INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-01 14:04 UTC

Enterprise AI Blind Spots: Shadow AI Inventory Gaps Expose Organizations Across Endpoint, SaaS, and Cloud

GOVERNANCE | MEDIUM | CVSS 5.0

SCC Item ID	SCC-GOV-2026-0025
Type	Governance
Severity	MEDIUM
CVSS Base Score	5.0
Affected Products	Organizations using unapproved AI tools, agents, browser extensions, IDE plugins, and MCP servers; CrowdStrike Falcon platform customers (all versions with Shadow AI Visibility Service)
Discovery Source	Rss:T1 Threatintel

Executive Summary

Enterprises are undercounting active AI tools, agents, and extensions, creating governance blind spots that existing web filtering and policy controls cannot close. Agentic AI tools operate autonomously on endpoints, inherit user-level permissions, and move sensitive data outside approved boundaries without triggering conventional detection. The business risk is unauthorized data exposure, loss of AI governance posture, and accelerating regulatory scrutiny over AI asset management.

Technical Analysis

The core exposure is structural: enterprises rely on URL-category filtering and policy attestation to govern AI tool usage, but these controls do not account for agentic AI that operates at the process level on endpoints. AI agents, MCP (Model Context Protocol) servers, IDE plugins, and browser extensions inherit the executing user's permissions (CWE-284), bypass network-layer controls by design (CWE-693), and process or stage sensitive data without audit trails (CWE-200, CWE-1059). Relevant MITRE ATT&CK techniques include T1048 (Exfiltration Over Alternative Protocol), T1078 (Valid Accounts, agent credential inheritance), T1059 (Command and Scripting Interpreter, agent-executed scripts), T1195 (Supply Chain Compromise, ungoverned MCP servers and IDE extensions), and T1560 (Archive Collected Data, AI tools staging sensitive content). There is no CVE; this is a governance and architectural gap, not a discrete vulnerability. CrowdStrike's Shadow AI Visibility Service, announced and expanded at RSAC 2026, addresses discovery through endpoint telemetry and behavioral detection rather than network controls. Source quality is vendor-tier (T3); findings are attributed to CrowdStrike professional services engagements and are not independently corroborated by NIST, CISA, or

academic research at this time.

Action Checklist

1. Inventory: Deploy telemetry-based AI discovery rather than relying on URL filtering or user attestation. Query endpoint telemetry (EDR process trees, DNS logs, browser extension registries) for known AI tool signatures including MCP server processes, IDE plugin activity (Copilot, Cursor, Codeium), and GenAI browser extensions. URL-category filtering will miss process-level and API-based AI activity.
2. Detection: Search EDR logs for processes associated with agentic AI frameworks (e.g., AutoGPT, LangChain agents, Claude MCP servers). Review outbound API call patterns to endpoints such as api.openai.com, api.anthropic.com, and generativeai.googleapis.com from non-approved applications. Flag IDE processes making outbound connections to LLM APIs outside approved tooling. Identify browser extensions with AI permissions via extension inventory queries.
3. Eradication: Block or quarantine unauthorized AI agents and MCP servers identified during discovery. Revoke or scope down user-level permissions inherited by AI processes where technically feasible. Remove ungoverned IDE plugins and browser extensions from managed endpoints via MDM or endpoint policy enforcement. Do not rely solely on URL blocking to prevent re-installation.
4. Recovery: Validate that AI tool inventory matches approved software lists post-remediation. Monitor for re-emergence of blocked AI processes via EDR alerting. Confirm that data loss prevention (DLP) controls cover GenAI prompt content, not just file transfers, before restoring normal operations.
5. Post-Incident: Document the gap between policy-attested AI tool counts and telemetry-discovered counts. Use this delta to make the structural case for telemetry-based AI governance. Update acceptable use policies to explicitly cover agentic AI, MCP servers, and IDE plugins. Establish a repeating AI asset discovery cadence, not a one-time audit.

IR / Forensic Enrichment

Triage Priority	STANDARD
Escalation Criteria	Escalate to urgent if telemetry analysis reveals that any unauthorized AI agent or MCP server was granted OAuth access to production data repositories (code, customer data, financial records), or if discovered AI tool count exceeds 3x the policy-attested count — indicating systemic governance failure — or if the organization is subject to EU AI Act, NYDFS Cybersecurity Regulation, or SEC cybersecurity disclosure rules and the shadow AI exposure constitutes a material risk event requiring regulatory notification.
Recovery Notes	Before restoring normal operations, verify that DLP controls have been explicitly extended to cover HTTP POST body content to LLM API endpoints, not just file transfer channels — GenAI prompt exfiltration is invisible to file-centric DLP. Monitor EDR process trees and DNS logs for re-emergence of blocked AI tool process names for a minimum of 30 days post-remediation, as users will commonly reinstall tools through alternative channels (npm, pip, direct binary download) that bypass MDM app store controls. Confirm that all OAuth application authorizations and API keys associated with removed AI tools have been revoked in SaaS platforms and that the approved AI tool allowlist has been formally published to all developer and end-user populations before closing the incident.

Forensic Artifacts	MCP server configuration files (claude_desktop_config.json at %APPDATA%\Claude\, .cursor/mcp.json at %USERPROFILE%\.cursor\, ~/.continue/config.json on Linux/macOS) — these enumerate every registered MCP server binary, its granted filesystem scope, and data access permissions, establishing the exact data exposure boundary for unauthorized agentic AI activity | Sysmon Event ID 1 (Process Create) and Event ID 3 (Network Connection) logs correlated by ProcessGuid — specifically chains showing VS Code, JetBrains, or Cursor as parent processes spawning python.exe, node.exe, or uvx.exe child processes that then establish outbound connections to api.openai.com, api.anthropic.com, or generativeai.googleapis.com, confirming IDE plugin-to-LLM API call chains | Browser extension manifest files at %LOCALAPPDATA%\Google\Chrome\User Data\Default\Extensions\[extension_id]\[version]\manifest.json — the 'permissions' and 'host_permissions' arrays in each manifest confirm which GenAI extensions had access to page content, clipboard, native messaging, or specific web origins including internal SaaS applications | OAuth connected application authorization records from Google Workspace Admin (Security > API Controls > App Access Control) and Microsoft Entra ID (Enterprise Applications > User Consent) showing which AI tools were granted delegated access to corporate data under user-level permissions that survived endpoint remediation and require separate revocation | Windows Firewall or network proxy logs (Squid access.log, Zscaler transaction logs, or pfirewall.log) showing outbound HTTPS POST volume and frequency to LLM API endpoints by source workstation and username — POST request frequency and payload size patterns distinguish interactive use from autonomous agentic AI polling behavior, supporting data exposure scope estimation under MITRE ATT&CK T1567.002 (Exfiltration Over Web Service: Exfiltration to Cloud Storage)

Per-Action IR Details

Inventory: Deploy telemetry-based AI discovery rather than relying on URL filtering or user attestation. Query endpoint telemetry (EDR process trees, DNS logs, browser extension registries) for known AI tool signatures including MCP server processes, IDE plugin activity (Copilot, Cursor, Codeium), and GenAI browser extensions. URL-category filtering will miss process-level and API-based AI activity.

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Establishing visibility and detection capability before adverse events occur; CSF [GV, ID, PR]

Controls: NIST SI-4 (System Monitoring) — monitor endpoints for unapproved process execution and outbound API calls by AI agents, NIST CM-8 (System Component Inventory) — extend asset inventory to include AI tools, IDE plugins, browser extensions, and MCP server binaries, CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory) — add AI software instances to enterprise asset tracking, CIS 2.1 (Establish and Maintain a Software Inventory) — enumerate all installed IDE plugins (Copilot, Cursor, Codeium) and GenAI browser extensions alongside licensed software, CIS 8.2 (Collect Audit Logs) — ensure DNS query logs, EDR process trees, and browser extension registries are captured and retained for AI discovery queries

Compensating: On Windows endpoints: run 'Get-Process | Where-Object {$_.MainWindowTitle -like "*copilot*" -or $_.Name -like "*codeium*" -or $_.Name -like "*cursor*"}' and 'Get-ItemProperty HKLM:\SOFTWARE\WOW6432Node\Google\Chrome\Extensions -ErrorAction SilentlyContinue' to enumerate installed Chrome extensions. Use osquery with 'SELECT name, identifier, permissions FROM chrome_extensions WHERE permissions LIKE "%ai%" OR permissions LIKE "%openai%";' across managed endpoints. For MCP server discovery, query Sysmon Event ID 1 (Process Create) logs filtering on node.exe, python.exe, or uvx.exe spawned from IDE parent processes. Capture DNS query logs via dnscmd or Windows DNS debug logging filtered for api.openai.com, api.anthropic.com, and generativeai.googleapis.com.

Evidence: Capture before discovery sweep: (1) Snapshot of HKCU\Software\Google\Chrome\Extensions and equivalent Firefox/Edge extension registry paths on all managed endpoints to establish baseline extension inventory. (2) Sysmon Event ID 1 process creation logs for the prior 30 days filtered on known AI process names (node.exe with

--mcp flag, uvx.exe, aider, continue.dev). (3) DNS resolver cache dumps ('ipconfig /displaydns' on Windows) for LLM API hostnames. (4) CrowdStrike Falcon Shadow AI Visibility Service discovery report if licensed, which catalogs AI tool usage by process and user. (5) Browser extension manifest files at %LOCALAPPDATA%\Google\Chrome\User Data\Default\Extensions*\manifest.json for permissions scope review.

Detection: Search EDR logs for processes associated with agentic AI frameworks (e.g., AutoGPT, LangChain agents, Claude MCP servers). Review outbound API call patterns to endpoints such as api.openai.com, api.anthropic.com, and generativeai.googleapis.com from non-approved applications. Flag IDE processes making outbound connections to LLM APIs outside approved tooling. Identify browser extensions with AI permissions via extension inventory queries.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: Correlating telemetry sources to identify unauthorized AI process activity and data exfiltration patterns; CSF [DE]

Controls: NIST SI-4 (System Monitoring) — detect unapproved LangChain, AutoGPT, or MCP server processes making outbound API calls outside approved application allowlists, NIST AU-6 (Audit Record Review, Analysis, and Reporting) — review DNS and proxy logs on defined frequency for LLM API destinations from non-approved process contexts, NIST IR-5 (Incident Monitoring) — track and document discovered shadow AI instances as potential incidents requiring classification, CIS 8.2 (Collect Audit Logs) — confirm EDR process tree logs and outbound network connection logs are feeding detection queries, CIS 7.1 (Establish and Maintain a Vulnerability Management Process) — treat shadow AI discovery as a continuous detection task, not a one-time scan

Compensating: Deploy the Sigma rule 'proc_creation_win_lolbin_not_in_expected_path' adapted to flag python.exe or node.exe spawning from VS Code, JetBrains, or Cursor parent processes with outbound network connections. Use Sysmon Event ID 3 (Network Connection) filtered on destination IPs resolving to api.openai.com (104.18.x.x range) or api.anthropic.com (160.79.x.x range) — confirm current IP ranges via DNS at time of investigation. Run 'netstat -b 5' on suspect endpoints to capture active connections by owning process. For LangChain agent detection, search for Python processes with command lines containing 'langchain', 'autogpt', 'crewai', or 'mcp' via Sysmon Event ID 1. Use osquery 'SELECT pid, name, cmdline, local_address, remote_address FROM process_open_sockets WHERE remote_address NOT IN (SELECT allowed_ips FROM your_allowlist);' to correlate processes with outbound LLM API connections.

Evidence: Capture before analysis: (1) Sysmon Event ID 1 (Process Create) and Event ID 3 (Network Connection) logs correlated by ProcessGuid to map which IDE or agent process initiated each LLM API call. (2) Windows Firewall log (pfirewall.log) or network proxy access logs filtered for api.openai.com, api.anthropic.com, generativeai.googleapis.com, and huggingface.co destinations with source process attribution. (3) Browser extension permission manifests for extensions declaring 'tabs', 'webRequest', 'clipboardRead', or 'nativeMessaging' permissions — these are the permissions that enable GenAI extensions to read and transmit page content. (4) For MCP servers: contents of ~/.cursor/mcp.json, ~/.config/claude/claude_desktop_config.json, or equivalent MCP configuration files that enumerate registered MCP server binaries and their scope. (5) MITRE ATT&CK T1567 (Exfiltration Over Web Service) artifacts: outbound HTTPS POST volume to LLM API endpoints from developer workstations, captured via Zeek/Wireshark connection logs.

Eradication: Block or quarantine unauthorized AI agents and MCP servers identified during discovery. Revoke or scope down user-level permissions inherited by AI processes where technically feasible. Remove ungoverned IDE plugins and browser extensions from managed endpoints via MDM or endpoint policy enforcement. Do not rely solely on URL blocking to prevent re-installation.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: Removing unauthorized AI processes and revoking inherited permissions to eliminate the governance gap; CSF [RS]

Controls: NIST IR-4 (Incident Handling) — execute eradication actions consistent with the IR plan, including removal of unauthorized agentic AI software and permission revocation, NIST CM-7 (Least Functionality) — restrict endpoints to only approved AI tools by disabling installation pathways for unapproved IDE plugins and browser extensions, NIST AC-6 (Least Privilege) — revoke or scope down OAuth tokens, API keys, and user-level permissions inherited by AI agent processes that operated outside approved boundaries, CIS 2.3 (Address Unauthorized Software) — remove

unapproved AI tools, MCP server binaries, and IDE plugins; document exceptions with business justification, CIS 4.6 (Securely Manage Enterprise Assets and Software) — enforce removal via MDM policy push rather than manual user action to prevent re-installation through user-controlled channels

Compensating: Use Microsoft Intune (free tier for M365 subscribers) or Group Policy to deploy an extension blocklist for Chrome/Edge targeting known AI extension IDs (e.g., Merlin, Monica, Sider, ChatGPT for Google). For IDE plugins, push a VS Code settings policy via GPO or Intune that sets 'extensions.allowed' to an approved allowlist, blocking Cursor, Codeium, and unapproved Copilot instances. For MCP server removal on Windows: 'Get-ChildItem $env:APPDATA\npm\node_modules -Filter "*mcp*" | Remove-Item -Recurse -Force' and delete entries from claude_desktop_config.json. For API key revocation: enumerate and rotate any API keys stored in user-accessible locations ('Get-ChildItem $env:USERPROFILE -Recurse -Filter ".env" | Select-String "OPENAI_API_KEY|ANTHROPIC_API_KEY"'). Use Windows Firewall rules to block outbound 443 from specific process paths ('netsh advfirewall firewall add rule name="Block AutoGPT" program="C:\Users*\autogpt\run.bat" action=block dir=out') as a process-level block that survives URL filter bypasses.

Evidence: Capture before eradication: (1) Full dump of MCP server configuration files (claude_desktop_config.json, .cursor/mcp.json, ~/.continue/config.json) documenting what data scopes and filesystem access each registered server was granted — this establishes the data exposure boundary for post-incident review. (2) Screenshot or export of OAuth application authorizations in corporate SaaS platforms (Google Workspace, Microsoft 365 connected apps) for any AI tools that requested delegated access — these persist after tool removal and must be revoked separately. (3) List of API keys or tokens found in user-accessible .env files, IDE settings, or browser local storage before rotation. (4) Sysmon Event ID 11 (File Create) logs showing when MCP server binaries or AI agent scripts were first written to disk, establishing initial access timeline. (5) CrowdStrike Falcon process ancestry trees (if licensed) or manually reconstructed Sysmon parent-child chains showing how AI agent processes were launched and what child processes or network connections they spawned.

Recovery: Validate that AI tool inventory matches approved software lists post-remediation. Monitor for re-emergence of blocked AI processes via EDR alerting. Confirm that data loss prevention (DLP) controls cover GenAI prompt content, not just file transfers, before restoring normal operations.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery: Verifying clean state and confirming DLP coverage for GenAI prompt content before returning to normal operations; CSF [RC]

Controls: NIST SI-7 (Software, Firmware, and Information Integrity) — verify post-remediation endpoint state by comparing installed software and extension inventories against approved baselines, NIST SI-4 (System Monitoring) — establish ongoing EDR alerting for re-emergence of blocked AI process names and outbound LLM API connections from non-approved applications, NIST IR-4 (Incident Handling) — confirm recovery actions are complete and document restoration of normal operations with governance controls verified, CIS 2.2 (Ensure Authorized Software is Currently Supported) — confirm that only approved, vendor-supported AI tools remain installed post-remediation, CIS 7.2 (Establish and Maintain a Remediation Process) — track remediation completion against the discovered shadow AI inventory and document residual risk for items that cannot be fully removed

Compensating: Schedule a recurring osquery query (weekly via osquery scheduled queries config) to re-run the extension and process inventory from the Inventory step and diff against the approved software baseline — alert on any new AI-related entries. For DLP coverage of GenAI prompts without enterprise DLP tooling: configure Squid proxy or Pi-hole DNS sinkhole to log and alert on POST requests to LLM API endpoints (api.openai.com/v1/chat/completions, api.anthropic.com/v1/messages) — POST body size anomalies (>10KB) from developer workstations may indicate bulk prompt content. Use Sysmon Event ID 1 re-alerting by adding AI process names to a custom Sigma rule deployed to Windows Event Forwarding. Verify recovery completeness with a manual re-run of the osquery extension inventory and 'Get-InstalledModule' / 'pip list' checks for Python-based AI agent packages on developer workstations.

Evidence: Capture before declaring recovery complete: (1) Post-remediation software inventory snapshot from osquery or MDM compared against pre-remediation baseline to confirm removal of all discovered shadow AI tools — delta of zero is the acceptance criterion. (2) EDR alert configuration export confirming new detection rules for blocked AI process names are active and have been tested with a benign simulation. (3) DLP policy scope documentation confirming that HTTP POST body inspection or API proxy logging is covering LLM API endpoints — specifically validating that prompt content (not just file attachment transfers) is within policy scope. (4) Re-run of DNS log query for

LLM API hostnames from the prior 7 days post-remediation to confirm absence of ongoing unauthorized API calls. (5) Revocation confirmation receipts for all OAuth application authorizations and API keys identified during eradication, with timestamps.

Post-Incident: Document the gap between policy-attested AI tool counts and telemetry-discovered counts. Use this delta to make the structural case for telemetry-based AI governance. Update acceptable use policies to explicitly cover agentic AI, MCP servers, and IDE plugins. Establish a repeating AI asset discovery cadence, not a one-time audit.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Documenting the 3x undercounting gap as evidence for governance investment and updating policy to explicitly address agentic AI; CSF [GV, ID]

Controls: NIST IR-4 (Incident Handling) — incorporate lessons learned from the shadow AI discovery into updated incident handling procedures for AI governance violations, NIST IR-8 (Incident Response Plan) — update the IR plan to include agentic AI, MCP servers, and IDE plugins as asset categories with defined discovery, triage, and response procedures, NIST SI-5 (Security Alerts, Advisories, and Directives) — establish a process to receive and act on future AI tool governance advisories from CISA, NIST AI RMF updates, and industry sources, CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory) — institutionalize AI tool discovery as a permanent component of asset inventory, reviewed on a defined repeating cadence, CIS 7.1 (Establish and Maintain a Vulnerability Management Process) — extend vulnerability management process scope to include AI tool governance risk, treated with the same discovery cadence as software vulnerability scanning

Compensating: Formalize the osquery scheduled query from Recovery into a monthly automated report distributed to the security and IT governance teams — no SIEM required, just a cron job exporting query results to a shared mailbox. Draft acceptable use policy addendum using NIST AI RMF (AI 100-1) definitions of 'agentic AI system' and 'AI-enabled tool' to ensure the policy language covers MCP servers and IDE plugins explicitly, not just 'AI chatbots'. Store the pre/post telemetry counts (policy-attested vs. discovered) in a governance register as quantitative evidence for budget requests for CrowdStrike Shadow AI Visibility Service or equivalent. Schedule the next AI asset discovery run in 30 days and quarterly thereafter, documented in the IR after-action report as a standing operational commitment.

Evidence: Preserve as post-incident record: (1) The final telemetry-discovered AI tool count versus policy-attested count, broken down by category (MCP servers, IDE plugins, browser extensions, standalone agents) — this is the primary evidence artifact documenting the governance gap. (2) Timeline reconstruction from Sysmon and DNS logs showing how long unauthorized AI tools were active before discovery (dwell time equivalent for shadow AI governance incidents). (3) Data exposure boundary assessment documenting which AI tools had access to sensitive data repositories (code repos, SharePoint, file shares) via inherited user permissions, derived from the OAuth scope and MCP server configuration artifacts captured during eradication. (4) After-action report documenting policy gaps (acceptable use policy did not enumerate agentic AI or MCP servers), detection gaps (URL filtering did not detect process-level AI API calls), and control gaps (no DLP coverage for prompt content). (5) Updated acceptable use policy version with explicit definitions, approved AI tool allowlist, and mandatory disclosure requirement for new AI tool requests — retained as governance evidence for future regulatory inquiries.

Detection Guidance

Primary detection method is endpoint telemetry, not network logs. Query EDR process trees for known agentic AI process names and parent-child relationships (e.g., IDE spawning LLM API client processes). Review outbound TLS connections to LLM API hostnames from endpoints that lack explicit authorization. Audit browser extension inventories for AI-category extensions with broad host permissions. Review developer workstations for MCP server processes listening on localhost ports. DNS query logs can surface GenAI API domains but will miss locally executing agents that communicate over HTTPS without a distinct hostname pattern. There are no published IOC hashes or IP indicators for this governance gap; detection depends on behavioral baselines and approved-versus-discovered asset comparison.

Framework Mappings

MITRE-ATTACK

- **T1190** — Exploit Public-Facing Application
- **T1048** — Exfiltration Over Alternative Protocol
- **T1059** — Command and Scripting Interpreter
- **T1036** — Masquerading
- **T1560** — Archive Collected Data
- **T1567** — Exfiltration Over Web Service
- **T1530** — Data from Cloud Storage
- **T1078** — Valid Accounts
- **T1526** — Cloud Service Discovery
- **T1195** — Supply Chain Compromise

NIST-800-53R5

- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-7** — Software, Firmware, and Information Integrity
- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **AC-2** — Account Management
- **AC-6** — Least Privilege
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **SA-9** — External System Services
- **SR-2** — Supply Chain Risk Management Plan
- **SR-3** — Supply Chain Controls and Processes
- **AC-3** — Access Enforcement
- **SC-28** — Protection of Information at Rest

OWASP-TOP10-2021

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

CIS-V8

- **6.1** — Establish an Access Granting Process
- **6.2** — Establish an Access Revoking Process
- **8.2** — Collect Audit Logs

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets

HIPAA-SECURITY

- **164.312(a)(1)** — Access Control

NIST-CSF-2

- **DE.CM-01** — Networks and network services are monitored

ISO-27001-2022

- **A.5.23** — Information security for use of cloud services

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1190	Exploit Public-Facing Application	Initial-Access
T1048	Exfiltration Over Alternative Protocol	Exfiltration
T1059	Command and Scripting Interpreter	Execution
T1036	Masquerading	Defense-Evasion
T1560	Archive Collected Data	Collection
T1567	Exfiltration Over Web Service	Exfiltration
T1530	Data from Cloud Storage	Collection
T1078	Valid Accounts	Defense-Evasion
T1526	Cloud Service Discovery	Discovery
T1195	Supply Chain Compromise	Initial-Access

Sources

Source	URL	Tier
Blog	https://www.crowdstrike.com/en-us/blog/crowdstrike-shadow-AI-visibi...	T3
	https://www.crowdstrike.com/en-us/blog/crowdstrike-secures-growing-...	T3
	https://www.crowdstrike.com/en-us/blog/crowdstrike-stops-genai-data...	T3
	https://siliconangle.com/2026/03/23/crowdstrike-targets-ai-security...	T3

Source	URL	Tier
CrowdStrike Shadow AI Visibility Service | Reduce AI Footprint Risk	https://www.crowdstrike.com/en-us/services/ai-security-services/sha...	T3

Generated 2026-05-01 14:04 UTC by TJS Security Command Center