INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-11 14:24 UTC

Langflow Path Traversal CVE-2026-5027 Actively Exploited, Unauthenticated File Write on ~7,000 Exposed Instances

CVE VULNERABILITY | CRITICAL | CVSS 9.5

SCC Item ID	SCC-CVE-2026-0293
Type	CVE Vulnerability
CVE ID	CVE-2026-5027, CVE-2026-0770, CVE-2026-21445, CVE-2026-33017, CVE-2025-3248
Severity	CRITICAL
CVSS Base Score	9.5
EPSS Score	0.0408 (89th percentile)
Affected Products	Langflow langflow-base < 0.8.3, Langflow application < 1.9.0; patched in v1.10.0
Published	2026-06-10T17:23:44
Discovery Source	Rss

Executive Summary

A critical path traversal vulnerability in Langflow, an open-source AI pipeline development platform, allows unauthenticated attackers to write arbitrary files to exposed servers. Approximately 7,000 publicly accessible Langflow instances are exploitable without credentials due to a default auto-login configuration that issues valid session tokens to any requester. Active exploitation has been documented, with attackers compromising Langflow AI pipelines within 20 hours of gaining access, posing serious risk to organizations running AI development infrastructure.

Technical Analysis

CVE-2026-5027 is a CWE-22 path traversal flaw in Langflow (langflow-base < 0.8.3, Langflow application < 1.9.0) with a CVSS base score of 9.5. The root cause is CWE-306 (missing authentication for critical function): Langflow's default auto-login configuration issues a valid session token from a single unauthenticated HTTP request, removing all credential barriers. A secondary CWE-434 unrestricted file upload weakness enables post-traversal arbitrary payload delivery. The vulnerability cluster includes CVE-2026-5027, CVE-2026-0770, CVE-2026-21445, CVE-2026-33017, and CVE-2025-3248; these represent related path traversal and file upload weaknesses in the same codebase, all patched in v1.10.0 through a single security update. All are exploitable through the same default auto-login authentication bypass vector. MITRE ATT&CK techniques

observed in exploitation include T1190 (Exploit Public-Facing Application), T1505.003 (Web Shell), T1059 (Command and Scripting Interpreter), T1106 (Native API), T1083 (File and Directory Discovery), and T1036 (Masquerading). MuddyWater has been identified as a threat actor associated with this campaign. Sysdig incident research documented full AI pipeline compromise within 20 hours of exploitation. Censys identified approximately 7,000 publicly exposed instances. The patch is available in Langflow v1.10.0. EPSS score is 0.041 at the 88th percentile, indicating above-average exploitation probability relative to all scored CVEs. Not currently listed in CISA KEV.

Action Checklist

1. Step 1: Containment. Immediately block public internet access to all Langflow instances. Place Langflow behind a VPN or restrict inbound access to known IP ranges at the firewall or WAF layer. If Langflow must remain externally accessible, disable the auto-login feature in configuration to eliminate the unauthenticated session token issuance vector (CWE-306). Applies to all deployments running langflow-base < 0.8.3 or Langflow application < 1.9.0. Reference NIST AC-17 (Remote Access) and CIS Controls v8 (Firewall Infrastructure).
2. Step 2: Detection. Query your asset inventory for any Langflow deployments using NIST AM-1 (Inventory of Information Systems). Search web server and application logs for unexpected POST requests to file-handling API endpoints, particularly those that did not originate from authenticated sessions prior to token issuance. Look for newly created files in application directories outside expected paths, including web-accessible directories (T1505.003 web shell indicator). Review network flow logs for outbound connections from Langflow host systems to unfamiliar external IPs following file write events (T1059 post-exploitation execution). Cross-reference with NIST AU-6 (Audit Record Review, Analysis, and Reporting). Use Censys or runZero to confirm whether your Langflow instance is externally visible.
3. Step 3: Eradication. Upgrade all Langflow deployments to v1.10.0, which contains the patch for the full vulnerability cluster (CVE-2026-5027, CVE-2026-0770, CVE-2026-21445, CVE-2026-33017, CVE-2025-3248). After upgrade, audit all files written to the application directory during the exposure window; remove any unrecognized files, particularly scripts or executables. Disable auto-login in configuration if not already done. Reference NIST SI-2 (Flaw Remediation) and CIS Controls v8 7.4 (Automated Application Patch Management). Verify uploaded or written files for unexpected file types using file integrity monitoring.
4. Step 4: Recovery. After patching, verify the running Langflow version via the application's version endpoint or package manifest. Confirm auto-login is disabled and that all session token issuance requires valid credentials. Rotate all API keys, secrets, and credentials stored in or accessible by Langflow pipelines, as attacker access to the pipeline environment may have exposed them (NIST IA-4, Identifier Management). Enable enhanced logging on the Langflow host per NIST AU-2 (Event Logging) and AU-12 (Audit Record Generation) and monitor for anomalous process execution or outbound network activity for at least 30 days post-remediation.
5. Step 5: Post-Incident. Conduct a review of AI development platform deployment standards. Assess whether auto-login defaults on any other internally deployed developer tools create similar unauthenticated access exposure. Implement a documented process to identify and remediate internet-exposed development tools using NIST SI-7 (Software, Firmware, and Information Integrity) and NIST CA-7 (Continuous Monitoring). Review access control policies for AI pipeline infrastructure against NIST AC-6 (Least Privilege) and AC-3 (Access Enforcement) to ensure pipeline execution environments cannot reach sensitive internal systems. Restrict pipeline service account permissions to only required

resources.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to CISO, legal, and privacy counsel immediately if forensic review confirms successful file writes to the Langflow host during the exposure window, any downstream pipeline credentials (LLM API keys, database connections, internal service tokens) were accessible within compromised pipelines, or if the Langflow instance processed or had access to PII/PHI/PCI data — triggering applicable breach notification obligations under GDPR, HIPAA, or state privacy statutes.
Recovery Notes	After patching to v1.10.0, perform a full re-scan of the Langflow host filesystem for web shells and persistence mechanisms before restoring external access, paying particular attention to directories served by the application's static file handler and any Python `.pth` files or `.pyc` artifacts that could indicate backdoored package installation. Monitor Sysmon EventID 1 (Process Creation) for unexpected child processes spawned by the Langflow service account, and EventID 3 (Network Connection) for outbound connections to non-LLM-provider external IPs, for a minimum of 30 days given the documented 20-hour pipeline compromise timeline. All LLM provider API keys, internal service credentials, and database connection strings accessible within Langflow pipeline definitions must be treated as fully compromised and rotated prior to resuming production pipeline execution.
Forensic Artifacts	Langflow application log (`langflow.log` or journald output for the `langflow` service unit) — search for POST requests to `/api/v1/files/upload` and path traversal sequences (`../`, `%2e%2e`, `%252e%252e`) in URI fields, with source IPs that predate any credential-based session establishment, confirming exploitation of the CWE-306 auto-login token issuance path. | Filesystem timeline of the Langflow application directory (`~/.langflow/` or `LANGFLOW_CONFIG_DIR`) — files created or modified outside the application's normal write paths (e.g., `.py`, `.php`, `.sh`, or `.so` files in static asset directories or within the flows storage path) during the exposure window are primary web shell / persistence indicators for T1505.003. | Sysmon EventID 11 (FileCreate) records on the Langflow host — specifically file creation events attributed to the Langflow process PID in directories outside the expected `flows/` and `logs/` subdirectories, which would indicate successful path traversal exploitation writing attacker-controlled content. | Network flow records or `tcpdump` pcap showing outbound connections from the Langflow host process to non-LLM-provider external IPs within the 20-hour post-access window — particularly DNS lookups for previously unseen domains or TCP sessions to ports 4444, 1337, 8080, or other common C2 ports indicative of T1059 post-exploitation execution following pipeline compromise. | Langflow pipeline definition files (JSON/YAML stored in the `flows/` directory) modified during the exposure window — these may contain attacker-injected custom component code, modified LLM endpoint URLs pointing to attacker-controlled infrastructure for credential harvesting, or added pipeline steps designed to exfiltrate API keys and secrets accessible to the pipeline execution environment.

Per-Action IR Details

Step 1: Containment — Immediately block public internet access to all Langflow instances. Place Langflow behind a VPN or restrict inbound access to known IP ranges at the firewall or WAF layer. If Langflow must remain externally accessible, disable the auto-login feature in configuration to eliminate the unauthenticated session token issuance vector (CWE-306). Applies to all deployments running langflow-base < 0.8.3 or

Langflow application < 1.9.0. Reference NIST AC-17 (Remote Access) and 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), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: On Linux-hosted Langflow, apply immediate iptables rules: `iptables -I INPUT -p tcp --dport 7860 -j DROP` (replace 7860 with your configured Langflow port), then selectively allow known IPs: `iptables -I INPUT -p tcp --dport 7860 -s -j ACCEPT`. On Windows, use `netsh advfirewall firewall add rule name='Block Langflow' protocol=TCP dir=in localport=7860 action=block`. Set the `AUTO_LOGIN` environment variable to `false` in your Langflow `.env` file or docker-compose to immediately cut the unauthenticated token issuance path without requiring a full upgrade.

Evidence: Before modifying firewall rules, capture current active network connections to the Langflow port using `ss -tnp sport = :7860` (Linux) or `netstat -anop | findstr 7860` (Windows) to document any attacker sessions in progress. Preserve a snapshot of `~/.langflow/` or the configured `LANGFLOW_CONFIG_DIR` and all files in the Langflow application directory. Record current running process list (`ps aux` / `tasklist /v`) to identify any child processes spawned by the Langflow service prior to containment.

Step 2: Detection — Query your asset inventory for any Langflow deployments using CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory). Search web server and application logs for unexpected POST requests to file-handling API endpoints, particularly those that did not originate from authenticated sessions prior to token issuance. Look for newly created files in application directories outside expected paths, including web-accessible directories (T1505.003 web shell indicator). Review network flow logs for outbound connections from Langflow host systems to unfamiliar external IPs following file write events (T1059 post-exploitation execution). Cross-reference with NIST AU-6 (Audit Record Review, Analysis, and Reporting). Use Censys or runZero to confirm whether your Langflow instance is externally visible.

NIST Phase: Detection Analysis

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

Controls: NIST AU-6 (Audit Record Review, Analysis, And Reporting), NIST AU-2 (Event Logging), NIST AU-3 (Content Of Audit Records), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 8.2 (Collect Audit Logs)

Compensating: Parse Langflow's access logs (typically at `logs/langflow.log` or via stdout/journald) for POST requests to `/api/v1/files/upload` or any endpoint containing path traversal sequences (`../`, `%2e%2e%2f`, `%252e%252e%252f`). Use: `grep -E '(POST.*files|\.\./)' /var/log/langflow/access.log`. Deploy Sysmon (EventID 11 — FileCreate) on the Langflow host to alert on new file creation outside expected application directories. Use `find /path/to/langflow -newer /path/to/langflow/app.py -type f` to identify files written after the known exposure window. For outbound C2 detection without a SIEM, run `tcpdump -i eth0 -w langflow_capture.pcap` on the host and analyze with Wireshark filtering on the Langflow process PID using `ss -tp` correlation.

Evidence: Capture Langflow application logs (stdout/journald/`langflow.log`) covering the full exposure window — specifically POST requests to `/api/v1/files/upload`, `/api/v1/flows`, and any endpoint that accepted a session token issued by the auto-login mechanism without prior credential submission. Preserve web server access logs (nginx/Apache) showing source IPs and full URI paths including query strings. Export network flow data (NetFlow/sFlow or `tcpdump` capture) showing outbound connections originating from the Langflow process after any file write event. Run `find` output listing all files in the Langflow directory tree with creation/modification timestamps within the exploitation window.

Step 3: Eradication — Upgrade all Langflow deployments to v1.10.0, which contains the patch for the full vulnerability cluster (CVE-2026-5027, CVE-2026-0770, CVE-2026-21445, CVE-2026-33017, CVE-2025-3248). After upgrade, audit all files written to the application directory during the exposure window; remove any unrecognized files, particularly scripts or executables. Disable auto-login in configuration if not already done.

Reference CIS 7.3 (Perform Automated Operating System Patch Management) and CIS 7.4 (Perform Automated Application Patch Management). Apply D3-FMBV (File Magic Byte Verification) to scan uploaded or written files for unexpected file types.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 2.2 (Ensure Authorized Software is Currently Supported)

Compensating: Before upgrading, use `pip show langflow langflow-base` (or `pip3`) to confirm installed versions. Upgrade via `pip install --upgrade langflow==1.10.0`. For Docker deployments, pull `langflowai/langflow:1.10.0` and redeploy. For file magic byte verification without enterprise DLP: use `file ` on each suspicious file or run ClamAV (`clamscan -r /path/to/langflow`) against the full application directory. Write a YARA rule targeting PHP/JSP/Python web shell signatures and scan with `yara -r webshell_rule.yar /path/to/langflow/`. Check file extensions against actual magic bytes using: `find . -type f | xargs file | grep -v 'ASCII\|JSON\|Python\|empty'`.

Evidence: Before removing any files, forensically image or hash the full Langflow application directory: `find /path/to/langflow -type f -exec md5sum {} \; > pre_eradication_hashes.txt`. Preserve any suspicious files (web shells, scripts, unexpected executables) in an isolated evidence directory before deletion — these are primary indicators of T1505.003 persistence. Capture the output of `pip list` showing the pre-patch Langflow and langflow-base versions as evidence of the vulnerable configuration. If a web shell was found, preserve the full HTTP access log entries that show how it was accessed post-deployment.

Step 4: Recovery — After patching, verify the running Langflow version via the application's version endpoint or package manifest. Confirm auto-login is disabled and that all session token issuance requires valid credentials. Rotate all API keys, secrets, and credentials stored in or accessible by Langflow pipelines, as attacker access to the pipeline environment may have exposed them. Apply D3-CRO (Credential Rotation). Enable enhanced logging on the Langflow host per NIST AU-2 (Event Logging) and AU-12 (Audit Record Generation) and monitor for anomalous process execution or outbound network activity for at least 30 days post-remediation.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AU-2 (Event Logging), NIST AU-12 (Audit Record Generation), NIST AC-2 (Account Management), NIST AC-3 (Access Enforcement), CIS 5.1 (Establish and Maintain an Inventory of Accounts), CIS 6.2 (Establish an Access Revoking Process)

Compensating: Verify the patched version by querying Langflow's built-in version endpoint: `curl -s http://localhost:7860/api/v1/version` and confirm the response shows `1.10.0`. Confirm auto-login is disabled by attempting an unauthenticated request to `/api/v1/auto_login` — a 401/403 response confirms remediation. For credential rotation, enumerate all environment variables and `.env` file entries in the Langflow config directory for API keys (OpenAI, Anthropic, HuggingFace, etc.) and rotate each at the respective provider's console. Enable Sysmon EventIDs 1 (Process Create), 3 (Network Connect), and 11 (File Create) on the Langflow host and forward logs to a central syslog server using `syslog-ng` or `rsyslog` for the 30-day monitoring window.

Evidence: Before bringing the patched system online, verify the integrity of the restored Langflow configuration files by comparing hashes against known-good baseline from version control or a fresh install of v1.10.0. Document the auto-login configuration state (`AUTO_LOGIN=false`) in the `.env` file as a recovery verification artifact. Preserve a list of all API keys and secrets that were accessible within Langflow pipeline configurations during the exposure window — these represent the full credential exposure scope and must be included in any breach notification assessment.

Step 5: Post-Incident — Conduct a review of AI development platform deployment standards. Assess whether auto-login defaults on any other internally deployed developer tools create similar unauthenticated access exposure. Implement a documented process to identify and remediate internet-exposed development tools using CIS 7.1 (Establish and Maintain a Vulnerability Management Process) and CIS 7.2 (Establish and

Maintain a Remediation Process). Review access control policies for AI pipeline infrastructure against NIST AC-6 (Least Privilege) and AC-3 (Access Enforcement) to ensure pipeline execution environments cannot reach sensitive internal systems. Apply D3-UAP (User Account Permissions) to restrict pipeline service account scope.

NIST Phase: Post Incident

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

Controls: NIST AC-6 (Least Privilege), NIST AC-3 (Access Enforcement), NIST AU-6 (Audit Record Review, Analysis, And Reporting), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 4.6 (Securely Manage Enterprise Assets and Software), CIS 4.7 (Manage Default Accounts on Enterprise Assets and Software)

Compensating: Use Censys Search or Shodan (free tier) to scan your ASN and IP ranges for other internet-exposed developer tools with default authentication configurations (Jupyter Notebooks, Streamlit apps, Gradio interfaces, n8n, Flowise — all common AI/ML tooling with similar auto-login anti-patterns). Create a recurring monthly cron job that uses `nmap -p 7860,8501,8888,3000 -oG -` to detect newly exposed AI platform ports. For Langflow pipeline service accounts, audit the OS-level user running the Langflow process — it should not be root or have write access outside the application directory: `ps aux | grep langflow` to identify the service account, then `id ` to verify group memberships.

Evidence: Compile a lessons-learned document that includes: the full timeline from first scan/exploitation attempt (extracted from web logs) to detection and containment, confirming whether the 20-hour AI pipeline compromise window applies to your environment. Document which Langflow pipeline configurations were accessible during the exposure window, including any LLM provider API keys, internal endpoint URLs, or data source credentials embedded in pipeline definitions — this inventory drives both the credential rotation scope and any regulatory breach notification assessment.

Detection Guidance

Primary detection focus: unauthenticated file write activity and web shell placement on Langflow hosts. Query application and web server access logs for POST requests to Langflow file-handling or flow-execution API endpoints that were not preceded by a conventional authenticated login. Because the auto-login default (CWE-306) means a valid token may appear in logs despite no user credential being submitted, prioritize detection on file write events rather than authentication events alone. Look for file creation events outside expected application data directories, particularly in web-accessible paths or system directories, correlating with T1505.003 (Web Shell) and T1083 (File and Directory Discovery). Monitor process execution logs on Langflow host systems for unexpected shell invocations (bash, sh, cmd, powershell) or scripting interpreter activity spawned from the Langflow application process, consistent with T1059 and T1106. Review outbound network connections from Langflow hosts for C2 beacon patterns or data exfiltration to unfamiliar external IPs following any file write event. Use Censys or runZero to identify externally exposed Langflow instances in your network footprint. Cross-reference new or modified files in the application directory against known-good file hashes using file integrity monitoring. Alert on any Langflow process writing files to paths not consistent with normal pipeline execution. NIST AU-6 (Audit Record Review, Analysis, and Reporting) and CIS Controls v8 8.2 (Collect Audit Logs) provide the logging baseline required to execute these queries.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`https://www.bleepingcomputer.com/news/security/path-traversal-flaw-in-ai-dev-platform-langflow-exploited-in-attacks/`	BleepingComputer reporting on active exploitation of CVE-2026-5027 in Langflow	MEDIUM
URL	`https://www.sysdig.com/blog/cve-2026-33017-how-attackers-compromised-langflow-ai-pipelines-in-20-hours`	Sysdig research documenting 20-hour compromise timeline for Langflow AI pipelines	MEDIUM
URL	`https://nvd.nist.gov/vuln/detail/cve-2026-33017`	NVD detail record for CVE-2026-33017, part of the Langflow vulnerability cluster	HIGH
URL	`https://www.runzero.com/blog/langflow/`	runZero guidance on finding Langflow-impacted assets in your environment	MEDIUM

Framework Mappings

MITRE-ATTACK

- **T1036** — Masquerading
- **T1106** — Native API
- **T1505.003** — Web Shell
- **T1059** — Command and Scripting Interpreter
- **T1083** — File and Directory Discovery
- **T1190** — Exploit Public-Facing Application

NIST-800-53R5

- **CM-2** — Baseline Configuration
- **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
- **IA-2** — Identification and Authentication (Organizational Users)
- **AC-3** — Access Enforcement
- **SI-10** — Information Input Validation

OWASP-TOP10-2021

- **A07:2021** — Identification and Authentication Failures

- **A01:2021** — Broken Access Control
- **A04:2021** — Insecure Design

CIS-V8

- **6.3** — Require MFA for Externally-Exposed Applications
- **16.10** — Apply Secure Design Principles in Application Architectures
- **16.12** — Implement Code-Level Security Checks
- **7.3** — Perform Automated Operating System Patch Management
- **7.4** — Perform Automated Application Patch Management

HIPAA-SECURITY

- **164.312(d)** — Person or Entity Authentication

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures

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
T1036	Masquerading	Defense-Evasion
T1106	Native API	Execution
T1505.003	Web Shell	Persistence
T1059	Command and Scripting Interpreter	Execution
T1083	File and Directory Discovery	Discovery
T1190	Exploit Public-Facing Application	Initial-Access

Sources

Source	URL	Tier
Security News	https://www.bleepingcomputer.com/news/security/path-traversal-flaw-...	T3
CVE-2026-33017: How attackers compromised Langflow AI ... - Sysdig	https://www.sysdig.com/blog/cve-2026-33017-how-attackers-compromise...	T3

Source	URL	Tier
CVE-2026-33017 - CVE Record	https://www.cve.org/CVERecord?id=CVE-2026-33017	T3
CVE-2026-33017 Detail - NVD	https://nvd.nist.gov/vuln/detail/cve-2026-33017	T1
Langflow Flodrix vulnerability CVE-2026–33017: Find impacted assets	https://www.runzero.com/blog/langflow/	T3
NVD	https://nvd.nist.gov/vuln/detail/CVE-2026-5027, CVE-2026-0770, CVE-...	T1