INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-14 06:51 UTC

CrowdStrike Extends Falcon AIDR to Kubernetes AI Workloads, Addressing Prompt Layer Visibility Gap

SECURITY ANALYSIS | **MEDIUM** | CVSS 5.0

SCC Item ID	SCC-STY-2026-0127
Type	Security Analysis
Severity	MEDIUM
CVSS Base Score	5.0
Affected Products	CrowdStrike Falcon AIDR, Falcon Container Sensor (Kubernetes deployments), Falcon Cloud Security, Falcon Next-Gen SIEM; OpenAI-compatible LLM client applications running in Kubernetes
Discovery Source	Rss:T1 Threatintel

Executive Summary

CrowdStrike has extended its Falcon AIDR platform to monitor prompt-layer interactions in Kubernetes-hosted AI applications, addressing a visibility gap that has grown as organizations move large language model workloads into production cloud environments. Traditional security tooling cannot inspect what passes between an application and an LLM, leaving prompt injection and data leakage risks effectively blind to most security operations teams. This announcement signals that AI workload security is crossing from experimental discussion into operationally necessary capability, and organizations running LLM-backed applications in production should treat prompt-layer monitoring as a required control, not a future consideration.

Technical Analysis

The core problem Falcon AIDR's Kubernetes extension addresses is architectural: when an application sends a prompt to an LLM and receives a response, that interaction traverses an API layer that traditional EDR, SIEM log pipelines, and network-based inspection tools were not designed to parse for semantic threat content. Signature-based detection has no meaningful surface here. Log-based approaches capture request metadata but not prompt content. Network proxies can inspect traffic but require inline placement that introduces latency and architectural friction, particularly in containerized environments where workload density and ephemeral lifecycles complicate static proxy insertion.

The Falcon Container Sensor collector sits at the Kubernetes runtime layer, intercepting prompt interactions without requiring inline proxy deployment. This matters operationally: security teams can enable coverage without coordinating an architectural change with the application engineering team or accepting proxy-induced latency on LLM calls. The sensor feeds correlated alerts into Falcon Cloud Security and Falcon Next-Gen SIEM, meaning prompt-layer events can be contextualized against container telemetry, cloud control plane activity, and identity data in a single investigation workflow.

The threat classes targeted map directly to OWASP's LLM Top 10, published by the OWASP Foundation. Prompt injection (LLM01) sits at the top of that list for a reason: a successful injection can cause the model to ignore its system instructions, exfiltrate context window contents, or generate outputs that bypass policy controls. In a production application, the context window frequently contains system prompts encoding business logic, retrieved documents containing sensitive data, or prior conversation history. Sensitive data leakage (aligned with CWE-200) and policy violations represent the downstream consequences of successful injection rather than discrete attack classes.

The MITRE ATT&CK techniques associated with this capability announcement reflect the realistic attack surface: T1530 (Data from Cloud Storage), T1213 (Data from Information Repositories), T1087 (Account Discovery), T1611 (Escape to Host), T1059 (Command and Scripting Interpreter), T1190 (Exploit Public-Facing Application), and T1552 (Unsecured Credentials). Prompt injection as an initial access or lateral movement enabler maps cleanly to several of these, particularly T1059 and T1190, where a crafted prompt functions as a code injection or application exploitation vector. T1611 is relevant in multi-tenant or shared-context deployments where prompt manipulation might be used to escape isolation boundaries.

The announcement arrives as cloud-hosted adversary activity is rising. CrowdStrike's own reporting on cloud environment targeting, referenced in the supporting source material, establishes the threat context: cloud workloads are active targets, not just passive data stores, and the introduction of LLM workloads into those environments expands the exploitable surface without a corresponding expansion of visibility. This capability represents a vendor acknowledging that the detection tooling must evolve in step with the workload type, not lag behind it.

Action Checklist

1. Assess exposure: Inventory all Kubernetes-hosted AI workloads in your environment, specifically applications that make OpenAI-compatible API calls to any LLM, whether hosted internally or via a third-party provider.
2. Review controls: Audit whether your current EDR, CSPM, or SIEM tooling has any visibility into prompt-layer interactions for those workloads. If not, document the gap formally as an unmonitored attack surface.
3. Evaluate Falcon AIDR applicability: If your organization runs CrowdStrike Falcon, assess whether the Falcon Container Sensor collector for AIDR is deployed on Kubernetes nodes hosting AI workloads. If not deployed, engage your CrowdStrike account team for a deployment scoping conversation.
4. Update threat model: Add prompt injection (OWASP LLM01, CWE-77, CWE-20) as a tracked attack class in your threat register for any application that passes user-controlled input to an LLM, regardless of whether that application runs in Kubernetes.
5. Communicate findings: Brief application security and cloud security leadership on the prompt-layer visibility gap. Frame it around business risk: LLM applications processing sensitive business data or customer inputs are exposed to a class of attack that most current controls do not detect.

6. Monitor developments: Track OWASP LLM Top 10 updates and CrowdStrike Falcon AIDR release notes for additional detection coverage as the threat class matures. The OWASP LLM Top 10 is a living document updated as the community documents new attack patterns.

IR / Forensic Enrichment

Triage Priority	STANDARD
Escalation Criteria	Escalate to urgent if Falcon AIDR or compensating proxy telemetry detects any outbound LLM API response containing patterns matching internal data classification markers (PII regex, internal hostname patterns, credential formats), or if OWASP LLM01-style indirect prompt injection is detected via anomalous tool-call sequences originating from LLM responses in a production AI workload — either condition represents active exploitation of the prompt-layer blind spot and triggers potential breach notification assessment under applicable data protection regulations.
Recovery Notes	Following deployment of Falcon AIDR or a compensating interception proxy, conduct a 30-day baselining period during which prompt-layer telemetry is reviewed daily to establish normal interaction patterns for each AI workload before tuning alert thresholds — do not assume initial alert volume represents true positive rate, as legitimate applications frequently include structured data (JSON, code, internal terminology) in prompts that will trigger overly broad rules. After baselining, validate that all previously inventoried AI workload pods appear in AIDR telemetry with no gaps in coverage, cross-referencing pod UIDs from 'kubectl get pods -A' against AIDR-reported workload identifiers. Continue monitoring OWASP LLM Top 10 and MITRE ATLAS for new technique documentation that may require detection rule updates, with a 30-day SLA for incorporating new patterns into deployed rules after each framework update.
Forensic Artifacts	Kubernetes API server audit logs filtered for verb=create,exec on pods in AI workload namespaces — specifically look for unexpected kubectl exec events against LLM-connected pods that could indicate an attacker pivoting to exfiltrate prompt context or API keys stored as environment variables | Container stdout/stderr logs from AI workload pods collected via 'kubectl logs --timestamps --since=72h' — prompt injection payloads and LLM responses containing exfiltrated data will appear in application logs if the application logs request/response pairs, which many LangChain and similar framework defaults do at DEBUG level | Outbound TLS SNI records from Kubernetes node network interfaces captured via tcpdump or cloud VPC flow logs — filter on dst_port=443 and dst_host matching LLM API providers (api.openai.com, api.anthropic.com, .api.cognitive.microsoft.com for Azure OpenAI) to identify anomalous call frequency, payload size spikes, or calls to unexpected LLM endpoints not in the approved inventory | Kubernetes Secrets and ConfigMap access audit events for secrets named with patterns matching OPENAI_API_KEY, ANTHROPIC_API_KEY, or similar — unauthorized reads of these secrets by unexpected service accounts or user identities may indicate credential harvesting as a secondary objective of a prompt injection attacker who achieved application-layer code execution | mitmproxy or Envoy access log entries (if compensating controls are deployed) capturing HTTP POST request bodies to /v1/chat/completions containing anomalous system prompt override patterns such as 'ignore previous instructions', 'you are now', 'disregard your system prompt', or base64-encoded strings in the user message field — these are the primary forensic indicators of OWASP LLM01 prompt injection attempts specific to OpenAI-compatible API workloads

Per-Action IR Details

Assess exposure: Inventory all Kubernetes-hosted AI workloads in your environment, specifically applications that make OpenAI-compatible API calls to any LLM, whether hosted internally or via a third-party provider.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Establishing IR Capability and Asset Visibility

Controls: NIST IR-4 (Incident Handling) — preparation sub-phase requires knowing which assets are in scope before an incident occurs, NIST SI-4 (System Monitoring) — monitoring scope must include AI workload namespaces and their egress to LLM API endpoints, CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory) — extend inventory schema to tag Kubernetes workloads that issue HTTP POST requests to /v1/chat/completions or equivalent OpenAI-compatible endpoints, CIS 2.1 (Establish and Maintain a Software Inventory) — enumerate container images in AI workload pods to identify which LLM client libraries (openai-python, langchain, llama-index) are present

Compensating: Run 'kubectl get pods --all-namespaces -o json | jq ".items[] | select(.spec.containers[].env[]? | .name | test(\"OPENAI|LLM|API_KEY\")) | {namespace: .metadata.namespace, pod: .metadata.name}"' to identify pods with LLM API credentials in environment variables. Supplement with 'kubectl exec -n -- env | grep -iE "openai|anthropic|azure_openai|llm"' for runtime inspection. For network discovery, deploy a temporary tcpdump DaemonSet or use 'kubectl sniff' (ksniff plugin) to capture egress traffic from AI workload pods and filter on destination ports 443 to hostnames matching api.openai.com, api.anthropic.com, or internal LLM service DNS names.

Evidence: Before inventorying, capture: Kubernetes audit logs (kube-apiserver audit log at /var/log/kubernetes/audit.log or cloud provider equivalent — GKE: Cloud Audit Logs, EKS: CloudTrail with Kubernetes API server events, AKS: Azure Monitor) filtered for 'create' and 'exec' verbs against pods in namespaces suspected to host AI workloads. Export current pod specs ('kubectl get pod -A -o yaml > pod_inventory_baseline.yaml') as a point-in-time snapshot before any changes. Capture existing NetworkPolicy resources ('kubectl get networkpolicy -A -o yaml') to document what LLM egress restrictions, if any, were in place at time of assessment.

Review controls: Audit whether your current EDR, CSPM, or SIEM tooling has any visibility into prompt-layer interactions for those workloads. If not, document the gap formally as an unmonitored attack surface.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Identifying Capability Gaps Before Incidents Occur

Controls: NIST IR-8 (Incident Response Plan) — the IR plan must explicitly address detection gaps; prompt-layer blind spots in Kubernetes AI workloads must be documented as a known limitation with an accepted risk statement or compensating control, NIST AU-2 (Event Logging) — audit whether logging policy covers HTTP request/response bodies for LLM API calls originating from Kubernetes pods, not just connection metadata, NIST SI-4 (System Monitoring) — validate that monitoring coverage extends to application-layer prompt interactions, not only network flow data and OS-level telemetry from Falcon Container Sensor without AIDR, CIS 8.2 (Collect Audit Logs) — confirm audit log collection is enabled for the container runtime (containerd/CRI-O) and Kubernetes API server across all nodes hosting AI workloads, CIS 7.1 (Establish and Maintain a Vulnerability Management Process) — prompt injection as tracked in OWASP LLM01 must be formally assessed as part of the vulnerability management scope for any LLM-integrated application

Compensating: Without a SIEM or Falcon AIDR, deploy an mitmproxy instance as a sidecar container or transparent proxy within the AI workload pod to intercept and log HTTP traffic between the application and the LLM API endpoint. Use 'mitmproxy --mode transparent --ssl-insecure -w /tmp/llm_traffic.mitm' and export flows with 'mitmproxy -r /tmp/llm_traffic.mitm --flow-detail 3'. Alternatively, configure Envoy as a sidecar (Istio service mesh if present) with access logging set to log request body up to 4096 bytes, writing to stdout captured by the container logging driver. Document the gap using a simple risk register entry: asset name, LLM endpoint called, data classification of inputs, detection capability = none, accepted/mitigated/transferred.

Evidence: Capture Falcon sensor deployment status across Kubernetes nodes ('kubectl get ds -n falcon-system -o yaml' and 'kubectl describe ds falcon-sensor') to establish whether Falcon Container Sensor is present but AIDR collector is absent — this is the specific gap the advisory addresses. Export any existing SIEM data source inventory or parsing rules referencing HTTP body inspection for LLM API paths (/v1/chat/completions, /v1/completions, /v1/messages) to confirm zero coverage. Pull CSPM scan results for the AI workload namespaces from your cloud provider's security hub (AWS Security Hub, GCP Security Command Center, Azure Defender for Containers) and note

whether any findings relate to unencrypted egress, missing network policies, or over-permissioned service accounts bound to AI workload pods.

Evaluate Falcon AIDR applicability: If your organization runs CrowdStrike Falcon, assess whether the Falcon Container Sensor collector for AIDR is deployed on Kubernetes nodes hosting AI workloads. If not deployed, engage your CrowdStrike account team for a deployment scoping conversation.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Equipping the Team with Appropriate Tools and Detection Capability

Controls: NIST IR-7 (Incident Response Assistance) — engaging the CrowdStrike account team for AIDR deployment scoping is the correct application of vendor IR assistance resources defined in this control, NIST SI-4 (System Monitoring) — deploying Falcon Container Sensor with the AIDR collector on AI workload nodes directly implements system monitoring coverage for prompt-layer interactions that currently do not exist, NIST IR-4 (Incident Handling) — the AIDR collector is a detection capability component; its absence means the preparation sub-phase of IR is incomplete for AI workload incident classes, CIS 4.6 (Securely Manage Enterprise Assets and Software) — Falcon Container Sensor deployment must follow a documented, version-controlled deployment process; validate sensor version compatibility with your Kubernetes version and container runtime before rollout

Compensating: If Falcon AIDR is not available, deploy Falco (https://falco.org) as a DaemonSet on Kubernetes nodes hosting AI workloads and write custom rules to alert on container processes making outbound TLS connections to LLM API hostnames: 'rule: LLM_API_Egress; condition: evt.type=connect and fd.sip.name in (api.openai.com, api.anthropic.com) and container.name != host; output: LLM API call from container %container.name pod %k8s.pod.name; priority: NOTICE'. Pair with an eBPF-based tool such as Tetragon (Cilium) to log syscall-level network activity from AI workload pods without requiring a full service mesh. These provide process-level telemetry but not prompt content inspection — document this residual gap explicitly.

Evidence: Before engaging the account team, collect: current Falcon sensor version deployed ('kubectl exec -n falcon-system -- /opt/CrowdStrike/falconctl -g --version'), the Kubernetes version and node OS ('kubectl get nodes -o wide'), and the container runtime in use ('kubectl get nodes -o jsonpath="{.items[*].status.nodeInfo.containerRuntimeVersion}"') — AIDR collector compatibility is runtime-specific and this data is required for the scoping conversation. Also export the list of AI workload pod service accounts and their associated RBAC roles ('kubectl get rolebindings,clusterrolebindings -A -o wide | grep ') since AIDR sensor deployment will require specific permissions and security teams should review blast radius of sensor service account before approving.

Update threat model: Add prompt injection (OWASP LLM01, CWE-77, CWE-20) as a tracked attack class in your threat register for any application that passes user-controlled input to an LLM, regardless of whether that application runs in Kubernetes.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Establishing Incident Categories and Threat Classifications

Controls: NIST IR-4 (Incident Handling) — the incident handling capability must include threat categories relevant to the organization's technology stack; prompt injection against production LLM workloads is a distinct incident class requiring its own classification criteria and response triggers, NIST RA-3 (Risk Assessment) — prompt injection (OWASP LLM01/CWE-77/CWE-20) must be assessed as an explicit threat to confidentiality (data exfiltration via crafted prompts), integrity (LLM output manipulation), and availability (resource exhaustion via prompt flooding), NIST SI-10 (Information Input Validation) — CWE-20 (Improper Input Validation) is the root CWE for prompt injection; threat model entries should reference the specific input paths: user-supplied chat messages, document uploads processed by RAG pipelines, and tool-call parameters returned by the LLM that are executed by the application, CIS 7.1 (Establish and Maintain a Vulnerability Management Process) — OWASP LLM Top 10 should be added as a tracked reference standard in the vulnerability management process alongside CVE feeds, specifically for AI/ML components

Compensating: Use a free threat modeling tool such as OWASP Threat Dragon or Microsoft Threat Modeling Tool to create a data flow diagram for each AI application showing: user input → application → LLM API → response → application action. For each data flow crossing a trust boundary (especially user input entering the prompt construction layer), tag CWE-77 and CWE-20 and document whether input sanitization, prompt hardening (system prompt

separation), or output validation exists. Export the threat model as a JSON/XML artifact and store it in version control alongside the application's security documentation. Review quarterly against OWASP LLM Top 10 updates.

Evidence: Before updating the threat model, extract the current application architecture documentation and API interface specifications for each LLM-integrated application to identify all points where user-controlled data (HTTP request parameters, uploaded file content, database-retrieved user records) is concatenated into prompt strings without sanitization. Review application source code or container image layers ('docker history ' or 'dive ') for LLM client library versions (e.g., 'openai>=1.0.0', 'langchain>=0.1.0') since older versions may lack structured output enforcement or system prompt protection features. This pre-work grounds the threat model entries in actual code paths rather than hypothetical attack surfaces.

Communicate findings: Brief application security and cloud security leadership on the prompt-layer visibility gap. Frame it around business risk: LLM applications processing sensitive business data or customer inputs are exposed to a class of attack that most current controls do not detect.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Communicating IR Capability Status and Risk to Stakeholders

Controls: NIST IR-6 (Incident Reporting) — this control requires establishing reporting channels and escalation paths before incidents occur; briefing leadership on the prompt-layer gap is pre-incident reporting that ensures the organization can act decisively if a prompt injection incident is later confirmed, NIST IR-8 (Incident Response Plan) — the IR plan must reflect leadership awareness of novel attack classes; if leadership has not been briefed on prompt injection, the plan cannot include appropriate escalation thresholds or breach notification triggers for LLM data exfiltration scenarios, NIST AU-6 (Audit Record Review, Analysis, and Reporting) — the absence of audit records for prompt-layer interactions is itself a finding that must be reported; this control requires that gaps in audit coverage be communicated to responsible organizational officials, CIS 7.2 (Establish and Maintain a Remediation Process) — leadership briefing is a prerequisite to securing budget and prioritization for AIDR deployment or compensating controls; without documented risk acceptance or remediation commitment from leadership, the gap remains unowned

Compensating: Prepare a one-page risk brief using publicly available data to quantify the business risk without requiring proprietary tooling: reference OWASP LLM01:2025 (prompt injection), the MITRE ATLAS matrix (ML-specific adversary tactics, specifically AML.T0051 - LLM Prompt Injection), and any internal data classification assessments showing what data types flow through LLM applications (PII, financial records, IP). For teams without a GRC platform, use a simple spreadsheet risk register with columns: application name, LLM provider, data classification of inputs, current detection capability (none/partial/full), risk rating (likelihood × impact), and remediation owner. Present the detection gap as a binary: either Falcon AIDR or an equivalent intercepting proxy is deployed, or prompt injection attacks against these workloads are invisible to the SOC.

Evidence: Assemble evidence for the brief from: Kubernetes audit logs showing which service accounts and users have accessed AI workload namespaces in the last 90 days (to demonstrate the workloads are active and in use, not theoretical); cloud provider billing records or API gateway logs showing LLM API call volume and associated costs (to demonstrate production scale and business reliance on these workloads); and any existing WAF or API gateway logs from the 30 days prior that show inbound requests reaching the AI application layer — absence of prompt-content inspection in these logs is the concrete evidence of the visibility gap that leadership needs to see.

Monitor developments: Track OWASP LLM Top 10 updates and CrowdStrike Falcon AIDR release notes for additional detection coverage as the threat class matures. The OWASP LLM Top 10 is a living document updated as the community documents new attack patterns.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Lessons Learned and Continuous Improvement of Detection Capability

Controls: NIST SI-5 (Security Alerts, Advisories, and Directives) — establish a formal process to receive and act on OWASP LLM Top 10 updates and CrowdStrike AIDR release notes as authoritative advisories for this threat class, treating them equivalently to CISA KEV additions for AI workload risk, NIST IR-4 (Incident Handling) — the incident handling capability must be updated as new attack patterns in the OWASP LLM Top 10 are documented; detection rules, playbook triggers, and escalation criteria should be reviewed against each new OWASP LLM release, NIST SI-2

(Flaw Remediation) — new Falcon AIDR detection modules released by CrowdStrike for additional prompt injection variants or LLM data leakage patterns should be treated as flaw remediation updates and applied within the organization's standard patching SLA for security tooling, CIS 7.1 (Establish and Maintain a Vulnerability Management Process) — add OWASP LLM Top 10 and MITRE ATLAS as tracked reference sources in the vulnerability management process, with a defined review cadence (quarterly minimum) and a responsible owner for translating new entries into updated threat model entries and detection rules

Compensating: Subscribe to the OWASP LLM Top 10 GitHub repository (https://github.com/OWASP/www-project-top-10-for-large-language-model-applications) via GitHub's 'Watch > Releases' notification feature to receive alerts on document updates at no cost. For CrowdStrike AIDR release notes, if you have a Falcon console subscription, configure email digest notifications for product updates in the Falcon console notification settings. Maintain a simple changelog document in your team's wiki that maps each new OWASP LLM entry or AIDR capability release to: (1) whether any of your inventoried AI workloads are affected, (2) whether your current compensating controls cover the new pattern, and (3) any required updates to Falco rules, mitmproxy filters, or threat model entries. Review this document at each quarterly threat model review.

Evidence: This is a forward-looking monitoring step; however, establish a baseline artifact set before beginning monitoring so that future changes are measurable: export the current OWASP LLM Top 10 version and date ('curl -s https://raw.githubusercontent.com/OWASP/www-project-top-10-for-large-language-model-applications/main/Archive/0_1_vulns/LLM01_Prompt_Injection.md | head -5' or download the current PDF and hash it with 'sha256sum'), snapshot the current Falcon AIDR sensor version and detection rule version from the Falcon console, and export the current state of your AI workload threat register entries. These baselines allow you to demonstrate to auditors and leadership that monitoring is active and that changes to the threat landscape are being tracked and incorporated.

Detection Guidance

Detection focus for this story falls into two categories: detecting prompt injection attempts in flight, and detecting the downstream consequences of successful injection.

For prompt injection attempts, behavioral indicators include anomalous prompt lengths or structures inconsistent with the application's expected interaction pattern, prompt content containing instruction-override syntax (phrases directing the model to ignore prior instructions, reveal system prompts, or act as an unrestricted agent), and sudden shifts in model output character that suggest the model's instruction context was altered. These are not log events most SIEMs will surface natively; they require prompt-layer instrumentation of the type Falcon AIDR is designed to provide.

For downstream consequences, hunt in your cloud and Kubernetes telemetry for: unusual data egress from containers running LLM workloads (T1530, T1213), API calls to cloud storage or internal services initiated by application processes that would not normally make those calls (T1530), and credential material appearing in application logs or LLM response payloads (T1552). Falcon Next-Gen SIEM correlation between container sensor events and cloud control plane logs is the intended detection path for CrowdStrike customers.

For organizations without prompt-layer tooling, compensating controls include: output filtering on LLM responses before they reach downstream systems or users, strict least-privilege IAM for the service identity running the LLM application (limiting the blast radius of a successful injection), and logging the full request-response payload to an immutable log store for post-incident review. The last control does not prevent injection but preserves forensic evidence.

Policy gap audit: Review whether your AI application security policy formally addresses prompt injection as a threat class, and whether your secure development lifecycle includes prompt injection testing for applications that accept user input and pass it to an LLM.

Framework Mappings

MITRE-ATTACK

- **T1530** — Data from Cloud Storage
- **T1213** — Data from Information Repositories
- **T1087** — Account Discovery
- **T1611** — Escape to Host
- **T1059** — Command and Scripting Interpreter
- **T1190** — Exploit Public-Facing Application
- **T1552** — Unsecured Credentials

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
- **SI-10** — Information Input Validation
- **AC-3** — Access Enforcement
- **SC-28** — Protection of Information at Rest
- **AT-2** — Literacy Training and Awareness

OWASP-TOP10-2021

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

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks
- **8.2** — Collect Audit Logs

HIPAA-SECURITY

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

ISO-27001-2022

- **A.8.26** — Application security requirements
- **A.8.8** — Management of technical vulnerabilities

- **A.5.21** — Managing information security in the ICT supply chain
- **A.5.23** — Information security for use of cloud services

NIST-CSF-2

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

SOC2-TSC

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1530	Data from Cloud Storage	Collection
T1213	Data from Information Repositories	Collection
T1087	Account Discovery	Discovery
T1611	Escape to Host	Privilege-Escalation
T1059	Command and Scripting Interpreter	Execution
T1190	Exploit Public-Facing Application	Initial-Access
T1552	Unsecured Credentials	Credential-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
	https://www.crowdstrike.com/en-us/blog/crowdstrike-falcon-cloud-sec...	T3
	https://www.crowdstrike.com/en-us/blog/adversaries-increasingly-tar...	T3
CrowdStrike Falcon AIDR: AI Detection & Response	https://www.crowdstrike.com/en-us/platform/falcon-aidr-ai-detection...	T3

Solutions is not liable for any damages resulting from the use of this information.

Generated 2026-05-14 06:51 UTC by TJS Security Command Center