NDSS Symposium 2026

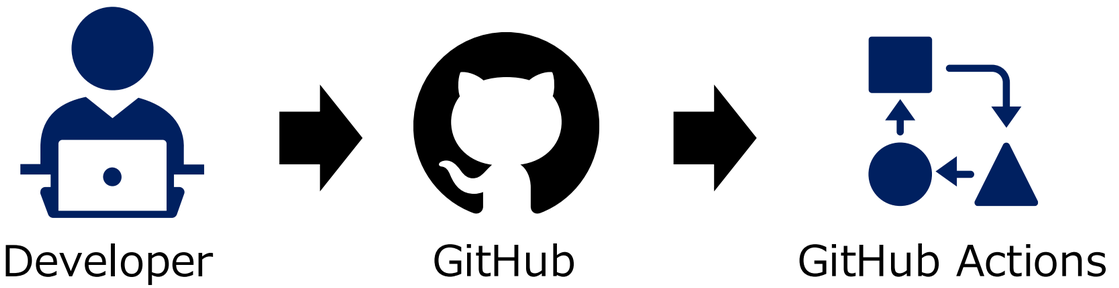
Action Required: A Mixed-Methods Study of Security Practices in GitHub Actions

Yusuke Kubo[*‡], Fumihiro Kanei[*], Mitsuaki Akiyama[†],
Takuro Wakai[‡], Tatsuya Mori[‡, §, ¶]

[*] NTT DOCOMO BUSINESS, Inc. [‡] Waseda University
[†] NTT, Inc. [§] NICT [¶] RIKEN AIP

GitHub Actions

- Continuous Integration / Continuous Delivery (CI/CD) platform provided by GitHub
- Automates build, test, and deployment pipelines using workflows
- Mainstream CI/CD platform driven by seamless GitHub integration and ease of setup



Major Components

.github/example-workflow.yml

```yaml
name: Example Workflow #Workflow Name
on:
  pull_request: #Trigger Event
  branches:
    - main #Target Branch
jobs:
  build: #Job Name
    runs-on: ubuntu-latest #Running Environment
    steps:
      - name: Check out code
        uses: actions/checkout@v4 #Using action

      - name: Show Pull Request Title
        env:                                    #Access github context
          PR_TITLE: ${{ github.event.pull_request.title }}
        run: echo "$PR_TITLE"
```

Security Incidents involving GitHub Actions

CVE-2025-30066 (Mar 2025)

Summary

A supply chain attack compromised the **tj-actions/changed-files** GitHub Action, impacting over 23,000 repositories. Attackers retroactively modified multiple version tags to reference a malicious commit, exposing CI/CD secrets in workflow logs. The vulnerability existed between **March 14 and March 15, 2025**, and has since been mitigated. This poses a significant risk of unauthorized access to sensitive information.

This has been patched in v46.0.1.

Details

The attack involved modifying the **tj-actions/changed-files** GitHub Action to execute a malicious Python script. This script extracted secrets from the Runner Worker process memory and printed them in GitHub Actions logs, making them publicly accessible in repositories with public workflow logs.

Key Indicators of Compromise (IoC):

- **Malicious commit**: 0e58ed8671d6b60d0890c21b07f8835ace038e67
- **Retroactively updated tags pointing to the malicious commit**:
  - `v1.0.0`: 0e58ed8671d6b60d0890c21b07f8835ace038e67
  - `v35.7.7-sec`: 0e58ed8671d6b60d0890c21b07f8835ace038e67
  - `v44.5.1`: 0e58ed8671d6b60d0890c21b07f8835ace038e67

https://github.com/advisories/GHSA-mrrh-fwg8-r2c3

S1ngularity (Aug 2025)

Attack Vector

Vulnerable Workflow

The root cause the introduction of a vulnerable workflow which contained the possibility for injecting executable code. The vulnerable workflow was reverted in `master` almost immediately after the team learned it could have been malicious. However, this proved to be inadequate to address the vulnerability.

The workflow contained the 2 issues.

Bash Injection

```
- name: Validate PR title
  run: |
    echo "Validating PR title: ${{ github.event.pull_request.title }}"
```

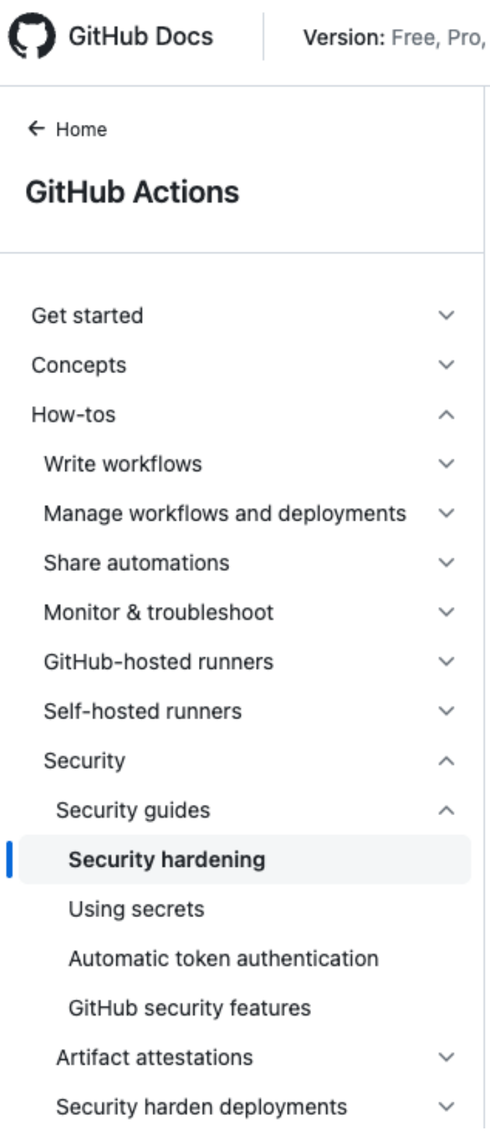
The intention of these lines was to print out the pull request titles being validated via our commit format checks.

However, if a PR was opened with a title such as `$(echo "You've been compromised")` the code would be executed within the workflow. We understood this once it was reported but we did not fully understand how this would compromise any secrets because the PR title validation workflow itself did not have access to any secrets.

https://github.com/nrwl/nx/security/advisories/GHSA-cxm3-wv7p-598c

Security Practices for GitHub Actions

Official document published by GitHub



GitHub Actions / How-tos / Security / Security guides /

Security hardening for GitHub Actions

Good security practices for using GitHub Actions features.

Overview

This guide explains how to configure security hardening for certain GitHub Actions features. If the GitHub Actions concepts are unfamiliar, see Understanding GitHub Actions.

Using secrets

Sensitive values should never be stored as plaintext in workflow files, but rather as secrets. Secrets can be configured at the organization, repository, or environment level, and allow you to store sensitive information in GitHub.

Secrets use Libsodium sealed boxes, so that they are encrypted before reaching GitHub. This occurs when the secret is submitted using the UI or through the REST API. This client-side encryption helps minimize the risks related to accidental logging (for example, exception logs and request logs, among others) within GitHub's infrastructure. Once the secret is uploaded, GitHub is then able to decrypt it so that it can be injected into the workflow runtime.

To help prevent accidental disclosure, GitHub uses a mechanism that attempts to redact any secrets that appear in run logs. This redaction looks for exact matches of any configured secrets

Version used in our study
(before the update on July 11, 2025)

In this article

- Overview
- Using secrets

16 Practices

- Using CODEOWNERS to monitor changes
- Understanding the risk of script injections
- Good practices for mitigating script injection attacks
- Using OpenID Connect to access cloud resources
- Using third-party actions
- Reusing third-party workflows
- Using Dependabot version updates to keep actions up to date
- Preventing GitHub Actions from creating or approving pull requests
- Using code scanning to secure workflows
- Using OpenSSF Scorecards to secure workflow dependencies
- Potential impact of a compromised runner
- Considering cross-repository access
- Hardening for GitHub-hosted runners
- Hardening for self-hosted runners
- Auditing GitHub Actions events

https://web.archive.org/web/20250701141038/https://docs.github.com/en/actions/how-tos/security-for-github-actions/security-guides/security-hardening-for-github-actions

Research Gap

	Summary	Target Service	Topic
[5]	An empirical study characterizing security risks in GitHub workflows	**GitHub Actions** and other CI/CD Services	**Security Risk**
[6]	A systematic study on token security in continuous integration services	**GitHub Actions** and other CI/CD Services	**Security Risk**
[22]	A Mixed-methods (surveys + interviews) study of OSS maintainers challenges	**GitHub**	**Security Practice**

Research Gap

NDSS'26	**[Our Research] A Mixed-methods study of security practices in GitHub Actions**	**GitHub Actions**	**Security Practice**

[5] I. Koishybayev et al., "**Characterizing the security of github CI workflows**," in Proc. USENIX Security 2022.
[6] Y. Gu et al., "**Continuous Intrusion: Characterizing the Security of Continuous Integration Services**," in Proc. IEEE S&P 2023.
[22] J.Ayala et al., "**A mixed-methods study of open-source software maintainers on vulnerability management and platform security features**," in Proc. USENIX Security 2025.

Research Questions

RQ1.

To what extent are GitHub Actions security practices implemented in real-world repositories?

RQ2.

What repository characteristics are associated with the implementation or non-implementation of security practices?

RQ3.

What factors prevent developers from implementing security practices?

Mixed-Methods Study

Security Practices for GitHub Actions

Measurement Study

Public Repositories
using GitHub Actions

Repository
Analysis

User Study

Developers
using GitHub Actions

Questionnaire

RQ1

RQ2

RQ3

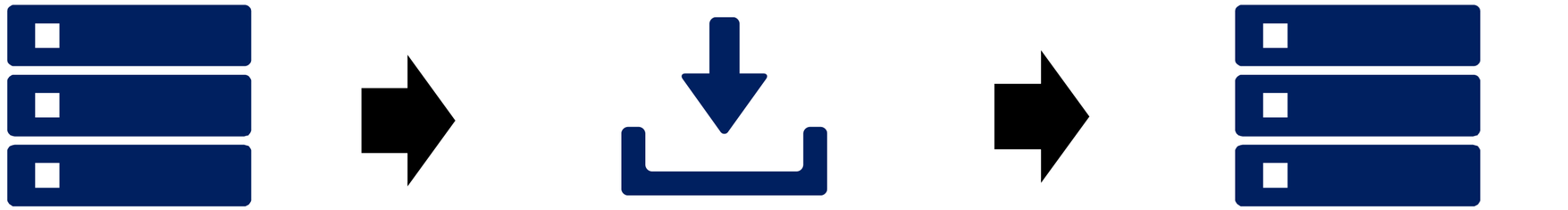
Targeted Security Practices

ID	Practice	Type	Objective
P1	CODEOWNERS	Feature	Ensure mandatory review of workflow changes to protect against unauthorized or accidental modifications
P2	Mitigating Script Injection	Workflow Coding	Secure the use of github context in workflows to avoid script injection
P3	OpenSSF Scorecard	Tool	Evaluate repository security posture through automated risk assessment
P4	Pinning Third-party Actions	Workflow Coding	Ensure the integrity of third-party actions by secure referencing • Pinning to SHA (P4-1) • Pinning to TAG when the creator is trusted (P4-2)
P5	Dependabot	Tool	Maintain up-to-date and secure GitHub Actions dependencies through automated updates

In user study, P4 is divided into P4-1 and P4-2

Measurement Study: Repository Dataset

- SEART-GHS as the base dataset
  - → GitHub repositories dataset including metadata such as stars and contributors
- GitHub Actions usage identified via GitHub REST API
- Repositories using GitHub Actions cloned locally for dataset construction



SEART-GHS

1,675,884
(created before 2024-12-31)

GitHub REST API

GitHub Actions
Repository Dataset

338,812

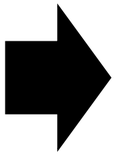
Measurement Study: Repository Analysis

Practice Status（RQ1）

Detect applicability and implementation status for each security practice

- Workflow Files
- Configuration Files

Repository Code



Detection Framework

Applicability

Implementation

Statistical Modeling （RQ2）

Identify repository characteristics associated with practice implementation

Dependent Variables

Implementation
(True or False)

Independent Variables

e.g.,
- Number of Stars
- Number of Contributors
- Owner Type

- Odds Ratio
- p-value
- 95% CI

Logistic Regression Analysis

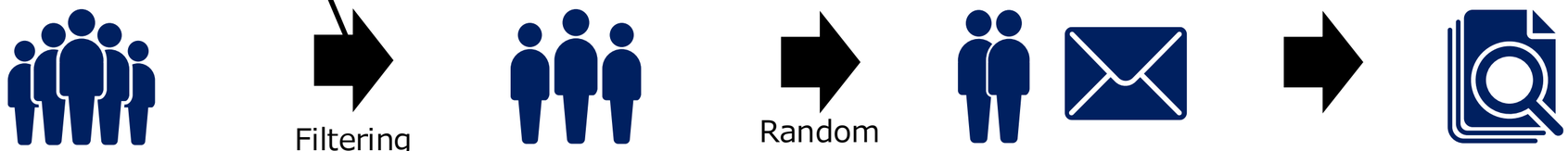
User Study: Recruiting Participants

- Targeted developers with experience using GitHub Actions
- No monetary incentives were provided; results were shared upon request
- Recruited via publicly listed email addresses in GitHub account profiles

Criteria

✓Public email address in profile
✓Contributed to GitHub Actions workflows
✓Human (non-bot) account

Our targeted recruitment strategy helped to reduce potential **survey fatigue**



Top-100 Contributors per Repository	Highly Relevant Candidates	Survey Participants	Valid Responses
1,439,072	3,405	1,800	102

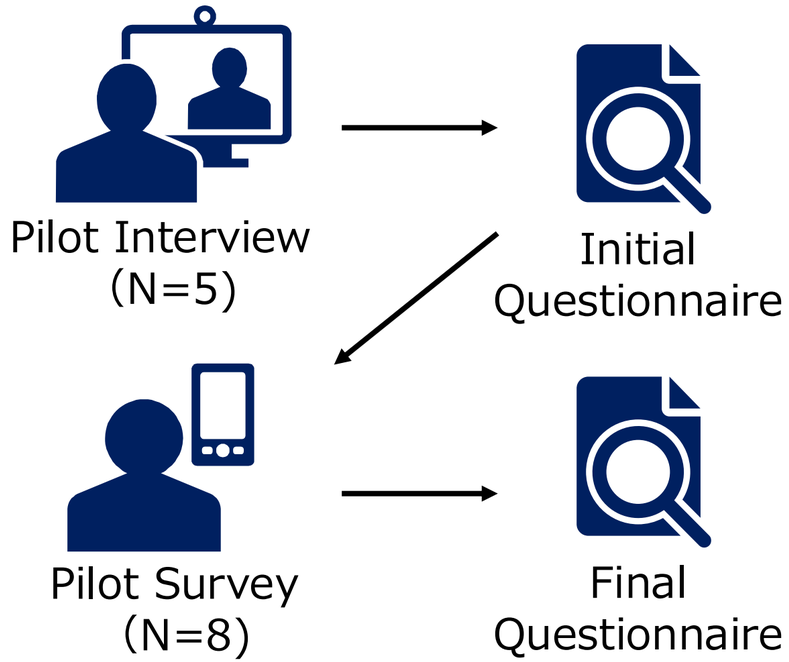
User Study: Online Survey

- Questionnaire designed through two pilot studies
- Multiple-choice questions with an “Other” free-text option

Descriptive Statistics

Qualitative Coding

Designing Process



Questionnaire Overview

Number of Questions: MAX 56

Estimated Time: 20 minutes

Sections:
- About GitHub Repository
- About GitHub Actions
- About Security Practices (for each)
- Demographics and Professional Background

(Full questionnaire in our research artifact)

RQ1: Implementation Rates

Security Practice	Target Repo.	Implementation Rate
P1 CODEOWNERS	338,812	7.1% (23,890)
P2 Mitigating Script Injection	43,516	52.9% (23,005)
P3 OpenSSF Scorecard	338,812	0.6% (1,965)
P4 Pinning Third-party Actions	233,124	16.2% (37,693)
P5 Dependabot	332,928	10.7% (35,588)

Answer to RQ1

None of the security practices are widely implemented
(0.6% – 52.9% implementation rates)

RQ2: Factors Associated with Implementation

Independent Variables	P1	P2	P3	P4	P5
Number of Stars	.99998*	1.00000	1.00000	1.00001	.99999
Number of Contributors	1.00167*	.99895*	1.00232*	.99956*	.99825*
Number of Commits	1.00000	1.00000	1.00000	1.00000	.99999*
Codebase Size	1.00000	1.00000	1.00000	1.00000	1.00000
Repository Age	.99989*	1.00007*	1.00002	.99999	1.00011*
Recent Activity	1.41553*	.98246	2.69133*	1.07503*	2.33900*
Number of Workflow Files	1.05185*	.97287*	1.03052*	.97246*	1.05856*
Number of Workflow Developers	1.10088*	.98085*	1.04417*	1.01408*	1.12425*
Owner Type (User)	.24995*	1.05489*	.35556*	1.06167*	.89690*

The listed values indicate the odds ratio.
* indicates statistical significance ($p < 0.05$ and 95% CI does not include 1).

Answer to RQ2

Small Effect Sizes for Repo Scale:
Popularity, maturity, and available human resources are sometimes significant,
but the magnitude is small (odds ratios close to 1)

Active Maintenance:
Ongoing maintenance is associated with higher implementation (P1, P3, P4, and P5)

Workflow Complexity:
A higher number of workflows hinders consistent implementation of coding-based practices (P2, P4)

Governance Effect:
Organizational ownership correlates with higher implementation of features (P1) and tools (P3, P5)

RQ3: Reasons for Not Implementing

SPs	Reasons for Not Implementing (Top 3 and N>=2)		
P1	**Lack of awareness**	**41.4%**	**(29/70)**
	Unnecessary/overly strict	40.0%	(28/70)
	Maintenance/operational costs	4.3%	(3/70)
P2	**Lack of awareness**	**50.0%**	**(19/38)**
	Unnecessary/overly strict	31.6%	(12/38)
	Maintenance/operational costs	5.3%	(2/38)
	Unclear risks or benefits	5.3%	(2/38)
P3	**Lack of awareness**	**71.6%**	**(63/88)**
	Unnecessary/overly strict	20.5%	(18/88)
P4-1	Unnecessary/overly strict	30.7%	(23/75)
	Maintenance/operational costs	25.3%	(19/75)
	Lack of awareness	**21.3%**	**(16/75)**
P4-2	**Lack of awareness**	**38.7%**	**(12/31)**
	Unnecessary/overly strict	22.6%	(7/31)
	Maintenance/operational costs	12.9%	(4/31)
P5	**Lack of awareness**	**36.2%**	**(17/47)**
	Unnecessary/overly strict	25.5%	(12/47)
	Maintenance/operational costs	8.5%	(4/47)

RQ3: Reasons + Important Factors

SPs	Reasons for Not Implementing (Top 3 and N>=2)		
P1	Lack of awareness	41.4%	(29/70)
	Unnecessary/overly strict	40.0%	(28/70)
	Maintenance/operational costs	**4.3%**	**(3/70)**
P2	Lack of awareness	50.0%	(19/38)
	Unnecessary/overly strict	31.6%	(12/38)
	Maintenance/operational costs	**5.3%**	**(2/38)**
	Unclear risks or benefits	5.3%	(2/38)
P3	Lack of awareness	71.6%	(63/88)
	Unnecessary/overly strict	20.5%	(18/88)
P4-1	Unnecessary/overly strict	30.7%	(23/75)
	Maintenance/operational costs	**25.3%**	**(19/75)**
	Lack of awareness	21.3%	(16/75)
P4-2	Lack of awareness	38.7%	(12/31)
	Unnecessary/overly strict	22.6%	(7/31)
	Maintenance/operational costs	**12.9%**	**(4/31)**
P5	Lack of awareness	36.2%	(17/47)
	Unnecessary/overly strict	25.5%	(12/47)
	Maintenance/operational costs	**8.5%**	**(4/47)**

Important Factors (Top 5)	
Low maintenance and operational overhead	**79.4% (81/102)**
Easy to set up and quick to adopt	46.1% (47/102)
Clearly effective in mitigating relevant risks	37.3% (38/102)
Does not interfere with the existing development process	32.4% (33/102)
Well-documented setup procedures and implementation details	25.5% (26/102)

RQ3: Qualitative Coding

Code	Example of Non-Implementation Reasons (open-ended response)	
Not applicable	[P1]	*As single owner, this step is unnecessary. [...]*
	[P2]	*I don't use external variables nor any substantial non-linear code in my actions.*
	[P4-1]	*I use only GitHub actions avoiding 3rd-party actions.*
	[P5]	***Dependabot doesn't support C++***
Negative impact on development	[P1]	*Some of my collaborators are not as GitHub savvy and might get upset if something like this inhibited their ability to contribute.*
	[P5]	*We don't want to increase deps without deeper discussions with community or the team so this is sure a manual process.*
Negative impact on security	**[P4-1]**	***This could make us more vulnerable by not automatically using the latest version of an action that might have important security fixes.***
	[P5]	*Rubber stamp updating the dependencies violates the point of pinning to a commit.*
Lack of resource	[P4-1]	*I'm willing to do this, but I have no time at the moment.*

Dependabot (P5) manages action dependencies independent of programming language (in GitHub Actions)

Pinning to SHA (P4-1) does not prevent automatic updates via dependency management tools

RQ3: Summary of Prevention Factors

Answer to RQ3

Three Key Barriers:

1. Lack of awareness

- 21.3% - 71.6% of non-implementers were unaware of the practices

2. Maintenance and operation concerns

- 25.3% cited maintenance burden for Pinning to SHA (P4-1)
- 79.4% emphasized low operational overhead as important

3. Misconceptions

- Some misunderstood the applicability of Dependabot (P5)
- Some perceived Pinning to SHA (P4-1) as reducing security

Mixed-Methods Study

Answer to RQs

Action Required

Recommendations

Developer Intervention	Notification	• Detect unimplemented practices and notify developers • Design notifications to minimize notification fatigue (e.g., context-aware, well-timed, and applicable repositories only)

	Platform-Level Support (P1, P3, P5)	• Match the automation level to each practice
		  • P1: support context-dependent configuration
		  (e.g., reviewer suggestions/file generation)
		  • P3 & P5: leverage templates and enable one-click activation
		• Prefer partial, human-approved automation over full enforcement

	IDE-Level Support (P2, P4)	• Shift left: provide real-time IDE assistance for secure workflow authoring
		  • P2: suggest safe patterns for handling github context values
		  • P4: suggest secure pinning refs (e.g., SHA/tag candidates)

Documentation Improvement	Mapping Practices to Risks	• Link each practice to the risks it mitigates and explain the expected benefits • Use concrete security incidents as examples where applicable
	Clearly “Who should use”	• Add a “Who should use this?” section per practice • State applicability criteria (repository type/workflow usage/prerequisites)

Disclosure and Feedback

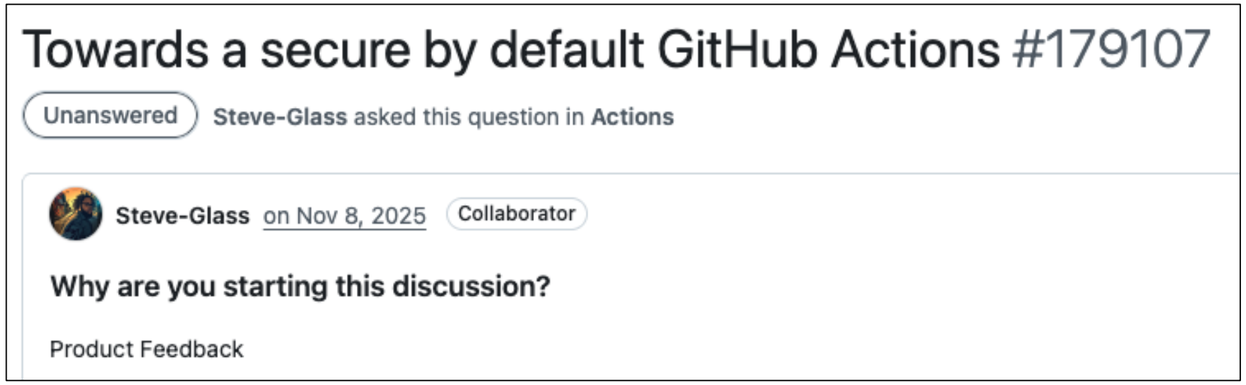
Disclosure

- Findings and recommendations shared with GitHub (Oct 2025)
- Aimed to contribute to platform-level, ecosystem-wide security improvements

Feedback

- Aligned with GitHub’s shift toward stronger default security protections
- Expressed appreciation for research that raises awareness of GitHub Actions security
- Indicated that our insights may help inform future platform improvements

Community discussion initiated by GitHub (Nov 2025)



https://github.com/orgs/community/discussions/179107

Conclusion

- Designed and conducted a mixed-methods study to answer the research questions on GitHub Actions security practices
- Revealed critically low implementation rates through a large-scale analysis of 338.8K public repositories
- Identified three key barriers through a survey of 102 developers: lack of awareness, maintenance concerns, and misconceptions
- Proposed empirically grounded design recommendations and shared them with GitHub

ykubo@nsl.cs.waseda.ac.jp

https://github.com/yksec14/gha_security_research

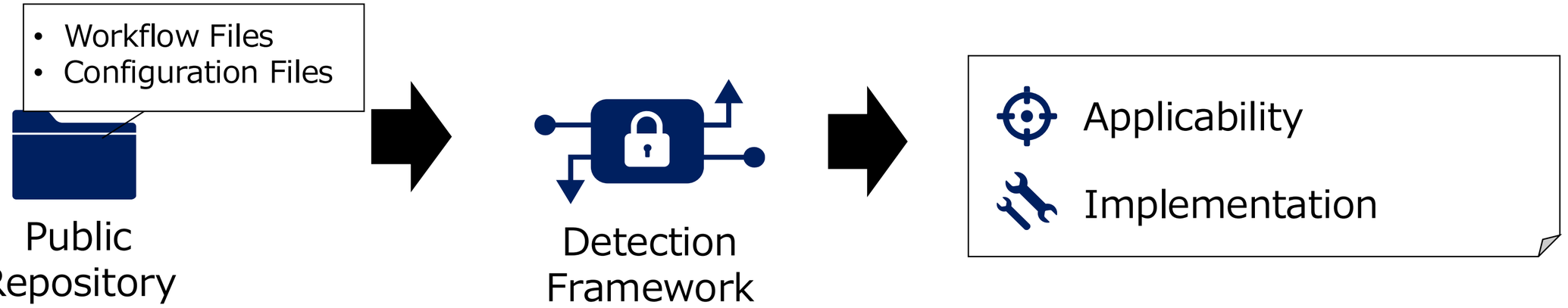
Appendix

Criteria-Based Practice Selection

No.	Select	Practice Type	Detectability	Eligibility	Privilege
1		Setting	✓	×	Admin
2	✓	Feature	✓	✓	User
3		Knowledge			
4	✓	Coding	✓	✓	User
5		Setting	✓	×	User
6	✓	Coding	✓	✓	User
7		Coding	✓	✓	User
8	✓	Tool	✓	✓	User
9		Setting	×		Admin
10		Tool	✓	✓	Admin
11	✓	Tool	✓	✓	User
12		Knowledge			
13		Setting	×		Admin
14		Knowledge			
15		Setting	×		Admin
16		Feature	×		Admin

↻ Merged with No.6

Measurement Study: Detection Framework



ID	Practice	Applicability		Implementation	
		Condition	Source	Condition	Source
P1	CODEOWNERS	(All repositories)	-	CODEOWNERS file present in the correct location	CODEOWNERS File
P2	Mitigating Script Injection	Use of github context	Workflow Files	No unsafe use of github context in workflows	Workflow Files
P3	Scorecard	(Public repositories)	-	Use of *ossf/scorecard* action	Workflow Files
P4	Pinning	Use of third-party actions	Workflow Files	All third-party actions securely pinned	Workflow Files
P5	Dependabot	Use of public actions	Workflow Files	Dependabot config file present in the correct location	dependabot.yml (or .yaml)

Detection Framework: P5 Dependabot

Applicability

.github/example-workflow3.yml
.github/example-workflow2.yml
.github/example-workflow1.yml

```
name: Example Workflow

on:
  push:

jobs:
  example:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout repository
        uses: actions/checkout@v4

      - name: Print message
        run: echo "Hello from GitHub Actions!"
```

[Check]
Use of public actions in workflows

Implementation

[Check]
Correct file path (.github directory)

[Check]
Dependabot configuration file present

.github/dependabot.yml

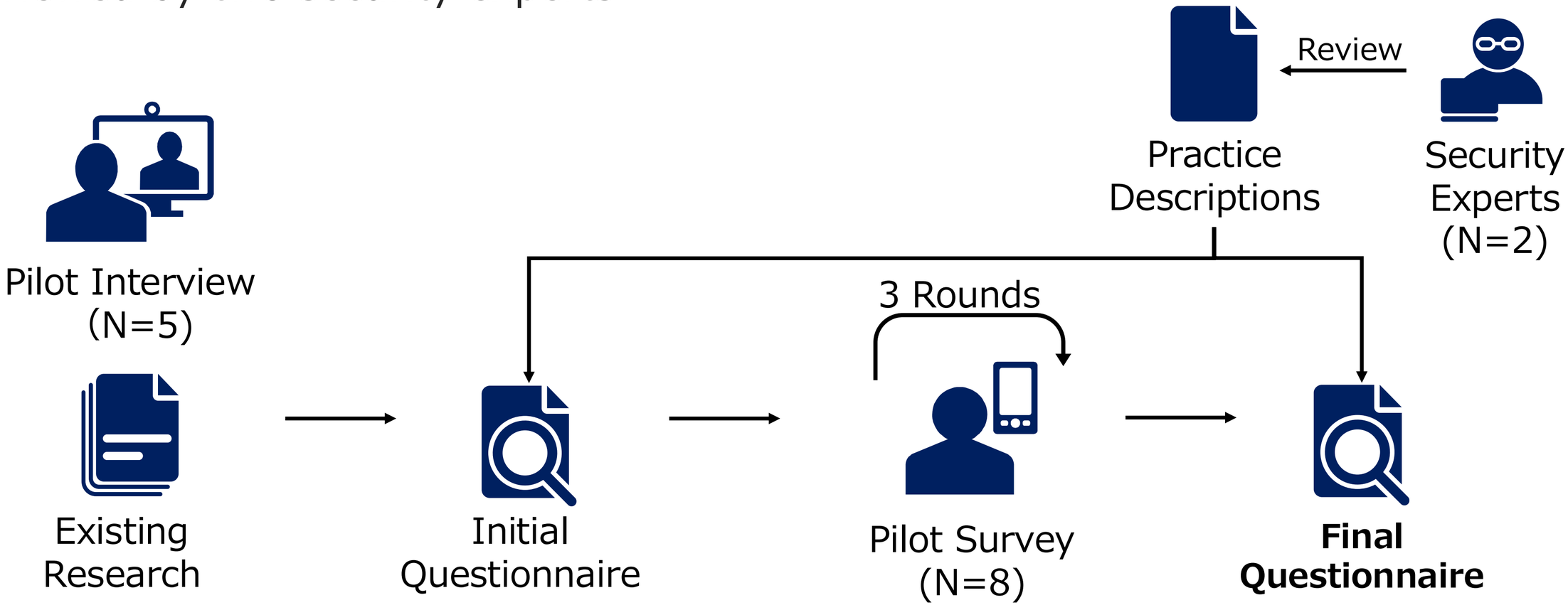
```
version: 2

updates:
  - package-ecosystem: "github-actions"
    directory: "/"
    schedule:
      interval: "weekly"

  - package-ecosystem: "pip"
    directory: "/"
    schedule:
      interval: "weekly"
```

[Check]
package-ecosystem set to "github-actions"

[Check]
Valid YAML format

User Study: Designing and Piloting

- Pilot interviews and prior research informed the questionnaire design
- Three rounds of pilot surveys ensured clarity and consistency
- Security practice descriptions were summarized by the authors and reviewed by two security experts



User Study: Questionnaire

- Up to 56 optional questions; Estimated completion time: 20 minutes
- Multiple-choice format with optional open-ended responses

Section	Question Content
GitHub Repository	• Primary programming language • Number of stars, contributors, and commits • Repository ownership and participant role
GitHub Actions	• Usage patterns (e.g., build, test, deployment) • Workflow configuration information sources • Operational reliance on GitHub Actions
Security Practices (for each practice)	• Implementation status • Reasons for non-implementation • Use of alternative approaches • Willingness to implement (if previously unaware) • Important factors for implementing decisions
Demographics and Professional Background	• Gender, age, residence, preferred language, etc. • Years of experience (software development and GitHub Actions) • Secure Software Development Self-Efficacy Scale (SSD-SES)

RQ2: Logistic Regression Results

Independent Variables	P1	P2	P3	P4	P5
Number of Stars	.99998*** [0.99997, 0.99998]	1.00000+ [0.99999, 1.00000]	1.00000+ [1.00000, 1.00001]	1.00001*** [1.00000, 1.00001]	.99999*** [0.99999, 1.00000]
Number of Contributors	**1.00167*** [1.00135, 1.00198]**	**.99895*** [0.99856, 0.99934]**	**1.00232*** [1.00164, 1.00300]**	.99956* [0.99921, 0.99991]	**.99825*** [0.99792, 0.99858]**
Number of Commits	1.00000*** [0.99999, 1.00000]	1.00000+ [1.00000, 1.00000]	1.00000** [1.00000, 1.00000]	1.00000+ [1.00000, 1.00000]	.99999*** [0.99999, 0.99999]
Codebase Size	1.00000* [1.00000, 1.00000]	1.00000** [1.00000, 1.00000]	1.00000** [1.00000, 1.00000]	1.00000*** [1.00000, 1.00000]	1.00000*** [1.00000, 1.00000]
Repository Age	.99989*** [0.99987, 0.99990]	1.00007*** [1.00006, 1.00009]	1.00002+ [0.99999, 1.00006]	.99999** [0.99998, 1.00000]	1.00011*** [1.00010, 1.00012]
Recent Activity	**1.41553*** [1.37468, 1.45758]**	.98246+ [0.94361, 1.02291]	**2.69133*** [2.41585, 2.99821]**	**1.07503*** [1.05091, 1.09971]**	**2.33900*** [2.28188, 2.39755]**
Number of Workflow Files	**1.05185*** [1.04799, 1.05572]**	**.97287*** [0.96831, 0.97745]**	**1.03052*** [1.02526, 1.03581]**	**.97246*** [0.96803, 0.97691]**	**1.05856*** [1.05475, 1.06238]**
Number of Workflow Developers	**1.10088*** [1.09676, 1.10502]**	**.98085*** [0.97661, 0.98511]**	**1.04417*** [1.03814, 1.05024]**	**1.01408*** [1.01041, 1.01776]**	**1.12425*** [1.12001, 1.12849]**
Owner Type	**.24995*** [0.24078, 0.25947]**	**1.05489* [1.00966, 1.10216]**	**.35556*** [0.31570, 0.40044]**	**1.06167*** [1.03721, 1.08672]**	**.89690*** [0.87462, 0.91974]**

The listed values indicate the odds ratio. Brackets show the 95% confidence interval. Significance levels are + $p > .05$; * $p < .05$; ** $p < .01$; *** $p < .001$. Values are shown in bold when the odds ratio is statistically significant and differs from 1.000 by at least 0.001. For binary variables such as Recent Activity, the odds ratio compares cases with value 1 against those with value 0. For categorical variables such as Owner Type, User is treated as the reference category, and the odds ratio for Organization is interpreted relative to it. For all other (continuous) variables, the odds ratio represents the change in odds associated with a one-unit increase in the variable.

RQ3: Implementation Rate and Reasons

SPs	Implementation Rate (User Study)	Reasons for Not Implementing (Top 3 and N>=2)	
P1	17.6% (18/102)	Lack of awareness	41.4% (29/70)
		Unnecessary/overly strict	40.0% (28/70)
		Maintenance/operational costs	4.3% (3/70)
P2	46.1% (47/102)	Lack of awareness	50.0% (19/38)
		Unnecessary/overly strict	31.6% (12/38)
		Maintenance/operational costs	5.3% (2/38)
		Unclear risks or benefits	5.3% (2/38)
P3	5.9% (6/101)	Lack of awareness	71.6% (63/88)
		Unnecessary/overly strict	20.5% (18/88)
P4-1	19.4% (19/98)	Unnecessary/overly strict	30.7% (23/75)
		Maintenance/operational costs	25.3% (19/75)
		Lack of awareness	21.3% (16/75)
P4-2	61.2% (60/98)	Lack of awareness	38.7% (12/31)
		Unnecessary/overly strict	22.6% (7/31)
		Maintenance/operational costs	12.9% (4/31)
P5	43.4% (43/99)	Lack of awareness	36.2% (17/47)
		Unnecessary/overly strict	25.5% (12/47)
		Maintenance/operational costs	8.5% (4/47)

Limitations

Generalizability & Applicability

- 5 of 16 security practices analyzed, selected based on observability and feasibility
- Findings may not generalize to excluded practices due to differing purposes and implementations

Sample Size of Participants

- User study included 102 experienced developers, comparable to prior GitHub-focused research
- Limited statistical power for inferential hypothesis testing
- Therefore, analysis relied on descriptive statistics and qualitative coding

Developers' Roles and Responsibilities

- The survey did not capture fine-grained roles or security decision authority
- More detailed role analysis may provide deeper insight into implementation barriers

Ethical Considerations

General

- Conducted in accordance with the Menlo Report and institutional ethics policies
- Confirmed to be exempt from IRB review requirements

Measurement Study

- API rate limits were carefully monitored and controlled to minimize server load
- Non-intrusive testing (local cloning, no repository modifications)

User Study

- Targeted recruitment to reduce survey fatigue
- Informed consent was obtained prior to participation
- No personally identifiable information collected
- Responses anonymized and securely stored
- Survey results shared upon request, instead of monetary incentives