SCA-Tools (Lieferkettensicherheits-Tools) in der Übersicht

Die Software Composition Analysis soll Risiken aufdecken, die Entwickler beim Einsatz von Open-Source-Komponenten eingehen, und die Softwarelieferkette absichern. Der Markt für passende Produkte ist riesig und in ständiger Bewegung.

-tract

- Software Composition Analysis (SCA) ist eine Form der Codeanalyse, die ermittelt, welche Open-Source-Bibliotheken eine Software verwendet, welche bekannten Schwachstellen in ihnen enthalten sind und unter welcher Lizenz sie stehen.
- SCA-Werkzeuge unterstützen dabei und automatisieren diesen Prozess, indem sie sich in Code-Repositorys, CI/CD-Pipelines und oft auch IDEs integrieren.
- Der Markt ist geprägt von sehr vielen Anbietern und oft recht jungen Produkten. Eine Auswahl von Angeboten von etablierter Herstellern, die auf dem deutschen Markt aktiv sind, stellt diese Übersicht vor.

Sicherheit rückt nach links. Nicht politisch, sondern im Sinne des Left Shift der DevOps-Bewegung. Es bedeutet, dass immer mehr Kompetenzen am Anfang, also links im gesamten Prozess angesiedelt sind: bei den Entwicklern. Mit DevSecOps wird aus der Verantwortung für den Betrieb (DevOps) nun Verantwortung für den sicheren Betrieb. Das macht aber Entwickler nicht auf magische Weise zu Securityspezialisten. Deshalb ist jede Unterstützung in Form von Werkzeugen oder Frameworks gefragt, die helfen, möglichst viele Risiken so früh es geht zu entdecken und Sicherheitslücken zu stopfen.

Da nahezu jedes größere Softwareprojekt Open-Source-Komponenten enthält, betrifft dies nicht nur die vom eigenen Entwicklerteam zu verantwortenden Schwachstellen, sondern auch die in den eingebundenen Abhängigkeiten. Die Aufgabe der Software Composition Analysis besteht darin, herauszufinden, welche Komponenten in welchen Versionen in der eigenen Software stecken, und dann zu ermitteln, welche schon bekannten Schwachstellen diese haben. Über diese absolute Mindestanforderung an eine SCA-Software gehen aber alle am Markt vorhandenen Systeme hinaus und bieten Einbindung in CI/CD-Pipelines oder Entwicklertools, automatische Lösungsvorschläge für gefundene Schwachstellen (Remediation), diverse Dashboards, Frameworks zum Festlegen von Richtlinien und so weiter.

Entdecken, dokumentieren, beheben

In aller Regel erfüllt SCA zudem eine Doppelfunktion. Zusätzlich zu Sicherheitsrisiken soll sie auch Compliance-Risiken identifizieren, indem sie die Open-Source-Lizenzen findet, unter denen verwendete Komponenten veröffentlicht sind. Das macht auch Rechtsabteilungen und Management zu SCA-Anwendern, die Software darf also unter Umständen nicht ausschließlich auf die Bedürfnisse von Developern zugeschnitten sein. So gut wie immer kann ein SCA-Werkzeug SBOMs (Software Bills of Materials) erzeugen, also „Zutatenlisten“, die beispielsweise US-Behörden per Präsidentenerlass verlangen müssen [1].

Um die Komponenten zu ermitteln, lesen SCA-Tools die Abhängigkeiten aus den Manifestdateien verschiedener Paketmanager aus, etwa NPM, Maven oder NuGet; manche scannen darüber hinausgehend auch den Sourcecode selbst oder sogar Binärdateien. Für den Abgleich mit bekannten Sicherheitslücken nutzen kommerzielle Anbieter in der Regel eigene Datenbanken, Open-Source-Programme greifen oft auf frei verfügbare Quellen zurück, wie die National Vulnerability Database (NVD), die das

US-amerikanische National Institute of Standards and Technology (NIST) pflegt, oder die ebenfalls recht umfassenden GitHub Security Advisories.

Der Markt für SCA-Software ist ausgesprochen vielfältig, neben ausgereiften und von großen Organisationen unterstützten Open-Source-Programmen tummeln sich Spezialhersteller und die etablierten Anbieter großer Sicherheitslösungen, die in den letzten zwei Jahren SCA entweder in ihre Suiten integriert oder separate Produkte lanciert haben. Zur großen Fülle an Herstellern und Produkten mag beitragen, dass es technisch eher eine Fleißarbeit ist, die Grundfunktionen zur Verfügung zu stellen: möglichst viele unterschiedliche Manifestformate der Paketmanager parsen und mit Datenquellen zu Sicherheitslücken abgleichen, Export in gängige SBOM-Formate, dazu noch etwas Integration in bereits vorhandene Frameworks zu Datenaufbereitung, Nutzermanagement, Entwicklertools und DevOps-Pipelines – fertig ist die SCA-Lösung.

Deshalb grenzen sich die führenden Anbieter auf diesem Gebiet auch alle durch spezielle Alleinstellungsmerkmale von ihren Mitbewerbern ab. Häufig haben sie weitere Analyseverfahren im Angebot und gestatten das zusätzliche Scannen von Source- oder Binärcode. Oft pflegen sie erweiterte Schwachstellendatenbanken, können interne Projekte in die Analyse einbeziehen, oder sie positionieren sich gezielt als umfassende Enterprise-Lösung, die alle Anwendungsfälle abdeckt und sich an ein heterogenes Anwenderfeld richtet.

Viel Bewegung im Markt

Die OWASP listet auf ihrer Website zum SBOM-Format CycloneDX 170 Plattformen und Werkzeuge auf, die ganz oder teilweise SCA-Funktionen haben (siehe ix.de/zvbm). Beim Eingrenzen der Auswahl ist auf die jährlichen Analysen von Gartner, Forrester und Co. nur bedingt Verlass. Manche Hersteller, die laut dem einen Analysten seit Jahren eine stabile, besonders starke Marktposition haben, werden bei dem anderen nicht einmal

erwähnt, andere rutschen von einem Jahr zum anderen zwischen Gartners magischen Quadranten hin und her.

Auch ist es fraglich, ob die Orientierung an den dort gelisteten Produkten immer sinnvoll ist, denn deren Schwerpunkt liegt auf großen Lösungen für den unternehmensweiten Einsatz. Nicht nur deren Implementierung kann aufwendig sein. Auch die Prozesse, an die sich alle Anwender gewöhnen müssen, sind nur mit großem Aufwand durchzusetzen. Mit etwas Pech ist das Produkt gekauft und eingerichtet, aber kaum einer nutzt seine elaborierten Features.

Für einzelne Projekte und kleinere Teams kann eine weniger umfangreiche, aber auch weniger komplexe Software die bessere Entscheidung sein – vorausgesetzt, sie lässt sich gut mit den vorhandenen Tools und Abläufen verheiraten. Open-Source-Werkzeuge, aber auch manche auf Cloud-native gebürsteten Spezialhersteller mit ihren SaaS-Angeboten kommen da am ehesten infrage.

Die hier vorgestellten Werkzeuge zählen zu den eher etablierten Produkten dieses Segmentes und stammen hauptsächlich von SCA-Spezialisten oder zumindest von Herstellern, deren sonstige Expertise in der Codeanalyse liegt und die auch im deutschsprachigen Raum aktiv sind. Hinzu kommt eine kleine Auswahl Open-Source- oder anderer kostenloser Tools. Die Angaben in dieser Übersicht beruhen auf öffentlich zugänglichen Informationen und auf Nachfragen bei den Herstellern; soweit verfügbar wurden die aktuellen technischen Dokumentationen der Produkte herangezogen. Es sind sowohl Produkte dabei, die sich on Premises installieren lassen, als auch solche, die komplett als Service in der Cloud angeboten werden. Manche Anbieter lassen ihren Kunden die Wahl zwischen verschiedenen Bereitstellungsmethoden, andere haben hybride Modelle im Angebot.

Übersicht ausgewählter SCA-Anbieter									
Anbieter	Aqua		Anchore/Community	Synopsys	Checkmarx	FOSSA	Mend	OWASP/Community Dependency-Track	Snyk
Produkt	Aqua Supply Chain Security	Aqua Trivy	Syft/Grype	Black Duck	Checkmarx SCA	FOSSA	Mend SCA	OWASP Dependency-Track	Snyk Open Source
Bereitstellungsmodell	Public Cloud, Private Cloud, on Premises, AWS, GCP	lokal	lokal	on Premises	SaaS, Private Cloud, on Premises	SaaS, on Premises	SaaS, lokaler Scan möglich	on Premises	SaaS, lokaler Scan möglich
Export von SBOM-Formaten	SPDX, CycloneDX	SPDX, CycloneDX	SPDX, CycloneDX, eigenes Format	SPDX, CycloneDX, Protex	CycloneDX (über API auch SPDX)	SPDX, CycloneDX, weitere Formate	CycloneDX und SPDX mit separatem Tool	CycloneDX	SPDX und CycloneDX mit API und CLI (Beta)
Abgleich mit Schwachstellendatenbanken	eigene Datenbank	eigene Datenbank	durch Integration mit Grype	NIST NVD oder Black Duck Security Advisories (mit separater Lizenz)	eigene Schwachstellendatenbank, zusätzlich Datenbank bösartiger Pakete	eigene Datenbank	eigene Datenbank	NIST NVD und weitere	eigene Datenbank
Integration in DevOps-Tools	u. a. GitHub Actions, GitLab, CI CD, Jenkins, CircleCI, Terraform Cloud	GitHub Actions und Azure DevOps (offiziell), CircleCI und weitere (Community)	teilweise Community-Plug-ins	ca. 15 CI-Plattformen	Jenkins, Azure DevOps, TeamCity, Bamboo	ca. 15 CI-Plattformen	u. a. Azure DevOps, Jenkins, CircleCI, Travis	Jenkins, Maven, Gradle, GitHub Actions	CircelCI, GitHub Actions, Jenkins, Maven, TeamCity, Terraform
Code-Repositorys	GitHub, GitLab, Bitbucket, Azure	Git-basierte	n. a.	GitHub, GitLab, Bitbucket	GitHub, GitLab, Bitbucket, Perforce, Azure	GitHub, GitLab, Bitbucket, Azure, Custom Imports	Hosted Integration für GitHub, Bitbucket, Azure, Self-hosted GitHub Enterprise, BB Server, GitLab	n. a.	Git-basierte
Scan von Artefakt-Repositorys	JFrog Artifactory, Nexus	nein	n. a.	JFrog Artifactory, Nexus	JFrog Artifactory, Nexus	nein	JFrog Artifactory, GitHub Packages	nein	JFrog, Nexus und weitere
Scan von Container-Images	Docker	Docker	Docker, OCI, Singularity	Docker (OpenShift, Kubernetes Package Manager, Pivotal Cloud Foundry)	Docker, AWS ECR (mittels Syft)	OCI-Container (apt, RPM und apk)	Docker, GCR, ACR, ECR	nein	mit Snyk Container
Compliance, Lizenzinformationen	ja	eingeschränkt	eingeschränkt	ja	ja	ja	ja	nein	mit Enterprise-Lizenz
IDE-Integration	(ja)	JetBrains IDEs, VS Code, (Vim mit Community-Plug-in)	VS Code für macOS und Linux	möglich mittels Code Sight	JetBrains IntelliJ, Visual Studio Code	nein	Visual Studio, VS Code, IntelliJ IDEA, GitHub Codespaces	nein	Eclipse, JetBrains-IDEs, VS, VS Code, Language Server (Beta)
CLI	ja	ja	ja	ja	ja	ja	ja	ja	ja
API	ja	nein	nein	ja	ja	ja	ja	ja	mit Enterprise-Lizenz
unterstützte Sprachen	Java, C/C++, .NET, Node.js, PHP, Python, Go, Ruby, Rust	Go, Java, .NET, PHP, Python, Ruby , Node.js	ca. 20	ca. 25	Java, C++, .NET, Python, PHP, Swift, Objective-C, Go, Ruby	ca. 20	über 200	Java, .NET, experimentell: Python, PHP, Node.js, Ruby, Swift	ca. 15
unterstützte Paketmanager	□	ca. 10□	ca. 20	ca. 20	ca. 15	ca. 20	ca. 30	ca. 20, davon 2□3 experimentell	ca. 15
Scan von Binärdaten	Go	nein	□nein	Java, .NET, Go	nein□	nein	ja	nein	□
Prüfung von Codeerreichbarkeit[1]	□nein	nein	nein	für Java	ja, Exploitable Path	nein	ja, Reachability Path Analysis	nein	für Java, mit Snyk Code
automatisierte Remediation	nein	nein	nein	nein	Remediation Manifests für npm	automatisierte Pull Requests	automatisierte Pull Requests	nein	automatisierte Pull/Merge Requests
Definition von Richtlinien	ja	nein	nein	ja	ja	ja	ja	ja	mit Enterprise-LIzenz
Preis	auf Anfrage (Lizenzierung nach Repositorys)	kostenlos (Open Source)	kostenlos (Open Source)	auf Anfrage	auf Anfrage	ab 104 Dollar pro Entwickler und Monat, Enterprise	ab 16 000 Euro pro Jahr (für 20 Entwickler)	kostenlos (Open Source)	ab 23 Dollar pro Entwickler und Monat, limitierte Version kostenlos, Enterprise auf Anfr.
□□□□□□□n. a. – nicht anwendbar, BB – Bitbucket; [1] Tests, ob die kompromittierte Funktion/Methode tatsächlich aufgerufen wird									

OWASP Dependency-Track

Eines der am längsten verfügbaren SCA-Tools kommt vom Open Worldwide Application Security Project: das Open-Source-Werkzeug OWASP Dependency-Track (ODT). Es gibt SBOMs im CycloneDX-Format aus und man kann sie in diesem Format auch

importieren. Für den Abgleich mit bekannten Schwachstellen nutzt das Werkzeug die National Vulnerability Database, GitHub Advisories und den Sonatype OSS Index als Datenquellen. Weitere, zum Teil kostenpflichtige Quellen können Anwender über Plug-ins freischalten. Metadaten über Abhängigkeiten gewinnt ODT aus einer Reihe von verbreiteten Paketformaten, neben den üblichen NuGet, PyPi, Maven oder NPM sind auch Cargo für Rust-Projekte oder Hex für Elixir/Erlang darunter. ODT ist nicht auf die Analyse von Abhängigkeiten und Schwachstellen beschränkt, es ermöglicht auch, Richtlinien (Policies) festzulegen, die bestimmte Pakete, Softwarelizenzen oder Software mit Schwachstellen eines definierten Schweregrades ausschließen.

ODT ist lokal installierbar. Ab Version 4 der Software sind Backend und Frontend voneinander getrennt. Das Backend – der API-Server – ist eine klassische Serverapplikation, die die API per Jetty zur Verfügung stellt und ihre Daten im lokalen Dateisystem und einer relationalen Datenbank speichert. Das Frontend ist eine Single-Page-Webapplikation. Sie stellt Dashboards und Reports dar und dient der Konfiguration. Am einfachsten ist die Installation als Docker-Container. Über ein offizielles Jenkins-Plug-in oder GitHub Actions wird ODT in die CI/CD-Pipeline integriert.

Syft und Grype

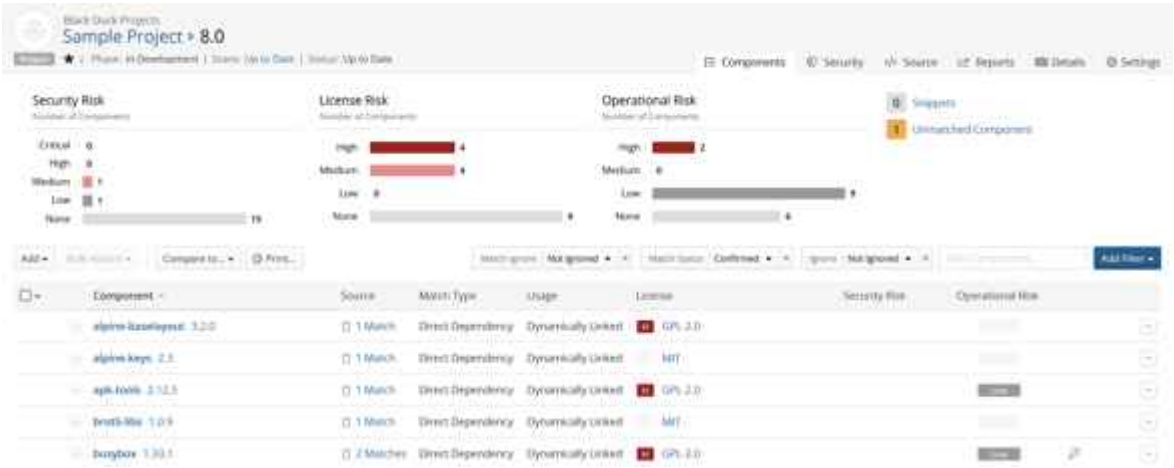
Das US-Unternehmen Anchore stellt mit Syft ein Open-Source-Tool zur Verfügung, das hauptsächlich der Erstellung von Software Bills of Materials (SBOMs) dient. Syft verfügt ausschließlich über eine Kommandozeilenschnittstelle, es stellt keine grafische Benutzeroberfläche bereit. Zusätzlich bietet Anchore das Tool Grype an, das in Verbindung mit Syft oder auch einzeln Vulnerability-Scans durchführt. Mit der Kombination von Syft und Grype lassen sich viele Anwendungsszenarien der großen kommerziellen Lösungen abdecken, wenn auch mit manuellem Konfigurationsaufwand. Für

die Integration in CI/CD-Pipelines stellen Anchore und die Community Werkzeuge zur Verfügung, beispielsweise Jenkins-Plug-ins oder GitHub Actions, selbst ein IDE-Plug-in für VS Code gibt es.

Syft und Grype lassen sich lokal installieren; die Schwachstellendatenbank kommt im SQLite-Format und wird in der Standardkonfiguration automatisch über das Netz aktualisiert. Die beiden Open-Source-Werkzeuge von Anchore haben keine eigene API, als CLI-Tools mit wohldefinierten Ausgabeformaten lassen sie sich aber prinzipiell von anderen APIs benutzen. Eine der Stärken beider Anwendungen ist die Ausrichtung auf containerisierte Applikationen.

Synopsys Black Duck

Black Duck gehört zu den am längsten verfügbaren und am häufigsten eingesetzten SCA-Angeboten am Markt. Dazu kommt, dass es nach der Übernahme des Herstellers durch Synopsys mit dessen Codeanalysewerkzeugen verzahnbar ist: So lässt sich SCA mit statischen und dynamischen Codeanalysemethoden (DAST, SAST) und Fuzzing aus einer Hand kombinieren. Es ist auf den unternehmensweiten Einsatz ausgerichtet und soll dabei helfen, in großen Projekten zentrale Sicherheits- und Compliancerichtlinien zu definieren und durchzusetzen (siehe Abbildung 1).



Die BOM-Ansicht von Black Duck listet Lizenz-, Sicherheits- und Betriebsrisiken einer Komponente gemeinsam auf. Letztere

ergeben sich zum Beispiel aus Paketen, die kaum noch gepflegt werden oder eine geringe Reputation besitzen (Abb. 1). *Synopsys*
Damit einher gehen ein umfassendes Rollen- und Berechtigungsmodell, komplexe Policies und Regelsätze, die definieren, wie mit bestimmten Risiken umzugehen ist, sowie Reportgeneratoren. Entsprechend langwierig kann die Einführung des Produkts sein. Synopsys gibt die Black Duck Security Advisories heraus und verspricht, dass seine SCA-Software viele Schwachstellen schon meldet, bevor sie in der National Vulnerability Datebase auftauchen.

Black Duck integriert sich via Plug-ins in alle verbreiteten CI/CD-Frameworks, Code- und Artefakt-Repositorys. Für die IDE-Integration ist Synopsys Code Sight zuständig, ein separates Produkt, das Black-Duck-Anwender kostenlos nutzen können.

Neben Sourcecode analysiert Black Duck Java-, .NET- und Go-Binaries, binäre Repositorys im JFrog-Artifactory- und Nexus-Format und bestimmte Firmwareformate. Bei der Codeanalyse verlässt es sich nicht nur auf die Deklarationen in den Manifesten der Pakete. Der Hersteller wirbt mit einer Multi-Faktor-Open-Source-Erkennung und integriert eine proprietäre Methode namens Codeprint, um Open-Source- und Fremdanbieter-Komponenten zu identifizieren.

Aqua Supply Chain Security

Aqua Security gilt als Spezialist für die Absicherung von containerisierten Anwendungen. Nach der Übernahme von Argon Ende 2021 – eines auf Supply Chain Security spezialisierten Start-ups aus Israel – bietet das Unternehmen mit Aqua Supply Chain Security ein Produkt an, das die wesentlichen Aspekte der SCA abdeckt und darüber hinaus weitere Sicherheitsüberprüfungen durchführt. So scannt es per statischer Codeanalyse bei Abhängigkeiten auch den Quellcode selbst, sucht nach Fehlkonfigurationen in den Build-Tools und in Infrastructure as Code. Go-Code können die Aqua-Scanner

auch in Binärform untersuchen.

Eine Besonderheit stellen die erweiterten SBOMs dar, die die Plattform erzeugen kann. Die als Next Generation SBOMs bezeichneten Dokumente sind mit zusätzlichen Informationen angereichert, etwa ob Peer-Reviews stattfanden oder ob das Code-Repository eine Zwei-Faktor-Authentifizierung verlangt. Zusätzlich soll Code Signing die Integrität sicherstellen.

Compliance- und Sicherheitsfunktionen sind integriert, Aqua Supply Chain Security eignet sich also auch zur Top-Level-Beurteilung der Risiken durch Open-Source-Einsatz im gesamten Unternehmen. Für die einzelnen Open-Source-Komponenten erstellt das Produkt einen Reputation Score, aus dem Maintenance-Zustand, der Beliebtheit, der Zahl und Schwere von Sicherheitslücken und anderen Faktoren.

Aqua vermarktet Supply Chain Security innerhalb seiner Cloud-native Application Protection Platform (CNAPP), in dessen Variante Dev Security. Es wird dort von den Komponenten Risk &Vulnerability Scanning sowie Advanced Malware Protection ergänzt.

Aqua Trivy

Ein Kernbestandteil von Aqua Supply Chain Security ist der Security-Scanner Trivy, der als separates CLI-Tool vor allem in der Container-Welt häufig eingesetzt wird. Für sich genommen ist er zwar kein vollwertiges SCA-Produkt, aber er ist Open Source und deckt so viele SCA-Aspekte ab, dass er in Kombination mit ein paar Skripten und anderen Open-Source-Werkzeugen die Grundlage für eine kleine, flexible, selbst gebaute SCA-Lösung sein kann. Trivy ist kein reiner Container-Scanner, sondern kann auch Code in Git-Repositorys, auf dem lokalen Filesystem oder in Images virtueller Maschinen prüfen. Er identifiziert dort bekannte Schwachstellen, findet Abhängigkeiten, Konfigurationsfehler und sensible Informationen wie Zugangsdaten. Außerdem identifiziert er

Open-Source-Lizenzen. Seit Kurzem kann Trivy auch SBOMs im SPDX- oder CycloneDX-Format erzeugen.

Trivy ist ein reines Kommandozeilenwerkzeug und somit automatisierungsfreundlich. Aqua Security stellt sogar selbst Integrationen für GitHub Actions und Azure DevOps zur Verfügung. Trivy bringt seine eigene kompakte Schwachstellendatenbank mit, bei gefundenen Lücken verlinkt er in der Ausgabe auf den entsprechenden Eintrag in der Aqua Vulnerability Database, die auch das kommerzielle Produkt Aqua Supply Chain Security nutzt.

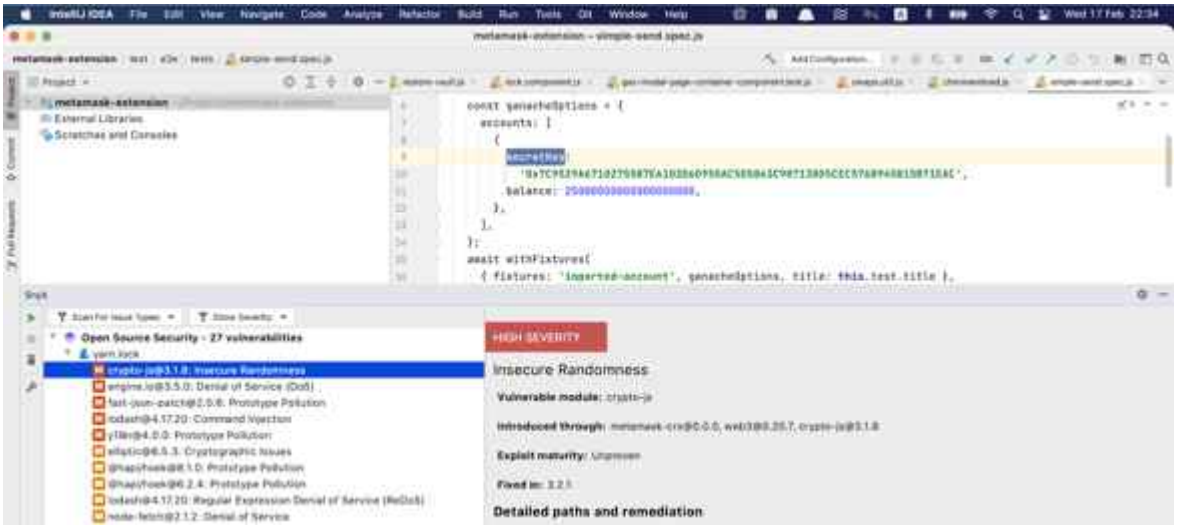
FOSSA

Das Produkt FOSSA (Free Open Source Software Analysis) des gleichnamigen Anbieters bezeichnet dieser als Open Source Risk Management Platform. Sein Schwerpunkt liegt darauf, rechtliche und Sicherheitsrisiken gemeinsam zu betrachten und die Nutzung von Open Source unternehmensweit durch Richtlinien abzudecken. Eine zentrale Policy Engine soll Rechts- und Entwicklungsabteilungen bei der gemeinsamen Ausarbeitung dieser Richtlinien unterstützen und garantieren, dass sie im Softwarelebenszyklus durchgesetzt werden. FOSSA wirbt mit rechtssicheren, auditfähigen Berichten und automatisierten Risikobewertungen, die beispielsweise den Due-Diligence-Prozess bei Firmenübernahmen beschleunigen sollen. Für DevOps-Teams bietet FOSSA neben Integrationsmöglichkeiten in alle relevanten CI-Produkte auch eine generische CI-Schnittstelle für individuelle Pipelines an, es scannt Container nach OCI-Standard und unterstützt rund 20 verbreitete Programmiersprachen.

Snyk Open Source

Snyk vereint mehrere Produkte auf einer Plattform. Für SCA zuständig ist die Komponente mit dem Namen Snyk Open Source, daneben bietet Snyk Code eine statische Codeanalyse. Snyk Container und Snyk Infrastructure as Code sind weitere

Komponenten. Snyk ist in erster Linie ein SaaS-Anbieter. In dieser Variante sind die Komponenten auch einzeln buchbar. Eine Enterprise-Lizenz umfasst immer alle Produkte; sie ist auch Voraussetzung, um Features nutzen zu können, die zu einem umfassenden SCA-Produkt gehören, wie Lizenzcompliance, Verwaltung von Richtlinien und Erstellung von Berichten sowie die Option, auf den Unternehmensservern gehostete Code-Repositorys einzubinden.



Mit einem Plug-in für JetBrains-IDEs informiert Snyk schon beim Schreiben des Codes über Sicherheitslücken in den eingebundenen Open-Source-Bibliotheken (Abb. 2). *Snyk*

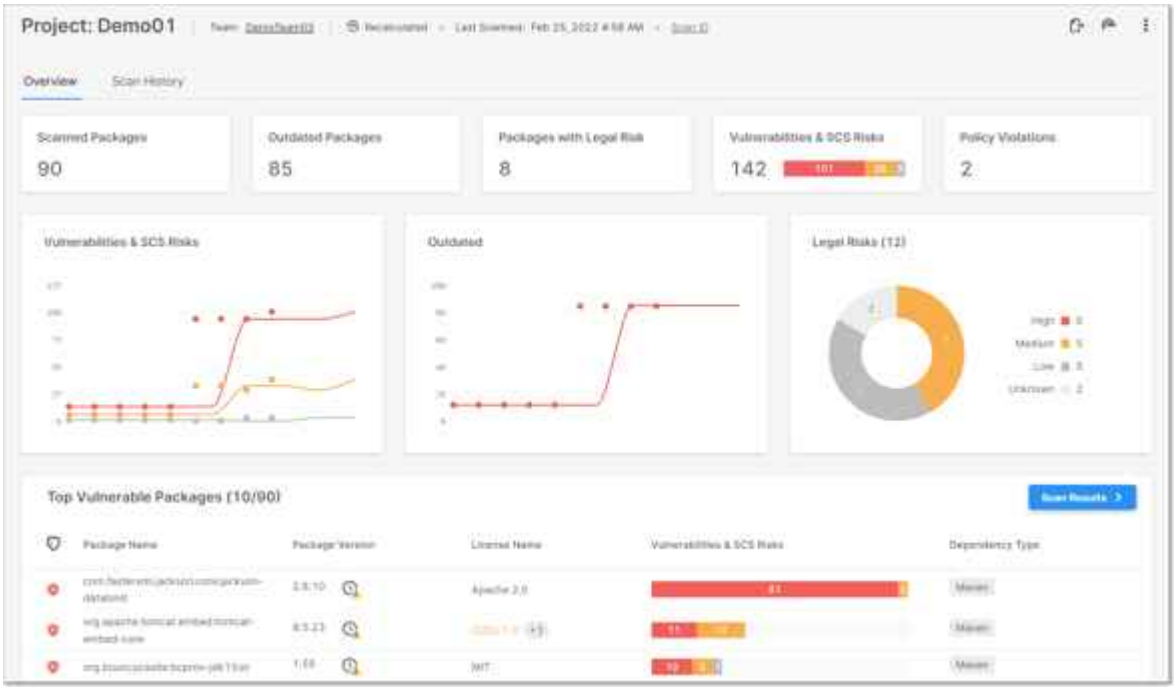
Eine vollständige On-Premises-Installation bietet Snyk nicht an – die Enterprise-Variante erlaubt es aber, ein als Snyk Broker bezeichnetes Proxysystem einzurichten, das Codescans lokal ausführt und die Kommunikation mit den Snyk-Servern über einen Tunnel absichert. Mit einem normalen SaaS-Abo erfolgt der Scan auf den Servern von Snyk.

Snyk Open Source lässt sich mit IDEs, den gängigen CI/CD-Tools und Git-basierten Repositorys verknüpfen (siehe Abbildung 2). Im Vergleich zu anderen Produkten ist vor allem die IDE-Unterstützung gut ausgebaut. So schlägt zum Beispiel das JetBrains-Plug-in mit einer als Open Source Advisor bezeichneten Funktion geeignete Open-Source-Pakete vor und bezieht dabei Popularität, Maintenance-Zustand und Bewertungen der Community ein.

Der Umgang mit SBOMs zählte lange Zeit nicht zu den Stärken von Snyk. Kürzlich hat das Unternehmen aber angekündigt, dass die API und das Kommandozeilenwerkzeug künftig SBOMs in den CycloneDX- und SPDX-Formaten exportieren sollen. In der aktuellen Betaversion der API ist das Feature bereits zu finden.

Checkmarx SCA

Checkmarx ist ein 2006 in Israel gegründetes IT-Security-Unternehmen, dessen Sicherheitsforscher wiederholt wichtige Schwachstellen aufgedeckt haben und federführend an der Erstellung der OWASP API Top Ten beteiligt sind. Erstes Produkt der Firma war CxSAST, ein Werkzeug zur statischen Codeanalyse, Checkmarx SCA (CxSCA) kam erst 2020 hinzu. Wie alle der größeren Anbieter betreibt Checkmarx seine eigene Schwachstellendatenbank, zusätzlich dazu auch eine Datenbank bösartiger Pakete, die gezielt dafür entwickelt werden, Softwareprojekte zu infiltrieren.



Dashboards wie hier bei Checkmarx gehören zur Grundausstattung aller umfangreicheren SCA-Tools (Abb. 3). *Checkmarx*

Checkmarx SCA ist Teil von Checkmarx One, dem integrierten Hauptprodukt des Herstellers, das von diesem als Application Security Testing Platform bezeichnet wird. CxSCA kann aber

auch separat lizenziert werden. Am günstigsten ist die Nutzung als Managed Service, optional ist der Betrieb in einer Private-Cloud-Umgebung oder vollständig on Premises möglich. Checkmarx SCA implementiert eine Methode namens Exploitable Path, die im Sourcecode des Projekts danach sucht, welche Funktionen in den Abhängigkeiten tatsächlich aufgerufen werden. Laut Hersteller funktioniert das für jede Programmiersprache, die sich mit CxSAST untersuchen lässt. Bei Scans über die SCA-Website lädt das Tool auch den Sourcecode hoch und dort bleibt er für bis zu 24 Stunden gespeichert. Ein Resolver kann Abhängigkeiten aber auch on Premises ermitteln und schickt diese Daten dann an die Plattform zur Risikoanalyse.

Bei Verwendung von Agents oder des Resolvers gelangen nur Metadaten, Manifestdateien und Fingerprints des Sourcecodes auf die Checkmarx-Server. Zu den Metadaten zählt Checkmarx auch sämtliche Dateinamen. Daten landen in einem verschlüsselten S3-Bucket, Sourcecode wird höchstens 24 Stunden aufbewahrt.

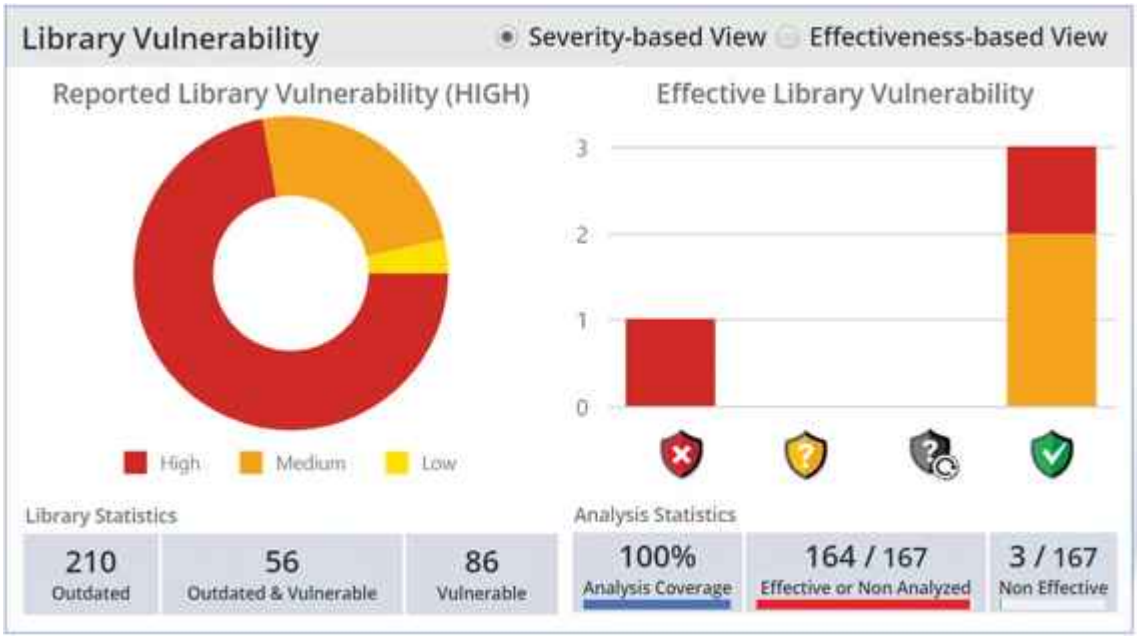
Mend SCA

Mend, vormals Whitesource, ist ein weiterer Hersteller im Umfeld der Anwendungssicherheit, der seine Wurzeln in Israel hat. Hier war das SCA-Produkt zuerst da, SAST kam später hinzu. Mend entwickelt auch den von Entwicklern viel gelobten Renovate Bot, ein Open-Source-Werkzeug zur automatischen Aktualisierung von Dependencies. Diesen wird iX in einer der kommenden Ausgaben vorstellen.

Mend sammelt Schwachstellen und Security Advisories aus zahlreichen Quellen in einer eigenen Datenbank und scannt auch Software, die in den Manifesten der Paketmanager nicht deklariert ist. Eine der Stärken des Produkts ist das Bewertungssystem von Schwachstellen (siehe Abbildung 4). Hier berücksichtigt Mend vor allem, ob der eigene Code verwundbare Funktionen aufruft (Reachable Path Analysis). Aber auch

andere, nicht direkt die Schwachstelle selbst betreffende Faktoren, die insgesamt die Auswirkungen auf die Geschäftstätigkeit widerspiegeln sollen, gehen ein.

Damit gehen entwicklerfreundliche Benachrichtigungs- und Remediation-Möglichkeiten einher. Ist Mend SCA in ein Repository integriert, kontrolliert es bei jedem Commit den Code auf vom Entwickler eingebaute Schwachstellen, Vulnerabilities in verwendetem Open-Source-Code und Lizenzverletzungen. Das Tool öffnet Pull Requests mit einem Upgrade des Pakets auf eine nicht verwundbare Version. Im einfachsten Fall ist somit die Schwachstelle mit einem Klick aus dem Abhängigkeitsbaum verschwunden.



Mend priorisiert Schwachstellen anhand verschiedener Metriken. Eine davon ist die Erreichbarkeit des Codes von der eigenen Anwendung aus (Abb. 4). *Mend*

Seine IDE-Pug-ins nennt der Anbieter Mend Advise, es gibt sie für IntelliJ Idea, WebStorm und PyCharm von JetBrains, für Visual Studio und VS Code sowie für Eclipse. Eine clevere Idee ist eine Browsererweiterung, die beim Stöbern auf Stack Overflow oder GitHub auf Sicherheitsrisiken in den gerade dargestellten oder erwähnten Komponenten hinweist.

Compliance- und Sicherheitsrichtlinien kann Mend SCA ebenfalls über entsprechende Regelwerke definieren und durchsetzen –

insgesamt stehen bei diesem Produkt aber eher die Bedürfnisse der Developer als die der Rechtsabteilung im Vordergrund. In den letzten Monaten hat Mend seine API um einen SBOM-Export erweitert, vorher musste man SBOMs mit einem Tool aus dem internen Softwareinventarformat erzeugen. Jetzt lässt sich der Prozess automatisieren.

Weitere Anbieter

Contrast SCA ist Teil der vor allem im Java-Umfeld verbreiteten Secure Code Platform des Herstellers. Sie verfolgt den Ansatz, Agenten in den Code einer Anwendung zu integrieren, die im laufenden Betrieb Schwachstellen identifizieren. Diese Agenten liefern auch Informationen zu den verwendeten Open-Source-Komponenten, aus denen die Plattform Schwachstellen identifiziert und detaillierte SBOMs generiert. Neben Java unterstützt Contrast weitere Sprachen und Plattformen etwa .NET, Python, Ruby und Go.

Die kanadische Firma **MergeBase** bewirbt ihr SCA-Produkt mit niedriger Falsch-positiv-Rate und Laufzeitüberwachung des Produktivcodes. Als SaaS ist MergeBase relativ günstig (ab 38 US-Dollar pro Entwickler), Enterprise-Varianten lassen sich auch on Premises installieren. Der Funktionsumfang ist mit dem von Snyk vergleichbar.

Revenera FlexNet Code Insights lädt entweder die gesamte Codebasis eines Projekts zum Scannen auf den Server oder verbindet den Scanserver mit einem Software-Repository, das er dann automatisch nach Vorgaben scannt. Im Unterschied zu Werkzeugen, die auf die Cloud und DevOps-Prozesse ausgerichtet sind und sich an verschiedene andere Tools andocken, hat FlexNet Code Insight eine eher konservative Herangehensweise: Das System dient als „Single Source of Truth“ für den gesamten Code des Projekts, erstellt SBOMs und identifiziert Schwachstellen.

Unternehmen, die bei ihren Artefakt-Repositorys auf JFrog

setzen, können mit **JFrog XRay** die dazu passende SCA-Lösung einsetzen, die eine native Artifactory-Anbindung bietet und Zugriff zu sämtlichen Metadaten im Repository hat und auch Binaries scannt. XFrog XRay identifiziert Lizenzen und Schwachstellen, erlaubt die Definition von Policies und exportiert SBOMs, JFrog pflegt eine Schwachstellendatenbank, die sich aus der VulnDB und eigenen Einträgen speist. Mit dem FrogBot lässt sich JFrog XRay auch in GitHub-Repositorys einbinden.

Veracode kombiniert SCA mit statischer Codeanalyse. Bei Letzterer versteht es auch Cobol, PRG oder verschiedene SQL-Dialekte, ist also auch im traditionellen IT-Umfeld zu Hause. **Veracode SCA** kommt mit 13 verbreiteten moderneren Sprachen und den entsprechenden Paketformaten zurecht. Es ist ein umfangreiches, sowohl auf Security als auch auf Compliance ausgerichtetes SCA-Produkt, das alle entscheidenden Funktionen und Integrationsmöglichkeiten mitbringt.

Die Nexus-Plattform von Sonatype ist bei Cloud-Entwicklern vor allem für ihr Artefakt-Repository bekannt, das direkt mit JFrog Artifactory konkurriert. Sicherheitsforscher kennen Sonatype eher wegen seiner Schwachstellendatenbank. Mit **Nexus Lifecycle** hat das Unternehmen ein SCA-Produkt im Angebot, das zwar auf seine übrigen Securityprodukte abgestimmt, aber nicht auf Anwender der Nexus-Repositorys beschränkt ist. Nexus Lifecycle ist ein umfassendes Produkt für den Enterprise-Einsatz.

Fazit

Für die Auswahl einer der großen kommerziellen Lösungen ist auf jeden Fall eine genaue Anforderungsanalyse sowohl seitens der Entwickler und des Sicherheitsteams als auch – wenn Complianceaspekte wichtig sind – der Rechtsabteilung notwendig. Sehr empfehlenswert zur Vorbereitung ist der 13-seitige „Open Guide to Evaluating Software Composition Tools“ der Linux Foundation, der die wichtigsten Metriken

identifiziert und dabei hilft, ihre Relevanz für das eigene Projekt oder Unternehmen einzuschätzen.

Eine längere, gut geplante Testphase vor der Lizenzierung des Produktes ist unabdingbar und bei allen seriösen Anbietern möglich. Bei Herstellern, die ihre komplette Nutzer- oder Administrationsdokumentation frei verfügbar machen, lassen sich einige Anforderungen schon vorher klären, denn nicht selten zeigen die Dokumente, wie die in Fact Sheets beworbenen Features tatsächlich funktionieren, oder sie decken Einschränkungen auf.

Zu beachten ist auch, dass bei möglicherweise schnell eingekauften SaaS-Angeboten Sourcecode und Metadaten das Unternehmen verlassen können und unter Umständen auf US-Servern landen. Im Sinne der DSGVO dürfte das meist zwar unproblematisch sein, da es sich nicht um personenbezogene Daten handelt. Aber das eine oder andere Unternehmen hat vielleicht doch gute Gründe, den Sourcecode lokal zu halten – speziell, wenn es um Auftragsentwicklung geht. Zum Glück gehen die meisten Anbieter mit Informationen, wo und wie lange Kundendaten gespeichert werden, recht transparent um.

Aus technischer Sicht essenziell ist, dass sich das SCA-Produkt an möglichst viele der im Unternehmen eingesetzten Entwicklungs- und Deployment-Werkzeuge anbinden lässt – am besten auch an solche, die für später auf der Wunschliste stehen. Kleinere Integrationen lassen sich über die API nachrüsten.

Darüber hinaus ist eine niedrige Falsch-positiv-Rate bei den gemeldeten Schwachstellen wichtig, damit das Werkzeug den Entwicklern nicht im Weg steht. Idealerweise kommt eine Überprüfung dazu, ob der Code mit der Schwachstelle überhaupt aufgerufen wird. Dieses Feature ist unter verschiedenen Namen (Reachable Path, Exploitable Path etc.) bei Anbietern verfügbar, die auch SAST-Produkte im Portfolio haben, manchmal jedoch nur für ausgewählte Sprachen.

Eine gute Integration in IDEs ist ein großes Plus, denn so verhindert man, dass Schwachstellen überhaupt den Weg in den Code finden und nicht erst beim Einchecken in das Repository oder noch später auffallen. Automatisierung und permanente Überwachung der CI-Pipelines sollte möglich sein.

Schwieriger wird es, wenn das Werkzeug dazu benutzt werden soll, unternehmensweite Policies durchzusetzen und Complianceanforderungen zu überwachen. Dann bringt ein Test der Software innerhalb eines Entwicklerteams keinen nennenswerten Erkenntnisgewinn. Hier könnte ein abteilungsübergreifendes Projektteam die Anforderungen möglichst genau spezifizieren und nach einer sinnvollen Vorauswahl eine kleine Zahl von Anbietern genauer unter die Lupe nehmen.

Wenn es darum geht, überhaupt erstmalig werkzeuggestützte Software-Composition-Analyse zu betreiben, ließe sich alternativ in einem Developer-Team ein eher an den Bedürfnissen der Entwickler ausgerichtetes Produkt einführen. Es muss aber zumindest von seinen Spezifikationen her den Compliancebereich mit abdecken könnte und ginge erst nach positiven Erfahrungen der Developer in den unternehmensweiten Einsatz. Auch ein nicht ganz optimales Werkzeug zur Ermittlung von Risiken durch Open-Source-Software sichert die Softwarelieferkette besser ab als gar keines. (ulw@ix.de)

1. Quellen
2. Udo Schneider; SBOMs – Stücklisten für Software; iX 10/2022, S. 54
3. Weitere Infos zu Tools und Auswahlkriterien: ix.de/zvbm