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

A Pattern of Code Execution Vulnerabilities in Low-Code AI Development Tools

Unofficial AI-assisted Research

Cloud Security Alliance AI Safety Initiative

2026-03-22

Key Takeaways

- CVE-2026-33017 is a CVSS 9.3/10.0 unauthenticated remote code execution vulnerability in Langflow, a widely-deployed open-source AI workflow builder with over 140,000 GitHub stars. A single unauthenticated HTTP request is sufficient to achieve full server compromise.
- Active in-the-wild exploitation was observed within 20 hours of the advisory's publication on March 17, 2026 — before any public proof-of-concept existed. Sophisticated threat actors appear to have reverse-engineered the exploit from the advisory description alone, with no prior proof-of-concept required.
- The root cause is structural: Langflow's public flow-sharing endpoint passes attacker-controlled flow data directly to Python's `exec()` with no authentication and no sandboxing. This is not an edge-case oversight; it reflects a systemic design pattern in AI platform development where code execution is treated as a feature rather than a critical attack surface.
- CVE-2026-33017 is the fourth critical unauthenticated or near-unauthenticated RCE vulnerability discovered in Langflow since early 2025, indicating a recurring failure mode rather than isolated defects. Similar vulnerabilities have been identified in n8n, Lightning AI Studios, and AI development environments across the ecosystem.
- All organizations running Langflow should treat this as an immediate incident-response priority. The patched release is version 1.9.0. Instances that cannot be upgraded immediately must be isolated from public network access.

Background

Langflow is a low-code visual framework for building AI agents, retrieval-augmented generation (RAG) pipelines, and agentic workflows. Developed originally by Logspace and subsequently acquired by DataStax — which has since been absorbed into IBM — Langflow provides a drag-and-drop interface in which developers connect LLM providers, vector databases, APIs, and arbitrary Python logic into deployable pipelines [1][2]. The platform has achieved substantial adoption in enterprise and research settings, with over 140,000 GitHub stars [2] and integrations supported by major enterprise vendors.

The fundamental value proposition of Langflow — and of the low-code AI platform category it represents — is the ability to compose and execute complex AI workflows without writing production code. This convenience carries an inherent tension: to be useful, these platforms must execute user-supplied code and make that capability accessible through web interfaces. The design challenge of securing this execution surface has proven consistently underestimated across the category.

Langflow versions are distributed as a Python package via PyPI and as Docker images. Prior to the March 2026 patch wave, the current production release was 1.8.1, itself a partial patch for the March 17 advisory. The fully remediated release, version 1.9.0, shipped in the days following the advisory [3].

Security Analysis

CVE-2026-33017: Technical Root Cause

The vulnerability resides in a single API endpoint: `POST /api/v1/build_public_tmp/{flow_id}/flow`. This endpoint exists to support Langflow's public flow-sharing feature, which allows users to share pipeline configurations with collaborators who may not have authenticated accounts on the platform. Because the feature is intended for public access, the endpoint carries no authentication requirement by design [4][5].

The flaw arises from how the endpoint handles an optional `data` parameter. When this parameter is present in the request body, the endpoint substitutes the caller-supplied flow data for the stored flow data that would normally be retrieved from the database. Flow definitions in Langflow can include custom Python code nodes, which the platform executes to enable arbitrary business logic within a pipeline. When the endpoint processes the caller-supplied flow data, those embedded Python nodes are passed directly to Python's `exec()` function with no validation, no sandboxing, and no privilege separation [4][5][6].

The consequence is complete. An attacker with no credentials and no prior relationship with the target instance can send a single HTTP POST request containing a crafted flow definition, and the server will execute any Python code embedded in that definition under the process privileges of the Langflow application. In common Docker-based deployments, the Langflow process runs as root or under a high-privilege user, giving it broad access to environment variables and files on the host, including any credentials stored in `.env` files or mounted volumes.

Researcher Aviral Srivastava, who discovered and reported the vulnerability on February 26, 2026, described the root cause as the absence of any verification that the flow data supplied by the caller matches the stored flow for the given `flow_id` [5]. The endpoint was designed to retrieve data from the database and execute it — the authentication gap created an implicit assumption that only legitimate, pre-stored flows would reach the execution path. Accepting caller-supplied data without authentication or validation invalidates that assumption entirely.

The vulnerability is assigned CVSS 4.0 score 9.3 (Critical) with vector `AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:L/SI:L/SA:L`, and CVSS 3.1 score 10.0 with vector `AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H` [3][7]. The GitHub Security Advisory identifier is GHSA-vwmf-pq79-vjvx. The CWEs implicated are CWE-95 (Eval Injection) and CWE-306 (Missing Authentication for Critical Function) [3].

Exploitation Timeline and Observed Threat Activity

The advisory for CVE-2026-33017 was published on March 17, 2026, alongside the partial patch in Langflow 1.8.1. Sysdig Threat Research Team observed active exploitation activity within 20 hours of the advisory's publication — a window that predates any publicly released proof-of-concept exploit [8][14][15].

Sysdig researchers observed three operationally distinct activity patterns across six source IP addresses. Four IP addresses conducted automated internet-wide scanning, consistent with automated tools probing for vulnerable Langflow instances at scale. A fifth IP address conducted active reconnaissance with pre-staged attack infrastructure, indicating deliberate preparation. A sixth IP address performed data exfiltration, focusing on credential harvesting from `.env` files and `/etc/passwd` [8]. Attacker infrastructure included payloads staged at `173.212.205[.]251:8443`.

The 20-hour exploitation window illustrates a trend that has become increasingly pronounced across the AI platform vulnerability class: sophisticated threat actors monitor CVE advisories and are capable of independently reconstructing exploits from textual descriptions, without requiring published proof-of-concept code. Security teams should not treat the absence of public exploit code as a meaningful delay before exploitation begins.

The Langflow Vulnerability Pattern

CVE-2026-33017 is not an isolated incident. The following table summarizes the critical vulnerabilities identified in Langflow since 2025, all of which involve code execution or credential compromise:

CVE	CVSS	Type	Affected Versions	Fixed In
CVE-2025-3248	9.8	Unauth RCE via code validation endpoint	< 1.3.0	1.3.0
CVE-2025-34291	9.4	CORS/CSRF chain → account takeover + RCE	≤ 1.6.9	1.7.x
CVE-2026-27966	9.8	Prompt injection → Python REPL via CSV Agent	< 1.8.0	1.8.0
CVE-2026-33017	9.3/10.0	Unauth RCE via public flow build endpoint	≤ 1.8.1	1.9.0
CVE-2026-33309	9.9	Arbitrary file write via v2 API (low-privilege auth required)	1.2.0–1.8.x	1.9.0

Each of these vulnerabilities reflects a structurally similar failure: code execution pathways that exist by design are insufficiently protected by authentication, authorization, or sandboxing controls. CVE-2025-3248 involved an endpoint explicitly named "validate/code" that executed the submitted Python by parsing it — Python evaluates decorator expressions at parse time, so the boundary between validation and execution was illusory [9][10]. CVE-2025-34291 required a three-weakness chain: permissive CORS, missing CSRF protection with overly permissive cookie attributes, and an authenticated code execution endpoint — but in combination these produced unauthenticated RCE via a malicious webpage [11]. CVE-2026-27966 emerged because the CSV Agent node hardcoded `allow_dangerous_code=True` in LangChain, unconditionally exposing a Python REPL that prompt-injection attacks could invoke directly [12]. CVE-2026-33309, while requiring low-privilege authentication, allowed path traversal in multipart uploads to write arbitrary files to the host filesystem, enabling code execution through planted configuration or plugin files [7].

The pattern that emerges across this vulnerability history is a failure to treat code execution surfaces as adversarial attack surfaces. In each case, the code execution capability was intentional and legitimate; the security failure was an absence of defense-in-depth controls — authentication, sandboxing, input validation, and least-privilege execution — around those capabilities.

The Broader AI Platform Attack Class

Low-code and no-code AI development platforms are converging on architectures that must execute user-supplied code, call external APIs, and interact with LLM providers — all through web-accessible interfaces. The vulnerability patterns documented here suggest that the security maturity of this category has not kept pace with its rapid adoption.

Platform	Vulnerability	CVSS	Type
Langflow	CVE-2026-33017	9.3/10.0	Unauth RCE, public endpoint code injection
Langflow	CVE-2026-27966	9.8	Prompt injection → Python REPL
n8n	CVE-2025-68613, CVE-2026-21877	Critical	Multiple RCE, full platform takeover [18]
Lightning AI Studios	(no CVE)	9.4	Hidden URL parameter → root RCE
AI IDEs ("IDEsaster")	30+ flaws	Various	Prompt injection + RCE across Cursor, Windsurf, GitHub Copilot [17]

The Lightning AI Studios vulnerability is particularly instructive: a hidden URL parameter that researchers discovered through source code review could be used to obtain root-level code execution on the remote compute backing a user's AI development session — a defect that arose because the feature was designed for internal tooling and its attack surface was not reconsidered when it became publicly accessible [13]. This parallels the Langflow pattern precisely.

AI platforms are an attractive target not merely because of their attack surface, but because of what they hold. Langflow instances are commonly configured with API keys for LLM providers, database connection strings, and cloud credentials — making credential exfiltration a primary post-exploitation risk, as the Sysdig analysis confirmed [8]. A compromised Langflow instance can serve as a pivot point into the organization's broader AI infrastructure, given the breadth of credentials typically accessible from the host.

Recommendations

Immediate Actions

Organizations should treat CVE-2026-33017 as an active incident requiring same-day response given the confirmed in-the-wild exploitation. Any Langflow instance running version 1.8.1 or earlier that is network-accessible should be considered potentially compromised.

The immediate priority is to upgrade all Langflow instances to version 1.9.0 or later. If immediate upgrade is operationally infeasible, the instance must be isolated from all public and untrusted network access using firewall rules or a reverse proxy that requires authentication before routing requests to the application. The specific endpoint `POST /api/v1/build_public_tmp/` should be blocked at the network perimeter for any instance that cannot be upgraded immediately.

For any Langflow instance that was publicly accessible prior to the March 17, 2026 advisory, security teams should initiate a full credential rotation covering all API keys, database passwords, cloud provider credentials, and service tokens stored in environment variables or configuration files on the host. Forensic review of outbound network connections, process execution logs, and file system changes should be conducted to identify evidence of the exploitation patterns Sysdig documented, particularly outbound connections to external IP addresses and access to `.env` or credential files.

Short-Term Mitigations

Organizations should audit all AI development platform deployments for exposure patterns similar to those in the Langflow vulnerability history. The key questions are whether these platforms are accessible from untrusted networks, whether code execution endpoints are authenticated, and whether the execution environment is isolated from production credentials and systems.

Network segmentation is a high-value control for AI development platforms. These tools are typically used in development and research workflows, not in production service paths, yet they are frequently deployed with the same network accessibility as production applications. Placing AI development platforms in isolated network segments with outbound egress controls limits the post-exploitation blast radius significantly. The Sysdig observation that attackers targeted `.env` files and `/etc/passwd` underscores that credential exposure is the primary post-exploitation risk.

Web application firewall (WAF) rules targeting the specific endpoint patterns associated with these vulnerabilities provide a compensating control while upgrade processes are completed. FortiGuard has published IPS signatures for CVE-2025-3248 [9] and has issued an outbreak alert for the broader

Langflow unauthenticated RCE class [16], both of which can serve as references for WAF rule construction.

Dependency and runtime monitoring should be applied to all Langflow deployments. Because Langflow executes arbitrary Python at runtime, behavioral anomaly detection — monitoring for unexpected process spawning, unusual outbound network connections, or file access patterns inconsistent with normal operation — offers detection coverage that signature-based controls may miss, particularly for novel exploit variants not yet represented in signature databases.

Strategic Considerations

The Langflow vulnerability series represents a systemic challenge for the low-code AI platform category that warrants strategic attention beyond the immediate patch cycle. Organizations that have standardized on Langflow or similar platforms should evaluate the security architecture of those platforms as a first-class vendor selection criterion going forward.

Specifically, security teams should assess whether AI development platforms provide: authenticated access controls on all code execution endpoints, including those designed for sharing; execution sandboxing that prevents code running within the platform from accessing host credentials and filesystem paths outside the application's working directory; a principle of least privilege for the application process itself; and a documented vulnerability disclosure program with a history of timely response. It is worth noting that the Langflow CVE history, while concerning, shows that the project responds to disclosures with patches — the recurring challenge lies in the architectural posture that generates these vulnerabilities, which requires deeper remediation than individual patches alone can address.

The emergence of prompt injection as an RCE vector (CVE-2026-27966) deserves particular attention at the strategic level. As AI platforms integrate LLM-powered components that process arbitrary external data — documents, web content, user inputs — and expose Python execution tools to those LLMs, the attack surface expands in ways that traditional code review and authentication controls do not address. Organizations should evaluate whether LLM-powered components in their AI pipelines have access to code execution capabilities, and if so, whether those capabilities can be invoked through data the LLM processes.

CSA Resource Alignment

This research note connects directly to several active Cloud Security Alliance frameworks and working groups.

MAESTRO (Agentic AI Threat Modeling): The Langflow vulnerability class exemplifies the MAESTRO concern around agentic AI systems that combine code execution, external API access, and LLM inference within a single attack surface. MAESTRO's threat model addresses AI agent execution environments and tool and API integration layers — both of which are directly implicated by the conditions that make Langflow deployments high-value targets. The prompt-injection-to-RCE vector (CVE-2026-27966) is a concrete example of the cross-layer attack chains MAESTRO characterizes.

AICM (AI Controls Matrix): The vulnerability patterns described in this note correspond to control deficiencies in the AICM's Application Security and Identity and Access Management domains. Controls addressing authenticated access to code execution interfaces, runtime sandboxing requirements, and credential separation for AI workflow platforms are directly applicable. Organizations using the AICM for AI governance should review their control posture against the execution environment security requirements it specifies.

CCM (Cloud Controls Matrix): The credential exfiltration risk highlighted by the Sysdig exploitation analysis maps to the CCM's Cryptography and Key Management (CEK) and Identity and Access Management (IAM) domains. The observation that attackers prioritize `.env` files and service credentials underscores that CCM controls governing secrets management and least-privilege access must extend to AI development platform deployments, which are frequently treated as development infrastructure rather than security-relevant systems.

Zero Trust Architecture Guidance: The most effective architectural control for limiting the blast radius of AI platform compromises is network segmentation consistent with Zero Trust principles — specifically, ensuring that AI development platforms cannot reach production credential stores or service APIs through implicit network trust. CSA's Zero Trust guidance on microsegmentation and workload identity is directly applicable to AI platform deployment architectures.

AI Organizational Responsibilities: The recurring nature of critical vulnerabilities in AI development platforms raises questions about vendor security posture that procurement and vendor management processes should address. CSA's guidance on AI organizational responsibilities includes vendor assessment criteria that security teams should apply when evaluating low-code AI platform vendors.

References

[1] DataStax, "DataStax Acquires Logspace, Creator of Langflow," DataStax Blog, 2024. https://www.datastax.com/blog/datastax-acquires-logspace-to-aid-gen-ai-app-development [Note: URL redirects to IBM's DataStax product page following DataStax's subsequent acquisition by IBM; the acquisition fact is confirmed by multiple secondary sources.]

[2] Langflow Project, "What is Langflow?" Langflow Documentation, 2026. https://docs.langflow.org/

[3] GitLab Advisory Database, "CVE-2026-33017 – Langflow Code Injection RCE," March 2026. https://advisories.gitlab.com/pkg/pypi/langflow/CVE-2026-33017/

[4] Aviral Srivastava, "CVE-2026-33017: How I Found an Unauthenticated RCE in Langflow by Reading the Code They Already Had," Medium, March 2026. https://medium.com/@aviral23/cve-2026-33017-how-i-found-an-unauthenticated-rce-in-langflow-by-reading-the-code-they-already-dc96cdce5896

[5] CVEfeed.io, "CVE-2026-33017 Detail," March 2026. https://cvefeed.io/vuln/detail/CVE-2026-33017

[6] Wiz, "CVE-2026-33017 – Vulnerability Database," March 2026. https://www.wiz.io/vulnerability-database/cve/cve-2026-33017

[7] GitLab Advisory Database, "CVE-2026-33309 – Langflow Arbitrary File Write RCE," March 2026. https://advisories.gitlab.com/pkg/pypi/langflow/CVE-2026-33309/

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

[9] FortiGuard Labs, "Langflow Missing Authentication Vulnerability – Threat Signal Report," FortiGuard, May 2025. https://www.fortiguard.com/threat-signal-report/6085/langflow-missing-authentication-vulnerability

[10] Keysight Technologies, "CVE-2025-3248: Langflow Unauthenticated Code Validation," Keysight Blog, June 2025. https://www.keysight.com/blogs/en/tech/nwvs/2025/06/29/cve-2025-3248-langflow-unauthenticated-code-validation

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

[12] GitLab Advisory Database, "CVE-2026-27966 — Langflow CSV Agent RCE," February 2026. https://advisories.gitlab.com/pkg/pypi/langflow/CVE-2026-27966/

[13] Noma Security, "Noma Research Discovers RCE Vulnerability in AI Development Platform Lightning AI," Noma Security Blog, 2025. https://noma.security/blog/noma-research-discovers-rce-vulnerability-in-ai-development-platform-lightning-ai/

[14] The Hacker News, "Critical Langflow Flaw CVE-2026-33017," March 2026. https://thehackernews.com/2026/03/critical-langflow-flaw-cve-2026-33017.html

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

[16] FortiGuard Labs, "Langflow Unauthenticated RCE Outbreak Alert," FortiGuard, 2026. https://fortiguard.fortinet.com/outbreak-alert/langflow-unauth-rce

[17] The Hacker News, "Researchers Uncover 30+ Flaws in AI Development Environments ('IDEsaster')," December 2025. https://thehackernews.com/2025/12/researchers-uncover-30-flaws-in-ai.html

[18] CSO Online, "Critical RCE Flaw Allows Full Takeover of n8n AI Workflow Platform," 2026. https://www.csoonline.com/article/4113980/critical-rce-flaw-allows-full-takeover-of-n8n-ai-workflow-platform.html