Langflow CVE-2026-33017: Unauthenticated RCE in AI Orchestration

Active Exploitation of Critical Code Injection Flaw Threatens AI Pipeline Infrastructure

Unofficial AI-assisted Research

Cloud Security Alliance AI Safety Initiative

2026-03-23

Key Takeaways

A critical unauthenticated remote code execution vulnerability in Langflow, tracked as CVE-2026-33017 with a CVSS v4.0 score of 9.3, came under active exploitation within approximately 20 hours of public disclosure on March 17, 2026 [1][2][5]. The flaw allows any unauthenticated network-accessible attacker to achieve full server-side code execution on vulnerable Langflow instances by submitting a crafted HTTP POST request — no credentials, no prior access, and no public proof-of-concept were required for adversaries to weaponize the advisory within less than a day [2][3]. Observed post-exploitation activity has focused on harvesting AI infrastructure credentials: LLM API keys, cloud provider credentials, and database connection strings that are routinely stored in the environment of a Langflow deployment [2]. Organizations running Langflow versions 1.8.1 or earlier should treat this as an emergency patching priority and audit affected systems for signs of compromise regardless of whether they consider their instances internet-exposed.

Four points demand immediate attention. First, Langflow 1.9.0 is the only fully remediated version; all prior releases remain vulnerable [3]. Second, the attack requires only a single POST request with a malicious payload to a public-facing endpoint — no authentication and no prior access required — and leaves no anomalous error responses in default logging configurations if monitoring is not configured to capture API request bodies [2]. Third, Langflow instances commonly hold the keys to an organization's entire AI stack, making compromise a supply chain event rather than an isolated host incident in organizations where Langflow holds connections to multiple downstream systems [2][6]. Fourth, this is the third critical RCE vulnerability in Langflow within a twelve-month period, and the pattern suggests structural design tension between the platform's extensibility model and secure authentication enforcement [4][7][8].

Background

Langflow is an open-source, Python-based low-code visual builder for AI agents and Retrieval-Augmented Generation (RAG) pipelines [6]. Its drag-and-drop node graph interface allows developers to assemble complex AI workflows — connecting large language models, vector databases, external APIs, and custom Python logic — without writing extensive integration code. The platform has attracted broad adoption among enterprises and developers building production AI systems, accumulating over 145,000

GitHub stars as of the disclosure date [1]. Langflow was originally developed by Logspace and acquired by DataStax in April 2024, integrating it into DataStax's generative AI product stack; IBM subsequently announced plans to acquire DataStax, further expanding Langflow's enterprise footprint [6].

The security significance of Langflow extends beyond the platform itself. Because Langflow functions as an orchestration layer, deployed instances commonly hold direct connections to databases, cloud storage buckets, vector stores, and LLM provider API keys. A compromised Langflow server therefore provides an attacker with a credential store that can unlock substantial downstream infrastructure — a characteristic that positions it as a disproportionately high-value target relative to its apparent network footprint [2]. This positioning in the AI supply chain is not unique to Langflow; similar orchestration platforms — including n8n, Flowise, and Dify — occupy architecturally analogous intermediary roles between AI models and downstream systems, and are likely subject to comparable credential-exposure risks. The security posture of these systems warrants the same scrutiny applied to traditional middleware and integration platforms.

CVE-2026-33017 is the third critical RCE-class vulnerability disclosed in Langflow within approximately twelve months. CVE-2025-3248 (CVSS 9.8) affected the `/api/v1/validate/code` endpoint and was added to the CISA Known Exploited Vulnerabilities catalog following observed delivery of the Flodrix botnet [7]. CVE-2025-34291 (CVSS v4.0: 9.4) was a critical account takeover and RCE chain discoverable by inducing a target user to visit a malicious webpage [8]. The recurrence of high-severity vulnerabilities in publicly accessible API endpoints reflects a structural challenge: Langflow is architected to execute arbitrary user-defined Python code as a core design feature, making the enforcement of authentication boundaries on any code-execution pathway an exceptionally high-stakes engineering requirement.

Security Analysis

Vulnerability Mechanics

CVE-2026-33017 was discovered by security researcher Aviral Srivastava and reported through responsible disclosure channels on February 26, 2026, approximately three weeks before the GitHub Security Advisory was published [3]. The flaw exists in the `POST /api/v1/build_public_tmp/{flow_id}/flow` endpoint, which was designed to allow unauthenticated users to build and preview "public flows" — a feature intended to support shareable, publicly accessible AI workflow demonstrations [2][3].

The vulnerability arises from an optional `data` parameter that, when supplied, causes the endpoint to substitute attacker-controlled payload data for the flow data that would normally be fetched from the database. Langflow's flow node definitions support arbitrary Python code as a first-class element of the workflow graph. When the endpoint processes attacker-supplied flow data, it passes the embedded Python directly to Python's `exec()` function with no sandboxing, no process isolation, and no filtering [1][2][3]. The result is unauthenticated arbitrary code execution running with the full privileges of the Langflow server process, triggered by a single HTTP request. The vulnerability is classified under CWE-306 (Missing Authentication for Critical Function), CWE-94 (Improper Control of Code Generation), and CWE-95 (Improper Neutralization of Directives in Dynamically Evaluated Code) [3].

The patch applied in Langflow 1.9.0 removes the `data` parameter from the `build_public_tmp` endpoint entirely, forcing the endpoint to always call `build_graph_from_db` and execute only administrator-controlled, database-resident flow data [3]. The fix is architectural rather than cosmetic: the endpoint should never have accepted caller-supplied payloads that could be interpreted as executable code. EPSS assigns CVE-2026-33017 a score of 0.50, placing it in the 64th percentile for exploitation probability [4]. Given confirmed active exploitation within 20 hours, the EPSS score underestimates realized exploitation intensity — a reminder that predictive scores reflect probability distributions, not ceilings, and should be read alongside threat intelligence reporting.

Exploitation Timeline and Attacker Behavior

The Sysdig Threat Research Team documented the exploitation campaign in detail, providing the clearest public record of attacker methodology to date [2]. The GitHub Security Advisory GHSA-vwmf-pq79-vjvx was published on March 17, 2026 at approximately 20:05 UTC. By March 18, 2026 at approximately 16:04 UTC — a window of roughly 20 hours — the first exploitation attempts were observed against live Langflow instances [2]. No public proof-of-concept code existed at the time; the advisory contained sufficient endpoint and parameter detail that attackers appear to have reconstructed the attack without requiring a public exploit — a conclusion supported by the speed and precision of the scanning campaign observed within 20 hours [2].

The observed campaign proceeded through three distinct phases. In the initial mass-scanning phase, four source IP addresses conducted coordinated internet-wide scans delivering identical payloads, indicating a single actor with pre-built exploit tooling executing a broad reconnaissance sweep. A fifth IP subsequently transitioned to deeper reconnaissance on instances confirmed as vulnerable, performing environment enumeration rather than simply testing reachability. A sixth IP conducted targeted credential harvesting operations, staging exfiltrated data for retrieval. Post-exploitation techniques observed across the campaign included extraction of environment variables targeting LLM API keys and cloud provider credentials, reading of `/etc/passwd` and `.env` files, database file enumeration

using filesystem search commands, and delivery of next-stage Python dropper payloads from a staging server at `173.212.205[.]251:8443` [2]. As of available reporting, no named threat group has been publicly attributed to this activity.

The 20-hour exploitation window is consistent with the accelerating weaponization timelines observed in recent high-severity vulnerabilities affecting developer and AI tooling — a pattern well documented in current threat intelligence reporting. It reflects a threat environment in which organizations can no longer assume days or weeks of safety following a security advisory — adversaries with automated scanning infrastructure and pre-staged exploit frameworks treat advisory publication as an operational trigger.

Scope of Exposure

Assessing exposure requires acknowledging what is known and what is uncertain. Langflow instances exposed directly to the internet are at highest risk, but the threat is not limited to publicly routable deployments. Many enterprise Langflow deployments operate within internal networks or behind VPNs, yet remain vulnerable if an attacker has already achieved any degree of network proximity — whether through a compromised developer endpoint, a lateral movement path from a different system, or a misconfigured network boundary. The platform's design as a multi-user collaboration environment means that a single compromised instance may hold credentials and connections spanning multiple projects, teams, and cloud accounts. Organizations that have deployed Langflow in containerized environments should additionally be aware that an authenticated user with code execution capability can potentially invoke Langflow's built-in `superuser` CLI command to create administrator accounts, extending the blast radius of a successful compromise [4].

Attribute	Detail
CVE	CVE-2026-33017
CVSS v4.0 Score	9.3 (Critical)
Affected Versions	All Langflow versions ≤ 1.8.1
Fixed Version	Langflow 1.9.0
Attack Vector	Network (unauthenticated HTTP POST)
Affected Endpoint	`POST /api/v1/build_public_tmp/{flow_id}/flow`

Attribute	Detail
CWE Classifications	CWE-306, CWE-94, CWE-95
EPSS Score	0.50 (64th percentile)
First Exploitation Observed	March 18, 2026 (~20 hours post-disclosure)
CISA KEV Status	Not listed as of 2026-03-23 (point-in-time; verify at cisa.gov/known-exploited-vulnerabilities-catalog)
Reporter	Aviral Srivastava (Aviral2642)

Recommendations

Immediate Actions

The most critical step for any organization running Langflow is upgrading to version 1.9.0 without delay. The vulnerability requires no credentials and no prior foothold, and exploitation tooling is demonstrably available to adversaries who were scanning the internet within 20 hours of the advisory's publication [2] [3]. For organizations that cannot immediately upgrade, disabling the `build_public_tmp` API endpoint through a reverse proxy or web application firewall rule provides a temporary mitigation, though it is not a substitute for patching. Organizations should block or rate-limit all access to `/api/v1/build_public_tmp/` endpoints at the network perimeter until patching is confirmed complete.

Parallel to patching, any Langflow instance that was internet-accessible between March 17 and the date of patching should be treated as potentially compromised pending investigation. Incident response teams should examine application logs for POST requests to the `build_public_tmp` endpoint, extract and rotate any credentials stored in environment variables or `.env` files accessible to the Langflow process, audit cloud IAM activity for any LLM API keys or cloud credentials associated with the

instance, and inspect host filesystems and network connections for indicators of the observed dropper activity associated with `173.212.205[.]251:8443` [2]. Credential rotation should occur before re-enabling external access, since stolen credentials remain valid after patching.

Short-Term Mitigations

Organizations should verify that AI orchestration tools are not deployed with broader network access than their security posture justifies. These platforms are often introduced as internal prototypes before gradually accumulating production responsibilities — a lifecycle that can leave network segmentation and access controls miscalibrated for their eventual role as credential stores. Langflow instances should not be directly internet-accessible unless explicitly required by use case, and even internal deployments should be placed behind authentication proxies that enforce organizational identity verification before requests reach the application. Authentication should not be delegated to the application itself as the sole control.

Credential hygiene for AI infrastructure requires specific attention because LLM API keys and vector database credentials are frequently issued with broad permissions and rotated infrequently. Organizations should adopt a least-privilege approach to LLM API keys used by Langflow — scoping keys to specific models and rate limits where provider APIs support it — and implement automated rotation on a schedule that assumes the possibility of undetected compromise. Cloud credentials accessed through Langflow should use short-lived tokens via IAM roles rather than long-lived static keys stored in environment variables.

Vendor support for behavioral monitoring in AI orchestration environments is currently limited relative to traditional application security tooling — a gap organizations should account for when designing their detection strategy. Organizations should configure logging to capture full API request paths and, where possible, request body metadata for Langflow's API endpoints, enabling detection of exploitation attempts that might otherwise appear only as successful requests with no error response. SIEM rules specifically targeting the `build_public_tmp` endpoint should be implemented as an ongoing detection capability, not just for incident response.

Strategic Considerations

The recurrence of critical RCE vulnerabilities in Langflow — three in twelve months — signals a meaningful security risk that organizations should weigh when evaluating AI orchestration platforms, particularly those designed to execute arbitrary user-defined code as a core capability. The architectural characteristic that makes Langflow especially susceptible — direct `exec()` access via API-accessible endpoints — is not universal across the category, and organizations should explicitly assess whether their

chosen platform shares this design pattern. Platforms enabling users to define and execute arbitrary code as a first-class feature cannot rely on input validation alone as their primary security control; defense in depth, runtime isolation, and strong authentication are required at every layer because the attack surface scales with the platform's extensibility. Organizations evaluating AI orchestration tools should explicitly assess the vendor's security track record, patch velocity, and the architectural mechanisms they have implemented to isolate user-defined code execution from administrative and authentication pathways.

Supply chain thinking is essential for AI infrastructure. A Langflow compromise is not merely a server compromise — it is a potential keyring for an organization's entire AI stack, including LLM providers, vector databases, document stores, and cloud services. Inventory and classification of AI orchestration deployments should be maintained with the same rigor applied to other credential stores and integration middleware, and incident response playbooks should pre-plan the credential rotation and downstream audit steps that will be required when one of these platforms is compromised.

The recurrence of critical vulnerabilities in Langflow suggests that the security review and hardening cycle for this platform has not kept pace with its enterprise adoption — a risk organizations should explicitly account for when onboarding open-source AI tooling. Organizations should apply the same pre-deployment security assessment to AI orchestration platforms — including penetration testing, API surface review, and authentication architecture evaluation — that they would apply to any other internet-adjacent system holding privileged credentials.

CSA Resource Alignment

This vulnerability has direct relevance to multiple active CSA research workstreams and published frameworks. The MAESTRO threat model for agentic AI systems specifically addresses execution-layer vulnerabilities in orchestration platforms, identifying code injection and authentication bypass as high-priority threat categories for AI agents that hold privileged access to downstream services [10]. CVE-2026-33017 is a concrete instantiation of the MAESTRO execution-layer threat category, demonstrating in production conditions how a flaw in an orchestration platform's authentication boundary translates directly into supply chain risk for AI-connected infrastructure.

The AI Controls Matrix (AICM) v1.0.3 provides the most directly applicable governance framework for organizations seeking to implement durable controls around AI orchestration deployments. The AICM Implementation Guidelines for Orchestrated Service Providers (OSPs) address the specific governance challenges of platforms that serve as intermediaries between AI models and downstream systems, including authentication controls, credential management, runtime isolation, and security incident

response [11]. Organizations operating Langflow in production are functioning as OSPs under the AICM taxonomy and should align their security controls to the OSP implementation guidelines regardless of whether they self-identify as service providers. The AICM's requirement for authenticated access controls on all privileged endpoints, and its guidance on secrets management for AI pipeline credentials, would directly address the categories of failure demonstrated by CVE-2026-33017.

CSA's published guidance on securing LLM-backed systems addresses the authorization architecture failures that underlie this class of vulnerability, specifically highlighting the risks of AI platforms that accept external inputs in code-execution pathways without enforcing authentication [12]. The Zero Trust for LLM Environments guidance is equally applicable: the observed post-exploitation behavior — immediate extraction of credentials and API keys — demonstrates why LLM infrastructure should be designed on the assumption of compromise, with short-lived credentials, minimal-scope permissions, and continuous behavioral monitoring rather than perimeter-based trust models [13]. The Agentic AI Red Teaming Guide provides organizations with a methodology for proactively identifying authentication bypass and code injection vulnerabilities in their AI orchestration deployments before adversaries do [9].

References

[1] The Hacker News, "Critical Langflow Flaw CVE-2026-33017 Triggers Attacks within 20 Hours of Disclosure," March 20, 2026. https://thehackernews.com/2026/03/critical-langflow-flaw-cve-2026-33017.html

[2] Sysdig Threat Research Team, "CVE-2026-33017: How Attackers Compromised Langflow AI Pipelines in 20 Hours," Sysdig Blog, March 2026. https://sysdig.com/blog/cve-2026-33017-how-attackers-compromised-langflow-ai-pipelines-in-20-hours

[3] langflow-ai, "GitHub Security Advisory GHSA-vwmf-pq79-vjvx: Unauthenticated Remote Code Execution in Langflow via Public Flow Build Endpoint," GitHub, March 17, 2026. https://github.com/langflow-ai/langflow/security/advisories/GHSA-vwmf-pq79-vjvx

[4] Wiz, "CVE-2026-33017 – Impact, Exploitability, and Mitigation Steps," Wiz Vulnerability Database, March 2026. https://www.wiz.io/vulnerability-database/cve/cve-2026-33017

[5] SecurityWeek, "Critical Langflow Vulnerability Exploited Hours After Public Disclosure," March 2026. https://www.securityweek.com/critical-langflow-vulnerability-exploited-hours-after-public-disclosure/

[6] Langflow / DataStax, "Langflow – Open-Source Low-Code AI Builder," langflow.org. https://www.langflow.org/

[7] Trend Micro, "Critical Langflow Vulnerability CVE-2025-3248 Actively Exploited to Deliver Flodrix Botnet," Trend Micro Research, 2025. https://www.trendmicro.com/en_us/research/25/f/langflow-vulnerability-flodric-botnet.html

[8] Obsidian Security, "CVE-2025-34291: Critical Account Takeover and RCE Vulnerability in the Langflow AI Agent Workflow Platform," Obsidian Security Blog, 2025. https://www.obsidiansecurity.com/blog/cve-2025-34291-critical-account-takeover-and-rce-vulnerability-in-the-langflow-ai-agent-workflow-platform

[9] Cloud Security Alliance, "Agentic AI Red Teaming Guide," CSA AI Safety Initiative, 2025. https://cloudsecurityalliance.org/research/working-groups/artificial-intelligence/

[10] Cloud Security Alliance, "MAESTRO: Multi-Agent Environment, Security, Threat, and Risk Ontology," CSA AI Safety Initiative, 2025. https://cloudsecurityalliance.org/research/working-groups/artificial-intelligence/

[11] Cloud Security Alliance, "AI Controls Matrix (AICM) v1.0.3," CSA AI Safety Initiative, 2026. https://cloudsecurityalliance.org/artifacts/ai-controls-matrix

[12] Cloud Security Alliance, "Securing LLM-Backed Systems: Essential Authorization Practices," CSA, 2024. https://cloudsecurityalliance.org/research/working-groups/artificial-intelligence/

[13] Cloud Security Alliance, "Using Zero Trust to Secure Enterprise Information in LLM Environments," CSA, 2025. https://cloudsecurityalliance.org/artifacts/using-zero-trust-to-secure-enterprise-information-in-llm-environments

