Fredrik Skogman
@kommendorkapten

- Staff Engineer @ GitHub
- Supply Chain Security
- Active OSS maintainer
  - Sigstore
  - The Update Framework (TUF)

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

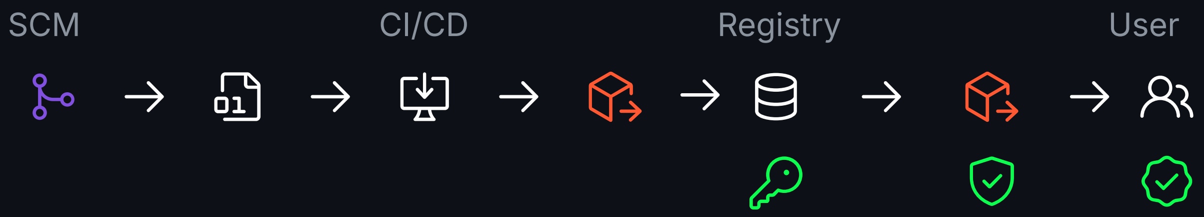
Free for open source

Early adopters include Homebrew

All bottles built with attested build provenance

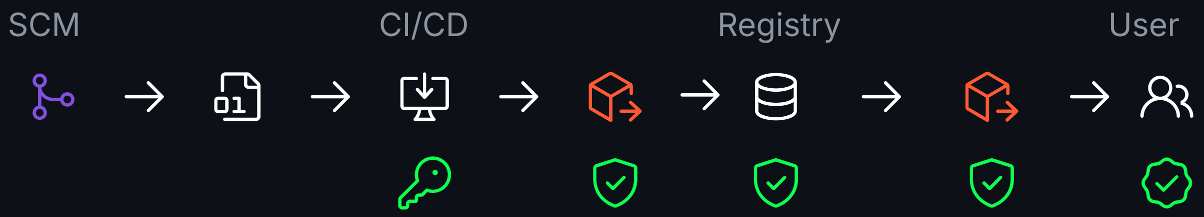
Feature is still in beta

“Classical signing”



- Raw signature over artifact
- Integrity is verifiable
- Signature is lacking context
    - No verifiable metadata
- PKI can be complex

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

Components

- GitHub Action (OSS)
- CA + PKI (OSS)
- Attestation Store
- gh cli (OSS)
- K8s admission controller (OSS, Sigstore policy controller)

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

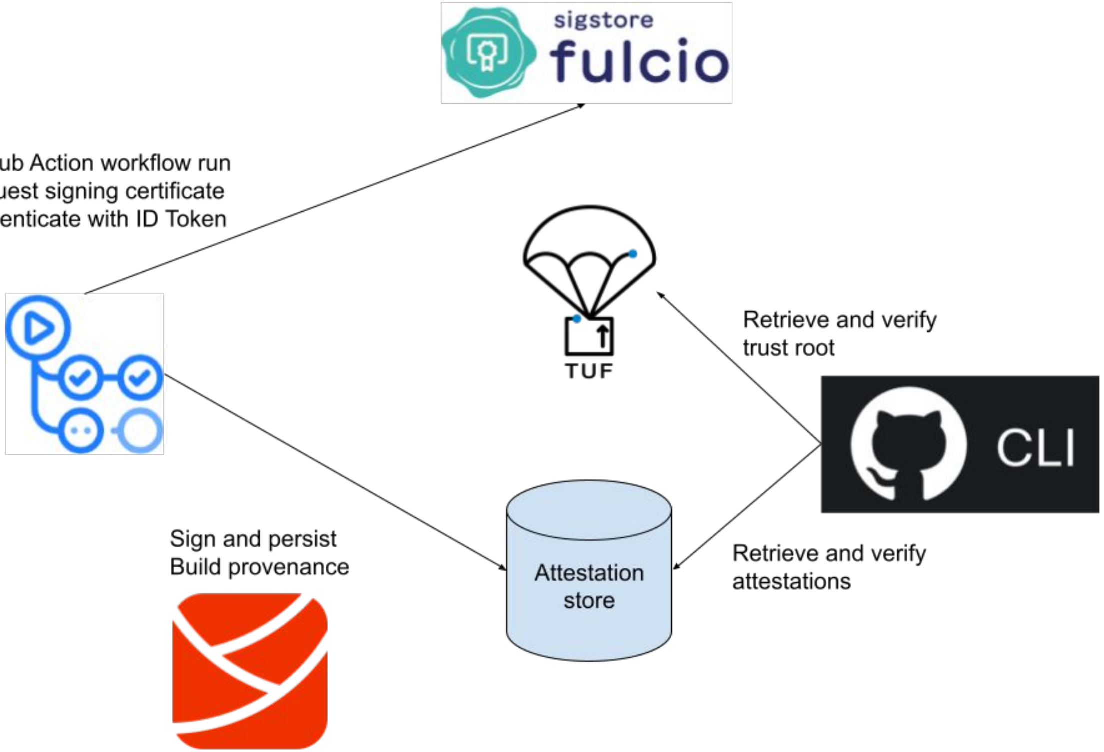
CNCF project

Secure updates over untrusted channels

Secure trust root management and delivery

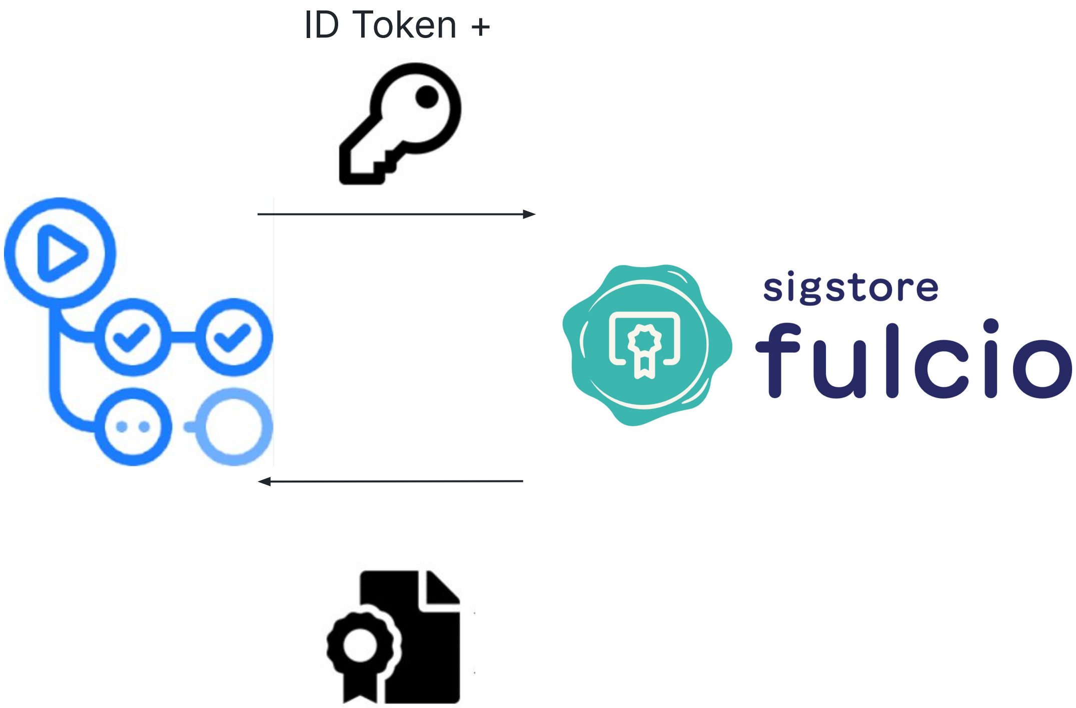
Simplified Overview

Signed timestamps are not shown



Sigstore ephemeral signing overview

1. Generate in-memory private key-pair
2. Send public key and ID token to Fulcio
3. Receive a signing certificate (10 minutes expiration)
4. Sign build provenance
5. Acquire a signed timestamp
6. Persist signature on a transparency log (optional)



Signing and verification

👎

- Key management
    - HSM or file on disk?
    - Is the key lost or compromised?
- Key distribution and selection
    - Which key to verify with
- Identity management
    - Is this person allowed to sign

👍

- Sigstore Fulcio as CA
    - Identity federation
- No key management
    - Ephemeral (one-time use) keys
- Use workload identities instead of human identities
    - Artifact was signed by org/repo/workflow

GitHub Artifact Attestations

The details

Sigstore operations

Public Good Instance (PGI)

For public repositories

Free for all

Operated by the sigstore community

Signing and identity information persisted on public append only ledgers

GitHub private instance

For private repositories

Privacy guarantee

All happens within GitHub

Fully compatible with PGI, tooling can be reused

Build provenance

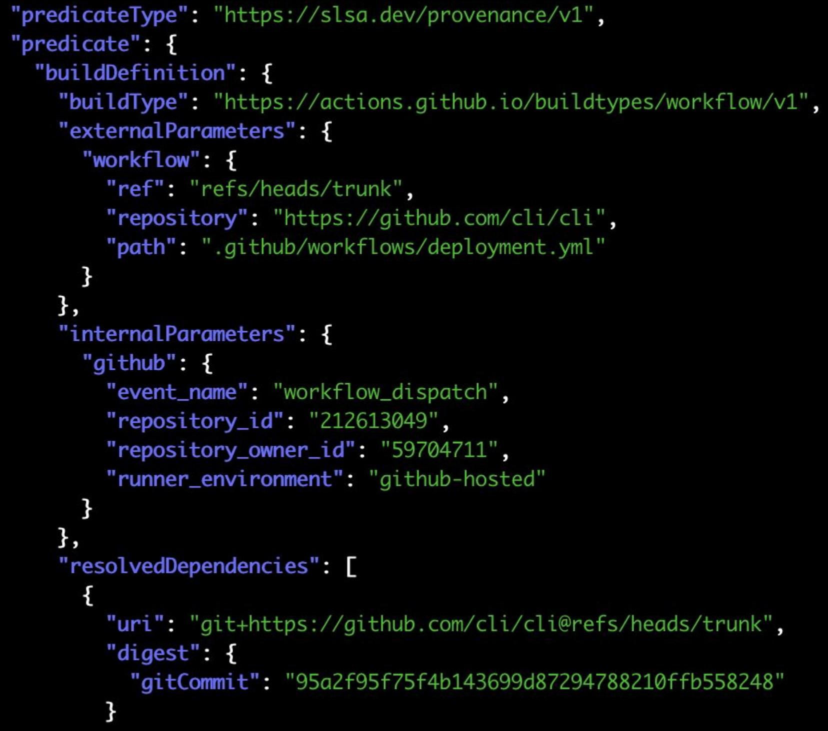
Automatically generated during build

Captures metadata of the build which includes

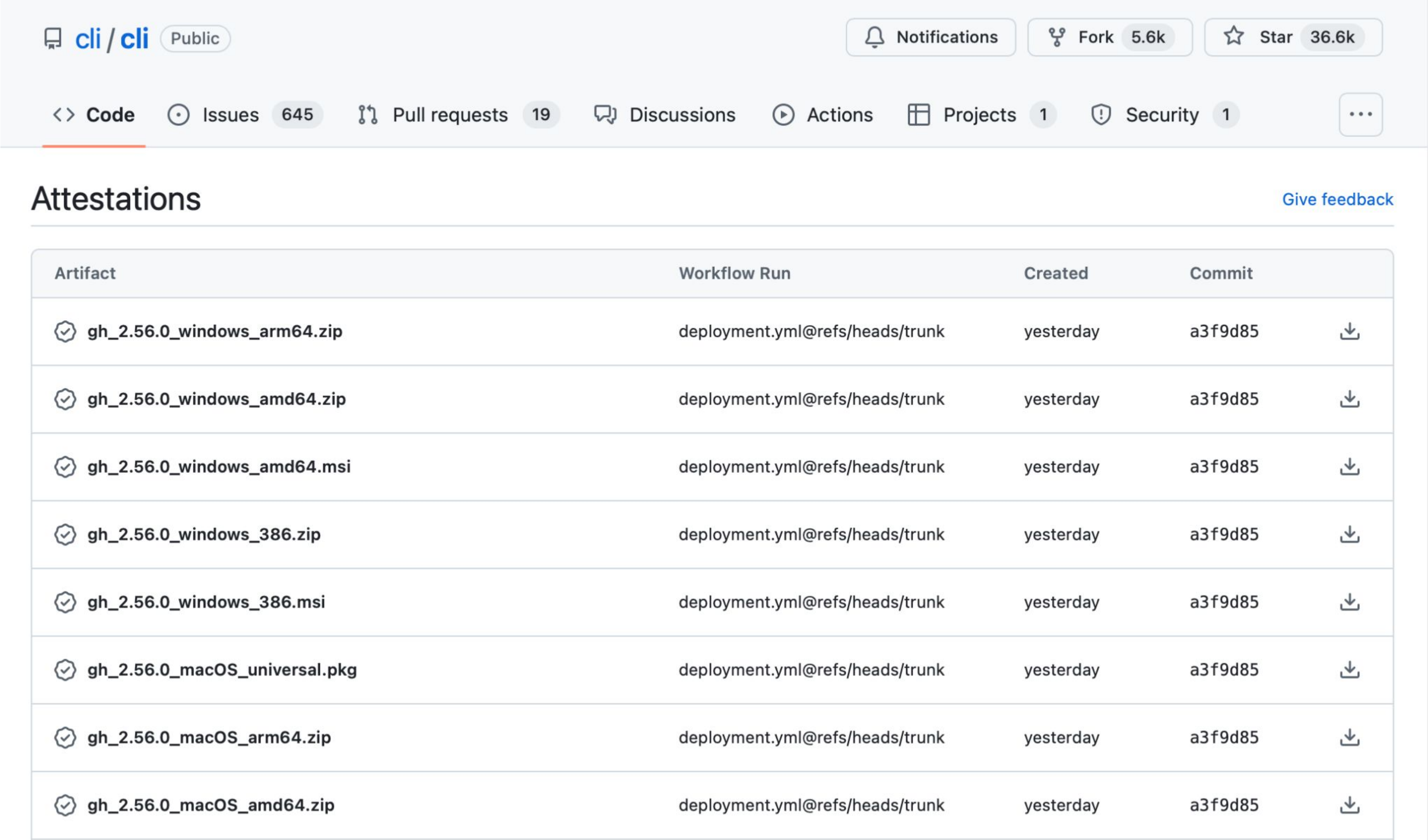
Owner/repository

Git commit/ref

Workflow used

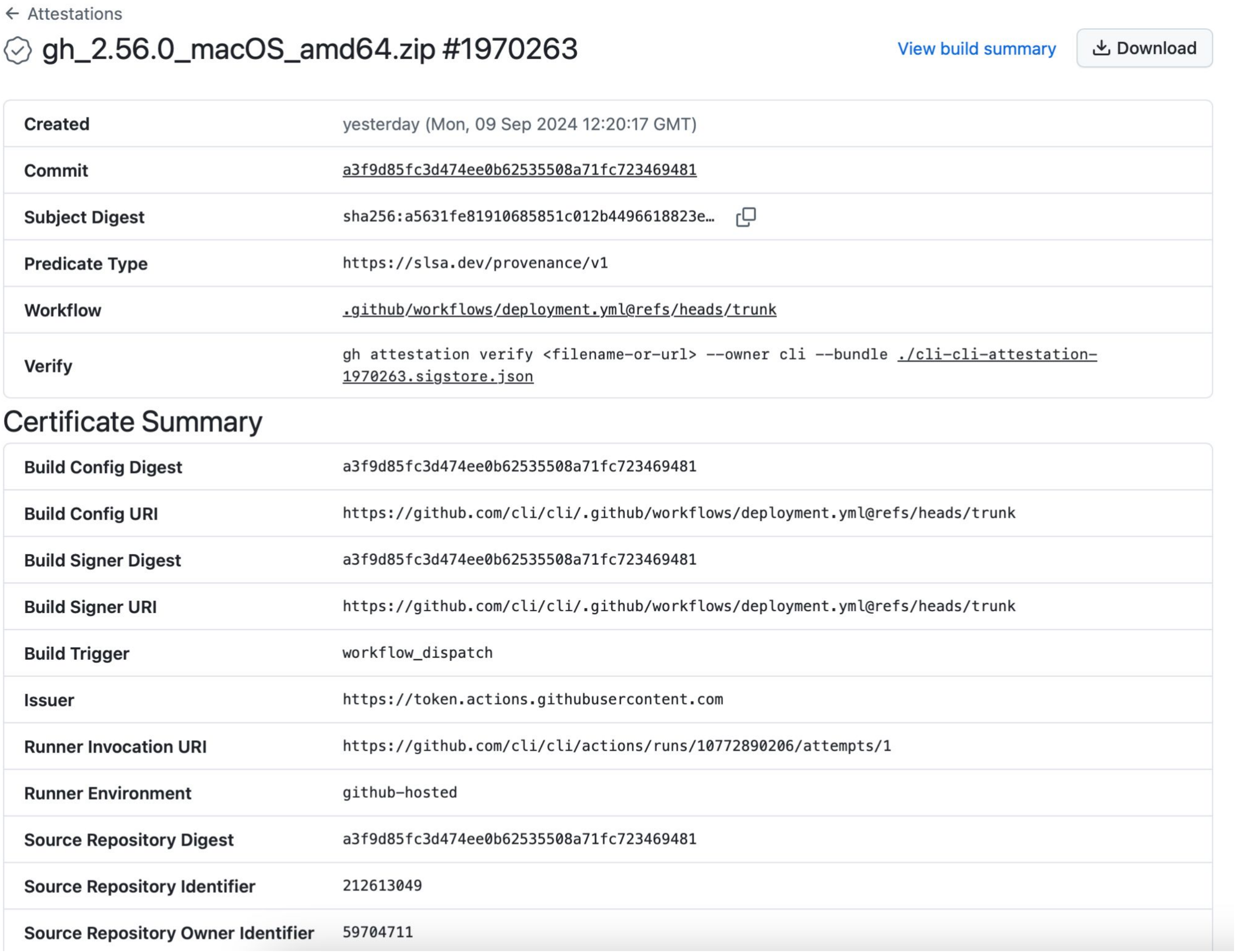


```
"predicateType": "https://slsa.dev/provenance/v1",
"predicate": {
  "buildDefinition": {
    "buildType": "https://actions.github.io/buildtypes/workflow/v1",
    "externalParameters": {
      "workflow": {
        "ref": "refs/heads/trunk",
        "repository": "https://github.com/cli/cli",
        "path": ".github/workflows/deployment.yml"
      }
    },
    "internalParameters": {
      "github": {
        "event_name": "workflow_dispatch",
        "repository_id": "212613049",
        "repository_owner_id": "59704711",
        "runner_environment": "github-hosted"
      }
    },
    "resolvedDependencies": [
      {
        "uri": "git+https://github.com/cli/cli@refs/heads/trunk",
        "digest": {
          "gitCommit": "95a2f95f75f4b143699d87294788210ffb558248"
        }
```



Attestations

Artifact	Workflow Run	Created	Commit
gh_2.56.0_windows_arm64.zip	deployment.yml@refs/heads/trunk	yesterday	a3f9d85
gh_2.56.0_windows_amd64.zip	deployment.yml@refs/heads/trunk	yesterday	a3f9d85
gh_2.56.0_windows_amd64.msi	deployment.yml@refs/heads/trunk	yesterday	a3f9d85
gh_2.56.0_windows_386.zip	deployment.yml@refs/heads/trunk	yesterday	a3f9d85
gh_2.56.0_windows_386.msi	deployment.yml@refs/heads/trunk	yesterday	a3f9d85
gh_2.56.0_macOS_universal.pkg	deployment.yml@refs/heads/trunk	yesterday	a3f9d85
gh_2.56.0_macOS_arm64.zip	deployment.yml@refs/heads/trunk	yesterday	a3f9d85
gh_2.56.0_macOS_amd64.zip	deployment.yml@refs/heads/trunk	yesterday	a3f9d85



gh_2.56.0_macOS_amd64.zip #1970263

Created	yesterday (Mon, 09 Sep 2024 12:20:17 GMT)
Commit	`a3f9d85fc3d474ee0b62535508a71fc723469481`
Subject Digest	`sha256:a5631fe81910685851c012b4496618823e…`
Predicate Type	`https://slsa.dev/provenance/v1`
Workflow	`.github/workflows/deployment.yml@refs/heads/trunk`
Verify	`gh attestation verify <filename-or-url> --owner cli --bundle ./cli-cli-attestation-1970263.sigstore.json`

Certificate Summary

Build Config Digest	`a3f9d85fc3d474ee0b62535508a71fc723469481`
Build Config URI	`https://github.com/cli/cli/.github/workflows/deployment.yml@refs/heads/trunk`
Build Signer Digest	`a3f9d85fc3d474ee0b62535508a71fc723469481`
Build Signer URI	`https://github.com/cli/cli/.github/workflows/deployment.yml@refs/heads/trunk`
Build Trigger	`workflow_dispatch`
Issuer	`https://token.actions.githubusercontent.com`
Runner Invocation URI	`https://github.com/cli/cli/actions/runs/10772890206/attempts/1`
Runner Environment	`github-hosted`
Source Repository Digest	`a3f9d85fc3d474ee0b62535508a71fc723469481`
Source Repository Identifier	`212613049`
Source Repository Owner Identifier	`59704711`

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

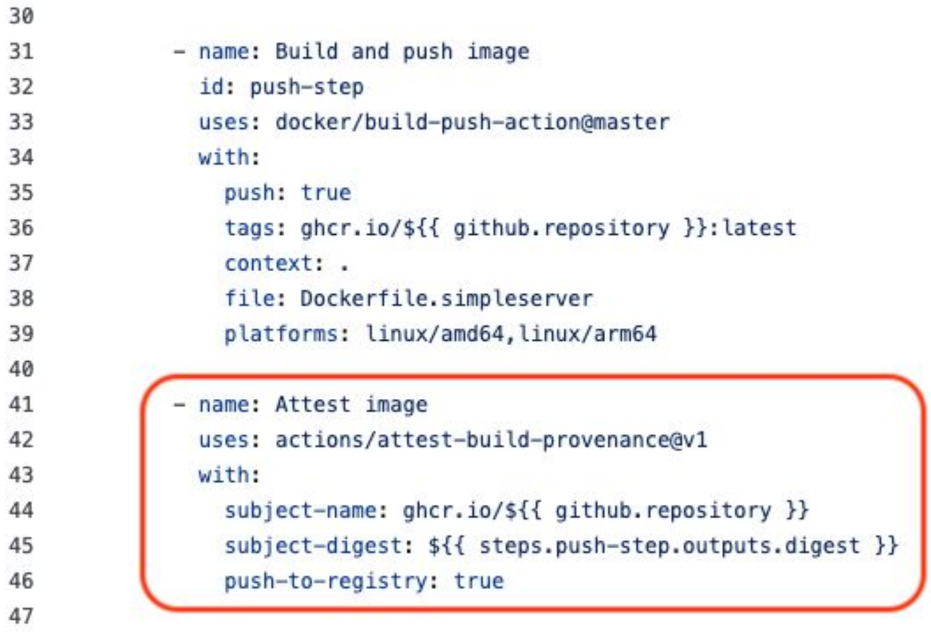
Attestation discovery and storage

- Sigstore does offer a solution
- Access controls
- Content addressable storage

GitHub Artifact Attestations

How to use it

Enablement



```yaml
- name: Build and push image
  id: push-step
  uses: docker/build-push-action@master
  with:
    push: true
    tags: ghcr.io/${{ github.repository }}:latest
    context: .
    file: Dockerfile.simpleserver
    platforms: linux/amd64,linux/arm64

- name: Attest image
  uses: actions/attest-build-provenance@v1
  with:
    subject-name: ghcr.io/${{ github.repository }}
    subject-digest: ${{ steps.push-step.outputs.digest }}
    push-to-registry: true
```

Ready to use action

A few lines of yaml

Arbitrary artifacts can be attested

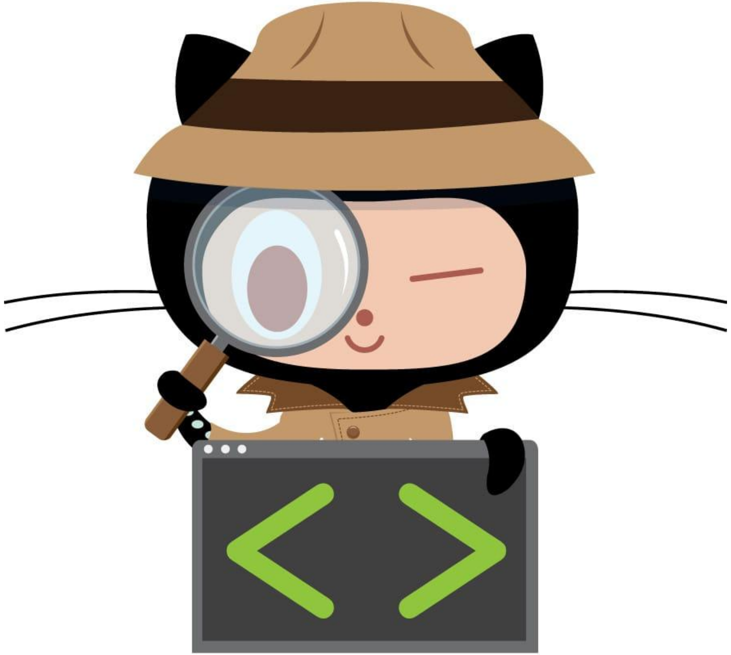
👍 **Yes, SBOMs**

Verification

Compatible with cosign (primary sigstore cli)

The GitHub cli

Sigstore policy controller (k8s admission controller)



Security considerations

Signed != secure

Security best practices for builds have to be followed

CODEOWNERS

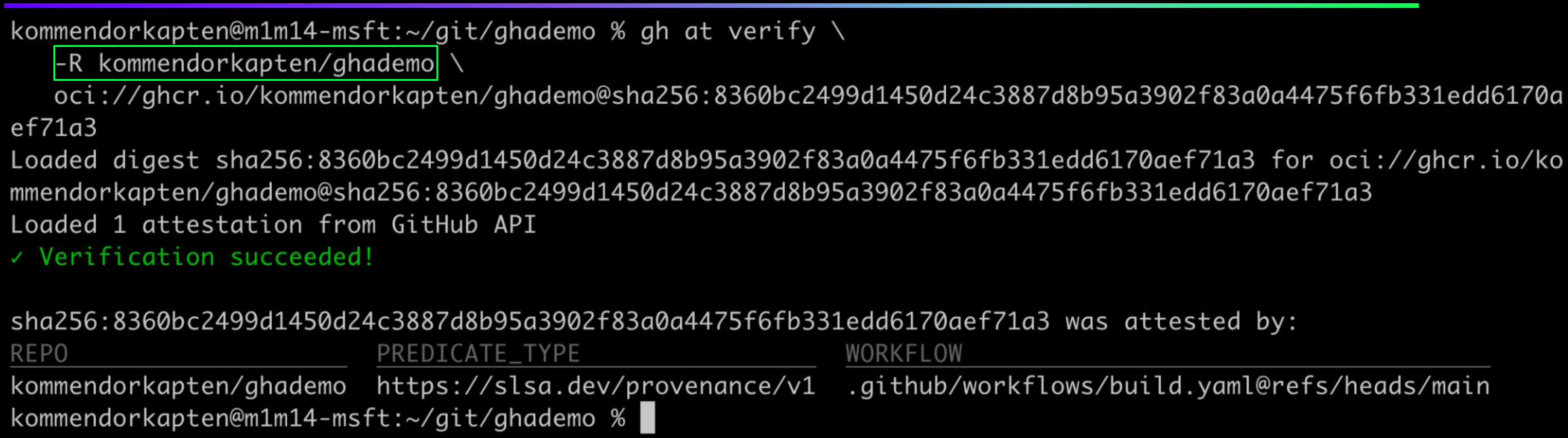
Reusable workflows (SLSA provides one)

- 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

Demo - OCI



```
kommendorkapten@m1m14-msft:~/git/ghademo % gh at verify \
   -R kommendorkapten/ghademo \
   oci://ghcr.io/kommendorkapten/ghademo@sha256:8360bc2499d1450d24c3887d8b95a3902f83a0a4475f6fb331edd6170a
ef71a3
Loaded digest sha256:8360bc2499d1450d24c3887d8b95a3902f83a0a4475f6fb331edd6170aef71a3 for oci://ghcr.io/ko
mmendorkapten/ghademo@sha256:8360bc2499d1450d24c3887d8b95a3902f83a0a4475f6fb331edd6170aef71a3
Loaded 1 attestation from GitHub API
✓ Verification succeeded!

sha256:8360bc2499d1450d24c3887d8b95a3902f83a0a4475f6fb331edd6170aef71a3 was attested by:
REPO                        PREDICATE_TYPE                  WORKFLOW
kommendorkapten/ghademo   https://slsa.dev/provenance/v1  .github/workflows/build.yaml@refs/heads/main
kommendorkapten@m1m14-msft:~/git/ghademo %
```

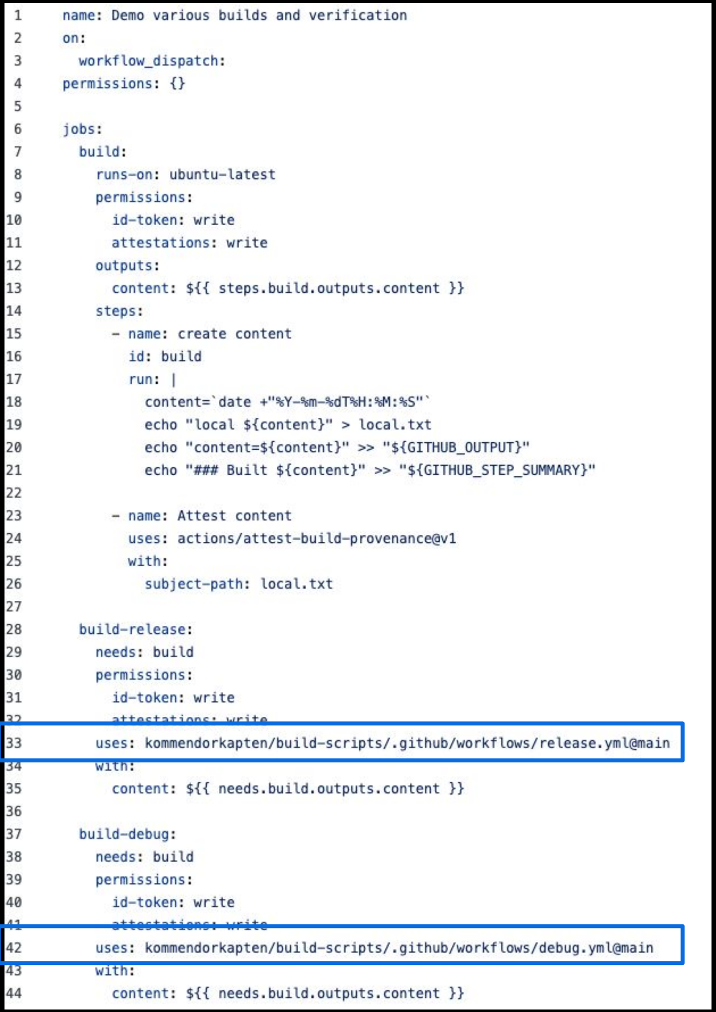
The cli fetches the referenced manifest
Retrieves attestations via the message digest of the downloaded manifest
Verify cryptographic signatures up to GitHub’s root CA
Ensure the index manifest originated from kommendorkapten/ghademo repository
Flag ‘--format json’ can be added for machine readable output

Demo workflow

Three artifacts built
local.txt is built and signed by this workflow
debug.txt is “built” and signed by a reusable workflow
release.txt is “built” and signed by a reusable workflow

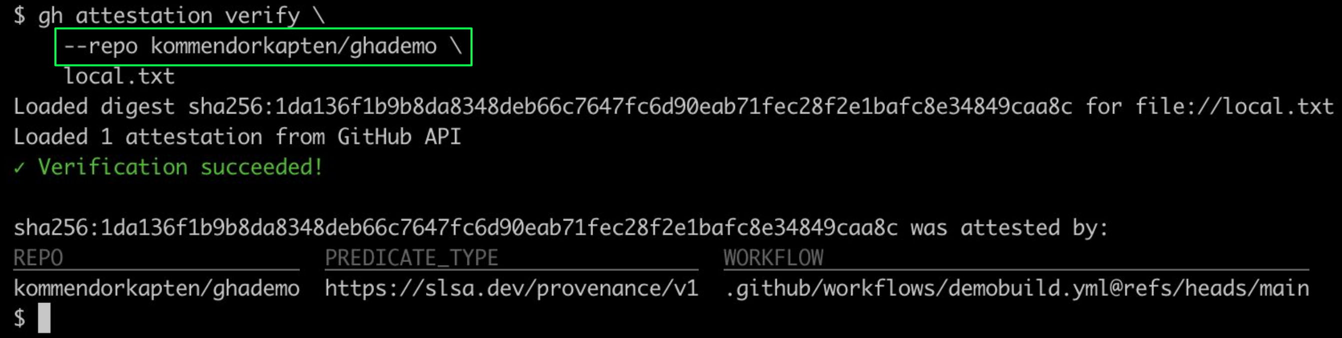
This provides us with capabilities to understand two important properties:

1. Where did the source materials originate from
2. Who built and signed the artifact



```yaml
name: Demo various builds and verification
on:
  workflow_dispatch:
permissions: {}

jobs:
  build:
    runs-on: ubuntu-latest
    permissions:
      id-token: write
      attestations: write
    outputs:
      content: ${{ steps.build.outputs.content }}
    steps:
      - name: create content
        id: build
        run: |
          content=`date +"%Y-%m-%dT%H:%M:%S"`
          echo "local ${content}" > local.txt
          echo "content=${content}" >> "${GITHUB_OUTPUT}"
          echo "### Built ${content}" >> "${GITHUB_STEP_SUMMARY}"

      - name: Attest content
        uses: actions/attest-build-provenance@v1
        with:
          subject-path: local.txt

  build-release:
    needs: build
    permissions:
      id-token: write
      attestations: write
    uses: kommendorkapten/build-scripts/.github/workflows/release.yml@main
    with:
      content: ${{ needs.build.outputs.content }}

  build-debug:
    needs: build
    permissions:
      id-token: write
      attestations: write
    uses: kommendorkapten/build-scripts/.github/workflows/debug.yml@main
    with:
      content: ${{ needs.build.outputs.content }}
```

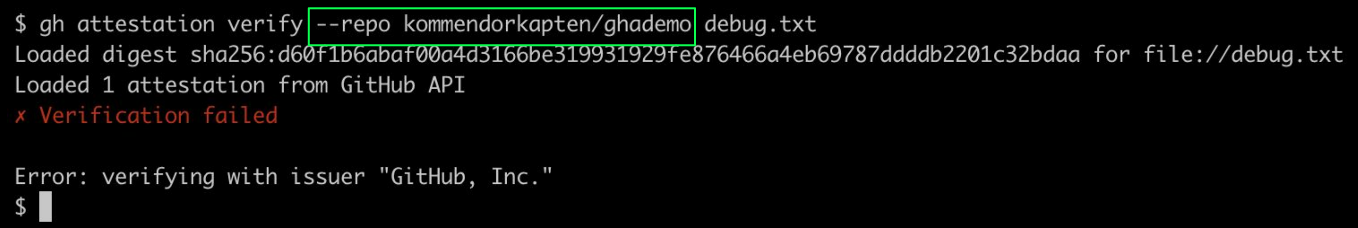
Demo - 1



```
$ gh attestation verify \
    --repo kommendorkapten/ghademo \
    local.txt
Loaded digest sha256:1da136f1b9b8da8348deb66c7647fc6d90eab71fec28f2e1bafc8e34849caa8c for file://local.txt
Loaded 1 attestation from GitHub API
✓ Verification succeeded!

sha256:1da136f1b9b8da8348deb66c7647fc6d90eab71fec28f2e1bafc8e34849caa8c was attested by:
REPO                     PREDICATE_TYPE                  WORKFLOW
kommendorkapten/ghademo  https://slsa.dev/provenance/v1  .github/workflows/demobuild.yml@refs/heads/main
$ 
```

The cli computes the message digest of local.txt
Retrieves attestations via the message digest
Verify cryptographic signatures up to GitHub’s root CA
Ensure artifact (local.txt) originated from kommendorkapten/ghademo repository
Flag ‘--format json’ can be added for machine readable output

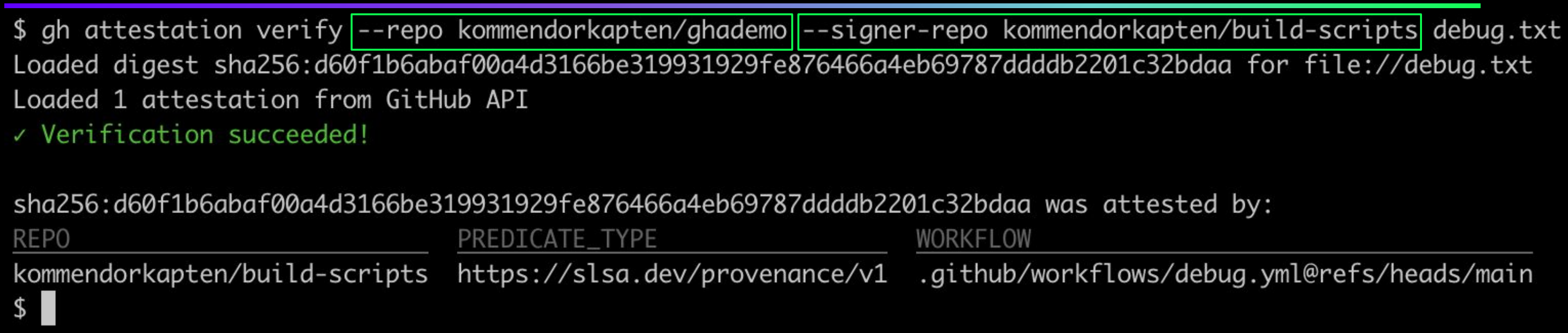
Demo - 2



```
$ gh attestation verify --repo kommendorkapten/ghademo debug.txt
Loaded digest sha256:d60f1b6abaf00a4d3166be319931929fe876466a4eb69787ddddb2201c32bdaa for file://debug.txt
Loaded 1 attestation from GitHub API
✗ Verification failed

Error: verifying with issuer "GitHub, Inc."
$ 
```

The cli computes the message digest of local.txt
Retrieves attestations via the message digest
Verify cryptographic signatures up to GitHub’s root CA
As the artifact is signed by a different repository (the reusable workflow), the signer’s identity does not match the provided one

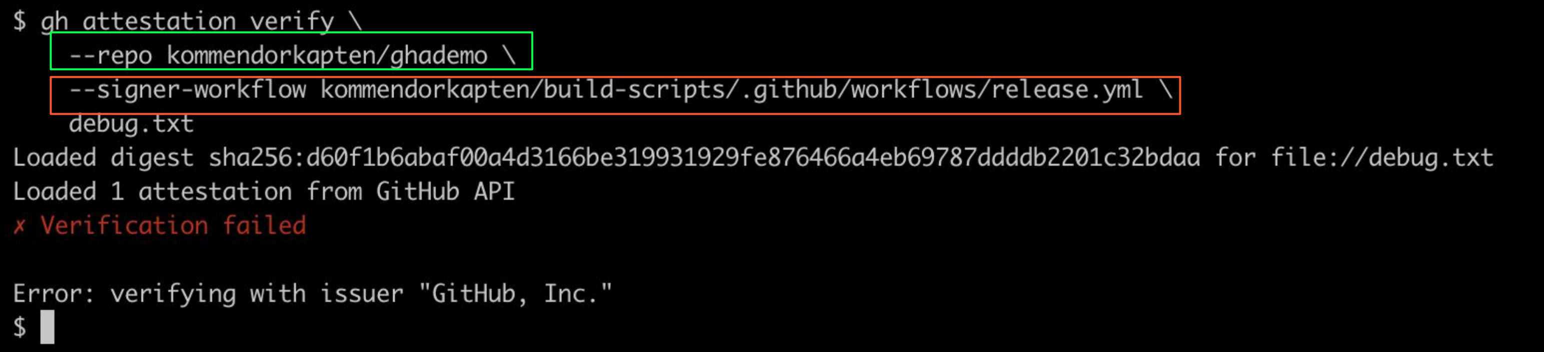
Demo - 3



```
$ gh attestation verify --repo kommendorkapten/ghademo --signer-repo kommendorkapten/build-scripts debug.txt
Loaded digest sha256:d60f1b6abaf00a4d3166be319931929fe876466a4eb69787dddb2201c32bdaa for file://debug.txt
Loaded 1 attestation from GitHub API
✓ Verification succeeded!

sha256:d60f1b6abaf00a4d3166be319931929fe876466a4eb69787dddb2201c32bdaa was attested by:
REPO                          PREDICATE_TYPE                 WORKFLOW
kommendorkapten/build-scripts https://slsa.dev/provenance/v1 .github/workflows/debug.yml@refs/heads/main
$ 
```

The cli computes the message digest of local.txt
Retrieves attestations via the message digest
Verify cryptographic signatures up to GitHub's root CA
Originating and signing repository matches – verification succeeds

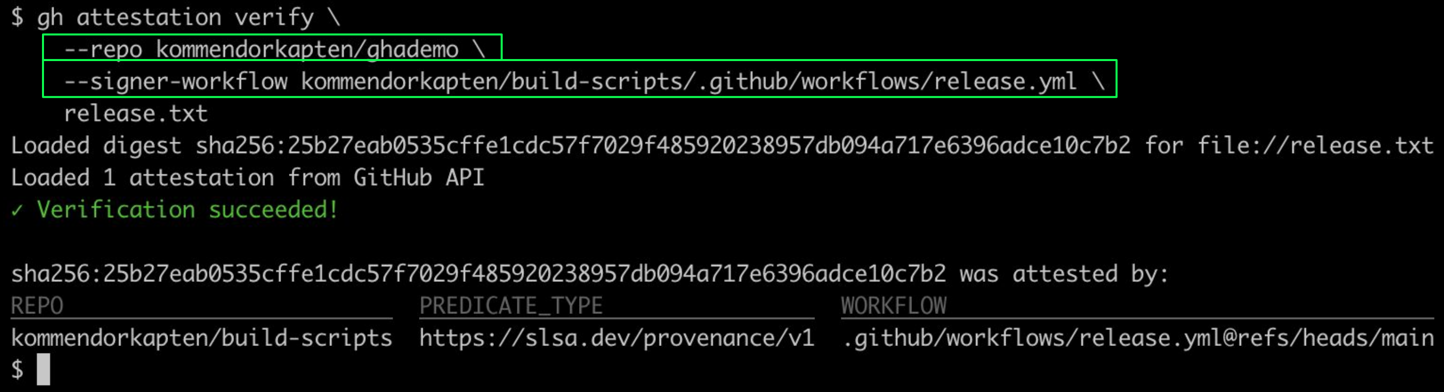
Demo - 4



```
$ gh attestation verify \
    --repo kommendorkapten/ghademo \
    --signer-workflow kommendorkapten/build-scripts/.github/workflows/release.yml \
    debug.txt
Loaded digest sha256:d60f1b6abaf00a4d3166be319931929fe876466a4eb69787ddddb2201c32bdaa for file://debug.txt
Loaded 1 attestation from GitHub API
✗ Verification failed

Error: verifying with issuer "GitHub, Inc."
$ 
```

The cli computes the message digest of local.txt
Retrieves attestations via the message digest
Verify cryptographic signatures up to GitHub’s root CA
Artifact was not built and signed by the release workflow

Demo - 5



```
$ gh attestation verify \
    --repo kommendorkapten/ghademo \
    --signer-workflow kommendorkapten/build-scripts/.github/workflows/release.yml \
    release.txt
Loaded digest sha256:25b27eab0535cffe1cdc57f7029f485920238957db094a717e6396adce10c7b2 for file://release.txt
Loaded 1 attestation from GitHub API
✓ Verification succeeded!

sha256:25b27eab0535cffe1cdc57f7029f485920238957db094a717e6396adce10c7b2 was attested by:
REPO                           PREDICATE_TYPE                  WORKFLOW
kommendorkapten/build-scripts  https://slsa.dev/provenance/v1  .github/workflows/release.yml@refs/heads/main
$
```

The cli computes the message digest of local.txt
Retrieves attestations via the message digest
Verify cryptographic signatures up to GitHub's root CA
Originating repository and signing workflow matches – verification succeeds

Thank you!

Questions?