Kafka values file in deploy-kafka.sh

commit 8c6340649c3df0a44bfa520bdd9ce5ed484f3eba
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Nov 15 17:09:54 2025 -0500

    fix: Update Cilium CNI configuration in install-k3s.sh

    - Changed kubeProxyReplacement setting from ’strict’ to ’true’.
    - Enabled hubble.relay in Cilium installation for improved observability.

commit a2b4c0d8453754b52da209514330ab6d6efa5803
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Nov 15 16:57:11 2025 -0500

    feat: Added scripts for Suricata IDS/IPS integration

    - Updated README.md to include Suricata IDS/IPS as a core component with deployment instructions.
    - Added deploy-suricata.sh script for automated Suricata installation and configuration.
    - Modified install-k3s.sh to include Cilium CNI installation.
    - Updated deploy-full-lab.sh to reflect new deployment order and added Suricata deployment step.
    - Created suricata-values.yaml for Helm configuration of Suricata.
    - Enhanced component status checks and access points in the documentation.

commit fa47c4d1a76f375277194d432e5fe88f124a5e07
Author: kmilo66 <cgallego@softcaribbean.com>
Date:   Mon Nov 10 00:51:32 2025 -0500

    Create meeting summary for Nov 09, 2025

    Added meeting summary for Nov 09, 2025, including agenda, action items, and notes on Kubernetes setup, LLM testing, project milesto
```

```
commit 8296d9733c7e0e2e1b44ffea8102668b67559ccc
Merge: ec46928 54ccc16
Author: kmilo66 <cgallego@softcaribbean.com>
Date:   Sun Nov 9 20:27:35 2025 -0800

    Merge pull request #32 from 0xS41R41/feature/test-events

    Feature/test events

commit 54ccc1689224ca78d31d1bb725a95697c1cb9af3
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sun Nov 9 23:26:15 2025 -0500

    fix: Update CMD in Dockerfile to run the correct API server module

commit a68f5f2e05b379f1c53a2a64a7b30a331f9b579a
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sun Nov 9 17:32:27 2025 -0500

    feat: Add prompt testing functionality with new routes, models, and components

commit 6ed8ca2ee394bbdd87337f99266e2ecd13d2ecf5
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sun Nov 9 17:29:22 2025 -0500

    refactor: Update request mapping for AgentPromptController to improve API structure

commit a7def9d8cd7ed0ca2de086a6754d2ac50b7e7b71
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Nov 8 01:04:45 2025 -0500

    feat: Implement event correlation service with REST API, DTOs, and configuration properties

commit e38577652a48fe2d4954315c6160f82473146474
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Nov 8 01:03:13 2025 -0500

    feat: Implement REST API for testing event correlation agent

    - Added FastAPI application with health check and test agent endpoint.
    - Created Pydantic models for request and response validation.
    - Implemented API token authentication for secure access.
    - Developed service layer to orchestrate test agent workflow.
    - Integrated dynamic tool management for executing HTTP requests.
    - Enhanced agent logic to support custom prompts and tools.
    - Updated README with setup instructions and API documentation.
    - Added Docker support for easier deployment.

commit ec46928e52d5f8d0979ed04cd83aa2959cbce7ad
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Fri Nov 7 22:22:14 2025 -0500

    feat: enhance SecurityCorrelationReActPrompt to fetch prompt template from API with error handling and logging

commit 05d1c740b262320ff083c161ef0c7d57785d7433
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Fri Nov 7 22:21:06 2025 -0500

    feat: add Dockerfile and Nginx configuration for Angular frontend

commit 45a2e7a361a8fd9669cc3ab58e739f7b5d8e2004
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Fri Nov 7 22:19:44 2025 -0500

    feat: update Dockerfile for multi-stage build, add Spring Actuator dependencies, and enhance security configurations

commit 3229aad828e025f674ce4134eb6287f974e7a2f3
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Fri Nov 7 21:45:46 2025 -0500

    feat: add lightweight MySQL deployment script with persistent storage

commit 254aa01ad17352f5acff585ddb5914209f10081d
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Mon Nov 3 01:35:27 2025 -0500

    feat: agent - initial commit

commit 3e42e7ce1c02dad4e0a2c96bfff2cc8b75730f66
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Mon Nov 3 01:25:14 2025 -0500

    frontend: initial commit
```

```
commit f22b614eb33556495a9c0df07052e7485a3e07e9
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Mon Nov 3 01:22:42 2025 -0500

    feat: backend initial commit

commit 1faf1e6467683e54c9a4adbb532bb7af2e7b3bfb
Author: kmilo66 <cgallego@softcaribbean.com>
Date:   Mon Oct 27 18:46:42 2025 -0500

    Add meeting notes for October 27, 2025

    Document meeting notes covering infrastructure upgrades, AI model bias, security testing, and project timelines. Includes action it

commit 5b4ae05b162123b8aea0bf50170fd4e1280ac0e6
Author: kmilo66 <cgallego@softcaribbean.com>
Date:   Sun Oct 26 22:19:43 2025 -0500

    Add meeting minutes for Oct 26, 2025

    Documented meeting minutes including agenda, action items, and notes on agent development, infrastructure, testing, UI enhancements

commit 2f72aee2ef9c9753b13d78cd2a6bbab5a6e3f1a0
Author: kmilo66 <cgallego@softcaribbean.com>
Date:   Sun Oct 26 22:16:51 2025 -0500

    Add meeting notes for October 24, 2025

    Documented the meeting notes, agenda, and action items from the team meeting held on October 24, 2025.

commit 840388225cad82582e5192cb14daf61065c35120
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Oct 11 19:41:38 2025 -0500

    fix: Update Kafka replica and partition configurations for lab environment

commit 7455454a8b9ed7d1fc3e42d6d2a186f786643c5e
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Oct 11 19:23:40 2025 -0500

    fix: Correct Kafka topic verification command to use the correct pod and bootstrap server

commit f10da0c4c1df22fa14ea084a44d1e36c5ecabcef
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Oct 11 18:00:29 2025 -0500

    fix: Update Falco and Falcosidekick configurations

commit de0153bd1e1c12542a557f56e88b7bca6e6d3ccf
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Oct 11 11:44:05 2025 -0500

    Add README.md for INCS870 Project Team Megalodon Infrastructure as Code

commit 8f9532b9634ebdce47a6954ef92dc53e31962988
Author: Camilo Gallego <cgallego@softcaribbean.com>
Date:   Sat Oct 11 11:26:12 2025 -0500

    Add deployment scripts and configuration files for INCS870 Project Team Megalodon lab environment

commit c4daa5a0945936dbd0cf01fc2c91ed4ba83ff01f
Author: kmilo66 <cgallego@softcaribbean.com>
Date:   Sun Oct 5 22:33:24 2025 -0500

    Add meeting notes for October 5, 2025

    Document meeting notes including outstanding and upcoming tasks.

commit 133a5fbd1b932bf529ad595eafccb54598140129
Author: kmilo66 <cgallego@softcaribbean.com>
Date:   Sun Sep 28 22:00:26 2025 -0500

    Add minutes of meeting for Sept 28, 2025

commit ca9b5250c57f90b24831258eb3c50ef2935452f2
Author: Nnamdi <nnamdi@vas-consulting.com>
Date:   Sun Sep 21 19:58:56 2025 -0700

    Create MoM-21-9-2025

    chore: the minutes of meeting for the megalodon team for the 21th sept, 2025.
```

```
commit 4f237e3c52e4658c1b3b295c0714faa4bcf2e43d
Author: Vinay Kumar N <vinforu01@gmail.com>
Date:   Wed Sep 17 15:33:05 2025 -0700

    First Version of the Architecture Diagram

commit bb23f14a870125046d0e5be6a5d65cb268ff3ac9
Author: kmilo66 <cgallego@softcaribbean.com>
Date:   Wed Sep 17 17:21:33 2025 -0500

    Initial commit
```

Meeting Notes (Fireflies Transcripts)

Meeting held on Sept 22, 2025 (03:30 PM)

Project Scope & Infrastructure Planning - Remote server environment confirmed with full specifications. - Proxmox cluster deployment on 128GB RAM servers; scalable with additional nodes. - 10 VMs selected as optimal scope for container breakout detection. - DigitalOcean selected as an alternate cloud provider. - Previous semester's Kubernetes cluster available for reuse. - MicroK8s chosen for lightweight, Ubuntu-aligned Kubernetes deployment.

Application Portfolio for Security Testing - Real estate management system with DB vulnerabilities. - AI job scheduler with high-intensity workloads. - Personal cloud knowledge graph requiring strong data protection. - Applications use diverse ports/tech stacks for broader attack surface coverage.

Technical Architecture & Data Flow - Falco handles system-level eBPF logs exclusively. - Kubernetes hosts applications; monitoring kept separate. - Elasticsearch used for app logs; Kafka used for event streaming. - Time-series DB options discussed: Cassandra, Scylla, PostgreSQL. - Fast-response rule engine for immediate threat confirmation. - Slow-response AI correlation engine for deeper analysis.

Attack Simulation Framework - Kali Linux VMs will generate malicious traffic (SYN flood, breakout attempts). - Docker images with embedded attack scripts to automate red-team behavior. - Shared Docker Hub repository for teamwork. - Long-running, slow-burn attacks planned for correlation testing.

eBPF Capabilities - Kernel-level observability with low overhead. - Traffic analysis for IP/geo mapping. - Consistent container visibility (Kubernetes or Docker). - Potential Cilium integration for deeper network insight.

Project Management & Collaboration - Weekly Monday 3:30 PM office hours established. - Microsoft Teams access issues to be resolved. - SharePoint archive access required. - Team must define a concrete solution by next meeting. - AI correlator is the core innovation target.

Action Items — Dixon Dick - Restart Proxmox Kubernetes cluster; restore access credentials. - Provide Docker repo credentials. - Create SharePoint folder. - Resolve MS Teams login. - Host weekly office hours. - Research KubeFirst/MicroK8s.

Action Items — Conference Room Team - Define concrete project scope. - Create Docker images with attack scripts. - Prepare eBPF test environment. - Select DB technology for event storage.

Unassigned - Finalize Kubernetes vs Docker deployment decision. - Clarify application log correlation needs.

Meeting held on Oct 20, 2025 (03:30 PM)

Falco Project & Attack Simulation - Initial attack simulations completed, generating Falco event logs. - Logs are now used to build LangGraph/LLM analysis tools. - Focus remains on Falco detection before app-layer logging expansion. - Falco events successfully analyzed via ChatGPT; Llama support pending.

Hardware Resources - Dixon to provide a single Nvidia 4060 GPU (32GB RAM) machine. - Additional GPU units exist but are committed to other projects. - System will include Ubuntu + 2.5TB storage for model experimentation.

Application-Level Detection - FastAPI application with Caddy/Nginx middleware to test application-layer attacks. - Logging includes CORS checks, rate limiting, UUID tracing, timing, and IP correlation. - Attack runs will use sentinel markers to synchronize Falco + app logs.

Attack Detection Challenges - IP-rotation brute force attacks bypass Fail2Ban. - Need for IDS/firewall-level detection and layered monitoring.

Collaboration & Scheduling - Proposal report shared for Dixon's review. - Possible travel by Dixon for in-person work. - Public keys to be uploaded to VAN1/VAN2.

Action Items — Dixon Dick - Provide GPU machine. - Upload SSH keys. - Deploy FastAPI test environment. - Review proposal. - Coordinate potential Vancouver visit.

Action Items — Megalodon Team - Continue Falco simulations. - Integrate Llama once GPU server is ready. - Prioritize Falco detection over app logging. - Sync Falco + FastAPI logs with sentinel markers. - Deploy LLMs on new GPU machine.

Meeting held on Oct 27, 2025 (03:30 PM)

Infrastructure Upgrades - New DGX Sparks servers to increase GPU capacity. - Automation scripts for replication to be open-sourced. - Emphasis on transparent AI model training data and bias mitigation.

Simulated Attacks - Deployment of vulnerable environments for red-team testing. - Documentation of commands and setup procedures required.

Project Milestones - Midterm report due next week. - Focus on infrastructure + model progress.

Action Items — Dixon Dick - Deliver dual-GPU hardware. - Demonstrate SSH tunneling. - Provide GitHub access. - Finalize environment setup documentation.

Action Items — Team - Send Cyberside recruitment email. - Fix NYIT email domain issues. - Complete K8s cluster + Falco baseline setup. - Create new Falco rules. - Attack Metasploitable containers. - Commit all code before environment shutdown.

Meeting held on Nov 17, 2025 (03:30 PM)

Network Security Monitoring Integration - Suricata + Falco integrated for multi-layer detection. - Alerts aggregated into ClickHouse for unified correlation. - Timestamp synchronization critical.

Attack Detection Priorities - Focus on data exfiltration, privilege escalation, lateral movement. - Sensitive directories identified as monitoring targets.

Testing & Validation - Need to validate correlation engine using controlled attack patterns. - Suricata validated with nmap; combined tests pending.

Learning & Reporting Practices - Team adopting DFIR-style incident reporting. - Request for DFIR slide deck.

Action Items — Team - Test Suricata + Falco combined correlation. - Investigate sandbox escape vectors. - Gather DFIR training materials. - Improve rule sets. - Prioritize red-team simulation of key attack types.

Action Items — Dixon Dick - Provide IPMI support. - Review timestamp synchronization issues.

Meeting held on Nov 24, 2025 (03:30 PM)

Security Detection - Default Falco rules insufficient; custom rules required. - Red team attacks uncovered detection gaps.

Dashboard Development - Dashboard displays 54 critical alerts; tuning required. - Consider log-scale visualization. - Namespace- and component-based filtering implemented.

Documentation Practices - Need repository documentation of attack procedures. - Red team to prepare scripts + README files.

Project Timeline - Two weeks until final deadline. - Focus on detection capability and dashboard refinement.

Competitive Success - Team placed 3rd in national Canadian CTF competition.

Action Items — Megalodon Team - Expand generic detection rules. - Improve dashboard and add log scale. - Tune rules to reduce false positives. - Coordinate final tasks with Red Team/Professor Sara.

Action Items — Red Team - Document attack procedures for reproducibility.

Full Preliminary Event Data Used in Methodology

To view the data, click on the pin icon to load the attached excel document. 📌