Characterizing the Security of GitHub CI Workflows

USENIX SEC'22 | Scooped our Submission | Prepublication ➦

Igibek Koishybayev [1], Aleksandr Nahapetyan [1], Raima Zachariah [3], Siddharth Muralee [2], Bradley Reaves [1], Alexandros Kapravelos [1], Aravind Machiry [2]

July 29, 2022

[1]North Carolina State University

[2]Purdue University

[3]Independent Researcher

Table of contents

Before the Paper

- RQ **驱动类型的** Paper（Trending in Security）
- Phenomenal Topic（e.g. Asleep at the Keyboard? Assessing the Security of GitHub Copilot’s Code Contributions ➦）

研究背景与动机

背景

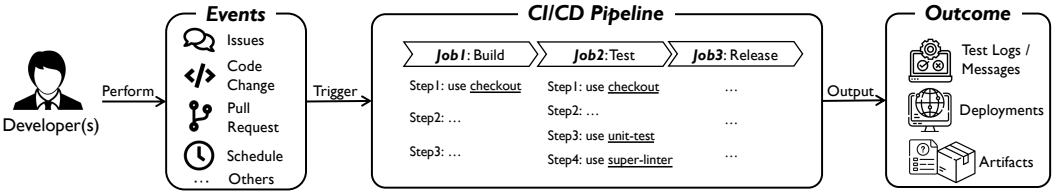

图 1: An overview of CI/CD pipelines.

安全威胁

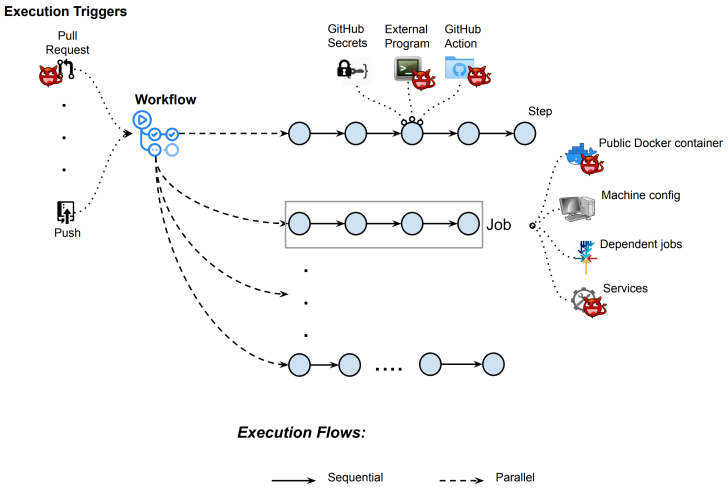

图 2: Threat Model

Contribution

- Security properties (SP)
- Analysis of the five most popular CI/CD platforms
- Attack scenario through third-party scripts hosted on GitHub
- 18% of repositories in dataset use vulnerable third-party scripts

Research Questions

1. What are the **security properties** that need to hold to have a secure CI/CD?
2. How does GitHub CI **compare to other public CI/CD platforms** according to SPs?
3. How does **usage behavior** of workflows affect GitHub CI SPs?

经验性研究

Answer to RQ1: Security Properties

利用**最小权限原则**总结出的影响 CI 工作流的安全要素。

- Admittance Control (AC)：**管理** CI **工作流**；
- Execution Control (EC)：**触发** CI **工作流**；
- Code Control (CC)：**控制** CI **的内容**；
- Access to Secrets (AS)：**管理密钥**；

Answer to RQ2: Permissions

CI/CD Platforms	Permissions	
	Code read	Code write
TravisCI	●✔	◑✖
CircleCI	●✔	◑✖
Jenkins	●✔	●✖
Gitlab CI external	●✔	◑✖
Gitlab CI internal	●✔	○✔
GitHub CI	●✔	●✖

绿勾代表对 *SP* 有益，红叉代表对 *SP* 有害。

Answer to RQ2: Scripting

CI/CD Platforms	Plugins			
	First-party	Third-party	Mutable	Review
TravisCI	●✓	◑✗	○✓	○✗
CircleCI	●✓	●✗	○✓	○✗
Jenkins	○✓	●✗	○✓	○✗
Gitlab CI external	●✓	○✓	○✓	○✗
Gitlab CI internal	●✓	○✓	○✓	○✗
GitHub CI	●✓	●✗	●✗	○✗

表格中的 *Plugins* 即为 *Scripts* 的意思，*Mutable* 代表 *Scripts* 是否可以被平台更新，绿勾代表对 *SP* 有益，红叉代表对 *SP* 有害。

Answer to RQ2

		TravisCI	CircleCI	Jenkins	Gitlab CI extrernal	Gitlab CI internal	Github CI
Admittance Control	(C1) Contributor can add workflow	●	●	●	●	●	●
	(C2) CI/CD run can NOT add new workflow	●	●	○	●	●	○^{W}
	(C3) Executes workflow from PR only after merge	●	◑	○	●	◑	◑^{W}
Execution Control	(C4) Contributors can modify the triggers	●	●	●	●	●	●
	(C5) CI/CD run can NOT modify the triggers	●	●	○	●	●	○^{W}
Code Control	(C6) CI/CD run can NOT modify the code	●	●	○	●	●	○^{W}
	(C7) CI/CD run is deterministic based on config	●	●	●	●	●	○^{W}
Access to Secrets	(C8) Masked	●	●	◑	●	●	●
	(C9) Accessible only to explicitly authorized steps	○	○	◑	●	●	○
	(C10) Restricted from pull requests	●	◑	◑	●	◑	◑^{W}

表格中的红色代表对 *SP* 有害，*W* 上标表示该属性是 *workflow-dependent* 的（即 *sementic* 的）。

数据分析

数据收集

GHArchive ➦

GitHub REST API

11,438 Scripts

213,854 Repositories

Workflows

Workflow Permissions only 0.2% of all workflows use permissions

Workflow Triggers 51.7% of public repositories run on **self-hosted machines** can be triggered by PR

Workflow Secrets Third-party scripts can **access the secrets** & Some developers pass the secrets in **plain text** to allow forked versions to run the workflows

Third-party Scripts

Verified vs Unverified Scripts The majority of the scripts are from non-verified creators (97%).

Third-Party Scripts' References Developers do not reference Third-party scripts by using **commit hash**, despite the security risks.

Vulnerability Analysis 38,315 or 17.9% use **at least one potentially vulnerable scripts** due to not upgrading the version.

在漏洞分析中使用了 *git-vuln-finder* ➦，**通过** *Git Commit Log* **寻找可能的漏洞**。

两篇 Paper 的比较

两篇 Paper 的比较

对比项	USENIX SEC'22	Our paper
包装 & 抽象	Security Properties	✗
主次 Domain 处理	对比法	✗
对 Vulnerablities 的分析	git-vuln-finder	已有的 CVE
炫技性质的工具	GWChecker	✗

启示与改进

Next Paper

基本风格	**检测** + Measurement
PoC Writing	
学习方法论	**静态分析的基本原理** & codeql **的使用**
Measurement	**基于静态分析的结果做** Measurement
参考	Probe the Proto ➦

谢谢大家，敬请指正！