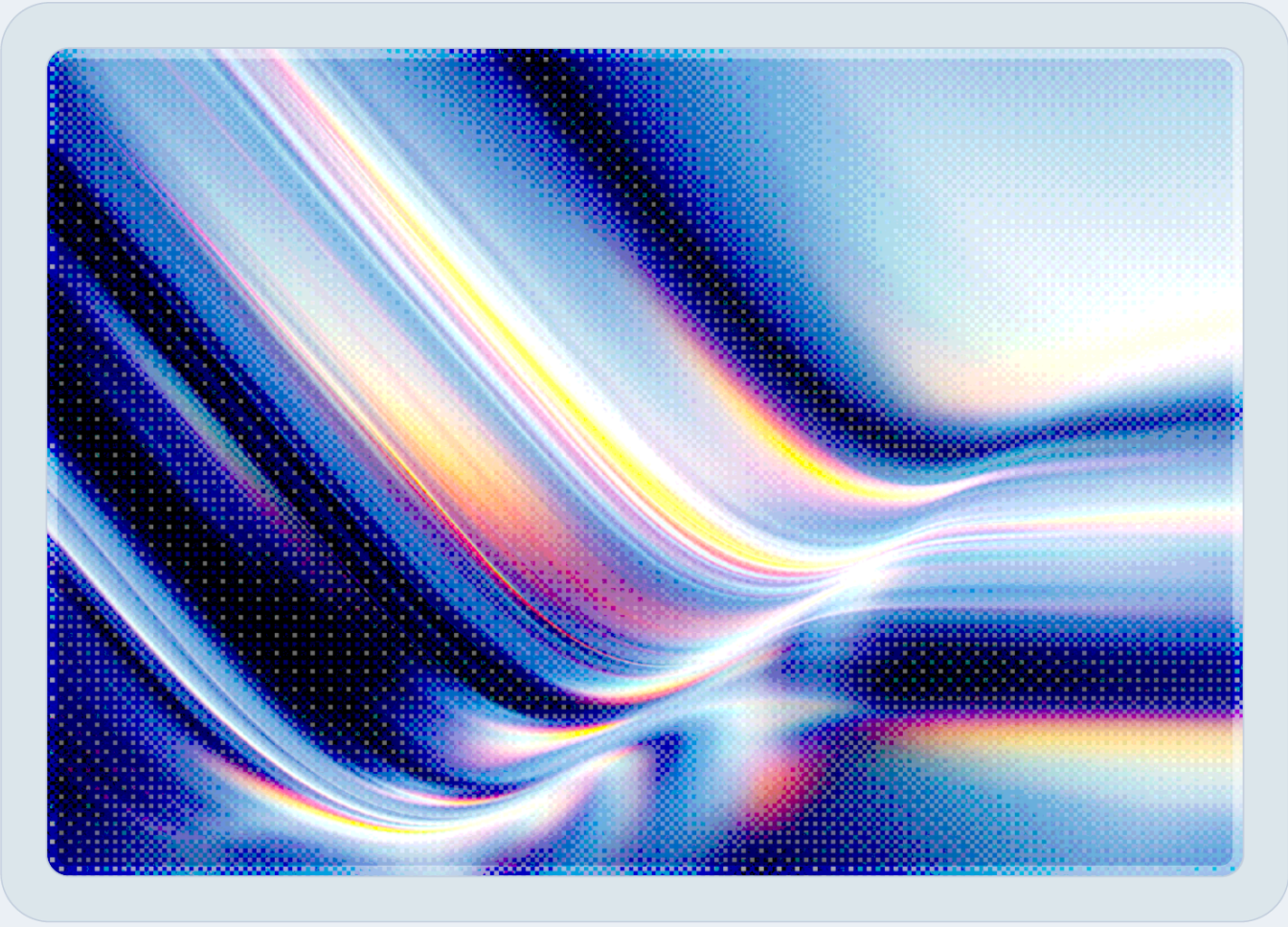
CSAI Foundation | Cloud Security Alliance

Sustained Multi-Actor Exploitation Cements Langflow as Persistent Target

2026-08-30

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

- Langflow, an open-source low-code platform for building AI agents and RAG pipelines with more than 150,000 GitHub stars, has had at least six distinct CVEs actively exploited in the wild since mid-2025. This note profiles those six in depth; VulnCheck's broader tracking puts the total count of exploited Langflow vulnerabilities at twelve — up from just one before 2026 — a pattern VulnCheck describes as unprecedented for the platform [1].
- Canary telemetry from VulnCheck recorded more than 15,000 successful exploitation attempts against Langflow across three vulnerabilities alone (CVE-2026-0769, CVE-2025-3248, and CVE-2026-5027), and separate scanning telemetry logged roughly 650 additional attempts against a fourth, CVE-2026-9198, from 244 unique source addresses spanning 41 countries [1][7].
- VulnCheck's honeypot research documents two unrelated threat actors running independent campaigns against the same population of exposed Langflow instances: one built a credential-harvesting and remote-access toolkit and ran for 27 days, the other built a cryptomining and network-pivoting operation and ran for more than two months, and the two campaigns overlapped on the same target population for nearly four weeks before either was fully observed [1].
- The most recent addition to CISA's Known Exploited Vulnerabilities catalog, CVE-2026-9198, was added on August 5, 2026 — just weeks before this note — with a federal remediation deadline of August 7, 2026, underscoring that exploitation of Langflow is not a resolved, historical incident but an ongoing condition [7].
- The recurring root causes are structural rather than incidental: Langflow ships with authentication disabled by default (`AUTO_LOGIN=true`), executes user-supplied code as a core product feature, and routinely stores credentials for connected LLM providers and cloud services on the same host — meaning each new CVE reopens a similar blast radius regardless of which specific flaw is exploited.
- Organizations should treat any low-code or no-code AI orchestration platform — not just Langflow — as requiring production-grade security governance, because the underlying design pattern of unauthenticated code execution plus aggregated credentials recurs across the category.

Background

Langflow is an open-source, low-code visual framework for assembling AI agents, retrieval-augmented generation (RAG) pipelines, and multi-model workflows through a drag-and-drop interface. Its accessibility to non-specialist builders is likely a factor in its rapid adoption: the project has surpassed 150,000 GitHub stars and is commonly deployed by both individual developers experimenting with agentic AI and enterprise teams standing up internal AI tooling [1][2]. That accessibility comes from a design choice that also defines its risk profile — Langflow treats code execution as a first-class feature rather than an edge case, evaluating user-supplied Python within flow nodes, validation endpoints, and build pipelines so that workflows can be assembled and tested interactively.

Cloud Security Alliance research has tracked this risk profile closely over the past several months, publishing separate analyses of CVE-2026-5027 (a path traversal flaw enabling unauthenticated arbitrary file write and RCE) and CVE-2026-33017 (an unauthenticated RCE in the public flow build endpoint) as each was disclosed and weaponized [3][4]. What distinguishes this note is not a new CVE but a new observation about pattern: VulnCheck's August 2026 research, drawn from purpose-built Langflow honeypots ("canaries"), shows that the platform has not experienced a single incident followed by remediation, but a sustained, overlapping sequence of independent campaigns exploiting different vulnerabilities toward different ends, some running concurrently for weeks without detection [1]. That shift — from isolated CVE to persistent contested territory — is the subject of this analysis.

Security Analysis

A widening vulnerability surface

Langflow's 2026 vulnerability history reads less like a series of isolated disclosures and more like a running inventory of the same underlying weaknesses surfacing through different code paths. CVE-2025-3248, disclosed in mid-2025, allowed unauthenticated remote code execution through the `/api/v1/validate/code` endpoint by invoking Python's `exec()` on attacker-supplied input; it was later linked to deployment of the Flodrix botnet [5][6]. CVE-2026-33017, disclosed in March 2026, achieved the same outcome through the public flow build endpoint and was weaponized within 20 hours of disclosure — before any public proof-of-concept existed — leading to credential exfiltration and a documented Monero cryptomining campaign, as CSA's earlier research on that disclosure documented in detail [9]. CVE-2026-5027, a path traversal flaw in the file upload handler, was independently confirmed exploitable against roughly 7,000 internet-accessible instances, most located in North America [2][3]. CVE-2026-55255, disclosed in June 2026, introduced a distinct class of flaw — an

insecure direct object reference in the agent execution endpoint that let any authenticated user hijack another tenant's workflow by supplying a harvestable UUID — and was observed chained with CVE-2026-33017 within 48 hours of disclosure [8]. CVE-2026-0769, an unauthenticated code injection flaw in the platform's custom-component evaluation path, was the vulnerability the cryptomining actor described below used to establish persistence after initial access [1]. Most recently, CVE-2026-9198, a code injection flaw permitting full RCE on default deployments, was added to CISA's KEV catalog on August 5, 2026, with telemetry showing roughly 650 exploitation attempts from 244 distinct source addresses across 41 countries beginning in early July [7].

Table 1 summarizes the pattern across all six disclosures.

CVE	Vulnerability Class	CVSS	Disclosed	Time to Exploitation	Fixed In
CVE-2025-3248	Unauthenticated code injection (`exec()`)	9.8	2025	Not reported (later linked to Flodrix botnet deployment)	1.3.0
CVE-2026-33017	Unauthenticated RCE (public flow build)	9.3	Mar. 17, 2026	~20 hours	1.9.0
CVE-2026-5027	Path traversal to RCE (file upload)	8.8	~Mar. 2026	~73 days	1.9.0
CVE-2026-0769	Unauthenticated code injection (custom component eval)	9.8	2026	Not reported (used for post-compromise persistence)	Not specified in source
CVE-2026-55255	IDOR / cross-tenant authorization bypass	8.4–9.9	Jun. 23, 2026	~2 days	1.9.1
CVE-2026-9198	Unauthenticated code injection (RCE)	9.8	~Jun. 2026	Within days of patch	1.10.1

Disclosure dates for CVE-2026-5027 and CVE-2026-9198 are approximate where cited sources did not report an exact date.

Individually, each of these vulnerabilities has already been documented by CSA and others as a discrete incident. Read together, they describe a platform whose disclosure-to-exploitation cycle has compressed and whose vulnerability classes span the full range of an AI orchestration platform's attack surface: unauthenticated code execution, path traversal, and authorization bypass. No single patch closes this surface, because the surface is a product of the platform's core design rather than a specific coding defect.

Two attackers, two objectives, one target

The most significant addition to this picture is VulnCheck's direct observation of concurrent, independent threat actors operating against the same exposed Langflow population with entirely different objectives [1]. The first, active from May 12 to June 8, 2026 (27 days), entered through CVE-2026-5027 and built a credential-extraction operation: it deployed a custom Python harvester that ran for roughly 20 seconds per execution and exfiltrated results to an external collection server, installed proxy agents and a commercial remote-access tool (SimpleHelp) for hands-on-keyboard access, established cron-based persistence, and maintained command-and-control over IRC. Its objective was straightforward — collect whatever credentials a compromised Langflow host held, likely including API keys for connected LLM providers, and use those credentials for broader access.

The second actor, active from April 22 to June 25, 2026 (64 days), pursued an entirely different goal. It gained initial access through CVE-2025-3248, later added persistence through CVE-2026-0769, deployed a SOCKS5 tunneling tool to route traffic through compromised hosts, disabled the system's audit daemon to reduce forensic visibility, and activated a Monero cryptocurrency miner. It also used a vulnerability scanning framework to identify additional exploitable Langflow instances, and pivoted via SSH to at least one further host, indicating an operation focused on expanding compute-resource theft rather than data exfiltration.

The two campaigns overlapped from May 12 to June 8, 2026 — nearly four weeks during which both operations ran simultaneously against the same target population, apparently without either actor being aware of, or coordinating with, the other. This is a meaningfully different threat picture than "a vulnerability was exploited": it indicates that Langflow's exposed population has become contested infrastructure, attractive enough to draw multiple, differently motivated threat actors who are willing to run sustained campaigns rather than opportunistic scans.

Why the same design flaws keep recurring

Three structural characteristics explain why Langflow keeps reappearing in exploitation telemetry regardless of which specific CVE is current. First, the platform's default configuration disables authentication (`AUTO_LOGIN=true`), so any newly discovered endpoint-level flaw is immediately reachable without credentials — a configuration choice that turns ordinary bugs into unauthenticated RCE. Second, code execution is not a bug class Langflow occasionally exhibits; it is the platform's core function, meaning that new features (a public-sharing endpoint, a validation endpoint, an agent-execution endpoint) each represent a fresh opportunity for insufficiently sandboxed code evaluation to be exposed. Third, Langflow instances routinely aggregate credentials for the very services they orchestrate — OpenAI, Anthropic, cloud provider, and database credentials are commonly present in environment variables or configuration files on the same host — so a successful compromise, regardless of entry vector, tends to yield the same high-value outcome. These three characteristics are independent of any individual patch, consistent with a pattern in which remediating one CVE has repeatedly been followed by exploitation of the next.

Recommendations

Immediate Actions

Organizations running Langflow in any capacity should confirm they are on version 1.10.1 or later, the first release addressing CVE-2026-9198, and should not rely on version-string claims alone given the documented history of incomplete patches in this codebase — version 1.8.2 was reported as fixed but remained fully exploitable [9]. Any instance still reachable from the public internet should be placed behind authentication and network access controls immediately, regardless of patch status, since `AUTO_LOGIN=true` remains the default and each newly disclosed endpoint has repeatedly proven reachable without credentials. Security teams should rotate credentials known to be aggregated on Langflow hosts — LLM provider API keys, cloud credentials, and, per CSA's prior analysis, database connection strings — treating rotation as warranted by exposure history rather than confirmed compromise, given that both VulnCheck-documented campaigns ran for weeks before detection.

Short-Term Mitigations

Teams should conduct a retrospective review of Langflow access and system logs covering at least the past several months, looking for the indicators VulnCheck documented: unexpected cron entries (including the specific pattern `/usr/bin/3WA72N.sh`), unfamiliar SSH keys or remote-access

tooling such as SimpleHelp, disabled audit logging (`auditd`), unrecognized cryptomining processes, or outbound connections to unfamiliar IRC or C2 infrastructure [1]. Because two unrelated campaigns coexisted on the same asset population for nearly four weeks, the absence of one indicator does not rule out the other, and organizations should check for both credential-theft and cryptomining/botnet indicators rather than assuming a single compromise pattern. Organizations should also inventory all Langflow deployments across the enterprise, including informal or shadow instances stood up by individual teams experimenting with agentic AI, since these are the deployments most likely to be internet-exposed and least likely to be included in existing vulnerability management scope.

Strategic Considerations

The persistence of exploitation against Langflow across at least six CVEs and more than a year argues that patch-by-patch remediation is not, by itself, a sufficient security strategy for AI low-code and orchestration platforms. Organizations should treat any platform in this category — Langflow, LangGraph, LangChain-based tooling, or comparable agent-orchestration frameworks — as requiring the same production-grade hardening as customer-facing infrastructure: authentication enabled by default, network segmentation restricting access to authorized users, short-lived or vaulted credentials rather than long-lived secrets in environment variables, and inclusion in continuous vulnerability management rather than one-time deployment review. Security and procurement teams evaluating new low-code AI tooling should specifically ask vendors how code execution is sandboxed, whether authentication is enabled by default, and how credential aggregation risk is scoped — the three characteristics this note identifies as the recurring root cause across Langflow's exploitation history. Finally, given that VulnCheck observed two independent campaigns operating undetected for weeks at a time, organizations should evaluate whether their monitoring would actually surface anomalous behavior — unexpected outbound connections, new persistence mechanisms, unfamiliar processes — on AI development infrastructure with the same rigor applied to production systems.

CSA Resource Alignment

This note builds on three prior CSA analyses of individual Langflow vulnerabilities, each of which remains directly relevant to understanding the sustained campaign documented here. CSA's research note on CVE-2026-33017: Langflow RCE Exploits Enterprise AI Pipelines [9] analyzed the credential-exfiltration and cryptomining activity that followed that vulnerability's disclosure, mapping it to MAESTRO's Layer 3 (Agent Frameworks and Orchestration) and Layer 4 (Deployment and Infrastructure); the cryptomining campaign documented in this note extends that same pattern through a second, independently discovered persistence mechanism (CVE-2026-0769) and initial-access vector (CVE-2025-3248),

reinforcing that the orchestration layer itself — not any single CVE — is the durable target. CSA's analysis of Langflow Path Traversal: Unauthenticated RCE Actively Exploited [10] documented the roughly 7,000 internet-exposed instances that gave the credential-harvesting actor described here its entry point, and its Zero Trust-aligned recommendation to eliminate unauthenticated public exposure applies with equal force to every CVE in Table 1, since each one is reachable specifically because instances are exposed without authentication. Finally, Langflow Authorization Bypass Added to CISA's KEV Catalog [11], CSA's analysis of CVE-2026-55255, addresses the identity and access management dimension of this problem: its finding that Langflow's cross-tenant authorization checks were inconsistently applied across resource-resolution paths illustrates the same pattern — security hardening added reactively, one endpoint at a time, rather than designed in from the start — that this note observes across the platform's broader exploitation history. Collectively, these findings map to the AI Controls Matrix (AICM) v1.1, particularly its supply chain security, runtime/sandboxing, and identity and access management domains, all of which speak directly to the structural weaknesses — default-open authentication, unsandboxed code execution, and aggregated credentials — that this note identifies as the common thread across a year of Langflow exploitation.

References

[1] VulnCheck. "Same Target, Different Playbooks: Two Attackers, Two Different Paths to Pwning the AI Stack." VulnCheck, August 2026.

[2] VentureBeat. "7,000 Langflow Servers Under Attack. LangGraph and LangChain Have the Same Holes." VentureBeat, June 2026.

[3] Bleeping Computer. "Path Traversal Flaw in AI Dev Platform Langflow Exploited in Attacks." Bleeping Computer, 2026.

[4] The Hacker News. "Langflow Vulnerability CVE-2026-5027 Exploited for Unauthenticated RCE." The Hacker News, June 2026.

[5] Horizon3.ai. "Unsafe at Any Speed: Abusing Python Exec for Unauth RCE in Langflow AI." Horizon3.ai, 2025.

[6] runZero. "Langflow Flodrix Vulnerability CVE-2026-33017: Find Impacted Assets." runZero, 2026.

[7] The Hacker News. "CISA Flags Langflow RCE, Tomcat, and N-central Flaws as Actively Exploited." The Hacker News, August 2026.

[8] Sysdig. "Understanding Langflow CVE-2026-55255, and Why Higher CVSS Vulnerabilities Aren't Always the Most Exploited." Sysdig, 2026.

[9] Cloud Security Alliance. "CVE-2026-33017: Langflow RCE Exploits Enterprise AI Pipelines." CSA AI Safety Initiative, July 2026.

[10] Cloud Security Alliance. "Langflow Path Traversal: Unauthenticated RCE Actively Exploited." CSA AI Safety Initiative, 2026.

[11] Cloud Security Alliance. "Langflow Authorization Bypass Added to CISA's KEV Catalog." CSA AI Safety Initiative, July 2026.

