contact@namiq.io · namiq.io

NamiQ

SECURITY ADVISORY

GitLab Community Edition and Enterprise Edition exploited

CVE-2026-97229 · on CISA's exploited list since 29 October 2026

Analysis Report

Any team running self-managed GitLab CE/EE on a version between 19.2 and 19.4.1 must upgrade now — the CISA deadline of 1 November 2026 has passed. Check your installed version first.

PREPARED BY	NamiQ	DATE	3 November 2026
FACTS AS OF	3 November 2026	VERSION	1.0
SEVERITY	**Critical: CVSS 3.1 10.0**	STATUS	On CISA's exploited list; due 1 November 2026

NamiQ © 2026 NamiQ. All rights reserved. This document is the property of NamiQ. It may be shared unaltered, with attribution.
Companion article: namiq.io/insights/gitlab-community-edition-and-enterprise-cve-2026-97229

The bottom line

> *Any team running self-managed GitLab CE/EE on a version between 19.2 and 19.4.1 must upgrade now — the CISA deadline of 1 November 2026 has passed. Check your installed version first.*

At a glance

Fact	Detail
Product	GitLab Community Edition and Enterprise Edition (CVE-2026-97229)
Severity	Critical: 10 (CVSS 3.1)
CISA	On its exploited list since 29 October 2026. Deadline for US federal agencies: 1 November 2026.
Chance of exploitation (EPSS)	0.28, the estimated chance in the next 30 days, as at 3 November 2026
Ransomware use	Not recorded by CISA. That is not the same as none.
Fixed in	GitLab CE/EE 19.2.8, 19.3.5, 19.4.2
Who looks first	Teams running self-managed GitLab CE/EE on versions 19.2 before 19.2.8, 19.3 before 19.3.5, or 19.4 before 19.4.2. GitLab.com and GitLab Dedicated users are not affected — the vendor's advisory confirms those are already patched.

Does this apply to you?

Answer in order. Stop at the first no.

Question	If no	If yes
Do you run a self-managed GitLab CE/EE instance (not GitLab.com or Dedicated)?	Nothing to do — GitLab-managed services are already patched.	Check which version you are running.
Is your GitLab server reachable from outside your organisation's network?	Still patch this week — no login is needed to exploit it even from inside.	Treat as the highest priority and patch before any other work today.

What it is

An unauthenticated attacker who can reach the GitLab server can submit a specially crafted regular expression via the CI/CD configuration interface and execute arbitrary code on the server, gaining

full control of it.

Why it matters

The CVSS score of 10.0 reflects that no authentication is required and a successful exploit gives an attacker full control of the GitLab server. The underlying flaw is an integer overflow in the Onigmo regular expression engine that GitLab uses to compile patterns submitted by users. The Ruby project's notice (posted 23 October 2026) explains that CVE-2026-97229 shares the same underlying Ruby bug as CVE-2026-93577 — an integer overflow in the size of a compiled pattern — fixed together in Ruby change #19020. Our reading is that the CI/CD lint path is likely the unauthenticated entry point, as GitLab's six threat detections name 'ci lint' worker crashes and client error bursts specifically.

What the vendor's advisory says

We read three pages on docs.gitlab.com and gitlab.com and www.ruby-lang.org. In short:

- The same release fixes CVE-2026-92883 (CVSS 8.7): XSS in syntax highlighting; an authenticated user can run JavaScript in other users' browser sessions. Affected: 19.1 before 19.2.8, 19.3 before 19.3.5, and 19.4 before 19.4.2.
- CVE-2026-16301 (CVSS 8.5) is also fixed: a Developer-role user could trigger CI/CD pipelines on protected branches without authorisation. Affected: 18.7 before 19.2.8, 19.3 before 19.3.5, and 19.4 before 19.4.2.
- GitLab has released six named threat detections for self-managed instances, including 'ci lint workhorse worker crash', 'ci regexp guard rejection', and 'ci lint scripted client error burst'.
- GitLab.com and GitLab Dedicated instances are confirmed patched — the vendor's advisory states this explicitly.

Things people will get wrong

- Our GitLab.com account means we are exposed. The vendor's advisory states GitLab-managed infrastructure, including GitLab.com and GitLab Dedicated, is already patched. Only self-managed installations are at risk.
- Restricting who can commit CI/CD configuration files is a sufficient workaround. The Ruby project's notice says limiting pattern length does not reliably protect against these flaws. The vendor has not published a supported workaround short of upgrading.
- A CVSS score of 10.0 must mean active mass exploitation is already happening. CISA records ransomware use as not recorded for this CVE. The score reflects the absence of any login requirement and the severity of the outcome, not confirmed mass exploitation.

How we would triage it

This is our judgement.

Step 1: Find the GitLab version you run

- On a self-managed Omnibus instance, the version is shown in the Admin Area under Components, or by running the command below as the system user.
- Confirm whether your version falls in the ranges 19.2 before 19.2.8, 19.3 before 19.3.5, or 19.4 before 19.4.2 — those are the ranges the record identifies as affected.

Step 2: Reduce exposure while you prepare the patch

- If the server is internet-facing and cannot be patched immediately, consider restricting access to the CI/CD lint endpoint at the network layer until the upgrade is applied.
- The vendor's advisory does not name a supported workaround, so network restriction is a temporary measure only and does not substitute for patching.

Step 3: Upgrade to the fixed release

- The vendor's advisory names 19.2.8, 19.3.5, and 19.4.2 as the first fixed versions in each series. Upgrade to the release that matches your current minor version.
- GitLab.com and GitLab Dedicated do not require any action — the advisory confirms those are already patched.

Step 4: Check for signs of earlier exposure

- GitLab has released six named threat detections. Look for 'ci lint workhorse worker crash', 'ci regexp guard rejection', 'ci regexp oversized non ascii path', and 'ci lint scripted client error burst' in GitLab Workhorse logs.
- Unexpected crashes in the ci-lint path before the patch was applied should be treated as possible exploitation attempts and investigated.

To close it, read the version from the product itself, not from an old inventory, and save the output with the date. If you cannot update today, write down who owns it, the new date, and what limits exposure until then. CISA's wording adds that you should stop using the product if no mitigation is available.

Send this to your IT provider

Copy it, change the names and dates, and send it. It is a draft for you to check, not a template tested with any provider.

> *We are writing about CVE-2026-97229, a CVSS 10.0 flaw in GitLab Community Edition and Enterprise Edition. It allows an unauthenticated attacker to execute code on a self-managed GitLab server by sending a crafted regular expression via the CI/CD configuration interface. The affected versions are 19.2 before 19.2.8, 19.3 before 19.3.5, and 19.4 before 19.4.2. The fixed versions are 19.2.8, 19.3.5, and 19.4.2. The CISA remediation deadline was 1 November 2026. Please confirm in writing: whether we run a self-managed GitLab CE/EE instance, which version it is on, and when the fixed version has been or will be applied. We would like written confirmation by 7 November 2026.*

Signs it was used

An update closes the hole. It does not tell you whether someone used it first. CISA's required action points to its forensic triage requirements (BOD 26-04). Look for:

- Look in GitLab Workhorse logs for crashes or errors in the ci-lint path — the vendor names 'ci lint workhorse worker crash' and 'ci lint scripted client error burst' as specific detection signals.
- Look for 'ci regexp guard rejection' and 'ci regexp oversized non ascii path' entries in Workhorse logs, which the vendor identifies as exploitation attempt indicators.
- Look for 'Onigmo native worker crash' entries in GitLab logs, which the vendor names as a threat detection signal specifically for this CVE.

The weakness behind it

The record classes this as CWE-190 (Integer Overflow or Wraparound).

The sources, in their words

CISA's catalogue entry:

> *GitLab Community Edition and Enterprise Edition contain a integer overflow or wraparound vulnerability that could allow an unauthenticated user to execute arbitrary code on the GitLab server via a specially crafted regular expression in a CI/CD configuration, due to insufficient regular expression validation.*

The CVE record, published by GitLab on 13 October 2026:

> *GitLab has remediated an issue in GitLab CE/EE affecting all versions from 19.2 before 19.2.8, 19.3 before 19.3.5, and 19.4 before 19.4.2 that, under certain conditions, could have allowed an unauthenticated user to execute arbitrary code on the GitLab server via a specially crafted regular expression in a CI/CD configuration, due to insufficient regular expression validation.*

Versions of GitLab the record lists as affected:

- 19.2 up to, but not including, 19.2.8
- 19.3 up to, but not including, 19.3.5
- 19.4 up to, but not including, 19.4.2

The affected ranges end before 19.2.8, 19.3.5, 19.4.2. Those are the first versions the record does not list as affected; confirm them in the vendor's advisory.

CISA's required action:

> *Apply mitigations in accordance with vendor instructions.*

Check your own project

If you build software, you can compare what a project declares with this CVE from a terminal. The tool reads requirements files, package.json, Maven, Gradle, Cargo, CMake and Dockerfile files, runs nothing, and only the CVE id leaves your machine. Naming the CVEs needs no key.

```
lattice-scan project . --cve CVE-2026-97229
```

Each row says in-range, outside-range, undetermined or lead, with the file and line. That is evidence about what a manifest declares, not about what runs, and a person decides. The verdicts, the CSV columns and a pipeline recipe are in the guide.

Check the servers you run

The same id can be checked against a host. Run this on the machine itself, not on the laptop that deploys to it: it reads installed packages (dpkg, rpm or apk), snaps, the names of running container images and the kernel version, compares them on the spot, and only the CVE ids leave the machine; the inventory stays on it, in ~/.lattice-scan. Naming the CVEs needs no key. Do it on each server after a deployment, and again when a new entry lands.

```
lattice-scan --cve CVE-2026-97229
```

A match is a candidate for a person to confirm, not a finding. A distribution can fix a CVE under an unchanged version number, so a match is a candidate; on Debian-family hosts the report also shows

whether the package's own changelog names the CVE. It reads the names of the running images, not what is inside them; for what an image declares, use the project check on its Dockerfile.

Guide: namiq.io/lattice/guides/choose-your-channel. MCP server (io.namiq/lattice in the MCP registry): namiq.io/lattice. Package: pypi.org/project/lattice-scan, or namiq.io/apt.

What we do not know yet

- The sources do not say whether the CI/CD lint endpoint is reachable by unauthenticated users on a default GitLab configuration or only under specific settings.
- No workaround short of patching is described in the vendor's advisory or the Ruby project's notice.
- The sources do not state whether versions older than 19.2 are affected by this specific flaw or whether backported fixes have been issued for them.
- We have not tested this against any environment. Versions and scores are as the sources state them and as at 3 November 2026; the vendor's advisory wins where it differs.
- The sections headed "The bottom line", "Does this apply to you?", "What it is", "Why it matters", "Exposure by situation", "Things people will get wrong", "How we would triage it" and "Send this to your IT provider" are our interpretation of the sources, not a statement by CISA or the vendor.

Appendix 1. Severity decoded

CVE-2026-97229: score 10, `CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H` As the CVE record gives it. The record's own score can differ from the base score on the NVD record.

Metric	Value	What it means here
Attack vector	Network	Reachable over a network, with no need to be on the same segment.
Attack complexity	Low	No special conditions beyond the flaw itself.
Privileges required	None	No account is needed.
User interaction	None	No one has to do anything.
Scope	Changed	Damage reaches beyond the vulnerable component.
Confidentiality	High	Total loss of confidentiality.
Integrity	High	Total loss of integrity.
Availability	High	Total loss of availability.

Appendix 2. Timeline

Date	Event
13 October 2026	CVE record published by GitLab.
29 October 2026	CISA added it to its list of exploited vulnerabilities.
1 November 2026	CISA's deadline for US federal agencies.

Appendix 3. Exposure by situation

These priorities are our judgement from the attacker prerequisites in the sources. They are not vendor ratings.

Your situation	Priority	Timeframe
Self-managed GitLab CE/EE on an affected version, internet-facing	High	Immediately — CISA deadline passed 1 November 2026
Self-managed GitLab CE/EE on an affected version, internal network only	High	This week — no authentication is required to exploit
Self-managed GitLab CE/EE on an affected version with Developer-role users	High	This week — also exposed to CVE-2026-16301 in the same release

Self-managed GitLab on 19.1 or earlier (outside CVE-2026-97229 range)	Medium	Next patch cycle — still exposed to CVE-2026-16301, which begins at 18.7
GitLab.com or GitLab Dedicated — no self-managed install	None	None for this CVE

Appendix 4. Definition of done

- Every GitLab Community Edition and Enterprise Edition you run is listed, with its version.
- Each is outside the affected list, or the vendor's mitigation is in place.
- The version output or a screenshot is saved with the date.
- Your projects' declared dependencies were checked against this CVE, and every row has a written reason.
- Logs for the weeks before the update were searched, and anything odd was written down.
- Anything not done has an owner and a date.

Sources

- CISA Known Exploited Vulnerabilities catalogue: https://www.cisa.gov/known-exploited-vulnerabilities-catalog
- CVE record for CVE-2026-97229: https://www.cve.org/CVERecord?id=CVE-2026-97229
- FIRST: Exploit Prediction Scoring System: https://www.first.org/epss/
- Advisory or release notes read: docs.gitlab.com/releases/patches/patch-release-gitlab-19-4-2-re: https://docs.gitlab.com/releases/patches/patch-release-gitlab-19-4-2-released/
- Advisory or release notes read: gitlab.com/gitlab-org/gitlab/-/work_items/630686: https://gitlab.com/gitlab-org/gitlab/-/work_items/630686
- Advisory or release notes read: www.ruby-lang.org/en/news/2026/10/23/regexp-fixes-for-gitlab-cves: https://www.ruby-lang.org/en/news/2026/10/23/regexp-fixes-for-gitlab-cves/
- Reference in the CVE record: hackerone.com: https://hackerone.com/reports/4055749
- Reference in the CVE record: hackerone.com: https://hackerone.com/reports/4058589