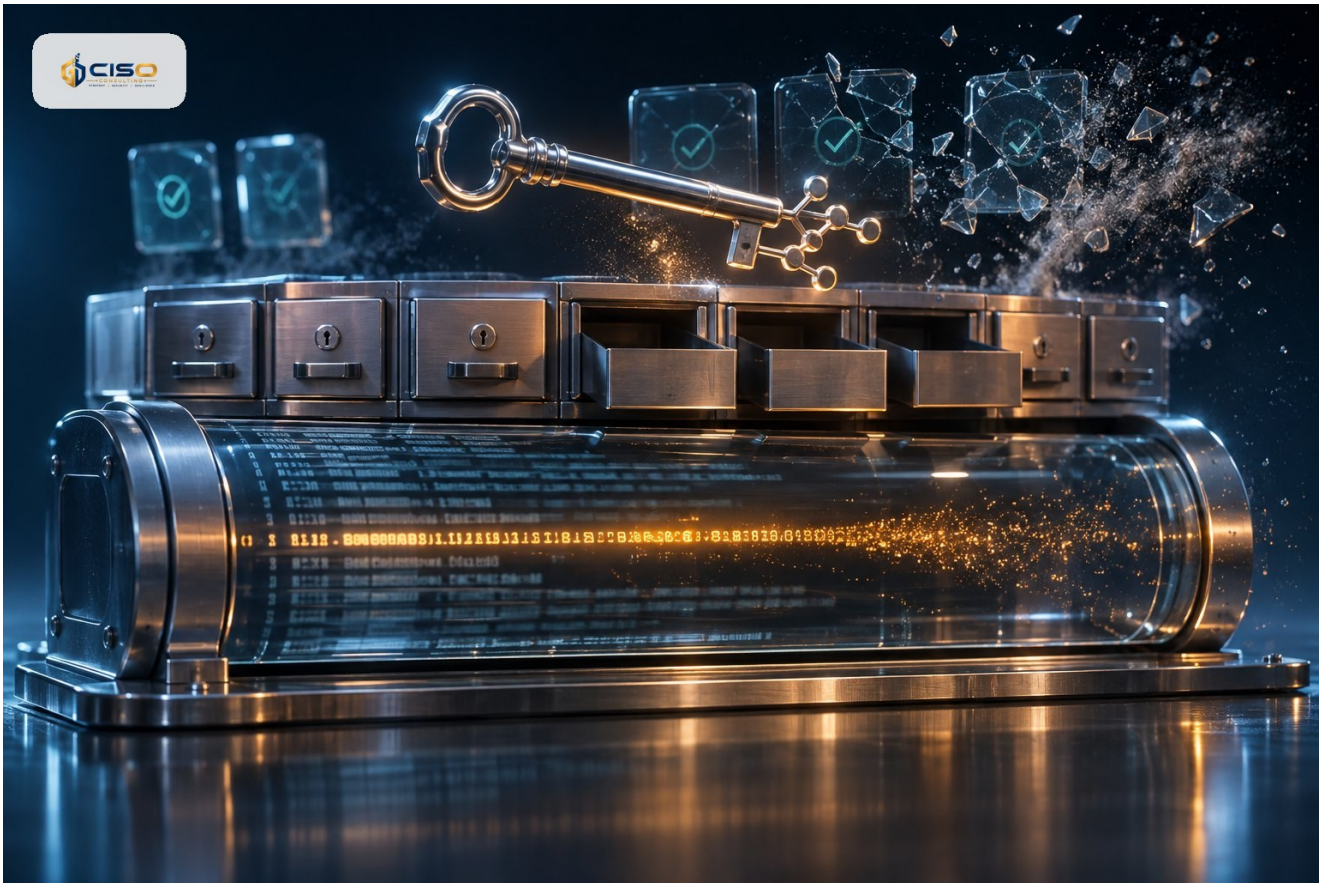
Gheist campaign uses stolen GitHub tokens to pull secrets from Actions logs



Wiz describes an automated campaign that uses stolen GitHub tokens to add workflows that print repository secrets into Actions logs, then downloads the logs and deletes the runs. Private repositories and GitHub Enterprise were affected.

- The campaign, called Gheist, is automated and starts from stolen GitHub tokens.
- Attackers plant workflows that write a repository's secrets into GitHub Actions logs.
- They download the logs and then delete the runs, which reduces visible traces.
- Wiz reports success against private repositories and GitHub Enterprise environments.
- The source mentions a GitHub safeguard, but the excerpt does not give its details.

What was reported

According to an analysis by cloud security company Wiz, reported by The Hacker News, an automated campaign named Gheist abuses stolen GitHub tokens. With a valid token, the attackers add a workflow to a repository. When it runs, the workflow prints the repository's secrets into the GitHub Actions logs.

The attackers then download those logs and delete the workflow runs. Wiz says the technique has worked against private repositories and GitHub Enterprise environments, so private or enterprise hosting alone is not a safeguard.

Why it matters

Secrets stored for pipelines, such as credentials and keys for cloud and production systems, may be exposed without any change to application code. Because the runs are deleted, a review of the visible run history may not show the activity. The source also refers to a GitHub safeguard that pauses something, but the available text is cut off, so we do not describe how it works or how far it helps.

Practical stance

Treat any suspected token theft as a possible secrets exposure, not only as unauthorized repository access. Teams should review which tokens exist, what they can reach, and which secrets sit within their scope.

What to do now

1. Inventory GitHub tokens in use, remove unused ones, and limit the rest to the least access needed.
2. Rotate secrets stored in repositories and Actions if token theft is suspected.
3. Review recent workflow changes and audit-log entries for unexpected workflows, runs or deletions.
4. Restrict who can create or modify workflows, and require review for workflow file changes.
5. Add token compromise and pipeline secrets exposure to your incident response playbooks.