FINOS

Fintech
Open Source
Foundation

Evaluating Compliance Solutions and Vendors

Aaron Williamson
Open Source Readiness Lead

December 4, 2019

Metrics to evaluate and compare source code scanning tools

Ibrahim Haddad

There are a number of existing open source compliance tools in the market. How to decide which one is best for you?
We provide a number of metrics that you can refer to to compare and contrast such tools.
If you have any suggestions to improve this list, please reach out.

- ***Size of the knowledge base*** against which scanned code is being compared
- ***Frequency of updates to the knowledge base*** - how often does the tool provider update the knowledge base to keep up with the pace of open source development?
- ***Speed of scans*** for the same loads
- ***Supported deployment models*** - cloud, on premise, hybrid
- ***Ability to identify origin and license of snippets*** - many tools do not provide such support and are only capable of identifying whole open source components, others have poor support
- ***Ability to auto-identify open source snippets*** in scanned code flagging their component of origin and license - saving endless hours on manual labor
- ***Support for vulnerability discovery*** - is the tool capable of identifying vulnerable code that was copy/pasted from one component into another? Or simply just able to identify vulnerabilities found in their original components
- ***Ability to represent and manage end-to-end review and approval process*** directly from within the tool via a self defined workflow
- ***Total cost of ownership*** - *which include the yearly license cost, training cost, cost of customizations (workflow, features, integration, etc.), cost of servers required for your specific install and Internal sys admin support for your install*
- ***An intuitive UI*** - easy and inviting to use – minimizing learning curve and making it less of a chore
- ***Support for APIs and a CLI*** that you can interconnect with your CD/CI environment to for ease of integration with existing development and build systems
- ***Ability to use the tool for M&A transactions*** without restrictions on the use of the tool as part of the licensing agreement
- ***Support for different audit methods*** - several methods exist
- ***Programming languages agnostic*** - the tool should be able to process any source code regardless of the programming language
- ***Support for SPDX*** - discovering licenses declared using SPDX identifiers and exporting scan results in SPDX format
- ***Ability to represent company policies and apply them on scanned code*** triggering specific actions depending on the license of the scanned code and related policies

Read more on practical open source compliance: Download free ebook "Open Source Compliance for the Enterprise" (2nd Edition).

Open source dependency identification

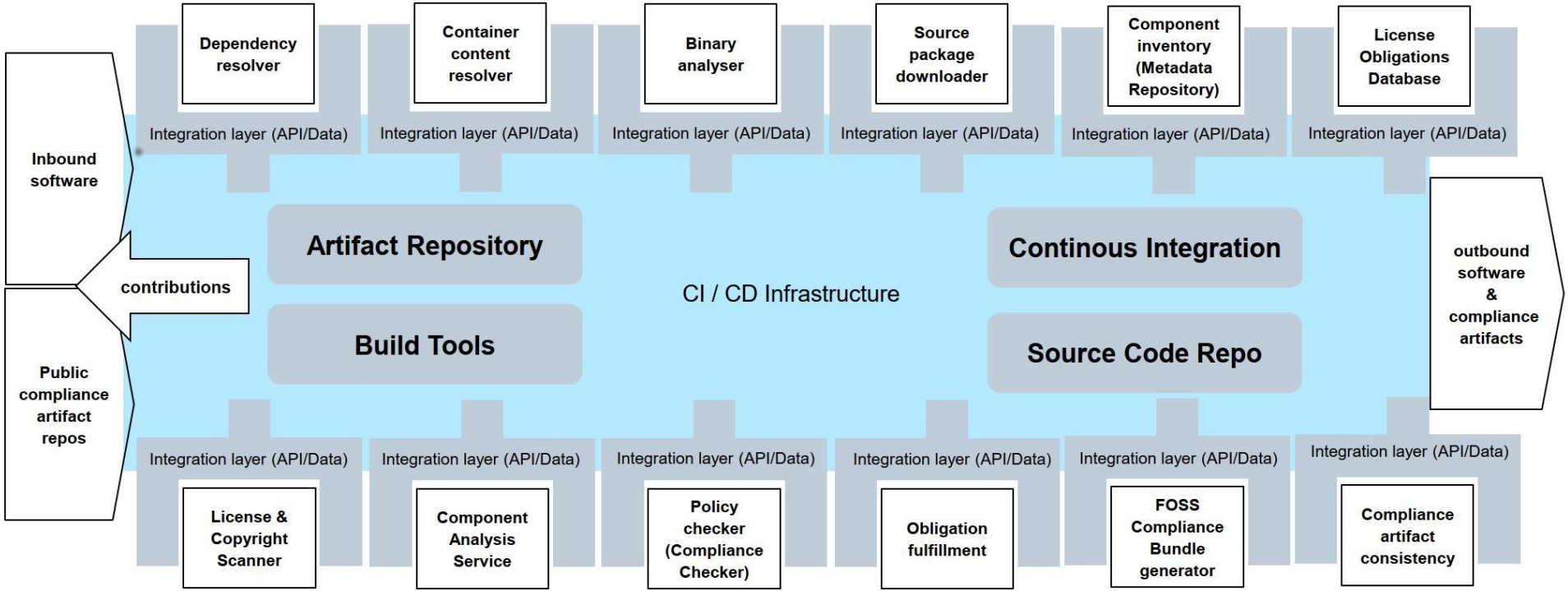
Open source dependencies show up in several forms:

- Dependencies declared in code, package manager manifests, or build files, and transitive dependencies of these
- Original or modified components included as source code
- “Snippets” or partial open source components embedded in proprietary source code
- Compiled binaries of open source components
- Source or binary components embedded in archives or container images

Open source dependency identification

Finding every dependency requires multiple approaches:

- Automated reporting by build system/package manager
- Scanning source code for:
  - declared dependencies
  - “fingerprints” of known open source components
  - open source license text & statements
- Inspection of binaries for text associated with OSS components & licenses
- Recursive inspection of archives and containers for source & binary components



Unrestricted 2019 License: CC-BY-SA-4.0 Oliver Fendt

Source code scanning - completeness

Does the product:

- integrate with build tools to identify transitive dependencies?
- identify complete open source components stored as source code in a product's source code repository?
- Identify partial or modified components in source tree?
- identify partial open source components (i.e. snippets) embedded in proprietary source code?
- scan source code for license statements?

Source code scanning - features

- Is the product technology-agnostic — does it provide the same accuracy/completeness regardless of programming language or technology?
- For products that fingerprint source code/snippets, how many open source components and versions are contained in the product's knowledgebase?
- How frequently is the knowledgebase updated?
- What is the average speed of a scan (e.g. files/second)?

Scanning non-source artifacts

Does the product:

- identify compiled/binary open source components?
- Identify open source components embedded in proprietary binaries?
- inspect the contents of archive files (e.g. zip, apk, jar) to identify open source components?
- inspect the contents of container images to identify open source components?

OSS identification — reporting

- Does the tool:
  - identify when a product is using an outdated version of an open source component (regardless of vulnerabilities)?
  - identify when a product includes an open source component that is not used in the product?
  - provide information on the frequency with which open source components are historically updated and/or released?
  - produce a software bill of materials in SPDX format?
  - produce complete compliance materials for a product?
  - provide useful guidance re: compliance obligations?

Security vulnerability scanning

- What public sources of vulnerabilities does the product draw from?
- Does the product's vulnerability database include any proprietary information or analysis? To what extent?
- Can vulnerabilities be traced to snippets of open source code?
- Is your tool capable of performing static analysis on source code to identify non-listed security vulnerabilities?

Integration with SDLC

- Is the product available as on-prem? Cloud-only? Hybrid?
- Does the product:
  - integrate with your existing tools?
    - SCM
    - Issue tracking
    - Product/project management
  - provide an API to enable additional integrations?
  - support complete review & approval process (via self-defined workflow)?
  - allow license and security policies to be defined and automatically enforced?

Product features

- Is the product available as on-prem? Cloud-only? Hybrid?
- Does the product support complete review & approval process (via self-defined workflow)?
- Is the product's UI intuitive for everyone to be involved in process?
- What is the total cost of ownership, taking into account:
  - License
  - Servers
  - Training
  - Internal sysadmin support FTEs
  - Necessary integrations
- Does the product & license support all use cases, including M&A and audit of legacy products?

FINOS

Fintech
Open Source
Foundation

info@finos.org

finos.org