How to Manage Open Source at Scale in a Global Enterprise?

Peter Giese, SAP Open Source Program Office
14. March 2019
Gardener

UI5 Web Components

CLA Assistant

OpenUI5

Vulas

OrientDB

Kyma

InfraBox
Linux Foundation
In addition to involvement with the Linux Foundation projects listed here, SAP is a Silver member of the Linux Foundation itself
Visit Site

Apache Software Foundation
SAP is involved with the Apache Olingo project, and with Apache Hadoop through Altiscale
Visit Site

OpenJDK
SAP is a contributor to the OpenJDK project
Visit Site

Cloud Native Computing Foundation
Platinum member; SAP is represented on the Board
Visit Site

Eclipse Foundation
Founding member and a strategic development member of the Eclipse Foundation.
Visit Site

Open Stack Foundation
SAP is a Corporate Sponsor of the Open Stack Foundation
Visit Site

Cloud Foundry Foundation.
Founding member and a Platinum member; SAP is represented on the Board of Directors
Visit Site

Open API
Silver member.
Visit Site

ToDo Group
General member.
Visit Site
SAP: The World’s Largest Provider of Enterprise Application Software

96,000+ Employees

180+ Countries

18,300+ Partners

77% of the world’s transaction revenue

78% of the world’s food

82% of the world’s medical devices
OSPO Working Mode – Scrum

One joint OSPO product backlog in Jira

Divided into seven epics

Each epic handled by a cross-functional scrum team
OSPO Working Mode – Scrum

Source: https://commons.wikimedia.org/wiki/File:Scrum_process.svg

After nine sprints:

- 150+ sprint backlog items completed
- 400+ product backlog items created
Exponential Growth of Open Source in Enterprise Application Software

SAP Inbound Open Source

- <2015
- 2016
- 2017
- 2018

Customer Expectations:
- SAP product standards
- Compliance

SAP Outbound Open Source

- 1000+ Contributors on Github

Customer & Community Expectations:
- SAP product standards
- CII Badge Program
Core Infrastructure Initiative (CII) Badge Program

- CII Badge is an open source **secure development** maturity model
- CII Badge criteria **codify best-practices** used by open source projects
- **CII Gold Badge** best matches SAP’s Product Standards
  
  Some projects cannot fulfill the [contributors_unassociated](#) continuity criterion that “the project MUST have at least two unassociated significant contributors”
  
  ➔ **CII Silver Badge**
CII Gold & Silver Badge Criteria – Open Source Tools from SAP

[**dco**]  →  CLA Assistant

[**vulnerabilities_fixed_60_days**]  →  Vulas

[**no_leaked_credentials**]  →  SCCS; not open sourced (yet)
The project SHOULD have a legal mechanism where all developers of non-trivial amounts of project software assert that they are legally authorized to make these contributions.

The most common and easily-implemented approach for doing this is by using a Developer Certificate of Origin (DCO), where users add "signed-off-by" in their commits and the project links to the DCO website.

However, this MAY be implemented as a Contributor License Agreement (CLA), or other legal mechanism.
CLA Assistant (github.com/cla-assistant) helps to handle the legal side of contributions to open source projects and to streamline the contribution workflow.

It is also provided as free hosted offering (cla-assistant.io) which allows contributors to sign a CLA from within a pull request by authenticating themselves with their GitHub account.
CLA Assistant Demo
CLA Assistant - Adoption

4.300+ linked GitHub repositories
300+ linked GitHub organizations
52.000+ CLA signatures
43.000+ users
Vulnerability Assessment Tool (Vulas)

CII Badge Gold [vulnerabilities_fixed_60_days]:

There MUST be no unpatched vulnerabilities of medium or high severity that have been publicly known for more than 60 days.
Vulnerable Open Source Components

80% to 90% of software products on the market include OSS components

Using components with known vulnerabilities:
- Root cause of major data breaches
  - Mossack Fonseca (*Panama Papers*) breach
  - *Equifax* breach

Impact Analysis is Difficult

Vulnerabilities are assigned to entire projects (e.g., Apache POI, Tomcat), but sub-components (e.g., Jar archives) are used separately.
Existing approaches (based on meta-data)

- Most tools “somehow” map finer-grained OSS components (e.g., JAR archives) to vulnerabilities using the project metadata
- Actual code often ignored

Limitations:
- False-positives (e.g. multi-module projects)
- False-negatives (e.g. re-bundling)
- Focus only on detection (no app-specific analysis)
Vulnerability descriptions (in natural language) often not useful

**CVE-2012-5633**: “The URIMappingInterceptor in Apache CXF before 2.5.8, 2.6.x before 2.6.5, and 2.7.x before 2.7.2, when using the WSS4JInInterceptor, bypasses WS-Security processing, which allows remote attackers to obtain access to SOAP services via an HTTP GET request.”
Vulas – Vulnerable Constructs

Repository:
https://github.com/some/project

Commits: 236AF0619, 1234E4D5C1A, ....

(Vulnerable advisory)

(CVE, GitHub issue)

Method x.y.z ADDED
Method a.b.c MODIFIED
Constructor i.j.k DELETED

(Vulnerable Constructs)
Vulnerable Constructs

Repository: https://github.com/some/project
Commits: 236AF0619, 1234E4D5C1A, ....

Method x.y.z ADDED
Method a.b.c MODIFIED
Constructor i.j.k DELETED

…………………………

Vulnerability advisory (CVE, GitHub issue)

Patch Information

Vulnerable Constructs

https://github.com/SAP/vulnerability-assessment-kb

Open-source vulnerability assessment knowledge base

The open-source vulnerability assessment knowledge base aggregates public information about security vulnerabilities in open source projects, the fuel required to run the vulnerability assessment tool.
Vulas Contributions

- From vulnerability to **vulnerable constructs** (actual code)
- Code-centric **detection** of known vulnerabilities
- Static and Dynamic **assessment** of vulnerable code
- Metrics to support selection of non-vulnerable libraries (**mitigation**)

Supported languages: Java, Python
Vulas Approach

**Assumption:** If an application contains and executes **vulnerable constructs**, then there is a significant risk that the vulnerability can be exploited in the application context.

- Static reachability analysis
- Dynamic analysis
- Combination of static & dynamic analysis
Vulnerability Assessment Tool Demo
Vulas – Adoption @ SAP

- 1+ Mio. scans of 800+ applications
- Recommended @ SAP after comparison with existing open / commercial tools

**Enterprise-ready**
- Usable in CI/CD pipelines, but also for legacy software
- Aggregated reports and audit of findings → could be exported to SPDX
- Support of CERT: Which of our apps are impacted by vulnerability X?
- Non-disclosed (internal) vulnerabilities can also be added to knowledge base

**Client-side tools**
- Plugins for Maven and Gradle (Java) and setup tools (Python)
- Command Line Interface (CLI) for everything else
Vulas is Open Source

Establish Collaboration to Reduce risks coming from usage of vulnerable OSS

- Open source foundations/projects: Upstream fixes, contributions to vulnerability knowledge base
- Enterprises: Productivity features (reporting, usability, etc.)
- Universities: New languages, new analysis techniques

Links

- Newsletter: [vulas-news-request@listserv.sap.com](mailto:vulas-news-request@listserv.sap.com) (“subscribe” in the body)
- Knowledge base with public vulnerabilities: [https://github.com/SAP/vulnerability-assessment-kb](https://github.com/SAP/vulnerability-assessment-kb)
SCCS – SAP Credential Code Scanner

CII Badge Gold [no_leaked_credentials]:

The public repositories MUST NOT leak a valid private credential (e.g., a working password or private key) that is intended to limit public access.

Source: https://breachlevelindex.com/
SCCS – SAP Credential Code Scanner

SCCS is a standalone static source code scanner that looks for credentials and sensitive data using:

• Regular expressions
• Inference rules
• CNN deep learning for passwords

SCCS looks for

• Passwords, encryption keys, hashed data, tokens (cloud APIs), signatures, e-mail addresses, users / user IDs, internal domains, IP addresses
Dashboard

Repositories: 31

Scans: 0

Findings: 0

Scan New Repo

Repo

Scan Options: (hold ctrl for multiple)
- c_crea.yml
- default_config.yml
- model 0
- model 1
- model 2

Fresh Scan: On

Submit
Run Better Together With Open Source

CLA Assistant → cf. Github issue list

Vulas → Support for additional languages like JavaScript & Go
→ collaboration on vulnerability assessment knowledge base

SCCS → collaboration on credential code scanners

https://commons.wikimedia.org/wiki/File:Collaboration_logo_V2.svg
Thank you!