# Software supply chain: would a poisoned package or action reach production?

This month, make new package versions wait three days before they install, lock every GitHub Action to a commit and replace long-lived publishing and cloud tokens with trusted publishing and OIDC. Judge every other control by whether it keeps credentials away from code you didn't write, or tells you within an hour where a bad version ran.


| Reviewed | Next review | Version | Written for |
|---|---|---|---|
| **5 Nov 2026** | **5 Dec 2026** | **1.2** | **Development, platform and DevOps teams that build and ship software, including small teams on GitHub** |

Latest version
**sheets.einblick.dev/supply-chain/**


## If you only do five things

Third parties, including flaws in vendor software, were involved in 48% of the breaches in Verizon's 2026 DBIR, up 60% in a year[1].

1. **Make new versions wait three days.**
   Poisoned releases are often caught within hours: PyPI quarantined the LiteLLM and Telnyx ones within four hours of upload[2]. Dependabot waits three days by default and pnpm one day[3,4]; set npm's `min-release-age=3`[5] and a P3D cooldown in uv or pip[2]. CISA suggested seven days after the axios compromise[6].

2. **Block install scripts and install from the lockfile.**
   Shai-Hulud 2.0 ran from a `preinstall` script, before any test or security check[7]. npm 12 blocks dependency install scripts by default and pnpm runs only approved ones[8,9]; approve the few that need it with `npm approve-scripts`[8]. On older npm set `ignore-scripts=true`[6]. Commit lockfiles with hashes[2] and install with `npm ci`.

3. **Lock every action to a commit.**
   In March 2026 an attacker force-pushed 76 of 77 version tags of `trivy-action` to code that read secrets from runner memory[12]. Turn on **Require actions to be pinned to a full-length commit SHA**, which a workflow lockfile satisfies, and set **Workflow permissions** to read-only[13].

4. **Publish and deploy without long-lived tokens.**
   GitHub calls removing long-lived credentials from CI/CD the number one thing you can do[14]. Publish to npm and PyPI with trusted publishing[15,2], set npm's **Require two-factor authentication and disallow tokens**[15], delete the old tokens and deploy to clouds with OIDC[16]. Require phishing-resistant MFA on developer accounts[6].

5. **Record what every build contained.**
   Generate an SBOM in the build for each release and keep it with the artifact[17,18], so you can tell within an hour whether you shipped a bad version. The CRA requires one from December 2027[19].

## When a package or action you use is reported compromised

1. **Did a bad version run anywhere?**
   Search lockfiles, caches and artifact registries, then CI pipelines and developer machines that installed packages in the exposure window[6]. OSV-Scanner checks lockfiles and SBOMs against OSV.dev[10], which includes OpenSSF's malicious-package (`MAL-`) records[11].

2. **Is the machine still infected?**
   Isolate it before you revoke anything: Mini Shai-Hulud wipes the home directory if its GitHub token is revoked[20]. Shai-Hulud 2.0 registered a GitHub Actions runner called `SHA1Hulud` on infected hosts[7]. Remove persistence and rogue runners, or rebuild.

3. **Which secrets could it reach?**
   Rotate every secret the job or machine could read, as CISA advises for a compromised CI run[6], and do it in one go: the Trivy attacker may have kept access because an earlier rotation wasn't atomic[12].

4. **What did the attacker do with them?**
   Look for new repositories (Trivy's payload created `tpcp-docs-` ones[12]), workflow commits by automated accounts[21], versions you didn't publish and unusual cloud API calls.

5. **Do you have to report it?**
   If malicious code could have reached a product you place on the EU market, treat it as a possible severe CRA incident, with an early warning due within 24 hours[19]. The EU security laws sheet has the other clocks.

## If you place software on the EU market: the CRA

- **Reporting applies now.** Since 11 September 2026, manufacturers must report actively exploited vulnerabilities and severe incidents, also for products placed on the market before the rest of the CRA applies on 11 December 2027[19].
- **A poisoned release counts.** An incident that has led or could lead to malicious code in your product or your users' systems is severe, such as malicious code in your update channel[19].

## What changed in 2026

- **Attacks moved into CI and release workflows.** Trivy's compromise spread through retagged actions in March[12]. Mini Shai-Hulud scraped runner memory for identity tokens and swapped them for npm trusted-publisher tokens[20]. In May CISA reported that a poisoned VS Code extension update led to theft of GitHub's internal repositories[21].
- **Safer defaults shipped.** npm 12 blocks dependency install scripts and git and URL dependencies[8,5], and since July Dependabot waits three days before version updates[3].
- **GitHub Actions got lockfiles.** Since 28 October, `.github/workflows/actions.lock` pins actions to commits on Free, Team and Enterprise Cloud plans, and a **Require lockfile** policy enforces it[22].

- **The SBOM must be machine-readable, not public.** It needs a commonly used format, must cover at least top-level dependencies and goes to authorities on request; sharing it with users is optional[19]. CISA's 2026 minimum elements add transitive dependencies, hashes and licenses[17].
- **Vulnerability handling is due by December 2027.** Have a disclosure policy, a reporting contact, free security updates and public notes on fixed flaws for at least 5 years, or the expected use time if shorter. Report flaws in components to their maintainers[19].

## What you can skip

- **Chasing unreachable CVEs deep in the tree.** The CRA asks you to remediate in relation to the risks[19]. Fix exploited and reachable ones within the patching sheet's deadlines and batch the rest.
- **Polishing SBOMs nobody reads.** Generate them in the build with tools you already have, as ENISA advises for a first implementation[18]. Buy an SBOM platform only when a customer or regulator consumes them.
- **SLSA Build L3 before the basics.** Close the moving tags, long-lived tokens and install scripts behind most 2026 incidents first.
- **A registry proxy for a small team.** A cooldown, a lockfile and blocked install scripts give most of its benefit without the upkeep.

## Mistakes that cost the most

- **Publishing from a busy job.** Malware in the same job can read the OIDC token from runner memory[20]. Publish from a separate job that installs nothing, in an environment that needs a reviewer's approval[24].
- **Running fork code with secrets.** `pull_request_target` workflows are privileged, bypass fork approval and must not check out untrusted code[16,13]. From 2 December GitHub blocks them in public repositories by default[26].
- **Forgetting developer machines.** The poisoned Nx Console extension reached users through VS Code's automatic updates[21]. Allowlist extensions and pin approved versions with the `AllowedExtensions` policy[25].

## What this looks like at your size

### Small team, no security staff

- In the organization's Actions settings, set read-only workflow permissions and require SHA pinning.
- Keep Dependabot's default cooldown and turn on its malware alerts, security updates and secret scanning (Secret Protection for private repos)[28].
- Move npm and PyPI publishing to trusted publishing (npm needs cloud-hosted runners for it[15]), and run npm 12 or current pnpm in CI.

### Growing, with IT or security staff

- Add `.github/workflows` to CODEOWNERS and turn on **Require review from Code Owners** (Team plan or higher for private repos)[27].
- Publish from an environment with required reviewers and **Prevent self-review** (public repositories only on Free, Pro and Team)[24].
- Run OSV-Scanner on lockfiles in every pull request.

### Enterprise or regulated

- Route installs through a registry proxy that enforces a cooldown and checks for malware.
- Verify GitHub artifact attestations before deploy; private repos need Enterprise Cloud[28].
- Keep each release's SBOM for 10 years or the support period, if longer[19], and add build pipeline compromises to your CRA reporting path.

### Version history

**5 Nov 2026, version 1.2** Noted that npm trusted publishing works only on cloud-hosted CI runners.

### Sources


1. 2026 Data Breach Investigations Report: Executive Summary. Verizon, 2026. verizon.com
2. Incident Report: LiteLLM/Telnyx supply-chain attacks, with guidance. PyPI Blog, 2 Apr 2026. blog.pypi.org
3. Dependabot version updates introduce default package cooldown. GitHub Changelog, 14 Jul 2026. github.blog
4. Settings: Dependency resolution. pnpm, checked 3 Nov 2026. pnpm.io
5. npm config (v12): min-release-age, allow-git, allow-remote. npm Docs, checked 3 Nov 2026. docs.npmjs.com
6. Supply Chain Compromise Impacts Axios Node Package Manager. CISA, 20 Apr 2026. cisa.gov
7. Shai-Hulud 2.0: Guidance for detecting, investigating, and defending against the supply chain attack. Microsoft Defender Security Research Team, 9 Dec 2025, updated 13 May 2026. microsoft.com
8. npm install-time security and GAT bypass2fa deprecation. GitHub Changelog, 8 Jul 2026. github.blog
9. Settings: Build (allowBuilds, strictDepBuilds). pnpm, checked 3 Nov 2026. pnpm.io
10. Project Source Scanning. OSV-Scanner Docs, checked 3 Nov 2026. google.github.io
11. OSV.dev data sources. OSV.dev Docs, checked 3 Nov 2026. google.github.io
12. Trivy ecosystem supply chain temporarily compromised (GHSA-69fq-xp46-6x23). Aqua Security, 21 Mar 2026. github.com
13. Managing GitHub Actions settings for a repository. GitHub Docs, checked 3 Nov 2026. docs.github.com
14. Disrupting supply chain attacks on npm and GitHub Actions. GitHub Blog, 28 Jul 2026. github.blog
15. Trusted publishing for npm packages. npm Docs, updated 6 Oct 2026. docs.npmjs.com
16. Secure use reference (GitHub Actions). GitHub Docs, checked 3 Nov 2026. docs.github.com
17. 2026 Minimum Elements for a Software Bill of Materials (SBOM). CISA, NSA, FBI and international partners, 29 Jul 2026. cisa.gov
18. ENISA Technical Advisory for Software Bill of Materials. ENISA, Oct 2026. enisa.europa.eu
19. Regulation (EU) 2024/2847 (Cyber Resilience Act), Articles 13, 14, 69 and 71, recital 68 and Annexes I, II and VII. EUR-Lex, 2024. eur-lex.europa.eu
20. Mini Shai-Hulud (S9043). MITRE ATT&CK, modified 31 Jul 2026. attack.mitre.org
21. Supply Chain Compromises Impact Nx Console and GitHub Repositories. CISA, 28 May 2026. cisa.gov
22. GitHub Actions workflow lockfiles are generally available. GitHub Changelog, 28 Oct 2026. github.blog
23. Stage-only npm tokens for safer automation. GitHub Changelog, 18 Sep 2026. github.blog
24. Deployments and environments. GitHub Docs, checked 3 Nov 2026. docs.github.com
25. Manage extensions in enterprise environments. Visual Studio Code Docs, checked 3 Nov 2026. code.visualstudio.com
26. pull_request_target blocked by default December 2. GitHub Changelog, 2 Nov 2026. github.blog
27. About code owners. GitHub Docs, checked 3 Nov 2026. docs.github.com
28. GitHub security features. GitHub Docs, checked 5 Nov 2026. docs.github.com