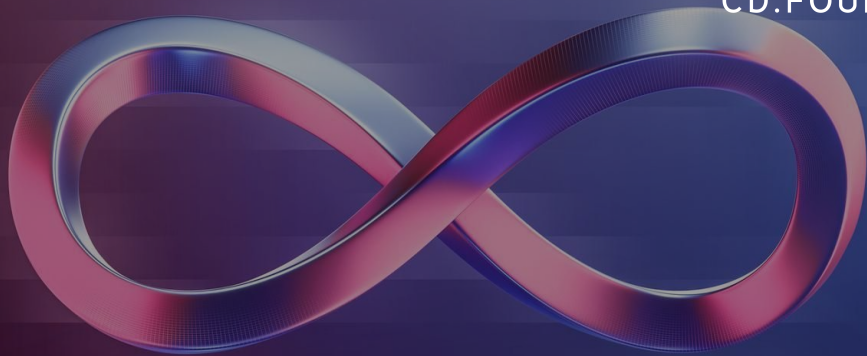


# Securing Your CI/CD Pipeline From Code to Deployment

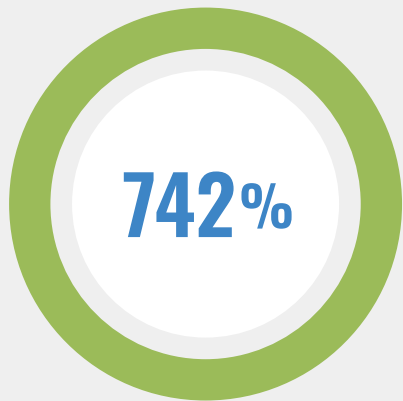
Presented by Steve Taylor,  
CTO DeployHub  
Ortelius Architect



CD.FOUNDATION



# Software Security is Complex



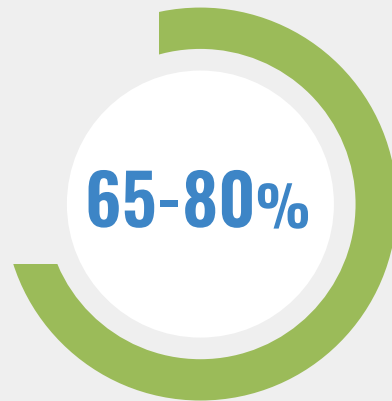
**The astonishing growth rate of malicious supply chain attacks.**

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](https://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](https://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](https://securityscorecards.dev)

Implement Scorecard GitHub Actions to perform a full security audit.

## SLSA

[slsa.dev](https://slsa.dev)

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

## Sigstore Cosign

[github.com/sigstore/cosign](https://github.com/sigstore/cosign)

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

## SPDX

[spdx.dev](https://spdx.dev)

An open standard for communicating software bill of materials information.

## OSV

[osv.dev](https://osv.dev)

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

## Syft (Anchore - OpenSSF Member Company)

[anchore.com/open-source/](https://anchore.com/open-source/)

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



# CLOUD NATIVE COMPUTING FOUNDATION

## Artifact Hub

[artifacthub.io](https://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/](https://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](https://hub.docker.com)

Container Image Library.

## Quay (Red Hat CNCF Member)

[quay.io/repository](https://quay.io/repository)

Secure Container Storage.

## Trivy (Aqua –CNCF Member)

[github.com/aquasecurity/trivy](https://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](https://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](https://github.com/dependabot)

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

## GPG

[github.com/gpg/gnupg](https://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](https://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](https://github.com/features/actions)

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

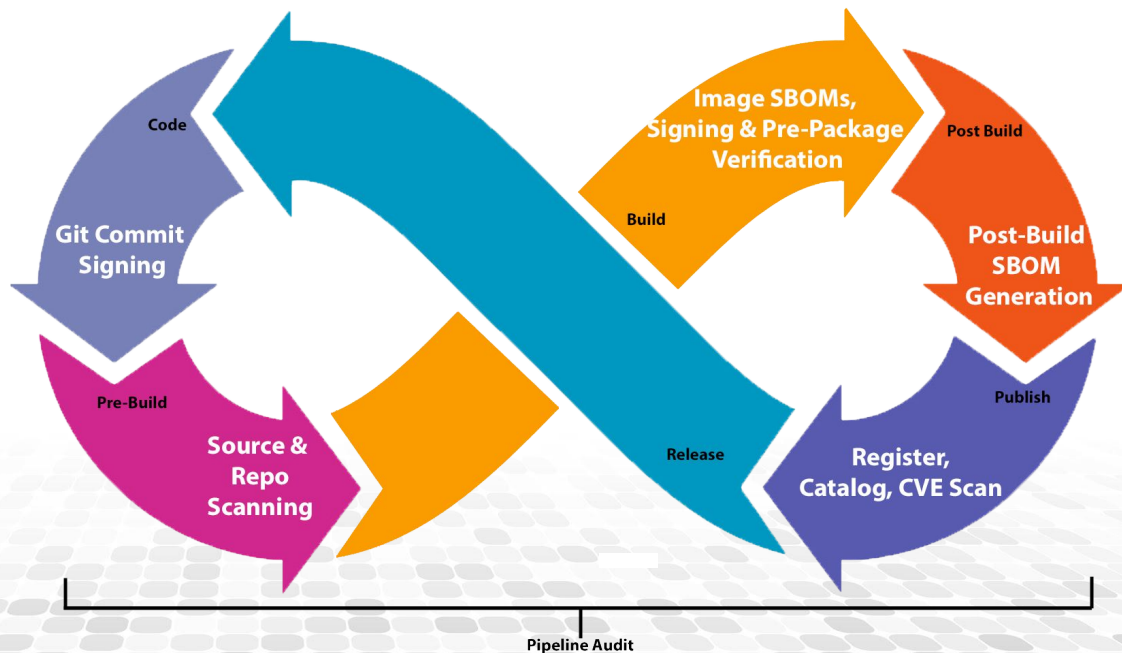
## Microsoft SBOM Tool

<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

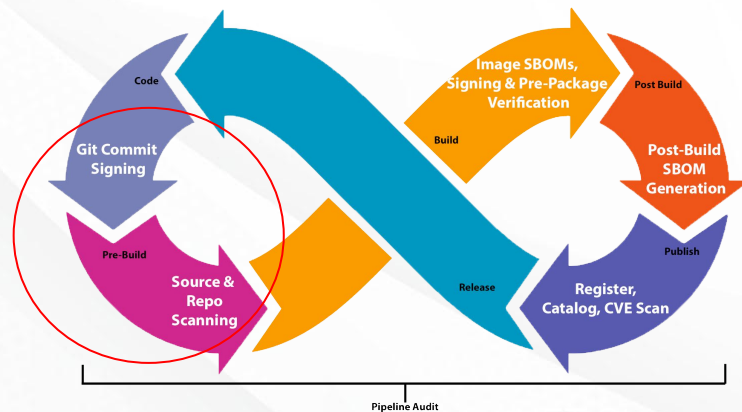




# Phase 1 - Code and Pre Build

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

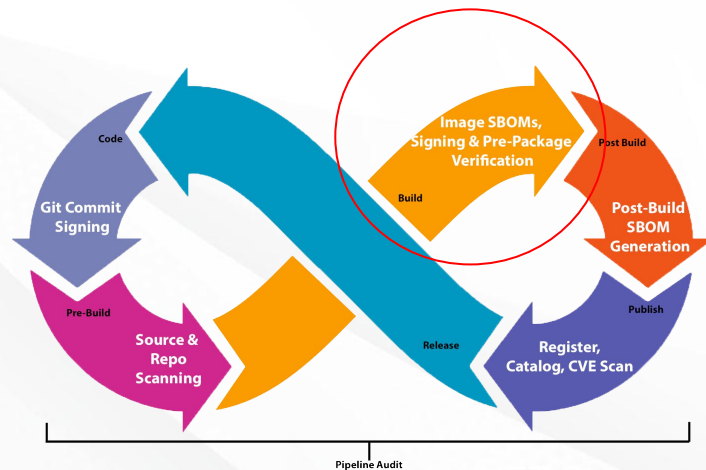
- [Veracode](#)
- [SonarQube](#)

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 Build Signing Tools

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

### Open-Source Package Verification Tools

- [Pyrsia.io](#)

### Hosted Build Systems

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

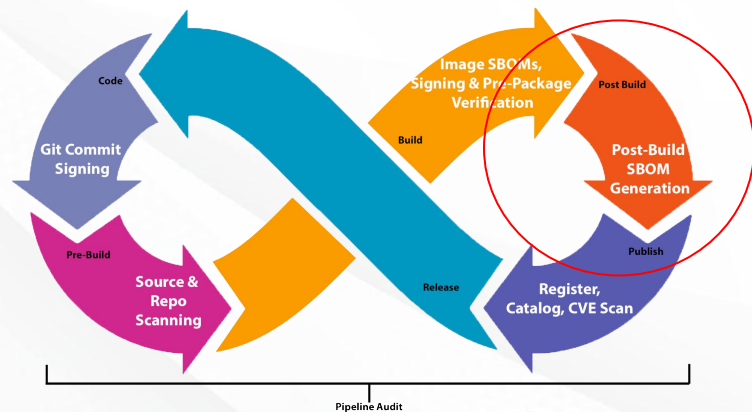
## Phase 3 - Post Build SBOM

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

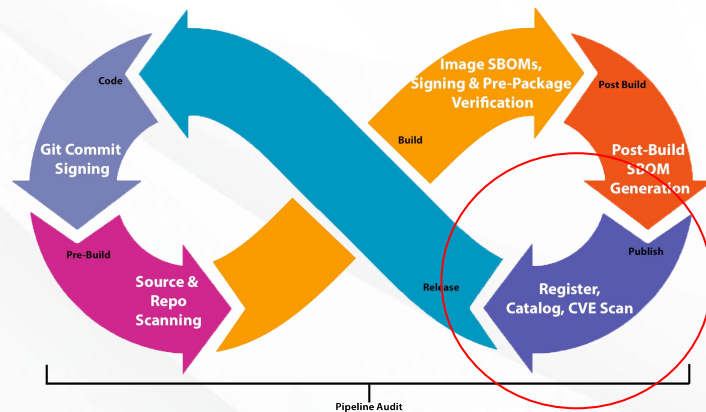
- [ArtifactHub \(OCI\)](#)
- [DockerHub \(OCI\)](#)
- [Quay \(OCI\)](#)
- [Maven Central](#)
- [NPM JS](#)
- [Pypi](#)

### Open-Source Evidence Catalogs

- [Ortelius.io](#)

### CVE Databases

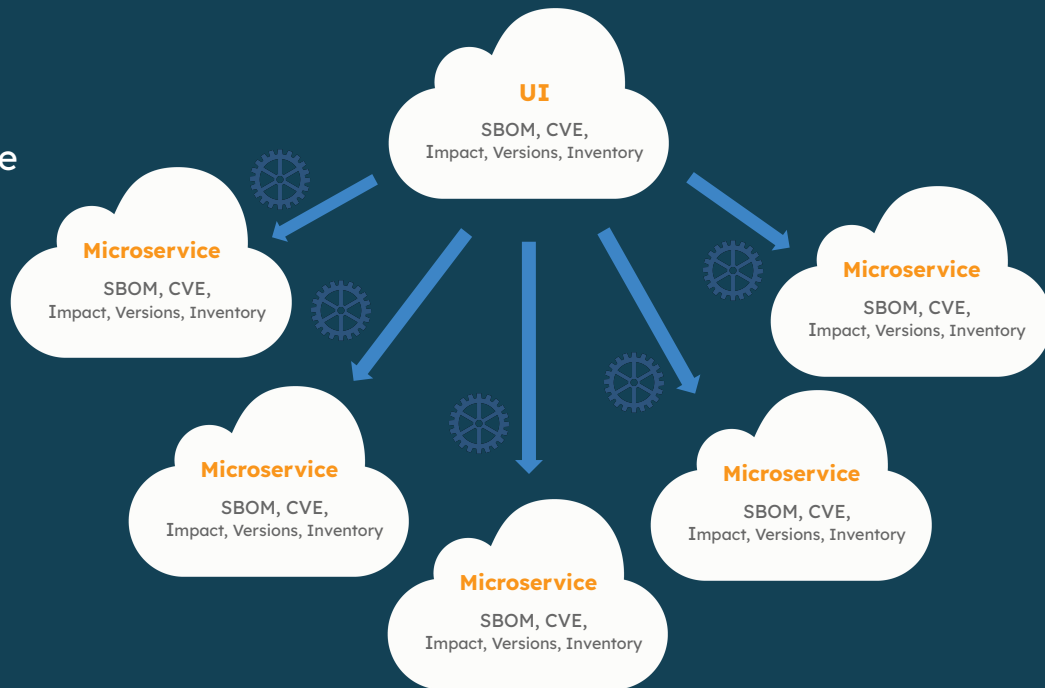
- <https://www.cvedetails.com/>
- <https://github.com/advisories>
- <https://www.cisa.gov/known-exploited-vulnerabilities-catalog>
- [osv.dev](https://osv.dev)
- <https://nvd.nist.gov/>
- <https://cve.mitre.org/>



# 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:

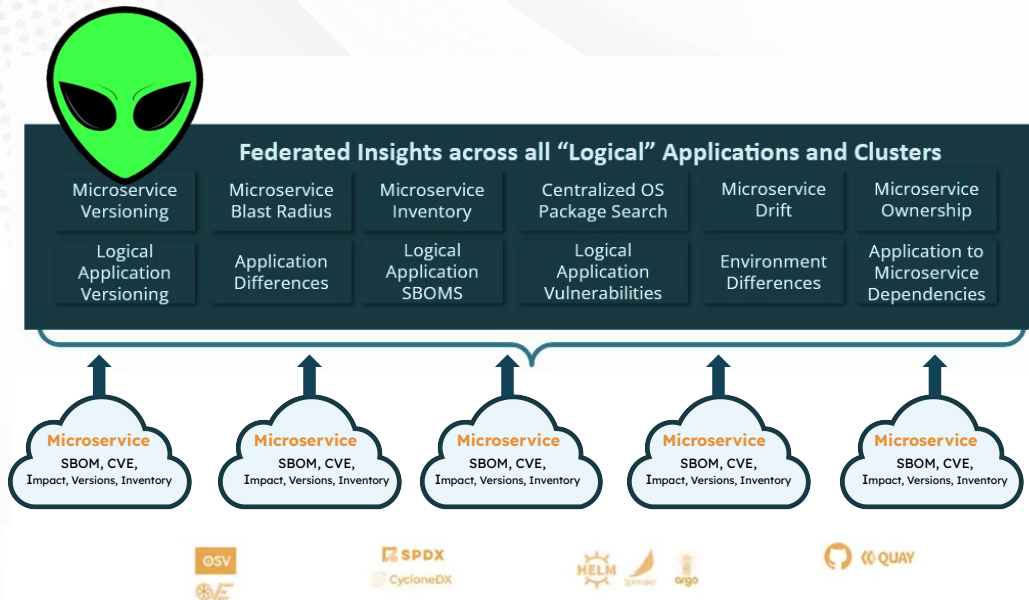
- Software Bill of Materials (BOM)
- CVE Reports
- Release Versions
- Change Tracking
- Application Impact Analysis



# 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



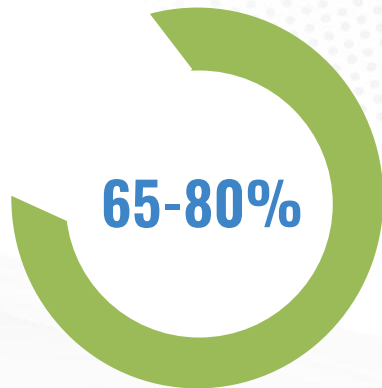
CD.FOUNDATION



The CDF is part of the Linux Foundation.

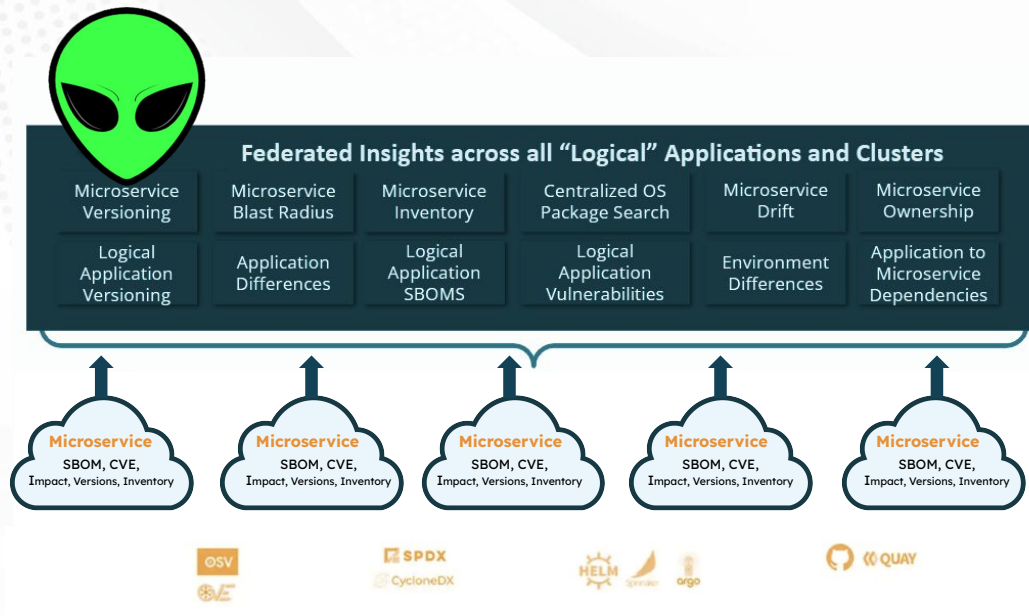


# Ortelius Addresses “Log Visibility”



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.

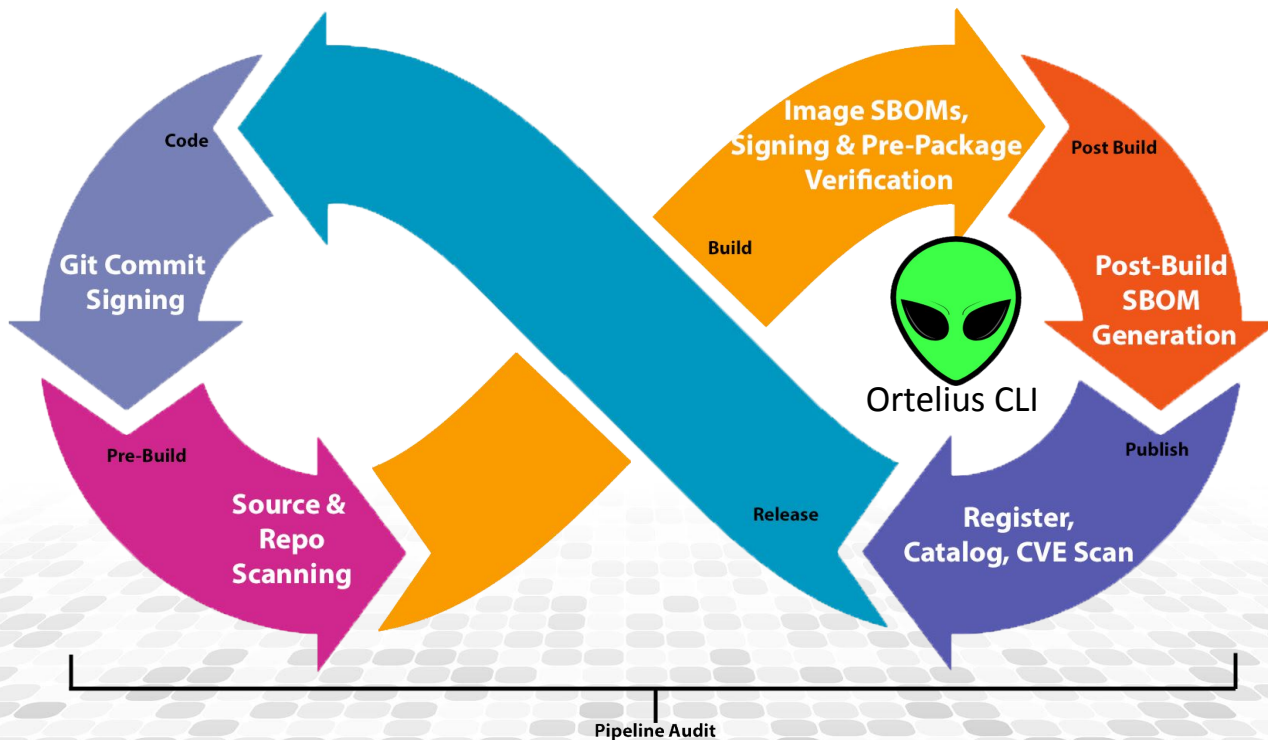
Version  
microservices  
each time their  
composition  
changes.

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 bar shows the ORTELIUS logo and the status "7 of 7 Reverse Proxy running". The left sidebar contains navigation icons for Applications, Components, Domains, Environments, Endpoints, Actions, Func/Procs, and Customize Types, with a Setup icon at the bottom.

The main content area is divided into two panels:

### Vulnerabilities

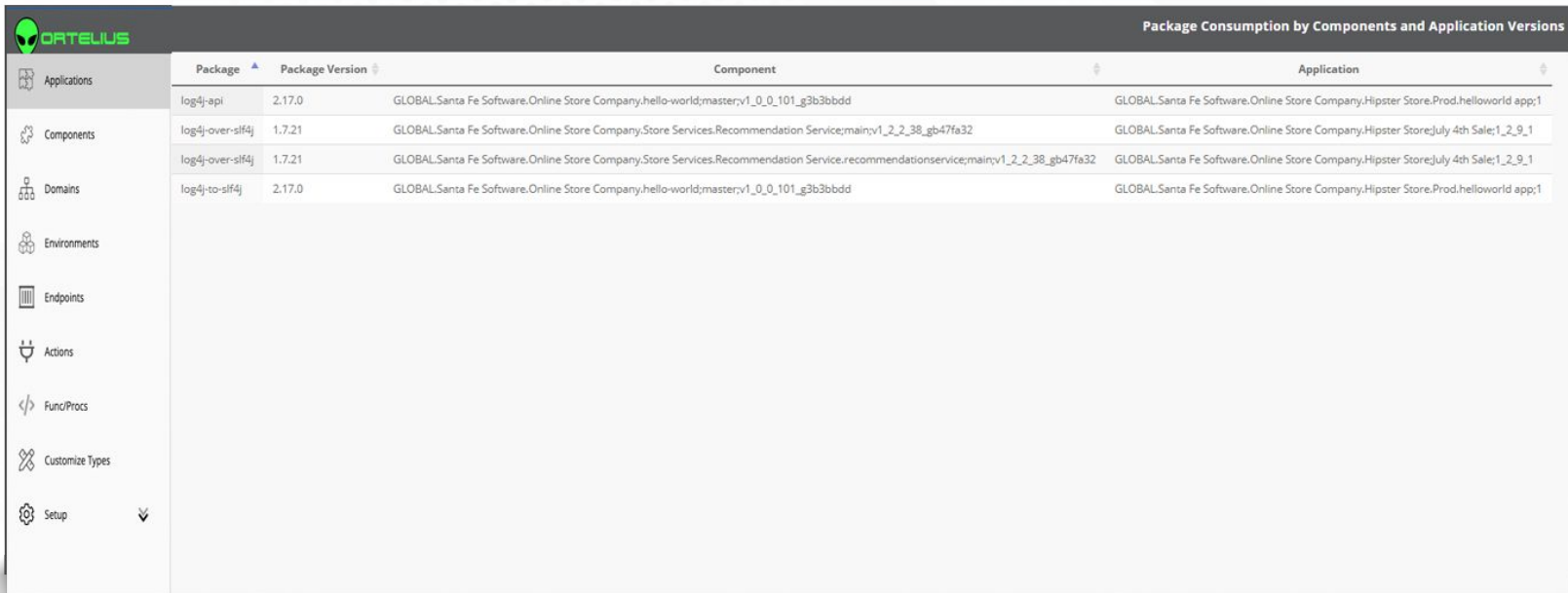
Package	Version	ID	Summary	Component
hibernate-validator	5.2.4.Final	<a href="#">GHSA-cxgp-pcfc-3u8c</a>	CVE-2017-7536 : Privilege Escalation in Hibernate Validator	GLOBAL.Santa Fe Software.Online Store Company.Store Services.Recommendation Service.recommendationservice;main/v1_2_2_36_g178b682
hibernate-validator	5.2.4.Final	<a href="#">GHSA-cxgp-pcfc-3u8c</a>	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	<a href="#">GHSA-z88c-cq4h-98gq</a>	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	<a href="#">GHSA-4g05-ch57-c2mg</a>	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

### Software Bill of Materials (SBOM)

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
dissect		No License	GLOBAL.Santa Fe Software.Online Store Company.Store Services.Recommendation Service.recommendationservice;main/v1_2_2_36_g178b682

# Actionable Evidence - Open Source Package Search

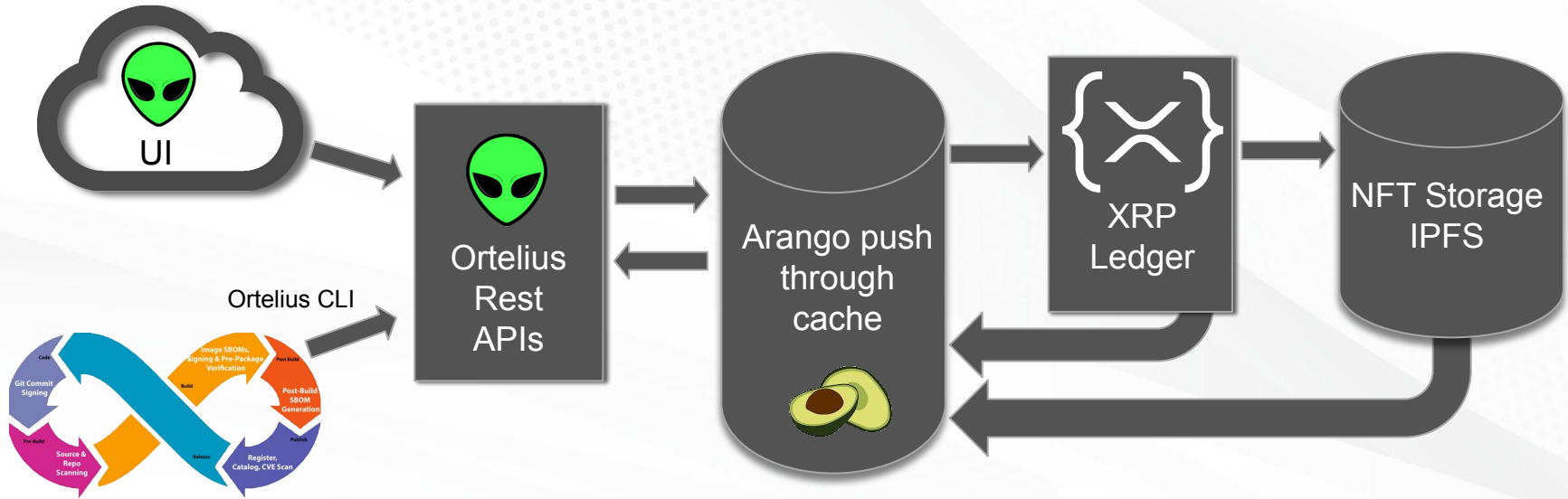
Answer the question “who is using Log4J?”



The screenshot displays the ORTELIUS application interface. On the left is a sidebar with navigation icons and labels: Applications, Components, Domains, Environments, Endpoints, Actions, Func/Procs, Customize Types, and Setup. The main area is titled "Package Consumption by Components and Application Versions". It contains a table with the following columns: Package, Package Version, Component, and Application. The table lists three entries for the log4j-api package, two for log4j-over-slf4j, and one for log4j-to-slf4j, each with its version and associated component and application details.

Package	Package Version	Component	Application
log4j-api	2.17.0	GLOBAL.Santa Fe Software.Online Store Company.hello-world;master:v1_0_0_101_g3b3bbdd	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_g3b3bbdd	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>

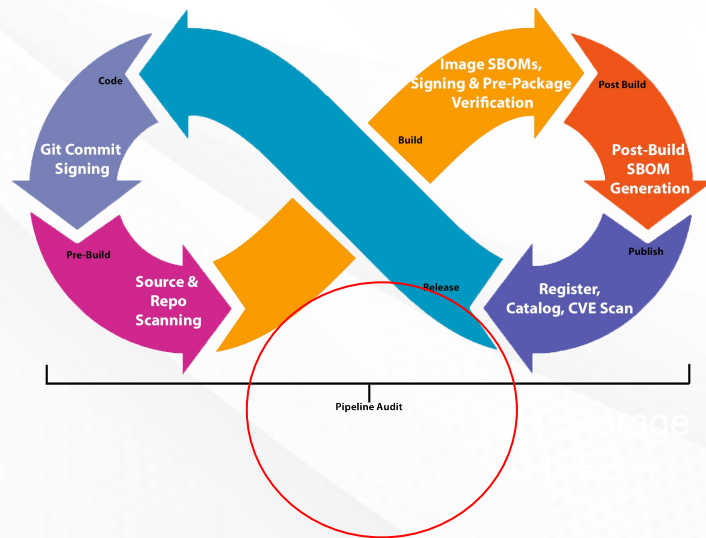


## Phase 5 - Pipeline Audit

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:

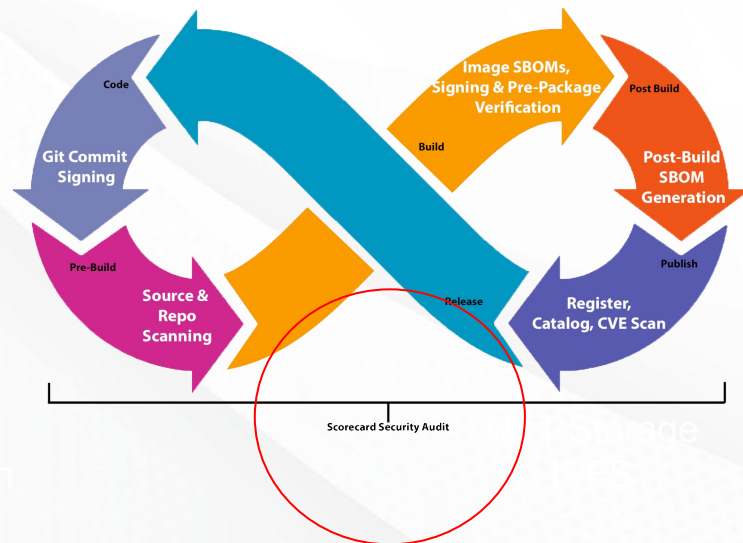
- Jenkins Audit Trail
- Tekton Chains



# 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](https://securityscorecards.dev)



# Open Source Security Tools Landscape

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

## 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>

