

# What immutable releases protect in GitHub Actions workflows

EasyxLab · EasyByte Hub S. Coop. Mad. · Study S18

 Working draft Published 2026-10-03

---

## Abstract

Since 2025-10-28 the Git tag of an immutable GitHub release cannot be moved or deleted. GitHub's documentation keeps major version tags such as `v1` outside releases, so that they stay movable, and suggests authors "recommend that your users specify a major version". We measured what immutable releases protect in the workflows of the top 1,000 repositories by stars with recent activity (A, cut-off 29,565 stars) and of 500 repositories drawn at random from the 20,553 with 1,000–5,000 stars (B), on 2026-10-03, through the GitHub API: 58,690 remote `uses:` references in A and 10,472 in B.

Of the tag references to actions whose latest release is immutable, 312 of 6,898 (4.5%) point to a protected tag in A (95% interval 2.8–7.1%) and 76 of 1,564 (4.9%) in B; for 3,238 of 6,586 (49.2%) of the unprotected ones in A, an immutable tag already points to the same commit. 28 of 50 (56.0%) of the most-used action repositories publish immutable releases; 12 of 28 (42.9%) for `actions/*`, 3 of 6 (50.0%) for `github/*`. SHA pinning covers 32,978 of 58,690 (56.2%) references in A and 4,812 of 10,472 (46.0%) in B, but the median repository pins 5.0% (A); at most 263 of 891 (29.5%) repositories in A and 65 of 398 (16.3%) in B pin every written reference. Immutable tags: 0.6% of references in A, 0.8% in B. GitHub's workflow-level dependency lock had not shipped on the measurement date.

---

## 1. Introduction

In March 2025 the tags of the popular action `tj-actions/changed-files` were moved to a malicious commit, and every workflow that referenced the action by tag ran the new code. Since then GitHub has shipped two mechanisms that bear on the problem. On 2025-08-15 the allowed-actions policy gained blocking and a rule that requires actions to be pinned to a full-length commit SHA. On 2025-10-28 *immutable releases* became generally available: "Tags for new immutable releases are protected and can't be deleted or moved."

Immutable releases protect a tag only if that tag belongs to a release, and GitHub's guidance keeps the tag most workflows use outside any release: authors should "create a major version tag (for example, `v1`)" without a release, and "recommend that your users specify a major version" (§3). A workflow that writes `@v1` does what the documentation recommends. GitHub's stated direction is to move "away from mutable references and towards immutable releases" (roadmap, 2026-03-26).

The question is not whether projects "do it wrong" but what the mechanism protects in practice: how many references point to a tag that cannot move, how many could, and how much SHA pinning already covers. We answer it for 2026-10-03, before GitHub's announced dependency lock ships.

---

## 2. Prior work

**Search (2026-10-03).** OpenAlex ( "immutable releases" , "GitHub Actions" pinning , "GitHub Actions" "commit SHA" , "GitHub Actions" security workflows , the titles below, and title filters), reading the first 25 results of each query (for "immutable releases" , 25 of 147; none concerns GitHub releases); Crossref for the cited DOIs; GitHub repository search for tools. arXiv and DBLP search were not queried (their robots.txt disallows it). Two of the items below and the DSN-S 2026 paper were found by the independent review, not by our first search. Log: `data/sources/prior_work_search.json` .

### Trackers and tools.

- **datosh/pinned-actions** (open source, MIT; results site `pin-gh-actions.kammel.dev`) has measured SHA pinning in the top 10,000 repositories by stars since 2024. Its page says "only 7% of GitHub repositories fully embrace this security best practice" and "The data was last updated in April 2026". Its published April 2026 archive gives 506 of 6,290 repositories that use Actions (8.0%) with every action reference pinned (a 40-hex SHA or a `sha256:` image digest). Since a commit of 2026-04-27 its code also records `latest_release_immutable` ( `GET /repos/{o}/{r}/releases/latest` ) for each repository, the same rule as our *immutable publisher* and §6.5; the April 2026 archive has no such field, and we found no published figure from it.
- **joshjohanning/ensure-immutable-actions** (Marketplace action, created 2025-11-07) "validates that third-party actions in your workflows ... use immutable releases": a reference passes if it is a full SHA or the tag of an immutable release, so major tags without a release fail. With `suggest-pins` it proposes a full commit SHA (with the referenced tag, or the latest release, as comment).
- **zizmor** (2026-10-02: "GitHub does not have immutable branches or tags") and **pinact** model SHA pinning only.

### Papers.

- **Koishybayev et al., USENIX Security 2022.** Of 601,338 references to third-party actions, 78.8% use a tag name, 20% a branch name and 1% (6,539) a commit hash (Table 5); the text says "only 0.1% of references use commit hash" and that "less than 2% of all repositories follow the security guidelines provided by Github regarding commit hash references".
- **Decan et al., ICSME 2022.** Among steps using an action from another repository, 93.0% (258,647) use a version tag, 5.3% a branch or other tag and 1.7% (4,601) a commit SHA.
- **Huang and Lin, ICPC 2025.** 18,938 workflows from 5,246 repositories in 2022 and 2024; 16,439 "Unverified Action Version Unpinned" issues in 2024 (issue counts, not shares).
- **Kubo et al., NDSS 2026.** 338,812 repositories and a survey. Practice P4, "Pinning Third-party Actions", is implemented by 16.2% (37,693 of 233,124) of repositories, where every third-party action must be referenced by a full 40-character SHA *or* be from a creator with a verified badge and be referenced by a tag. It is therefore not a SHA-only measure.
- **Chaiwut and Nikiforakis, SecDev 2025.** Over 23K Marketplace actions over four months; typosquatting and dangling references (abstract only; no open copy found).
- **Boschanski and Vieira, DSN-S 2026, Evaluating Security Best Practices in the GitHub Actions Documentation.** Directly relevant to our point about what the documentation recommends, but we

- found neither an abstract nor an open copy (Crossref and OpenAlex, 2026-10-03), so we cite it without describing its findings.

**Reconciling "pin everything".** Our at most 263 of 891 (29.5%) in A against datosh's 8.0% differ in population (our A is the top 1,000 with a push in the last 90 days and no archived repositories; datosh takes the top 10,000 regardless of activity, where stale workflows weigh more), in date (October against April 2026, six months of post-tj-actions migration), and in unit (datosh also counts container image references). Its first 1,000 entries give 100 of 781 (12.8%). We did not re-run their tool, so we cannot apportion the gap.

**What is new here, as far as we can find.** H1, the share of tag references that immutable releases actually protect where the action already publishes them, and the share of unprotected references that have an immutable tag at the same commit (49.2% in A). The adoption figures (H2) and SHA pinning (H3) extend existing trackers to a dated, activity-filtered population. As a tool, `uses_check.py` adds to ensure-immutable-actions the immutable *tag* at the same commit as a movable tag (a readable, protected pin that runs the same code) and the form of each movable tag.

---

### 3. Technical baseline, with dates

All quotations are literal and dated in `data/sources/excerpts.json`.

| DATE       | SOURCE                                                                                | WHAT IT SAYS                                                                                                                                                                                                                     |
|------------|---------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2025-08-15 | changelog, <i>GitHub Actions policy now supports blocking and SHA pinning actions</i> | "GitHub recommends that workflows pin dependency versions to a specific commit SHA"                                                                                                                                              |
| 2025-10-28 | changelog, <i>Immutable releases are now generally available</i>                      | "Tag protection : Tags for new immutable releases are protected and can't be deleted or moved." / "Existing releases remain mutable unless you republish them."                                                                  |
| 2025-12-11 | github/roadmap #592 and #1103                                                         | the earlier "Immutable Actions" (OCI packages) closed as <i>not planned</i>                                                                                                                                                      |
| 2026-03-13 | docs, actions how-to (last commit)                                                    | "If you want to be able to update the Git tag of a release later, do not create a release on GitHub."                                                                                                                            |
| 2026-03-26 | blog, <i>What's coming to our GitHub Actions 2026 security roadmap</i>                | "We're introducing a dependencies: section in workflow YAML that locks all direct and transitive dependencies with the commits SHA" / "Public preview 3-6 months General availability 6 months"                                  |
| 2026-07-16 | docs, <i>Managing custom actions</i> (last commit)                                    | "Instead, you can recommend that your users specify a major version when using your action"                                                                                                                                      |
| 2026-07-20 | docs, <i>prevent release changes</i> (last commit)                                    | "Be aware that immutability will only apply to future releases."                                                                                                                                                                 |
| 2026-09-03 | docs, <i>Immutable releases</i> concept (last commit)                                 | "its associated Git tag is locked to a specific commit, cannot be changed, and cannot be deleted while the release exists. If you delete the immutable release, you can delete the tag, but you cannot reuse the same tag name." |
| 2026-09-29 | docs, <i>Secure use reference</i> (last commit)                                       | "Pinning an action to a full-length commit SHA is currently the only way to use an action as an immutable release."                                                                                                              |

**State of the dependency lock on 2026-10-03.** Six months and one week after the roadmap post, the roadmap issues *Locked Dependencies for GitHub Actions Workflows [Public Preview]* (#1253) and *[GA]* (#1268) are open, last updated 2026-09-03, without the "Shipped" label that the shipped immutable-releases items carry. None of the 542 changelog entries published between 2026-03-26 and 2026-10-02 announces it, and the workflow-syntax reference has no `dependencies` key. **It had not shipped on the measurement date.**

---

## 4. Data

- **Population A:** the first 1,000 by stars of the 1,686 public, non-fork, non-archived repositories with at least 20,000 stars pushed on or after 2026-07-05. **Population B:** 500 drawn with seed 20261003 from the 20,553 such repositories with 1,000–5,000 stars. Both frames came from 242 search-API queries in star slices ( `data/frame_queries.csv` ). No repository had to be excluded.
- **Workflows:** 10,819 files in 892 repositories of A and 1,945 files in 399 of B, read at the default branch with the GraphQL API. All were readable.
- **Action repositories:** 1,435 distinct repositories (1,427 resolved) after merging names that redirect to the same repository, with their latest release, their 100 most recent releases, every non-SHA ref used against them, and whether each of the 1,881 distinct pinned SHAs is a commit.

---

## 5. Method

The unit is a remote `uses: reference ( owner/repo[/path]@ref )` in a workflow file, step or reusable-workflow call. The classifier ( `scripts/uses_check.py` , rules in `METHOD.md` §4) labels each one **sha** (40 hexadecimal characters that are a commit of the action repository), **sha-unresolvable** (40 hexadecimal characters that are not: a deleted commit, an annotated-tag object, a vanished repository), **immutable-tag** (a tag whose GitHub release has `immutable: true` ), **mutable-tag** (any other tag; form major `vN` , minor `vN.N` , full `vN.N.N` or other), **branch** or **unresolved**. Both SHA classes count as SHA-pinned. For a mutable tag it also reports the *immutable alternative*: an immutable release tag that points to the same commit. An action's author is an *immutable publisher* when the action repository's latest release is immutable. First party means the current owner is `actions` or `github` .

The extractor is a dependency-free, YAML-aware line scanner. Against a full YAML parse of all 12,764 files, 12,760 of the 12,761 files that PyYAML parses agree. Changes made after the method was frozen are listed with their effect in `METHOD.md` (Deviations); none changed H1.

Reference-level intervals are percentile bootstraps that resample repositories (2,000 replicates), because references cluster within repositories. Repository-level shares carry Wilson intervals in B (a simple random sample) and none in A, which is a census of its frame.

---

## 6. Results

### 6.1 How references are pinned

| VERDICT          | A      | %    | B     | %    |
|------------------|--------|------|-------|------|
| sha              | 32,868 | 56.0 | 4,806 | 45.9 |
| sha-unresolvable | 110    | 0.2  | 6     | 0.1  |
| immutable-tag    | 333    | 0.6  | 87    | 0.8  |
|                  |        |      |       |      |

| VERDICT                        | A             | %    | B             | %    |
|--------------------------------|---------------|------|---------------|------|
| mutable-tag                    | 23,720        | 40.4 | 5,178         | 49.4 |
| · major ( <code>v7</code> )    | 22,543        |      | 4,949         |      |
| · full ( <code>v7.0.1</code> ) | 1,025         |      | 169           |      |
| · minor and other              | 152           |      | 60            |      |
| branch                         | 1,632         | 2.8  | 386           | 3.7  |
| unresolved                     | 27            | 0.0  | 9             | 0.1  |
| <b>remote references</b>       | <b>58,690</b> |      | <b>10,472</b> |      |

SHA-pinned, pooled: 32,978 of 58,690 (56.2%) in A (interval 50.7–61.7%) and 4,812 of 10,472 (46.0%) in B (29.0–60.1%). A few large workflow sets drive the pooled shares: without its three largest repositories (PostHog/posthog, openclaw/openclaw, github/awesome-copilot) A gives 30,016 of 55,728 (53.9%); without open-telemetry/opentelemetry-python, B gives 3,421 of 9,080 (37.7%). Per repository, the median SHA share is 5.0% in A (mean 39.3%) and 0.0% in B (mean 19.1%). 110 SHA pins in A, to 25 action repositories, are not a commit of the action repository. In 90.7% of SHA pins in A the line carries a version comment ( `# v4.2.0` ). 390 references in A and 93 in B call reusable workflows.

## 6.2 What immutable releases protect

Of the tag references to actions whose latest release is immutable, 312 of 6,898 (4.5%) point to a protected tag in A (2.8–7.1%) and 76 of 1,564 (4.9%) in B (2.4–8.2%). Counting every action with an immutable release among its latest 100 gives 333 of 6,926 (4.8%) and 87 of 1,579 (5.5%).

The protected references are concentrated. In A, langflow-ai/langflow holds 68 of the 312 and the top five repositories 132; leaving out any one repository gives 3.6–4.6%. In B, libimobiledevice/ideviceinstaller holds 18 of the 76 and the top five 44; leaving one out gives 3.8–5.1%. The unprotected references sit in 503 of the 516 repositories of A that reference immutable publishers by tag.

The unprotected references are almost all major tags. For 3,238 of 6,586 (49.2%) in A (43.2–56.0%) and 743 of 1,488 (49.9%) in B (42.6–56.8%), an immutable release tag already points to the same commit as the major tag: switching `@v7` to that `@v7.0.0` would not change the code that runs, but would stop automatic updates within the major version. Most unprotected references in A go to `actions/setup-node` (1,729) and `actions/setup-python` (1,080). 37 immutable-tag references in A use a major tag that is itself an immutable release (for example `tj-actions/changed-files@v47` ).

## 6.3 Which actions publish immutable releases

Of the 1,221 resolved action repositories referenced in A, 154 (12.6%) publish their latest release as immutable; 154 of the 1,039 that have any release (14.8%). Third party: 139 of 1,187 (11.7%); first party: 15 of 34 (44.1%). The most-used actions matter most: 28 of 50 (56.0%) of the action repositories used by the most repositories in A are immutable publishers, including 10 of the 19 first-party ones in that list. First

party: 12 of 28 (42.9%) for `actions/*` and 3 of 6 (50.0%) for `github/*`. Immutable include `actions/setup-node`, `setup-python`, `setup-go`, `setup-java`, the `attest*` actions and `github/codeql-action`; not immutable include `actions/checkout` (latest v7.0.1; used by 874 repositories of A), `cache`, `upload-artifact`, `download-artifact`, `github-script` and `deploy-pages` (full lists in `data/tables.md`). The most-used action of all offers no immutable tag to pin to.

## 6.4 Repositories that pin every written reference

Every remote reference written in the workflow files is SHA-pinned in at most 263 of 891 (29.5%) repositories in A. 89 of those 263 also call local actions ( `./path`, outside `.github/workflows/` ) whose own `uses:` we did not read; without them the figure is 174 of 891 (19.5%). In B: at most 65 of 398 (16.3%, Wilson 13.0–20.3%), or 55 of 398 (13.8%, 10.8–17.6%). Counting only third-party actions: 284 of 783 (36.3%) and 62 of 329 (18.8%). At least one SHA pin appears in 490 repositories of A and 100 of B.

## 6.5 Secondary: the repositories' own releases

The population repositories' own latest release is immutable in 71 of 806 (8.8%) in A and 10 of 353 (2.8%) in B. An SBOM-like asset is attached to it in 23 of 806 (2.9%) and 4 of 353 (1.1%). This only describes release assets; the CRA's SBOM duty (Annex I, Part II, from 11 December 2027) does not require publishing the SBOM.

---

## 7. Discussion

**The mechanism protects what it was designed to protect.** An immutable release locks a full version tag; the documentation keeps the major tag movable so that `@v7` receives fixes. References to actions that publish immutable releases go to the movable tag 95% of the time.

**Half of the gap is free in code terms.** For about half the unprotected references, a protected tag names the commit that runs today; moving to it trades automatic updates within the major line for a tag that cannot move, which Dependabot can then update.

**SHA pinning carries most of the protection, in few repositories.** Over half of the references in A are SHA-pinned, yet the median repository pins 5%. GitHub's secure-use reference (2026-09-29) still calls SHA pinning "the only way to use an action as an immutable release", which is no longer accurate for the tag of an immutable release. Neither form protects an action's own dependencies, which the announced lock targets; `actions/checkout`, `cache` and `upload-artifact` offer no immutable tag at all.

---

## 8. Limitations

- **One day.** Releases, tags and workflows change daily; every figure is "on 2026-10-03".
- **Populations.** A is the top of GitHub by stars with recent activity; B is a random sample of 500 (398 with remote references) from one star band. The frame and the B sample were drawn 9 minutes before the method was frozen (the rules do not depend on the sample). The frames are not published, because they list individuals' repositories.
- **Scope.** Default branch only; local composite actions and the dependencies of actions are not read, so the "pin every written reference" figures are upper bounds.

- **Alternatives** are searched among the 100 most recent releases and the release of the ref.
- **SHA pins.** A SHA counts as a commit if GitHub resolves it in the action repository, which includes its fork network, so "impostor" commits from forks are not detected.
- **Organisation policies** (required SHA pinning, required immutable releases) are not observable.
- **Extractor.** One file still disagrees with the YAML parse (two values written as folded block scalars); 3 files are not parsable by PyYAML (a tab inside a plain scalar), and their 7 references are counted by the extractor. 19 refs name both a tag and a branch and were classified as tags.
- **Residual re-identification risk.** Per-repository counts for user-owned repositories and per-action counts for user-owned actions could in principle fingerprint a public repository by re-crawling GitHub; that information is already public in each repository, and no published field links a pseudonym to a name, rank, star count, file, line or SHA.

---

## 9. Data, code and licences

Code (Apache-2.0) and data and text (CC BY 4.0) are listed in `README.md`. `scripts/run.sh` recomputes every figure offline; `scripts/check_headline.py` checks the numbers of the README abstract and of this paper's abstract and sections 4, 6 and 7 against `data/` (`VERIFICATION.md` says exactly how). Personal owners are pseudonymised with a key that is not published; raw API responses are not published. Data came only from the GitHub REST and GraphQL APIs; no github.com page was fetched.

---

## Automation and review

AI agents carried out this study: collection, classifier and tests, analysis, reading of the prior work and documentation, and all of this text. Every figure is computed by the scripts from the published tables, and every quotation is checked verbatim against the saved source. An independent AI review agent recomputed the figures, re-checked 33 references and 54 latest releases against the API, and raised the issues fixed in this version (prior work, pseudonymisation, concentration, bounds).

---

## Competing interests

EasyByte, the cooperative behind EasyxLab, develops attest-lint, an open-source tool that flags attestation regressions in lockfiles (study S5). EasyByte may release `scripts/uses_check.py` as a free tool; a comparable free tool, ensure-immutable-actions, already exists (§2). No figure here was produced by attest-lint. No other interest.

---

## References

- GitHub changelog: *GitHub Actions policy now supports blocking and SHA pinning actions* (2025-08-15); *Releases now support immutability in public preview* (2025-08-26); *Immutable releases are now generally available* (2025-10-28); *Actions steps can now be run in parallel* (2026-06-25); *Reference same-repository actions with self-repository syntax* (2026-07-30). <https://github.blog/changelog/>
- GitHub blog: *What's coming to our GitHub Actions 2026 security roadmap* (2026-03-26).

- GitHub Docs (github/docs): *Using immutable releases and tags to manage your action's releases; Immutable releases; Preventing changes to your releases; Managing custom actions; Secure use reference*. Paths and commits in `data/sources/excerpts.json`.
- github/roadmap issues #592, #1103, #1137, #1138, #1253, #1268.
- datosh/pinned-actions, <https://pin-gh-actions.kammel.dev/> (data of April 2026; `analysis_immutable.go`, 2026-04-27).
- joshjohanning/ensure-immutable-actions (v2.7.2, 2026-09-22).
- zizmor, `docs/audits.md` (2026-10-02); pinact, `README.md` (2026-09-12).
- I. Koishybayev et al. Characterizing the Security of GitHub CI Workflows. USENIX Security 2022.
- A. Decan et al. On the Use of GitHub Actions in Software Development Repositories. ICSME 2022. doi:10.1109/ICSME55016.2022.00029
- Huang, Lin. Revisiting Security Practices for GitHub Actions Workflows. ICPC 2025. doi:10.1109/ICPC66645.2025.00016
- Kubo, Kanei, Akiyama, Wakai, Mori. Action Required: A Mixed-Methods Study of Security Practices in GitHub Actions. NDSS 2026. doi:10.14722/ndss.2026.240483
- N. Chaiwut, N. Nikiforakis. Time for Actions: A Longitudinal Study of the GitHub Actions Marketplace. SecDev 2025. doi:10.1109/SecDev66745.2025.00023
- Boschanski, Vieira. Evaluating Security Best Practices in the GitHub Actions Documentation. DSN-S 1. doi:10.1109/DSN-S70715.2026.00079

## Cite this study

### CITATION

EasyxLab (2026). What immutable releases protect in GitHub Actions workflows. Study S18. EasyByte Hub S. Coop. Mad.  
<https://github.com/easybytehub/easyxlab/tree/main/studies/s18-actions-immutable-releases>

### BIBTEX

```
@techreport{easyxlab_s18,
  title      = {What immutable releases protect in GitHub Actions workflows},
  author     = {{EasyxLab}},
  institution = {EasyByte Hub S. Coop. Mad.},
  number     = {S18},
  year       = {2026},
  url        = {https://github.com/easybytehub/easyxlab/tree/main/studies/s18-actions-immutable-releases}
}
```