SHA Pinning Alone Does Not Stop GitHub Actions Supply Chain Attacks



A report says attackers have bypassed SHA pinning of GitHub Actions. In March 2026, those who hijacked the Trivy scanner planted malicious code made to look like it came from GitHub's official checkout tool.

- SHA pinning is widely used as a main defense against software supply chain attacks.
- The source says attackers have already found a way around it.
- In March 2026, the hijackers of the open-source scanner Trivy planted malicious code.
- The code was made to look as if it came from GitHub's official checkout tool.
- The source is a webinar promotion, so technical detail is limited.

What was reported

Pinning GitHub Actions to an exact code version, known as SHA pinning, has become a standard defense against software supply chain attacks. According to The Hacker News, attackers have already found a way around it.

The example cited is from March 2026. The people who hijacked Trivy, a widely used open-source security scanner, planted malicious code that appeared to originate from GitHub's official checkout tool.

Why it matters

Many teams treat a pinned reference as proof that the code is trusted. The report suggests that assumption is incomplete: a pin fixes a version, but it does not by itself show that the code behind it is what it appears to be. Security tooling such as scanners also runs with access to build environments, which makes it an attractive target.

The source is a short promotion for a webinar and does not describe the technique in full, so teams should avoid drawing conclusions beyond what is stated.

What to do next

Treat SHA pinning as one control among several. Review how your pipelines reference third-party actions, who can change them, and what access they hold. The actions below offer a practical starting point.

What to do now

1. Inventory all third-party GitHub Actions and scanners used in your pipelines, including Trivy and checkout steps.
2. Check whether Trivy was used in your pipelines around March 2026 and review the related build logs.
3. Restrict pipeline tokens and secrets to the minimum access each job needs.
4. Add review and approval for changes to workflow files and action references.
5. Include CI/CD supply chain risk in your third-party and secure development risk assessments.