Over 1,000 internet-exposed GitLab servers remain unpatched against actively exploited RCE flaw



More than 1,000 internet-exposed GitLab servers are still unpatched against a maximum-severity flaw that attackers are actively using for remote code execution. Organisations running GitLab should verify their patch status.

- Over 1,000 internet-exposed GitLab servers are still unpatched.
- The flaw is rated maximum severity.
- It is being actively exploited in remote code execution (RCE) attacks.
- Exposure to the internet is the common factor among the affected servers.

Over 1,000 internet-exposed GitLab servers have still not been patched against a maximum-severity vulnerability, according to BleepingComputer. The flaw is being actively exploited in attacks that achieve

remote code execution (RCE).

Why it matters

GitLab commonly holds source code, build pipelines and credentials used to deploy software. A server that allows an attacker to run code remotely can therefore become a foothold into development and production environments. Because exploitation is already under way, unpatched and exposed instances should be treated as at risk now, not in a future maintenance window.

The source does not give further technical detail in the excerpt available to us, so this item does not name the specific vulnerability or fixed versions. Teams should consult the vendor's official security advisory for those details.

What to do next

Institutions that run self-managed GitLab should confirm quickly whether their instances are reachable from the internet and whether the vendor's fix has been applied. Where patching cannot happen immediately, reducing exposure is the sensible interim step. If an instance was exposed while unpatched, it is prudent to review it for signs of compromise and to rotate secrets stored in it.

This is also a reminder to keep development infrastructure inside the same asset inventory, vulnerability management and change processes as other critical systems.

What to do now

1. Identify all self-managed GitLab instances and confirm which are reachable from the internet.
2. Check the vendor's security advisory and apply the fixed version without delay.
3. If patching must wait, restrict access (for example VPN or IP allow-listing) to the instance.
4. Review exposed instances for signs of compromise, such as unexpected accounts, jobs or access logs.
5. Rotate tokens, keys and credentials stored in any instance that was exposed while unpatched.