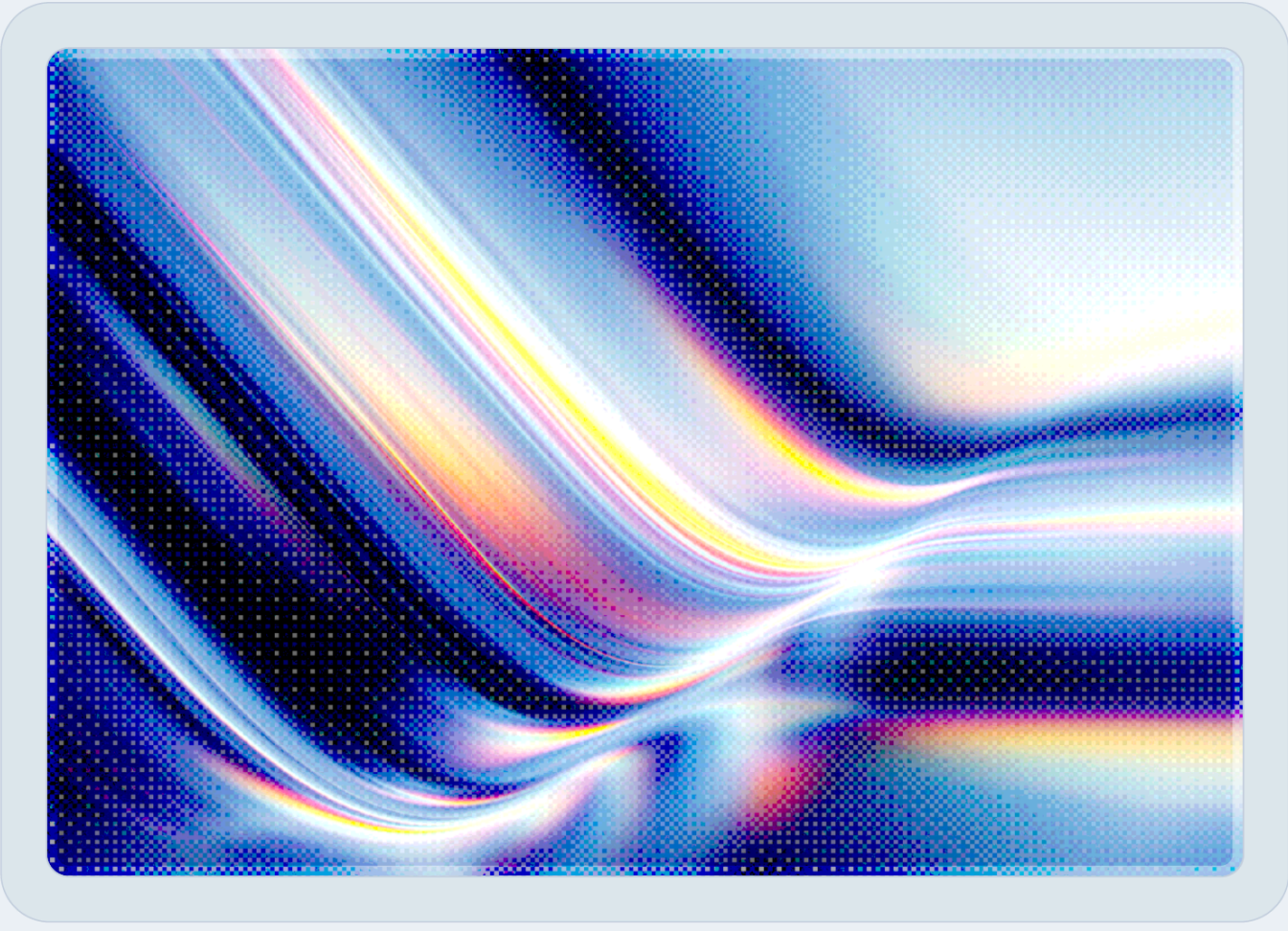
CSAI Foundation | Cloud Security Alliance

Langflow Zero-Day Exploited to Harvest AI and Cloud Credentials

CVE-2026-0768 Marks Langflow's Twelfth Actively Exploited Flaw of 2026

2026-09-04

AI-assisted Rapid Research



CSAI | CSA cloud security alliance®

Key Takeaways

CVE-2026-0768, a critical (CVSS 9.8) unauthenticated remote code execution vulnerability in the Langflow AI development platform, is under active exploitation as of early September 2026, nearly eight months after Trend Micro's Zero Day Initiative (ZDI) publicly disclosed it as a zero-day in January 2026 [1][3]. Threat intelligence firm VulnCheck detected exploitation attempts against internet-facing honeypot sensors beginning August 29–30, 2026, recording more than 50 detections within hours and escalating to over 360 total detections by the following Monday, with the bulk of observed traffic originating from Russia [2][4].

Rather than deploying ransomware or destructive payloads, attackers exploiting CVE-2026-0768 are conducting systematic reconnaissance and credential harvesting: querying environment variables for Langflow superuser tokens, OpenAI API keys, and AWS access keys; reading Langflow's cached secret key from disk; and probing for SSH access and shell history to support lateral movement [1][2][3]. This behavior reflects a broader pattern documented across the AI tooling ecosystem in 2026, in which orchestration platforms that aggregate credentials for large language model providers, cloud accounts, and downstream integrations appear to be priority targets in part because a single compromised host can yield access to dozens of connected services [1][9].

CVE-2026-0768 is not an isolated event. Industry reporting indicates it is the twelfth Langflow vulnerability exploited in the wild during 2026 alone, compared with only one known exploited Langflow flaw before that year, and total exploitation attempts across the platform's 2026 CVE portfolio now exceed 15,000 [9]. CSA has previously documented two of these prior incidents — CVE-2026-33017 and CVE-2026-5027 — both of which share the same unauthenticated-RCE-to-credential-theft pattern now repeating with CVE-2026-0768 [10][11]. Organizations running self-hosted Langflow should treat this as an emergency patching and credential rotation event regardless of whether their specific instance shows signs of compromise.

Background

Langflow is an open-source, Python-based visual framework for building large language model pipelines, retrieval-augmented generation applications, and autonomous AI agents, distributed under the stewardship of DataStax, Langflow's parent company [14], and popular enough to have accumulated

more than 145,000 GitHub stars [13]. Its drag-and-drop workflow editor lets developers assemble AI applications from prebuilt and custom components, including a code editor that allows users to write and validate arbitrary Python logic as part of a flow. That code-validation feature is the root of the vulnerability at the center of this note.

CVE-2026-0768 resides in the code validator underlying Langflow's custom component editor. According to Zero Day Initiative advisory ZDI-26-034, the flaw stems from "the lack of proper validation of a user-supplied string before using it to execute Python code" [3][6]. Because the platform's `validate` endpoint passes attacker-supplied input directly into Python's execution path without sanitization, and because the endpoint requires no authentication, a remote attacker can submit a single crafted HTTP request and have arbitrary Python code run with root privileges on the Langflow host [1][3][4]. Public reporting places the affected range at Langflow releases up to and including version 1.4.2 [1][3][4]; SentinelOne's vulnerability database classifies the flaw under CWE-94 (Improper Control of Generation of Code) with the vector CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H, reflecting network-reachable, low-complexity exploitation requiring no privileges or user interaction and yielding complete compromise of confidentiality, integrity, and availability [5].

Trend Research analysts Peter Girnus, William Gamazo Sanchez, and Alfredo Oliveira reported the flaw to ZDI in July 2025; ZDI published it as a coordinated zero-day advisory on January 9, 2026, after the disclosure window elapsed without a vendor fix being confirmed [3][6]. This is notable because it means CVE-2026-0768 sat in public advisory form for roughly eight months before the exploitation activity documented in this note began, giving defenders ample lead time that appears to have gone largely unused across a meaningful share of internet-exposed deployments. Vulnerability trackers including Qualys ThreatPROTECT and SentinelOne describe version 1.4.2 or later as remediated, but as of this writing Langflow's maintainers have not published a dedicated security advisory confirming the exact patched release for this specific CVE — an ambiguity that echoes the confusion CSA previously documented around the CVE-2026-33017 patch timeline, where version 1.8.2 was mistakenly reported as fixed when it remained fully exploitable [4][5][10]. Organizations should not rely on a specific point release number alone and should instead verify against the current stable Langflow release and any forthcoming vendor advisory.

The `validate` endpoint has a documented history of code-injection weaknesses: an earlier, distinct vulnerability in the same `/api/v1/validate/code` endpoint, CVE-2025-3248, was patched in Langflow 1.3.0 by removing the endpoint's reliance on Python's `exec()` function in favor of compile-only validation [8]. CVE-2026-0768 represents a subsequent code-injection weakness in the same functional area, reported roughly four months after that earlier fix shipped — suggesting either an incomplete remediation of the underlying design pattern or a reintroduction of unsafe execution logic

during later development. This recurrence in a single, well-known endpoint underscores a structural pattern security teams should weigh heavily when evaluating Langflow's development practices: the same class of bug in the same code path was reported twice within roughly a year.

Security Analysis

Exploitation Timeline and Attacker Behavior

VulnCheck's Canary honeypot network first observed exploitation attempts targeting CVE-2026-0768 on August 29–30, 2026, recording more than 50 detection events within hours against sensors positioned in the United Kingdom [2][4]. By the following Monday, September 1, cumulative detections had risen to more than 360, with VulnCheck VP of Threat Research Caitlin Condon characterizing the observed activity as "a mix of reconnaissance and credential harvesting" rather than destructive attack behavior [1][2][4]. The bulk of attacking traffic traced back to infrastructure in Russia [1][2].

The documented attacker playbook is methodical. After achieving code execution through the `validate` endpoint, operators query a specific set of environment variables — `LANGFLOW_SUPERUSER`, `OPENAI_API*`, `AWS_ACCESS*`, and `AWS_SECRET*` — that map directly to the credentials Langflow instances commonly hold for their own administrative access and for the LLM and cloud providers they integrate with [1][2][3]. Attackers additionally read `/root/.cache/langflow/secret_key`, the file Langflow uses to encrypt stored credentials and session data, and check for SSH key material and `.bash_history` size as indicators of further lateral-movement opportunity [1][2][4]. A community technical write-up of the exploitation chain further describes attackers searching source code and `.env` files for embedded secrets, exfiltrating harvested material to external infrastructure, and checking whether a given host had already been backdoored by a prior operator before adding their own persistence — behavior consistent with opportunistic, low-sophistication scanning rather than a single coordinated campaign [7].

No public proof-of-concept exploit is known to have been published at the time exploitation began, according to VulnCheck's reporting [1][4] — a notable data point suggesting independent derivation of an exploit from the January 2026 advisory's technical description nearly eight months after publication, rather than reuse of previously published exploit code. That gap illustrates that public disclosure timing does not reliably predict when exploitation will begin, and that organizations cannot treat an unexploited advisory as a lower operational priority simply because time has passed since disclosure.

Concurrent Exploitation of a Ruby on Rails Flaw

The Hacker News reporting on this campaign notes that VulnCheck's Langflow detections coincided with a parallel wave of exploitation targeting CVE-2026-66066, a critical (CVSS 9.5) Ruby on Rails vulnerability dubbed "KindaRails2Shell," in which a discrepancy between Rails' Active Storage component and the libvips image-processing library allows attackers to coerce arbitrary file reads from crafted image uploads, exposing Rails' `secret_key_base`, database credentials, and cloud tokens [2]. VulnCheck identified more than 7,100 exposed vulnerable Rails instances as of early August 2026 and observed exploitation targeting canary sensors in Singapore, Israel, and the United Kingdom, with attacking infrastructure in France establishing command-and-control connections to Israel [2]. While the article does not establish that the same threat actors are behind both campaigns — the Rails activity's originating and C2 infrastructure differs from the primarily Russian traffic observed against Langflow — the concurrent timing illustrates that credential-harvesting operators are opportunistically working through multiple classes of internet-exposed application infrastructure rather than focusing exclusively on AI-specific tooling [2].

Part of a Broader Langflow Exploitation Pattern

CVE-2026-0768 extends a pattern of Langflow exploitation that has intensified markedly during 2026. Industry analysis published alongside this exploitation wave counts twelve distinct Langflow vulnerabilities exploited in the wild during 2026, against a single known exploited Langflow flaw in all prior years combined, with cumulative successful exploitation attempts across that vulnerability set exceeding 15,000 [9]. CSA has previously covered two of these incidents in dedicated research notes. CVE-2026-33017, an unauthenticated RCE in Langflow's public flow build endpoint, was weaponized within roughly 20 hours of its March 2026 disclosure and was used to harvest OpenAI, Anthropic, and AWS credentials before pivoting to Monero cryptomining deployment [10]. CVE-2026-5027, a path traversal vulnerability in Langflow's file upload endpoint disclosed in March and exploited beginning in June 2026, allowed attackers to write arbitrary files and was linked by researchers to the Iranian state-sponsored group MuddyWater [11]. A third disclosed flaw, CVE-2026-55255, an authorization-bypass (IDOR) vulnerability that let an authenticated attacker execute another user's flows by specifying their flow ID, was added to CISA's Known Exploited Vulnerabilities catalog on July 7, 2026; threat intelligence firm Sysdig had already observed a single operator chaining it with CVE-2026-33017 for credential theft in a campaign it tracked between June 22 and 25, 2026 [15].

In CSA's assessment, the recurrence of unauthenticated RCE and credential-harvesting patterns across at least four distinct CVEs in a single calendar year — each in a different Langflow subsystem (the code validator, the public flow build endpoint, the file upload endpoint, and the workflow execution API) — points to a systemic gap in Langflow's security engineering practices rather than a series of unrelated,

isolated defects. Each of these endpoints independently lacked either authentication enforcement or input sanitization sufficient to prevent unauthenticated code execution or credential exposure, despite the platform's role as a credential-aggregating orchestration layer for enterprise AI pipelines.

Why AI Orchestration Platforms Are High-Value Targets

Langflow's architectural role explains why attackers persistently target it. Enterprise deployments configure Langflow with API keys for LLM providers such as OpenAI and Anthropic, cloud provider credentials for AWS and other infrastructure, database connection strings, and third-party integration secrets, all so that assembled workflows can call these services without developers re-entering credentials for every flow [1][2][11]. A successful compromise of the host therefore yields not just server-level access but authenticated access to every service whose credentials the instance stores — a single point of failure that, in CSA's assessment, is disproportionate to the security posture organizations typically apply to development and orchestration tooling relative to production application infrastructure. Attackers who obtain OpenAI or Anthropic keys can consume an organization's AI spend, exfiltrate proprietary prompts and outputs processed through connected workflows, or use the stolen keys to impersonate the victim organization in downstream API calls. Attackers who obtain AWS credentials can pivot into cloud infrastructure well beyond the Langflow host itself.

Recommendations

Immediate Actions

Organizations running self-hosted Langflow should immediately verify their running version and upgrade to the current stable release, given the ambiguity around the exact patched version documented for this CVE. Any instance exposed to the public internet without a confirmed upgrade should be treated as compromised until proven otherwise, and taken offline or placed behind authenticated network access while remediation is completed. Because the observed attacker behavior focuses on silent credential harvesting rather than visible disruption, the absence of obvious symptoms does not indicate the absence of compromise.

Credential rotation should follow immediately for every secret an affected Langflow instance had access to: the Langflow superuser password, any OpenAI or other LLM provider API keys configured in the instance, AWS access and secret keys, and the Langflow secret key itself. Because attackers specifically targeted `/root/.cache/langflow/secret_key`, rotating this key and re-encrypting stored credentials should be treated as mandatory rather than precautionary for any instance that was internet-

reachable while running an affected version. Organizations should also notify their AI service providers of potential key exposure so that inference logs can be audited for unauthorized usage tied to the compromised keys.

Security teams should audit access logs for requests to the Langflow `validate` endpoint, unexpected outbound connections from the Langflow host, unfamiliar SSH keys added to `authorized_keys`, and any evidence of prior attacker persistence — including signs that a previous operator had already backdoored the instance before the activity documented in this note began.

Short-Term Mitigations

Organizations should restrict network exposure of Langflow instances that do not require public internet accessibility, placing them behind a VPN, reverse proxy, or network segmentation controls rather than exposing the application directly. Where Langflow must remain internet-facing for legitimate business reasons, disabling or gating the `validate` endpoint and any other code-execution-capable functionality not actively required should be evaluated as a compensating control pending confirmed patch application.

Given the credential-aggregation risk inherent to Langflow's design, organizations should move away from storing long-lived, high-privilege API keys directly within the platform wherever feasible. Scoped, short-lived credentials issued through a secrets manager and injected at workflow runtime limit the value of any individual host compromise, reducing the blast radius of a future vulnerability in this or any other orchestration platform.

Strategic Considerations

The recurrence of unauthenticated RCE and credential-theft vulnerabilities across at least four distinct Langflow subsystems within a single year — the code validator, the public flow build endpoint, the file upload endpoint, and the workflow execution API — should inform how organizations weigh the operational risk of self-hosting AI orchestration platforms generally. Security teams evaluating Langflow, or any comparable AI development or orchestration tool, should require evidence of a mature secure development lifecycle, including adversarial testing of code-execution and file-handling endpoints, before granting the platform access to production-grade credentials. Vulnerability management programs should formally include AI development tooling in the same patch-cadence and exposure-monitoring processes applied to customer-facing production systems, rather than treating developer and orchestration infrastructure as lower priority by default. Given that exploitation of CVE-2026-0768

began nearly eight months after public disclosure, organizations cannot assume that time since advisory publication correlates with declining risk; unpatched, internet-exposed instances can remain viable targets for years after disclosure, as this case demonstrates.

CSA Resource Alignment

CSA has published targeted analysis of two prior Langflow vulnerabilities that share substantial technical and behavioral overlap with CVE-2026-0768, both of which security teams evaluating this incident should review directly.

CVE-2026-33017: Langflow RCE Exploits Enterprise AI Pipelines: This CSA research note documents an earlier unauthenticated RCE in Langflow's public flow build endpoint that was weaponized within 20 hours of disclosure and used for the same credential-harvesting objective — OpenAI, Anthropic, and AWS keys — now observed with CVE-2026-0768 [10]. The note's discussion of incomplete patching, where a version reported as fixed remained fully exploitable, is directly applicable to the version ambiguity documented in this note, and its recommendations on credential rotation and AI-tooling inclusion in vulnerability management programs apply without modification to the current incident.

Langflow Path Traversal: Unauthenticated RCE Actively Exploited: This note covers CVE-2026-5027, a file-upload path traversal flaw exploited by suspected state-sponsored actors to achieve unauthenticated RCE. Read alongside CVE-2026-0768, it reinforces that Langflow's exposure is not concentrated in a single component; distinct endpoints across the platform have independently failed to enforce authentication or input validation sufficient to prevent code execution, a pattern organizations should factor into platform-level risk assessments rather than treating each CVE as a standalone event [11].

AI Controls Matrix (AICM) v1.1: As the superset successor to the Cloud Controls Matrix, AICM's application and interface security and vulnerability management control domains provide the framework organizations should use to evaluate whether AI orchestration tooling like Langflow receives the same security testing rigor — including adversarial review of code-execution and file-handling endpoints — expected of production application infrastructure [12]. AICM's identity and access management domain further supports the credential-scoping and secrets-management recommendations in this note, guiding organizations toward short-lived, least-privilege credential issuance rather than static API keys stored directly within orchestration platforms.

References

1. Bill Toulas. "Critical Langflow flaw exploited to steal OpenAI and AWS keys." BleepingComputer, September 2026.
2. Ravie Lakshmanan. "Attackers Exploit Critical Langflow and Rails Flaws in Credential-Probing and C2 Activity." The Hacker News, September 2026.
3. Pierluigi Paganini. "Hackers Target Langflow in CVE-2026-0768 Attacks." Security Affairs, September 2026.
4. Qualys ThreatPROTECT. "Langflow Remote Code Execution Vulnerability Exploited in Attacks (CVE-2026-0768)." Qualys, September 2, 2026.
5. SentinelOne. "CVE-2026-0768: Langflow Code Injection RCE Vulnerability." SentinelOne Vulnerability Database, 2026.
6. Zero Day Initiative. "ZDI-26-034." Trend Micro Zero Day Initiative, January 9, 2026.
7. anoymask. "Exploitation of Langflow CVE-2026-0768: From Unauthenticated Root RCE to Secret Theft and Lateral Movement." DEV Community, September 2026.
8. GitHub Security Advisories. "Langflow Unauth RCE – GHSA-rvqx-wpfh-mfx7 (CVE-2025-3248)." GitHub, 2025.
9. Forkast News. "Langflow's 12th Exploited CVE Confirms AI Frameworks Are Now Credential Harvesting Infrastructure." Forkast News, September 2026.
10. Cloud Security Alliance AI Safety Initiative. "CVE-2026-33017: Langflow RCE Exploits Enterprise AI Pipelines." CSA Labs, July 2, 2026.
11. Cloud Security Alliance AI Safety Initiative. "Langflow Path Traversal: Unauthenticated RCE Actively Exploited." CSA Labs, June 12, 2026.
12. Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." CSA, 2026.
13. GitHub. "langflow-ai/langflow." GitHub repository, accessed September 2026.
14. Langflow. "Big News for Langflow!." Langflow Blog, February 25, 2025.

15. Ravie Lakshmanan. "CISA Adds 4 Actively Exploited Adobe, Joomla, and Langflow Flaws to KEV." The Hacker News, July 8, 2026.