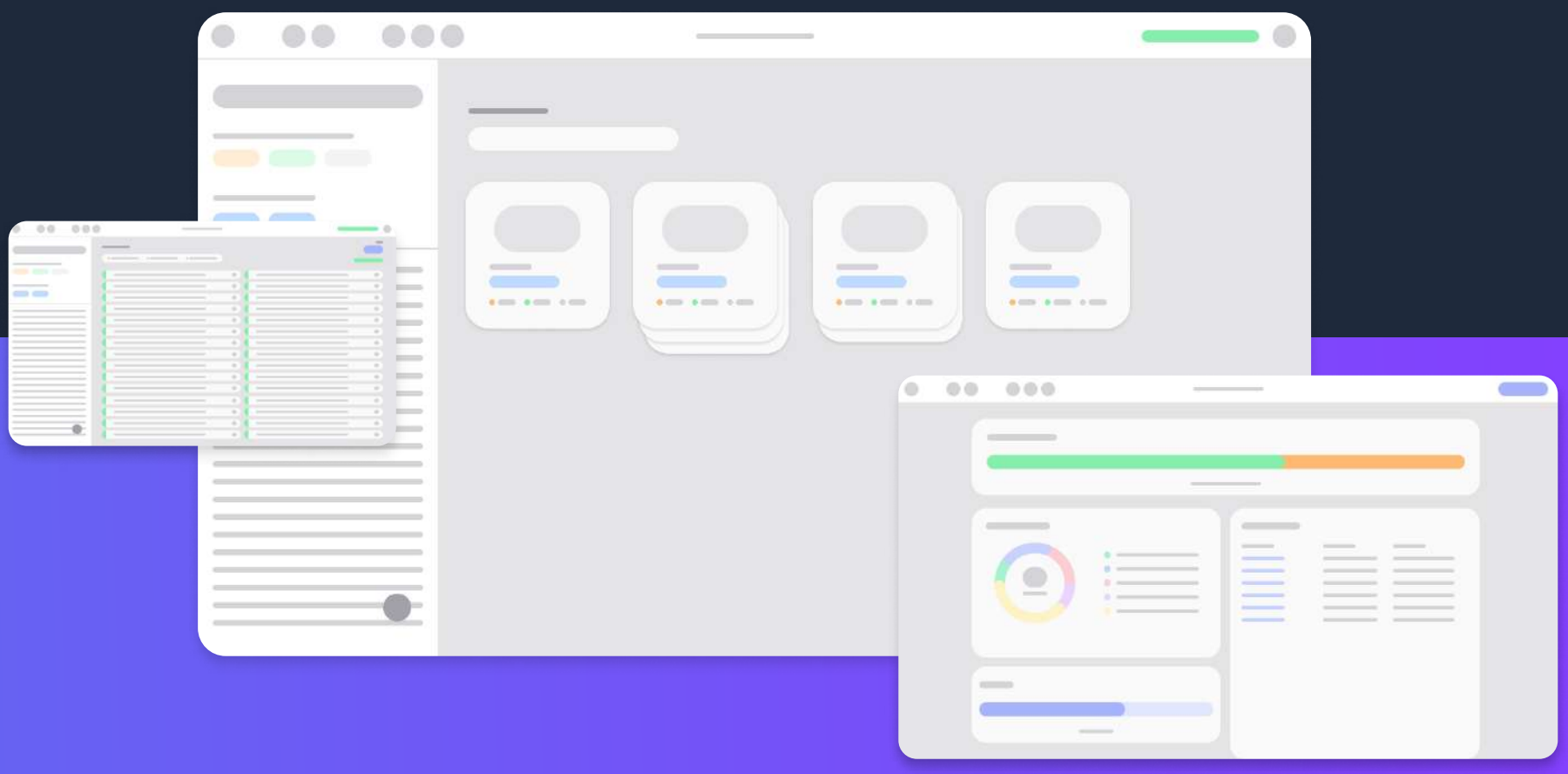
Whitepaper

Standaridizing Open Source Inventorying

The first Open Source Inventory Engine
built for developers & modern DevOps organizations.



SCANOSS

Content

Questions?

Please get in touch through https://www.scanoss.com
or info@scanoss.com

Background

Software Composition Analysis (SCA) tools perform analysis, comparison and identification of Open Source components. Sadly, none of the SCA vendors have embraced Open Source themselves. The proprietary nature of their tools leads to struggle at the customer side. Customers fall into vendor lock-in scenarios and it is very difficult to integrate these tools into their existing infrastructure. This leads to expensive post-processing of results, expensive maintenance of glue code, slow performance and increased hardware costs.

SCA Tooling That Fits a Modern DevOps Environment.

SCA tools were originally conceived with the specific purpose of creating a Software Bill of Materials (SBOM) from static snapshots of source trees. The SBOM quickly became a de facto requirement for due diligence processes during M&As, but also gained traction as part of pre-delivery audits at the end of development cycles.

However, development teams have started to struggle to adopt these tools in continuous development environments. Given the closed nature of existing SCA tooling, it's hard to integrate them in the development process and even harder to compare the results. There is a clear need for modern development teams to start left in the development process by performing continuous validations, rather than waiting for a final audit.

Software Development is a continuous process. Performing audits on static source code is costly and the result is most likely obsolete by the time assessment is completed. Open Source Inventorying will expose interfaces that allow natural integration to the validation process of a modern DevOps environment.

REDEFINING SCA: the first open source solution

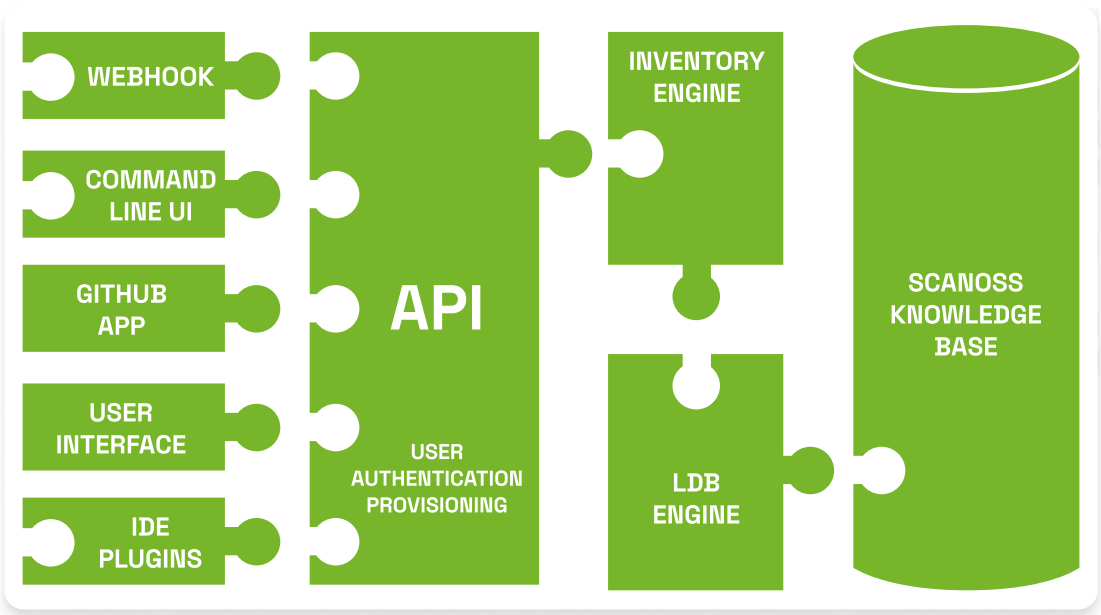
SCANOSS believes now is the time to reinvent Software Composition Analysis with a goal of 'start left' and a focus first on the foundation of reliable SCA, the SBOM. An SBOM that does not require a small army of auditors to make it usable. So, SCANOSS provides an SBOM that that is 'always on'.

SCANOSS now releases the first entirely Open Source software platform for Open Source Inventorying, specifically designed for modern development (DevOps) environments. The platform includes:

- **Open Data Mining tool (minr)**
- **Open Database Engine (ldb)**
- **Open Inventory Engine & SBOM**
- **Compliant RESTful API**
- **Webhooks and Command Line Interface (CLI)**

Open Platform Architecture

The complete SCANOSS-platform is released as Open Source. The pillar of the architecture is an OpenAPI compliant RESTful API that exposes functionality for the inventory engine to be interfaced from CI/CD tools as well as a full set of features for OSS auditing and other user interfaces.



The pillar of our architecture: a RESTful API

Client-side applications and middleware connect to the RESTful API via https. The API handles interaction with the Inventory Engine, which uses the LDB interfaces to query the database.

Open SBOM (Software Bill of Materials)

The SCANOSS platform uses indistinctively SPDX or CycloneDX (in their JSON variant) as the core SBOM storage format. XML variants are also supported both for importation and exportation of SBOMs.

Open Indexing Algorithm (https://github.com/scanoss/wfp)

The knowledge base stores source code fingerprints for entire files and code fractions (snippets). These fingerprints are used for matching source code during analysis. Full file fingerprints have been standardized by using cryptographic hash algorithms, such as MD5 and SHA. However, looking at the market for SCA tooling, there are no standards defined for comparing snippets. SCA vendors implement their own proprietary algorithms for calculating snippet fingerprints, which means it is virtually impossible to compare results from different tools. Moreover, the proprietary nature of such algorithms imply that closed binaries are used for performing analysis, which is always a concern given the sensitive nature of corporate source code.

SCANOSS uses an open algorithm known as Winnowing, that has been used extensively in academic circles to obtain and compare fingerprints from documents and source code. These fingerprints are used to detect plagiarism against known texts and source code. There are several open source implementations of the Winnowing algorithm available today. Given the wide adoption and broad availability of open source implementations, SCANOSS has adapted this algorithm for indexing and comparing massive amounts of source code.

Further details and source code available at:
https://www.scanoss.com/winnowing.html

Open Database Engine (https://github.com/scanoss/ldb)

One of the main challenges faced by SCA tool manufacturers given thevast number of data records, is the performance of existing database engines and the overall database footprint.

Comparing a single file may require hundreds of database queries. In a large source code project, the latter easily multiplies to millions of database queries. This means query performance becomes critical. In database terms the requirement is to perform very simple queries on a perfectly balanced tree of numeric keys. Given such simple requirements, the broad set of features offered by general purpose SQL or NoSQL database engines becomes an expensive overhead.

SCANOSS designed a database engine specifically for this use case andhas already passed the 2 trillion fingerprints mark. SCANOSS' DatabaseEngine (called: 'LDB') organizes data in mapped linked lists, enablinghardware-speed searches. Comparisons are performed in microseconds,which allows the scanning of thousands of files per second while thedatabase footprint is kept to a bare minimum and can be distributed across devices.

Open Inventory Engine (https://github.com/scanoss/engine)

The SCANOSS Inventory Engine performs comparison of source code against the Open Source Knowledge Base using the provided LDB Database interfaces.

Analysis is performed either on source code files or on pre-calculated Winnowing fingerprints. Output is presented in JSON format.

Open RESTful API (https://github.com/scanoss/API)

The SCANOSS API interfaces with the Inventory Engine and provides multiple functionality for Audit Management, User Management, Provisioning, Project Management, Authentication and Authorisation. Reference code is provided with the API, including an auditing UI and a Webhook and a Command Line Interface scanner (scanner.py).

Open Webhook (https://github.com/scanoss/webhook)

The SCANOSS Webhook features integration to the main software repository providers and enables a secured & automated source-code check that triggers upon every Git PUSH. The Webhook automatically retieves the changed files along with the optional Open Source Assets declaration and posts them to the inventory server for analysis.
The build status for the commit is automatically updated and a comment is posted highlighted by a SCANOSS badge.
If the presence of an undeclared Open Source asset is detected, regadless of whether it is a complete file or a small code snippet, the commit is marked as ”build failed” and a ”failed” badge is added to the commit comment, regardless of whether it is a complete file or a small code snippet.

Open Command-line Interface (https://github.com/scanoss/scanoss.py)

SCANOSS.py is a Python client that performs direct recursive scanning of the provided directory, among many other things. The scan is performed using the SCANOSS API.

Fragment sample use of the scanner.py CLI:

```
scanoss-py scan .
Searching . for files to fingerprint...
Scanning fingerprints...
{
 "Dockerfile":[
 {
  "id": "none",
  "server": {
   "hostname": "p8",
   "version": "4.3.4",
   "flags": "0",
   "elapsed": "0.000054s"
  }
 }
],
 "version.py":[
 {
  "id": "file",
  "status": "pending",
  "lines": "all",
  "oss_lines": "all",
  "matched": "100%",
  "purl": [
   "pkg:github/scanoss/scanoss.py",
   "pkg:pypi/scanoss"
  ],
  "vendor": "scanoss",
  "component": "scanoss.py",
  "version": "0.6.11",
  "latest": "0.6.11",
  "url": "https://github.com/scanoss/scanoss.py",
  "release_date": "2021-10-18",
  "file": "version.py",
  "url_hash": "b80bc2e3d5dfa13f6a6c26ae44c824d8",
  "file_hash": "7c4e6ef45455b344810945caf995ded7",
  "file_url": "https://osskb.org/api/file_contents/7c4e6ef45455b344810945caf995ded7",
  "dependencies": [],
  "licenses": [

.... Continues
```

Open Mining Tool (https://github.com/scanoss/minr)

The SCANOSS Mining Tool (called: 'Minr') is a command-line tool that performs download, extraction and indexing of a source code component. The tool is released as Open Source with the purpose of allowing users to create their own source code knowledge bases.

The arbitrary creation of knowledge bases can have a myriad of applications. E.g. a company could create a knowledge base of proprietary code to detect the unwanted release of an IP. A different application could be performing a comparison against only a designated set of OSS components.

Example use of the SCANOSS Minr tool:

```
$ minr scanoss, scanner.py, 1.0 http://github.com/scanoss/scanner.py/
archive/1.0.tar.gz
Downloading http://github.com/scanoss/scanner.py/archive/1.0.tar.gz
Completed in 0.858s
$find mined -not -empty
mined
mined/sources
mined/sources/2b29.mz
mined/components
mined/files
mined/files/2b.csv
mined/snippets/24.bin
mined/snippets/38.bin
...
mined/snippets/6f.bin
mined/snippets/7b.bin
```

Minr fetches the components and subsequently generates component, file and snippet data. Minr can be launched in a number of instances, from any number of machines and the resulting data can be joined by simple concatenation, to be inserted into the knowledge base at a later point in time.

Get involved

The SCANOSS Platform is made entirely available entirely as Open Source. The collaboration guidelines are available in the source code tree. Questions and suggestions are welcome at https://www.scanoss.com

Get in touch

SCANOSS offers commercial agreements with access to its complete Knowledge Base, additional features and Service Level Agreements.

Please contact info@scanoss.com for further information.

Published on 2020-07-08 by SCANOSS.com

SCANOSS