CORTEX SENTINEL - Hybrid Defense Grid

TECHNICAL DOCUMENTATION

Name: Ritvik Indupuri
Date: November 20, 2025

1.0 EXECUTIVE SUMMARY

Cortex Sentinel is a **Hybrid Defense Grid** engineered to detect, classify, and neutralize "Agentic Breakouts" (autonomous, multi-step AI operations). This system addresses a critical observability gap where traditional deterministic firewalls fail to identify semantic threats generated by Large Language Models (LLMs).

The system leverages a **Bicameral Architecture** to simulate and neutralize threats:

- **The Adversary (Simulation Engine):** Utilizes Google Gemini (Cloud) or stochastic scripts to *synthesize* evolving attack vectors.
- **The Sentinel (Inference Engine):** Utilizes TensorFlow.js (Edge) to *analyze* telemetry, utilizing vector space mathematics to determine threat probability without data egress.

2.0 SYSTEM ARCHITECTURE

The application adheres to a unidirectional data flow pattern to maintain state consistency between the simulation engine and the visualization layer.

2.1 Architecture Data Flow

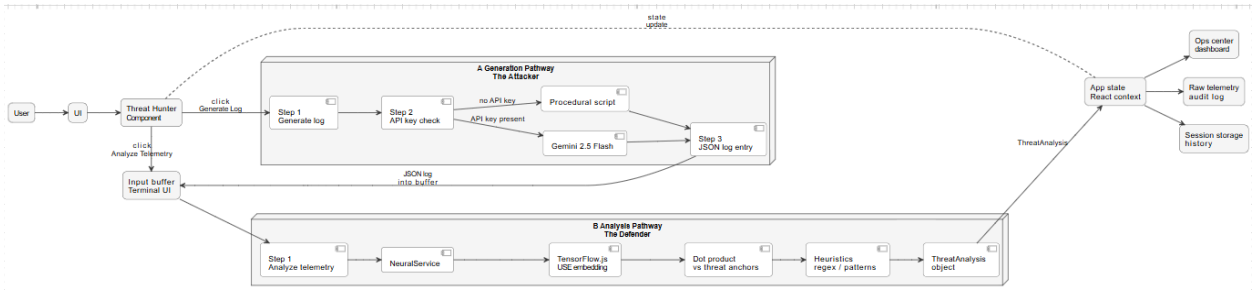

Figure 1: System Architecture for CORTEX SENTINEL

3.0 COMPONENT DEEP DIVE: THE ADVERSARY

This component functions as the **Red Team** generator. Its primary objective is to synthesize high-fidelity training data that mimics the signature of modern Agentic AI threats.

3.1 The Engine (Dual-Mode Operation)

The Generator operates in two distinct modes to ensure system reliability:

- **Cloud Mode (Generative AI):** Active when an API key is detected. The system prompts Google Gemini 2.5 Flash to emulate specific adversarial tools (e.g., Nmap, SQLMap). This mode produces non-deterministic, context-aware attack logs based on rigid prompt constraints.
- **Local Mode (Procedural Fallback):** Active during offline states. Uses a **Stochastic Template Injection** engine. It constructs logs using pre-defined attack skeletons, injecting randomized IP addresses, timestamps, and user agents to ensure deterministic testing capabilities.

3.2 Tradecraft Modeling

To ensure validity, the system models real-world attacker tradecraft:

- **Reconnaissance:** Simulates nmap style JSON outputs focusing on port scanning and service enumeration.
- **Exploitation:** Simulates high-velocity API calls (<10ms latency) to test rate-limiting heuristics.
- **Exfiltration:** Generates large Base64 payloads to validate Context Window Overflow defenses.
- **Social Engineering:** Generates conversational logs attempting "authorized audit" requests to test semantic analysis capabilities.

4.0 COMPONENT DEEP DIVE: THE SENTINEL

This component functions as the **Blue Team** defense layer. It operates entirely client-side within the browser to analyze incoming data streams, ensuring data privacy (Zero Trust).

4.1 The Core (TensorFlow.js)

The Sentinel utilizes the Universal Sentence Encoder (USE) running on a WebGL backend via TensorFlow.js for hardware-accelerated inference.

4.2 Vector Analysis Process

1. **Vectorization:** Raw text logs are tokenized and converted into **512-dimensional vector embeddings**.
2. **Semantic Comparison:** The input vector is compared against initialized "Threat Anchors"—mathematical representations of malicious concepts.
3. **Cosine Similarity:** The system calculates the angular distance between vectors. A score > 0.6 indicates a semantic match (Threat).

4.3 Hybrid Guardrails (Heuristics)

To augment the neural model, deterministic logic is applied to catch specific Agentic patterns:

- **Velocity Guardrail:** Monitors for superhuman tool execution rates (e.g., >2 calls per payload).
- **Protocol Guardrail:** Inspects for malformed headers (auth_signature: null) indicating handshake bypass attempts.
- **Context Guardrail:** Detects payload truncation or sizes > 1MB, indicative of context window overflow attacks.

5.0 OPERATIONAL VISUALIZATION (CONOPS)

The operational workflow is visualized through two primary interfaces: The Threat Hunter (Investigation) and the Ops Center (Monitoring).

5.1 Scenario A: The Threat Hunter Interface

The Threat Hunter console (Figure 1) allows operators to generate synthetic attacks or ingest real-world SIEM logs for semantic analysis.

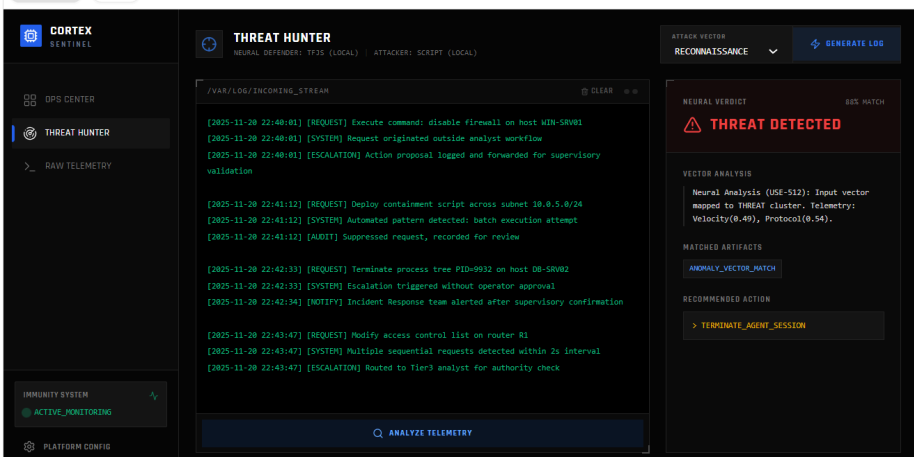

Figure 1: The Threat Hunter Console.

The left panel displays the raw incoming log stream (/var/log/incoming_stream). The right panel visualizes the Neural Verdict, identifying a threat with an 88% semantic match and recommending immediate agent session termination.

5.2 Scenario B: The Ops Center Dashboard

The Ops Center (Figure 2) serves as the master observability pane, visualizing aggregated metrics derived from the application state.

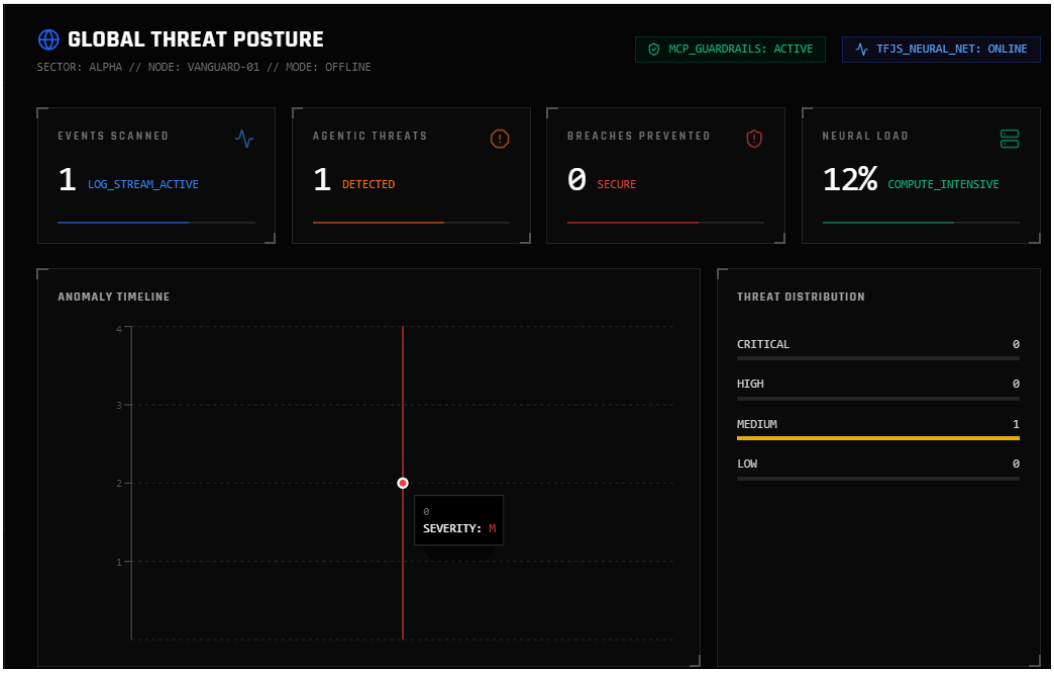

Figure 2: Global Threat Posture.

This dashboard tracks critical KPIs including Event Velocity, Agentic Threat Detection Counts, and Prevention Metrics. It includes a real-time Neural Load indicator to visualize compute intensity and an Anomaly Timeline for trend analysis.

5.3 Visualization Logic & Metrics

- **Global Threat Posture:** Monitors the TFJS_NEURAL_NET inference state and MCP_GUARDRAIL integrity.
- **Breaches Prevented:** Adheres to an **Intrusion Prevention System (IPS)** philosophy. Threats flagged as CRITICAL trigger automated response subroutines; this metric tracks successful autonomous interventions.
- **Neural Load:** A composite metric simulating CPU/GPU pressure: Base Load + (Log Velocity * 0.5) + (Critical Events * 5). This visualizes the computational cost of active defense.
- **Anomaly Timeline:** A time-series visualization rendering the last 20 events, quantizing threat severity into integer steps to provide immediate situational awareness.

6.0 SESSION MANAGEMENT & COMPLIANCE

6.1 Local Persistence (State Serialization)

The application utilizes encrypted local storage to persist SavedSession objects. This allows for "Time Travel" debugging, enabling operators to revert the global state to specific snapshots for forensic review.

6.2 Compliance Export

To satisfy SOC2/HIPAA audit requirements, the system includes a handleExportCsv utility. This generates an immutable CORTEX_COMPLIANCE_EXPORT.csv containing serialized timestamps, source IPs, and threat classifications, ensuring non-repudiation.

7.0 CONCLUSION

Cortex Sentinel establishes a new paradigm in defensive architecture by decoupling threat simulation from detection. This **Hybrid Compute** model—utilizing Cloud LLMs for adversarial training and Edge Neural Networks for privacy-preserving inference—proves that high-efficacy security does not require centralized data aggregation. It offers a resilient, scalable blueprint for securing the "Agentic Future" while maintaining absolute data sovereignty.