INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-07-08 14:45 UTC

CrowdStrike Expands AI Attack Taxonomy to 200+ Techniques as Agentic Systems Become Prime Targets

SECURITY ANALYSIS | HIGH | CVSS 7.5

SCC Item ID	SCC-STY-2026-0335
Type	Security Analysis
Severity	HIGH
CVSS Base Score	7.5
Affected Products	AI agents and LLM-integrated systems broadly; systems with tool-calling, file access, or shell execution capabilities; CrowdStrike Falcon AIDR; Google Gemini (referenced example); Anthropic Claude Desktop (referenced example)
Discovery Source	Rss:T1 Threatintel

Executive Summary

CrowdStrike's AI security research team has published an expanded prompt injection taxonomy documenting more than 200 distinct techniques targeting LLM-integrated systems, including 18 newly identified methods, five of which are described in detail. The research argues that agentic AI systems, those granted tool-calling, file access, or shell execution permissions, represent a qualitatively different and more consequential attack surface than standard chatbot interfaces, where successful injection can translate directly into code execution, data exfiltration, or privilege escalation. This signals a maturing threat category: AI integration is no longer an abstract risk surface but a documented, technique-catalogued attack vector requiring the same systematic defensive treatment organizations apply to traditional endpoint and network threats.

Technical Analysis

CrowdStrike's research, published on its official blog, frames prompt injection against agentic systems as a structural trust boundary problem rather than a simple input validation failure. The five publicly disclosed techniques focus on exploiting the context pipeline that connects user input, system prompts, retrieved data, and tool invocations in multi-step agentic workflows. In these architectures, a malicious payload embedded in an external data source, a document, a web page, an email, can traverse the pipeline and be interpreted as a legitimate instruction by the agent's orchestration layer, triggering downstream tool calls the operator never authorized.

The attack surface expands significantly when agents hold shell execution or file system permissions. CrowdStrike assesses that standard input validation controls are insufficient against this technique set because the injection point is often not the user-facing interface but an upstream data feed the agent trusts by design. CrowdStrike's research, supported by a documented proof-of-concept against Anthropic Claude Desktop (referenced in https://cyberpress.org/claude-desktop-command-execution-attack/), demonstrates that at least one technique has moved from theoretical to demonstrated risk.

MITRE ATT&CK for MLSystems techniques AML.T0051 (LLM Prompt Injection) and AML.T0054 (Prompt Injection via Untrusted Data) map directly to the described attack patterns, alongside conventional enterprise techniques T1059 (Command and Scripting Interpreter) and T1190 (Exploit Public-Facing Application), reflecting how successful agentic compromise bridges the AI and traditional infrastructure threat models. CrowdStrike simultaneously announced Falcon AIDR and related platform capabilities framed around AI agent governance and shadow AI detection, positioning the research as both a threat disclosure and a product category argument. Security teams should evaluate the research findings independently of the commercial framing.

Action Checklist

1. Step 1: Assess exposure, inventory every AI agent, LLM-integrated workflow, or AI-assisted tool deployed in your environment, including shadow AI deployments; specifically identify any instance granted tool-calling, file access, code execution, or shell permissions (CIS 1.1: Establish and Maintain Detailed Enterprise Asset Inventory; CIS 2.1: Establish and Maintain a Software Inventory)
2. Step 2: Review controls, audit the trust boundary configuration of each agentic system; verify that external data sources (documents, web content, email) fed into agent context pipelines are treated as untrusted input and do not carry instruction-level authority; validate that least-privilege principles govern what tools agents can invoke (NIST AC-6: Least Privilege; NIST AC-3: Access Enforcement; NIST AC-4: Information Flow Enforcement)
3. Step 3: Update threat model, add AML.T0051 and AML.T0054 to your threat register as active technique categories; document which internal agentic systems present viable attack paths for context pipeline injection leading to T1059-class execution or data exfiltration
4. Step 4: Evaluate detection coverage, assess whether your SIEM and EDR telemetry can attribute unexpected shell execution or file access events to an AI agent process rather than a human user; log AI agent tool invocations with sufficient context to support post-incident analysis (NIST AU-2: Event Logging; NIST AU-3: Content of Audit Records; CIS 8.2: Collect Audit Logs)
5. Step 5: Communicate findings, brief engineering and product leadership on the specific risk that external data ingested by agentic systems can carry executable instructions; this is an architectural conversation, not a patch cycle; frame it as a design constraint requiring explicit trust boundary enforcement before expanding agent permissions
6. Step 6: Monitor developments, track CrowdStrike's full taxonomy publication for the remaining 13 of the 18 newly identified techniques not yet publicly disclosed; monitor the Claude Desktop proof-of-concept disclosure for technical detail that may illuminate detection opportunities applicable to your deployed agents

IR / Forensic Enrichment

Triage Priority	URGENT
Escalation Criteria	Escalate immediately to CISO and legal if EDR or Sysmon telemetry records a shell execution or unauthorized file access event with an LLM agent process (e.g., python.exe, node.exe running a LangChain or MCP-backed agent) as the parent, particularly if outbound network connections to non-whitelisted destinations are observed in the same time window, as this indicates a likely successful prompt injection achieving code execution and potential data exfiltration requiring breach notification assessment.
Recovery Notes	After remediating trust boundary misconfigurations — restricting agent tool permissions to least-privilege and enforcing untrusted-input handling for all external context sources — verify recovery by replaying representative benign workloads through each agent and confirming that tool invocations are logged with full command-line context and attributable to the agent process identity. Monitor agent process trees and outbound network connections for a minimum of 30 days post-remediation, as prompt injection payloads can be embedded in long-lived data sources (cached documents, email archives, shared knowledge bases) that may not be immediately purged. Update the threat register with confirmed detection rule coverage for AML.T0051 and AML.T0054 before returning any agent with shell or file-write permissions to full operational status.
Forensic Artifacts	LLM agent process tree logs (Sysmon Event ID 1 — Process Creation): captures parent-child relationships showing the agent runtime (python.exe, node.exe) spawning unexpected child processes such as cmd.exe or powershell.exe, which is the direct forensic signature of a successful prompt injection achieving T1059-class execution through an agentic tool-call | Agent context window / conversation logs: if the agent platform retains prompt history (e.g., LangChain callback logs, Claude Desktop session files at '~/.config/claude/', OpenAI Assistants thread logs via API), these contain the injected instruction payload itself and are the primary artifact for reconstructing the attack vector and identifying the malicious data source that introduced it | File system change logs for agent working directories (Sysmon Event ID 11 — FileCreate): documents files written by the agent process outside its expected output paths, which may represent staged exfiltration payloads or persistence artifacts dropped as a result of injected shell commands | Outbound network connection records (Sysmon Event ID 3 — Network Connection or 'netstat -ano' snapshot tied to agent PID): captures exfiltration attempts initiated by the agent process to attacker-controlled infrastructure, particularly unexpected DNS lookups or HTTP POST requests from the agent runtime process identity | MCP configuration file and tool manifest snapshots (e.g., '~/.config/claude/claude_desktop_config.json' for Claude Desktop, or equivalent tool-declaration files for LangChain/AutoGPT deployments): establishes the declared tool permission scope at the time of the incident, which is essential for determining the maximum blast radius of any successful injection and for distinguishing authorized from unauthorized tool invocations in post-incident review

Per-Action IR Details

Step 1: Assess exposure — inventory every AI agent, LLM-integrated workflow, or AI-assisted tool deployed in your environment, including shadow AI deployments; specifically identify any instance granted tool-calling, file access, code execution, or shell permissions (CIS 1.1: Establish and Maintain Detailed Enterprise Asset Inventory; CIS 2.1: Establish and Maintain a Software Inventory)

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Establish IR capability and asset visibility before incidents occur

Controls: CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.1 (Establish and Maintain a Software Inventory), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Run 'Get-Process | Where-Object {$_.Path -like "*python*" -or $_.Path -like "*node*"}' on Windows endpoints to surface interpreter processes that may be backing LLM agent runtimes; on Linux, use 'ps aux | grep -E "langchain|autogpt|ollama|litellm|openai"' to identify active agent processes. Maintain a spreadsheet enumerating each agent's granted OS permissions and external data sources.

Evidence: This is a preparation/inventory step and does not alter live system state; no volatile capture is required before execution. Document discovered agents' process names, parent processes, and permission scopes as a baseline for later comparison if anomalous behavior is detected.

Step 2: Review controls — audit the trust boundary configuration of each agentic system; verify that external data sources (documents, web content, email) fed into agent context pipelines are treated as untrusted input and do not carry instruction-level authority; validate that least-privilege principles govern what tools agents can invoke (NIST AC-6: Least Privilege; NIST AC-3: Access Enforcement; NIST AC-4: Information Flow Enforcement)

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Implement preventive controls and validate architecture before exploitation occurs

Controls: NIST AC-6 (Least Privilege), NIST AC-3 (Access Enforcement), NIST AC-4 (Information Flow Enforcement)

Compensating: For each agent, manually review its tool manifest or MCP (Model Context Protocol) configuration file — in Claude Desktop this is '~/.config/claude/claude_desktop_config.json' — and enumerate every declared tool and its filesystem or shell scope. For LangChain-based agents, inspect the 'tools=[]' constructor argument in source code to confirm no unrestricted shell or filesystem tools are present without explicit scope constraints.

Evidence: This step reviews configuration without altering live state; no volatile capture required. Preserve a read-only snapshot of each agent's configuration files and tool permission manifests (e.g., MCP config JSON, LangChain tool definitions, OpenAI function-calling schemas) as timestamped baseline artifacts before any remediation changes are applied.

Step 3: Update threat model — add AML.T0051 and AML.T0054 to your threat register as active technique categories; document which internal agentic systems present viable attack paths for context pipeline injection leading to T1059-class execution or data exfiltration

NIST Phase: Preparation

Reference: NIST 800-61r3 §2 — Preparation: Maintain threat intelligence and update IR procedures to reflect current adversary techniques

Compensating: Create a threat register entry in a shared spreadsheet or wiki page for each agentic system, mapping it to the MITRE ATLAS techniques AML.T0051 (LLM Prompt Injection) and AML.T0054 (LLM Jailbreak) as threat categories, and document the specific tool permissions that would make exploitation consequential (e.g., 'Agent X has shell execution — successful injection could achieve command execution equivalent to the agent's OS user context').

Evidence: No live system state is altered by this step; no volatile capture required. The output artifact is the updated threat register itself, which should be versioned and dated to establish a documented pre-incident awareness baseline.

Step 4: Evaluate detection coverage — assess whether your SIEM and EDR telemetry can attribute unexpected shell execution or file access events to an AI agent process rather than a human user; log AI agent tool invocations with sufficient context to support post-incident analysis (NIST AU-2: Event Logging; NIST AU-3: Content of Audit Records; CIS 8.2: Collect Audit Logs)

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: Establish monitoring capability to identify and attribute anomalous events to their source

Controls: NIST AU-2 (Event Logging), NIST AU-3 (Content Of Audit Records), CIS 8.2 (Collect Audit Logs)

Compensating: Deploy Sysmon with a configuration that sets ProcessCreation (Event ID 1) and FileCreate (Event ID 11) rules targeting the specific process names of your deployed LLM agent runtimes (e.g., 'python.exe', 'node.exe', 'ollama.exe') as parent processes; any child process spawned by these — especially cmd.exe, powershell.exe, or bash

— should generate an immediate alert. Use the free Sigma rule 'proc_creation_win_susp_script_exec_from_env' as a starting template, modified to filter on AI agent parent processes.

Evidence: Before modifying any logging configuration on a host where an agent is actively running and potentially under exploitation, capture: (1) current running process tree via 'Get-CimInstance Win32_Process | Select ProcessId,ParentProcessId,Name,CommandLine' or 'ps auxf' on Linux; (2) active network connections via 'Get-NetTCPConnection' or 'netstat -ano' to identify any unexpected outbound connections initiated by the agent process; (3) recent file system changes in the agent's working directory. These volatile artifacts establish the pre-change baseline and may reveal in-progress exfiltration if an active injection is underway.

Step 5: Communicate findings — brief engineering and product leadership on the specific risk that external data ingested by agentic systems can carry executable instructions; this is an architectural conversation, not a patch cycle; frame it as a design constraint requiring explicit trust boundary enforcement before expanding agent permissions

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Translate findings into organizational awareness and architectural improvement to reduce future risk

Controls: NIST AC-1 (Policy And Procedures)

Compensating: Prepare a one-page brief using a concrete example drawn from the CrowdStrike taxonomy — specifically the demonstrated Claude Desktop proof-of-concept where a malicious document in the context window triggered tool invocations — to make the risk tangible for non-technical leadership; pair it with a matrix showing each deployed agent, its current tool permissions, and the blast radius if a prompt injection succeeded (e.g., 'Agent Y has email send permission — successful injection could exfiltrate data to attacker-controlled address').

Evidence: No live system state is altered by this step; no volatile capture required. Documentation artifacts from Steps 1–4 (asset inventory, permission audit, threat register, detection gap analysis) serve as the evidentiary basis for this briefing and should be preserved in their pre-briefing state.

Step 6: Monitor developments — track CrowdStrike's full taxonomy publication for the remaining 13 non-publicly-disclosed techniques; monitor the Claude Desktop proof-of-concept disclosure for technical detail that may illuminate detection opportunities applicable to your deployed agents

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: Continuously integrate new threat intelligence to improve detection and update IR procedures

Controls: NIST AU-6 (Audit Record Review, Analysis, And Reporting)

Compensating: Subscribe to CrowdStrike's adversarial ML research RSS feed and the MITRE ATLAS update feed (atlas.mitre.org) via a free RSS reader; configure a GitHub search alert for 'prompt injection' and 'MCP exploit' to surface new proof-of-concept disclosures; assign a named team member to review and summarize new technique disclosures weekly and map them against the agent inventory produced in Step 1.

Evidence: No live system state is altered by this step; no volatile capture required. Maintain a dated intelligence log recording each new technique disclosure, its assessed applicability to deployed agents, and any detection rule updates triggered, to support audit and post-incident review if a future incident exploits a technique first documented during this monitoring period.

Detection Guidance

Behavioral patterns to hunt for center on unexpected tool invocations originating from AI agent processes. Specifically: (1) Shell or command interpreter spawning (cmd.exe, bash, PowerShell) where the parent process is an AI agent runtime or LLM integration service, correlate with NIST AU-2 event logging requirements and flag any such chain absent an explicit operator-initiated workflow. (2) File system reads or writes by agent processes against directories outside the agent's declared operational scope, particularly reads of credential stores,

configuration files, or SSH keys. (3) Outbound network connections initiated by agent processes to destinations not present in the agent's configured tool registry, potential indicator of exfiltration following a successful injection. (4) Context pipeline inputs arriving from external sources (web retrieval, document ingestion, email parsing) that contain instruction-formatted text patterns, including role redefinition strings, jailbreak prefixes, or tool invocation syntax embedded in what should be data payloads. (5) Audit log gaps or suppression: agents that stop generating expected tool-call audit records mid-session may indicate a session hijack or context override. Log AI agent activity at the tool-invocation level, not just at the session level, per NIST AU-3 and AU-12 requirements. MITRE D3FEND countermeasures applicable: D3-UAP (User Account Permissions, restrict agent runtime accounts to minimum required tool scope), D3-LAM (Local Account Monitoring, monitor agent service accounts for anomalous access patterns), D3-SFA (System File Analysis, monitor for unexpected reads of sensitive system files by agent processes). No specific IOC values (hashes, domains, IPs) were present in the provided source material. The CrowdStrike blog post at https://www.crowdstrike.com/en-us/blog/crowdstrike-uncovers-new-prompt-injection-techniques/ may contain additional technical indicators, consult that source directly for any published values.

Framework Mappings

MITRE-ATTACK

- **T1059** — Command and Scripting Interpreter
- **T1566** — Phishing
- **T1078** — Valid Accounts
- **T1190** — Exploit Public-Facing Application
- **T1203** — Exploitation for Client Execution

NIST-800-53R5

- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-7** — Software, Firmware, and Information Integrity
- **AT-2** — Literacy Training and Awareness
- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **SI-8** — Spam Protection
- **AC-2** — Account Management
- **AC-6** — Least Privilege
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SI-2** — Flaw Remediation
- **SI-10** — Information Input Validation

OWASP-TOP10-2021

- **A03:2021** — Injection

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
- **5.4** — Restrict Administrator Privileges to Dedicated Administrator Accounts

ISO-27001-2022

- **A.8.26** — Application security requirements
- **A.5.34** — Privacy and protection of personal information

HIPAA-SECURITY

- **164.308(a)(6)(ii)** — Response and Reporting

SOC2-TSC

- **CC7.4** — Responds to identified security incidents
- **CC6.3** — Authorizes, modifies, or removes access

NIST-CSF-2

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1059	Command and Scripting Interpreter	Execution
T1566	Phishing	Initial-Access
T1078	Valid Accounts	Defense-Evasion
T1190	Exploit Public-Facing Application	Initial-Access
T1203	Exploitation for Client Execution	Execution

Sources

Source	URL	Tier
Blog	https://www.crowdstrike.com/en-us/blog/crowdstrike-uncovers-new-pro...	T1
Cyberpress	https://cyberpress.org/crowdstrike-5-new-prompt-injection-techniques/	T3
Gbhackers	https://gbhackers.com/crowdstrike-uncovers-5-new-prompt-injection-t...	T2

Source	URL	Tier
Cyberpress	https://cyberpress.org/claude-desktop-command-execution-attack/	T3
CrowdStrike Innovations Secure AI Agents and Govern ...	https://www.crowdstrike.com/en-us/blog/new-crowdstrike-innovations-...	T1

Generated 2026-07-08 14:45 UTC by TJS Security Command Center