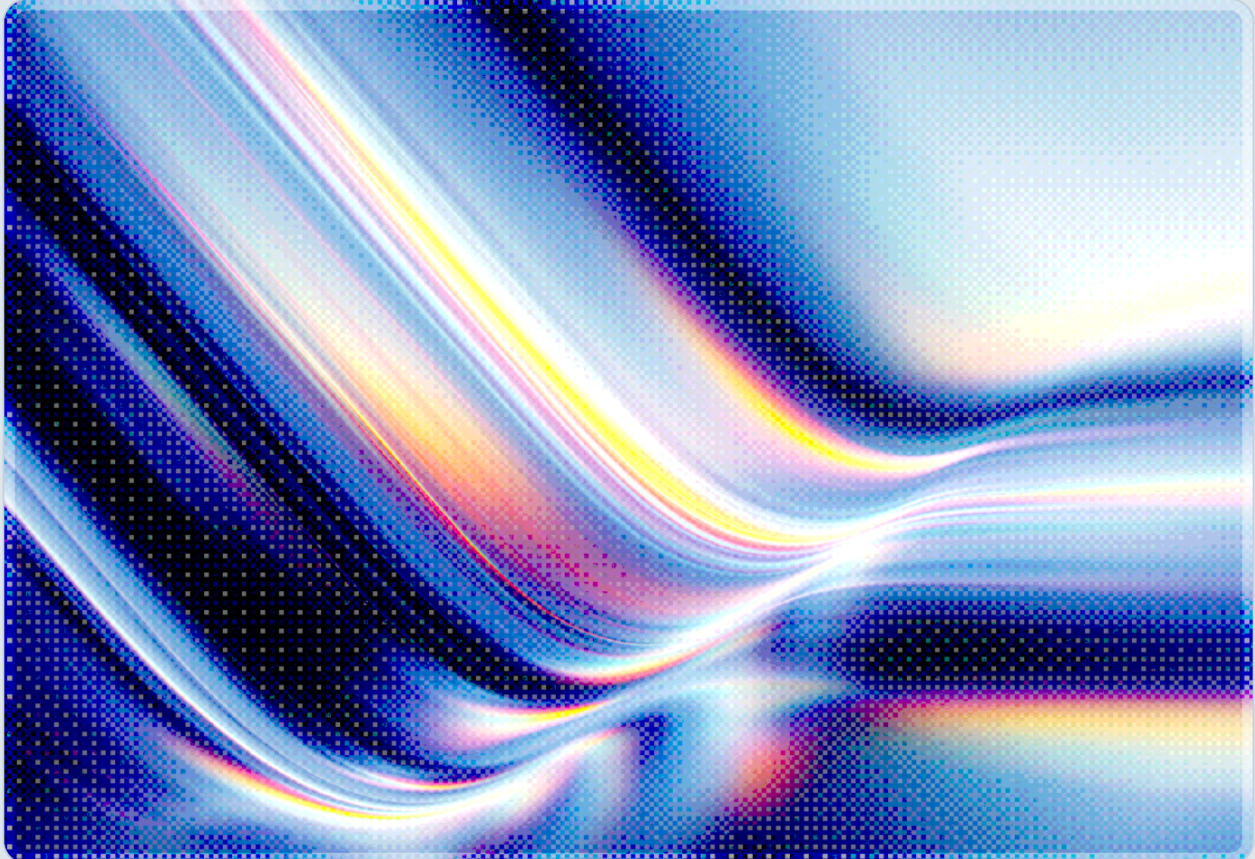


GhostAction Returns: Compromised Maintainer Accounts Steal AI API Keys

Credential-Stealing GitHub Actions Workflows Now Mine Entire Git Histories

2026-10-12

 AI-assisted Rapid Research



CSAI

cloud
security
alliance®

© 2026 Cloud Security Alliance. Some rights reserved.

You may download, store, display, view, print, redistribute, and link to this document in its original, unmodified form, provided that attribution to the Cloud Security Alliance is maintained and all trademark and copyright notices remain intact.

This document may not be modified or altered. You may quote portions of the document as permitted by the Fair Use provisions of the United States Copyright Act, provided that attribution is given to the Cloud Security Alliance.

This document may be shared on professional and social media platforms in its original form with attribution.

This document was generated with AI assistance and has not undergone official CSA review and approval processes.

Key Takeaways

On October 8, 2026, an attacker used two compromised open-source maintainer accounts to push a workflow named `security-audit.yml` to 345 repositories across two bursts on the same day, 279 of them forks. The activity reproduces the GhostAction pattern first documented in September 2025 [1][2]. The workflow sends repository Actions secrets, along with credentials found anywhere in the full git history, to a hard-coded IP address over plain HTTP, and the harvested patterns include Anthropic (`sk-ant-`), OpenAI (`sk-proj-`) and OpenRouter (`sk-or-`) API keys [1][2].

History mining changes the remediation burden. Rotating only the currently configured secrets does not address credentials in commit history, which this workflow reads, and credentials deleted from the working tree years ago remain recoverable there [1]. Published estimates of scale also differ. StepSecurity counted 378 repositories with the workflow present by code search on October 9, while The Hacker News, citing Socket, reports more than 500 accounts and "tens of thousands" of repositories [1][2]. Defenders should treat any repository touched by an affected account as in scope until they have checked it directly. Nearly all technical details in this note rest on a single vendor analysis [1], so readers should weigh them accordingly.

On defenses, StepSecurity observed one repository where exfiltration was blocked by a required-approval setting. Whether that result generalizes to push-triggered workflows from write-access accounts is unconfirmed [1].

Background

GhostAction is a name applied to a family of attacks in which an adversary with write access to a GitHub repository adds a CI workflow that masquerades as a security check but exfiltrates secrets. The first documented wave came to light on September 5, 2025, when GitGuardian reported 3,325 secrets stolen from 817 repositories belonging to 327 users. The stolen secrets included DockerHub, GitHub, and npm tokens, as well as PyPI and AWS credentials [3]. That campaign relied on HTTP POST requests to an attacker-controlled endpoint. PyPI moved the first project it found to read-only status within hours of the alert [3].

The current wave differs in two respects that matter to AI practitioners. First, the target list is explicitly oriented toward AI services. StepSecurity's analysis of the October workflow identifies thirteen credential patterns, including Anthropic, OpenAI, and OpenRouter API keys, alongside AWS keys, GitHub and GitLab tokens, Google and Firebase keys, Slack bot tokens, and SendGrid keys [1]. Second, the workflow searches the full git history rather than only the checked-out tree, which extends the exposure window to secrets that developers believed they had already removed [1].

StepSecurity and Socket reported the October activity publicly on October 9, 2026 [1][2]. According to StepSecurity, an attacker used the reportedly compromised account of Takashi Kitao, author of the pyxel game engine, to push the workflow to 27 repositories between 13:20 and 13:44 UTC on October 8. Roughly eight hours later, between 21:10 and 21:26 UTC, the account of Henry Wu, original author of Uber's athenadriver, was used to push it to 318 repositories, 39 of them source repositories and 279 forks [1]. Nothing in the reporting suggests the account holders were involved, and the public reporting reviewed for this note does not state how the accounts were compromised, so the initial access vector remains unconfirmed.

StepSecurity also attributes earlier infections to a third account dating to September 5, 2026, and describes command-and-control infrastructure first seen on September 4–5, 2026 [1]. Treating these accounts and infrastructure as the work of one operator is StepSecurity's assessment and has not been independently corroborated.

Security Analysis

The reporting supports a clear picture of how the attack proceeds, why AI credentials are exposed to it, and how far it has spread. This section follows that sequence, and then turns to the controls that appear to have mattered and the relationship to earlier supply chain campaigns.

Attack Chain

The attacker obtains credentials for a maintainer account with write access, then commits a workflow file under `.github/workflows/`. Reported file names are `security-audit.yml`, `github_actions_security.yml`, and `security-check.yml`, and the commit messages observed include "Add security audit workflow" and "Trigger security scan" [1][2]. These names are chosen to look routine in a repository that already runs CI. Because the push comes from a legitimate maintainer identity, review habits that focus on unfamiliar contributors offer little protection.

Once the workflow runs, it checks out the repository with full fetch depth, then extracts secrets from two places. Named Actions secrets configured for the repository are exposed through the workflow's own environment, and a `git log -p --all` pass collects credential-shaped strings from every commit on every branch [1]. The workflow tags output with markers such as `AKIA_CTX_START` and `AKIA_CTX_END` so that AWS access key IDs can be paired with nearby secret keys, and it posts the result to `193.32.204[.]199` over unencrypted HTTP without a DNS lookup [1][2]. StepSecurity's run-log evidence for `uber/athenadriver` shows the server returning an acknowledgment about four seconds after the workflow started, which suggests that in that run exfiltration finished before a human could have noticed [1].

Why AI API Keys Are in Scope

CSA assesses that AI provider keys are attractive for reasons that differ from cloud credentials. They are often handled casually during prototyping, in test fixtures, notebooks, `.env` examples, and CI configuration, which is the kind of handling that leaves them in history. They also carry direct monetary value, since a stolen key lets an attacker consume paid model capacity at the victim's expense, and a key with broad scope may expose prompts, stored files, or connected tools. This reasoning is the author's assessment and not a finding of the cited reporting. No figure for stolen Anthropic, OpenAI, or OpenRouter keys has been published for this wave, so the impact is described by capability and not by count.

CSA's earlier research on LiteLLM gateway vulnerabilities and on IDE plugin campaigns shows the same pattern from other directions: attackers go after the place where many AI credentials are concentrated or casually handled [6][7]. GitHub repositories and their CI systems are another such place. A single secret-bearing repository can hold the keys for model providers, cloud accounts, package registries, and source hosting at once.

Scale and Uncertainty

Published counts do not fully reconcile, and the table below sets them side by side with the source and window each one covers.

Figure	Reported by	Scope or window
345 repositories (27 + 318, of which 279 are forks)	StepSecurity [1]	Pushes on October 8, 2026 by two accounts

Figure	Reported by	Scope or window
378 repositories with the workflow	StepSecurity [1]	Code search on October 9, 2026
182 repositories with the history-mining marker	StepSecurity [1]	Code search on October 9, 2026
88 repositories whose requests carried the named-secrets marker	StepSecurity [1]	Code search on October 9, 2026
More than 500 accounts and "tens of thousands" of repositories	The Hacker News, citing Socket [2]	Window not stated
772 public repositories across 373 users or organizations	The Hacker News, attributing the figure to GitGuardian [2]	August 31 to September 30, 2026

The differences may reflect different measurement methods, such as repositories found by search versus repositories reached by account, as well as forks inflating counts. The forks matter in particular because they do not inherit the parent repository's Actions secrets, so their exposure differs from that of source repositories. Readers should not treat any one figure as authoritative, and an organization's own audit is the only reliable measure of its exposure.

Control Effectiveness

Two defensive observations stand out. StepSecurity reports that `typecho-fans/plugins` was protected because the repository required approval before workflow runs, which it describes as the control that stopped the campaign in that case [1]. GitHub's documentation on approving workflow runs focuses on runs triggered from forks, and the behavior for pushes by an account with write access depends on repository configuration, so this single observation should not be read as proof that the setting blocks the attack generally [4].

The history sweep also means secret scanning and rotation must cover past commits. GitHub's secret scanning can flag known token formats [5], but in CSA's assessment it cannot undo exposure for a credential that was already readable to a workflow that ran.

Relationship to Earlier Supply Chain Campaigns

This activity belongs to a series of 2026 attacks that use stolen developer credentials and CI trust. CSA has documented the Miasma worm's use of a stolen GitHub credential and Actions OIDC abuse against Red Hat npm packages, and a related CI/CD bypass in the AsyncAPI compromise [8][9]. Unlike the Miasma payload, GhostAction contains no worm or implant. Its payload is a short script, yet it reached hundreds of repositories, which makes it a useful indicator of how readily a single maintainer identity converts into broad access. The StepSecurity report also advises package maintainers to delay releases until cleanup is complete, because a compromised maintainer account can publish to registries as well as push workflows [1].

Recommendations

The recommendations below are ordered by time horizon, beginning with the steps an affected organization should take now and moving toward structural changes to how CI credentials are held.

Immediate Actions

Search every organization and personal repository for `security-audit.yml`, `github_actions_security.yml`, and `security-check.yml` on all branches and in forks, and review Actions run history since August 31, 2026 [1][2]. If one appears, treat it as a confirmed breach. Rotate all configured Actions secrets, and also rotate every credential that was ever committed to the repository, including AI provider keys, cloud keys, registry tokens, and GitHub or GitLab tokens [1]. Identify and revoke the compromised personal access token or session, delete the malicious workflows from every branch, and pause package releases until the review is done [1]. Finally, hunt in network logs for outbound connections to `193.32.204[.]199` and the earlier addresses `45.139.104[.]115` and `170.39.218[.]2` [1].

Short-Term Mitigations

Enable required approval for workflow runs where the repository settings allow it, and restrict who can create or modify files in `.github/workflows/` using branch protection and CODEOWNERS rules. Documentation [4] covers fork-triggered runs only, so the branch and ownership controls are the more reliable guard against a push from a write-access account. Set the default `GITHUB_TOKEN` to read-

only and grant write permissions per job, in line with GitHub's hardening guidance [10]. Move secrets from repository scope to environment scope with required reviewers, so a newly added workflow cannot read them without passing the environment's reviewer and branch rules.

Where an AI provider supports it, use short-lived or project-scoped keys with spend limits, and set up alerts on unusual usage. Require phishing-resistant multi-factor authentication for maintainers and prefer tokens with narrow scopes and expiry, noting that the compromise vector for these accounts has not been published and that stolen tokens or sessions may bypass MFA.

Strategic Considerations

Treat the CI system as a credential store that needs the same monitoring as a secrets vault. This means egress allow-listing for runners, so plain-HTTP posts to arbitrary IP addresses fail, and alerting on new workflow files created outside a reviewed pull request. Rely less on long-lived static keys by adopting OIDC federation to cloud providers, and ask AI providers for scoped, expiring keys and workload-identity federation as procurement requirements.

Run pre-commit and server-side push protection to stop credentials entering history in the first place, and assume history is exposed in any repository where the workflow executed. Finally, include maintainer account recovery and session revocation in incident response plans, since this campaign shows that one account can reach hundreds of repositories, including forks, in minutes.

CSA Resource Alignment

CSA's research note on the Miasma worm and Red Hat npm compromise is the closest prior work, because it describes how a stolen GitHub credential and Actions trust were used to reach package registries and to pull credentials from developer environments [8]. Its findings on CI identity and publication controls are relevant to the maintainer-takeover step observed here. The related AsyncAPI note documents a CI/CD bypass that led to a malicious release, which reinforces the recommendation to gate workflow creation and releases separately [9].

CSA's analysis of AI developer toolchain supply chain attacks addresses the broader pattern in which AI API keys and developer credentials are the target of attacks on tools, plugins, and dependencies, and it frames AI keys as high-value non-human credentials [6]. The LiteLLM gateway advisory shows the consequence of concentrated AI keys when an intermediary is compromised, and supports the guidance on scoping, spend limits, and rotation [7]. As a standing framework, the AI Controls Matrix v1.1 provides

control objectives for identity and access management, supply chain management, and application and interface security that map to the mitigations above, including least-privilege tokens, secret handling, and change control over pipeline definitions [11].

References

- [1] StepSecurity. "[GhostAction Returns: Malicious “Security Audit” Workflows Now Mine Credentials from Entire Git Histories](#)." StepSecurity, October 9, 2026.
- [2] The Hacker News. "[Credential-Stealing GitHub Actions Workflows Planted in Tens of Thousands of Repositories](#)." The Hacker News, October 2026.
- [3] GitGuardian. "[GhostAction Campaign: 3,325 Secrets Stolen](#)." GitGuardian Blog, September 2025.
- [4] GitHub. "[Approving workflow runs from forks](#)." GitHub Docs.
- [5] GitHub. "[About secret scanning](#)." GitHub Docs.
- [6] Cloud Security Alliance. "[AI Developer Toolchain Supply Chain Attack Research Note](#)." CSA Labs, June 2026.
- [7] Cloud Security Alliance. "[LiteLLM AI Gateway: Critical Vulnerability Chain Exposes API Keys](#)." CSA Labs, 2026.
- [8] Cloud Security Alliance. "[Miasma npm Supply Chain Worm: Red Hat Compromise](#)." CSA Labs, June 2026.
- [9] Cloud Security Alliance. "[AsyncAPI npm Compromise: CI/CD Bypass Delivers Miasma RAT](#)." CSA Labs, July 2026.
- [10] GitHub. "[Security hardening for GitHub Actions](#)." GitHub Docs.
- [11] Cloud Security Alliance. "[AI Controls Matrix v1.1](#)." Cloud Security Alliance.