INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-21 19:02 UTC

# Langflow CORS Origin Validation Error Enables RCE and Full System Compromise (CVE-2025-34291)

CVE VULNERABILITY | CRITICAL | CVSS 9.3 | CISA KEV

| | |
|---|---|
| SCC Item ID | SCC-CVE-2026-0208 |
| Type | CVE Vulnerability |
| CVE ID | CVE-2025-34291 |
| Severity | CRITICAL |
| CVSS Base Score | 9.3 |
| EPSS Score | 0.0949 (93th percentile) |
| KEV Status | Yes — CISA Known Exploited Vulnerability (due: 2026-06-04) |
| Affected Products | Langflow (Langflow), specific patched version not confirmed from available sources; all versions with permissive CORS and SameSite=None refresh token cookie configuration are affected |
| Published | 2026-05-21 |
| Discovery Source | Cisa Kev |

## Executive Summary

A critical vulnerability in Langflow, an AI agent workflow platform, allows attackers to hijack authenticated user sessions by tricking users into visiting a malicious webpage. Once session tokens are stolen, attackers gain full access to the platform's API and can execute arbitrary code on the underlying system. This vulnerability is confirmed actively exploited and appears on the CISA Known Exploited Vulnerabilities catalog, requiring immediate action from any organization running Langflow.

## Technical Analysis

CVE-2025-34291 is an origin validation error (CWE-346) in Langflow caused by an overly permissive CORS policy combined with a refresh token cookie configured as SameSite=None (CWE-565). This combination enables cross-site request forgery-class attacks (CWE-352): a malicious origin can issue credentialed cross-origin requests to the Langflow refresh token endpoint, silently obtain valid session tokens, and call authenticated API endpoints without user awareness. Because Langflow executes user-defined code and LLM pipelines by design, authenticated API access translates directly to arbitrary code execution (MITRE T1059) and full system compromise (T1190, T1548). Token theft maps to T1550.001 (Use Alternate Authentication Material:

Application Access Token). CVSS base score is 9.3 (Critical); EPSS score is 0.09488 at the 92.9th percentile, indicating elevated exploitation likelihood relative to the broader CVE population. Patch availability and affected version details should be confirmed from the Langflow security advisory before final remediation planning. Consult the upstream Langflow advisory and the CISA KEV catalog entry (due date 2026-06-04) for the latest patch and mitigation status. Sources: NVD (https://nvd.nist.gov/vuln/detail/CVE-2025-34291), CISA KEV (https://www.cisa.gov/known-exploited-vulnerabilities-catalog).

## Action Checklist

1. Step 1: Containment. Immediately restrict public internet access to all Langflow instances. If Langflow must remain accessible, place it behind a VPN or zero-trust access gateway so only authenticated internal users can reach it. Block unauthenticated or cross-origin requests to the /api/*/refresh, /api/*/login, and /api/*/auth/token endpoints at the WAF or reverse proxy layer. Consult Langflow's API documentation for the complete list of token-issuing endpoints. Reference: NIST AC-17 (Remote Access), CIS 4.4 (Implement and Manage a Firewall on Servers).
2. Step 2: Detection. Review web server and application logs for anomalous cross-origin requests to Langflow's token refresh endpoint, particularly requests with an Origin header that does not match your organization's authorized domains. Look for session token issuance events followed immediately by API calls from different IP addresses or user agents. Enable audit logging per NIST AU-2 (Event Logging) and AU-12 (Audit Record Generation) if not already active. Check SIEM for T1550.001 indicators: token reuse from geographically or behaviorally inconsistent sources. Reference: CIS 8.2 (Collect Audit Logs).
3. Step 3: Eradication. Apply the vendor-supplied patch when confirmed available from Langflow's official security advisory. Until a patch is available, configure Langflow's CORS policy to allowlist only explicitly authorized origins and change the refresh token cookie from SameSite=None to SameSite=Strict or SameSite=Lax. Rotate all active session tokens and refresh tokens to invalidate any that may have been stolen. Reference: NIST CM (Configuration Management), D3-CRO (Credential Rotation), D3-CH (Credential Hardening).
4. Step 4: Recovery. After applying configuration changes or the vendor patch, validate that cross-origin requests from unauthorized origins are rejected with a 403 or CORS error. Confirm refresh token cookies carry SameSite=Strict or Lax attributes. Re-audit active user sessions and terminate any sessions created during the exposure window. Monitor Langflow API logs for anomalous code execution or pipeline invocation activity for at least 30 days post-remediation. Reference: NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST SI-4 (System Monitoring).
5. Step 5: Post-Incident. Conduct a configuration review of all AI/ML workflow platforms for CORS policy and cookie security attribute gaps. Update your vulnerability management process to track CISA KEV due dates as hard remediation deadlines. Evaluate whether Langflow and similar code-executing platforms require additional network segmentation and least-privilege API scoping. Reference: NIST AC-6 (Least Privilege), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), D3-UAP (User Account Permissions).

## IR / Forensic Enrichment

| | |
|---|---|
| **Triage Priority** | IMMEDIATE |

| | |
|---|---|
| **Escalation Criteria** | Escalate to CISO and legal counsel immediately if Langflow pipeline execution logs show any flow_run records during the exposure window — particularly any involving LLM tool calls, shell execution nodes, or file system access — as these indicate the CORS session hijack was leveraged for RCE on the host, triggering breach notification assessment obligations if the system processed PII, PHI, or held credentials for downstream systems. |
| **Recovery Notes** | After CORS hardening and token rotation are confirmed, validate the underlying Langflow host for signs of post-exploitation by checking for new cron jobs (`crontab -l; ls -la /etc/cron*`), new user accounts (`getent passwd \| awk -F: '$3 >= 1000'`), and recently modified files in the Langflow application directory (`find /path/to/langflow -newer /path/to/langflow/app.py -type f`) — because CVE-2025-34291 enables RCE through the platform's pipeline execution engine, not just API access. Monitor Langflow's API execution endpoint (`/api/v1/run` and `/api/v1/build`) in access logs for 30 days post-remediation, alerting on any invocations by user accounts that were active during the exposure window. Treat any confirmed pipeline execution during the exposure window as a full host compromise until host forensics are completed. |
| **Forensic Artifacts** | Langflow reverse proxy or WAF access logs (30-day retention window): filter for POST requests to /api/v1/refresh and /api/v1/login where the Origin request header is present and does not match the Langflow host's FQDN — these are the direct evidence of the CORS hijack attempt. \| Langflow application database flow_run table (SQLite langflow.db or PostgreSQL): records of all pipeline executions during the exposure window, including user_id, flow_id, input parameters, and execution status — this determines whether session hijack progressed to RCE via Langflow's code-executing pipeline nodes. \| HTTP response headers captured at the reverse proxy (specifically Set-Cookie headers on /api/v1/refresh responses): confirm whether SameSite=None; Secure was set on refresh token cookies issued during the exposure window, establishing that stolen tokens were viable for cross-origin reuse. \| Host-level process creation logs from the Langflow server during the exposure window: on Linux, check `/var/log/auth.log`, auditd logs (`ausearch -sc execve`), or Sysmon for Linux (sysmonforlinux) Event ID 1 for processes spawned as the Langflow service account — Langflow's pipeline engine can invoke shell commands, so unexpected child processes of the Langflow Python process are indicators of RCE. \| Langflow application log file (`~/.langflow/langflow.log` for pip installs; container stdout for Docker) filtered for ERROR and WARNING entries and any log lines referencing CORS policy decisions, token validation failures, or pipeline execution errors — these may reveal attacker probing activity or failed exploitation attempts that preceded a successful session hijack. |

### Per-Action IR Details

**Step 1: Containment — Immediately restrict public internet access to all Langflow instances. If Langflow must remain accessible, place it behind a VPN or zero-trust access gateway so only authenticated internal users can reach it. Block unauthenticated or cross-origin requests to the /refresh and /login token endpoints at the WAF or reverse proxy layer. Reference: NIST AC-17 (Remote Access), CIS 4.4 (Implement and Manage a Firewall on Servers).**

**NIST Phase:** Containment

**Reference:** NIST 800-61r3 §3.3 — Containment Strategy

**Controls:** NIST AC-17 (Remote Access), NIST AC-4 (Information Flow Enforcement), NIST SC-7 (Boundary Protection), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

**Compensating:** On NGINX or Apache reverse proxy fronting Langflow, add a location block that checks the Origin header against an explicit allowlist and returns 403 for non-matching origins on /api/v1/refresh and /api/v1/login: `if ($http_origin !~* '^https://(yourdomain\.com)$') { return 403; }`. For iptables-based host firewall, restrict Langflow's

default port (7860) to VPN subnet only: `iptables -I INPUT -p tcp --dport 7860 ! -s -j DROP`. Document the rule change with timestamp for the incident record.

**Evidence:** Before restricting access, capture a full snapshot of Langflow's active session store (if Redis-backed, run `redis-cli KEYS 'session:*'` and dump all session entries with TTLs). Export the reverse proxy or WAF access logs covering the 30-day window prior to containment — specifically retain all requests to `/api/v1/refresh` and `/api/v1/login` with their Origin, Referer, X-Forwarded-For, and User-Agent headers intact. These headers are the primary forensic record of whether cross-origin token theft occurred before containment was applied.

**Step 2: Detection — Review web server and application logs for anomalous cross-origin requests to Langflow's token refresh endpoint, particularly requests with an Origin header that does not match your organization's authorized domains. Look for session token issuance events followed immediately by API calls from different IP addresses or user agents. Enable audit logging per NIST AU-2 (Event Logging) and AU-12 (Audit Record Generation) if not already active. Check SIEM for T1550.001 indicators: token reuse from geographically or behaviorally inconsistent sources. Reference: CIS 8.2 (Collect Audit Logs).**

**NIST Phase:** Detection Analysis

**Reference:** NIST 800-61r3 §3.2 — Detection and Analysis

**Controls:** NIST AU-2 (Event Logging), NIST AU-3 (Content of Audit Records), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-12 (Audit Record Generation), CIS 8.2 (Collect Audit Logs)

**Compensating:** Without a SIEM, run this against Langflow's access log (adjust path for your deployment — commonly `/var/log/langflow/access.log` or the reverse proxy log): `grep '/api/v1/refresh' access.log | awk '{print $1, $7, $NF}' | sort | uniq -c | sort -rn` to surface high-frequency refresh attempts. Then cross-reference token issuance events with subsequent API calls using: `grep -E '(POST /api/v1/refresh|POST /api/v1/login)' access.log | awk '{print $1}' | while read ip; do grep $ip access.log | grep -v 'refresh\|login'; done` to find post-auth API activity from the same IP. For T1550.001 (Use Alternate Authentication Material: Application Access Token) detection without EDR, use GoAccess (`goaccess access.log --log-format=COMBINED`) to visualize session token reuse across divergent IPs and user agents in a browser-based report.

**Evidence:** Extract all Langflow application log entries (default location varies by deployment method — check `~/.langflow/langflow.log` for pip installs or container stdout for Docker deployments) for HTTP 200 responses to POST `/api/v1/refresh` where the Origin header is present and does not match the Langflow host's own FQDN. Capture the full HTTP request headers for each hit, including Set-Cookie response headers, to confirm whether a new refresh token with SameSite=None was issued to the cross-origin requestor. Additionally preserve any Langflow pipeline execution logs (typically in the application database — check `langflow.db` SQLite file or PostgreSQL `flow_run` table) from the same timeframe to assess whether stolen tokens were used to trigger pipeline or code execution activity.

**Step 3: Eradication — Apply the vendor-supplied patch when a confirmed patched version is published by the Langflow maintainers. Until a patch is available, configure Langflow's CORS policy to allowlist only explicitly authorized origins and change the refresh token cookie from SameSite=None to SameSite=Strict or SameSite=Lax. Rotate all active session tokens and refresh tokens to invalidate any that may have been stolen. Reference: NIST CM (Configuration Management), D3-CRO (Credential Rotation), D3-CH (Credential Hardening).**

**NIST Phase:** Eradication

**Reference:** NIST 800-61r3 §3.4 — Eradication

**Controls:** NIST CM-6 (Configuration Settings), NIST CM-7 (Least Functionality), NIST SI-2 (Flaw Remediation), NIST IA-5 (Authenticator Management), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

**Compensating:** For CORS hardening without touching Langflow source: in NGINX, add `add_header 'Access-Control-Allow-Origin' 'https://yourdomain.com' always;` and `add_header 'Access-Control-Allow-Credentials' 'false' always;` to the Langflow proxy block, and ensure no wildcard ACAO header is emitted. For cookie attribute enforcement at the proxy layer (when you cannot modify Langflow's cookie issuance), use NGINX `proxy_cookie_flags` directive: `proxy_cookie_flags refresh_token samesite=strict secure;`. For token rotation without

vendor tooling, invalidate all current sessions by rotating the Langflow SECRET_KEY environment variable (set in `.env` or Docker Compose), which invalidates all JWT-signed tokens immediately — then notify all users to re-authenticate.

**Evidence:** Before rotating tokens or applying configuration changes, export the Langflow application database snapshot (SQLite: `cp langflow.db langflow_pre_eradication_$(date +%Y%m%d).db`; PostgreSQL: `pg_dump langflow > langflow_pre_eradication_$(date +%Y%m%d).sql`) to preserve the pre-remediation session and flow execution state as forensic evidence. Document the current CORS configuration as found — retrieve it from Langflow's `config.yaml` or environment variables (`printenv | grep -i cors`) and the reverse proxy config — and retain both as timestamped artifacts for the incident record and any future regulatory inquiry.

**Step 4: Recovery — After applying configuration changes or the vendor patch, validate that cross-origin requests from unauthorized origins are rejected with a 403 or CORS error. Confirm refresh token cookies carry SameSite=Strict or Lax attributes. Re-audit active user sessions and terminate any sessions created during the exposure window. Monitor Langflow API logs for anomalous code execution or pipeline invocation activity for at least 30 days post-remediation. Reference: NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST SI-4 (System Monitoring).**

**NIST Phase:** Recovery

**Reference:** NIST 800-61r3 §3.5 — Recovery

**Controls:** NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST SI-4 (System Monitoring), NIST CM-6 (Configuration Settings), NIST CA-7 (Continuous Monitoring), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

**Compensating:** Validate CORS fix without a commercial scanner using curl: `curl -v -H 'Origin: https://evil.example.com' -H 'Cookie: refresh_token=' -X POST https:///api/v1/refresh 2>&1 | grep -E '(Access-Control|/api/v1/login -d '{"username":"test","password":"test"}' 2>&1 | grep Set-Cookie` and verify `SameSite=Strict` or `SameSite=Lax` is present. For 30-day post-remediation pipeline monitoring, set a cron job to run nightly: `grep 'flow_run\|code_execution\|/api/v1/run' /var/log/langflow/access.log | awk '$9==200' >> /var/log/langflow/post_remediation_executions.log` and review weekly for unexpected pipeline invocations.

**Evidence:** Before restoring full user access, enumerate all Langflow user accounts and their last session creation timestamps from the application database (`SELECT username, last_login, created_at FROM user WHERE last_login BETWEEN '' AND ''`) to identify accounts whose sessions were active during the exposure window — these accounts are candidates for credential reset and additional review. Preserve the full list of pipeline execution records (`SELECT id, flow_id, user_id, created_at, status FROM flow_run WHERE created_at BETWEEN '' AND ''`) to determine whether any stolen session was used to invoke Langflow's code execution capability, which could indicate a secondary compromise of the underlying host.

**Step 5: Post-Incident — Conduct a configuration review of all AI/ML workflow platforms for CORS policy and cookie security attribute gaps. Update your vulnerability management process to track CISA KEV due dates as hard remediation deadlines. Evaluate whether Langflow and similar code-executing platforms require additional network segmentation and least-privilege API scoping. Reference: NIST AC-6 (Least Privilege), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), D3-UAP (User Account Permissions).**

**NIST Phase:** Post Incident

**Reference:** NIST 800-61r3 §4 — Post-Incident Activity

**Controls:** NIST AC-6 (Least Privilege), NIST AC-4 (Information Flow Enforcement), NIST RA-5 (Vulnerability Monitoring and Scanning), NIST SI-2 (Flaw Remediation), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 4.2 (Establish and Maintain a Secure Configuration Process for Network Infrastructure)

**Compensating:** For CORS and cookie attribute auditing across AI/ML platforms without a commercial ASM tool, build a checklist-driven curl script that tests each platform's token endpoints for wildcard ACAO headers and SameSite=None cookies, running it against all internal AI platforms (n8n, Flowise, Dify, ComfyUI, etc.) on a monthly

cron schedule. For CISA KEV tracking without a vulnerability management platform, subscribe to the CISA KEV RSS feed (`https://www.cisa.gov/sites/default/files/feeds/known_exploited_vulnerabilities.json`) and use a Python script with `requests` + `difflib` to diff the JSON daily and alert via email when new entries match your software inventory maintained in a simple CSV asset register per CIS 1.1.

**Evidence:** For the lessons-learned record, compile the complete incident timeline artifact set: (1) the pre-remediation CORS configuration as found, (2) the access log extract showing the exposure window with cross-origin refresh requests, (3) the pipeline execution records from the exposure window, (4) the session database export, and (5) the post-remediation curl validation output confirming the fix. This evidence package supports both internal lessons-learned per NIST 800-61r3 §4 and any CISA KEV compliance documentation demonstrating remediation within the required due date window.

## Detection Guidance

Query web application and reverse proxy logs for requests to Langflow's token refresh endpoint (typically /api/v1/refresh or equivalent) where the Origin or Referer header does not match your organization's authorized Langflow hostname. Flag any response that issues a Set-Cookie or returns a token payload to an unauthorized origin. In your SIEM, correlate: (1) token issuance events tied to unusual origin values, (2) API calls using the issued token from a different IP or user agent within a short time window, and (3) subsequent Langflow pipeline or code execution events. Behavioral indicator: a user session that executes a pipeline or invokes the code execution API shortly after a token refresh from an unfamiliar origin is a strong signal of exploitation. No publicly confirmed IOC hashes or IPs are available from T1-tier sources as of 2026-03-04. Enable audit logging per NIST AU-2 and CIS 8.2 across all Langflow nodes before beginning log review.

## Indicators of Compromise

| Type | Value | Context | Confidence |
|---|---|---|---|
| URL | `No confirmed IOCs available from T1-tier sources at time of publication` | No specific IP addresses, domains, or file hashes associated with active exploitation have been confirmed by NVD or CISA as of this writing. Monitor vendor advisories and CISA KEV for updates. | LOW |

## Framework Mappings

### MITRE-ATTACK

- **T1550.001** — Application Access Token
- **T1059** — Command and Scripting Interpreter
- **T1190** — Exploit Public-Facing Application
- **T1548** — Abuse Elevation Control Mechanism

### NIST-800-53R5

- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection

- **SI-4** — System Monitoring
- **SI-7** — Software, Firmware, and Information Integrity
- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **AC-6** — Least Privilege
- **CM-6** — Configuration Settings
- **SC-23** — Session Authenticity
- **SI-10** — Information Input Validation
- **AT-2** — Literacy Training and Awareness

### OWASP-TOP10-2021

- **A01:2021** — Broken Access Control

### CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks

### ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities

### NIST-CSF-2

- **DE.AE-08** — Incidents are declared when adverse events meet the defined incident criteria

## MITRE ATT&CK Mapping

| Technique ID | Technique Name | Tactic |
|---|---|---|
| T1550.001 | Application Access Token | Defense-Evasion |
| T1059 | Command and Scripting Interpreter | Execution |
| T1190 | Exploit Public-Facing Application | Initial-Access |
| T1548 | Abuse Elevation Control Mechanism | Privilege-Escalation |

## Sources

| Source | URL | Tier |
|---|---|---|
| **cisa_kev** | https://www.cisa.gov/known-exploited-vulnerabilities-catalog | T1 |
| **CVE-2025-34291 Detail - NVD** | https://nvd.nist.gov/vuln/detail/CVE-2025-34291 | T1 |

| Source | URL | Tier |
|---|---|---|
| **CVE-2025-34291: Critical Account Takeover and RCE Vulnerability ...** | https://www.obsidiansecurity.com/blog/cve-2025-34291-critical-accou... | T3 |
| **CVE-2025-34291 Exploited in the Wild: LangFlow AI ... - CrowdSec** | https://www.crowdsec.net/vulntracking-report/cve-2025-34291 | T3 |
| **Langflow CORS misconfiguration enables Account Takeover and RCE** | https://github.com/advisories/GHSA-577h-p2hh-v4mv | T3 |



Generated 2026-05-21 19:02 UTC by TJS Security Command Center