Managing the Security Risk of Open Source Dependencies: Current Tools & Challenges

Nasif Imtiaz, Laurie Williams

North Carolina State University

Key insights

1. Software Composition Analysis (SCA) tools can detect open source dependencies and report known vulnerabilities in them.
2. A key strength of an SCA tool is the accuracy, up-to-dateness, and completeness of its vulnerability database.
3. Not all dependency vulnerabilities may pose a security risk to the application.
4. Future research opportunity lies in - i) Understanding how developers assess security risk and make fix decisions for dependency vulnerabilities.
   ii) Automation techniques to ensure continuous monitoring of vulnerability data in the open source ecosystem.

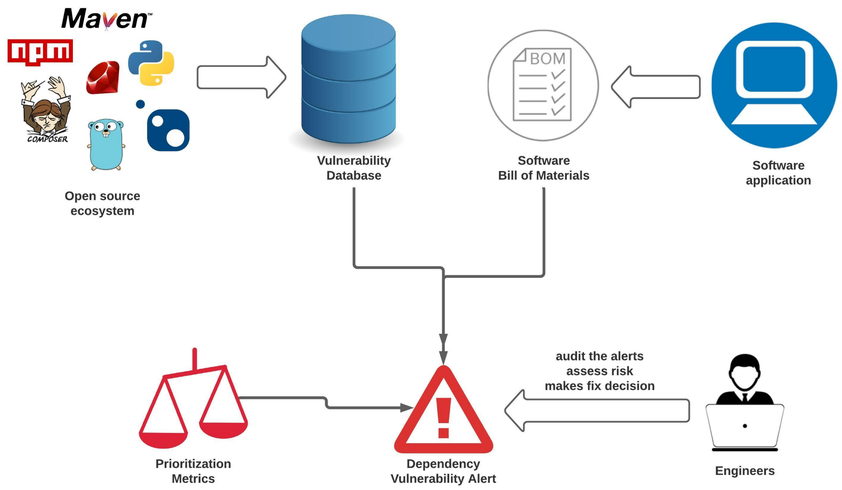

Figure: SCA workflow

SCA comparison

- We run 9 SCA tools on a web application, OpenMRS, consisting of 43 Maven and 5 npm projects
- We find the tools vary widely in their reporting of unique vulnerabilities and the unique dependencies to contain these vulnerabilities
- We characterize five type of metrics provided by the studied tools to aid in risk assessment of dependency vulnerabilities, most notably reachability analysis.

Table 2: Vulnerable Dependencies for Maven (Java) projects

Tool	Alert	Unique Dependency	Unique Package	Unique Vulnerability	CVE	Non-CVE	Scan Time (Minutes)
	Total (Median per project)						
OWASP DC	12,466 (254.0)	332 (38.0)	149 (36.0)	313 (117.0)	289	24	14.4
Snyk	4,902 (66.0)	96 (6.0)	46 (6.0)	189 (23.0)	178	11	15.1
Dependabot	136 (0.0)	20 (0.0)	11 (0.0)	61 (0.0)	61	0	NA
MSV	3,197 (58.0)	36 (12.0)	14 (12.0)	36 (22.0)	36	0	3.4
Steady	2,489 (51.0)	91 (20.0)	39 (19.0)	97 (41.0)	89	8	385.0
WhiteSource	434 (0.0)	76 (0.0)	44 (0.0)	146 (0.0)	127	19	NA
Commercial A	2,998 (70.0)	107 (24.0)	53 (24.0)	208 (70.0)	187	21	NA
Commercial B	205	35	35	127	127	0	NA

Table 3: Vulnerable Dependencies for npm (JavaScript) projects

Tool	Alert	Unique Dependency Path	Unique Dependency	Unique Package	Unique Vulnerability	CVE	non-CVE	Scan Time (Minutes)
	Total (Median per project)							
OWASP DC	1,379 (208.0)	498 (72.0)	239 (71.0)	160 (57.0)	234 (71.0)	78	156	4.4
Snyk	2,210 (135.0)	1,004 (44.0)	90 (20.0)	54 (17.0)	121 (26.0)	79	42	1.0
Dependabot	97 (8.0)	NA	32 (1.0)	30 (1.0)	45 (4.0)	29	16	NA
npm audit	1,266 (37.0)	852 (28.0)	58 (12.0)	45 (12.0)	62 (16.0)	31	31	0.1
WhiteSource	205 (32.0)	205 (32.0)	89 (14.0)	55 (9.0)	96 (18.0)	58	38	NA

Key differences

- Accuracy and completeness of vulnerability database
  - Tools can report unique non-CVEs not reported by other tools.
  - Tools may have different mapping of vulnerability to affected versions of packages.
- How dependencies are detected
  - OWASP DC and WhiteSource detected JavaScript dependencies in Maven projects though source code analysis.
  - Commercial B only reported dependencies under use during runtime.

Realsearch
Real People · Real Projects · Real Impact

8TH ANNUAL
HOT TOPICS *in the* SCIENCE OF SECURITY
APRIL 13-15, 2021 | VIRTUAL