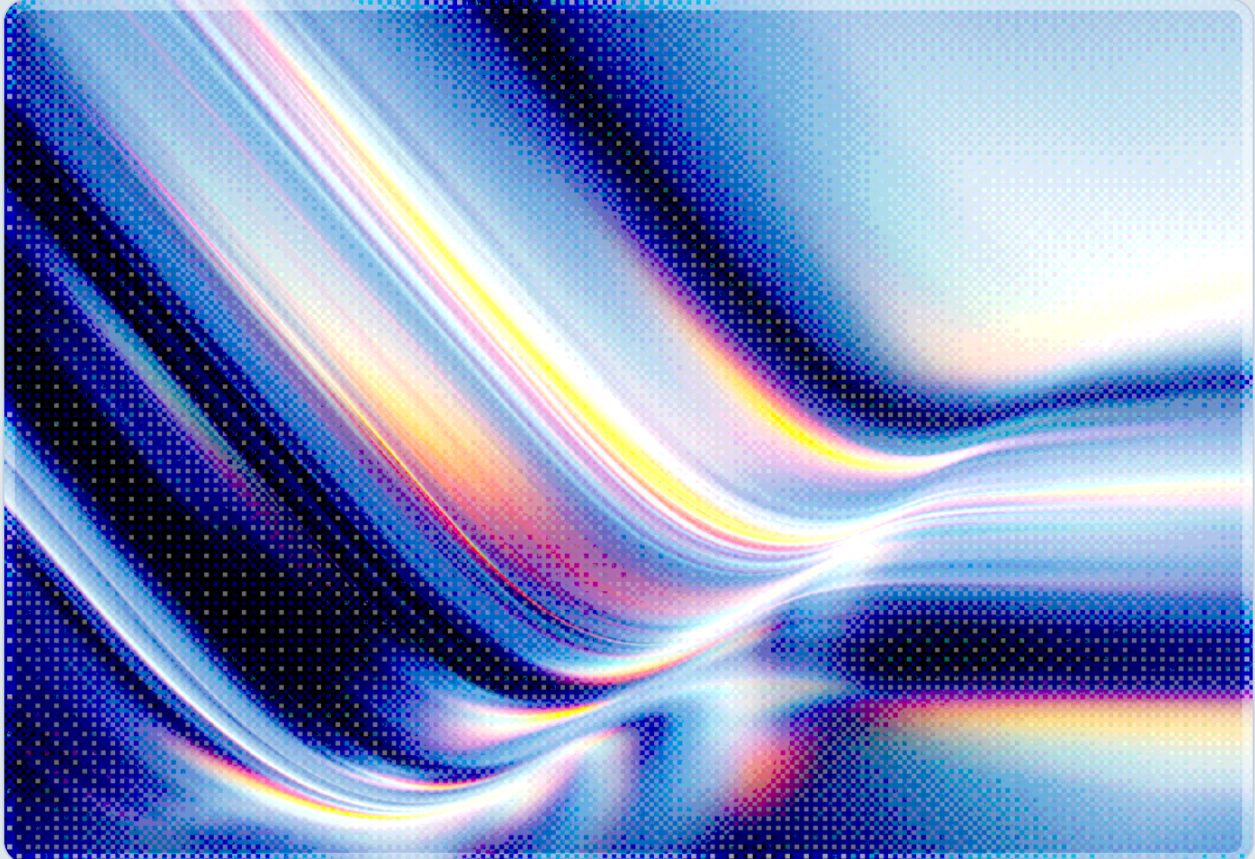


Harvesting Keys for Distillation: Attacks on AI Credential Platforms

How attackers target the platforms and pipelines that hold AI provider credentials, and what to do about it

2026-10-27

 AI-assisted Rapid Research



CSAI

cloud
CSA security
alliance®

© 2026 Cloud Security Alliance. Some rights reserved.

You may download, store, display, view, print, redistribute, and link to this document in its original, unmodified form, provided that attribution to the Cloud Security Alliance is maintained and all trademark and copyright notices remain intact.

This document may not be modified or altered. You may quote portions of the document as permitted by the Fair Use provisions of the United States Copyright Act, provided that attribution is given to the Cloud Security Alliance.

This document may be shared on professional and social media platforms in its original form with attribution.

This document was generated with AI assistance and has not undergone official CSA review and approval processes.

Key Takeaways

AI provider API keys have become a commodity of their own, and the people buying them now include unauthorized AI labs as well as fraudsters looking for free compute. Anthropic's September 2026 threat intelligence report describes proxy services ("transfer stations") that give unauthorized labs access to frontier models. The report says these services build account networks from "false identities, fake or stolen credit cards, and stolen API keys" and often use "stolen API credentials belonging to legitimate companies or individuals" [1]. Demand from distillation operators is now one of several forces driving key harvesting, alongside resale and cost-shifting fraud, and an organization's AI keys are what gets harvested.

Attackers still collect keys one at a time from developer laptops and public repositories, but recent incidents show them also going after the services that hold many customers' keys in one place. In October 2026, OpenCode said an intruder had reached a legacy system behind its hosted Console product. The intrusion may have exposed OpenAI, Anthropic, and Google keys that 1.4% of Console users had stored, and OpenCode described it as "related to a large-scale distillation attack" [3][4]. OpenCode has not named the attackers, and observers have questioned the distillation label [3][16]. In a separate incident earlier the same month, Fireworks AI reported that someone used internal Fireworks credentials to capture environment variables, including stored API keys and tokens, from evaluation jobs in a limited number of customer accounts [6]. Fireworks has not linked its incident to distillation.

Deleting a stored key does not revoke it. OpenCode's and Fireworks' guidance both made this point. A third-party key captured from a platform stays valid until the provider that issued it revokes it, and a platform's own budget controls cannot cap usage that bypasses the platform [3][6].

Anthropic describes the tradecraft as industrialized. It reports that attackers find exposed tokens through automated mining of application binaries, code repositories, container images, metadata endpoints, open storage, and victim-deployed AI agents. They then validate each find and grade it for resale value before using it [1]. One stolen key was used for roughly three weeks to attack other organizations [1].

Organizations should treat every AI provider key as a production bearer credential. That means an inventory of every third-party platform that holds one, a separate key for each tool and integration, spend limits enforced at the provider, and a revocation runbook that starts at the issuing provider.

Background

Why Distillation Creates Demand for Stolen Keys

Distillation is a legitimate training technique. A large "teacher" model generates responses, and those responses are used to train a smaller "student" model. Anthropic defines *illicit* distillation as "an industrial-scale, covert campaign to extract a model's capabilities and replicate them in another model without authorization" [1]. In February 2026, Anthropic attributed campaigns to three China-based labs. It said the campaigns ran through roughly 24,000 fraudulent accounts and produced more than 16 million exchanges with Claude [2]. In its September report, Anthropic said it had since disrupted distillation attacks from seven China-based labs. Anthropic attributes more than 151 million exchanges between May and July 2026 to a campaign it links to Alibaba [1][9]. Anthropic also notes that OpenAI and Google have described similar activity against their own models [1].

These campaigns need far more access than a lab could get through its own accounts, particularly when geographic restrictions and account verification stand in the way. Anthropic says the proxy services meeting this demand mix fraudulent accounts with stolen credentials from legitimate customers. Anthropic also says unauthorized labs buy transcripts of other users' exchanges from proxy operators, who "often save exchanges between users and US models without the knowledge or consent of those users" [1]. A stolen key therefore has two kinds of value. It provides compute that someone else pays for, and it provides access that does not trace back to the lab doing the distilling. In CSA's assessment, when the key belongs to an established enterprise account, it may also come with higher rate limits and more trust from abuse detection than a freshly created fraudulent account.

From LLMjacking to a Criminal AI Supply Chain

Abuse of stolen AI credentials is not new. Sysdig's Threat Research Team coined "LLMjacking" in May 2024 after attackers exploited a vulnerable Laravel deployment (CVE-2021-3129), took the AWS credentials stored on it, and used them to reach hosted models across several providers, apparently in order to resell the access [7]. Sysdig estimated the potential consumption costs to the victim, if the attacker maximized quota use, at more than \$46,000 per day [7]. CSA's own research has followed how the technique has changed since then. In the most recent variant, documented in October 2026, attackers take over a Google Cloud billing account outright rather than stealing an individual key [12].

What has changed in 2026 is how organized and specialized the market has become. Anthropic describes a "criminal AI supply chain" in which compromised API keys, session tokens, and devices are, for multiple criminal groups, "the sole objective." It says brokers then feed these into "fraudulent AI reseller networks

that rotate in new stolen API keys and session tokens until they exhaust their usage" [1]. One group ran websites that offered discounted access to frontier models and distributed client applications "spoofing as popular AI harnesses including Claude Code" that were in fact credential harvesters, collecting all of the victim's credentials and authenticated session tokens [1]. Distillation operators, cost-shifting fraudsters, and intrusion crews all draw on overlapping markets for stolen credentials.

The supply of exposed keys keeps growing. GitGuardian reports that it detected 1,275,105 leaked secrets tied to AI services on public GitHub in 2025, an 81% increase over 2024 [8]. It also reports that eight of the ten fastest-growing categories of leaked secrets were AI-related [8]. GitGuardian found 24,008 unique secrets in MCP-related configuration files, of which 2,117 were verified as valid. It also reports that 64% of secrets confirmed valid in 2022 could still be exploited four years later [8].

Security Analysis

The Shift to Concentration Points

Mining public repositories and infecting individual developer machines yields keys one at a time. A platform that stores keys on behalf of many customers can yield thousands of keys from one intrusion, which makes it the more efficient target. Several recent incidents fit this pattern, and the table below compares them. Not every row is the same kind of concentration point. The hosted OpenCode Console and Fireworks platforms are multi-tenant stores, while LiteLLM is a self-hosted gateway whose exposure is per deployment, and the JetBrains campaign shows a marketplace concentrating *distribution* rather than storage of keys.

Incident	Concentration point	Access vector (as reported)	Credentials at risk	Source
OpenCode Console (Oct 16–17, 2026)	Hosted gateway storing customers' provider keys	Intruder reached a legacy system being decommissioned; specifics undisclosed	OpenAI, Anthropic, and Google keys of 1.4% of Console users	[3][4] [5]
Fireworks AI (identified Oct 6, 2026)	Evaluation job environment variables	"Unauthorized use of internal Fireworks credentials"	API keys and tokens stored as secrets in	[6]

Incident	Concentration point	Access vector (as reported)	Credentials at risk	Source
			affected accounts	
LiteLLM proxy (2026, multiple CVEs)	Self-hosted AI gateway credential database (per deployment)	Poisoned PyPI packages, pre-auth SQL injection, chained RCE	Every upstream provider key the gateway manages	[11]
JetBrains Marketplace (Oct 2025–Jun 2026)	Distribution channel: IDE plugin marketplace, keys taken from individual developers' plugin settings	15 trojanized AI assistant plugins exfiltrating keys on save	Developer LLM API keys across ~70,000 installs	[10]
Enterprise software vendors (Anthropic case, 2026)	Vendors holding customers' AI keys	Intrusions starting from compromised credentials	Target organizations' AI keys, one reused for ~3 weeks	[1]

What these incidents have in common is that the organization that owns the key is often not the organization that loses it. OpenCode's CLI and desktop apps keep provider keys in a local SQLite database on the user's machine. Only the optional hosted Console, which calls the provider on a workspace's behalf so team members never see the key, held keys centrally [3]. That convenience is what made Console a single place where many customers' keys could be taken at once. Customers usually see a hosted gateway, evaluation platform, or SaaS integration as a productivity tool, but security teams should treat it as a third-party custodian of production credentials.

Why Platform Controls Do Not Contain the Damage

Platform-side remediation has structural limits, and both October disclosures show them. A provider key is a bearer credential: whoever holds it can make billable requests until the issuing provider revokes it [3]. OpenCode's guidance noted that deleting a key from Console does not revoke it, and that Console budgets cannot cap usage outside Console because Console sees only traffic that passes through its own

gateway [3]. Fireworks said the same thing more directly: "Removing a stored secret alone does not invalidate the underlying credential," and "third-party credentials require action with their issuing providers" [6]. Fireworks could revoke its own keys. Third-party keys could only be revoked by their owners, working with the providers that issued them [6].

As a result, remediation depends on customers reading a notice and acting on it, and the stolen keys stay usable in the meantime. OpenCode's notice gave 00:30 UTC on October 17 as the earliest possible exposure time [4]. Affected users reported unauthorized charges, by their own accounts ranging from a few hundred dollars to about \$1,700, and Google suspended some projects for "abusive activity consistent with hijacked resources" [3][4]. These loss figures come from individual users and have not been independently verified. OpenCode has not confirmed how many keys were copied or used [3]. Dealroom reported that Anthropic said it revoked affected Claude API keys [5]. That suggests providers may sometimes revoke keys on their own initiative, but organizations should not plan around it.

The link between the OpenCode incident and distillation should be read carefully. OpenCode described the breach as "related to a large-scale distillation attack" but did not name the attackers or explain how they relate to known distillation campaigns [3]. Some observers have questioned whether the available evidence establishes that characterization [16]. Running up unauthorized charges is consistent with resale or cost-shifting as well as with harvesting training data. What the public record does establish is narrower: frontier providers say stolen customer keys are a known input to distillation infrastructure, and attackers are actively going after platforms that hold such keys in bulk [1][3][6].

Pipelines as Collection Points

CI/CD and evaluation pipelines are a related exposure. Fireworks' exposed credentials were environment variables used in evaluation jobs [6]. Teams routinely place provider keys in such jobs so that benchmarks, regression tests, and agent evaluations can call models directly. Anthropic's report lists "CI/CD injection," "whole-cluster secret dumps," and "vendor-OAuth fan-out to every downstream tenant" among the ways one working credential is turned into broader access inside a victim environment [1]. The LiteLLM compromise followed a similar path. Attackers compromised a security scanner used in CI, used it to steal publishing tokens, and pushed malicious package versions that harvested credentials and moved laterally through Kubernetes [11]. Any pipeline that holds a long-lived key exposes it to anyone who can tamper with the pipeline.

Victim-deployed AI agents add another layer. Anthropic names them explicitly among the sources attackers mine for exposed tokens [1]. GitGuardian's finding of thousands of valid secrets in MCP configuration files points the same way: the configuration patterns that wire AI agents together are

leaking working credentials at scale [8]. Agents often hold several provider keys plus tokens for the tools they orchestrate, so a compromised agent host could supply credentials to distillation proxies and to intrusion operations at the same time.

Downstream Consequences Beyond Cost

Losing money is the most visible harm but not the only one. When a stolen key feeds a distillation proxy, the victim's account is used to extract capabilities that Anthropic says arrive without the original model's safeguards [2]. The traffic is attributed to the victim, which can lead to account suspension, as Google's response to the OpenCode incident showed [4]. The victim's audit trail also fills with activity it did not generate. Anthropic further reports that some labs fed real user conversations, relayed through third-party model routers, into Claude for distillation. Those sessions contained names, email addresses, and company data [1]. Organizations that send traffic through unvetted intermediaries therefore risk their prompts and outputs as well as their keys.

Recommendations

Immediate Actions

Organizations that stored keys in OpenCode Console, Fireworks AI, or any platform that has disclosed a similar incident should revoke those keys at the issuing provider and issue replacements, even if they see no suspicious usage. Fireworks explicitly advises rotation "even if you do not find suspicious activity" [6]. After rotation, teams should update every application, script, and stored secret that used the old key, then review provider-side usage and billing logs from the earliest possible exposure date onward [3]. Anthropic and Google both publish guidance on handling suspected key compromise and on API key hygiene, and teams should follow the provider-specific steps [14][15].

Security teams should also build a quick inventory of every third-party service that holds an AI provider key on the organization's behalf. Typical holders include hosted coding-agent consoles, AI gateways, evaluation and fine-tuning platforms, observability tools, CI/CD secret stores, and IDE plugins. Many organizations do not maintain such an inventory, and without one a third-party breach notice is difficult to act on quickly.

Short-Term Mitigations

Over the next one to three months, organizations should move to one key per tool, per environment, and per integration. That way a single compromise can be traced and revoked without disrupting other systems [3]. Spend limits and usage alerts should be set at the provider, because limits enforced only by a gateway or platform do not apply once the key is used elsewhere [3]. Where providers support it, keys should be scoped to the minimum models and capabilities required, restricted to known source networks, and issued for short lifetimes. Google's best practices for API keys specifically recommend API and application restrictions [15].

Pipeline hygiene needs the same attention. Evaluation and CI jobs should get provider credentials through workload identity or short-lived token exchange instead of long-lived keys in environment variables. Those jobs should run in isolated runners whose secret scope is limited to the job. Organizations should run secret scanning that covers MCP and agent configuration files as well as source code, given the volume of valid credentials GitGuardian found there [8]. Teams should allowlist IDE plugins and other developer tooling that can read stored AI keys, following the lessons of the JetBrains Marketplace campaign [10]. Finally, AI access should be bought only through authorized channels. Anthropic's report warns that any discount requiring traffic and credentials to pass through an unknown intermediary puts the user's data and systems at risk [1].

Strategic Considerations

In the longer term, organizations should treat AI provider keys as non-human identities with lifecycle governance: an owner, a purpose, a scope, an expiry, and continuous monitoring. Vendor risk assessments should ask whether a platform stores customer provider keys, how it encrypts and segments them, whether its legacy and decommissioned systems still hold credentials, and how quickly it will notify customers when exposure is possible. OpenCode's exposure came from a system it was in the process of retiring, which shows that decommissioning is part of the attack surface [3][4][5].

Detection also deserves investment. Distillation traffic run on stolen keys tends to look different from an organization's normal usage. It may arrive in high volume, use systematic prompt templates, attempt to extract reasoning, or come from unfamiliar geographies [1]. Feeding provider usage telemetry into the SIEM and setting baselines per key turns provider billing from a monthly surprise into a detection source. Organizations should also encourage AI providers to offer finer-grained key scoping, workload identity federation, and proactive revocation when bulk exposure at a third party is detected, since these controls belong at the issuing provider.

CSA Resource Alignment

CSA's rapid research on AI credential theft addresses several of the defenses described above, and this note extends that work from individual developer endpoints to centralized platforms and pipelines.

JetBrains Marketplace: AI Plugin Campaign Steals LLM API Keys documents fifteen trojanized AI assistant plugins that exfiltrated developer keys over roughly eight months, along with a credential-laundering monetization model [10]. The resale economy that note describes is the same supply chain Anthropic now links to distillation proxies, and its guidance on plugin allowlisting and key scoping applies directly to the short-term mitigations above.

LiteLLM AI Gateway: Critical Vulnerability Chain Exposes API Keys analyzes how a self-hosted AI gateway concentrates enterprise provider credentials and how supply chain poisoning, pre-authentication SQL injection, and chained remote code execution each exposed that store [11]. The OpenCode and Fireworks incidents show the same concentration risk in hosted and managed platforms, so that note's argument for treating credential aggregation points as tier-zero infrastructure applies here.

Cloud Billing Hijacking as LLMjacking traces LLMjacking from stolen keys to billing-account takeover and recommends spend caps that actually stop usage rather than only send notifications [12]. That recommendation matters equally when the key is taken from a third-party platform.

At the control framework level, the CSA *AI Controls Matrix (AICM) v1.1* addresses related controls [13]. Its Identity and Access Management and Cryptography, Encryption and Key Management domains cover key lifecycle and scoping. Its Supply Chain Management, Transparency and Accountability domain covers third-party custodians of credentials. Its Logging and Monitoring domain covers per-key usage baselines. Organizations can use AICM to turn the vendor-assessment questions above into measurable requirements.

References

- [1] Anthropic. "[Detecting and countering misuse of AI: September 2026](#)." Anthropic Threat Intelligence, September 2026.
- [2] Anthropic. "[Detecting and preventing distillation attacks](#)." Anthropic, February 2026.
- [3] Khandelwal, Swati. "[OpenCode Console Breach May Have Exposed OpenAI, Anthropic, and Google AI Keys](#)." The Hacker News, October 19, 2026.
- [4] Toulas, Bill. "[OpenCode hack exposes API keys amid AI distillation attacks](#)." BleepingComputer, October 19, 2026.
- [5] Dealroom News. "[OpenCode breach drains users' AI keys as platforms holding customer credentials come under attack](#)." Dealroom, October 18, 2026.
- [6] Fireworks AI. "[October 2026 Security Incident](#)." Fireworks AI Security Bulletin, October 2026.
- [7] Sysdig Threat Research Team. "[LLMjacking: Stolen Cloud Credentials Used in New AI Attack](#)." Sysdig, May 2024.
- [8] GitGuardian. "[The State of Secrets Sprawl 2026](#)." GitGuardian, March 2026.
- [9] Lakshmanan, Ravie. "[Anthropic Says Seven China-Based AI Labs Ran Industrial-Scale Claude Distillation Attacks](#)." The Hacker News, September 11, 2026.
- [10] Cloud Security Alliance. "[JetBrains Marketplace: AI Plugin Campaign Steals LLM API Keys](#)." CSA Labs, June 2026.
- [11] Cloud Security Alliance. "[LiteLLM AI Gateway: Critical Vulnerability Chain Exposes API Keys](#)." CSA Labs, June 2026.
- [12] Cloud Security Alliance. "[Cloud Billing Hijacking as LLMjacking: Security Implications and Guidance](#)." CSA Labs, October 2026.
- [13] Cloud Security Alliance. "[AI Controls Matrix \(AICM\) v1.1](#)." Cloud Security Alliance, accessed October 2026.
- [14] Anthropic. "[What should I do if I suspect my API key has been compromised?](#)" Claude Help Center, accessed October 2026.

[15] Google Cloud. "[Best practices for managing API keys.](#)" Google Cloud Documentation, accessed October 2026.

[16] TechCrunch. "[Hackers hit AI coding tool OpenCode, running up charges on users' AI accounts.](#)" TechCrunch, October 19, 2026.