INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-14 13:49 UTC

Prompt Injection Reaches Production: Runtime Detection at the LLM Layer Arrives for Kubernetes Workloads

SECURITY ANALYSIS | MEDIUM | CVSS 5.0

SCC Item ID	SCC-STY-2026-0129
Type	Security Analysis
Severity	MEDIUM
CVSS Base Score	5.0
Affected Products	CrowdStrike Falcon AIDR, Falcon Cloud Security, Falcon Container Sensor, Falcon Next-Gen SIEM, OpenAI-compatible LLM API clients running on Kubernetes
Discovery Source	Rss:T1 Threatintel

Executive Summary

CrowdStrike has released a new Falcon Container Sensor collector that brings runtime detection of prompt injection, data leakage, and AI policy violations directly to Kubernetes-hosted LLM workloads, without requiring proxy insertion or architectural changes. This closes a visibility gap that conventional endpoint and network security tools cannot address, because LLM-layer attacks occur at the application logic level, not the network or process layer. The release is a market signal: AI workload runtime security is solidifying into a dedicated product category, and organizations deploying LLMs in production without dedicated runtime monitoring now carry a measurable and increasingly indefensible gap.

Technical Analysis

Prompt injection, classified by OWASP as LLM01 in the OWASP Top 10 for LLM Applications, represents the highest-priority attack surface for production AI systems. The attack class exploits the fundamental design of LLMs: model inputs are instructions, not data, so a malicious instruction embedded in user input, a retrieved document, or an API response can override system-level directives, exfiltrate context, or cause the model to take unintended actions. Until now, no commercially available runtime sensor addressed this layer directly.

CrowdStrike's Falcon AIDR extension addresses that gap by deploying a Falcon Container Sensor collector that inspects OpenAI-compatible API traffic at runtime inside Kubernetes pods. The approach does not require a separate agent process, proxy insertion, traffic redirection, or changes to existing AI application deployments. Detection coverage spans three categories: prompt injection attempts (CWE-77, CWE-78, mapped to MITRE T1059 and T1190), data leakage events where model outputs may contain sensitive context (CWE-200,

mapped to T1552 and T1071.001), and AI policy violations, a broader category covering outputs that violate organizational acceptable-use or content policies (CWE-693, mapped to T1565).

The defensive gap this addresses is structural. Kubernetes network policies and conventional container security tools operate at layers 3 through 7 of the network stack or at the OS process level. LLM-layer attacks do not generate anomalous network signatures or abnormal syscalls; they are semantically malicious within otherwise valid HTTPS requests to legitimate API endpoints. SIEM correlation rules and EDR behavioral detections cannot differentiate a benign LLM API call from a prompt injection payload without understanding the content of the call at the application layer.

The product release responds to documented threat maturation. Prompt injection has transitioned from academic proof-of-concept to documented active exploitation in production AI systems, with publicly reported incidents in 2024-2025 across SaaS AI integrations, autonomous agent frameworks, and retrieval-augmented generation pipelines (per OWASP Top 10 for LLM Applications threat surveys). CrowdStrike's integration of AIDR detections into Falcon Next-Gen SIEM allows organizations to correlate LLM-layer events with broader telemetry, which matters for incident response: a prompt injection that exfiltrates API keys (T1552) or initiates command execution (T1059) will now generate a correlated alert chain rather than appearing as an isolated anomaly in application logs that no one is watching.

Action Checklist

1. Step 1: Assess exposure, inventory all Kubernetes workloads running LLM inference or proxying calls to OpenAI-compatible APIs; determine whether Falcon Container Sensor is deployed and whether AIDR is licensed and configured for those workloads.
2. Step 2: Review controls, verify whether your current detection stack has any coverage for LLM-layer prompt injection; check if application-layer logging captures full request and response payloads from LLM API calls, not just metadata.
3. Step 3: Update threat model, add OWASP LLM01 (Prompt Injection) as an explicit entry in your threat register for any AI-integrated application; map to MITRE T1059, T1190, T1552, and T1071.001 for detection rule development.
4. Step 4: Communicate findings, brief application and platform engineering teams that Kubernetes-hosted LLM workloads now have a supported runtime detection path; coordinate with the AI/ML platform owner on sensor deployment timelines.
5. Step 5: Monitor developments, track CrowdStrike AIDR release notes and OWASP LLM Top 10 updates; watch for follow-on coverage of indirect prompt injection (LLM01 sub-variant via retrieval pipelines), which is the more complex and currently less-detected attack path.

IR / Forensic Enrichment

Triage Priority	STANDARD
Escalation Criteria	Escalate to urgent if Falcon AIDR generates a prompt injection alert on a Kubernetes workload processing PII, PHI, or financial data (triggering breach notification assessment under GDPR, HIPAA, or PCI DSS), or if LLM API response logs contain evidence of credential extraction (AWS key patterns, OAuth tokens, or internal system prompt disclosure) indicating a successful data leakage event rather than a probe.

Recovery Notes	Post-incident, verify that Falcon Container Sensor daemonset coverage is confirmed across all LLM-hosting namespaces and that AIDR policy is enforced (not just monitoring) for prompt injection and data leakage categories. Monitor LLM API response logs for a minimum of 14 days post-remediation for recurrence of injection patterns, paying particular attention to indirect injection via any RAG retrieval pipelines that were not in scope for initial sensor coverage. Update the threat register entry for OWASP LLM01 with indicators observed during the incident to improve Sigma rule and AIDR policy precision before returning to steady-state monitoring.
Forensic Artifacts	LLM API application-layer request/response logs from the Kubernetes pod (kubectl logs or persistent log sink): the primary artifact for prompt injection — must contain full prompt text, system prompt, assistant response, model ID, and caller identity to reconstruct the injection sequence | Envoy/Istio sidecar access logs or API gateway access logs capturing HTTP POST bodies to the OpenAI-compatible inference endpoint: these reveal the raw injection payload, including any jailbreak prefixes or instruction-override strings sent at the application layer | Kubernetes audit log entries (kube-apiserver audit.log) for exec, port-forward, or pod/log API calls against the LLM inference pod: lateral movement following a successful injection may involve an attacker using kubectl to access the compromised pod directly | Vector database or document store query logs from the RAG retrieval pipeline (if applicable): for indirect prompt injection, the injected content originates in a retrieved document — the retrieval query log identifies which external document chunk introduced the malicious instruction into the LLM context | Falcon AIDR alert telemetry from the Falcon Next-Gen SIEM for the affected pod (if AIDR is deployed): provides ground-truth detection events with model-layer context that no other log source captures, including the specific policy rule triggered and the token-level content that caused the violation

Per-Action IR Details

Step 1: Assess exposure — inventory all Kubernetes workloads running LLM inference or proxying calls to OpenAI-compatible APIs; determine whether Falcon Container Sensor is deployed and whether AIDR is licensed and configured for those workloads

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: establishing visibility and asset inventory before an incident occurs

Controls: NIST IR-4 (Incident Handling) — ensuring IR capability covers AI/LLM workload scope, NIST SI-4 (System Monitoring) — monitoring coverage must extend to Kubernetes-hosted LLM inference pods, NIST RA-2 (Security Categorization) — categorizing AI workloads by data sensitivity and LLM API exposure, CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory) — LLM inference pods and OpenAI-compatible API proxy services must be enumerated as enterprise assets, CIS 2.1 (Establish and Maintain a Software Inventory) — Falcon Container Sensor deployment status and AIDR license assignments must be tracked per workload

Compensating: For teams without Falcon AIDR: run `kubectl get pods -A -o json | jq '.items[] | select(.spec.containers[].env[]? | .value | test("OPENAI|LLM|ANTHROPIC|MISTRAL"; "i")) | {namespace: .metadata.namespace, pod: .metadata.name}'` to surface pods with LLM API environment variables. Cross-reference against your Helm release inventory (`helm list -A`) for any chart names containing 'llm', 'inference', 'ollama', 'triton', or 'vllm'. Document each namespace, image, and whether it proxies external OpenAI-compatible endpoints.

Evidence: Before inventorying, capture the current state: export `kubectl get pods -A -o yaml` and `kubectl describe daemonset falcon-sensor -n falcon-system` to establish Falcon sensor coverage gaps across namespaces. Record which pods have `hostPID: true` or privileged security contexts, as these affect sensor telemetry scope for LLM container workloads.

Step 2: Review controls — verify whether your current detection stack has any coverage for LLM-layer prompt injection; check if application-layer logging captures full request and response payloads from LLM API calls, not just metadata

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: identifying coverage gaps in monitoring that would allow LLM-layer attacks to go undetected

Controls: NIST SI-4 (System Monitoring) — detection must cover application-logic-layer LLM API interactions, not only network/process events, NIST AU-2 (Event Logging) — event types must include full LLM request/response payloads, not solely HTTP metadata or container stdout, NIST AU-3 (Content of Audit Records) — audit records for LLM API calls must establish what prompt was submitted, what response was returned, which model was invoked, and by which identity, NIST AU-12 (Audit Record Generation) — LLM inference pods must be configured to generate structured logs of API interactions at the application layer, CIS 8.2 (Collect Audit Logs) — audit log collection must be validated to include LLM API gateway or sidecar logs, not only Kubernetes control plane and node-level logs

Compensating: For teams without Falcon AIDR or a WAF with LLM payload inspection: deploy an Envoy sidecar as a logging proxy in front of the LLM inference container to capture full HTTP request/response bodies to a local log volume. Alternatively, patch the application pod to emit structured JSON logs (prompt_text, response_text, model_id, token_count, caller_identity) via stdout and ship with Fluentd/Fluentbit to a retained log sink. Write a Sigma rule targeting log entries where response_text contains patterns like 'ignore previous instructions', 'disregard your system prompt', or exfiltration-indicative phrases (e.g., base64 blobs in assistant turns).

Evidence: Pull existing application logs from LLM inference pods (`kubectl logs -n --timestamps --previous`) and confirm whether request bodies are present or truncated. Check Kubernetes audit logs (`/var/log/kubernetes/audit.log`) for API server events against the pod but note these will not contain LLM prompt content — this gap itself is the key finding to document. If an Istio or Envoy mesh is in use, retrieve access logs from the sidecar to determine whether payload capture is enabled.

Step 3: Update threat model — add OWASP LLM01 (Prompt Injection) as an explicit entry in your threat register for any AI-integrated application; map to MITRE T1059, T1190, T1552, and T1071.001 for detection rule development

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: threat modeling and detection rule development as pre-incident readiness activities

Controls: NIST RA-3 (Risk Assessment) — formal risk assessment must now account for OWASP LLM01 prompt injection as a threat source for AI-integrated Kubernetes applications, NIST SI-4 (System Monitoring) — detection rules must be written for MITRE T1059 (Command and Scripting Interpreter) where an LLM is manipulated to emit executable instructions, T1190 (Exploit Public-Facing Application) for externally accessible inference endpoints, T1552 (Unsecured Credentials) for prompt attacks targeting credential extraction from system context, and T1071.001 (Application Layer Protocol: Web Protocols) for LLM API exfiltration channels, NIST IR-4 (Incident Handling) — incident classification criteria must include LLM-layer prompt injection as a distinct incident type with its own severity tiers, CIS 7.1 (Establish and Maintain a Vulnerability Management Process) — the vulnerability management process must incorporate OWASP LLM Top 10 as a source of threat intelligence alongside CVE feeds

Compensating: Write Sigma rules targeting application-layer logs for prompt injection indicators: match on assistant-turn responses containing shell command syntax (`/bin/sh`, `cmd.exe`, `powershell`), credential patterns (AWS key format `AKIA[0-9A-Z]{16}`), or LLM refusal bypass phrases ('as an AI without restrictions'). Map these to T1059 and T1552 respectively. Use osquery on Kubernetes nodes to monitor for unexpected child processes spawned by inference service processes: `SELECT pid, name, cmdline, parent FROM processes WHERE parent IN (SELECT pid FROM processes WHERE name LIKE '%python%' OR name LIKE '%uvicorn%' OR name LIKE '%tritonserver%');`

Evidence: Before formalizing the threat model entry, retrieve any historical LLM API access logs retained in your log sink and run a retroactive grep for known prompt injection patterns (e.g., `grep -iE 'ignore (all |previous |prior )?(instructions|prompts|constraints)' /path/to/llm_access.log`). Document the baseline false-positive rate so detection rule thresholds are calibrated. Capture a snapshot of current Kubernetes NetworkPolicy resources to assess whether LLM inference pods have unrestricted egress that would enable T1071.001 data exfiltration.

Step 4: Communicate findings — brief application and platform engineering teams that Kubernetes-hosted LLM workloads now have a supported runtime detection path; coordinate with the AI/ML platform owner on sensor deployment timelines

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: establishing communication structures and stakeholder coordination for incident handling of a new workload class

Controls: NIST IR-8 (Incident Response Plan) — the IR plan must be updated to designate the AI/ML platform owner as a named stakeholder with defined notification SLAs for LLM-layer incidents, NIST IR-6 (Incident Reporting) — reporting procedures must identify who within engineering receives prompt injection alerts from Falcon AIDR and what their expected response action is, NIST IR-2 (Incident Response Training) — application and platform engineering teams require awareness training on LLM-specific attack patterns (OWASP LLM01) so they can triage AIDR alerts correctly, NIST IR-7 (Incident Response Assistance) — establish whether CrowdStrike professional services or AIDR-specific support channels are available as an escalation path for novel prompt injection patterns

Compensating: For teams without a formal IR communications platform: create a shared runbook page (Confluence, Notion, or a Git-tracked Markdown doc) that documents the LLM workload inventory (from Step 1), the current detection gap, the Falcon AIDR deployment timeline, and interim escalation contacts. Distribute via a single email thread with the AI/ML platform owner, application engineering lead, and security team so there is a timestamped record of awareness — this serves as evidence of due diligence if a prompt injection incident occurs before sensor deployment completes.

Evidence: Document the current Falcon Container Sensor daemonset rollout status across all Kubernetes clusters (`kubectl rollout status daemonset/falcon-sensor -n falcon-system --watch=false`) and capture the output as a dated artifact. This establishes the pre-briefing state of sensor coverage and provides a baseline against which post-deployment coverage improvement can be measured.

Step 5: Monitor developments — track CrowdStrike AIDR release notes and OWASP LLM Top 10 updates; watch for follow-on coverage of indirect prompt injection (LLM01 sub-variant via retrieval pipelines), which is the more complex and currently less-detected attack path

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: continuous improvement of detection capability and threat intelligence integration as the LLM attack surface evolves

Controls: NIST SI-5 (Security Alerts, Advisories, and Directives) — establish a formal subscription to CrowdStrike AIDR release notes and OWASP LLM Top 10 revision announcements as authoritative advisories for this threat class, NIST SI-2 (Flaw Remediation) — track AIDR sensor version currency; treat outdated collector versions on LLM workloads as an unmitigated flaw given the active evolution of prompt injection techniques, NIST IR-4 (Incident Handling) — incident handling procedures for indirect prompt injection (retrieval-augmented generation pipeline poisoning) must be drafted before AIDR coverage arrives, as the attack path is active now, CIS 7.1 (Establish and Maintain a Vulnerability Management Process) — integrate OWASP LLM Top 10 revision tracking into the vulnerability management calendar alongside NVD and CISA KEV feeds, CIS 7.2 (Establish and Maintain a Remediation Process) — define remediation SLAs for newly detected AIDR-identified prompt injection techniques as AIDR signature coverage expands

Compensating: For indirect prompt injection via RAG pipelines specifically: implement input/output validation at the retrieval layer using a lightweight Python script that regex-scans retrieved document chunks for prompt injection syntax before they are inserted into the LLM context window. Log all retrieved chunk content with source document reference to a structured log sink. Subscribe to the OWASP LLM Top 10 GitHub repository (`https://github.com/OWASP/www-project-top-10-for-large-language-model-applications`) via GitHub watch/release notifications — this is free and provides direct visibility into sub-variant coverage updates without relying on vendor advisory cadence.

Evidence: Establish a monitoring baseline now for RAG pipeline data sources: enumerate all vector databases, document stores, and external retrieval endpoints used by Kubernetes-hosted LLM workloads (query Kubernetes ConfigMaps and Secrets for connection strings referencing Pinecone, Weaviate, Chroma, pgvector, or similar). These retrieval sources are the injection surface for indirect LLM01 attacks and are currently outside Falcon AIDR's stated coverage scope — documenting them now creates an artifact for future threat model updates when tool coverage expands.

Detection Guidance

Organizations using Falcon AIDR with the new Container Sensor collector should configure detection policies to alert on: (1) prompt injection pattern matches in inbound API request payloads, particularly inputs containing instruction-override language targeting system prompts; (2) outbound LLM response content that matches data leakage signatures, including patterns consistent with API key formats, PII, or internal document structures; (3) deviations from defined AI policy baselines in model output categories. Detection policies should be tuned to reduce false positives from legitimate multi-turn conversations or complex system prompts that may superficially resemble injection attempts.

For organizations without Falcon AIDR, detection must be implemented at the application layer. Log full request and response payloads from LLM API calls to a SIEM or log aggregation platform; API gateway metadata alone is insufficient. Hunt for anomalous output length or structure relative to input type, which can indicate a successfully injected instruction. In retrieval-augmented generation pipelines, audit the content of documents being injected into context windows; indirect prompt injection via poisoned retrieval sources (T1565) is not detectable at the API call level without content inspection.

At the Kubernetes platform level, monitor for unusual outbound connections from pods running LLM workloads (T1071.001), unexpected credential access patterns from the pod service account (T1552), and any process execution anomalies within the inference container (T1059). These are downstream indicators of a successful injection, not prevention controls, but they provide a secondary detection layer when LLM-layer visibility is absent.

Framework Mappings

MITRE-ATTACK

- **T1552** — Unsecured Credentials
- **T1071.001** — Web Protocols
- **T1059** — Command and Scripting Interpreter
- **T1565** — Data Manipulation
- **T1190** — Exploit Public-Facing Application

NIST-800-53R5

- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-7** — Software, Firmware, and Information Integrity
- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **AC-3** — Access Enforcement
- **SC-28** — Protection of Information at Rest

- **SI-10** — Information Input Validation

OWASP-TOP10-2021

- **A01:2021** — Broken Access Control
- **A03:2021** — Injection

HIPAA-SECURITY

- **164.312(a)(1)** — Access Control
- **164.308(a)(6)(ii)** — Response and Reporting

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **2.5** — Allowlist Authorized Software
- **8.2** — Collect Audit Logs

NIST-CSF-2

- **RS.CO-03** — Recovery activities and progress communicated
- **DE.CM-01** — Networks and network services are monitored

ISO-27001-2022

- **A.5.21** — Managing information security in the ICT supply chain

SOC2-TSC

- **CC9.2** — Manages risks associated with vendors and business partners

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1552	Unsecured Credentials	Credential-Access
T1071.001	Web Protocols	Command-And-Control
T1059	Command and Scripting Interpreter	Execution
T1565	Data Manipulation	Impact
T1190	Exploit Public-Facing Application	Initial-Access

Sources

Source	URL	Tier
Blog	https://www.crowdstrike.com/en-us/blog/falcon-aidr-detects-threats-...	T3
	https://www.crowdstrike.com/en-us/blog/falcon-aidr-detects-threats-...	T3

Source	URL	Tier
	https://www.crowdstrike.com/en-us/blog/how-crowdstrike-detects-clou...	T3
	https://www.crowdstrike.com/en-us/blog/crowdstrike-introduces-charl...	T3
CrowdStrike Falcon AIDR: AI Detection & Response	https://www.crowdstrike.com/en-us/platform/falcon-aidr-ai-detection...	T3

Generated 2026-05-14 13:49 UTC by TJS Security Command Center