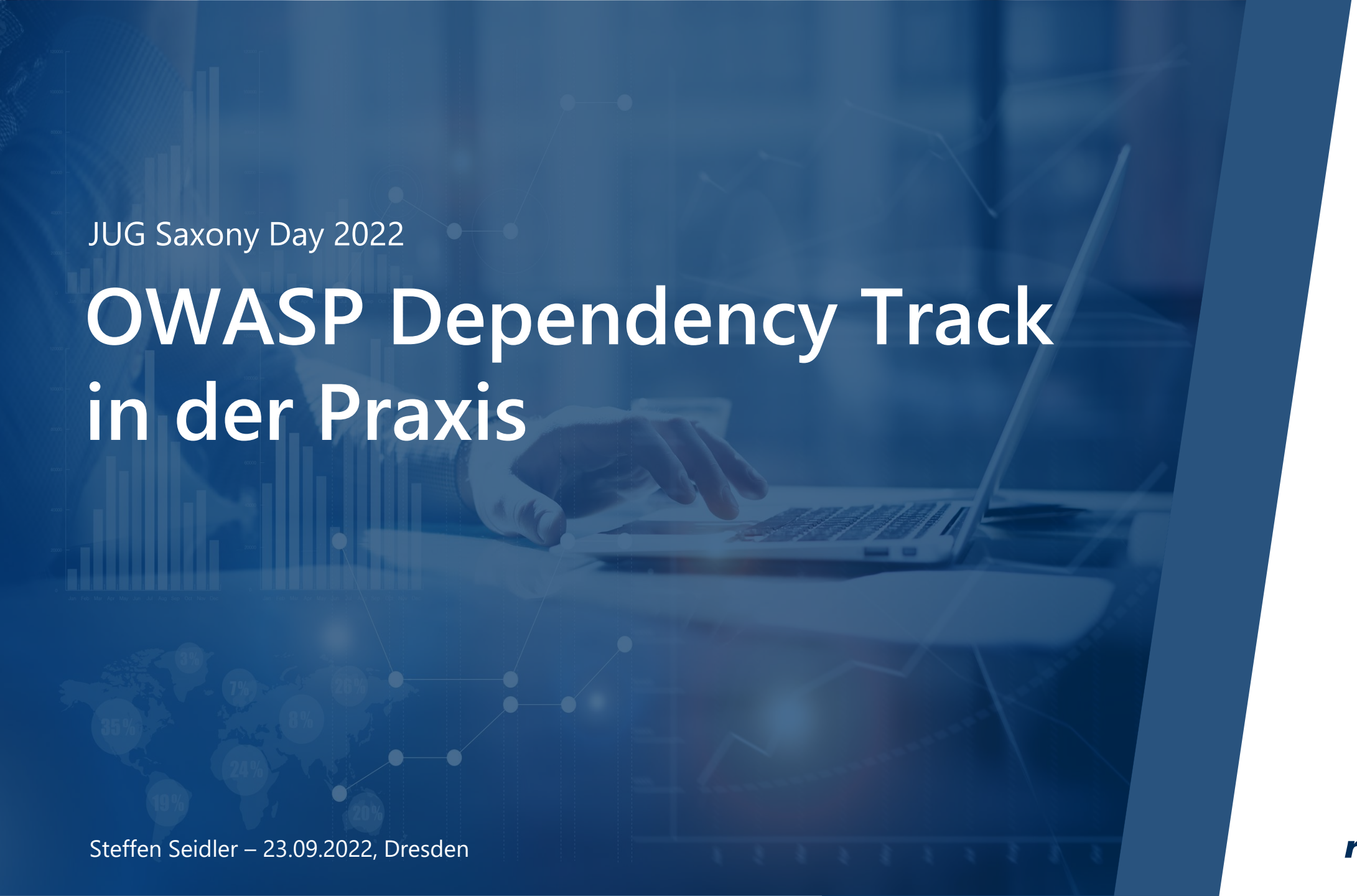


JUG Saxony Day 2022

OWASP Dependency Track in der Praxis

Steffen Seidler – 23.09.2022, Dresden

robotron®

Zur Person

Steffen Seidler

Dipl.-Mathematiker

Senior-Softwareentwickler (Geschäftsbereich Energiewirtschaft)

- SCRUM-Master
- T.P.S.S.E.-zertifiziert
  (TeleTrusT Professional for Secure Software Engineering)

Fokus auf sichere Softwareentwicklung im Gesamtprozess des Software Development Life Cycle

Roadmap

Risiken in der Software Supply-Chain

- Grundregeln bei Fremdbibliotheken

OWASP Dependency Track

- Funktionsweise und Features
- Herausforderungen der Integration

Unsere Integrationsvariante

Risiken in der Software Supply-Chain

Risiken bei der Sicherheit der Software Supply-Chain immer häufiger und vielfältiger

- Aufdeckung von Bugs
- Verwaisen von Fremdbibliotheken
- Hijacking von Fremdbibliotheken
- Einschleusung von Schwachstellen durch unzureichend geprüfte Commits
- Mutwillige Sabotage

- Oder schlicht Aufdeckung von Missbrauchspotenzials von (vermutlich gut gemeinter) Funktionalität wie bei Log4J im Dezember 2021

Grundregeln

IT-Sicherheit beim Umgang mit Fremdbibliotheken

Sorgfalt bei der Auswahl der Abhängigkeiten

- Sorgfältige Prüfung
  - Was wird wirklich gebraucht?
  - Was bekommt man eventuell unerwünscht mitgeliefert?
  - Risiko-Analyse zur Einbindung der Fremdbibliothek
    - Ist das Projekt noch aktiv? Wer ist Maintainer? (Company/Community/Einzelperson)
    - Wie stabil ist die Implementierung/API?

Überwachung und Ernstfall-Management

- Regelmäßige Prüfung aller Abhängigkeiten auf neue (Sicherheits-)Risiken und insbesondere neue Schwachstellen
- Ermittlung betroffener Projekte/Produkte/-versionen/Systeme/Kunden

OWASP Dependency Track

(Teil-)Automatisierung der Überwachung

robotron®

OWASP Dependency Track

Funktionsweise

Analogien zum älteren OWASP Dependency Check

- Verwendung der öffentlichen Schwachstellen-Datenbank NVD (National Vulnerability Database) von NIST
  - Matching per CPE (Common Platform Enumeration)

Regelmäßige Prüfung des hinterlegten Software Portfolios mit Abhängigkeiten auf bekannte Schwachstellen

- Zusätzliche Quellen: GitHub Advisories, Sonatype OSS Index, VulnDB
- Unterstützung zahlreicher Repositories diverser Sprachen zur Prüfung auf neuere Versionen
  - Maven (Java), NPM (JavaScript), …

OWASP Dependency Track

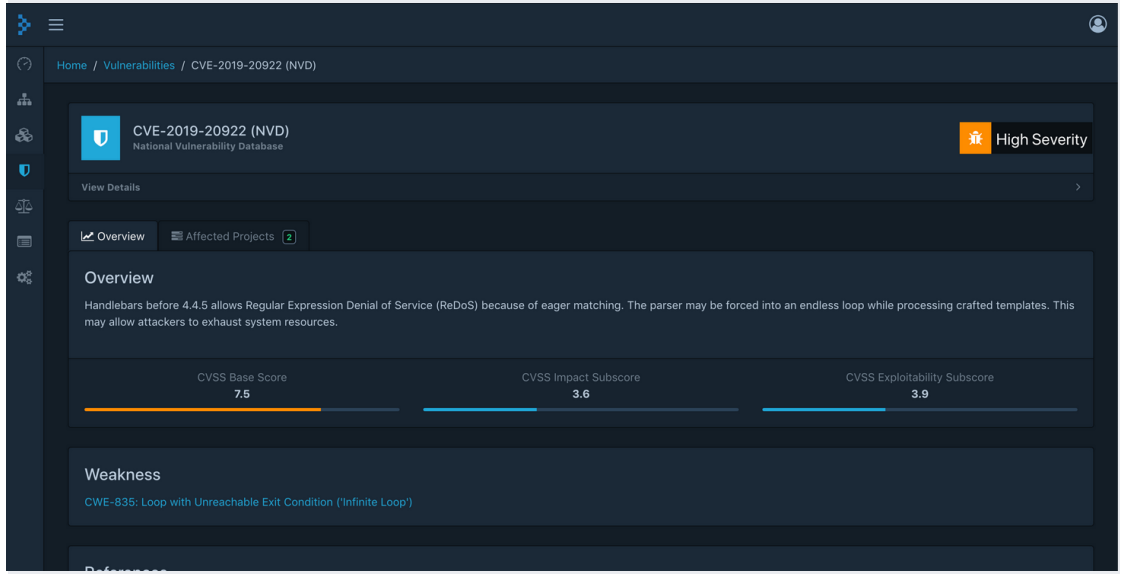
Ansichten

Portfolio Übersicht

Bedrohungslage im Überblick

Schwachstellendetails

Informationen zur Bewertung der Schwachstelle



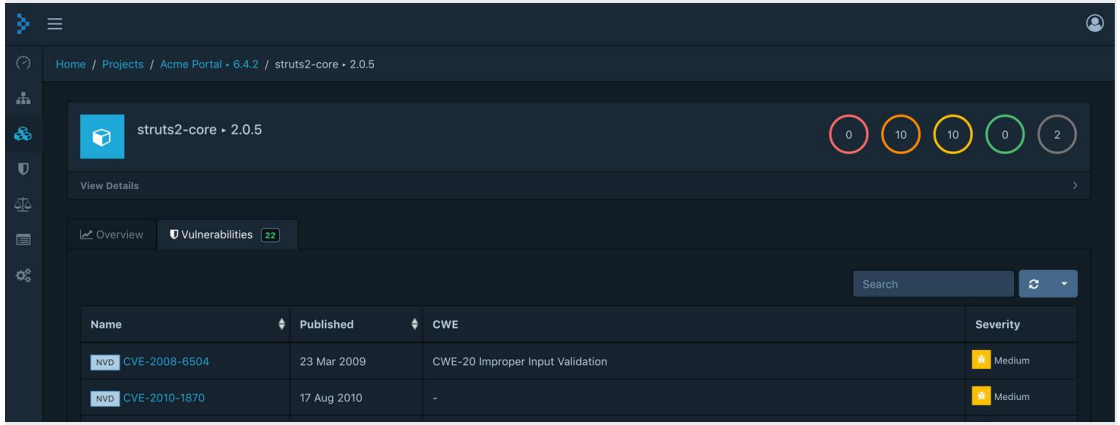
robotron®

OWASP Dependency Track

Ansichten

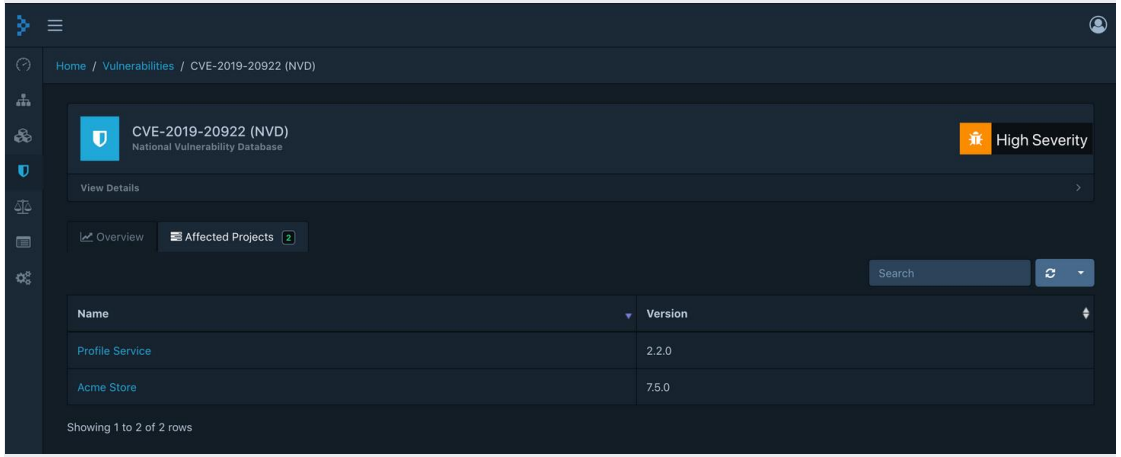
Produktsicht

Bekannte Schwachstellen im Projekt/Produkt



Schwachstellensicht

Betroffene Projekte/Produkte bei einer bekannten Schwachstelle



robotron®

OWASP Dependency Track

Im Ernstfall - Generell

Kühlen Kopf bewahren und strukturiert vorgehen

Analyse der Schwachstelle

Worum geht es? / Wie ist der Angriffsvektor?

Was ist der Impact?

Bewertung

Relevanz und Kritikalität für die eigenen Projekte/Produkte/Prozesse

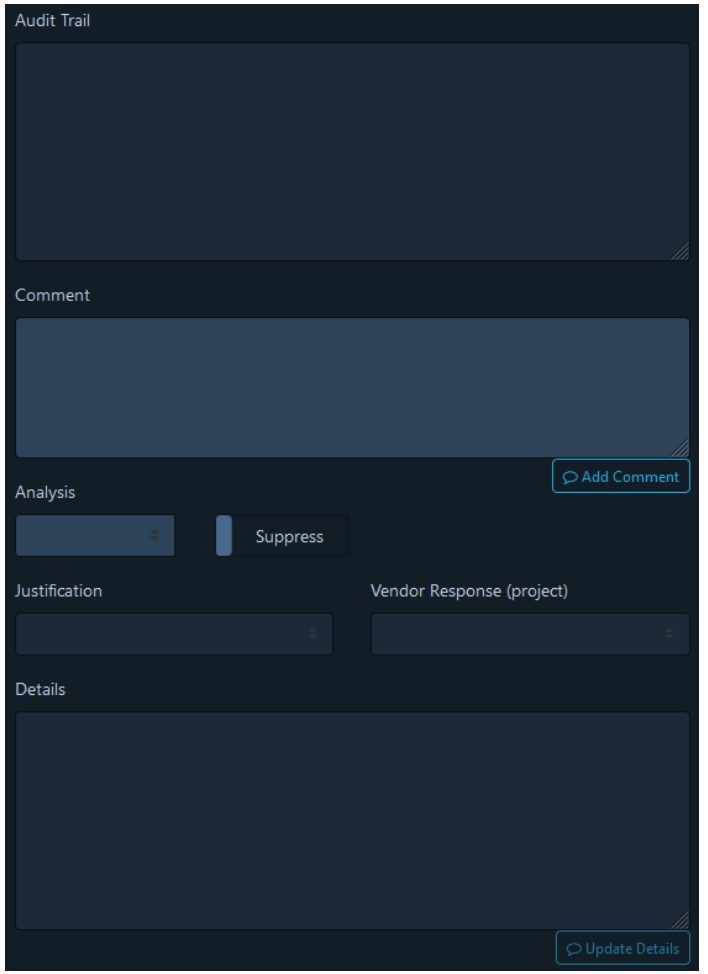
Maßnahmen bestimmen und einleiten

- **Es bleiben manuelle Schritte, denn die Einbindung verrät nichts über die tatsächliche Verwendung**

OWASP Dependency Track

Im Ernstfall - Unterstützung

- **Audit-Trail zur Dokumentation**
  - Einschätzung der Verwundbarkeit
  - Möglichkeit zur Unterdrückung
  - Begründung für die Entscheidung
    - Insbesondere bei Analyseergebnis „Nicht betroffen“
- **Alarmmeldungen**
  - Email, Slack, Teams, WebEx, WebHook
- **Anbindung an Aggregationssysteme**
  - Fortify SSC, Kenna Security, ThreadFix



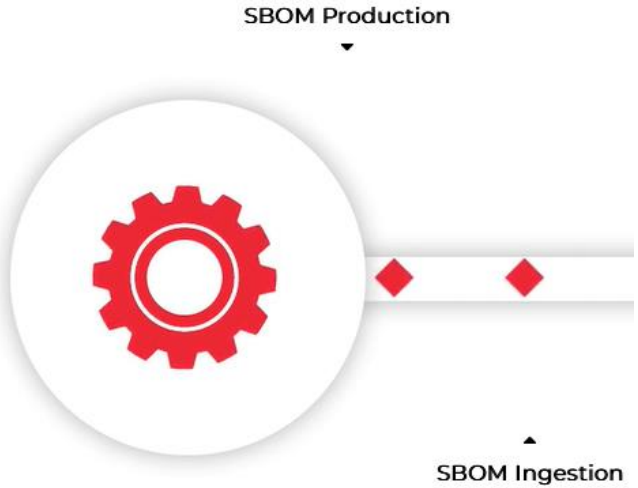
robotron®

OWASP Dependency Track

Herausforderungen der Integration - SBOM

SBOM (Software Bill Of Materials)

- Liste aller Abhängigkeiten (insbesondere transitive)
- Benötigtes Format CycloneDX von OWASP
- Erstellung
  - Manuell
  - Im Buildprozess
- Verarbeitung per CLI und Bibliotheken
- Verwaltung und Versionierung
- Übertragung and Dependency Track
- **Gute Nachricht: Zahlreiche Tools und Plugins im CycloneDX Tool Center**



Mit der U.S. Executive Order 14028 vom 12.05.2021 sind SBOMs in den USA sogar verpflichtend bei Verträgen mit der US-Regierung.

robotron®

OWASP Dependency Track

Herausforderungen der Integration - SBOM

Schlechte Nachricht: SBOM Generierung ist keine Magie!

- Sammlung der Informationen aus den Artefakten
- Analyse und Strukturierung der enthaltenen Informationen
- Erstellung einer rudimentären SBOM

Fehlende oder falsche Daten im Artefakt → SBOM möglicherweise unbrauchbar

- CPE/PackageURL in der Regel nicht enthalten → Versuch per Pattern zu generieren (geraten)
  - Schlechtes Matching vor allem Framework-Komponenten, da bei CVEs häufig das Framework an sich per CPE referenziert wird
- Lizenzinformationen fehlen oder mit kleineren Typos → Matching mit Lizenzpool schlägt fehl

➢ **Manuelle Prüfung und Korrektur fast immer notwendig!**

OWASP Dependency Track

Herausforderungen der Integration - SBOM

Pflege und Korrektur direkt in Dependency Track nicht sinnvoll!

- Änderungen wirken sich nicht immer wie erwartet aus
  - Ablage der Komponenten gemäß SBOMs → einzelne Einträge für jedes Produkt
    - Hoher Anpassungsaufwand bei manueller Pflege in Dependency Track
  - Korrektur von CPEs
    - Alte CVEs (zur falschen CPE) verbleiben
- Alternativ SBOMs korrigieren und neu einspielen (nur die letzte SBOM relevant)

Ansatz eines vorgelagerten Hint-Systems analog zu OWASP Dependency Check

OWASP Dependency Track

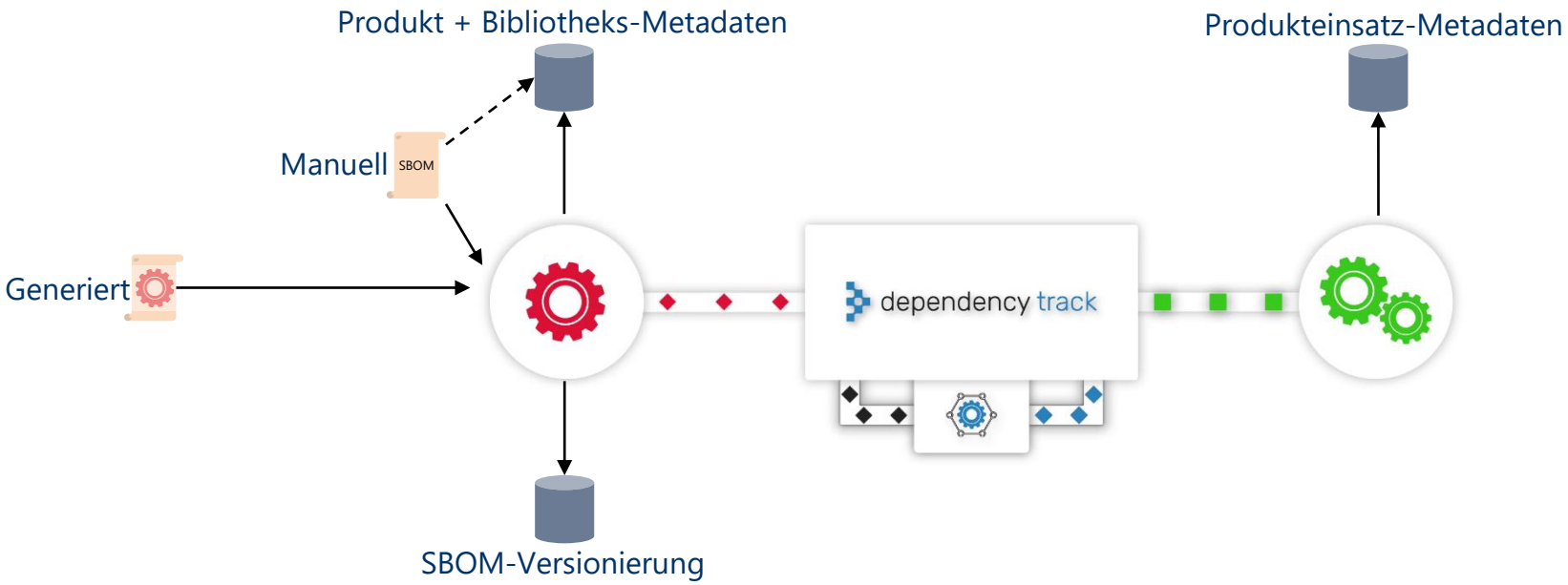
Unsere Integrationsvariante

Besonderheiten und Anforderungen

- Integration in bestehende Prozesse
- Nutzung von bestehenden Metadaten
  - Informationen über Produkte/-versionen und deren verwendete Bibliotheken
    - Unter anderem auch De-Support (d.h. ggfs. Deaktivierung/Entfernung aus Dependency Track)
    - Hilfestellung bei manueller Erstellung
    - Metadatenpflege bei neuer Abhängigkeit und bei Korrekturen (z.B. von generierten CPEs)
  - Informationen über Einsatz der Produkte/-versionen auf welchen Systemen/Kunden mit Feature Umfang
    - Identifizierung der betroffenen Systeme zur Entscheidung und Einleitung der Maßnahmen

OWASP Dependency Track

Unsere Integrationsvariante



- Kommunikation mit Dependency Track per REST-API
- SBOM-Verarbeitung per CycloneDX Core Java Library
- Vorgefertigter BOM Repository Server als Versionierung und Verwaltung leider unpassend

robotron®

Quellen und Weiterführendes

- OWASP Dependency Track: https://dependencytrack.org/
  - Dokumentation: https://docs.dependencytrack.org/

- CycloneDX: https://cyclonedx.org/
  - CycloneDX Tool Center: https://cyclonedx.org/tool-center/

Vielen Dank für die Aufmerksamkeit!

Steffen Seidler
Senior-Softwareentwickler

Robotron Datenbank-Software GmbH
www.robotron.de

robotron®

@Robotron Datenbank-Software GmbH

Robotron Datenbank-Software GmbH
@Robotron_DD

robotron.de/newsletter

@Robotron Datenbank-Software GmbH

robotron®