



Daily Technology Watch

Technology

30 September 2026 – Cadence: Daily (Monday to Friday)

Summary & Topic of the Day

Software Supply Chain Security on OpenShift: Tekton Chains, Sigstore (RHTAS) and SBOM/VEX

SLSA provenance signed by Tekton Chains, keyless Sigstore signing (Red Hat Trusted Artifact Signer), SBOM/VEX with Trusted Profile Analyzer and Enterprise Contract (Conforma) release policy: the OpenShift build chain as the trust boundary.

1. Scope

Watch family: 2. Jobs, Skills & Technologies.

Run cadence: Daily (Monday to Friday at 08:30).

Distribution: [Aperta Web Radar](#) — [LinkedIn Aperta Scientia](#) — [X @ApertaScientia](#).

2. Topic of the Day — Article Published on the Web Radar

Software Supply Chain Security on OpenShift: Tekton Chains, Sigstore (RHTAS) and SBOM/VEX

SolarWinds (2020), Log4Shell (CVE-2021-44228), the XZ Utils backdoor (CVE-2024-3094) and the typosquatting campaigns on npm and PyPI all converge on the same engineering finding: **the artefact executed in production is no longer the code you believe you wrote**. The question is no longer "is my application vulnerable?" but "can I prove where this binary came from, who built it, with which dependencies, and that it stayed intact between the commit and the Pod?". That is precisely the role of a tooled software supply chain. On OpenShift the assembly is now mature: **OpenShift Pipelines (Tekton)** for declarative execution, **Tekton Chains** for signed SLSA provenance, **Sigstore / Red Hat Trusted Artifact Signer (RHTAS)** for keyless signing, **Red Hat Trusted Profile Analyzer (TPA)** for SBOMs and VEX, and **Enterprise Contract (now Conforma)** as the release gate.

The pipeline as the trust boundary: Pipelines-as-Code

The first break with legacy practice is to remove all image builds from workstations:

- **Containerised execution by construction:** a Tekton Task runs in a dedicated Pod, with its own ServiceAccount, Workspaces and resource limits. There is no longer any untraceable local `docker build` or floating `latest` tag on a laptop.
- **Pipelines-as-Code (PaC):** PipelineRun definitions live in the application repository, under `.tekton/`, and are triggered by Git webhooks (push, pull request) through the Repository CRD and CEL event filtering. The pipeline becomes an artefact reviewed in a *pull request*, just like code.
- **Native traceability:** every run produces an immutable, timestamped TaskRun/PipelineRun referencing the



exact builder image (buildah, git-clone), the parameters and the commit SHA. This is the raw material of SLSA provenance.

Field caveat: without a *pinning* policy on `taskRef` (versioned OCI bundles, digests rather than tags), a "secure" pipeline can execute an upstream-modified Task. That is a direct attack vector on the build chain.

SLSA provenance signed by Tekton Chains

Tekton Chains is a controller that watches the cluster's `TaskRun` and `PipelineRun` objects and turns their results into **in-toto** provenance attestations in the **SLSA v1.0** format:

- **Automatic attestation:** Chains captures the inputs (base images, sources, parameters) and the outputs (image digests, artefacts), then produces a signed provenance document with no developer intervention.
- **Signing backends:** *keyless* x509 (OIDC → Fulcio, short-lived certificate), managed KMS (AWS KMS, GCP KMS, HashiCorp Vault) or a Kubernetes key. This choice determines the achievable SLSA level.
- **OCI publication:** by default the attestation and its signature are pushed to the registry as OCI artefacts in the format expected by `cosign`, alongside the matching transparency-log entry. The image then carries the annotation `chains.tekton.dev/signed: "true"`.
- **SLSA reading:** L1 = provenance generated; L2 = hosted build + signed provenance; L3 = isolated build, non-forgable provenance. Tekton Chains on OpenShift provides the L2/L3 building blocks, provided the builder isolation is hardened.

Keyless signing: Sigstore and Red Hat Trusted Artifact Signer

The Sigstore model solves the historical problem of managing signing keys (issuance, rotation, revocation):

- **Fulcio** issues a very short-lived ephemeral x509 certificate, bound to the OIDC identity of the operator or workload.
- **Rekor** records every signature in an append-only transparency log with a Merkle tree: proof of existence is public and verifiable, and compromise becomes detectable.
- **RHTAS** is Red Hat's supported enterprise distribution of Sigstore (Fulcio, Rekor, Trillian, TUF for trust-root distribution, `cosign`, `gitsign`, and the `ec` binary). `gitsign` lets you sign Git commits through simple OIDC authentication, with no local key.

Command-line verification:

```
cosign verify ghcr.io/org/app@sha256:<digest> \
--certificate-identity-regexp '^https://github.com/org/.*' \
--certificate-oidc-issuer https://token.actions.githubusercontent.com
```

Admission enforcement: the *Sigstore Policy Controller* (`ClusterImagePolicy` CRD) rejects any deployment whose image is not signed by an approved identity and logged in Rekor. RHACS also offers Cosign signature verification in its admission controller. Compliance is no longer a report, it is a **deny**.

SBOM and VEX: Red Hat Trusted Profile Analyzer

An SBOM is only useful if it is actionable:

- **Formats:** SPDX 2.3 and CycloneDX 1.6 now cover virtually all generation tools; the content must be attached to the image (attestation) rather than stored off-chain.



- **TPA (Trustify)** centralises the SBOMs produced by your builds *and* those supplied by vendors, then continuously correlates them with vulnerability databases and security advisories (including Red Hat's).
- **VEX (Vulnerability Exploitability eXchange)** is the missing piece: a component may be present in the image without being reachable (`vulnerable_code_not_in_execute_path`) or even compiled (`vulnerable_code_not_present`). Without VEX, a team drowns its developers in non-exploitable CVEs; with VEX, it prioritises what is genuinely in the attack surface and answers factually "*what is the blast radius of a new CVE?*".
- **PURL-based querying:** correlation by `pkg:npm/...`, `pkg:maven/log4j-core@2.14.1` lists in seconds every affected image, including in remote registries.

Enterprise Contract (Conforma): policy as the release gate

Verification must be blocking and identical everywhere:

- **ec validate image / ec verify** runs in the pipeline before any promotion, and refuses the release if a rule fails.
- **Typical rules:** signature verified, SLSA provenance of a minimum level, SBOM present and attached, no critical-severity CVE, and an allow-list of permitted Tekton Task bundles.
- **Rego/OPA policies packaged as OCI bundles:** versioned, reviewed, incrementally distributable — from the developer workstation (shift-left) all the way to cluster admission and the registry. The same rules can be reused by Quay (immutability, Clair scanning, signing) to lock down the registry.

Complete reference chain

1. Signed commit (gitsign, OIDC identity).
2. PaC trigger on a Git webhook → versioned `PipelineRun`.
3. Image build (Buildah, rootless) + vulnerability scan.
4. SBOM and attestation generation.
5. Tekton Chains → signed SLSA provenance (Fulcio) + Rekor entry + OCI push to Quay.
6. `ec validate image`: blocking release gate.
7. ClusterImagePolicy admission: rejection of any unsigned or non-compliant artefact.
8. Continuous SBOM/CVE/VEX correlation in TPA.

3. Impact on Aperta Scientia Curricula

AS200 Track — DevOps Platform Engineer

This watch directly feeds into the platform engineering and software supply chain security modules:

- **Builds & CI/CD (D0288)** : Designing declarative Tekton pipelines, managing rootless container builds, and automating workflows via Pipelines-as-Code.
- **Operations & Hardening (D0380)** : Multi-cluster management, delivery pipeline observability, and enterprise-wide compliance policies.



- **Container & Platform Security (D0430)** : Implementing admission enforcement (Sigstore Policy Controller, RHACS), keyless artifact signing (RHTAS), SBOM generation, and automated exploitability triage (TPA / VEX).
- **Target Certifications** : Practical preparation for **EX188** (Containers), **EX280** (OpenShift Administration), and **EX288** (OpenShift Development).

AS300 Track — AI Platform Engineer

Supply chain security principles are extended to inference infrastructure and MLOps pipelines:

- **GPU & ML Dependency Traceability** : Provenance auditing and SLSA verification for base images embedding CUDA, ROCm, and PyTorch runtimes.
- **Model Signing & Admission Control** : Cryptographic signing of model weights and vLLM / KServe serving runtimes before cluster deployment.
- **AI Governance & SBOM** : Automated license inventory and continuous vulnerability scanning across the entire MLOps stack (KubeFlow Pipelines, Ray, Triton).

4. Sources

- Red Hat Developer — Red Hat Trusted Software Supply Chain is now available — [Link](#)
- Red Hat Developer — Trusted Software Supply Chain — [Link](#)
- Red Hat Developer — Trusted Artifact Signer (RHTAS) — [Link](#)
- Red Hat Developer — Trusted Profile Analyzer (TPA) — [Link](#)
- Tekton Chains — Documentation — [Link](#)
- Sigstore — Documentation — [Link](#)
- SLSA — Supply-chain Levels for Software Artifacts — [Link](#)
- Enterprise Contract / Conforma — [Link](#)
- CycloneDX / SPDX — [Link](#), [Link](#)

Contacts & Community:

- Academic Programs: learning@apertascientia.io
- Disability Officer & Student Care: care@apertascientia.io
- X (Twitter): [@ApertaScientia](https://twitter.com/ApertaScientia)
- Discord Community: discord.gg/eDJzR5Vk85