Agentic AI Framework CVEs Under Active Exploitation

Langflow Remote Code Execution and LangChain/LangGraph Data Exfiltration

Unofficial AI-assisted Research

Cloud Security Alliance AI Safety Initiative

2026-03-28

Key Takeaways

- CVE-2026-33017, a CVSS 9.3 unauthenticated remote code execution vulnerability in Langflow, was exploited in the wild within 20 hours of its March 17, 2026 disclosure—attackers developed working exploits directly from the advisory description, without waiting for a public proof-of-concept [1][2].
- CISA added CVE-2026-33017 to its Known Exploited Vulnerabilities catalog on March 25, 2026, requiring federal agencies to remediate by April 8, 2026; this is the second Langflow critical RCE added to the KEV catalog since May 2025 [3][10].
- CVE-2026-34070 and CVE-2025-67644, disclosed in March 2026, together with CVE-2025-68664, originally disclosed by security researchers in December 2025 and patched in early 2026, collectively expose three separate classes of enterprise data: arbitrary filesystem files (CVE-2026-34070), environment secrets including API keys (CVE-2025-68664), and full conversation histories stored in SQLite checkpoints (CVE-2025-67644) [4][6].
- The root cause in Langflow traces directly to an architectural pattern—passing attacker-controlled flow definitions through an unsandboxed `exec()` call—that persisted across multiple vulnerability cycles despite prior patching, illustrating how fixing individual vulnerabilities cannot substitute for eliminating an unsafe design [5].
- CVE-2025-68664 in langchain-core, with a CVSS score of 9.3 (CNA assessment), enables secret exfiltration through LLM response fields, meaning a malicious prompt can cascade through serialization pipelines to leak AWS keys, database credentials, and other secrets stored as environment variables [6][9].

Background

Langflow is an open-source visual platform for composing LLM-powered applications and autonomous agent workflows. Developed originally by Logspace and now maintained as a community project, Langflow enables users to build agentic pipelines by connecting nodes representing LLMs, tools, data sources, and custom Python code into executable graph structures. Its visual interface and no-code

composition model attracted adoption among organizations beginning to pilot agentic AI workflows. That same flexibility—the capacity to accept arbitrary Python code in node definitions and execute it server-side—has made Langflow the subject of three critical CVEs in under twelve months.

LangChain and its stateful extension LangGraph represent a different segment of the agentic AI framework ecosystem. Where Langflow targets practitioners who prefer visual composition, LangChain targets developers building programmatic chains and agents in Python. LangGraph adds persistent state management and checkpoint capabilities on top of LangChain, enabling long-running agents to pause, resume, and branch across sessions. Together these libraries underpin a substantial fraction of production agentic AI deployments: langchain-core alone registered approximately 98 million monthly downloads as of late 2025 [6], and the three packages collectively accumulated more than 84 million downloads in a single week in early 2026 (LangChain: 52 million, LangChain-Core: 23 million, LangGraph: 9 million) [4]. The scale of that deployment footprint—approximately 98 million monthly downloads of langchain-core alone—means vulnerabilities in these libraries reach well beyond research and development contexts into production AI systems operating with access to enterprise data, secrets, and infrastructure.

The disclosure cluster described in this research note—spanning Langflow's third critical RCE in twelve months and three distinct exfiltration paths in LangChain and LangGraph—arrives at a moment when agentic AI deployments are transitioning from pilot to production across financial services, healthcare, and critical infrastructure sectors. The threat model for this class of vulnerability differs meaningfully from traditional application security: a successful exploit against a Langflow instance or a LangChain-backed agent pipeline typically yields not just server access but access to the entire constellation of credentials, API keys, and connected system permissions that the AI agent was provisioned with. In many observed deployments, AI agents are provisioned with broad access to justify their operational utility, and the blast radius scales with privilege.

Security Analysis

CVE-2026-33017: Langflow's Third Critical RCE in Twelve Months

The vulnerability identified as CVE-2026-33017 (CVSS 9.3, CNA assessment) exists in the public flow build endpoint at `POST /api/v1/build_public_tmp/{flow_id}/flow`, present in all Langflow versions through 1.8.1 [1]. The endpoint is designed to be unauthenticated—it exists to allow public flows to be executed without requiring the caller to hold a session—and this functional requirement created the conditions for exploitation. When a request arrives, the endpoint accepts

attacker-supplied flow data containing arbitrary Python code in node definitions. That code travels through a ten-step call chain ending in `exec(compiled_code, exec_globals)` in `src/lfx/src/lfx/custom/validate.py` [5]. No sandboxing, AST filtering, or privilege isolation applies. Assignment statements within the attacker-supplied code execute immediately during graph construction, before the flow completes a single step of its intended operation.

The structural relationship between CVE-2026-33017 and its predecessor CVE-2025-3248 is instructive. CVE-2025-3248, disclosed in May 2025 with a CVSS score of 9.8, resided in the `/api/v1/validate/code` endpoint and was remediated by adding an authentication decorator [12]. That fix addressed the symptom—a reachable execution path that lacked access control—without addressing the underlying design: the same `exec()` call pattern was replicated across multiple endpoints serving different authentication contexts. CVE-2026-33017 exploited an endpoint that was architecturally required to be public, meaning authentication could never be the correct mitigation. Security researcher Aviral Srivastava characterized the correct fix as "removing the data parameter from the public endpoint entirely, so public flows can only execute their stored (server-side) flow data" [5]. The patch in Langflow 1.9.0 reflects this principle.

Post-exploitation activity observed by Sysdig's Threat Research Team within the first 20-hour exploitation window followed a pattern indicative of pre-positioned actors rather than opportunistic scanners. Initial access was confirmed through custom Python scripts that extracted `/etc/passwd` contents to verify code execution. Subsequent payloads retrieved next-stage malware from `173.212.205[.]251:8443`, followed by systematic harvesting of environment variables, configuration files, and `.env` files containing database credentials and API keys [2]. The speed and sophistication of these operations—actors built working exploits from the advisory text alone and staged a prepared toolkit before public proof-of-concept code existed—suggests the vulnerability was treated as a high-priority target by organized threat groups.

A separate but overlapping thread connects CVE-2025-34291 (CVSS 9.4), disclosed in December 2025, to the same Langflow ecosystem. That vulnerability enabled cross-site account takeover combined with RCE by having an authenticated user visit a malicious webpage, and active exploitation was observed beginning January 23, 2026, deploying infrastructure associated with the Flodric botnet [7]. CVE-2025-3248 and CVE-2026-33017 both trace to Langflow's use of unsandboxed `exec()` for user-supplied Python—the same architectural root cause that each patch addressed differently. CVE-2025-34291 represents a structurally distinct vulnerability class (XSS-enabled privilege escalation), though the recurrence of critical exploitation across three CVEs in twelve months reflects systemic risk in the platform regardless of root cause.

CVE-2025-68664: Serialization Injection in LangChain Core

CVE-2025-68664 (CVSS 9.3, CNA assessment; NIST CVSS 3.1: 8.2) was first disclosed by the research group Cyata in December 2025 under the name LangGrinch and subsequently patched in langchain-core versions 0.3.81 and 1.2.5 [6][9]. The vulnerability originates in the `dumps()` and `dumpd()` serialization functions, which use an `lc` key in JSON structures to distinguish LangChain objects from ordinary Python dictionaries. Before the patch, these functions did not escape ordinary dictionaries containing an `lc` key, allowing an attacker to craft a data structure that the deserializer would treat as an already-serialized LangChain object. The `loads()` function, when processing such a structure, supported resolution of values from environment variables when `secrets_from_env=True` –the default configuration prior to patching. An attacker who could influence a field that passed through serialization and deserialization could thus exfiltrate environment secrets by embedding a request for their values in the injected structure.

The most consequential exploitation path involves LLM response fields. LangChain agents routinely serialize and log LLM outputs including `additional_kwargs` and `response_metadata`, fields that flow through `astream_events()`, `astream_log()`, and `RunnableWithMessageHistory`. A malicious prompt response from a compromised or attacker-controlled LLM backend, or from a user interacting directly with an exposed agent endpoint, can embed a serialization injection payload that cascades through these pipelines and extracts credentials to an attacker-controlled endpoint [6]. Twelve distinct vulnerable code flows were identified, spanning streaming, logging, caching, and hub manifest operations. The breadth of affected code paths means that partial patching or dependency pinning at the application layer provides limited assurance without upgrading langchain-core itself.

Beyond credential exfiltration, the deserialization injection mechanism enables instantiation of LangChain objects whose constructors perform network calls or file operations. Classes within langchain-core, langchain-openai, and langchain-aws are included in the deserializer's allowlist. Instantiating a `PromptTemplate` object through injection triggers Jinja2 template rendering, and instantiating connector classes can produce server-side request forgery by directing the application to make outbound calls to attacker-specified endpoints. The vulnerability thus spans a spectrum from credential leak to blind SSRF depending on what the attacker places in the injected structure and which code paths process it.

CVE-2026-34070 and CVE-2025-67644: Path Traversal and SQL Injection

The two remaining vulnerabilities in the March 2026 LangChain/LangGraph disclosure round address structurally simpler but operationally significant weaknesses. CVE-2026-34070 (CVSS 7.5) is a path traversal vulnerability in `langchain_core/prompts/loading.py` affecting the prompt-loading API [4]. The API accepts a path to a prompt template file and loads it without validating that the path resolves within an expected directory boundary. An attacker with any ability to influence the path argument—through user input, an agent tool call, or a compromised upstream component—can read arbitrary files accessible to the application process. In an agentic AI deployment where the application runs with access to configuration directories, credential stores, or private data repositories, this file read primitive translates directly into sensitive data disclosure.

CVE-2025-67644 (CVSS 7.3) affects the LangGraph SQLite checkpoint module (`langgraph-checkpoint-sqlite`), which persists agent state and conversation history to a SQLite database between sessions [4]. The vulnerability is a SQL injection flaw in metadata filter keys passed to checkpoint search operations. An attacker who can influence the filter keys—again achievable through user input or tool-call manipulation in many deployment patterns—can execute arbitrary SQL queries against the checkpoint database, exposing the complete conversation history for any session stored in that database. In enterprise deployments where a single LangGraph checkpoint store serves multiple users or sessions, this becomes a multi-tenant data isolation failure.

Patches for these two vulnerabilities are available in langchain-core version 1.2.22 and langgraph-checkpoint-sqlite version 3.0.1, respectively [4]. Unlike CVE-2025-68664, which requires upgrading langchain-core itself, the SQL injection fix requires updating `langgraph-checkpoint-sqlite` as a separate package, which may not appear as a direct dependency in many application manifests and therefore may not surface in automated vulnerability scans that report only direct requirements.

Convergence of Risk in the Agentic AI Stack

The six CVEs described in this research note collectively illustrate a convergence of risk properties that distinguishes the agentic AI framework layer from conventional application software. First, the credential blast radius is amplified: AI agent processes are typically provisioned with secrets for every external service they interact with, meaning an arbitrary file read or environment variable disclosure yields a more complete credential harvest than the same primitive against a conventional microservice. Second, the supply chain surface area is large: with langchain-core at approximately 98 million monthly downloads and tens of millions of indirect dependents through LangGraph and LangChain, unpatched versions persist across organizational boundaries long after patches are available. Third, the 20-hour exploitation window for CVE-2026-33017 illustrates how quickly high-profile agentic AI framework vulnerabilities can

be weaponized—a timeline that vulnerability management programs operating on weekly patch cycles cannot absorb. This pattern of short time-to-exploitation following public advisory has appeared in multiple recent agentic framework disclosures, suggesting that traditional patch cadences are structurally mismatched to the threat velocity of this software category.

Recommendations

Immediate Actions

Organizations running Langflow should treat CVE-2026-33017 as a zero-day requiring emergency remediation regardless of scheduled patch windows. Upgrade to Langflow 1.9.0 or later immediately. Any Langflow instance running version 1.8.1 or earlier that is reachable from the internet or from untrusted internal network segments should be considered potentially compromised. Rotate all credentials stored as environment variables or in configuration files on the affected host before restoring service, as post-exploitation activity consistently targeted environment variable exfiltration. Network access to Langflow's API port should be restricted to known, authenticated source addresses via firewall rules pending upgrade.

Organizations using LangChain or LangGraph in production should audit their installed versions against the following minimum versions: langchain-core 1.2.22 (addressing CVE-2026-34070 and CVE-2025-68664), and langgraph-checkpoint-sqlite 3.0.1 (addressing CVE-2025-67644). Applications running langchain-core in the 0.3.x legacy branch require version 0.3.81 or later. Because these packages are frequently installed as transitive dependencies rather than direct requirements, dependency lock files and software bill of materials tooling should be used to identify whether vulnerable versions are present in production environments.

Short-Term Mitigations

Where immediate patching is operationally infeasible, several mitigations can reduce exposure. For Langflow, disabling or firewalling the public build endpoint at `/api/v1/build_public_tmp/` eliminates the CVE-2026-33017 attack surface without requiring an application restart. Deploying Langflow instances with a least-privilege service account that lacks read access to sensitive configuration directories limits the value of post-exploitation file enumeration. Container-based deployments should ensure the Langflow process runs as a non-root user and that secrets are injected at runtime through a secrets manager rather than baked into environment files on disk.

For LangChain applications, explicitly setting `secrets_from_env=False` in serialization calls removes the primary credential exfiltration path for CVE-2025-68664 where full patching is delayed. Input validation on prompt template paths before passing them to the loading API reduces exploitability of CVE-2026-34070. Parameterized filter construction rather than direct key interpolation in LangGraph checkpoint queries mitigates CVE-2025-67644 at the application layer pending the package upgrade.

Strategic Considerations

The recurrence of critical RCE vulnerabilities in Langflow across three distinct CVEs in twelve months reflects a structural pattern rather than isolated defects. Organizations should evaluate whether deploying a framework that provides server-side arbitrary Python execution as a first-class feature is appropriate for their threat model without additional isolation controls. Containerization with restricted syscall profiles, network egress filtering to prevent callback to attacker infrastructure, and runtime process monitoring for unexpected child process spawning are necessary mitigating controls for any deployment that accepts user-influenced code execution paths, consistent with container security baselines recommended by NIST SP 800-190 and CIS Benchmarks for containers.

The LangChain/LangGraph vulnerability cluster highlights that trust model boundaries require explicit definition in agentic AI architectures. LLM outputs, user-supplied prompts, tool call results, and checkpoint metadata should each be treated as untrusted input throughout the application, even when the immediate consumer is an internal framework component rather than application code directly. Adopting this posture requires organizations to treat AI agent frameworks as attack surface rather than trusted infrastructure—a shift that has implications for security review processes, dependency management programs, and incident response playbooks.

At the program level, organizations deploying agentic AI at scale should establish a dedicated tracking mechanism for CVEs affecting AI framework dependencies, analogous to the tracking applied to web application frameworks and runtime libraries. The velocity of CVE filings against MCP-adjacent and agentic AI tooling—exceeding 30 filings in a sixty-day window in early 2026 [8]—exceeds the capacity of conventional vulnerability management programs to triage without a dedicated process.

CSA Resource Alignment

The vulnerabilities described in this research note map directly to threat categories defined in CSA's MAESTRO framework for agentic AI threat modeling. The Langflow RCE chain—attacker-controlled flow definitions executing through an unsandboxed interpreter—is a concrete instantiation of the Execution Hijacking threat pattern, in which an adversary gains control of the agent's execution environment by subverting the mechanism through which the agent interprets and runs instructions. The LangChain deserialization injection path, where malicious LLM output cascades through serialization pipelines to extract credentials, corresponds to MAESTRO's Memory and State Manipulation category, representing adversarial influence over the data structures the agent uses to maintain context and continuity.

CSA's AI Infrastructure and Compliance Matrix (AICM) provides the compliance mapping context for the organizational controls implied by these vulnerabilities. The AICM's controls addressing Model and Component Integrity (ensuring that AI framework dependencies are sourced from verified channels and maintained at patched versions) and Secrets and Credential Management (ensuring that credentials accessible to AI agent processes are isolated, rotated, and audited) are directly responsive to the attack patterns described here. Organizations using the CSA STAR program to assess AI vendor and product security should incorporate agentic framework CVE history into their assessment criteria, recognizing that a framework's record of security defect remediation is a material input to third-party risk evaluation.

CSA's Zero Trust guidance applies with particular force to agentic AI deployments. The principle of least-privilege access—granting each agent process only the minimum permissions required for its defined function—directly limits the blast radius of the credential exfiltration attacks enabled by CVE-2025-68664 and CVE-2026-33017. Agent processes that hold only narrowly scoped credentials yield less value to an attacker than processes provisioned with broad access for operational convenience. The Zero Trust architectural principle of verifying every access request rather than trusting ambient network position is equally applicable to the agentic layer: agent-to-service communication should authenticate with short-lived, rotatable credentials rather than static API keys stored in environment files.

References

1. 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

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

3. CISA / Help Net Security, "CISA Sounds Alarm on Langflow RCE, Trivy Supply Chain Compromise After Rapid Exploitation," Help Net Security, March 27, 2026. https://www.helpnetsecurity.com/2026/03/27/cve-2026-33017-cve-2026-33634-exploited/

4. The Hacker News, "LangChain, LangGraph Flaws Expose Files, Secrets, Databases in Widely Used AI Frameworks," March 2026. https://thehackernews.com/2026/03/langchain-langgraph-flaws-expose-files.html

5. Barrack AI, "Langflow Got Hacked Twice Through the Same exec() Call. Your AI Stack Probably Has the Same Problem," March 2026. https://blog.barrack.ai/langflow-exec-rce-cve-2026-33017/

6. Cyata, "All I Want for Christmas is Your Secrets: LangGrinch Hits LangChain Core (CVE-2025-68664)," December 2025. https://cyata.ai/blog/langgrinch-langchain-core-cve-2025-68664/

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

8. Cloud Security Alliance AI Safety Initiative, "CVE and CWE Agentic Vulnerability Catalog: Weakness Classes Introduced by Autonomous AI Agents," CSA Research Note, March 27, 2026.

9. NIST NVD, "CVE-2025-68664 Detail," National Vulnerability Database. https://nvd.nist.gov/vuln/detail/CVE-2025-68664

10. BleepingComputer, "CISA: New Langflow Flaw Actively Exploited to Hijack AI Workflows," March 2026. https://www.bleepingcomputer.com/news/security/cisa-new-langflow-flaw-actively-exploited-to-hijack-ai-workflows/

11. NIST NVD, "CVE-2025-3248 Detail," National Vulnerability Database. https://nvd.nist.gov/vuln/detail/CVE-2025-3248