

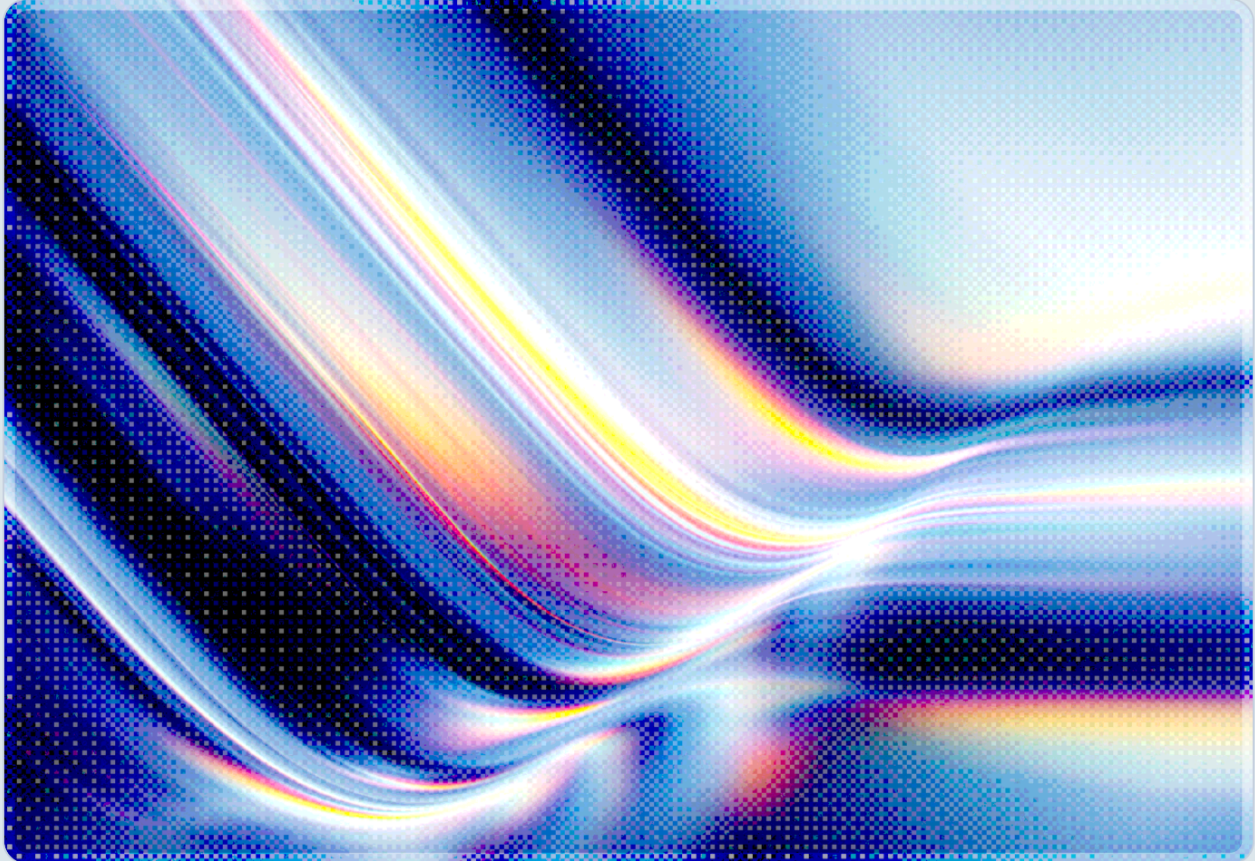
GitLab AI Gateway RCE: Securing the Gateway Tier

CVE-2026-90970 and the Recurring Template-Escape Pattern in Self-Hosted AI Infrastructure

2026-10-03



AI-assisted Rapid Research



CSAI

cloud
CSA security
alliance®

© 2026 Cloud Security Alliance. Some rights reserved.

You may download, store, display, view, print, redistribute, and link to this document in its original, unmodified form, provided that attribution to the Cloud Security Alliance is maintained and all trademark and copyright notices remain intact.

This document may not be modified or altered. You may quote portions of the document as permitted by the Fair Use provisions of the United States Copyright Act, provided that attribution is given to the Cloud Security Alliance.

This document may be shared on professional and social media platforms in its original form with attribution.

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

Key Takeaways

GitLab disclosed CVE-2026-90970 on October 2, 2026, a critical flaw (CVSS 9.9) in the self-hosted GitLab AI Gateway [1][2]. An authenticated user with Duo Agent Platform access can escape the prompt template sandbox through a crafted custom flow configuration and execute arbitrary commands on the gateway [1]. The exposure falls on organizations running their own gateway; fixes are available in versions 19.2.4, 19.3.2, and 19.4.1 [1][2]. The Hacker News reports that GitLab offers no fixes for versions 18.1.6 through 19.1.x [2]. GitLab's advisory for the earlier CVE-2026-1868 stated that GitLab.com, GitLab Dedicated, and the GitLab-hosted gateway were protected [3], but we did not confirm that the October advisory makes the same statement.

The flaw closely resembles CVE-2026-1868, which GitLab patched in February 2026 and which also scored 9.9 and involved template expansion of user-supplied flow definitions [2][3]. A second critical flaw of the same class within about eight months suggests that the template-handling design of the flow engine, rather than a single coding error, is a durable attack surface. That is an inference from two disclosures, not a statement from GitLab.

In our assessment, the wider lesson is that the AI gateway tier concentrates credentials and trust. According to reporting on the advisory, the gateway holds JWT signing keys and connects to both the GitLab instance and AI model providers [2]. Compromise of that host therefore has consequences beyond the gateway itself. Organizations should patch immediately, restrict who can author custom flows, and treat the gateway with controls comparable to those applied to identity and CI/CD infrastructure, as detailed below.

Background

GitLab Duo Self-Hosted lets organizations run AI-assisted development features against models they control. The AI Gateway is the component that mediates between the GitLab instance and the model backends, and it also hosts the Duo Agent Platform's workflow service, which executes "flows," configurable multi-step agent workflows [1][3]. Custom flows are defined by users and rendered through prompt templates, which is the part of the system implicated in this vulnerability.

GitLab's advisory describes CVE-2026-90970 as an improper neutralization issue in a custom flow prompt template. The CVSS 3.1 vector is AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:H, which reflects network reachability, low complexity, a low-privilege requirement, and a changed scope with high impact on confidentiality, integrity, and availability [1]. The scope change matters: the vulnerable component is the gateway, but the consequences extend to systems that trust it. The Hacker News reports the weakness classification as CWE-1336, a template engine weakness [2].

The affected ranges cover AI Gateway versions 18.1.6 through 19.2.3, 19.3.0 through 19.3.1, and 19.4.0. GitLab credits the report to a researcher using the handle invisiblemeerkat through its HackerOne program [1]. As of the reporting reviewed, CISA's assessment lists exploitation as "none," and we found no public evidence of active exploitation [2]. This note relies on GitLab's advisory and The Hacker News reporting; no additional independent analysis of the flaw was available at the time of writing.

Item	CVE-2026-1868 (February 2026)	CVE-2026-90970 (October 2026)
CVSS 3.1 score	9.9	9.9
Described cause	Insecure template expansion of user-supplied data in Duo Workflow Service via crafted flow definitions	Prompt template sandbox escape via crafted custom flow configuration
Access required	Authenticated GitLab user	Authenticated user with Duo Agent Platform access
Stated impact	Denial of service or code execution on the gateway	Arbitrary command execution on the gateway
Fixed versions	18.6.2, 18.7.1, 18.8.1	19.2.4, 19.3.2, 19.4.1

Sources: [1][3].

Security Analysis

The exploit path begins with a legitimate, low-privilege capability. Any account that can create or modify custom flows becomes a potential entry point, so the effective attack surface is the population of users granted Duo Agent Platform access, not the set of administrators. The size of this population depends on how widely an organization has enabled Duo Agent Platform; organizations that have enabled it broadly should assume that many accounts can reach the vulnerable code path, and that credential theft of any one of them is sufficient.

The consequences of code execution on the gateway depend on what the host holds and can reach. Reporting indicates the gateway stores JWT signing keys and has connectivity to the GitLab instance and to model providers [2]. We infer that an attacker with command execution could attempt to mint or forge tokens, read provider API credentials and prompts or code context passing through the service, and use the gateway's network position to pivot toward the GitLab instance or internal model endpoints. The exact impact will vary with deployment, and GitLab has not published exploitation details. Container isolation, egress rules, and secret storage choices will determine how far an attacker can go.

In our assessment, the recurrence matters more than the single flaw. Both CVE-2026-1868 and CVE-2026-90970 involve user-controlled flow definitions that reach a template engine in a context that permits command execution [1][3]. Template sandboxes have typically proved difficult to make airtight in general-purpose software, and prompt templating adds a further complication: flow templates are meant to carry dynamic content, tool calls, and variables. The second disclosure indicates that the February fix did not close every path to template-driven code execution, though GitLab has not said whether the two flaws share a root cause. Defenders should therefore plan for further variants.

This pattern is not unique to GitLab. CSA has documented a series of critical flaws in the LiteLLM AI gateway, including an unauthenticated RCE chain that combined an authenticated command injection with a framework-level authentication bypass, and a cascade of issues that exposed the API keys the gateway manages [4][5]. The common thread is architectural: gateways aggregate credentials and sit between users, code, and models, yet in our assessment they may receive less hardening and monitoring than the systems they serve, because they are frequently treated as internal infrastructure. We have not surveyed deployments to confirm this. Self-hosted gateways also place the patching burden on the customer, and The Hacker News reports that GitLab offers no fixes for versions 18.1.6 through 19.1.x [2], so lagging deployments must upgrade across release lines.

Recommendations

Immediate Actions

Inventory every self-hosted AI Gateway, including those stood up by individual teams for evaluation, and compare versions against the affected ranges. Upgrade to 19.2.4, 19.3.2, or 19.4.1 as appropriate, following GitLab's Self-Hosted AI Gateway installation documentation [1]. Organizations on versions older than 19.2.4 should plan a full upgrade rather than look for a backport.

Where immediate upgrade is not possible, reduce exposure by limiting Duo Agent Platform access to the users who need it and by restricting who can create or edit custom flows. Review the gateway's flow definitions and recent workflow activity for unexpected template constructs or commands. After upgrading, consider rotating the JWT signing keys and any model provider credentials stored on or reachable from the gateway if there is any indication of suspicious flow activity, since the advisory does not state whether past exploitation would leave traces.

Short-Term Mitigations

Restrict the gateway's network position. Egress should be limited to the GitLab instance and the specific model endpoints required, and inbound access should be limited to the GitLab instance. Run the gateway as an unprivileged container with a read-only filesystem where practical, and keep secrets in a managed secret store rather than in environment variables on the host. These controls do not prevent a sandbox escape, but they limit what the resulting command execution can reach.

Add detection for gateway hosts. Useful signals include child processes spawned by the gateway service, outbound connections to destinations outside the allowlist, and changes to flow configurations by accounts that do not normally author them. Subscribe the platform team to GitLab's security release notices so that gateway patches, which are published separately from the main GitLab patch releases, are not missed [1][3].

Strategic Considerations

Treat the AI gateway tier as a distinct asset class with its own owner, patch service-level agreement, and threat model. It aggregates credentials, mediates sensitive source code and prompts, and executes user-defined workflow logic, which together justify controls closer to those applied to identity and CI/CD

infrastructure than to ordinary internal applications. Procurement and architecture reviews for AI development tooling should ask how user-authored templates and workflows are isolated from the host, and whether the vendor's fix addressed a vulnerability class or a single instance.

Where possible, design for blast-radius containment: separate gateways by sensitivity, scope provider credentials per gateway, and avoid sharing signing keys across environments. Given that two critical flaws of the same class have been disclosed within eight months, organizations should also consider whether custom flow authoring needs to be enabled at all in production instances, or whether a reviewed set of flows is sufficient.

CSA Resource Alignment

CSA's rapid research on the LiteLLM AI gateway is the most directly relevant prior work. *LiteLLM AI Gateway: Active Exploitation via MCP Injection* [4] describes an actively exploited unauthenticated RCE chain scored CVSS 10.0 when its components are combined in an AI gateway, and its remediation and detection guidance for the gateway tier applies here, although the GitLab flaw requires authentication and has no known exploitation. *LiteLLM AI Gateway: Critical Vulnerability Chain Exposes API Keys* [5] addresses the credential-concentration risk that makes gateway compromise costly, and supports the recommendation to scope and rotate provider credentials.

For control mapping, the AI Controls Matrix v1.1 [6] provides the applicable control domains. Threat and vulnerability management (TVM) covers patch timelines and inventory for gateway components, and application and interface security (AIS) covers input handling, including the template-neutralization failure at issue. We also point to the identity and access management domain for restricting who can author flows and for managing signing keys and provider secrets.

References

- [1] GitLab. "[GitLab AI Gateway Patch Release 19.4.1, 19.3.2, 19.2.4 \(CVE-2026-90970\)](#)." GitLab Docs, October 2026.
- [2] The Hacker News. "[GitLab Patches Critical Self-Hosted AI Gateway Flaw](#)." The Hacker News, October 2, 2026.
- [3] GitLab. "[GitLab AI Gateway Critical Patch Release: 18.6.2, 18.7.1, and 18.8.1 \(CVE-2026-1868\)](#)." GitLab Docs, February 6, 2026.
- [4] Cloud Security Alliance. "[LiteLLM AI Gateway: Active Exploitation via MCP Injection](#)." CSA Labs, June 13, 2026.
- [5] Cloud Security Alliance. "[LiteLLM AI Gateway: Critical Vulnerability Chain Exposes API Keys](#)." CSA Labs, June 16, 2026.
- [6] Cloud Security Alliance. "[AI Controls Matrix v1.1](#)." Cloud Security Alliance, 2026.