CSAI Foundation

Cloud Security Alliance AI Safety Initiative

Critical RCE in Flowise AI Agent Builder: Active Exploitation

CVE-2025-59528 Analysis, Impact Assessment, and Remediation Guidance

Unofficial AI-assisted Research

2026-04-07

Key Takeaways

- CVE-2025-59528 is a CVSS 10.0 unauthenticated remote code execution vulnerability in Flowise — with more than 42,000 GitHub stars and over 12,000 internet-facing instances detectable via network scanning, among the most visible open-source AI agent builder platforms — currently under active exploitation [1][2].
- The root cause is the use of JavaScript's `Function()` constructor — functionally equivalent to `eval()` — to process attacker-controlled input in Flowise's CustomMCP node, granting full Node.js runtime privileges without authentication [3][4].
- Successful exploitation enables credential theft of LLM provider API keys, database connection strings, and cloud service secrets stored in the Flowise environment, followed by lateral movement into the broader AI and enterprise infrastructure the instance integrates with [2][5].
- Flowise was acquired by Workday in August 2025, making this vulnerability a supply chain concern extending into enterprise HR and finance platforms that have begun embedding Flowise capabilities [6].
- Organizations must upgrade to Flowise 3.0.6 immediately and treat any previously exposed instance as fully compromised, rotating all associated credentials [3][4].

Background

Flowise and Its Role in Enterprise AI Infrastructure

Flowise is an open-source, low-code visual builder for constructing AI agent workflows and large language model (LLM) applications. Built on LangChain, it provides a drag-and-drop interface through which developers and operations teams design, prototype, and deploy AI applications — including retrieval-augmented generation pipelines, multi-agent orchestration systems, chatbots, and tool-calling workflows — without requiring deep programming expertise. The platform supports integrations with more than 100 external tools, vector databases, APIs, and memory modules, making it a popular entry point for enterprise AI deployment [1][7].

The platform's adoption trajectory has been steep. With more than 42,000 GitHub stars and a substantial open-source contributor community, Flowise became one of the most frequently deployed components in production AI infrastructure. Fortune 500 companies including Thermo Fisher, Deloitte, and Accenture are among its reported enterprise users [1]. Workday's acquisition of Flowise in August 2025 extended this footprint further, embedding Flowise's agent-building capabilities directly into the Workday platform and connecting it to the HR, financial, and workforce management systems of thousands of enterprise customers [6].

This adoption profile is precisely what makes CVE-2025-59528 consequential. A critical vulnerability in a widely deployed AI infrastructure component is not merely a risk to the Flowise instance itself — it is a risk to every system, credential, and data source that instance touches. In architectures where Flowise orchestrates agent workflows and holds credentials for all integrated systems, compromising the Flowise instance grants equivalent access to the agent's full operational scope. For many organizations, Flowise sits at the center of their AI workflows, with access to production databases, cloud services, LLM provider accounts, and internal APIs.

The Model Context Protocol Integration

The vulnerability resides specifically in Flowise's implementation of the Model Context Protocol (MCP), an open standard introduced by Anthropic in late 2024 that enables AI agents to interact with external tools, data sources, and services through a standardized interface [8]. MCP has seen rapid adoption as a standardized mechanism for extending agent capabilities, with support integrated across major AI agent platforms since its introduction in late 2024 [8]. However, its security characteristics vary significantly across implementations. Security research has identified serious vulnerabilities across MCP implementations, including path traversal flaws, OS command injection via shell invocations in tool-calling integrations, and server-side request forgery — vulnerability classes observed across multiple independently developed MCP server implementations [9].

Flowise's `CustomMCP` node is the platform's interface for connecting to custom or user-defined MCP servers. It is intended to allow users to specify server connection parameters in JSON format, which the application then uses to establish the MCP connection. The vulnerability arises in how Flowise processes that configuration input before attempting to parse it as JSON.

Security Analysis

Vulnerability Mechanics

CVE-2025-59528 was assigned a CVSS v3.1 score of 10.0 — the maximum — reflecting its combination of unauthenticated network accessibility, zero required interaction, and complete compromise of confidentiality, integrity, and availability [1][3]. The full vector (`CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H`) indicates that no authentication is required, no special conditions or configuration must be in place, and the impact extends beyond the vulnerable component itself to affect dependent systems. The Exploit Prediction Scoring System (EPSS) rated the vulnerability at 84.07% probability of exploitation within 30 days, placing it at the 99.28th percentile of all scored CVEs, as of April 7, 2026 — EPSS scores are recalculated daily and will differ if checked at a later date [1].

The technical root cause is a design-level error in the `convertToValidJSONString` function within `CustomMCP.ts`, lines 262–270. Rather than using a strict JSON parser to process the configuration string supplied by the user, the function passes that string directly to JavaScript's `Function()` constructor:

```
Function('return ' + inputString)()
```

This construct compiles and immediately invokes the supplied string as JavaScript code with full Node.js runtime privileges. It is functionally equivalent to `eval()`, and the security implications are identical: any attacker who can supply the `inputString` parameter can execute arbitrary code on the server. The endpoint that exposes this path — `POST /api/v1/node-load-method/customMCP` — requires no authentication by default, meaning exploitation requires nothing more than HTTP access to the Flowise port [3][4].

The vulnerability was discovered by security researcher Kim SooHyun and reported to FlowiseAI in September 2025, resulting in a GitHub Security Advisory (GHSA-3gcm-f6qx-ff7p) published September 13, 2025 and NVD publication on September 22, 2025 [3]. The fix in Flowise 3.0.6 replaces the `Function()` constructor with `JSON5.parse()`, a strict data-only parser that does not evaluate code. The vulnerable surface spans four files in the taint chain: route registration in `node-load-methods/index.ts`, the controller in the `getSingleNodeAsyncOptions` handler, a forwarding layer in the service, and the vulnerable method in `CustomMCP.ts` [4].

Exploitation and Observed Threat Activity

Public proof-of-concept exploit code was submitted to Exploit-DB (entry #52440) on October 31, 2025, approximately six weeks after the advisory was published, and has been publicly available for more than five months [5]. A typical exploit payload takes the form of an Immediately Invoked Function Expression that wraps `child_process.execSync()` calls to execute arbitrary OS commands, with return type characteristics crafted to pass minimal validation checks [5].

Active exploitation was confirmed in April 2026, with attack traffic traced to a single Starlink IP address — a pattern researchers characterize as indicative of coordinated, potentially automated scanning and exploitation consistent with opportunistic mass-exploitation campaigns [2]. No specific named threat actor group has been publicly attributed as of this writing. The attack surface is substantial: more than 12,000 Flowise instances are currently detectable as internet-facing via network scanning, and a meaningful fraction of these are likely running vulnerable versions given that the affected version range spans `>= 2.2.7-patch.1` through `< 3.0.6` [2].

Post-exploitation access provides an attacker with the full capabilities of the Node.js process, including arbitrary OS command execution via `child_process`, unrestricted filesystem access via `fs`, and the ability to read all environment variables — which in production Flowise deployments commonly include LLM provider API keys (OpenAI, Anthropic, Google, and others), database connection strings, vector database credentials, and cloud service secrets, given Flowise's integration model for connecting to external services. These credentials provide the immediate basis for financial impact through API abuse, and for lateral movement into connected systems.

A Pattern of Systemic Vulnerability

CVE-2025-59528 is not an isolated incident in the Flowise security record. It is the third Flowise vulnerability confirmed exploited in the wild in 2025, following CVE-2025-26319 (CVSS 9.8, unauthenticated arbitrary file upload via path traversal in the `/api/v1/attachments` endpoint, reported February 2025) and CVE-2025-8943 (CVSS 9.8, unauthenticated OS command execution via shell invocation of `npx` in the Custom MCPs feature prior to version 3.0.1) [2][11][12]. Each of these vulnerabilities affects a different feature, but all three share a common failure pattern: user-controlled input reaches a high-privilege execution path without adequate validation or authentication controls.

The three exploited vulnerabilities in a single year, each affecting different features, suggest that security review may not have kept pace with feature development — though FlowiseAI's internal development processes are not publicly documented. The rapid expansion of Flowise's feature set, particularly its integration of MCP and other emerging AI protocols, is consistent with a development

environment where new capabilities are introduced at a pace that creates opportunities for security gaps to go undetected until they are found by external researchers or actively exploited. Organizations operating Flowise in production should treat this history as a signal that additional vulnerabilities may exist in less-examined parts of the codebase, and should plan for ongoing vigilance rather than a single-patch resolution.

The Workday acquisition introduces additional supply chain considerations. As Flowise components are integrated into the Workday platform, vulnerabilities in the Flowise codebase potentially become vulnerabilities in enterprise HR and finance systems with dramatically larger blast radii. The timeline of this integration relative to the patch cycle for CVE-2025-59528 is not publicly known, but enterprise customers using Workday AI features built on Flowise should verify with Workday the version of Flowise components in their environment [6].

Recommendations

Immediate Actions

The highest priority action for any organization operating Flowise is to upgrade to version 3.0.6 or later without delay. The upgrade path follows standard npm procedures; however, organizations upgrading from 2.x versions should review the Flowise 3.0.x release notes for potential breaking changes before upgrading, as this represents a major version increment. Organizations that cannot upgrade immediately should implement network-layer controls that block unauthenticated access to the Flowise API port (default: 3000) from any untrusted network segment.

Any Flowise instance that was internet-accessible and running a vulnerable version should be treated as potentially compromised, regardless of whether exploitation evidence is present in logs. Absence of log evidence is not a reliable indicator of absence of compromise: attackers with code execution can modify or suppress logging, and exploitation attempts may have occurred before logging was enabled or before log retention windows captured them. The appropriate response is to rotate all credentials that the Flowise instance had access to: LLM provider API keys, database passwords, vector database credentials, cloud service access keys, and any secrets stored in environment variables or Flowise's configuration files.

Detection of active exploitation attempts in logs should focus on POST requests to `/api/v1/node-load-method/customMCP` containing JavaScript keywords such as `process.mainModule`, `child_process`, `require`, `execSync`, or `exec`. SonicWall has published IPS signatures

21519 and 21918 for automated detection in supported environments [4].

Short-Term Mitigations

Organizations should audit whether Flowise's built-in API authentication is enabled. In older Flowise versions, authentication is disabled by default [3][4] — a configuration that is inappropriate for any deployment accessible beyond a single developer's local environment, absent compensating network controls. If authentication is not currently enforced, it should be enabled immediately and access tokens should be treated as production secrets.

Flowise should be deployed behind a reverse proxy or WAF configured to terminate TLS, enforce authentication, and log all API requests. Direct exposure of the Flowise application port to untrusted networks eliminates a critical layer of defense in depth. Network segmentation should restrict Flowise's ability to reach internal systems beyond what it genuinely requires — this limits lateral movement potential in the event of future compromise.

Organizations running Flowise in container environments should review whether the container runs with more privileges than necessary. A process running as root in a container with `child_process` access through a code injection vulnerability presents a substantially expanded blast radius compared to a properly constrained deployment.

Strategic Considerations

CVE-2025-59528 illustrates a broader challenge confronting AI security practitioners: the AI infrastructure tooling layer — agent builders, workflow platforms, MCP servers, and LLM proxies — is being deployed at enterprise scale faster than security review and patching disciplines have matured for these categories of software. Flowise's record of three exploited CVEs in a single year may reflect a broader pattern in the AI infrastructure tooling category, where rapid feature development has in multiple cases outpaced security validation — though no comprehensive cross-platform comparison data is currently available to quantify this trend with precision.

Organizations should apply to AI infrastructure the same scrutiny they apply to any internet-facing application server. This means mandatory authentication on all API endpoints, network segmentation, privilege minimization for the runtime process, credential rotation policies that account for the possibility of silent compromise, and explicit inclusion of AI tooling in vulnerability management programs. The "it's just an internal tool" framing that sometimes applies to developer platforms is not appropriate for Flowise deployments: these platforms hold production credentials, connect to production data sources, and execute code with production-level privileges.

For enterprise AI governance programs, this class of vulnerability reinforces the case for inventory completeness as a foundational control. Organizations cannot protect what they have not catalogued. Discovering that a Flowise instance has been running on an internal network since 2024, accumulating integrations and credentials, is a scenario that becomes increasingly likely as AI tooling proliferates through engineering and operations teams.

CSA Resource Alignment

CVE-2025-59528 intersects directly with several of CSA's AI security frameworks and guidance materials. MAESTRO — CSA's threat modeling framework for agentic AI systems [13] — identifies MCP server compromise as a distinct attack category representing a single point of failure in agent tool-calling architectures. The vulnerability in Flowise's CustomMCP node is a concrete instantiation of this threat: an attacker who exploits the RCE gains control over the interface through which the agent accesses its tools and external data sources, enabling both immediate credential theft and the insertion of malicious behavior into the agent's tool-calling pipeline.

The AI Controls Matrix (AICM) [14] addresses the relevant control objectives through its API security, access management, and secure software development domains. AICM control objectives around input validation, authenticated API access, and privilege minimization for AI service components apply directly to Flowise deployments. Organizations using the AICM to assess their AI security posture should explicitly include AI builder platforms and workflow tools in their control scope, not just model inference endpoints.

CSA's prior research on MCP security has documented vulnerability classes — including path traversal, OS command injection, and server-side request forgery — across commercial and open-source MCP server implementations, establishing the pattern that CVE-2025-59528 continues. The Flowise CustomMCP node failure is a code-level manifestation of the broader architectural risk that CSA has characterized as insufficient trust boundary enforcement between AI agents and the tool-calling layer.

CSA's STAR for AI certification program provides a structured framework for evaluating the security posture of AI service providers, including AI infrastructure vendors. Vendors seeking STAR for AI certification would be expected to demonstrate mature vulnerability management practices, including rapid patching, proactive security testing of new feature code, and disclosure practices that give customers adequate time to remediate before exploitation becomes widespread. The six-month window between CVE-2025-59528's publication and widespread active exploitation reporting, combined with

the availability of public exploit code for five of those months, represents a remediation window during which enterprise customers with internet-exposed instances were at material risk, and some may have been compromised.

References

[1] Wiz. "CVE-2025-59528." Wiz Vulnerability Database, 2025.

[2] Vulert. "Flowise AI CVE-2025-59528: RCE Exploitation and 12,000+ Exposed Instances." Vulert Blog, 2026.

[3] FlowiseAI. "GHSA-3gcm-f6qx-ff7p: Remote Code Execution in CustomMCP Node." GitHub Security Advisory, September 2025.

[4] SonicWall. "FlowiseAI Custom MCP Node Remote Code Execution." SonicWall Blog, 2025.

[5] Exploit-DB. "Flowise AI 2.2.7 - Remote Code Execution (CVE-2025-59528)." Exploit-DB, October 31, 2025.

[6] Workday. "Workday Acquires Flowise, Bringing Powerful AI Agent Builder Capabilities to the Workday Platform." Workday Newsroom, August 14, 2025.

[7] The Hacker News. "Flowise AI Agent Builder Under Active CVSS 10.0 RCE Exploitation; 12,000+ Instances Exposed." The Hacker News, April 7, 2026.

[8] Anthropic. "Model Context Protocol." Anthropic, November 2024.

[9] GitLab. "CVE-2025-59528 Advisory: flowise npm package." GitLab Advisory Database, 2025.

[10] NIST NVD. "CVE-2025-59528 Detail." National Vulnerability Database, September 22, 2025.

[11] NIST NVD. "CVE-2025-26319 Detail." National Vulnerability Database, 2025.

[12] NIST NVD. "CVE-2025-8943 Detail." National Vulnerability Database, 2025.

[13] Cloud Security Alliance. "MAESTRO: Agentic AI Threat Modeling Framework." CSA AI Safety Initiative, 2024.

[14] Cloud Security Alliance. "AI Controls Matrix (AICM)." CSA, 2024.

