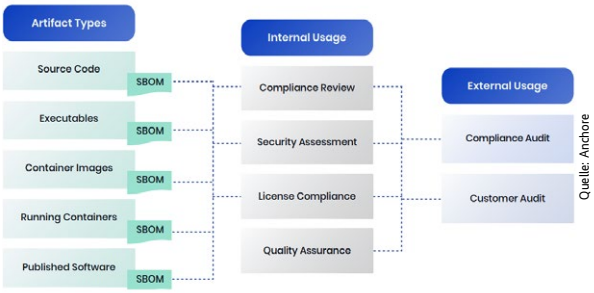
Syft: Sicherheit für die Softwarelieferkette



Quelle: Anchore

Syft erstellt Software Bills of Materials von komplexen Softwarearchitekturen in Containern und hilft, Schwachstellen zu finden. Diese entstehen etwa durch Komponenten, die als nicht genannte Abhängigkeiten des eigentlichen Produkts unerkannt bleiben.

Von Jürgen Schuck

Ein breites Angebot von Open-Source-Komponenten, agile Methoden und DevOps-Konzepte haben die Entwicklungszyklen von Software stark beschleunigt. Die Bereitstellungsintervalle verkürzen sich dadurch ebenfalls und Deployments erfolgen serverseitig häufig als Container, die nicht selten Hunderte verschiedene Softwarekomponenten unterschiedlichster Herkunft enthalten. Jede einzelne stellt ein potenzielles Sicherheitsrisiko dar, das sich durch die genaue Kenntnis der Komponenten in einem Container entscheidend reduzieren lässt.

Syft von Anchore ist ein solches Werkzeug und als Open Source verfügbar. Es erzeugt eine Software Bill of Materials (SBOM) mit ausführlichen Informationen zur Zusammensetzung eines Softwarecontainers. Der Begriff Bill of Materials stammt aus der Fertigungsindustrie und bezeichnet eine detaillierte Liste aller Materialien zur Herstellung eines Produkts. Analog dazu ist eine Software Bill of Materials eine Liste der Komponenten in einem Softwareprodukt.

Es gibt Syft für Linux, macOS und Windows. Das Repository auf GitHub stellt Releases bereit, die jeweils die ausführbaren Programme für alle berücksichtigten Betriebssysteme und Prozessorarchitekturen enthalten. Die Installation erfordert lediglich, das aktuelle Binary ins lokale Dateisystem zu kopieren. Syft arbeitet auf der Kommandozeile: Der Programmname syft startet das Programm, falls das Installationsverzeichnis in der Umgebungsvariablen `PATH` enthalten ist.

Die Bedienung von Syft erklärt das Wiki; der Link dazu steht im Readme des Repositorys (siehe ix.de/zm7z). Das Kommando `syft --help` listet die Optionen. Das Programm erwartet einen Parameter mit einer Quelle, für die es eine SBOM erstellt. Ohne zusätzliche Angaben fragt Syft zunächst die Daemons von Docker, Podman und containerd nacheinander nach einem Image für die angegebene Quelle. Mit der zusätzlichen Option `--from` lässt sich der Quelltyp bestimmen. Syft beherrscht neben Images, die die Container-Daemons bereitstellen, auch die Typen OCI (Open Container Initiative) und SIF (Singularity Image Format) sowie einige mehr für Archive (tar) und Verzeichnisstrukturen.

Das Programm analysiert die angegebene Quelle und listet die enthaltenen Softwarekomponenten tabellarisch auf `stdout`. Neben den Namen und Versionsnummern enthält die Liste die Pakettypen, denen Syft die Komponenten zuordnet, beispielsweise Debian, Java, npm, Python und Rust. Syft bezeichnet Pakettypen als Ecosystems.

SBOM-Formate für die automatisierte Generierung

Mit der Option `--output` kann man weitere Ausgabeformate einstellen, die detailliert über die Komponenten in einem Container informieren, beispielsweise zu Lizenzen und den Imagelayern, in denen Syft eine Komponente gefunden hat. Zu diesen Formaten zählen CycloneDX und System Package Data Exchange (SPDX), zwei Standards für SBOM-Formate. Da sie die Inhalte in JSON beschreiben, eignen sie sich für die automatisierte Verarbeitung, etwa durch Skripte in einer CI/CD-Pipeline. CycloneDX hat seine Stärken in der Detailtiefe von Informationen zu Softwarekomponenten und ihren Abhängigkeiten. Bei SPDX bilden unter anderem Lizenzinformationen einen Schwerpunkt.

Auch eigene Ausgabeformate sind möglich. Da Syft in Go geschrieben ist, lassen sich mit Kenntnissen der Programmiersprache Templates für spezielle Anforderungen definieren. Die Namen der erforderlichen Variablen für die Ausgabedaten finden sich in einer Ausgabe im proprietären Format `syft-json`, das den größten Detaillierungsgrad und damit den gesamten Vorrat an Informationen bietet, die Syft in einem Container ermitteln kann.

Syft exportiert eine Reihe von Funktionen, die als APIs in eigenen Programmen verwendbar sind. Auch die API erfordert Kenntnisse von Go und einen gewissen Willen zur Umsetzung eigener Projekte, denn als Dokumentation dienen ausschließlich Programmbeispiele. Die Dokumentation zum Erstellen eigener Templates ist ebenfalls eher dürftig und besteht überwiegend aus Beispielen (siehe ix.de/zm7z).

Die Arbeitsweise von Syft können Entwicklerinnen und Entwickler in YAML konfigurieren; sie ist im Repository via kommentierter Konfigurationsdatei ausführlich beschrieben. Interessant ist eine Option, mit der Syft zur Analyse mehrere Scanner parallel ausführt. Und auch das Ausblenden bestimmter unkritischer, aber massenhaft in einem Container enthaltener Dateien kann die Analyse beschleunigen.

Mit Syft erstellte SBOMs, versioniert abgelegt in einem Repository und eingebettet in den Softwarebereitstellungsprozess, ermöglichen automatisierte Compliancеprüfungen der Komponenten und fördern damit die Transparenz eines Softwareprodukts und die Nachvollziehbarkeit von Änderungen in der Lieferkette. Softwarekonsumenten können SBOMs – ebenfalls automatisiert – beispielsweise während des Deployment-Prozesses mit denen ihrer Lieferanten vergleichen und so gewährleisten, dass sie korrekte Software verwenden. (nb@ix.de)

Quellen

Dokumentation und Readme: ix.de/zm7z

JÜRGEN SCHUCK

arbeitet bei der Materna Information & Communications SE als IT-Projektleiter für Behörden und Unternehmen.