Red Hat Summit

Connect

Every organization needs a robust strategy for verifying the origin and integrity of container images

Gareth Healy

Dan Hawker

The Submission Blurb

You trust Red Hat as a provider of content, but how can you verify that the content has not been a victim of corruption or has been tampered with in transit?

Supporting Content (800 chars):
This talk will demonstrate how customers can verify the OCP release and core components via automated pipelines and built-in features of OCP using cryptographic keys, providing assurance that the images being used were built and provided by Red Hat.

We'll talk about the core concepts of signing and verifying container images and how customers can verify Red Hat content. We believe this will help them on their journey to start signing and verifying their own in-house images. We'll provide practical examples that customers can use today to enable them to start on their Software Supply Chain Security journey.

Topics: Application Platform, CI/CD, Compliance, Containers, DevSecOps, GitOps, Security, Software Supply Chain Security
Products: OCP, RHTAS
Demo: Yes

What we will discuss and show today…

Discuss..

- What do we mean by "Trust but Verify"?
- Why is this important and How have people been doing this?
- Introduction to Trusted Software Supply Chain
- What's next for you?

Show…

- Verifying Red Hat Content via
  - OpenShift
  - Podman/Buildah/Skopeo

Technology illustrations by Storyset

Gareth Healy

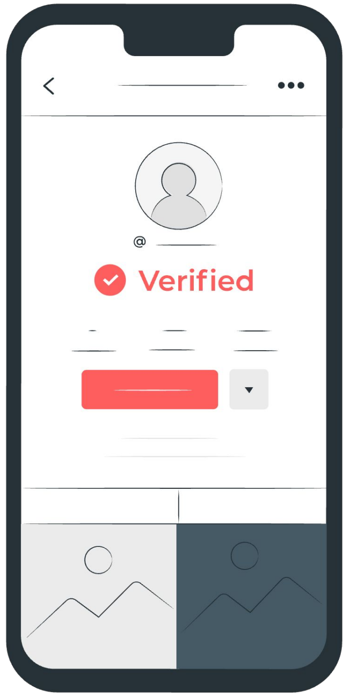
Principal Architect
Red Hat UK&I Services

Dan Hawker

Principal Architect
Red Hat UK&I Services

What and Why?

What do we mean by “Trust but Verify”?



- Trust is inferred between you and Red Hat, if you are a customer
- Trust doesn't stop bad-actors either within Red Hat, between Red Hat and your network or within you own network; *verification* does
- You’ve been automatically verifying RHEL RPMs for years, *even if you didn't realise*, so why should containers be any different?

Why is this important?

- Weekly occurrences of supply chain attacks and typosquatting which means developers are constantly under attack and run the risk of accidentally using malicious packages
- Access tokens are regularly stolen/leaked and typically, provide the attacker access to push compromised changes
- Can you *automatically* assert that the components running in your environment were built by the trusted source?

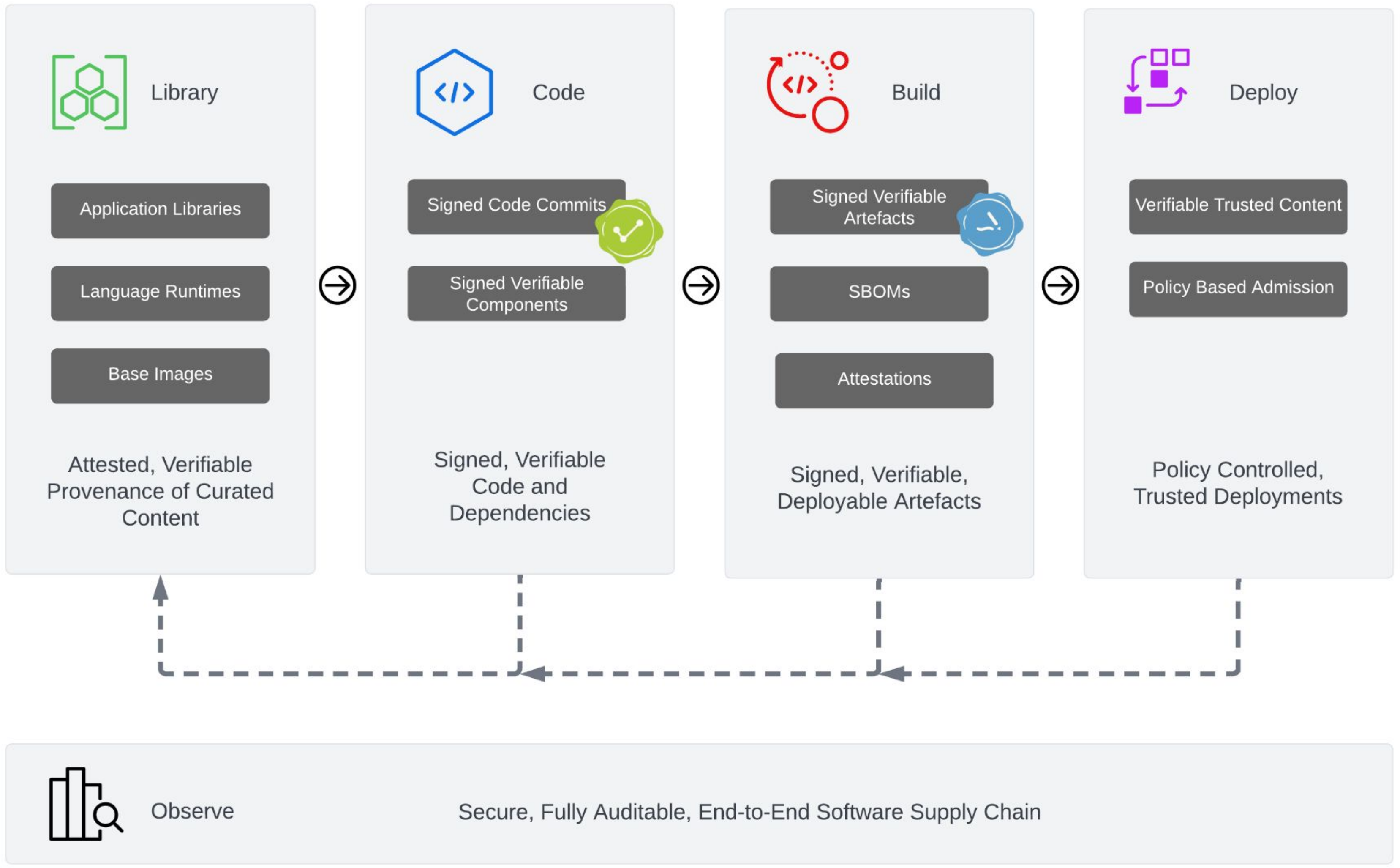
Technology illustrations by Storyset

How have people been doing this?

- Trust but verify is not a new concept in the container space:
  - Container PGP signing was supported in OCP3
  - Notary v1 was released in 2015 and has gone through multiple iterations since
  - Pull via SHA and verify remote SHA matches local SHA

How?

Trusted Software Supply Chain Concepts


Library
Application Libraries
Language Runtimes
Base Images
Attested, Verifiable Provenance of Curated Content
Code
Signed Code Commits
Signed Verifiable Components
Signed, Verifiable Code and Dependencies
Build
Signed Verifiable Artefacts
SBOMs
Attestations
Signed, Verifiable, Deployable Artefacts
Deploy
Verifiable Trusted Content
Policy Based Admission
Policy Controlled, Trusted Deployments
Observe
Secure, Fully Auditable, End-to-End Software Supply Chain

Open Source Collaboration in The TSSC Community

Innovations coming from Open Source communities:

- In-toto
- SLSA
- TUF
- Sigstore

Verifying via OpenShift

OCP 4.18 introduced a Tech Preview feature enabled via the **TechPreviewNoUpgrade** feature gate:

- ClusterImagePolicy
- ImagePolicy

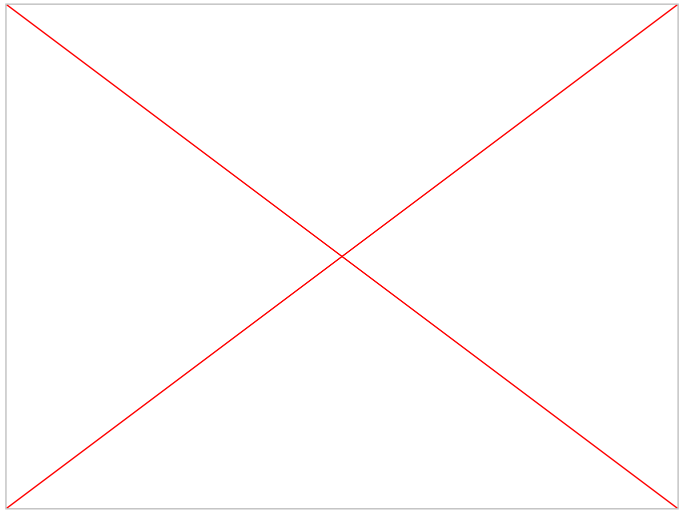
*What does Tech Preview mean?

Verifying via Podman/Buildah/Skopeo

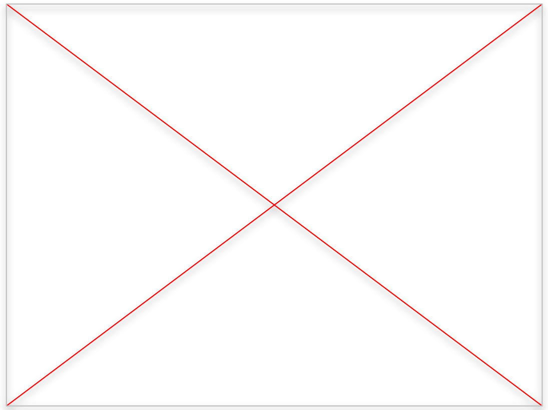
The OCP feature piggy-backs the containers/image feature used by cri-o…

…which means Podman/Buildah/Skopeo can all use the same config as well!

Demo



```
gahealy@gahealy-mac:~/Documents/Git/github.com/garethahealy/verifying-redhat-images
→ verifying-redhat-images git:(main) ✗
```



https://github.com/garethahealy/verifying-redhat-images

Next?

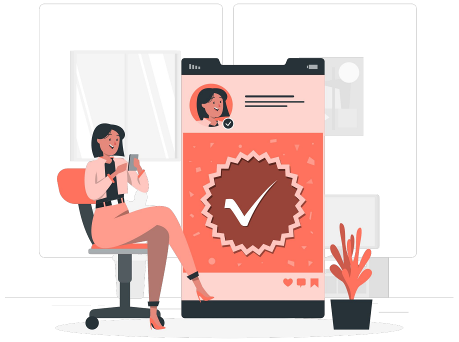
Thinking about Signing your own Images?

- Red Hat Trusted Artifact Signer (RHTAS)
  - Runs on OpenShift/RHEL and based on Sigstore
  - You control the PKI or OIDC for keyless

- Red Hat Advanced Cluster Security (RHACS)
  - You've signed the images with TAS, now lets verify and enforce with RHACS

Interested in exploring? Speak to your account manager on how Red Hat can help you on this journey

Signing is cool, but Attestations are great!



- What is an Attestation?
  - Define: attestation – "evidence or proof of something"

- Attestations can provide the verifier with signed metadata on:
  - What Git commit the build is from
  - Where it was built
  - Who triggered the release

- This creates an Audit trail for when things go bad…

- And decision points on whether to allow artifacts into your environments

Technology illustrations by Storyset

Other tools we like...

- GitSign
  - Tried using PGP GitSigning? GitSign is commit signing with zero developer headache
  - Supported binary via RHTAS
- Renovatebot
  - Remove the churn of updating dependency versions
  - Pin container base images, GitHub Actions and Gitlab Pipeline Includes
  - Can be part of an *Evergreening* approach
- Step-Security harden-runner GitHub Action
  - Detect egress and source code changes at pipeline run
  - Discovered several supply chain attacks against:
    - tj-actions/changed-files compromise (CVE-2025-30066)
    - Google's Open-Source Project Flank
    - Microsoft's Open-Source Project Azure Karpenter Provider in Real-Time
    - Popular Nx Build System Package Compromised with Data-Stealing Malware

Worth a read...

- Death by a Thousand Slops
  - How is AI affecting CVE reporting and PRs
- Adnan Khan and John Stawinski blogs
  - Security researchers focused on supply chain attacks

Final Thoughts...

- You are going to start verifying Red Hat content... right?
- But what's the health of the other artifacts you are pulling into your ecosystem, such as Java/Python/npm/Container Bases/etc?
- If you are generating metadata (i.e.: SBOMS); are they signed and verifiable?
- If you are already signing; is the *process* tamper-proof?

Technology illustrations by Storyset

Questions?

Red Hat Summit

Connect

Thank you

linkedin.com/company/red-hat

facebook.com/redhatinc

youtube.com/user/RedHatVideos

twitter.com/RedHat

Red Hat Summit

Connect

Trust, but Verify: Verifying Red Hat Container Images to Assure Platform Trust

Gareth Healy

Dan Hawker