Hive Pro

Threat Level

Red

Hiveforce Labs

# THREAT ADVISORY

## VULNERABILITY REPORT

# GitLab Path Traversal (CVE-2026-85706) Exploited to Steal Secrets


| Date of Publication | Admiralty Code | TA Number |
| --- | --- | --- |
| September 15, 2026 | A1 | TA2026268 |

# Summary

**First Seen:** September 10, 2026
**Affected Products:** GitLab Community Edition (CE) and Enterprise Edition (EE)
**Impact:** GitLab has patched CVE-2026-85706, a maximum-severity (CVSS 10.0) path traversal flaw in the repository commits API of its self-managed Community and Enterprise Edition products. The weakness stems from improper path confinement combined with missing authentication enforcement, allowing a remote attacker with no credentials to read arbitrary files from an affected GitLab server, provided the instance hosts at least one public project. Within hours of the September 10, 2026 patch release, researchers reproduced the flaw and observed opportunistic scanning against internet-exposed servers. Because GitLab instances routinely store source code, deploy keys, and CI/CD secrets, this file-read primitive translates directly into credential theft and downstream supply-chain compromise, making emergency patching essential for every self-managed deployment.

## CVE

| CVE | NAME | AFFECTED PRODUCT | ZERO-DAY | CISA KEV | PATCH |
|---|---|---|---|---|---|
| CVE-2026-85706 | GitLab Community Edition and Enterprise Edition Path Traversal Vulnerability | GitLab CE/EE | ✗ | ✓ | ✓ |

# Vulnerability Details

**#1**

GitLab has shipped emergency security releases to close a maximum-severity flaw in its self-managed Community Edition (CE) and Enterprise Edition (EE) products that lets unauthenticated attackers read arbitrary files from vulnerable servers. Tracked as CVE-2026-85706 and classified as a path traversal weakness (CWE-22), the issue sits in the repository commits API. It arises from two compounding defects: the endpoint fails to properly confine file paths to the intended directory, and it does not enforce authentication before servicing the request. Together, these gaps let a request escape the boundaries GitLab expects to impose and reach files elsewhere on the underlying server.

**#2**

Exploitation is strikingly simple. An attacker sends a crafted request to the commits API and manipulates the file-path parameter so that the application returns the contents of files it should never expose. Researchers who reproduced the issue confirmed that arbitrary files can be read in a single HTTP request, with the only practical prerequisite being that the target instance hosts at least one public project.

**#3**

The vulnerability affects all self-managed GitLab CE and EE builds from version 18.7 up to (but not including) 19.1.8, from 19.2 up to 19.2.6, and from 19.3 up to 19.3.2, across Omnibus, source, and Helm chart deployments. GitLab.com already runs a fixed build and GitLab Dedicated tenants require no action, so the exposure is concentrated on customer-operated instances, with internet-facing servers at greatest risk because no login is required to trigger the flaw.

**#4**

Active exploitation followed disclosure almost immediately. Security researchers reported behavioral probes against a honeypot network from around 06:00 UTC on September 11, 2026, and later observed the activity escalate from reconnaissance to successful file exfiltration over the weekend, including theft of configuration files and SSH settings used to harvest secrets. With the flaw confirmed under active exploitation and requiring no authentication to trigger, organizations running affected self-managed instances should treat this as an emergency and upgrade to a fixed release without delay.

# Vulnerability

| CVE ID | AFFECTED PRODUCTS | AFFECTED CPE | CWE ID |
|---|---|---|---|
| CVE-2026-85706 | GitLab CE/EE self-managed (18.7 before 19.1.8; 19.2 before 19.2.6; 19.3 before 19.3.2) | cpe:2.3:a:gitlab:gitlab:*:*:*:*:community:*:*:*, cpe:2.3:a:gitlab:gitlab:*:*:*:*:enterprise:*:*:* | CWE-22 |

# Recommendations

**Apply the GitLab Patch Immediately:** Upgrade every affected self-managed GitLab CE or EE instance to 19.1.8, 19.2.6, or 19.3.2 depending on the branch in use, treating this as an emergency change outside normal patch cycles. Verify the exact running version after the upgrade rather than trusting the maintenance record, and remember that single-node installations will incur brief downtime while database migrations run.

**Hunt for Signs of Exploitation:** Review GitLab, reverse-proxy, WAF, and network logs for HTTP POST requests to the "/api/v4/projects/{id}/repository/commits/" endpoint that contain file-path parameters, alongside traversal-like path values, repeated unauthenticated requests, unfamiliar source addresses, and unusually large responses. Because exploitation can succeed in one request and logs may be incomplete, a clean search does not prove the instance was untouched, so continue monitoring after patching.

**Rotate Potentially Exposed Secrets:** If suspicious file access is found or cannot be ruled out, rotate any credentials, deploy keys, access tokens, CI/CD variables, and database secrets that may have resided on the server. Review repository permissions, CI/CD runner activity, recent deployments, and package-publishing events for anomalies that could indicate the reuse of stolen secrets.

**Restrict Exposure While Remediating:** Where immediate patching is not possible, limit access to GitLab from trusted development, administrative, or VPN networks and remove unnecessary public exposure. This reduces risk during the remediation window but is a stopgap, not a substitute for upgrading to a fixed release.

**Vulnerability Management:** Maintain a current inventory of every GitLab instance, including secondary nodes, test systems, disaster-recovery environments, and containerized deployments behind proxies or load balancers. Track software versions and security patches, prioritize CISA KEV entries in the patch pipeline, and assess the security practices of third-party vendors for critical DevOps services.

**Attack Surface Monitoring:** Continuously discover and monitor internet-facing DevOps assets so that newly exposed or unpatched GitLab servers are identified before attackers reach them. Pair asset discovery with real-time exploit and KEV tracking to shorten the window between disclosure and remediation for high-severity, actively exploited flaws.

## Potential MITRE ATT&CK TTPs

| Tactic | Technique | Sub-technique |
|---|---|---|
| Initial Access | T1190: Exploit Public-Facing Application | |
| Collection | T1005: Data from Local System | |
| Credential Access | T1552: Unsecured Credentials | T1552.001: Credentials In Files |
| Resource Development | T1588: Obtain Capabilities | T1588.006: Vulnerabilities |

## Patch Link

https://docs.gitlab.com/releases/patches/patch-release-gitlab-19-3-2-released/

## References

https://docs.gitlab.com/releases/patches/patch-release-gitlab-19-3-2-released/

https://watchtowr.com/resources/rapid-reaction-gitlab-critical-path-traversal-vulnerability-cve-2026-85706/

# What Next?

At **Hive Pro**, it is our mission to detect the most likely threats to your organization and to help you prevent them from happening.

Book a Demo of **HivePro**.

REPORT GENERATED ON

**September 15, 2026 • 08:20 AM**



More at www.hivepro.com