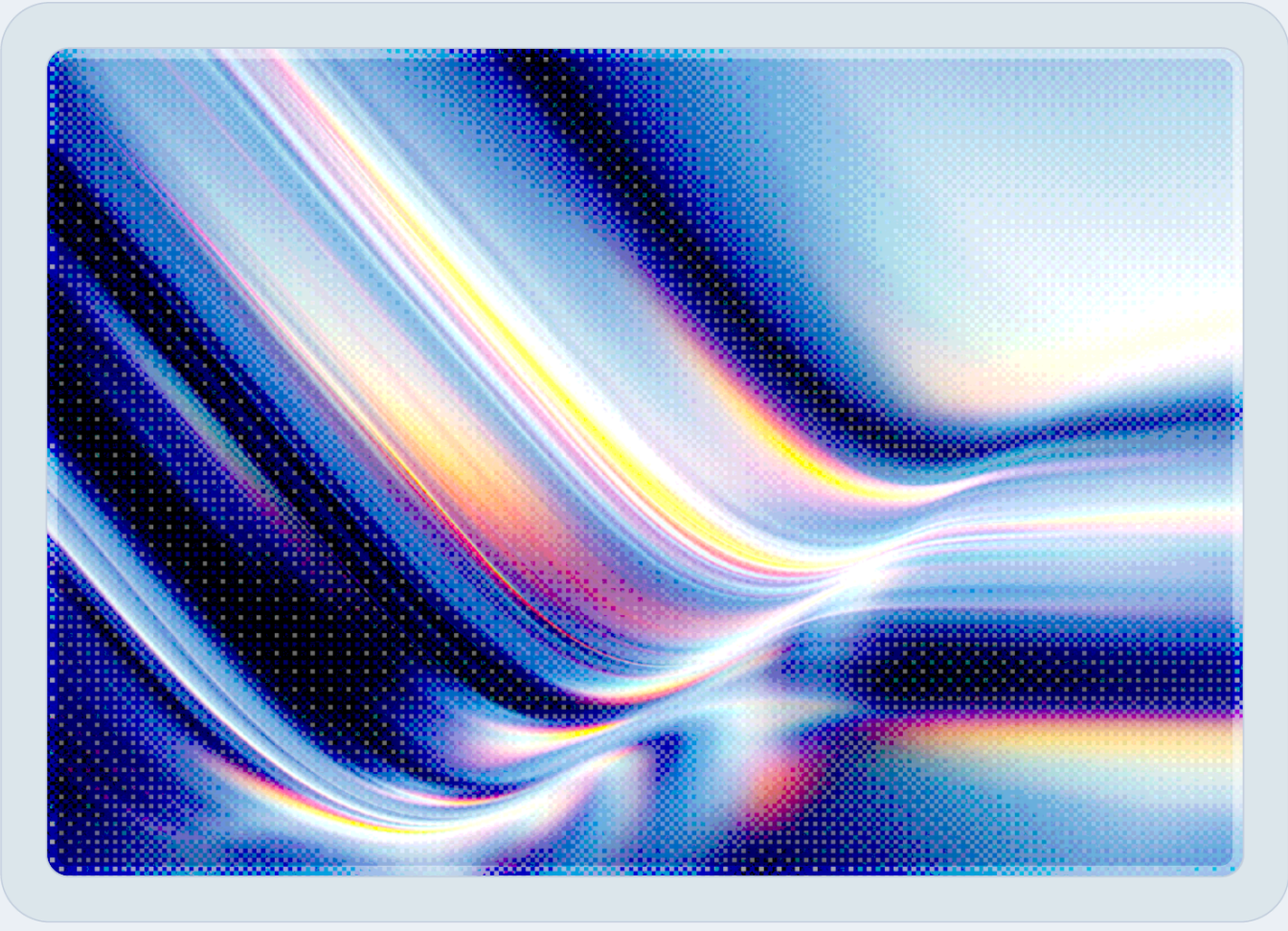
CSAI Foundation | Cloud Security Alliance

Poisoned AI IDE Plugins: Developer API Key Theft

Coordinated JetBrains Marketplace Campaign Targets OpenAI, DeepSeek, and SiliconFlow Credentials

2026-06-18

AI-assisted Rapid Research



CSAI | cloud security alliance®

Key Takeaways

A coordinated malware campaign distributed at least fifteen AI coding assistant plugins through the JetBrains Marketplace, collectively accumulating nearly 70,000 installations while silently exfiltrating developer API keys for OpenAI, DeepSeek, and SiliconFlow services [1][2][4][5][6][8]. The campaign operated for approximately eight months—from late October 2025 through June 2026—before Aikido Security researchers identified the shared malicious codebase and notified JetBrains [1].

The attack surface is the IDE plugin ecosystem itself. Developers integrating AI coding tools routinely enter high-value credentials directly into IDE settings panels, creating a credential-harvesting opportunity that requires no phishing lure, no social engineering, and no network exploitation—only a convincing plugin listing [2]. Credential theft fired the moment a user clicked "Apply," with no user notification, no consent prompt, and no visible behavioral anomaly during normal plugin use [1].

JetBrains removed all fifteen plugins, permanently banned the seven associated publisher accounts, and triggered a remote kill-switch to disable the extensions in users' IDEs [3]. However, developers who entered API keys into any of the affected plugins prior to removal should assume those credentials are compromised and revoke them immediately.

The incident demonstrates that AI-era supply chain risk has extended into developer tooling in a new, high-frequency way: as AI coding assistants become standard IDE components, the credential-stuffing attack surface expands with every new plugin category.

Background

JetBrains IntelliJ IDEA and its family of language-specific IDEs—PyCharm, WebStorm, GoLand, and others—are among the most widely used integrated development environments in professional software engineering. The JetBrains Marketplace, which hosts third-party plugins for these tools, offers a large library of extensions developed by independent vendors. Plugin installation is a routine developer activity, and the marketplace's approval process does not appear to have extended to deep behavioral analysis of plugin runtime conduct [3], a gap that allowed plugins to reach users while containing credential-exfiltration logic that only activates during settings configuration.

The rapid adoption of AI coding assistants has created fertile ground for adversarial plugin campaigns. Developers in 2025 and 2026 have come to expect AI-powered code completion, commit message generation, code review, and bug-finding features to arrive via IDE plugins—and they routinely configure these plugins by entering API keys provided by AI service vendors. Keys for services such as OpenAI, Anthropic Claude, DeepSeek, and SiliconFlow carry direct financial value: they authorize inference calls billed to the key owner's account, and compromised keys can be used for unauthorized model access until revoked.

Aikido Security, a software supply chain security firm, identified the campaign through code analysis of suspicious plugin listings that had appeared on the JetBrains Marketplace beginning in late October 2025 [1]. The company's researchers found that fifteen distinct plugins—published under seven vendor accounts using names such as CodePilot, StackSmith, CodeCrafter, CodeWeaver, JetCode, DailyCode, and ZenCoder—shared a near-identical malicious codebase that had been renamed and repackaged across listings [1][2]. Publicly reported download totals for the campaign are approximately 70,000 installations, though Aikido noted these metrics may be artificially inflated [2]. The two most prominently downloaded plugins were "DeepSeek AI Assist" (approximately 27,727 downloads) and "CodeGPT AI Assistant" (approximately 25,571 downloads) [1].

The campaign's longevity—new plugin variants continued to appear as recently as June 10, 2026—indicates that the marketplace's review pipeline did not flag the credential-exfiltration logic during routine submission checks [1][3].

Security Analysis

Attack Mechanism

The malicious plugins functioned as advertised. Chat interfaces, commit message generators, code reviewers, and unit test creators all operated normally, providing developers with no functional reason to suspect misbehavior. The credential theft was embedded in the settings save path: when a developer entered an API key into the plugin's configuration panel and clicked "Apply," the plugin's `save()` method intercepted the credential before storing it locally and transmitted it externally in the same call [1].

The exfiltration logic specifically targeted values matching the `sk-` prefix pattern with 51-character length—the characteristic format of OpenAI API keys—as well as credential formats for DeepSeek and SiliconFlow [1]. The code deduplicated keys before transmission, suggesting the campaign was designed to avoid redundant data collection.

Credentials were sent in plaintext over unencrypted HTTP to a hardcoded command-and-control server at IP address `39.107.60[.]51`, using the endpoint `/api/software/key` with a static bearer token (`F48D2AA7CF341F782C1D`) for authentication to the attacker-controlled infrastructure [1][2]. The plugins installed a custom `X509TrustManager` that effectively disabled standard certificate validation [3] — a modification consistent with suppressing TLS warnings that might otherwise alert users or network monitoring tools. This combination—plaintext transmission, raw IP address in code, and deliberate TLS weakening—represents multiple layers of operational security violation that, in retrospect, could have been detected through automated static analysis of plugin bytecode.

Monetization Model

Aikido researchers identified a secondary revenue mechanism that distinguishes this campaign from straightforward credential theft. The affected plugins included a "donation wall" that invited users to pay a small fee in exchange for access to AI model inference without requiring their own API key [1][2]. Aikido's analysis suggests the attacker-controlled server dispensed credentials to paying users from the pool of keys harvested from free-tier victims—a model in which the genuine key owners incurred the inference costs while the campaign operator collected both the stolen credentials and the subscription fees [2].

This monetization architecture transforms the attack from a simple data theft incident into an ongoing credential resale operation. The attacker's server functioned simultaneously as an exfiltration endpoint and a credential distribution service. This is an inference drawn from observed behavior rather than a confirmed operator statement, but the technical evidence—a server endpoint that both receives stolen keys and issues keys on payment—is consistent with this interpretation.

Indicators of Compromise

Organizations wishing to assess exposure should check for the following:

Indicator	Type	Value
C2 server IP	Network	`39.107.60[.]51`
Exfiltration endpoint	URL	`hxxp://39.107.60[.]51/api/software/key`
Static auth token	String	`F48D2AA7CF341F782C1D`

Indicator	Type	Value
Plugin IDs (partial)	String	`ord.cp.code.ai.kit`, `com.my.code.tools`, `org.sm.yms.toolkit`, `com.json.simple.kit`, and eleven additional variants
Publisher accounts	Identity	CodePilot, StackSmith, CodeCrafter, CodeWeaver, JetCode, DailyCode, ZenCoder

Traffic destined for `39.107.60[.]51` from developer workstations, particularly on port 80 from IDE processes, should be treated as indicative of a compromised installation. Security operations teams should inspect endpoint telemetry for any JetBrains process initiating outbound HTTP connections to raw IP addresses without a registered domain.

Broader Supply Chain Implications

This incident is an instance of a well-documented supply chain attack pattern applied to a new and rapidly growing category: AI tooling extensions in developer IDEs. Several structural features of the JetBrains Marketplace created the conditions for the campaign to persist for eight months.

First, the marketplace's approval process does not appear to have extended to runtime behavioral analysis [3], allowing plugins to pass review while containing credential exfiltration logic that only activates during settings configuration. Second, the use of AI-themed naming conventions (DeepSeek, CodeGPT) exploited developer familiarity and trust in widely recognized AI brand names, a pattern that does not require any technical vulnerability to exploit. Third, the IDE plugin model grants extensions broad access to the local environment, including the ability to read configuration inputs, make outbound network requests, and install custom security handlers—capabilities that are necessary for legitimate plugins but equally useful for malicious ones.

The campaign's eight-month lifespan suggests a gap in marketplace monitoring: post-publication behavioral review of already-approved plugins appears to have been insufficient to catch new variants as they were submitted or to detect the shared codebase across seven vendor accounts.

Recommendations

Immediate Actions

Any developer who installed one of the fifteen identified plugins should revoke all API keys that were entered into the plugin's settings panel immediately, regardless of whether anomalous account activity is yet visible. API providers typically allow key revocation and reissuance through their respective dashboards—OpenAI via platform.openai.com, DeepSeek via platform.deepseek.com, and SiliconFlow through their developer portal. Revocation should precede any investigation into downstream effects, as the window for unauthorized use begins at the moment of exfiltration.

Organizations should block the C2 IP address `39.107.60[.]51` at their network perimeter and endpoint firewall layers. Security operations teams should query endpoint detection logs for any prior outbound connections to this address originating from IDE processes. Developers who observe anomalous spending in their AI provider billing dashboards after using any of the named plugins should treat this as confirmation of key compromise.

Any JetBrains IDE installation that contained one of the fifteen plugins should be inspected to verify the plugin has been fully removed. JetBrains' remote kill-switch disables the plugin on next IDE launch, but the kill-switch does not revoke already-transmitted credentials [3].

Short-Term Mitigations

Teams that deploy AI coding assistants at scale should implement centralized API key management rather than allowing individual developers to configure per-developer keys in IDE settings. Distributing keys through a secrets management system—with per-developer or per-workstation key scoping, usage monitoring, and automatic rotation—limits the impact of a single-key compromise and provides visibility into anomalous usage before costs accumulate.

Organizations should extend their software bill of materials (SBOM) practices to include IDE plugins as a tracked component category. Developer workstations may fall outside the scope of enterprise SBOM programs, yet they represent a significant and growing attack surface as AI tooling proliferates. Plugin inventories should be reviewed against marketplace security advisories on a regular cadence.

AI provider API keys should be scoped to the minimum necessary permissions. Where providers support usage caps, rate limits, or IP-based access restrictions, these controls should be applied to developer keys. This does not prevent exfiltration but limits the attacker's ability to monetize stolen keys before they are detected and revoked.

Strategic Considerations

The JetBrains incident reflects a structural challenge for all plugin and extension marketplaces hosting AI tooling: the credential-entry UI pattern—where users type high-value secrets directly into extension settings—is both the expected user experience and an inherent risk concentrator. Marketplaces should evaluate whether plugin-level credential storage can be replaced with centralized, audited secrets vaults that extensions access through a mediated API rather than direct user input.

JetBrains has announced enhanced automated screening to detect unencrypted HTTP endpoints, hardcoded IP addresses, and custom TLS weakening patterns in submitted plugin code [3]. The company has also recommended that developers consider the Agent Client Protocol (ACP) as a structured communication standard between IDEs and AI agents, which JetBrains contends reduces the surface area for credential-handling abuse compared to ad hoc plugin configuration [3]. These are meaningful responses, but the broader challenge of detecting sophisticated credential-exfiltration logic in a high-volume plugin submission environment will require ongoing investment in behavioral analysis tooling and post-publication monitoring.

For organizations operating software development teams, this incident is a prompt to treat developer workstations as part of the enterprise threat model. The tools developers use daily—IDEs, package managers, CLI utilities—appear to be increasingly targeted as entry points for supply chain compromise, and the emergence of AI coding assistants as a high-adoption, credential-dependent category elevates this risk further.

CSA Resource Alignment

This incident maps directly to threat categories addressed within CSA's Agentic AI Threat Modeling Framework (MAESTRO), which structures security analysis across seven layers of AI system architecture [7]. At the **Agent Ecosystem layer**, MAESTRO explicitly identifies marketplace manipulation and tool misuse as threat vectors: malicious plugins in an IDE marketplace represent precisely this class of risk, where the trust model of a curated distribution channel is exploited to deliver adversarial tooling at scale. Organizations building or deploying AI coding assistants should apply MAESTRO's ecosystem layer threat catalog to their plugin vetting processes.

The CSA Cloud Controls Matrix (CCM) addresses the supply chain and third-party risk controls that are directly implicated here. CCM control domain **Supply Chain Management, Transparency, and Accountability (STA)** covers third-party component vetting, software provenance tracking, and the obligations of organizations to assess risk in external dependencies—categories that extend logically to

IDE plugins consumed by development teams. The broader AI Controls Matrix (AICM) extends CCM with additional controls for AI-specific deployments, including AI tool governance and credential management in AI-integrated workflows.

CSA's **Zero Trust guidance** is relevant to the credential management failure mode this incident exploits. A Zero Trust approach to developer API key management would enforce that credentials are never stored in plugin configuration files or transmitted by plugin code directly; instead, they would be issued through a mediated identity layer with continuous verification and usage anomaly detection. The principle of least-privilege access—issue keys scoped to specific models and rate limits, not open-ended access—reduces the blast radius of any single credential compromise.

CSA's **Software Transparency: Securing the Digital Supply Chain** guidance addresses the SBOM and provenance tracking practices that are applicable here. Extending SBOM programs to IDE plugin inventories, and requiring that development teams track which AI tooling extensions are installed across the developer fleet, aligns with this guidance and creates the visibility necessary to respond quickly when marketplace security advisories are issued.

References

[1] Aikido Security. "Multiple JetBrains IDE Plugins Caught Stealing AI Keys." Aikido Security Blog, June 2026.

[2] Lawrence Abrams. "Malicious JetBrains Marketplace Plugins Steal AI API Keys from Developers." BleepingComputer, June 16, 2026.

[3] JetBrains. "Marketplace Ecosystem Security Update: Addressing Malicious Third-Party AI Plugins." JetBrains Blog, June 16, 2026.

[4] The Hacker News. "Malicious JetBrains Plugins Steal AI API Keys as Chrome Extensions Capture Chat bot Chats." The Hacker News, June 2026.

[5] Infosecurity Magazine. "Fifteen JetBrains Marketplace Plugins Steal API Keys." Infosecurity Magazine, June 2026.

[6] GBHackers. "JetBrains Plugin Security Alert: 70,000+ Installs Linked to AI Key Theft." GBHackers, June 2026.

[7] Cloud Security Alliance. "Agentic AI Threat Modeling Framework: MAESTRO." Cloud Security Alliance, February 6, 2025.

[8] Developer Tech. "JetBrains Marketplace Malware Exposes Developer API Keys." Developer Tech, June 2026.

