AI Developer Tool Supply Chain Attacks: RCE, Fake Installers, and AI-Promoted Malicious Repos

Unofficial AI-assisted Research

Cloud Security Alliance AI Safety Initiative

2026-03-08

Key Takeaways

Between late 2023 and early 2026, threat actors mounted an escalating series of supply chain attacks targeting the AI developer tooling ecosystem. The incidents documented in this note span six distinct attack categories: malicious IDE extensions, fake AI tool installers distributed through malvertising, package ecosystem compromises exploiting AI brand recognition, a novel class of attacks exploiting LLM package hallucinations ("slopsquatting"), remote code execution vulnerabilities in AI coding assistants and model servers, and compromise of the emerging Model Context Protocol (MCP) infrastructure.

Several structural characteristics make AI developers a particularly high-value target. They typically operate with elevated local privileges and broad network access, manage sensitive credentials for cloud environments and code repositories, and in many cases follow AI-generated installation instructions with reduced verification friction — a behavior pattern threat actors appear to be actively exploiting. The emergence of AI coding assistants that can write files, execute shell commands, and manage system configurations has further raised the stakes: prompt injection attacks that might previously have produced nuisance-level outputs now represent viable remote code execution primitives with real-world CVEs documenting the risk (CVE-2025-53773 [3], CVE-2025-54135 [8], CVE-2025-54136 [8]).

Security teams and platform operators must recognize that the AI developer tooling stack—comprising IDE extensions, LLM-integrated coding assistants, local model servers, MCP infrastructure, and the package ecosystems these tools recommend—now constitutes a high-value, rapidly expanding attack surface requiring dedicated security controls.

Background

Developer Trust as the Fundamental Attack Surface

The AI developer tool supply chain inherits all the vulnerabilities of traditional software supply chains and adds several new ones. A fundamental challenge is trust: developers must necessarily trust the tools that accelerate their work, and the rapid adoption of AI coding assistants has created an environment in which, by many accounts, trust is being extended faster than security vetting processes can keep pace.

The IDE plugin ecosystem is the most visible expression of this problem. The Visual Studio Code Marketplace and its alternative Open VSX registry together host tens of thousands of extensions, many from individual contributors with minimal review. As AI coding assistants gained mainstream adoption, malicious actors recognized that an extension presenting as an AI productivity tool could attract large install counts and relatively permissive user trust. By January 2026, researchers documented that malicious VS Code extension detections had grown from 27 incidents in 2024 to 105 in the first 10 months of 2025—a roughly fourfold increase in a single year [1].

Parallel to the IDE extension threat is the problem of package ecosystem abuse. Developers installing AI libraries or interfacing with AI APIs rely on PyPI, npm, and similar registries. When AI brand names—Claude, ChatGPT, Hugging Face—appear in package names, they carry an implicit credibility that threat actors have learned to exploit. This trust exploitation extends in a novel direction through the emergence of LLM package hallucinations: when developers ask AI coding assistants to suggest installation commands, the model may recommend packages that do not exist, and attackers can register those hallucinated names with malicious payloads before the developer notices.

The MCP Ecosystem and a New Trust Boundary

The Model Context Protocol (MCP), an open standard introduced by Anthropic for connecting AI agents to external data sources and tools, introduced a new trust boundary into the developer environment during 2024–2025. As AI coding assistants began supporting MCP server integrations, each external MCP server a developer connects to becomes a potential attack vector: a malicious or compromised server can inject instructions into the AI agent's context, influence its code generation, and in the worst cases trigger execution of attacker-controlled commands on the developer's machine. One security researcher's tracking counted approximately 30 CVEs filed against MCP implementations within the first 60 days of the protocol's widespread adoption in early 2025 [13].

Security Analysis

Package Ecosystem Compromises: AI Brands as Lures

The most straightforward attack vector against AI developers exploits the package installation workflow. In November 2023, an attacker uploaded two PyPI packages—`gptplus`, which claimed to provide access to GPT-4 Turbo, and `claudeai-eng`, which impersonated Anthropic's Claude AI—to the Python Package Index. Kaspersky's Global Research and Analysis Team discovered the packages

approximately one year later, by which time they had accumulated over 1,700 downloads across 30 countries [2]. Both packages delivered JarkaStealer, a Java-based infostealer distributed as malware-as-a-service via Telegram that exfiltrates browser credentials, session tokens from Telegram, Discord, and Steam, and general system information.

The attack does not require sophisticated deception. The packages functioned superficially as described while executing the malicious payload, suggesting that association with recognized AI brand names was sufficient, in many cases, to drive organic installation by developers who did not verify the publisher before installing.

Slopsquatting: When AI Recommendations Become Attack Vectors

A structurally more novel threat emerged from the intersection of LLM code suggestions and package registries. Researchers analyzing 576,000 LLM-generated code samples found that 19.7% of samples—approximately 113,000 instances—referenced non-existent packages [4]. For open-source models, the hallucination rate reached 21.7% on average, while proprietary models hallucinated at approximately 5.2%. Critically, these hallucinations are not random noise: 58% of hallucinated package names reappeared consistently when the same prompt was issued ten times, indicating that specific model-topic combinations produce reliable, repeatable false recommendations [4]. This repeatability is precisely what makes slopsquatting a viable attack: an attacker who identifies which package names a popular model tends to hallucinate for common developer queries can register those names and wait for organic installs.

The threat is not merely theoretical. A researcher at Lasso Security published a benign Python package under the name `huggingface-cli`—a name multiple LLMs repeatedly hallucinated in response to Hugging Face installation queries—on January 28, 2024. The package accumulated more than 15,000 downloads in the months following publication [5]. The Alibaba GraphTranslator project's official installation documentation included a `pip install huggingface-cli` command, having incorporated an AI recommendation without verification [5]. Had the registered package been malicious rather than a research demonstration, the consequences—arbitrary code execution in the CI/CD pipeline and development environments of any project following that documentation—would have been severe and difficult to contain.

CI/CD Pipeline and Repository Compromise

Supply chain attacks are not limited to the package registry layer. The December 2024 compromise of the Ultralytics YOLO library—a computer vision AI project with a large cumulative PyPI download base—demonstrated that GitHub Actions CI/CD pipelines represent an attractive target for supply chain

attackers given the scale of downstream impact. Attackers exploited a misconfiguration in the `pull_request_target` GitHub Actions trigger and a cache poisoning technique to inject cryptomining payloads (XMRig) into the build pipeline without direct access to the repository's secrets. The attack subsequently advanced to a second phase in which a previously compromised PyPI API token was used to directly publish four malicious versions [6]. Each remained available for between one and twelve hours before removal, meaning projects with unpinned version dependencies and automated update tooling could have fetched malicious builds into production environments before removal.

The cascading GitHub Actions attack of March 2025 further illustrates the interconnected nature of CI/CD risk. A compromise of the `reviewdog/action-setup` GitHub Action (CVE-2025-30154) served as a stepping stone to compromise `tj-actions/changed-files` (CVE-2025-30066), a utility used in over 23,000 repositories. The malicious versions dumped CI/CD runner memory to public workflow logs, exposing environment variables, secrets, and API tokens [7]. Palo Alto Unit 42 assessed that the campaign began as a targeted attack against Coinbase repositories before being broadened. CISA issued an advisory on March 18, 2025 [7].

RCE in AI Coding Assistants: Prompt Injection as Exploitation Primitive

The integration of tool-use capabilities into AI coding assistants effectively elevated prompt injection, previously considered primarily a data manipulation or social engineering concern in most deployment contexts, into a code-execution vulnerability category. Three CVEs from mid-2025 illustrate the pattern.

CVE-2025-54135 ("CurXecute") and CVE-2025-54136 ("MCPoison"), both in Cursor IDE prior to version 1.3, allowed a malicious MCP server processing untrusted external data—such as a GitHub issue tracker, web search results, or a customer support inbox—to inject instructions that would silently overwrite `~/.cursor/mcp.json` and execute attacker-controlled commands [8]. MCPoison added a persistence dimension: after an initial approval by the user, the altered MCP configuration would execute malicious commands automatically every time the project opened, with no further prompts. A related bypass, CVE-2025-59944, affected Cursor versions up to 1.6.23 and exploited the IDE's case-sensitive path protection to perform the same MCP configuration overwrite via crafted paths that bypassed confirmation dialogs [9].

CVE-2025-53773 in GitHub Copilot and Visual Studio extended the threat model further. Researcher Johann Rehberger demonstrated that malicious content embedded in source code comments, GitHub issues, or web pages retrieved by Copilot during an agentic task could instruct the assistant to write `"chat.tools.autoApprove": true` into the project's `.vscode/settings.json`, enabling "YOLO mode"—disabling all user confirmation requirements for tool execution [3]. Hidden Unicode characters made the injected instructions invisible during code review. Rehberger characterized

the vulnerability as "wormable" given its potential for repository-to-repository propagation: an infected repository could propagate the poisoned instruction to collaborators and downstream projects, creating AI-mediated botnet-style propagation [3].

Vulnerabilities in Local AI Infrastructure

Local model-serving platforms represent a distinct but related attack surface. Ollama, the open-source platform for running AI models locally, was found to have a critical path traversal vulnerability (CVE-2024-37032, "Probllama") in May 2024 that allowed unauthenticated remote code execution via the `/api/pull` endpoint [10]. In Docker deployments, Ollama runs as root and listens on `0.0.0.0` by default, configurations that, in combination, create a high-severity exposure. Wiz Research identified over 1,000 Ollama instances reachable from the public internet with no authentication [10]. Subsequent security advisories documented additional Ollama vulnerabilities in 2024 and 2025 addressing heap overflows, memory exhaustion, and token exposure.

MCP Infrastructure as an Attack Vector

The mcp-remote library—an OAuth proxy used to connect local AI clients including Claude Desktop and Cursor to remote MCP servers, featured in integration guides from Cloudflare, Hugging Face, and Auth0—carried a critical OS command injection vulnerability (CVE-2025-6514, CVSS 9.6) in which a malicious MCP server could supply a crafted `authorization_endpoint` URL containing shell metacharacters that mcp-remote passed directly to the system shell [11]. With over 437,000 downloads, the library's vulnerability represented a single exploit point capable of compromising developer machines across a wide swath of the MCP ecosystem, potentially enabling theft of API keys, cloud credentials, SSH keys, and local repository contents depending on the developer's configuration.

Anthropic's own MCP Inspector development tool contained a complementary critical vulnerability (CVE-2025-49596, CVSS 9.4): the Inspector's proxy server accepted unauthenticated connections by default and could spawn local processes. Combined with a CSRF vulnerability and the "0.0.0.0 Day" browser-level design flaw—in which browsers permit local services to be contacted by visiting a malicious website—an attacker could achieve remote code execution on a developer's machine simply by directing them to a malicious URL [12]. The vulnerability was fixed in MCP Inspector version 0.14.1 in June 2025.

Fake Installer Campaigns and Malvertising

Among the delivery mechanisms documented in 2025–2026, malvertising campaigns have achieved rapid reach by targeting high-intent developer search queries. By purchasing search engine advertisements that surface above legitimate installation pages when developers search for AI tool installation instructions, attackers intercept traffic at the moment of highest intent.

Push Security documented the "InstallFix" technique in February–March 2026, in which attackers created pixel-accurate clones of the official Claude Code CLI installation page and distributed them via Google Ads targeting searches for "Claude Code," "Claude Code install," and "Claude Code CLI" [14]. The cloned pages substituted the legitimate installation commands with instructions that, on macOS, executed base64-encoded commands downloading the Amatera Stealer—a 2025-era infostealer successor to ACR Stealer that targets browser credentials, session cookies, and cryptocurrency wallets [14].

A separate technique documented in 2025 involved weaponizing Claude.ai artifacts—AI-generated content hosted on Anthropic's own domain—as delivery vehicles. Moonlock Lab and AdGuard researchers found that threat actors created publicly accessible Claude artifacts containing ClickFix-style terminal instructions and allowed them to be indexed by Google Search for queries including "online DNS resolver" and "macOS CLI disk space analyzer" [15]. Content hosted on the legitimate `claude.ai` domain may have increased user trust and bypassed reputation-based filtering that would have flagged a newly-registered third-party domain. Researchers estimated over 15,000 views and more than 10,000 user interactions with malicious artifacts before takedown [15].

Storm-2460, a ransomware group associated with RansomExx operations, was observed using a modified open-source ChatGPT desktop application as a delivery vehicle for the PipeMagic modular backdoor in 2025 [16]. PipeMagic is a fully modular, in-memory framework using encrypted named pipes for command-and-control communications. The campaign combined the fake AI application with CVE-2025-29824, a Windows Common Log File System zero-day enabling privilege escalation to SYSTEM, to deploy ransomware against financial and real estate organizations in the United States, Europe, and the Middle East [16].

Malicious IDE Extensions: Scale and Persistence

Two VS Code extension incidents from late 2025 and early 2026 illustrate the scale achievable when attackers succeed in placing malicious extensions on official marketplaces. In June–July 2025, a threat actor published a malicious extension named "Solidity Language" to the Open VSX marketplace,

artificially inflating its install count to approximately 54,000 to game search rankings above the legitimate extension [17]. The extension delivered PureLogs Stealer and ScreenConnect, ultimately enabling the theft of $500,000 USD in cryptocurrency from at least one affected developer [17].

In January 2026, researchers documented that two VS Code extensions with a combined 1.5 million installs—marketed as Chinese-language AI coding assistants (`ChatGPT – 中文版` and `ChatGPT – ChatMoss (CodeMoss)`)—had covertly exfiltrated every file opened and every edit made by developers to servers in China for an extended period [1]. The attack used a triple-channel architecture: real-time file surveillance that Base64-encoded and transmitted files immediately upon opening, an on-demand remote exfiltration pathway allowing up to 50 files per attacker command, and device fingerprinting through embedded Chinese analytics SDKs [1]. The extensions remained available through the official VS Code Marketplace for the duration of their operation.

Recommendations

Immediate Actions

Security teams and development organizations should treat the AI developer tooling stack with the same rigor applied to production software dependencies. IDE extensions should be subject to organizational approval processes rather than individual developer discretion; extension allow-listing using publisher verification and automated scanning services provides a practical control. Developers should be explicitly instructed to verify package publishers before installing AI-branded packages, and to cross-reference AI-recommended package names against the official documentation of the project or API they are using before executing any installation command.

Any MCP server that a development team connects to should be treated as a privileged integration requiring security review equivalent to that applied to third-party APIs. The principle of least privilege applies directly: AI agents should not be granted tool permissions—especially file write and shell execution—unless those permissions are specifically required for the task, and even then should operate within defined scopes. Organizations should monitor `~/.cursor/mcp.json`, `.vscode/settings.json`, and equivalent AI configuration files for unauthorized modifications.

AI coding assistant configurations that enable automatic tool approval ("YOLO mode" in Copilot, equivalent settings in other tools) should be disabled at the organizational level and treated as a high-severity misconfiguration if found enabled in developer environments. Patching of Cursor IDE (version

1.3+ for CVE-2025-54135/54136, 1.7+ for CVE-2025-59944) and GitHub Copilot (August 2025 Patch Tuesday and Visual Studio 2022 17.14.12+ for CVE-2025-53773) should be treated as priority updates.

Short-Term Mitigations

Package dependency review processes should be extended to cover AI-generated code recommendations, not only direct developer decisions. When AI coding assistants suggest package installations, those recommendations should be validated against official documentation and verified against the package registry's publisher information. Organizations using Ollama or similar local model servers should ensure those services are not reachable from the public internet, are protected by authentication, and are not running with root or elevated privileges in default configurations.

CI/CD pipeline configurations should be audited for `pull_request_target` trigger usage and cache configurations that could enable the attack pattern observed in the Ultralytics compromise. Third-party GitHub Actions should be pinned to specific commit SHAs rather than mutable version tags, a control that would have substantially limited the blast radius of the tj-actions/changed-files attack. Secret scanning and token rotation should be performed for any pipelines that may have been exposed during the March 2025 campaign window.

Strategic Considerations

The slopsquatting threat warrants a structural response at both the organizational and ecosystem levels. Organizations should evaluate whether the AI coding assistants they deploy have package recommendation hallucination rates that are acceptable given their risk tolerance, and should provide developers with guidance on treating AI-generated package names as unverified until confirmed. Package registry operators should consider accelerating programs to reserve or flag high-risk package name patterns associated with known hallucinations from major models.

The convergence of AI agentic capabilities—file system access, shell execution, network requests—with prompt injection vulnerabilities implies that AI coding assistants are likely to remain targets for prompt injection-based RCE while they act on untrusted content from the environment. Organizations should engage with their AI tool vendors on their prompt injection mitigations and require disclosure of the scope of tool permissions granted to AI agents in their products.

CSA Resource Alignment

This research note addresses threats that map directly to several CSA frameworks and publications.

The CSA **AI Controls Matrix (AICM) v1.0** establishes the AI supply chain security domain as one of its 18 control domains and defines controls for AI software component integrity, third-party AI library risk management, and model provenance verification [18][19]. The supply chain attack patterns documented in this note—compromised PyPI packages, malicious IDE extensions, CI/CD pipeline poisoning—represent precisely the threat scenarios AICM's supply chain controls are designed to address.

The CSA **Software Transparency: Securing the Digital Supply Chain** publication provides foundational guidance on CI/CD pipeline security, Software Bill of Materials (SBOM) adoption, and open-source software risk management that applies directly to the AI developer tooling context [20]. SBOM generation for AI development environments—capturing IDE extensions, AI library dependencies, and model-serving infrastructure—is an area where existing supply chain security practice can be extended.

The CSA **Cloud Controls Matrix (CCM)** supply chain management domain (STA-09, STA-10) covers third-party software integrity and supplier assurance requirements applicable to AI tool procurement and deployment decisions.

The **MAESTRO** threat modeling framework for agentic AI systems provides a structured approach to analyzing the threat categories described in this note. MAESTRO Layer 2 (data operations) covers threats arising from untrusted data processed by AI agents—directly applicable to the prompt injection RCE vulnerabilities in Cursor and GitHub Copilot documented here. MAESTRO Layer 5 (agent trust and orchestration) covers the MCP server trust model, addressing how compromised or malicious external services can subvert agent behavior.

The **CSA State of AI Security and Governance Report 2025** identifies third-party AI component risk as a top-three security concern among surveyed organizations [21], providing quantitative context for the threat scenarios this note documents.

References

[1] The Hacker News, "Malicious VS Code AI Extensions with 1.5M Installs Steal Developer Data," January 2026. https://thehackernews.com/2026/01/malicious-vs-code-ai-extensions-with-15.html

[2] Kaspersky GReAT, "Kaspersky Uncovers Year-Long PyPI Supply Chain Attack Using AI Chatbot Tools as Lure," November 2024. https://www.kaspersky.com/about/press-releases/kaspersky-uncovers-year-long-pypi-supply-chain-attack-using-ai-chatbot-tools-as-lure

[3] Embrace the Red (Johann Rehberger), "GitHub Copilot Remote Code Execution via Prompt Injection," 2025. https://embracethered.com/blog/posts/2025/github-copilot-remote-code-execution-via-prompt-injection/

[4] Spracklen, J. et al., "We Have a Package for You! A Comprehensive Analysis of Package Hallucinations by Code Generating LLMs," arXiv:2406.10279, 2024 (v3: March 2025). https://arxiv.org/abs/2406.10279

[5] The Register, "AI Bots Hallucinate Software Packages," March 28, 2024. https://www.theregister.com/2024/03/28/ai_bots_hallucinate_software_packages/

[6] PyPI Blog, "Ultralytics Supply Chain Attack Analysis," December 11, 2024. https://blog.pypi.org/posts/2024-12-11-ultralytics-attack-analysis/

[7] CISA, "Supply Chain Compromise of Third-Party tj-actions/changed-files (CVE-2025-30066) and reviewdog/action-setup (CVE-2025-30154)," March 18, 2025. https://www.cisa.gov/news-events/alerts/2025/03/18/supply-chain-compromise-third-party-tj-actionschanged-files-cve-2025-30066-and-reviewdogaction

[8] Cato Networks (formerly AIM Security), "CurXecute: When Public Prompts Turn Into Local Shells — RCE in Cursor via MCP Auto-Start," July 2025. https://www.catonetworks.com/blog/curxecute-rce/; Check Point Research, "Cursor IDE: Persistent Code Execution via MCP Trust Bypass," July 2025. https://research.checkpoint.com/2025/cursor-vulnerability-mcpoison/

[9] Lakera (Brett Gustafson), "Cursor Vulnerability CVE-2025-59944," 2025. https://www.lakera.ai/blog/cursor-vulnerability-cve-2025-59944

[10] Wiz Research, "Probllama: Critical RCE Vulnerability in Ollama (CVE-2024-37032)," June 2024. https://www.wiz.io/blog/probllama-ollama-vulnerability-cve-2024-37032

[11] JFrog Security Research, "CVE-2025-6514: Critical mcp-remote RCE Vulnerability," 2025. https://jfrog.com/blog/2025-6514-critical-mcp-remote-rce-vulnerability/

[12] Oligo Security, "Critical RCE Vulnerability in Anthropic MCP Inspector (CVE-2025-49596)," June–July 2025. https://www.oligo.security/blog/critical-rce-vulnerability-in-anthropic-mcp-inspector-cve-2025-49596

[13] Equixly, "MCP Server: The New Security Nightmare," March 29, 2025. https://equixly.com/blog/2025/03/29/mcp-server-new-security-nightmare/

[14] Push Security, "InstallFix: Fake Claude Code Install Guides Push Infostealers," March 2026. https://pushsecurity.com/blog/installfix; BleepingComputer, "Fake Claude Code Install Guides Push Infostealers in InstallFix Attacks," 2026. https://www.bleepingcomputer.com/news/security/fake-claude-code-install-guides-push-infostealers-in-installfix-attacks/

[15] BleepingComputer, "Claude LLM Artifacts Abused to Push Mac Infostealers in ClickFix Attack," 2025. https://www.bleepingcomputer.com/news/security/claude-llm-artifacts-abused-to-push-mac-infostealers-in-clickfix-attack/

[16] Microsoft Security Blog, "Dissecting PipeMagic: Inside the Architecture of a Modular Backdoor Framework," August 18, 2025. https://www.microsoft.com/en-us/security/blog/2025/08/18/dissecting-pipemagic-inside-the-architecture-of-a-modular-backdoor-framework/

[17] BleepingComputer, "Malicious VSCode Extension in Cursor IDE Led to $500K Crypto Theft," July 2025. https://www.bleepingcomputer.com/news/security/malicious-vscode-extension-in-cursor-ide-led-to-500k-crypto-theft/

[18] Cloud Security Alliance, "AICM Implementation and Auditing Guidelines," 2024–2025. https://cloudsecurityalliance.org/artifacts/aicm-implementation-auditing-guidelines-frameworks

[19] Cloud Security Alliance, "Introductory Guidance to the AI Controls Matrix (AICM)," 2024–2025. https://cloudsecurityalliance.org/artifacts/introductory-guidance-to-aicm

[20] Cloud Security Alliance, "Software Transparency: Securing the Digital Supply Chain," 2022 (updated November 2025). https://cloudsecurityalliance.org/artifacts/software-transparency-digital-supply

[21] Cloud Security Alliance, "The State of AI Security and Governance Report 2025," 2025. https://cloudsecurityalliance.org/artifacts/the-state-of-ai-security-and-governance-report-2025

Further Reading

The following sources provide additional context on related threat actors and supply chain incidents not covered in depth in this note:

[22] IBM Security, "2026 X-Force Threat Intelligence Index," February 25, 2026. https://newsroom.ibm.com/2026-02-25-ibm-2026-x-force-threat-intelligence-index-ai-driven-attacks-are-escalating-as-basic-security-gaps-leave-enterprises-exposed

[23] SentinelOne Labs, "NullBulge Threat Actor Masquerades as Hacktivist Group Rebelling Against AI," 2024. https://www.sentinelone.com/labs/nullbulge-threat-actor-masquerades-as-hacktivist-group-rebelling-against-ai/

[24] Aikido Security, "Supply Chain Attack on the React Native Aria / GlueStack Ecosystem," June 2025. https://www.aikido.dev/blog/supply-chain-attack-on-react-native-aria-ecosystem

[25] Pillar Security, "New Vulnerability in GitHub Copilot and Cursor: How Hackers Can Weaponize Code Agents Through Compromised Rule Files," March 18, 2025. https://www.pillar.security/blog/new-vulnerability-in-github-copilot-and-cursor-how-hackers-can-weaponize-code-agents

---

This research note was produced by the Cloud Security Alliance AI Safety Initiative as a point-in-time analysis based on publicly available information as of March 8, 2026. It is intended to inform security professionals and development teams about emerging threats and does not constitute legal, compliance, or audit guidance.