CVE-2026-33017: Unauthenticated RCE Hits Langflow AI Pipelines

Active Exploitation Within 20 Hours of Disclosure; AI API Keys Exfiltrated

Unofficial AI-assisted Research

Cloud Security Alliance AI Safety Initiative

2026-03-21

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

Key Takeaways

- CVE-2026-33017 is a CVSS 10.0 unauthenticated remote code execution vulnerability in Langflow, the open-source visual framework for building LLM-powered agents and RAG pipelines, affecting all versions up to and including 1.8.1 [1][2].
- The vulnerability combines missing authentication on a public-facing API endpoint with unsafe use of Python's `exec()` function, enabling any unauthenticated attacker to achieve full server compromise with a single HTTP POST request [3].
- Active exploitation was observed within approximately 20 hours of the public advisory, following a three-phase attack pattern culminating in the bulk exfiltration of AI provider API keys for OpenAI and Anthropic, AWS credentials, and database passwords from compromised instances [4].
- The fixed version, Langflow 1.9.0, was released concurrently with the advisory. Organizations running any prior version must upgrade immediately and treat all secrets stored on affected instances as compromised [2][5].
- CVE-2026-33017 is part of a larger, systemic pattern of unsafe code execution design in Langflow: the platform has accumulated at least four distinct RCE-class vulnerabilities since 2025, one of which (CVE-2025-3248) is listed on CISA's Known Exploited Vulnerabilities catalog [6][7].

Background

Langflow is an open-source, Python-based visual framework for composing large language model pipelines, autonomous agents, and retrieval-augmented generation (RAG) applications. It provides a drag-and-drop interface for wiring together AI components — LLM providers, vector databases, document loaders, tool integrations — into deployable workflows. Originally developed as an independent project, Langflow was acquired by DataStax, which was itself announced as an IBM acquisition in February 2025. Langflow is actively used in production agentic systems across a range of enterprise deployments, making its security posture consequential for organizations relying on it for AI workflow orchestration [8].

The vulnerability was discovered on February 26, 2026, by independent security researcher Aviral Srivastava, who identified it by reviewing the code diff of a prior patch — a technique that exploits the common pattern of incomplete fixes leaving structurally similar vulnerabilities in adjacent code paths [3]. Srivastava disclosed the vulnerability through responsible disclosure channels, and a public advisory was published on March 17, 2026 [1][3]. Despite the absence of a public proof-of-concept exploit at the time of disclosure, threat actors had reconstructed functional exploits directly from the advisory text within hours, with the first automated scanning activity observed at approximately the 20-hour mark [4].

Security Analysis

Vulnerability Mechanics

CVE-2026-33017 is rooted in two distinct weaknesses operating in combination. The first is CWE-306 (Missing Authentication for Critical Function): the endpoint `POST /api/v1/build_public_tmp/{flow_id}/flow` was designed to allow unauthenticated users to build flows marked as "public," a feature intended to support embedded or shared pipelines. The second is CWE-95 (Improper Neutralization of Directives in Dynamically Evaluated Code): when an optional `data` parameter is supplied in the request body, the endpoint uses the attacker-controlled payload to construct the flow graph rather than fetching the stored database record [1][3].

Because Langflow executes Python code within flow node definitions using Python's native `exec()` function without any sandboxing, containerization, or safe evaluation context, an attacker who controls the `data` parameter can embed arbitrary Python code that executes with the privileges of the Langflow server process. No authentication token, session cookie, or prior enumeration of a valid `flow_id` is required — the endpoint accepts the attacker-supplied `data` value unconditionally, making exploitation highly reliable against any network-accessible Langflow instance [3][4].

The CVSS 3.1 base vector `AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H` assigns this vulnerability a score of 10.0 [1]. Some secondary sources, including the CVSS 4.0 score assigned by GitHub, Inc., cited 9.3 [2]; the GitLab advisory database, which reflects the upstream CVSS 3.1 scoring, resolves to 10.0 under the above vector [1].

Exploitation Timeline and Attack Pattern

Sysdig's Threat Research Team documented exploitation activity across a 48-hour window following disclosure, organized into three distinct operational phases [4].

The initial phase, beginning roughly 20 hours after the public advisory, consisted of mass automated scanning using the `nuclei` vulnerability scanner routed through proxy chains. Payloads in this phase executed benign system commands (`id`) and exfiltrated results via DNS callbacks to `interactsh`-based infrastructure — a technique characteristic of broad-spectrum vulnerability hunters establishing target inventories [4].

The second phase, overlapping with and extending the first, involved custom Python exploit scripts conducting active reconnaissance on confirmed-vulnerable hosts. Attackers enumerated directory listings, read `/etc/passwd`, and staged payloads from pre-positioned infrastructure at `173.212.205[.]251:8443`. This phase indicates a deliberate escalation from opportunistic scanning to targeted intrusion [4].

The third and most consequential phase involved systematic credential harvesting. Attackers executed Python code to dump environment variables and extract `.env` files from the Langflow working directory. These files routinely contain the API keys and connection strings that Langflow users configure to connect the platform to LLM providers and downstream services. Confirmed exfiltrated data categories included OpenAI API keys, Anthropic API keys, AWS access credentials, and database passwords. Six unique source IPs were identified across the window, with coordinated behavior suggesting shared command-and-control infrastructure at `143.110.183[.]86:8080` [4].

The speed of this escalation — from automated scanning to targeted credential exfiltration within 48 hours of advisory publication — is consistent with a deliberate, multi-stage intrusion strategy rather than opportunistic scanning alone. AI pipeline credentials present amplified downstream risk beyond what traditional credential exfiltration entails. API keys for LLM providers such as OpenAI and Anthropic can be resold on criminal markets, used to generate synthetic content at victim expense, or repurposed for supply-chain attacks against the operator's downstream AI services — risks that industry analysis of underground market activity has documented in the context of prior LLM API key theft campaigns.

Systemic Context: Langflow's Pattern of Unsafe Code Execution

CVE-2026-33017 is not an isolated incident. A review of the Langflow CVE history reveals a persistent architectural tendency to pass user-influenced data to Python's `exec()` function or equivalent execution contexts without adequate boundary enforcement. CVE-2025-3248, a similarly unauthenticated RCE via the `/api/v1/validate/code` endpoint, was added to CISA's Known Exploited Vulnerabilities catalog in May 2025 and was used in the wild by the Flodrix botnet campaign [6][7]. CVE-2026-27966, disclosed on the same date Srivastava discovered CVE-2026-33017, exploited a hardcoded `allow_dangerous_code=True` parameter in the CSV Agent node to expose a Python REPL to prompt-injected commands [9]. CVE-2026-33309, also fixed in the 1.9.0 release alongside CVE-2026-33017, enables arbitrary file write and RCE via a path traversal bypass in the v2 files API — an incomplete fix for earlier CVE-2025-68478 [10][11].

CVE	Attack Type	Auth Required	CVSS	Fixed In
CVE-2025-3248	RCE via code validation endpoint	None	9.8	1.3.0
CVE-2026-27966	RCE via prompt injection in CSV Agent	None	9.8	1.8.0
CVE-2026-33017	RCE via public flow build endpoint	None	10.0	1.9.0
CVE-2026-33309	Arbitrary file write → RCE via path traversal	None	9.9	1.9.0

This pattern suggests a systemic architectural tendency rather than coincidental individual oversights — a judgment consistent with published security research on platforms that treat runtime code evaluation as a first-class design feature. Platforms that use `exec()` or equivalent runtime code evaluation as a design primitive, rather than as a carefully guarded implementation detail, create an attack surface that repeatedly manifests as critical vulnerabilities regardless of how many individual instances are patched. The architectural remediation required is not another point fix but a fundamental re-evaluation of whether user-visible, network-accessible endpoints should ever be permitted to drive arbitrary code execution.

Recommendations

Immediate Actions

Organizations running Langflow in any capacity should treat this as an emergency response situation. The first priority is upgrading to Langflow 1.9.0, which removes the `data` parameter from the vulnerable endpoint and forces flow building to use only database-stored records [5]. Any instance that was running Langflow ≤ 1.8.1 and was network-accessible — including instances behind a VPN or on internal networks reachable from compromised hosts — should be treated as potentially compromised. All API keys, database credentials, and secrets stored in `.env` files or environment variables on those hosts must be rotated immediately, with revocation completed before the rotated secrets are placed in use [4][5].

Security teams should review logs from the March 17–21, 2026 window for HTTP POST requests to the `/api/v1/build_public_tmp/` path. Indicators of compromise include outbound DNS resolution queries to `*.oast.fun` or `*.interactsh.com` domains, unexpected outbound connections to `173.212.205[.]251` or `143.110.183[.]86`, and process execution of commands such as `id`, `env`, or `cat /etc/passwd` from the Langflow server process [4].

Short-Term Mitigations

Organizations that cannot immediately upgrade should restrict network access to Langflow instances at the firewall or reverse proxy layer, blocking all unauthenticated access to the `/api/v1/build_public_tmp/` endpoint path. Where Langflow is deployed as an internet-facing service, it should be placed behind an authentication proxy that requires valid credentials before forwarding any request to the application. The "public flows" feature should be disabled at the configuration level if it is not actively required [5].

All Langflow deployments should be inventoried, including development, staging, and test instances, which are frequently overlooked in upgrade campaigns despite often sharing the same credential pools as production environments. Container orchestration systems (Kubernetes, ECS, Docker Compose) should be checked to ensure that Langflow service ports are not inadvertently exposed beyond their intended network boundary.

Strategic Considerations

The Langflow vulnerability pattern warrants broader strategic consideration for organizations building AI pipeline infrastructure. Any platform that enables visual composition of AI workflows — including Langflow alternatives such as n8n, Flowise, or Apache Airflow with LLM extensions — may present similar risks if it uses server-side code execution as a design primitive. Security architects evaluating AI pipeline platforms should specifically assess whether the platform isolates code execution in sandboxed environments (containers, isolated processes, WebAssembly runtimes) or relies on in-process `exec()` calls, and should require documented evidence of security review for any code-execution pathways exposed to network-reachable endpoints.

For organizations that must continue operating Langflow, the principle of least privilege should govern all deployment decisions. Langflow server processes should run as non-root users with minimal filesystem permissions, and their network egress should be restricted to explicitly required destinations. Secrets required by Langflow should be managed through dedicated secrets management infrastructure (AWS Secrets Manager, HashiCorp Vault, Kubernetes Secrets with appropriate RBAC) rather than exposed as plaintext environment variables, which are trivially readable from arbitrary code execution contexts.

As of this advisory, CVE-2026-33017 has not yet been added to the CISA Known Exploited Vulnerabilities catalog. Given confirmed active exploitation, a KEV listing should be anticipated; federal agencies and contractors operating under BOD 22-01 should monitor the catalog and treat any listing as triggering the standard 14-day remediation deadline. Regardless of KEV status, the confirmed exploitation severity warrants treating this as a priority-zero remediation.

CSA Resource Alignment

This advisory intersects with several foundational CSA frameworks and research initiatives that provide structural guidance for understanding and responding to the risks exposed by CVE-2026-33017.

The CSA **MAESTRO** (Multi-Agent Environment for Systematic Threat and Risk Operations) framework for agentic AI threat modeling is directly applicable. Langflow's architecture — where agent nodes communicate over internal APIs, external tools are invoked through configurable integrations, and orchestration logic is expressed as visual flow graphs — corresponds to the multi-layer agent stack that MAESTRO was designed to assess. Applying MAESTRO's layered threat model to this attack chain, the exploit traverses from Layer 1 (Foundation Model inputs) through Layer 6 (infrastructure) by exploiting the execution boundary between the orchestration layer and the underlying server environment. Organizations modeling AI pipeline risk should use MAESTRO to enumerate the attack surface across all layers, not only the model inference endpoints.

The CSA **AI Controls Matrix (AICM)** provides a framework for assessing AI system security controls that directly addresses the control categories violated by this vulnerability. The missing authentication control maps broadly to the AICM's Identity and Access Management domain, while the absence of sandboxed code execution maps broadly to the Workload Security and Application Security domains. Organizations using the AICM as an assessment instrument for AI systems should add Langflow deployments to their assessment scope and verify compliance with relevant controls prior to any production use. Analogous guidance is also available in the NIST AI Risk Management Framework and MITRE ATLAS, which address overlapping risk categories from complementary perspectives.

The CSA **AI Organizational Responsibilities** publication series addresses governance structures that are prerequisite to detecting this class of vulnerability before exploitation. The guidance on AI tools and applications specifically covers the obligation of organizations to maintain visibility into all AI software components deployed within their environment — a capability that would have enabled affected organizations to identify vulnerable Langflow versions before March 18, 2026 and take proactive remediation action [12].

The CSA **Agentic AI Red Teaming Guide**, published in 2025, provides methodologies for testing the exact class of vulnerability exposed here: supply chain and dependency attacks against AI agent infrastructure, and the exploitation of agent-to-system boundaries where attacker-controlled data reaches code execution contexts. The guide's recommendations for testing API injection against orchestration endpoints are directly applicable to Langflow's public flow build endpoint [13].

The **STAR for AI** program offers a mechanism for AI service providers to publicly document their security posture through the AI Consensus Assessments Initiative Questionnaire (AI-CAIQ). Organizations that operate Langflow as part of a service offering should assess whether CVE-2026-33017 requires an updated STAR for AI filing, and should treat the CVE cluster as evidence that the underlying platform requires heightened scrutiny in any third-party risk assessment process.

References

[1] GitLab Advisory Database, "CVE-2026-33017 – Langflow Unauthenticated RCE via Public Flow Build Endpoint," GitLab, March 17, 2026. `https://advisories.gitlab.com/pkg/pypi/langflow/CVE-2026-33017/`

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

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

[4] 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`

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

[6] CISA, "Known Exploited Vulnerabilities Catalog – CVE-2025-3248," CISA, May 2025. `https://www.cisa.gov/known-exploited-vulnerabilities-catalog`

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

[9] GitLab Advisory Database, "CVE-2026-27966 – Langflow RCE in CSV Agent via Hardcoded allow_dangerous_code Parameter," GitLab, February 2026. `https://advisories.gitlab.com/pkg/pypi/langflow/CVE-2026-27966/`

[10] GitLab Advisory Database, "CVE-2026-33309 – Langflow Arbitrary File Write via v2 Files API Path Traversal," GitLab, March 2026. `https://advisories.gitlab.com/pkg/pypi/langflow/CVE-2026-33309/`

[11] SC Media, "Critical Langflow RCE Vulnerability Exploited Within 20 Hours," SC Media, March 2026. `https://www.scworld.com/news/critical-langflow-rce-vulnerability-exploited-within-20-hours`

[12] Cloud Security Alliance, "AI Organizational Responsibilities: AI Tools and Applications," CSA, 2025.

[13] Cloud Security Alliance, "Agentic AI Red Teaming Guide," CSA AI Safety Initiative, 2025.