NETRISE

BeEF Compromise

1. Executive Summary

On January 12, 2026, NetRise identified a confirmed software supply-chain compromise affecting the BeEF (Browser Exploitation Framework) open-source project. The issue is a critical "pwn request" vulnerability (CVSS v3.1 9.3) introduced via a change to the project's GitHub Actions workflow. The workflow configuration enables any GitHub user to exfiltrate repository secrets and execute arbitrary code simply by opening a pull request, with no merge or maintainer approval required.

Using a proprietary AI-powered engine designed to uncover developer identities and detect code security threats at scale, we attribute the malicious workflow change to the contributor zinduolis (ilgakulnis@gmail.com) with high confidence and assess the actor as operating from Moscow, Russia (UTC+3), with a 90% confidence based on temporal commit-pattern alignment and corroborating signals. The contributor is also flagged due to operation from a DDTC-restricted country (Russia).

If exploited, the vulnerability enables theft of sensitive secrets including BROWSERSTACK_USERNAME, BROWSERSTACK_ACCESS_KEY, and a GITHUB_TOKEN with write permissions, which can lead to full repository compromise, workflow modification, and potential backdoored releases.

NetRise validated the exploit in a private, isolated reproduction environment and prepared a responsible disclosure package with immediate remediation options and hardening guidance.

2. Methodology

NetRise Contributor Intelligence is an AI-powered engine designed to uncover developer identities and detect code security threats at scale. It utilizes two primary capabilities:

Location Intelligence

- De-anonymizes actors through temporal pattern analysis (commit timestamp histograms), holiday-inactivity matching, linguistic fingerprinting, social media country detection, collaborative network inference, GitHub profile geocoding, and AI synthesis for weighted signal fusion.

Commit Intelligence

- Detects malicious intent via heuristic risk scoring with threshold flagging and AI-powered analysis for supply chain risks, data exfiltration, and credential harvesting using structured threat assessment.

3. Details of the research and its findings

Incident

Project: BeEF (Browser Exploitation Framework)
Threat Level: Critical (CVSS 9.3)

Using the capabilities described above, we identified a confirmed supply-chain compromise in the high-profile open-source security tool BeEF. Our analysis identified a foreign contributor who introduced a well-known "pwn request" vulnerability designed to exfiltrate secrets and execute malicious code.

The Contributor (Location Intelligence)

- We successfully de-anonymized the contributor responsible for the malicious code, zinduolis.
- **Identity:** zinduolis (Email: ilgakulnis@gmail.com)
    - **Detected Location:** Moscow, Russia
    - **Intelligence Signal:** High Confidence (90%). AI temporal analysis confirmed their work patterns match Moscow time (UTC+3) with consistent morning work schedules (7:30 AM–2:30 PM Moscow time).
    - **Risk Flags:** The contributor is flagged for operating from a DDTC-restricted country (Russia).

The Vulnerability (Commit Intelligence)

The actor introduced a "Pwn Request" vulnerability via commit e394f2e1.

What It Is: The workflow uses pull_request_target as a trigger, checks out untrusted code from the pull request's HEAD, and executes it. The pull_request_target event runs in the context of the base repository with full access to secrets, but the workflow checks out and executes code from the fork (the pull request HEAD). This allows an attacker to execute arbitrary code with access to repository secrets.

The Mechanism: The attacker modified the GitHub Actions workflow file (.github/workflows/github_actions.yml). They changed the trigger from pull_request to pull_request_target, which runs in the context of the base repository rather than the fork.

The Exploit: By removing explicit SHA pinning, the workflow now checks out and executes untrusted code from the attacker's fork while retaining access to high-value secrets.

This is a well-documented attack pattern known as a "pwn request" vulnerability.

Impact / Business Impact

If integrated or left in place, this vulnerability grants an attacker:

Arbitrary Code Execution: The ability to run any code they wish on the build servers.

Credential Theft: Immediate access to sensitive environment variables, specifically BROWSERSTACK_USERNAME and BROWSERSTACK_ACCESS_KEY.

GITHUB_TOKEN Exfiltration and Privilege Abuse: The GITHUB_TOKEN is persisted in git config (default behavior of actions/checkout) and can be extracted. Based on workflow logs, this token has:

- Contents: write — Can push code directly to master
- Actions: write — Can modify workflows
- Packages: write — Can publish releases
- PullRequests: write — Can merge pull requests

Supply Chain Poisoning / Full Repository Compromise: With the extracted GITHUB_TOKEN, an attacker can push malicious code, create backdoored releases, modify workflows to maintain persistence, or poison the build/release process.

Severity

CVSSv3 Score: **9.3 (Critical)**
Vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:N
Attack Complexity: Low — requires only a GitHub account
Privileges Required: None — any user can fork and submit a PR

Validation and Reproduction

We performed a security review and successfully reproduced the exploit in a private, isolated environment to confirm that GITHUB_TOKEN (with write permissions) and BrowserStack credentials could be exfiltrated without maintainer approval.

To avoid drawing attention to the issue in the BeEF project, reproduction was conducted in a separate test repository with equivalent configuration. Supporting logs and results were captured from the reproduction workflow run.

Responsible Disclosure & Remediation

We have delivered a responsible disclosure package to address this critical vulnerability (CVSS 9.3), to which contributors to the project responded very quickly, providing an action plan:

Immediate Block: Block commit e394f2e1 and flag the contributor zinduolis.

Fix Options:

`pull_request_target` reduces developer friction, and it's very convenient, but it's unsafe.

See: https://www.google.com/search?q=pull_request_target+supply+chain+attack&oq=pull_request_target+supply+chain+attack for a list of high profile supply chain attacks that exploited this weakness.

Immediate Fix (Option A): Switch to pull_request trigger

If secrets are not required for fork PRs, use `pull_request` instead. Tradeoff: BrowserStack tests won't run for fork PRs (no credentials). You could run them only on push to master after merge.

Immediate Fix (Option B): Don't checkout PR code

If you must use `pull_request_target`, never checkout and execute code from the PR

Recommended Fix (Option C): Split into two workflows

Workflow 1: Runs on pull_request (no secrets, safe to run untrusted code)
Workflow 2: Runs on workflow_run after PR merge (has secrets, runs trusted code)

Additional Hardening Recommendations:

- Enable branch protection on master:
- Require pull request reviews
- Require status checks
- Disable force pushes

Recommendations

Immediate containment (hours)

- Revert or block the malicious change: Remove/rollback workflow modifications associated with commit e394f2e1 and audit subsequent commits for related changes.
- Rotate exposed secrets immediately:
  - `BROWSERSTACK_USERNAME`
  - `BROWSERSTACK_ACCESS_KEY`

    - Any tokens accessible to GitHub Actions, including GITHUB_TOKEN-equivalent credentials (and any third-party tokens used in CI).
- Audit GitHub Actions permissions:
    - Reduce default token permissions where possible.
    - Review repository settings for workflow permissions and token scopes.

5. Conclusion

NetRise identified and validated a critical pwn-request class supply-chain vulnerability in the BeEF project’s GitHub Actions workflow that could allow any external GitHub user to execute arbitrary code and exfiltrate secrets without maintainer approval. The change is attributed with high confidence to the contributor zinduolis, assessed as operating from Moscow, Russia, and flagged due to operation from a DDTC-restricted jurisdiction.

This incident reinforces a recurring supply-chain lesson: CI/CD workflows are production-critical attack surfaces, and small configuration changes—especially around `pull_request_target`, permission scopes, and checkout behavior—can convert routine automation into a repository-compromise path. Immediate rollback, secret rotation, and secure workflow redesign (preferably split-workflow patterns) materially reduce risk and prevent recurrence.