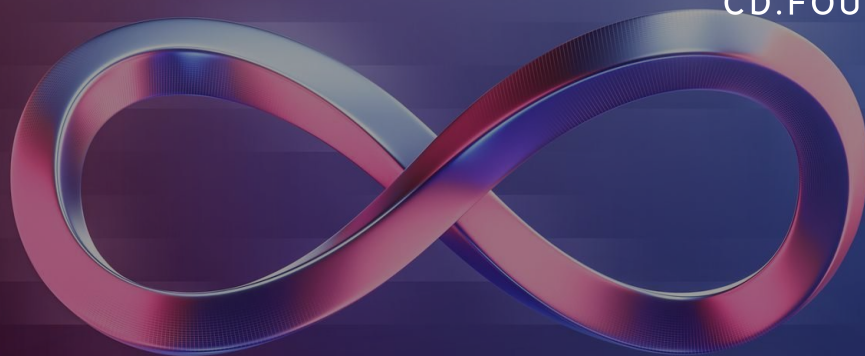


Securing Your CI/CD Pipeline From Code to Deployment

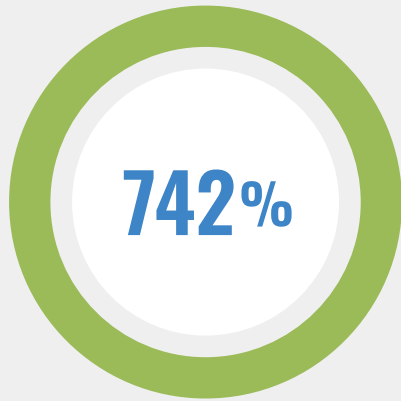
Presented by Steve Taylor,
CTO DeployHub
Ortelius Architect



CD.FOUNDATION



Software Security is Complex



The astonishing growth rate of malicious supply chain attacks.

Source: State of the Software Supply Chain - Sonatype



Boards that consider cybersecurity a business risk.

Source: Gartner



Companies seeking more 'log' visibility into application security.

Source: McKinsey & Company

New Tools, New Pipeline Phases to Secure Software

- The OpenSSF, CD.Foundation, CNCF, and security tool vendors have worked to address the issue of software security with new programs and open-source tooling.
- There are 5 phases of the DevOps pipeline where security tooling can be easily added.
- From code signing to cataloging the data, this roadmap will point you in the correct direction for hardening your DevOps pipeline against cyber attacks.



Jenkins

[Jenkins.io](https://jenkins.io)

The leading open source automation server, provides hundreds of plugins to support building, deploying and automating any project.

Tekton

[Tekton.dev](https://tekton.dev)

Tekton is a powerful and flexible open-source framework for creating CI/CD systems.

Google Cloud Build (Google – CDF Member)

cloud.google.com/build

Scales with no infrastructure to set up, upgrade, or scale. Run builds in a fully managed environment

Ortelius

[Ortelius.io](https://ortelius.io)

A centralized evidence catalog for publishing DevOp and Security intelligence creating a continuous view of an organization's security profile.

Pysia

[Pysia.io](https://pysia.io)

Pysia is a decentralized package network that enables developers to quickly and easily leverage any package with confidence and transparency.

JFrog FrogBot (JFrog - CDF Member)

github.com/jfrog/frogbot

Scans your pull requests and repositories for security vulnerabilities. You can scan pull requests when they are opened.

Security Scorecard

securityscorecards.dev

Implement Scorecard GitHub Actions to perform a full security audit.

SLSA

slsa.dev

SLSA is a set of incrementally adoptable guidelines for build level supply chain security.

Sigstore Cosign

github.com/sigstore/cosign

Sign and verify software artifacts, such as container images and blobs.

SPDX

spdx.dev

An open standard for communicating software bill of materials information.

OSV

osv.dev

An open, precise and distributed approach to producing and consuming vulnerability information.

Syft (Anchore - OpenSSF Member Company)

anchore.com/opensource/

Generates a Software Bill of Materials (SBOM) from container images and filesystems.



CLOUD NATIVE COMPUTING FOUNDATION

Artifact Hub

artifacthub.io

Web-based application that enables finding, installing, and publishing Kubernetes packages.

Docker BuildX (Docker - CNCF Member)

docs.docker.com/engine/reference/commandline/buildx/

CLI plugin that extends the docker command with the full support of the features provided by Moby BuildKit builder toolkit.

Docker Hub (Docker - CNCF Member)

hub.docker.com

Container Image Library.

Quay (Red Hat CNCF Member)

quay.io/repository

Secure Container Storage.

Trivy (Aqua -CNCF Member)

github.com/aquasecurity/trivy

A container scanner that looks for security issues, and *targets* where it can find those issues.

GitHub

Members of OpenSSF, CNCF, CDF

CodeQL

codeql.github.com

Discovers vulnerabilities across a codebase. Uses semantic code analysis engine that lets you query code as though it were data.

Dependabot

github.com/dependabot

Helps open-source users determine if they are running latest version of dependencies.

GPG

github.com/gpg/gnupg

Creates keys that are used to generate badges to indicate if your commits are verified.

Signed-off-by

dev.to/janderssonse/git-signoff-and-signing-like-a-champ-41f3

Verifies who authored the commit under certain conditions, or that you are passing on something which has been authored.

Actions

github.com/features/actions

Makes it easy to automate all your software CI/CD workflows. Build, test, and deploy your code right from GitHub.

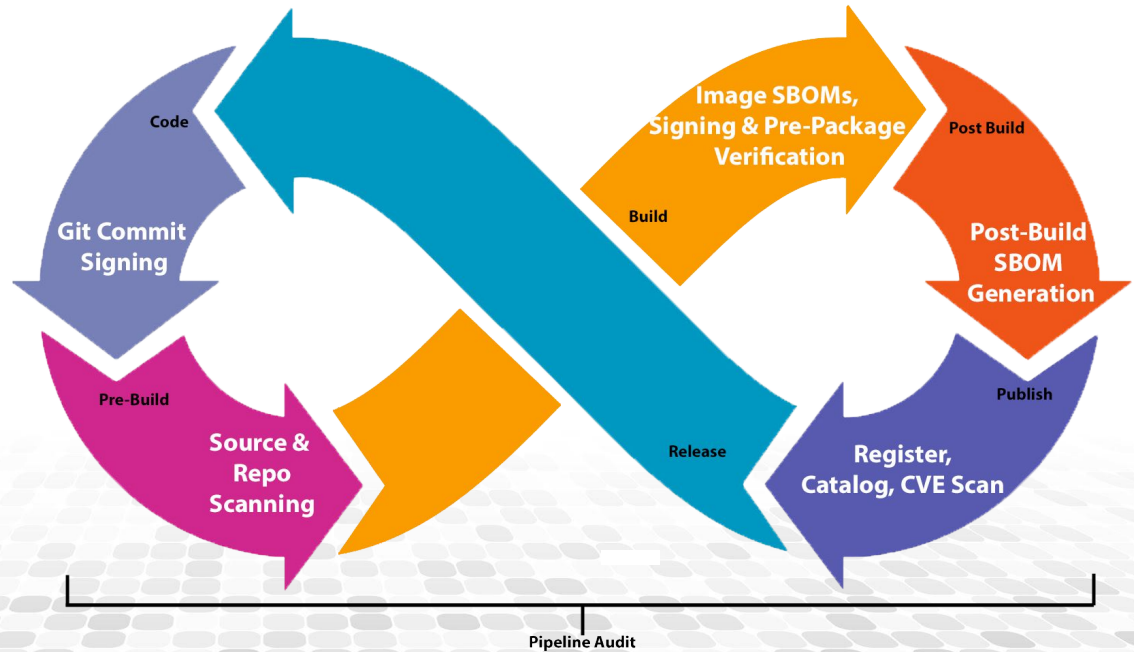
Microsoft SBOM Tool

<https://github.com/microsoft/sbom-tool>

Scans your pull requests and repositories for security vulnerabilities. You can scan pull requests when they are opened.

Application Security, as it relates to the DevOps Pipeline, should be implemented in 5 phases:

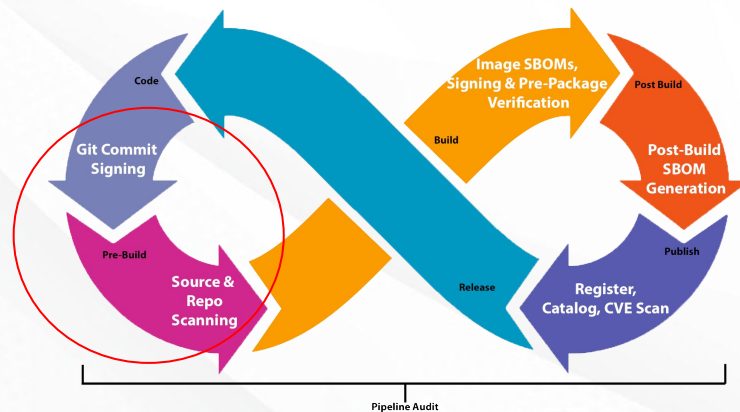
- 1) Code and Pre-Build
- 2) Build
- 3) Post Build (if needed)
- 4) Publish
- 5) Audit



Phase 1 - Code and Pre Build

Critical security steps include:

- code signing
- scanning individual files for code weaknesses
- scanning an entire code base.



Tools to consider:

Git Commit Signing Open-Source Tools

- [GitHub Signing](#)
- [GitLab Signing](#)
- [BitBucket](#)

Repo Security Scanning Tools

- [GitHub CodeQL](#)
- [AquaSec Trivy](#)
- [Dependabot](#)
- [FrogBot](#)

Open-Source SCA Code Scanning Tools

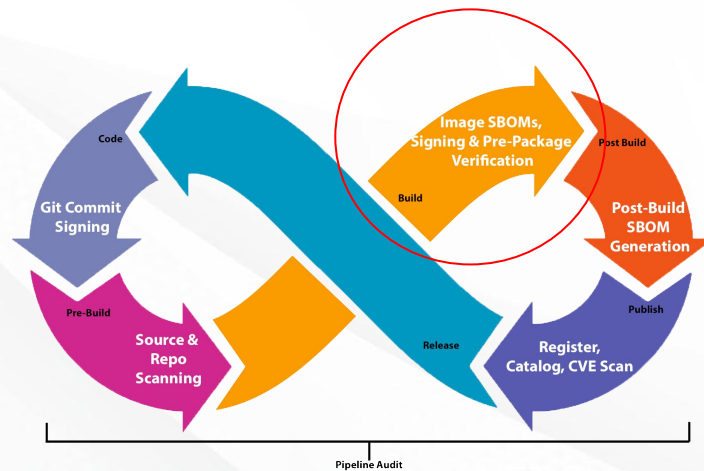
- [Veracode](#)
- [SonarQube](#)

Note: For a comprehensive list of free, commercial, and open-source SCA tooling, check out [Source Code Analysis Tools by OWASP](#).

Phase 2 - Build

These actions include:

- generating an image SBOM
- image signing
- Pre-package verification



Tools to consider:

Open-Source Image SBOM Tools

- [Apko](#)
- [Docker BuildX](#)

Open-Source Package Verification Tools

- [Pyrsia.io](#)

Open-Source Build Signing Tools

- [sigstore.dev](#)
- [Notary](#)

Hosted Build Systems

- [Google Cloud Build](#)
- [GitHub](#)
- [Tekton](#)
- [Jenkins](#)

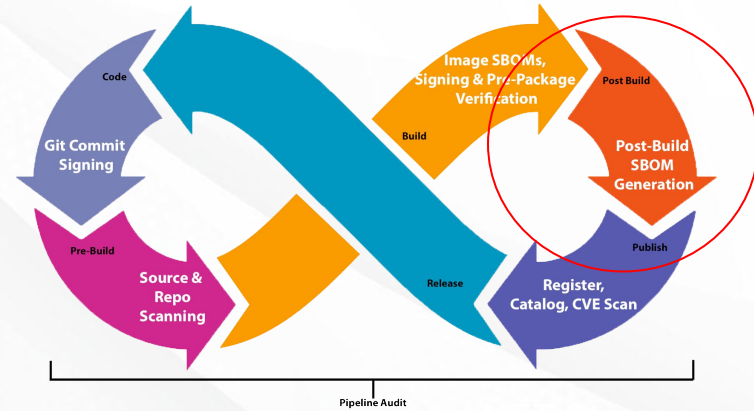
Phase 3 - Post Build SBOM

If the build step in Phase 2 does not include creating an SBOM image, a post-build effort is needed to add security actions for generating the SBOM for the build.

Tools to consider:

- [Anchore Syft](#)
- [Microsoft SBOM Tool](#)
- [OpenSSF SPDX](#)

Open-Source Post Build SBOM tools



Phase 4 - Store the Evidence

This phase includes:

- register containers
- collect security evidence to show an organization's security profile
- discover CVEs

Tools to consider:

Open Source Registries

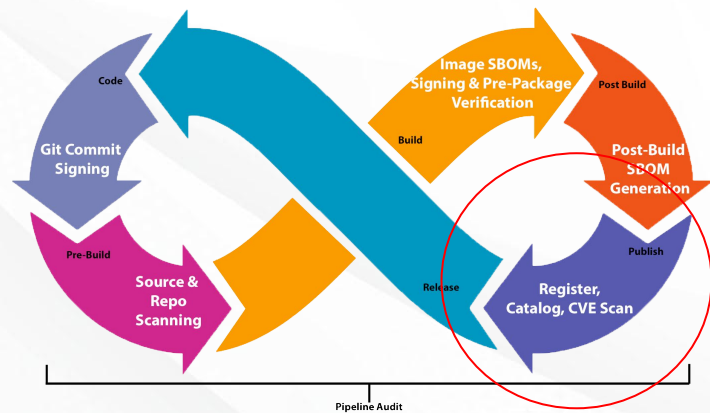
- [ArtifactHub \(OCI\)](#)
- [DockerHub \(OCI\)](#)
- [Quay \(OCI\)](#)
- [Maven Central](#)
- [NPM JS](#)
- [Pypi](#)

Open-Source Evidence Catalogs

- [Ortelius.io](#)

CVE Databases

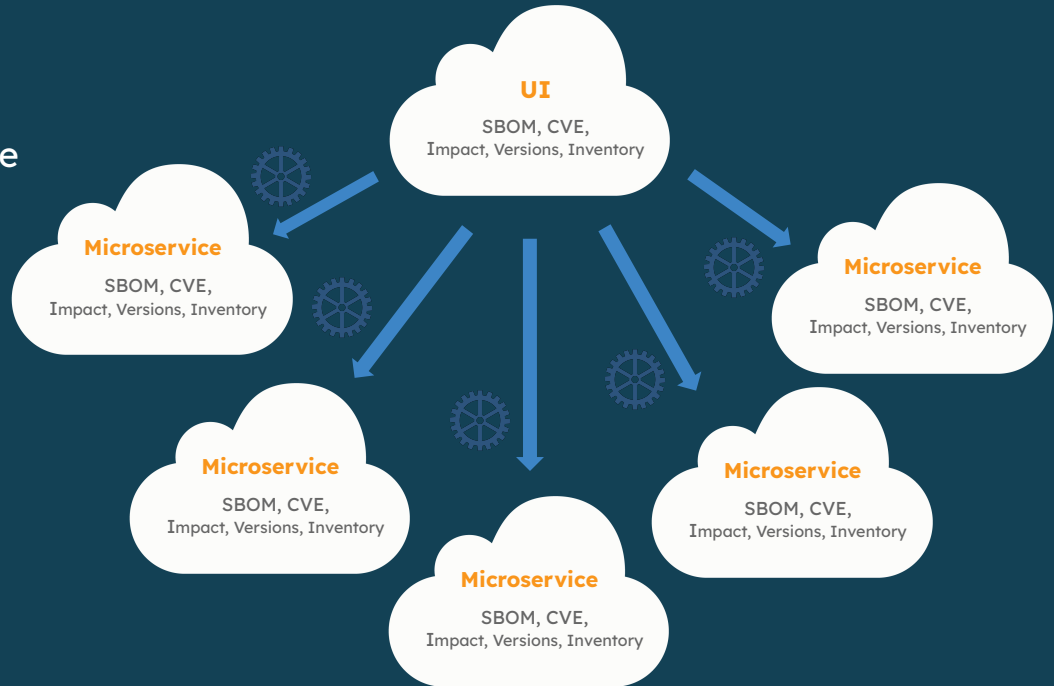
- <https://www.cvedetails.com/>
- <https://github.com/advisories>
- <https://www.cisa.gov/known-exploited-vulnerabilities-catalog>
- osv.dev
- <https://nvd.nist.gov/>
- <https://cve.mitre.org/>



The Importance of Publishing - Phase 4

Security and DevOps Data is trapped across siloed containers and pipelines making it hard to see a comprehensive picture of:

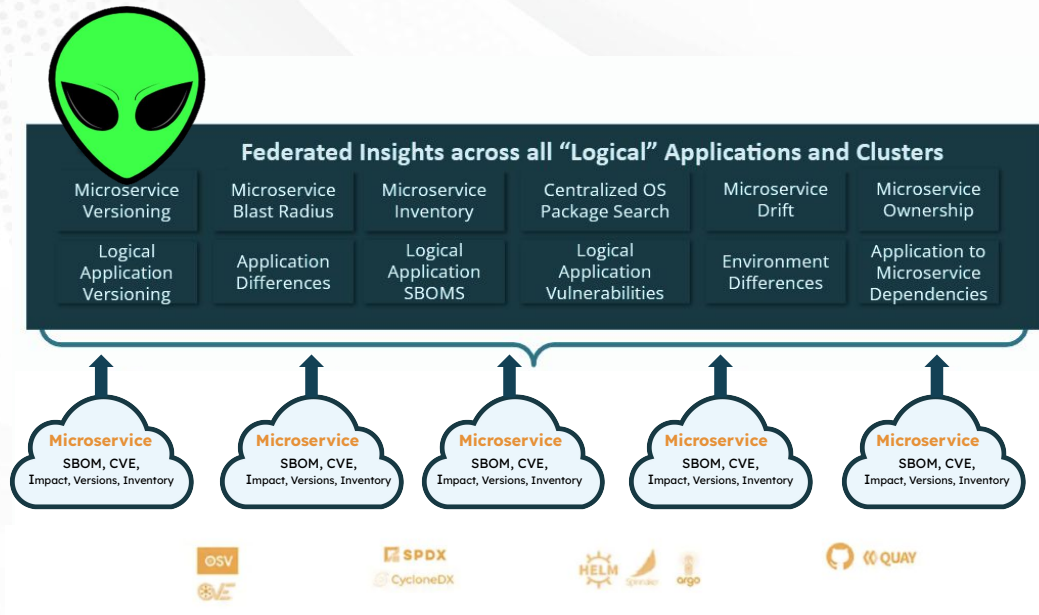
- Software Bill of Materials (BOM)
- CVE Reports
- Release Versions
- Change Tracking
- Application Impact Analysis



Publishing With Ortelius

Ortelius gathers and aggregates critical, security and DevOps insights across your organization.

- Capture actionable insights in minutes versus days.
- Continuously expose non-compliant services to improve application security.
- Improves site reliability response by as much as 50%.



Ortelius is Incubating at the CDF

Continuous Delivery Foundation

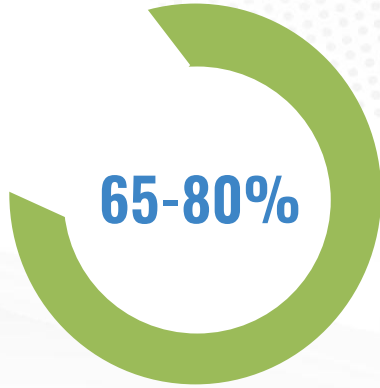


CD.FOUNDATION



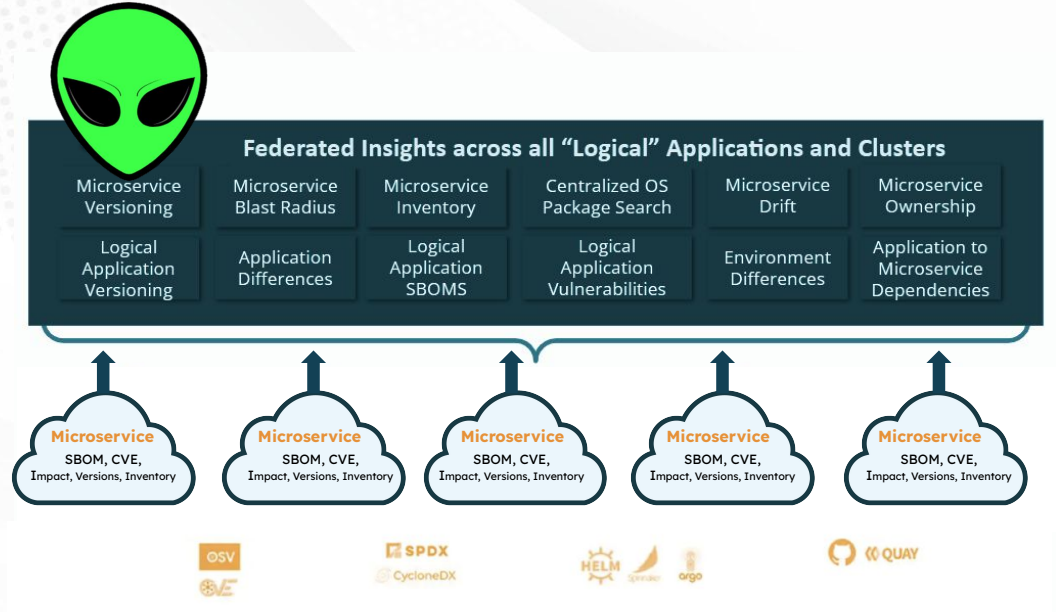
The CDF is part of the Linux Foundation.

Ortelius Addresses “Log Visibility”



Companies seeking more ‘log’ visibility into application security.

[Source: McKinsey & Company](#)



You Have the Data - Make It Actionable with Ortelius

Centralize
all security,
DevOps, and
SCA data.

View open
source package
usage across the
organization.

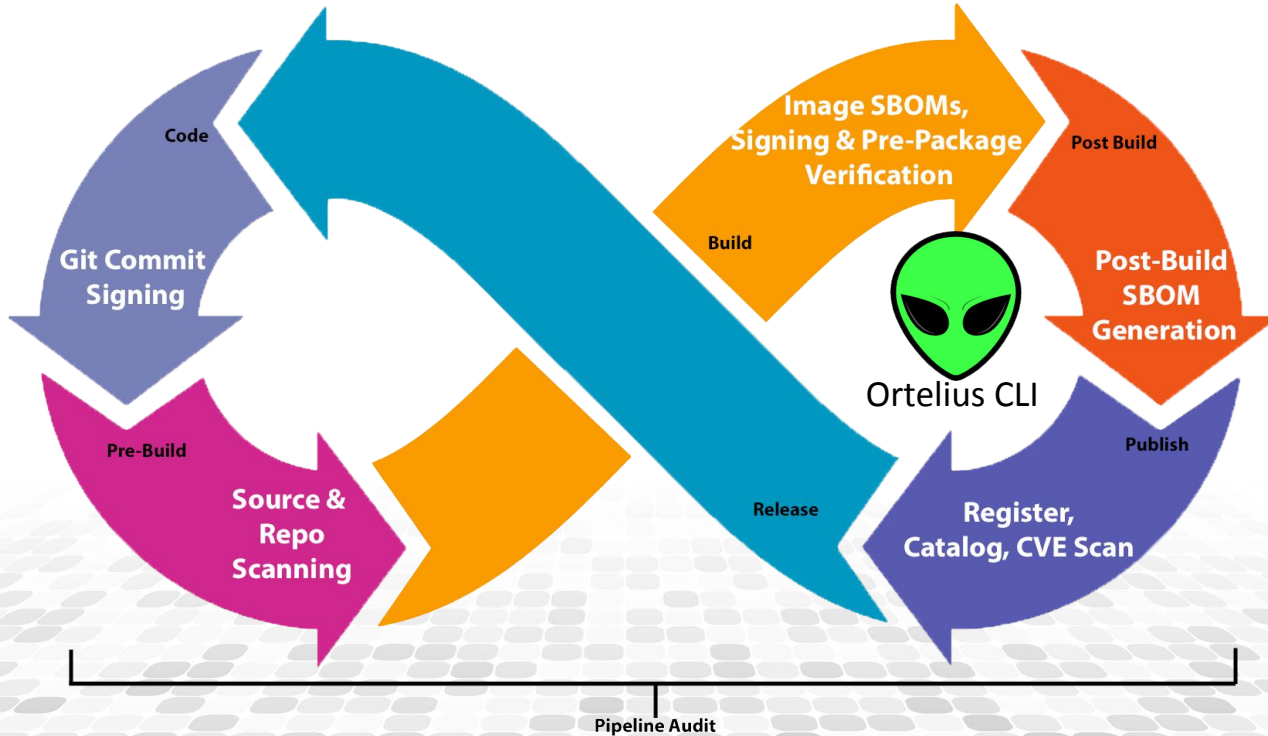
Version
microservices
each time their
composition
changes.

View the impact a
single service has
to all consuming
logical
applications.

Assign release
numbers to
'logical'
applications as
services change.

Track
microservice
versions and
usage across all
clusters.

Automate Evidence Collection with the Ortelius CLI



Actionable Evidence - Application Level SBOMs

In a Decoupled Microservices Environment

The screenshot displays the ORTELIUS application security tool interface. The top navigation bar includes the ORTELIUS logo, a user profile icon, and a status indicator "7 of 7 Reverse Proxy running". A left sidebar contains navigation options: Applications, Components, Domains, Environments, Endpoints, Actions, Func/Procs, Customize Types, and Setup.

The main content area is divided into two panels:

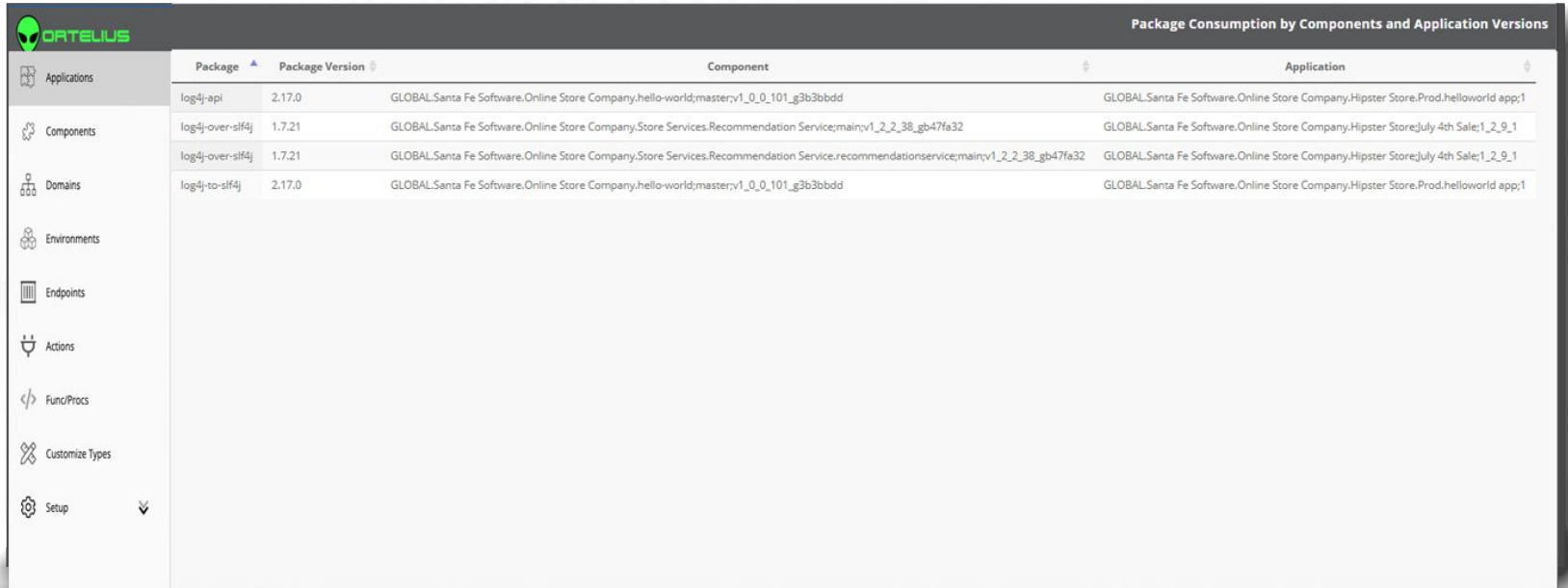
- Vulnerabilities:** A table listing identified vulnerabilities. The table has columns for Package, Version, ID, Summary, and Component. Two vulnerabilities are highlighted in blue, and one is highlighted in red.
- Software Bill of Materials (SBOM):** A table listing the components of the application. The table has columns for Package, Version, License, and Component. The first row is highlighted in blue.

Package	Version	ID	Summary	Component
hibernate-validator	5.2.4.Final	GHSA-cxgp-cffc-3u8c	CVE-2017-7536 : Privilege Escalation in Hibernate Validator	GLOBAL.Santa Fe Software.Online Store Company.Store Services.Recommendation Service.recommendationservice;main/v1_2_2_36_g178b682
hibernate-validator	5.2.4.Final	GHSA-cxgp-cffc-3u8c	CVE-2017-7536 : Privilege Escalation in Hibernate Validator	GLOBAL.Santa Fe Software.Online Store Company.Store Services.Recommendation Service.recommendationservice;main/v1_2_2_36_g178b682
jackson-databind	2.8.1	GHSA-z88c-cq4h-98gq	CVE-2020-25649 : XML External Entity (XXE) Injection in Jackson Databind	GLOBAL.Santa Fe Software.Online Store Company.Store Services.Recommendation Service.recommendationservice;main/v1_2_2_36_g178b682
jackson-databind	2.8.1	GHSA-4gq5-ch57-c2mg	CVE-2018-14719 : High severity vulnerability that affects com.fasterxml.jackson.core:jackson-databind	GLOBAL.Santa Fe Software.Online Store Company.Store Services.Recommendation Service.recommendationservice;main/v1_2_2_36_g178b682

Package	Version	License	Component
US_export_policy		No License	GLOBAL.Santa Fe Software.Online Store Company.Store Services.Recommendation Service.recommendationservice;main/v1_2_2_36_g178b682
ca-certificates-java		No License	GLOBAL.Santa Fe Software.Online Store Company.Store Services.Recommendation Service.recommendationservice;main/v1_2_2_36_g178b682
charset		No License	GLOBAL.Santa Fe Software.Online Store Company.Store Services.Recommendation Service.recommendationservice;main/v1_2_2_36_g178b682
cidrdata		No License	GLOBAL.Santa Fe Software.Online Store Company.Store Services.Recommendation Service.recommendationservice;main/v1_2_2_36_g178b682
dnsres		No License	GLOBAL.Santa Fe Software.Online Store Company.Store Services.Recommendation

Actionable Evidence - Open Source Package Search

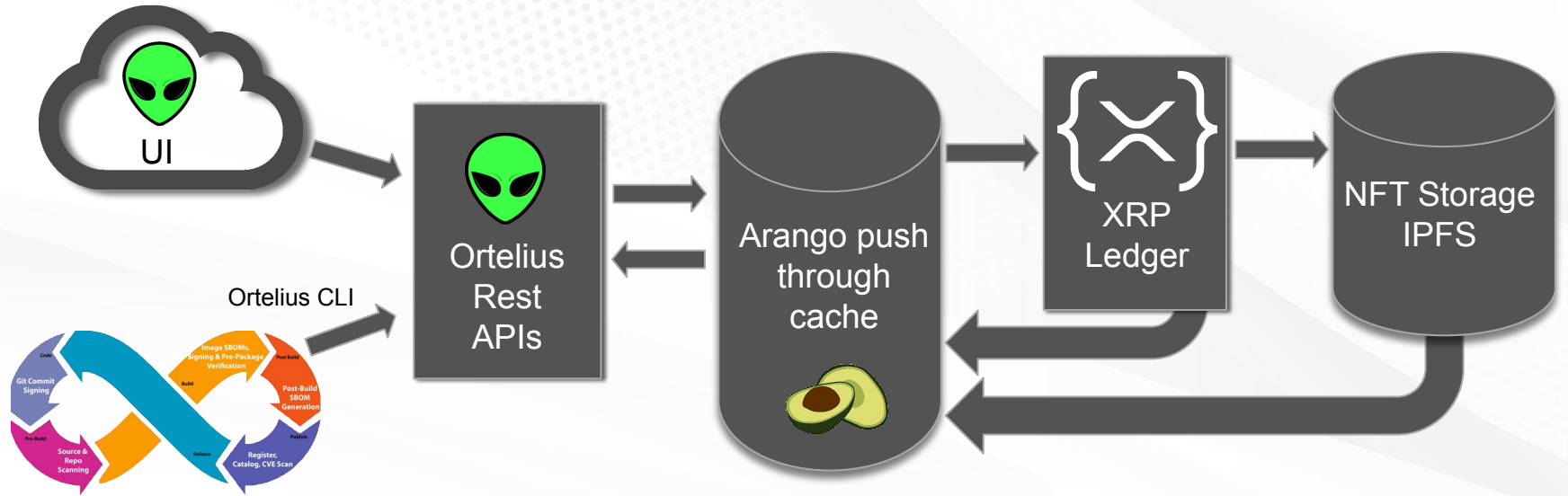
Answer the question “who is using Log4J?”



The screenshot displays the ORTELIUS interface, which is used for analyzing open-source package consumption. The main title is "Package Consumption by Components and Application Versions". The interface includes a sidebar with navigation options: Applications, Components, Domains, Environments, Endpoints, Actions, Func/Procs, Custom Types, and Setup. The main content area shows a table with the following data:

Package	Package Version	Component	Application
log4j-api	2.17.0	GLOBAL.Santa Fe Software.Online Store Company.hello-world;master;v1_0_0_101_g3b3bddd	GLOBAL.Santa Fe Software.Online Store Company.Hipster Store.Prod.helloworld app;1
log4j-over-slf4j	1.7.21	GLOBAL.Santa Fe Software.Online Store Company.Store Services.Recommendation Service;main;v1_2_2_38_gb47fa32	GLOBAL.Santa Fe Software.Online Store Company.Hipster Store;July 4th Sale;1_2_9_1
log4j-over-slf4j	1.7.21	GLOBAL.Santa Fe Software.Online Store Company.Store Services.Recommendation Service.recommendationservice;main;v1_2_2_38_gb47fa32	GLOBAL.Santa Fe Software.Online Store Company.Hipster Store;July 4th Sale;1_2_9_1
log4j-to-slf4j	2.17.0	GLOBAL.Santa Fe Software.Online Store Company.hello-world;master;v1_0_0_101_g3b3bddd	GLOBAL.Santa Fe Software.Online Store Company.Hipster Store.Prod.helloworld app;1

Ortelius Architecture



Learn More by Joining the Ortelius Team



ortelius.io



<https://www.linkedin.com/company/ortelius-open-source/>



@OrteliusOs



Ortelius Open Source GitHub: <https://github.com/ortelius>



Ortelius Discord Channel <https://discord.gg/hRCRYRQZ>

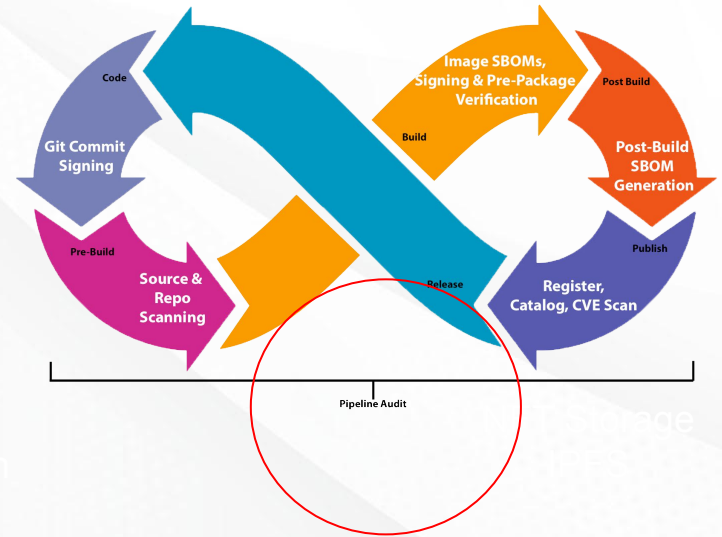


Phase 5 - Pipeline Audit

Beyond adding security to the phases of the pipeline, auditing the pipeline itself further hardens the application life cycle process.

This is a new area of pipeline management. Check out:

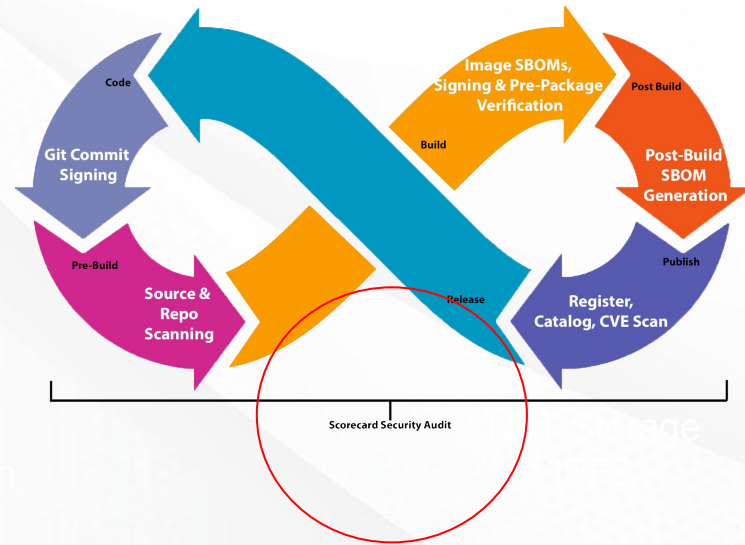
- Jenkins Audit Trail
- Tekton Chains



ScoreCard Security Audit

OpenSSF Scorecard checks for:

- Branch Protection
- CI Tests
- CII Best Practices
- Code-Review
- Contributors
- Dangerous Workflows
- Dependency Update Tool Usage
- License
- Packaging
- Maintained
- Fuzzing
- Pinned Dependencies
- SAST
- Security Policies
- Signed Releases
- Token Permissions
- Vulnerabilities



securityscorecards.dev

Open Source Security Tools Landscape

Code and Pre-Build	Build	Post-Build	Publish
Source Code Scanning <ul style="list-style-type: none">• Veracode• SonarQube	Image SBOM Generation <ul style="list-style-type: none">• Apko• Docker Buildx Hosted Build Systems <ul style="list-style-type: none">• Google Cloud• GitHub Actions• Tekton	Post Build SBOM Generation <ul style="list-style-type: none">• Syft• SPDX• Microsoft SBOM	Registries <ul style="list-style-type: none">• ArtifactHub• DockerHub• Quay• Maven Central• NPM JS• Pypi
Repository Scanning <ul style="list-style-type: none">• CodeQL• Trivy• FrogBot• Dependabot	Signing / Attribution / Provenance <ul style="list-style-type: none">• sigstore• Notary Package Verification <ul style="list-style-type: none">• Pyrsia		Evidence Catalog <ul style="list-style-type: none">• Ortelius

Thank You and Find Me:



<https://www.linkedin.com/in/steve-taylor-oms/>



DeployHub.com



@DeployHubProj



DeployHub



Ortelius Open Source GitHub: <https://github.com/ortelius>



Ortelius Discord Channel <https://discord.gg/hRCRYRQZ>

