OPENCHAIN

Official Checklist

Metrics to Evaluate Source Code Scanning Tools

Item	Meets Requirements?	Notes
Size of the knowledge base		Compare this 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		Can it auto-identify in scanned code flagging their component of origin and license?
Support for vulnerability discovery		Is the tool capable of identifying code that was copy/pasted from one component into another? OR, simply 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		Should include the yearly license cost, training, customization cost, cost of servers required for your specific install and internal sys admin support
Does it have an intuitive UI?		Should be easy to use, minimizing the learning curve
Support for APIs and a CLI that you can interconnect with your CD/CI environment?		
Ability to use the tool for M&A transactions?		No restrictions on the use as part of the licensing agreement
Support for different audit methods?		
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?		Will trigger specific actions depending on the license of the scanned code and related policies