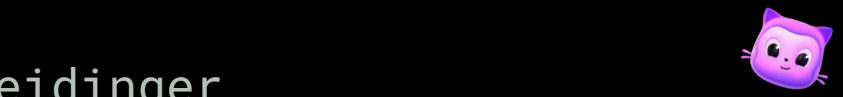
GitHub

S3C Update GitHub 2025

@kommendorkapten @tinaheidinger



Who are we?

Tina Heidinger

Senior Product Manager

Fredrik Skogman

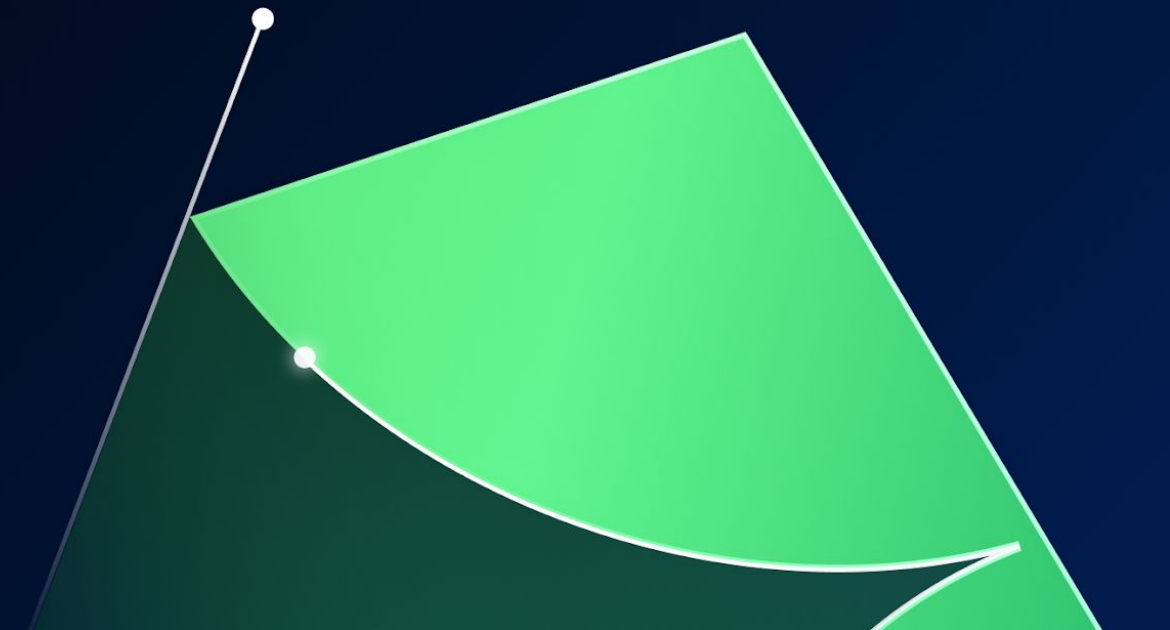
Staff Software Engineer

@tinaheidinger

@kommendorkapten

Agenda

- Artifact Attestations
- Demo
- Customer adoption
- Lessons learned
- Future plans & roadmap



GitHub Artifact Attestations

Guarantees integrity for artifacts built on GitHub Actions

Offering a simple path to Sigstore based signing for all OSS

GA since June 2024

Available for OSS npm since October 2023

Free for open source

Early adopters include Homebrew

All bottles built with attested build provenance

Feature is still in beta (Homebrew verification)

GitHub Artifact Attestation

- Capture non-forgeable metadata about the build (provenance)
- Prove integrity from source to build step to consumer
  - Verifiable metadata allows for rich policies
- Use workload identities instead of human identities
- GitHub provides PKI
  - Developer doesn’t need to manage keys

Built on open source

Sigstore

OpenSSF project

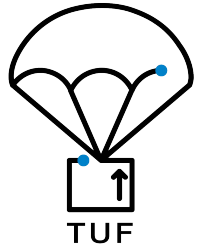
Signing and verification of binary artifacts

Public Good Instance

SLSA (Supply-chain Levels for Software Artifacts)

OpenSSF Project

Open specification for build provenance as **in-toto** predicates



The Update Framework

CNCF project

Secure updates over untrusted channels

Secure trust root management and delivery

Why not just use PGI Sigstore?

- Offer a “battery included” experience of using Sigstore
- Sigstore only signs and verifies – integration to build systems has to be provided
- Build provenance generation
- Attestation discovery and storage
  - Sigstore doesn’t offer a solution
  - Access controls
  - Content addressable storage

GitHub Artifact Attestations

How to use it

Create an attestation:

Add the following to your GitHub Actions workflow that creates the artifact, and has `attestations:write` permissions:

```
- name: Attest Build Provenance
    uses: actions/attest-build-provenance@v2
    with:
        subject-path: “bin/my-artifact.tar.gz”
```

Verify an attestation:

- GitHub CLI

  ```
  gh attestation verify
  my-artifact.tar.gz -o
  my-organization
  ```

- Kubernetes Admission Controller
- Download attestation for offline verification

Security considerations

- Signed != secure
- Security best practices for builds have to be followed
- CODEOWNERS
- Reusable workflows (SLSA provides one)
  - Separation of build instructions and code
  - Build isolation

Relation to real world attacks

- **@solana/web3.js** (December 2024)
  - It appears malicious actors got push access to npm
  - No build provenance generated
- **Ultralytics** (December 2024)
  - GitHub Action template injection attack
  - Exfiltrated push token to PyPI
  - Transparency log entry/attestation proved very useful during forensic analysis
  - Second release did not contain build provenance
- **Kong ingress controller** (January 2025)
  - DockerHub push credentials stolen via Pwn Request
  - No build provenance generated

Customer adoption & success stories

- Loved by thousands of organizations
- Integrated with OSS ecosystems
- Adoption drivers: Build integrity, compliance, adherence to industry standards and best practices (SLSA)
- Integration through consolidation of CI/CD on an enforced set of reusable workflows

Lessons learned

- **Ease of use is key!**
- Customers find it challenging to determine what to verify and where in the SDLC to enforce policies
- Custom use cases require a flexible attestation framework
- Tool interoperability and vencor-agnostic solutions matter
- Provenance is just the beginning

Future plans & roadmap

- Enhanced attestation lifecycle management capabilities
- Tooling compatibility - meet customers where they are
- Deeper integration into GitHub (immutable actions, immutable releases)
- Artifacts as first-class citizens
- Prioritization of security alerts based on artifact metadata


Thank you