Risiken in der Software Supply Chain identifizieren, mit OWASP Tools

Was haben wir aus Log4j gelernt?

@kitenco1

Stephan Kaps

Leiter Softwareentwicklung Bundesamt für Soziale Sicherung

Gründer Java User Group Bonn

Java Entwickler & Architekt seit 2002

Weitere Schwerpunkte:

- Softwareentwicklungsprozesse
- BizDevSecOps
- OpenSource
- Speaker & Autor

bizdevops.org

BONN JAVA USER GROUP

Wer sind wir uns was machen wir?

(Aufsichts-) Behörde mit ca. 700 Mitarbeiter

ca. 60 Fachanwendungen (Produkte)

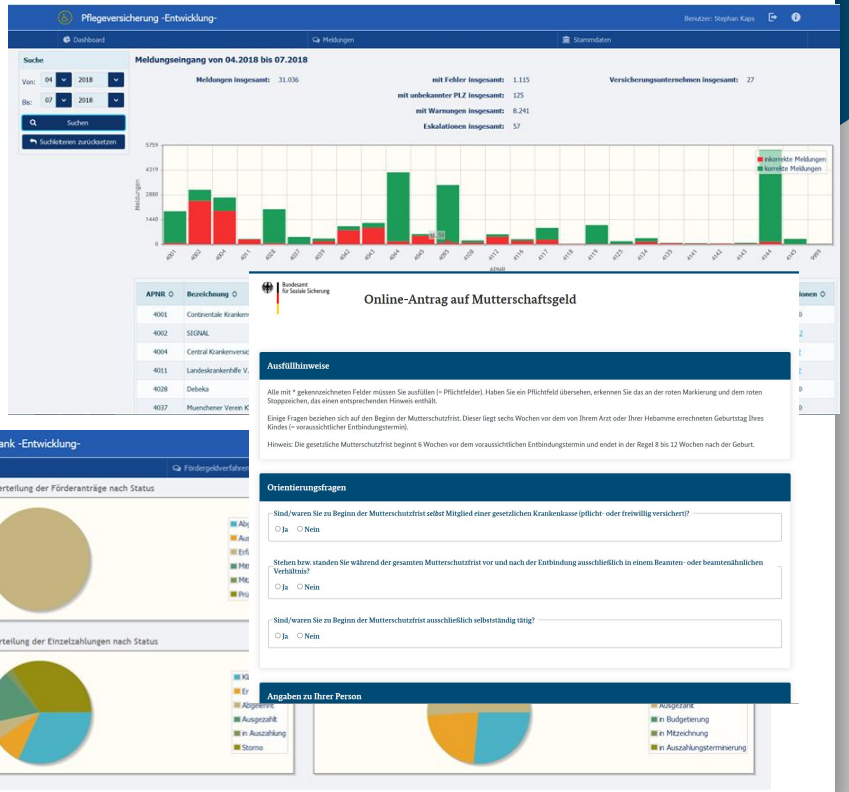
für interne Fachbereiche & Landesprüfdienste

seit 2016 agile Produktentwicklung (Scrum)

10 Entwickler und 2 Product Owner

Private-Cloud mit Containertechnologien

Sehr hohe Mitarbeiterzufriedenheit



SPIEGEL Netzwelt

Log4j-Sicherheitslücke

Wie löscht man ein brennendes Internet?

IT-Fachleute sind wegen der Log4j-Sicherheitslücke alarmiert, erste Angriffe laufen bereits. Welche Folgen hat das für Nutzer? Wie ließe sich so etwas verhindern? Antworten auf die wichtigsten Fragen.

Von **Patrick Beuth**

13.12.2021, 17.20 Uhr

Was ist passiert?

CVE-2021-44228: Risiko kritisch, CVSSv3 10/10

Cyber-Sicherheitswarnung des BSI Stufe Rot

Pressekonferenz in der Tagesschau am 13.12.2021

https://www.tagesschau.de/multimedia/video/video-960365.html

Maßnahmen

Identifizieren & Patchen

Härtung -> Flag setzen (log4j2.formatMsgNoLookups=true)

In Web Application Firewall (WAF) erkennen und blockieren

Ausschalten?

Patchen!

Patchen!!1!!

Maßnahmen

Identifizieren

Day 0 = Bekanntwerden der Lücke

Day 1 = Fixes und Patches ausrollen

Day 2 = Was haben wir gelernt?

2 Herausforderungen

Wie können wir sicherstellen, dass wir Fragen zu dem, was wir in unserer Software verwenden, schnell beantworten können, wenn es eine weitere Schwachstelle (dieser Größenordnung) gibt?

Wie können wir sicherstellen, dass wir nicht anfällige Versionen von log4j aufnehmen und bereitstellen?

Welche Produkte sind eigentlich betroffen?

Kaufsoftware

= Black Box

Ist die Anwendung überhaupt in Java geschrieben?

Steht vielleicht was auf der Webseite?

Können wir den Hersteller kontaktieren?

Echt jetzt?

Eigenentwickelte Software

= Analyse der Third-Party-Dependencies

Welche Anwendung verwendet überhaupt Log4j?

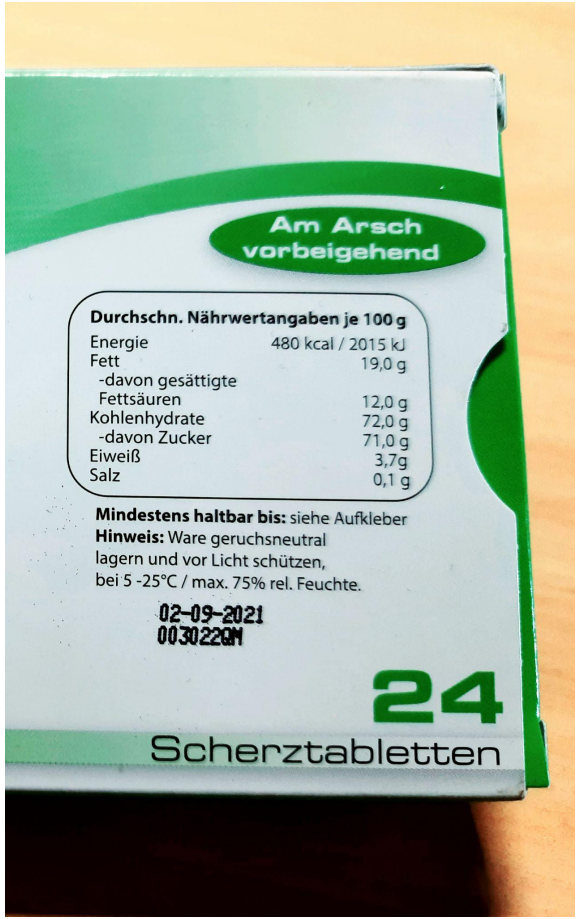
Und falls ja, in welcher Version?

Haben wir einen Überblick über die gesamte Anwendungslandschaft?

@kitenco1

Software Bill of Materials (SBOM)

- Beipackzettel zu einer Software
- Auflistung aller verwendeter Bibliotheken bzw. Komponenten
- Maschinenlesbare Inventarliste
- Beinhaltet direkte und transitive Abhängigkeiten und deren hierarchische Beziehung
- Für Software, die in Amerikanischer Regierung verwendet wird, inzwischen quasi verpflichtend https://www.cisa.gov/sbom
- Spezifikation Standard SPDX oder CycloneDX https://cyclonedx.org/specification/overview/ https://cyclonedx.org/tool-center/



Was ist CycloneDX?

“OWASP CycloneDX ist ein leichtgewichtiger Software Bill of Materials (SBOM)-Standard, der für den Einsatz in Anwendungssicherheitskontexten und der Analyse von Lieferkettenkomponenten (SCA) entwickelt wurde.”

SCA = supply chain component analysis

Technology

Technology Radar

Search Techniques

Techniques

Adopt

1. Four key metrics
2. Single team remote wall

Trial

3. Data mesh
4. Definition of production readiness
5. Documentation quadrants
6. Rethinking remote standups
7. Server-driven UI
8. Software Bill of Materials

Themes for this volume

Software Supply Chain Innovations

We've seen very public and severe problems that are caused by poor software supply chain governance. Now teams are realizing that responsible engineering practices include validating and governing project dependencies.

Why Do Developers Implementing State Management in Rea

Ever since Redux was releas
seen a steady stream of too
frameworks that manage sta
different ways, each with a
of trade-offs. We don't know
can only speculate: Is this th
churn the JavaScript ecosys
to promote?

Download

Adopt

No change

o see?

lecting what we came might have covered ar already. We re too many to talk e the Radar reflects our experience, it is not based on a comprehensive market analysis.

what you are looking for on a previous Radar already. We sometimes cull things just because there are too many to talk

the older SPDX standard with a

Tool Center

All tools 149 | Open Source 112 | Proprietary 39 | Build Integration 61 | Analysis 45 | Author 2 | GitHub Action 13 | Transform 8 | Library 10 | Signing / Notary 5 | Distribute 4

proprietary analysis build-integration

Apiiro

Apiiro

Apiiro enables security & development teams to proactively remediate critical risks in their cloud-native applications such as design flaws, secrets, IaC misconfigurations, API & OSS vulnerabilities across the software supply chain.

proprietary analysis

AppSonar

CyberTest

AppSonar Static Code Analyzer helps improve the security and quality of application code and can generate CycloneDX BOMs during analysis.

opensource build-integration

Auditjs

Sonatype

Audits an NPM package.json file to identify known vulnerabilities

Forks 51 Stars 184

opensource distribute

BOM Repository Server

CycloneDX

A lightweight repository server used to publish, manage, and distribute CycloneDX SBOMs

Forks 8 Stars 38

proprietary analysis

Black Duck

Synopsys

Black Duck software composition analysis (SCA) helps teams manage the security, quality, and license compliance risks that come from the use of open source and third-party code in applications and containers.

proprietary analysis

BlackBerry Jarvis

BlackBerry

Software composition analysis (SCA) and security testing solution that detects and list open-source software and software licenses within embedded systems and associated cybersecurity vulnerabilities and exposures

proprietary analysis

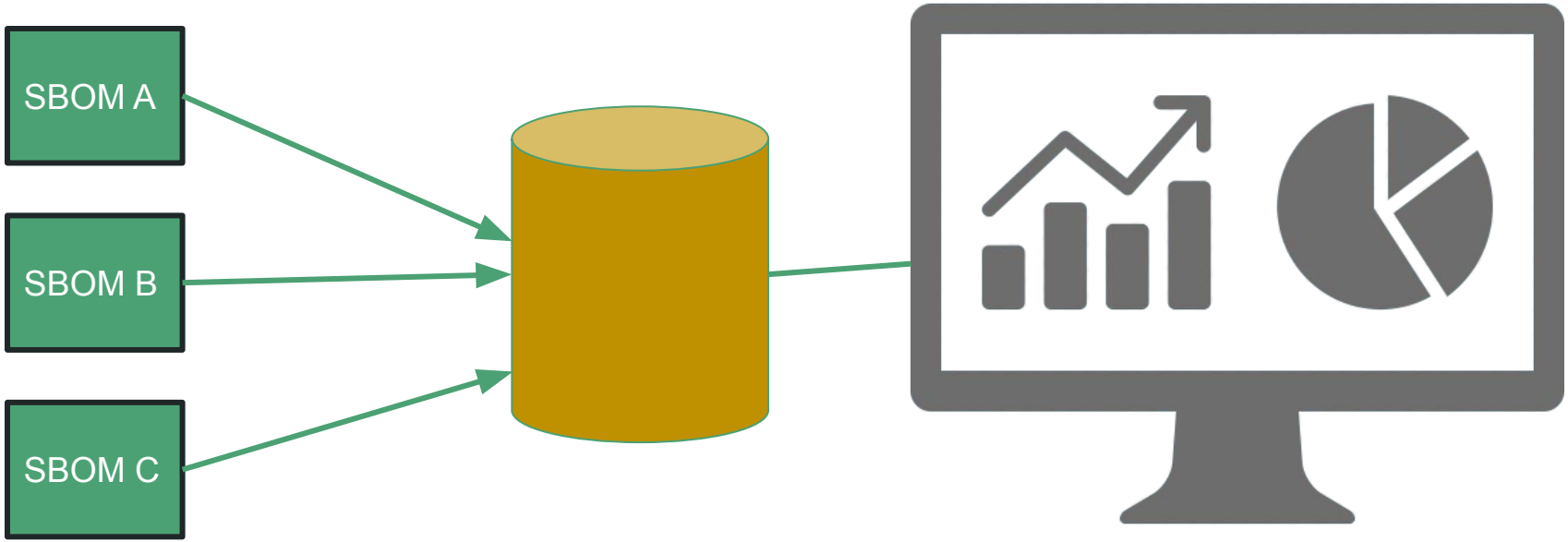
proprietary build-integration analysis github-ac...

opensource analysis

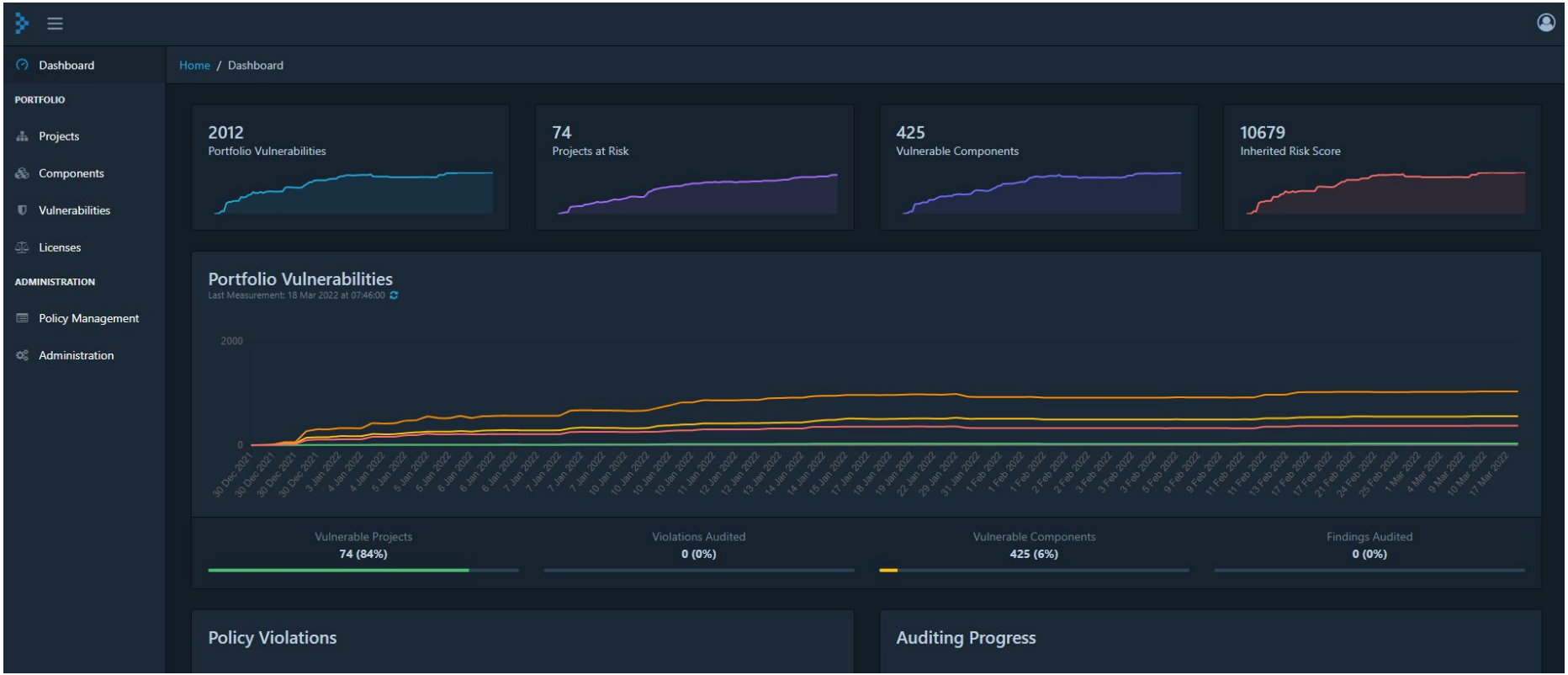
```xml
<component type="library" bom-ref="pkg:maven/org.apache.poi/poi@3.17?type=jar">
  <publisher>Apache Software Foundation</publisher>
  <group>org.apache.poi</group>
  <name>poi</name>
  <version>3.17</version>
  <description>Apache POI - Java API To Access Microsoft Format Files</description>
  <scope>optional</scope>
  <hashes>
    <hash alg="MD5">243bc3d431e4fadb79738719504c64f7</hash>
    <hash alg="SHA-1">0ae92292a2043888b40d418da97dc0b669fde326</hash>
    <hash alg="SHA-256">30181821dd2e849727b638b9e329aeff4a64f3445c4142b13cf7a18bb3552edd</hash>
    <hash alg="SHA-384">f9a4db2b945cdba835b3c076c3f2bfa72767a5c02f81a57ffb4a5782651d5da66453c03e5c1be0505d1edad8a35c6f82</hash>
    <hash alg="SHA-512">a0c018d8999356000b1077b343aac1a2e9050f84e3097e28b77869be9fc23475d4ac59bf375a2c96b6d824f06ba2d1873ee8d4495fb9bb412f848379ac7d11fb</hash>
    <hash alg="SHA3-256">84ba8a8001b44c04048d5ab1208640f0f5d54183231962cb180de7d7cfa7a386</hash>
    <hash alg="SHA3-384">2101589e294edd8d55f29be193e74c84afb8db754566b08b369c6c235e969af25ee8962e70f7c3dc262f7cbdebe3d57d</hash>
    <hash alg="SHA3-512">cc8b929d7f54aa98c606977bab56fcfe198e0bb21403b7b2a1bff6bf2940901aee9d47bde493a98af376b4b1e75d110284023bd174b7774d454fac6e14b0606c</hash>
  </hashes>
  <licenses>
    <license>
      <id>Apache-2.0</id>
    </license>
  </licenses>
  <purl>pkg:maven/org.apache.poi/poi@3.17?type=jar</purl>
  <externalReferences>
    <reference type="website"><url>http://www.apache.org/</url></reference>
    <reference type="mailing-list"><url>http://mail-archives.apache.org/mod_mbox/poi-user/</url></reference>
  </externalReferences>
</component>
```


@kitencol

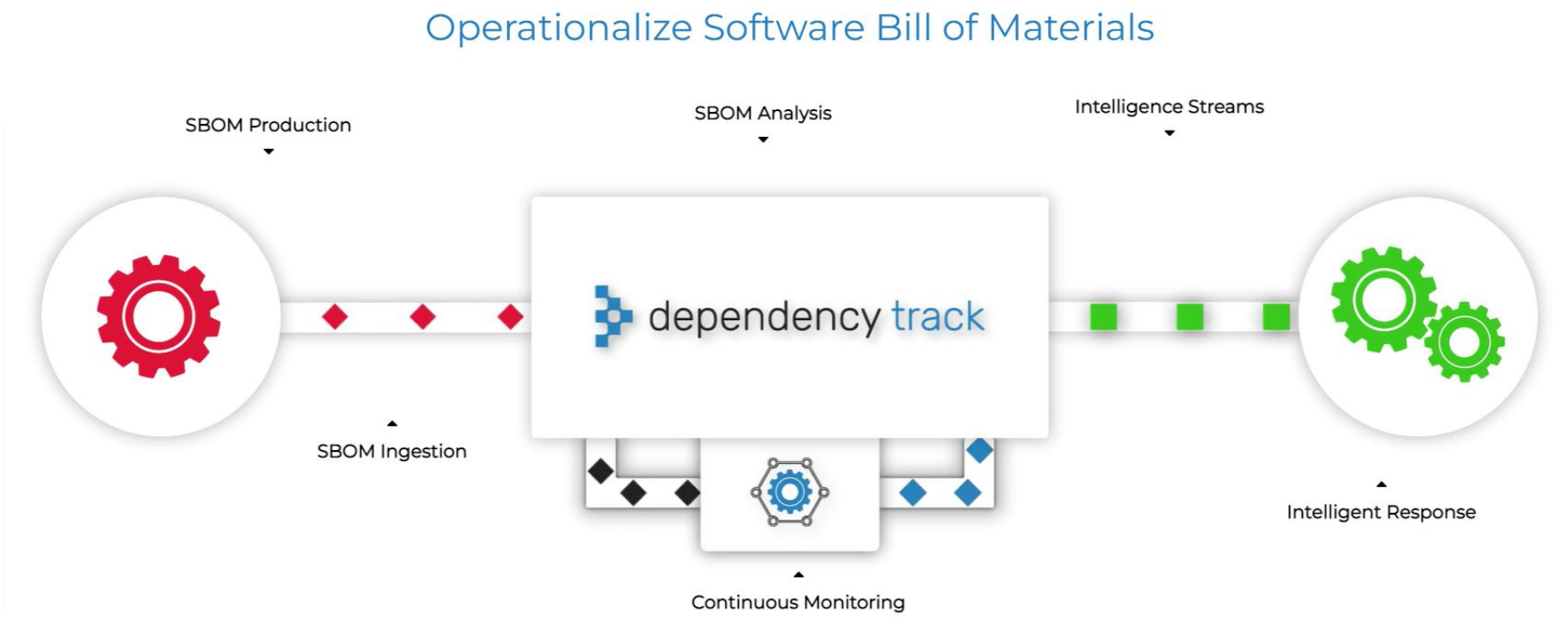
Zusammenführen

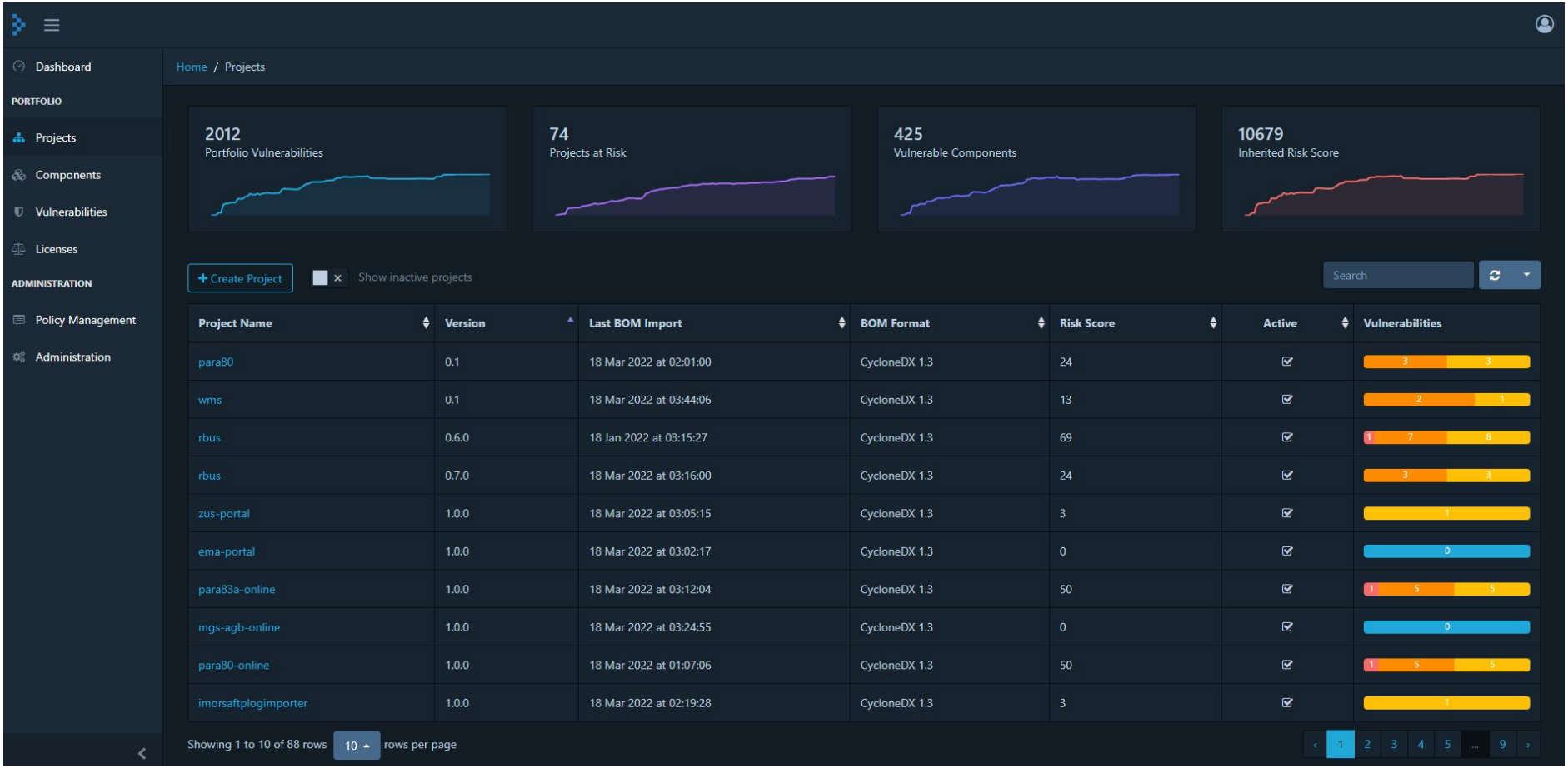


OWASP Dependency Track

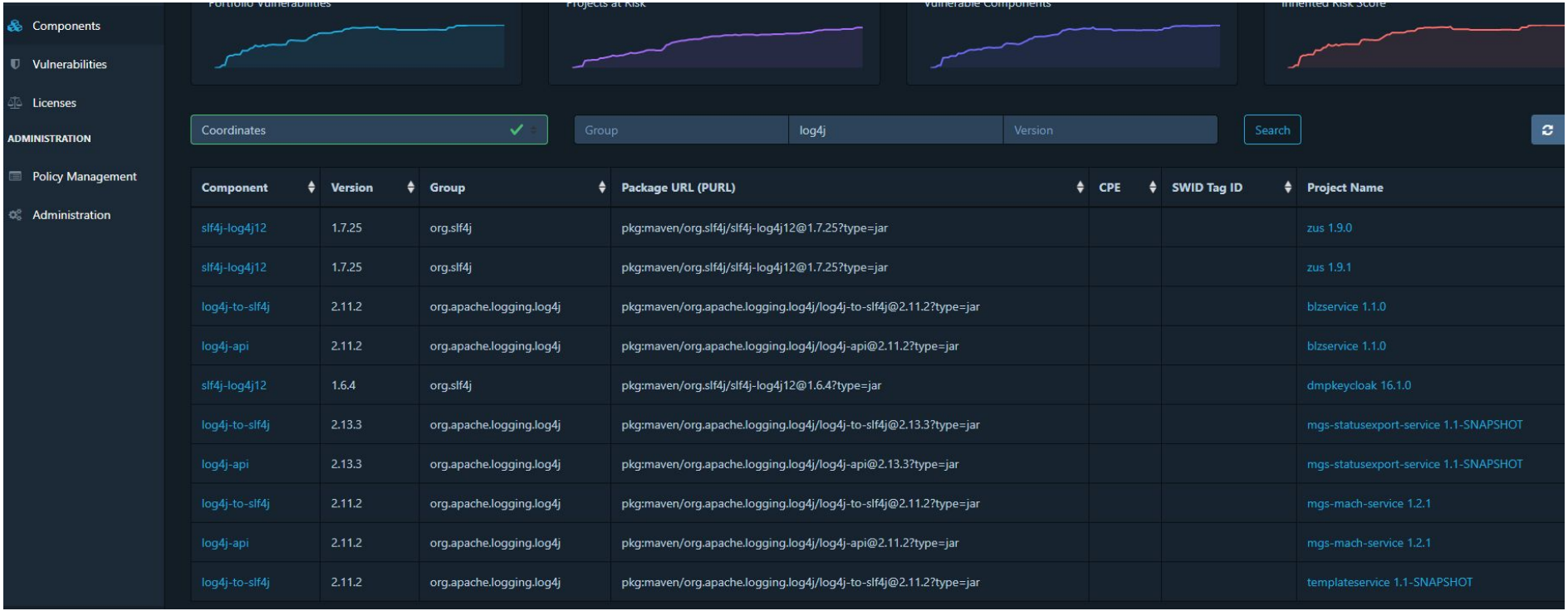


Continuous Security Monitoring



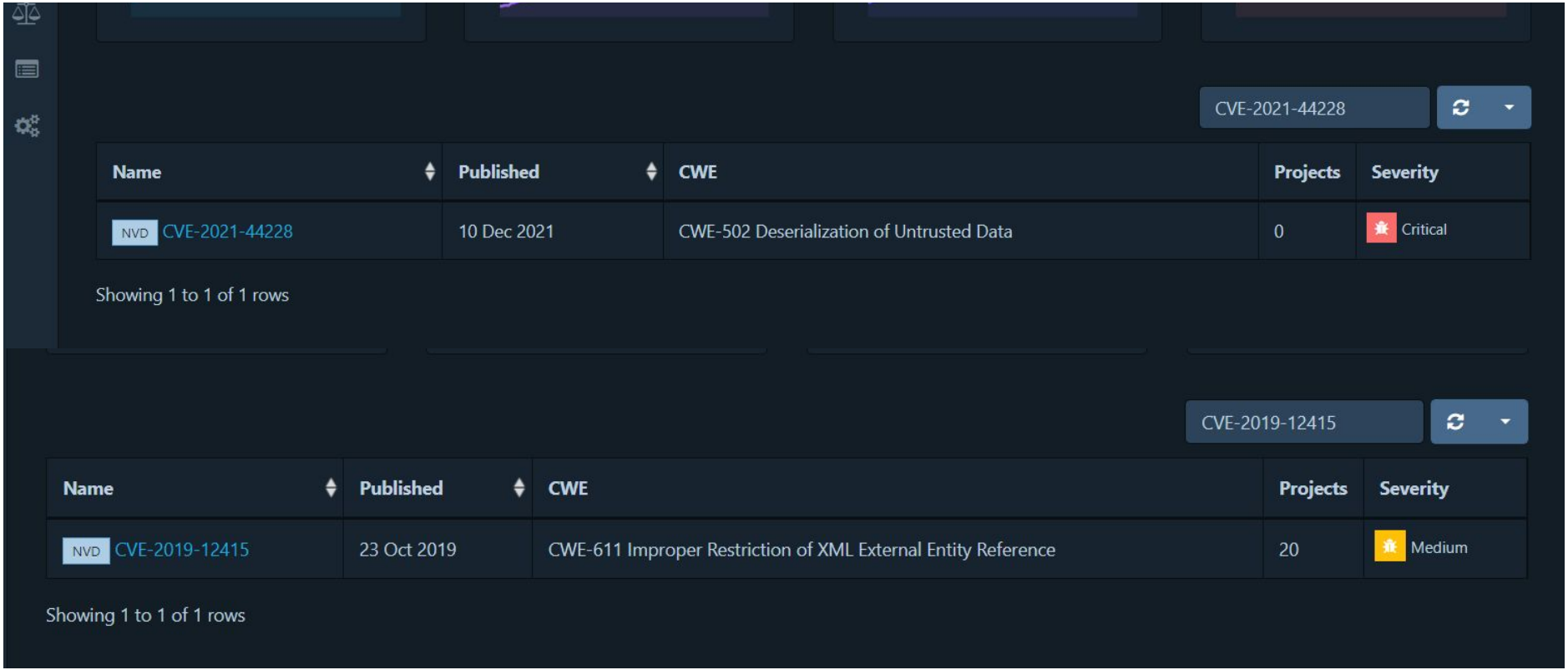

Dashboard
PORTFOLIO
Projects
Components
Vulnerabilities
Licenses
ADMINISTRATION
Policy Management
Administration
Home / Projects
2012
Portfolio Vulnerabilities
74
Projects at Risk
425
Vulnerable Components
10679
Inherited Risk Score
+ Create Project
Show inactive projects
Search
Project Name
Version
Last BOM Import
BOM Format
Risk Score
Active
Vulnerabilities
para80 0.1 18 Mar 2022 at 02:01:00 CycloneDX 1.3 24
wms 0.1 18 Mar 2022 at 03:44:06 CycloneDX 1.3 13
rbus 0.6.0 18 Jan 2022 at 03:15:27 CycloneDX 1.3 69
rbus 0.7.0 18 Mar 2022 at 03:16:00 CycloneDX 1.3 24
zus-portal 1.0.0 18 Mar 2022 at 03:05:15 CycloneDX 1.3 3
ema-portal 1.0.0 18 Mar 2022 at 03:02:17 CycloneDX 1.3 0
para83a-online 1.0.0 18 Mar 2022 at 03:12:04 CycloneDX 1.3 50
mgs-agb-online 1.0.0 18 Mar 2022 at 03:24:55 CycloneDX 1.3 0
para80-online 1.0.0 18 Mar 2022 at 01:07:06 CycloneDX 1.3 50
imorsaftplogimporter 1.0.0 18 Mar 2022 at 02:19:28 CycloneDX 1.3 3
Showing 1 to 10 of 88 rows
10
rows per page
1 2 3 4 5 ... 9

Suche nach Komponenten



Component	Version	Group	Package URL (PURL)	CPE	SWID Tag ID	Project Name
slf4j-log4j12	1.7.25	org.slf4j	pkg:maven/org.slf4j/slf4j-log4j12@1.7.25?type=jar			zus 1.9.0
slf4j-log4j12	1.7.25	org.slf4j	pkg:maven/org.slf4j/slf4j-log4j12@1.7.25?type=jar			zus 1.9.1
log4j-to-slf4j	2.11.2	org.apache.logging.log4j	pkg:maven/org.apache.logging.log4j/log4j-to-slf4j@2.11.2?type=jar			blzservice 1.1.0
log4j-api	2.11.2	org.apache.logging.log4j	pkg:maven/org.apache.logging.log4j/log4j-api@2.11.2?type=jar			blzservice 1.1.0
slf4j-log4j12	1.6.4	org.slf4j	pkg:maven/org.slf4j/slf4j-log4j12@1.6.4?type=jar			dmpkeycloak 16.1.0
log4j-to-slf4j	2.13.3	org.apache.logging.log4j	pkg:maven/org.apache.logging.log4j/log4j-to-slf4j@2.13.3?type=jar			mgs-statusexport-service 1.1-SNAPSHOT
log4j-api	2.13.3	org.apache.logging.log4j	pkg:maven/org.apache.logging.log4j/log4j-api@2.13.3?type=jar			mgs-statusexport-service 1.1-SNAPSHOT
log4j-to-slf4j	2.11.2	org.apache.logging.log4j	pkg:maven/org.apache.logging.log4j/log4j-to-slf4j@2.11.2?type=jar			mgs-mach-service 1.2.1
log4j-api	2.11.2	org.apache.logging.log4j	pkg:maven/org.apache.logging.log4j/log4j-api@2.11.2?type=jar			mgs-mach-service 1.2.1
log4j-to-slf4j	2.11.2	org.apache.logging.log4j	pkg:maven/org.apache.logging.log4j/log4j-to-slf4j@2.11.2?type=jar			templateservice 1.1-SNAPSHOT

Suche nach Vulnerabilities

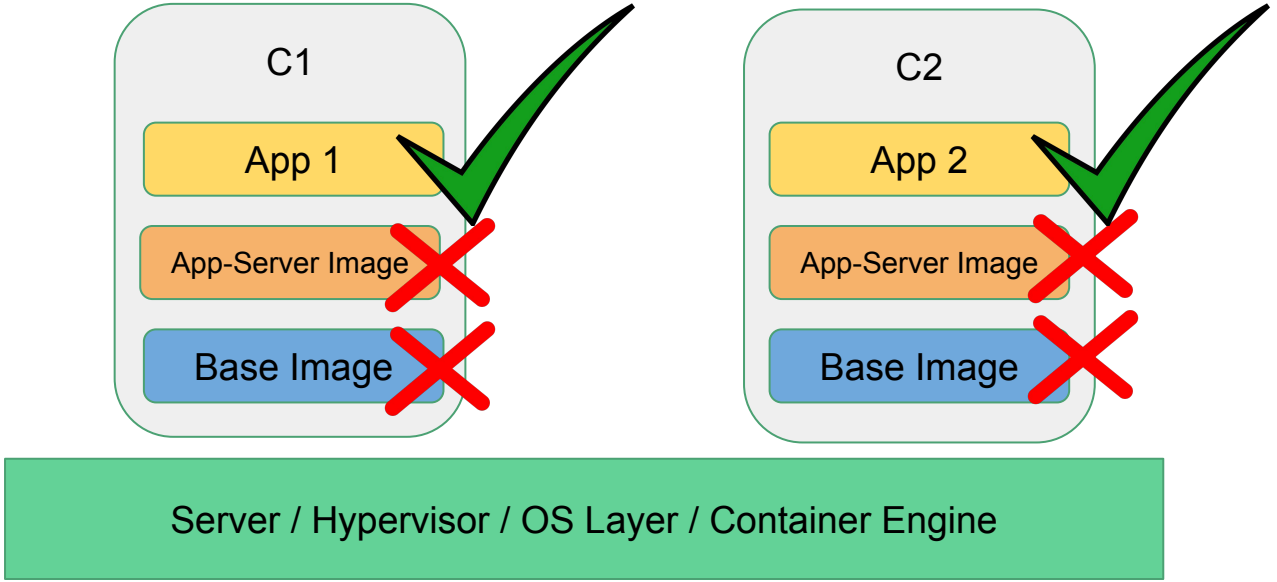


Integration in Build-Prozess (Nightly)

```
stage('Stat. Analysen') {
    steps {
        withMaven(maven: 'Maven 3.8.6') {
            sh "mvn -U spotbugs:spotbugs org.cyclonedx:cyclonedx-maven-plugin:makeAggregateBom"
        }
        withCredentials([string(credentialsId: 'deptrack-apikey-text', variable: 'API_KEY')]) {
            dependencyTrackPublisher artifact: 'target/bom.xml', projectName: "${env.PROJECT}",
            projectVersion: "${env.OWNVERSION}", synchronous: true, dependencyTrackApiKey: API_KEY,
            failedNewCritical: 3, failedNewHigh: 5, failedNewLow: 15, failedNewMedium: 10,
            failedTotalCritical: 5, failedTotalHigh: 10, failedTotalLow: 50, failedTotalMedium: 30,
            unstableNewCritical: 1, unstableNewHigh: 2, unstableNewLow: 10, unstableNewMedium: 5,
            unstableTotalCritical: 1, unstableTotalHigh: 5, unstableTotalLow: 30, unstableTotalMedium: 15
        }
    }
}
```

Was ist mit der Ausführungsschicht?

Container-Technologien



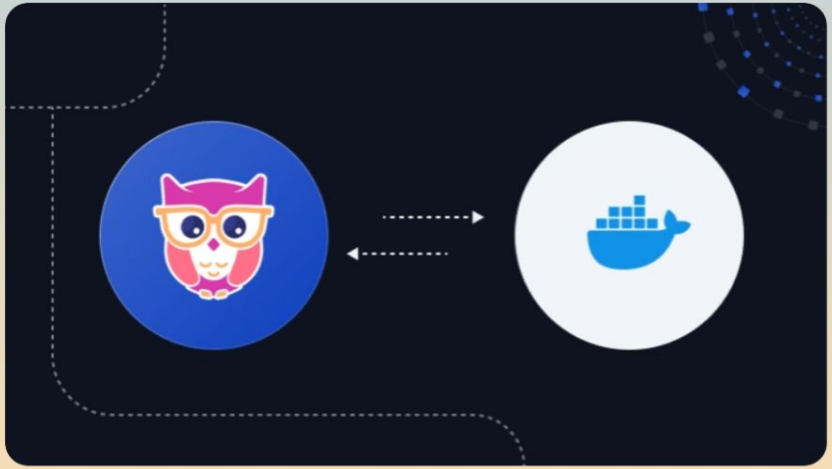
SBOMs für Container Images und Filesystem

https://github.com/anchore/syft

https://github.com/anchore/grype

Anchore and Docker Release 'docker sbom' to Create Comprehensive SBOMs Based on Syft

By: Dan Nurmi APR 07, 2022 4 MIN READ



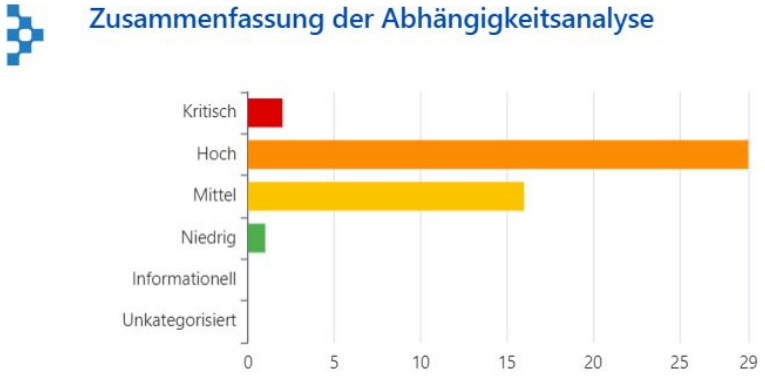
Today Anchore and Docker released the first feature in what we anticipate will be an ongoing initiative to bring the value of the software bill of materials (SBOM) to all container-oriented build and publication systems. Now included in the latest Docker Desktop version is an operation called 'docker sbom' that is available via the 'docker' command. This new operation, which is built on top of Anchore's open source Syft project, enables Docker users to quickly generate detailed SBOM documents against container images using the native Docker CLI.

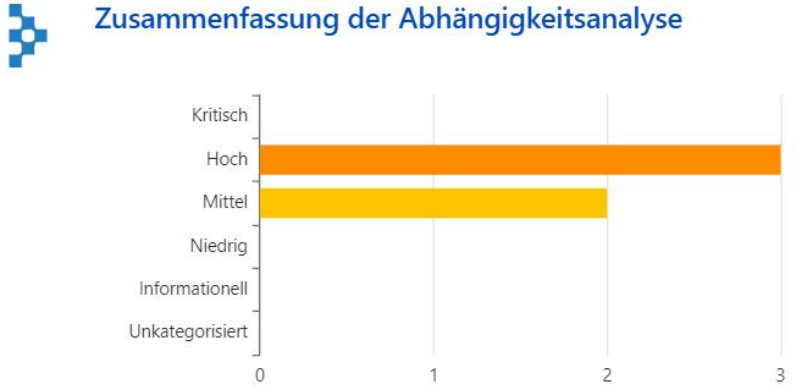
CLI Aufruf und direkte Auswertung

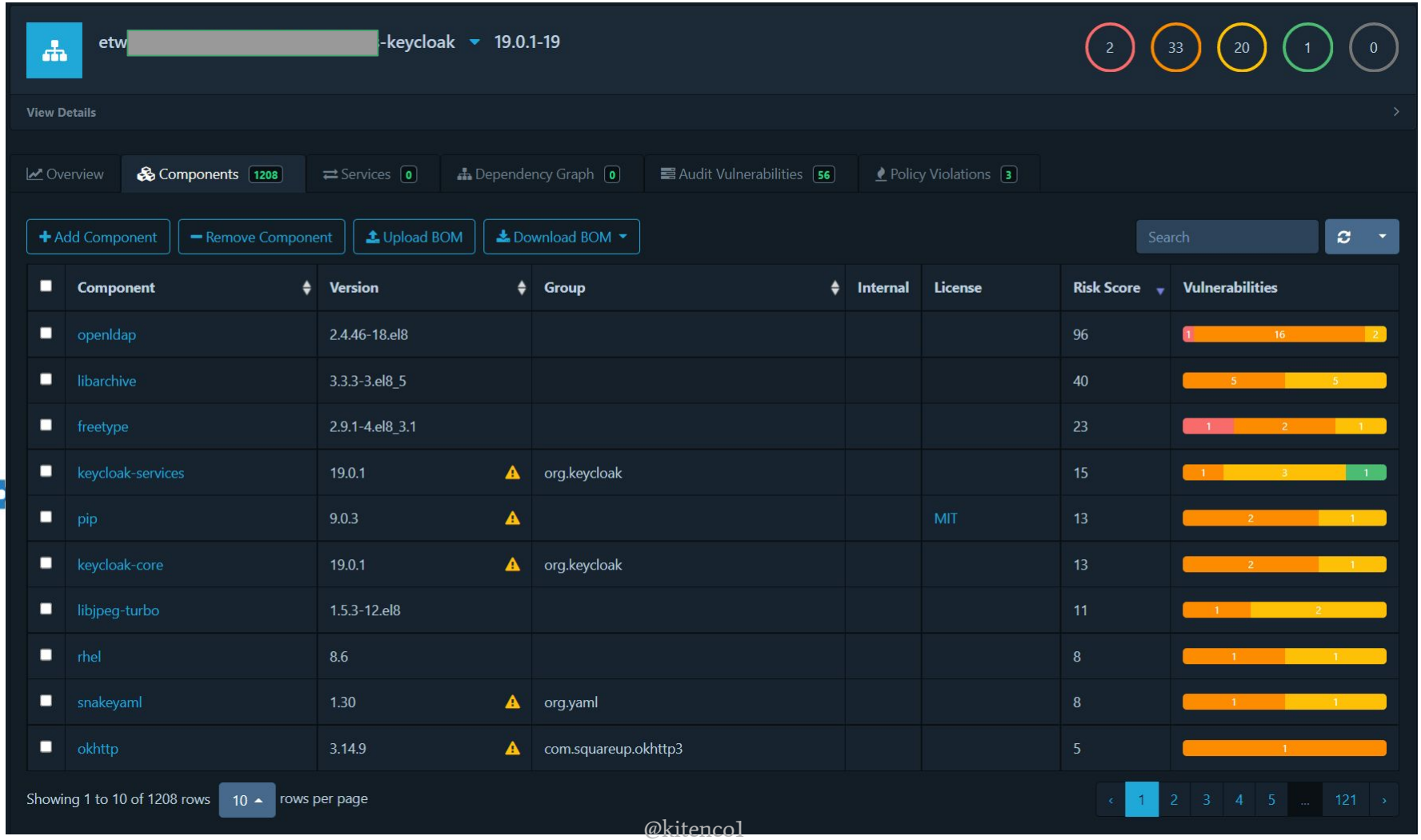
```
stage('Analyze Dependencies of Docker Image'){
    steps{
        script {
            dir ('out') {
                sh "syft -o cyclonedx-xml=sbom.xml ${env.IMAGE}:${env.TAG}"
                withCredentials([string(credentialsId: 'deptrack-apikey-text', variable: 'API_KEY')]) {
                dependencyTrackPublisher artifact: 'sbom.xml', projectName: "${env.IMAGE}",
                projectVersion: "${env.TAG}", synchronous: true, dependencyTrackApiKey: API_KEY,
                failedNewCritical: 3, failedNewHigh: 5, failedNewLow: 15, failedNewMedium: 10,
                failedTotalCritical: 5, failedTotalHigh: 10, failedTotalLow: 50, failedTotalMedium: 30,
                unstableNewCritical: 1, unstableNewHigh: 2, unstableNewLow: 10, unstableNewMedium: 5,
                unstableTotalCritical: 1, unstableTotalHigh: 5, unstableTotalLow: 30, unstableTotalMedium: 15
                }
            }
        }
    }
```

Syft findet ein bisschen mehr …

Project Name	Version	Classifier	Last BOM Import	BOM Format	Risk Score	Active	Policy Violations	Vulnerabilities
mgs-keycloak	19.0.1	Library	14 Sep 2022 at 01:33:29	CycloneDX 1.4	21	☑	11	3 2
etw-docker-03.bvaetw.de/mgs/mgs-keycloak	19.0.1-19	Container	18 Aug 2022 at 17:45:40	CycloneDX 1.4	246	☑	3	2 33 20



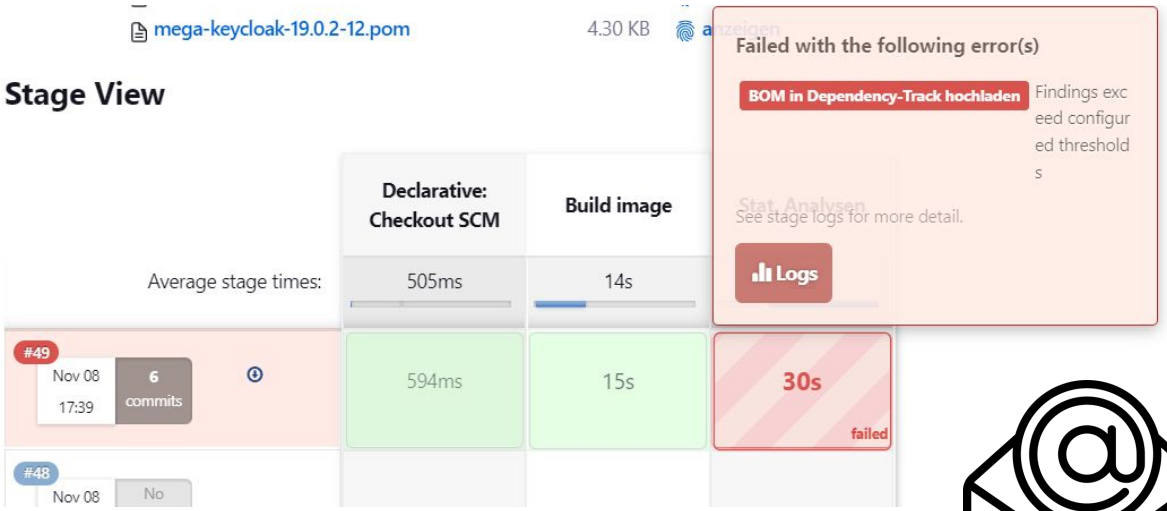



etw -keycloak 19.0.1-19
2
33
20
1
0
View Details
Overview
Components 1208
Services 0
Dependency Graph 0
Audit Vulnerabilities 56
Policy Violations 3
+ Add Component
− Remove Component
Upload BOM
Download BOM
Search
Component
Version
Group
Internal
License
Risk Score
Vulnerabilities
openldap 2.4.46-18.el8 96
libarchive 3.3.3-3.el8_5 40
freetype 2.9.1-4.el8_3.1 23
keycloak-services 19.0.1 org.keycloak 15
pip 9.0.3 MIT 13
keycloak-core 19.0.1 org.keycloak 13
libjpeg-turbo 1.5.3-12.el8 11
rhel 8.6 8
snakeyaml 1.30 org.yaml 8
okhttp 3.14.9 com.squareup.okhttp3 5
Showing 1 to 10 of 1208 rows
10
rows per page
1
2
3
4
5
121

@kitenco1

Was jetzt?

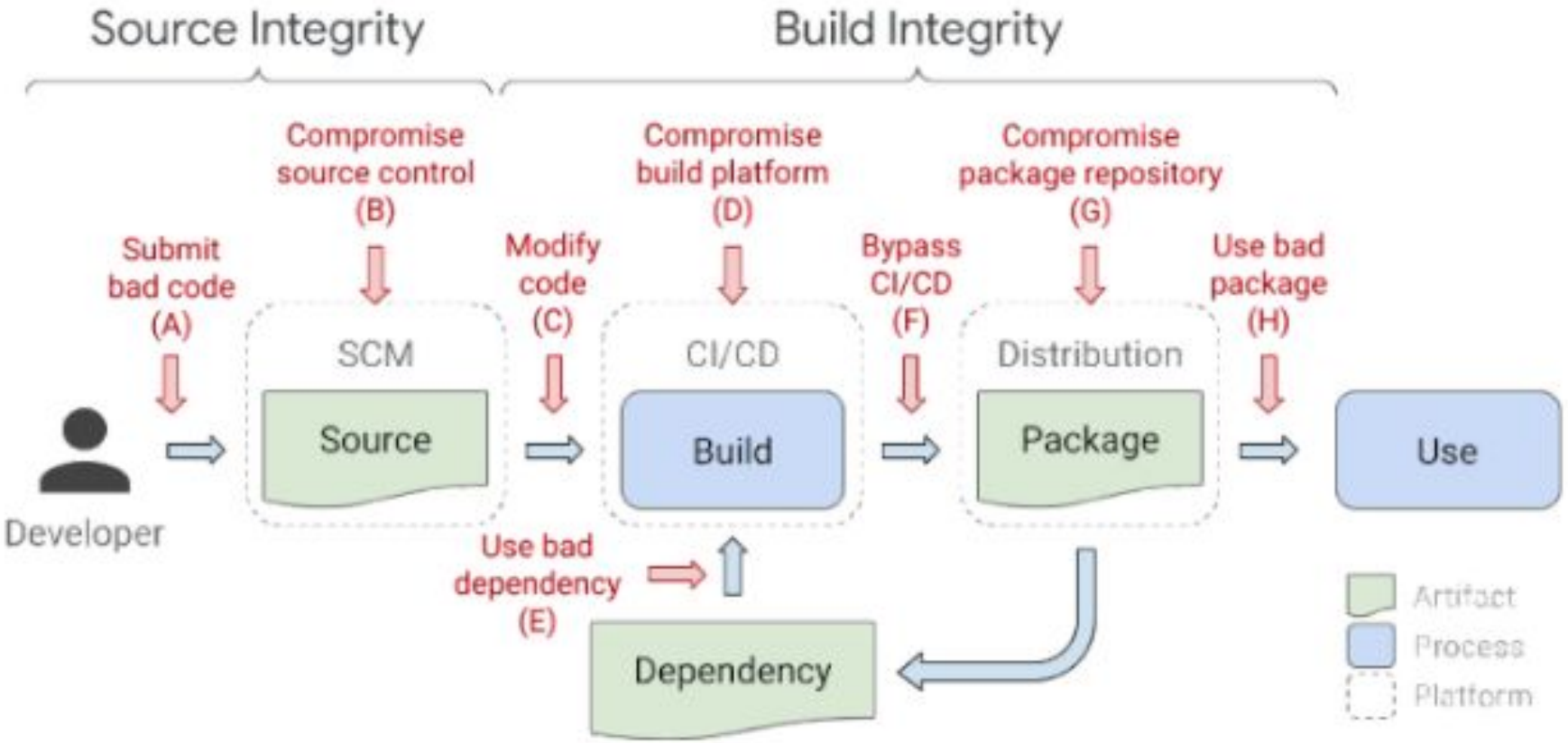
Organisatorisch



CVE-2022-12345

DEVOPS-1016 Der Nightly-Build von 'mega-keycloak' ist fehlgeschlagen IN ARBEIT *Nicht zugewiesen*

Software Supply Chain ist mehr!



https://opensource.googleblog.com/2021/10/protect-your-open-source-project-from-supply-chain-attacks.html

Trusted Commits

Multi-Faktor Authentifizierung

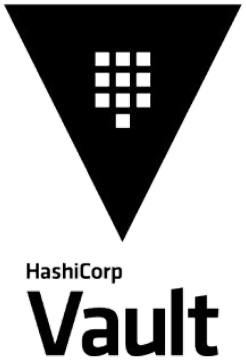
IP Allowlist

Github Branch Protection Rules

Secrets

Keine Secrets im Repo (z.B. Thoughtworks Talisman)

Keine Secrets im Build Tool (z.B. Hashicorp Vault)



Automatische Updates für Dependencies

- Dependabot, Renovate und Ähnliches

https://github.com/dependabot

https://github.com/renovatebot/renovate

Fähigkeit, schnell auszuliefern

- Continuous Delivery & Deployment

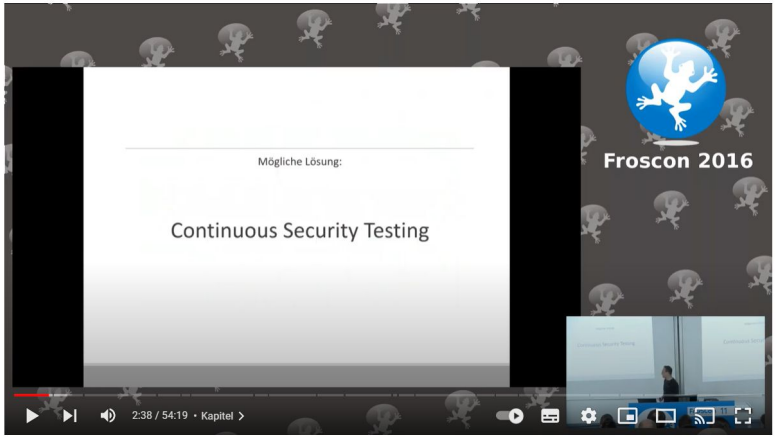
https://entwickler.de/devops/roadmap-einer-spannenden-reise
https://devm.io/continuous-delivery/prerequisites-continuous-deployment-enterprises-001

Secure Coding & Testing

- Sichere Softwareentwicklung & Security Testing (SAST, DAST)


Sichere Softwareentwicklung: Einstieg in Secure-Coding und Continuous-Security-Testing

Sichere Softwareentwicklung:
Einstieg in Secure-Coding und Continuous-Security-Testing



FrOSCon 2016
Stephan Kaps: Sichere Softwareentwicklung

Container Security

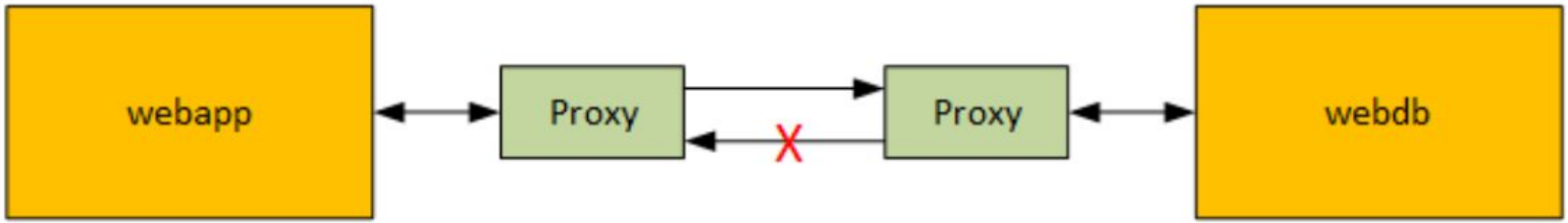
- Angriffsfläche verkleinern
- Privilegien minimieren
- Docker Engine härten

DevOps

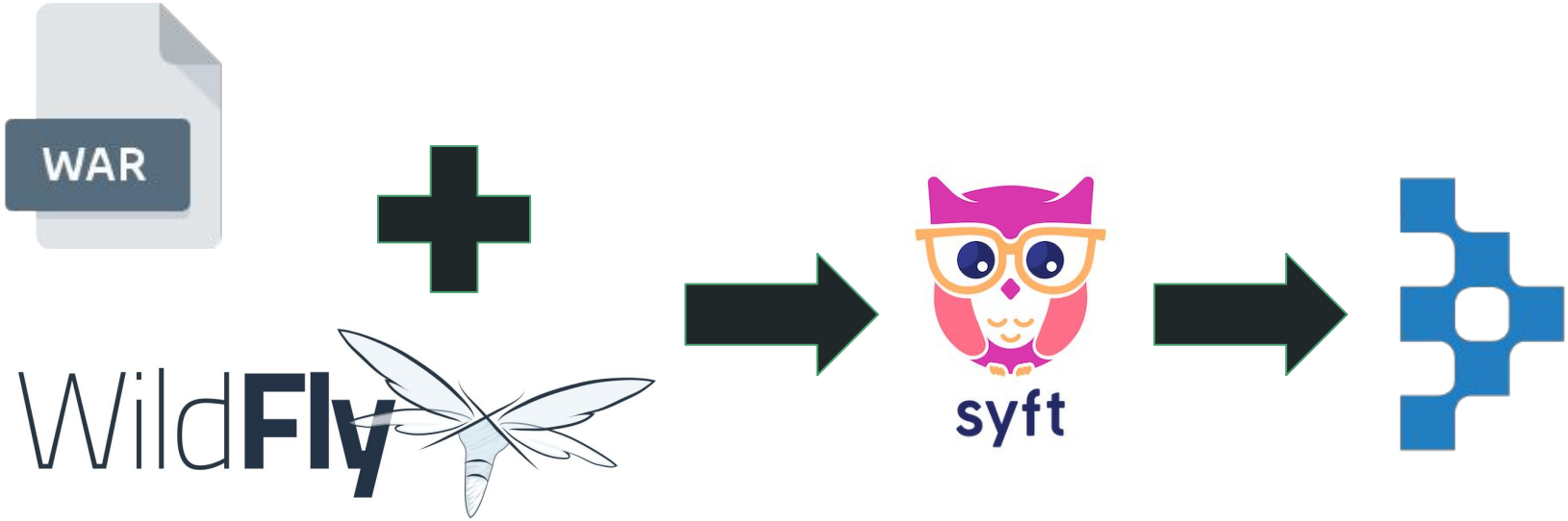
In den sicheren Hafen: Einstieg in Container Security - Stephan Kaps | JCON 2020

Isolierung

- Service-Mesh
- Mikro-Segmentierung
- Kommunikations-Regeln (Policies / Intentions)



Was war jetzt mit Kaufsoftware?



Aufnahme von Anforderung in Ausschreibungen

“Bereitstellung einer SBOM bei jeder ausgelieferten Version”

“Zeitnahe Bereitstellung von Patches bei neuen Sicherheitslücken”

Weitere Maßnahmen:

Wie können wir sicherzustellen, dass wir nicht versehentlich anfällige Versionen von log4j aufnehmen und bereitstellen?

... oder dass jemand uns etwas schadhaftes unterschiebt?

Aber Gott sei Dank benutzen wir Maven und Java

6.1.11. Downloading and Verifying Dependencies

The following command line options affect the way that Maven will interact with remote repositories and how it verifies downloaded artifacts:

-C, --strict-checksums
Fail the build if checksums don't match

-c, --lax-checksums
Warn if checksums don't match

-U, --update-snapshots
Forces a check for updated releases and snapshots on remote repositories

https://books.sonatype.com/mvnref-book/reference/running-sect-options.html#running-sect-deps-option

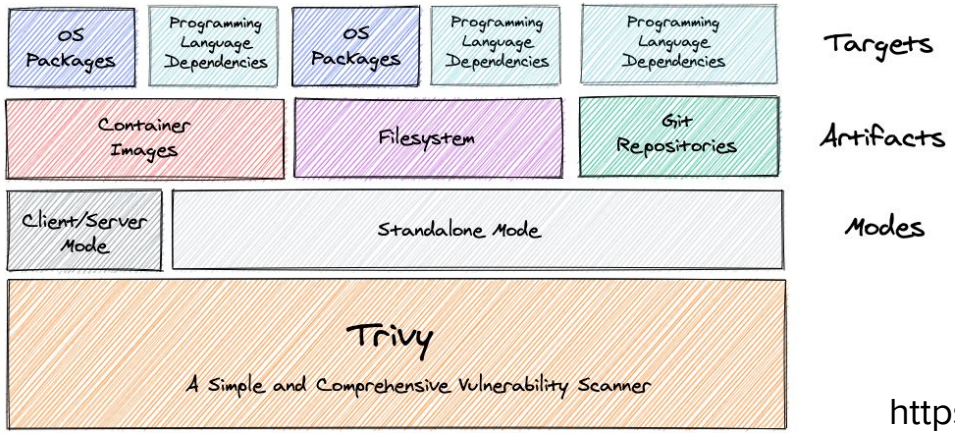
Ab Version 4.0.0 ist das der default!

https://issues.apache.org/jira/browse/MNG-5728

Secure Container Registry

- Vulnerability Scans
- Image Signing
- Pull Policies

https://goharbor.io/



https://github.com/aquasecurity/trivy

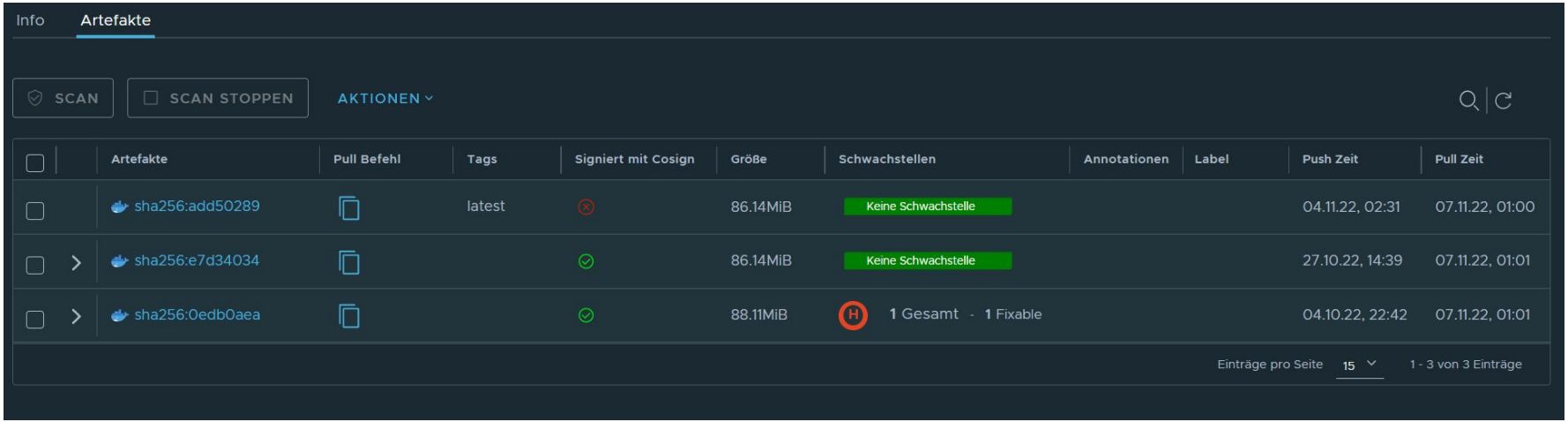
Vulnerability Scanning

>	CVE-2020-16156	Hoch		perl-base	5.28.1-6+deb10u1	
>	CVE-2020-25649	Hoch		com.fasterxml.jackson.core:jackson-databind	2.10.4	2.10.5.1, 2.9.10.7, 2.6.7.4
>	CVE-2021-40690	Hoch		org.apache.santuario:xmlsec	2.1.4	2.1.7, 2.2.3
>	CVE-2020-13949	Hoch		org.apache.thrift:libthrift	0.13.0	0.14.0
>	CVE-2020-13936	Hoch		org.apache.velocity:velocity-engine-core	2.1	2.3
>	CVE-2020-28052	Hoch		org.bouncycastle:bcprov-jdk15on	1.65	1.67
>	CVE-2020-25638	Hoch		org.hibernate:hibernate-core	5.3.17.Final	5.3.20.Final, 5.4.24.Final
>	CVE-2021-37714	Hoch		org.jsoup:jsoup	1.8.3	1.14.2
>	CVE-2014-3530	Hoch		org.picketlink:picketlink-common	2.5.5.SP12-redhat-00009	2.6.1.Final
>	CVE-2019-25013	Mittel	4.8	libc-bin	2.28-10	

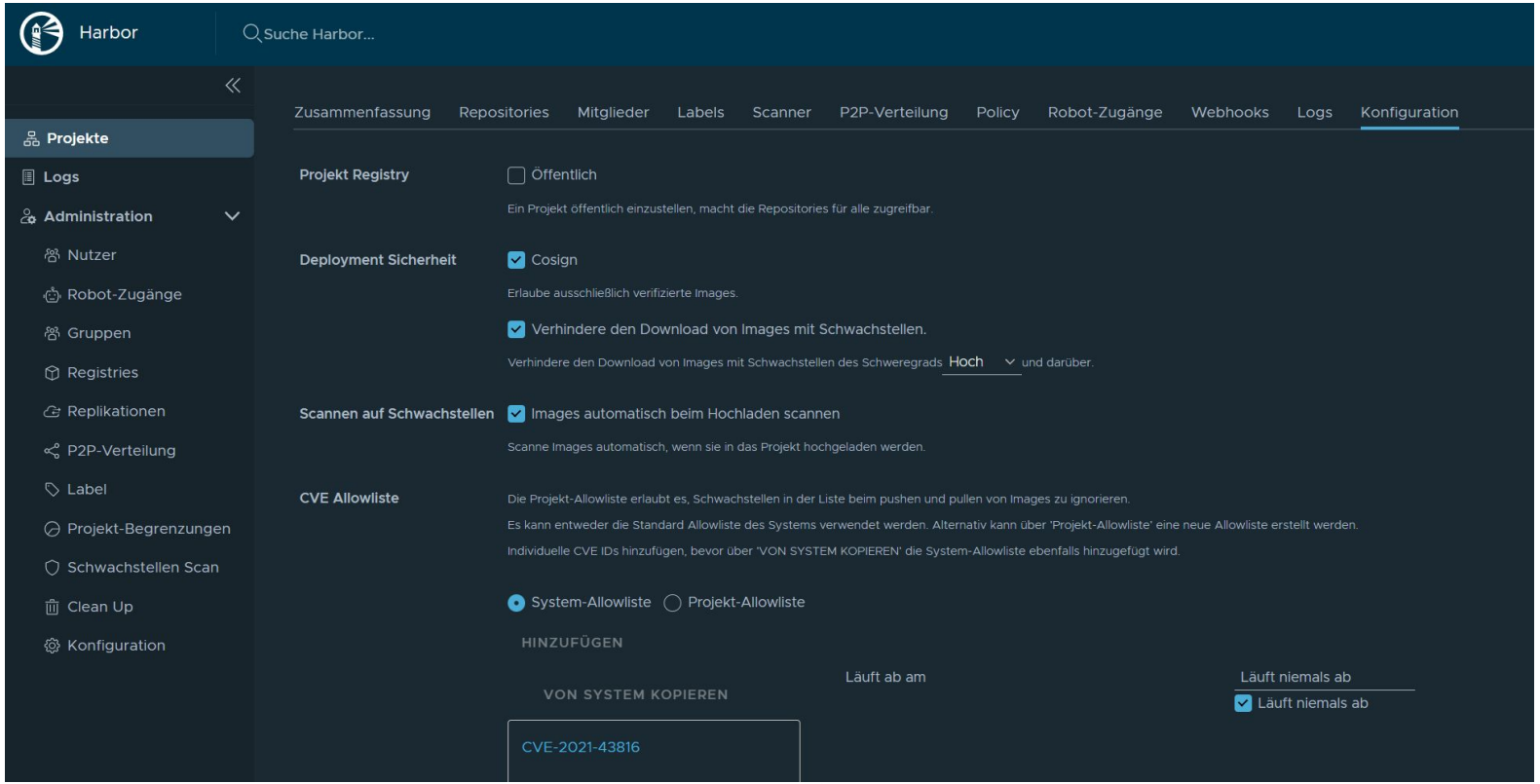
Image Signing mit Sigstore cosign


```
stage('Sign Image'){
    steps{
        script {
            dir ('out') {
                withCredentials([string(credentialsId: 'cosignkey', variable: 'COSIGN_KEY')]) {
                    sh "cosign sign --key $COSIGN_KEY ${env.IMAGE}:${env.TAG}"
                }
            }
        }
    }
}
```

Image Signing mit Sigstore cosign



Pull Policy


Harbor
Suche Harbor...
Projekte
Logs
Administration
Nutzer
Robot-Zugänge
Gruppen
Registries
Replikationen
P2P-Verteilung
Label
Projekt-Begrenzungen
Schwachstellen Scan
Clean Up
Konfiguration
Zusammenfassung
Repositories
Mitglieder
Labels
Scanner
P2P-Verteilung
Policy
Robot-Zugänge
Webhooks
Logs
Konfiguration
Projekt Registry
Öffentlich
Ein Projekt öffentlich einzustellen, macht die Repositories für alle zugreifbar.
Deployment Sicherheit
Cosign
Erlaube ausschließlich verifizierte Images.
Verhindere den Download von Images mit Schwachstellen.
Verhindere den Download von Images mit Schwachstellen des Schweregrads Hoch und darüber.
Scannen auf Schwachstellen
Images automatisch beim Hochladen scannen
Scanne Images automatisch, wenn sie in das Projekt hochgeladen werden.
CVE Allowliste
Die Projekt-Allowliste erlaubt es, Schwachstellen in der Liste beim pushen und pullen von Images zu ignorieren.
Es kann entweder die Standard Allowliste des Systems verwendet werden. Alternativ kann über 'Projekt-Allowliste' eine neue Allowliste erstellt werden.
Individuelle CVE IDs hinzufügen, bevor über 'VON SYSTEM KOPIEREN' die System-Allowliste ebenfalls hinzugefügt wird.
System-Allowliste
Projekt-Allowliste
HINZUFÜGEN
VON SYSTEM KOPIEREN
Läuft ab am
Läuft niemals ab
Läuft niemals ab
CVE-2021-43816

Fazit

Was ist zu tun?

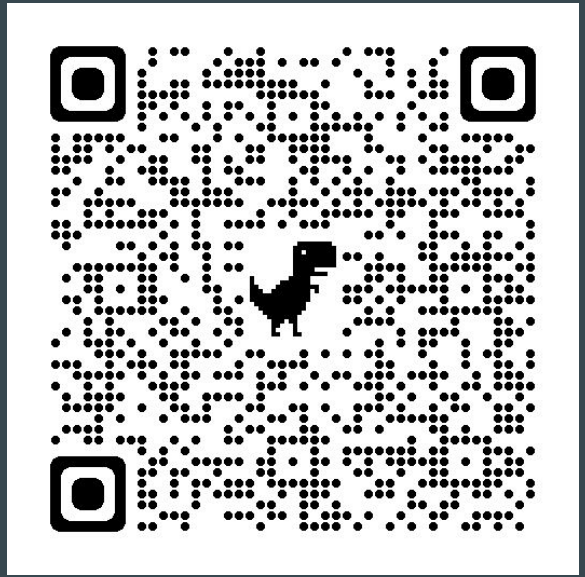
- Generierung von SBOMs für eigenentwickelte Software (automatisiert im Rahmen von CI/CD)
- Verpflichtung der Hersteller von Software zur Bereitstellung einer SBOM zu jeder ausgelieferten Version
- Einführung einer Software zur kontinuierlichen Analyse von SBOMs
- Zusätzlich Ausführungsschicht überwachen
- Plan zum Umgang mit Ergebnissen überlegen
- Kontinuierlich fixen und deployen

Infographic zur freien Verfügung

https://kitenco.de/download/SBOM-Infographic.png

Verpflichtung der Hersteller zur Bereitstellung eines Beipackzettels (SBOM) für Ihre Software

Cloud- / DevOps Engineer (m/w/d)



bis 75.000 €
Standort: Bonn
bis zu 100% remote

@kitenco1

New Work

Vielen Dank!

Kontakt:

Stephan Kaps
info@kitenco.de

www.kitenco.de

@kitenco1
https://www.xing.com/profile/Stephan_Kaps2
https://www.linkedin.com/in/stephan-kaps-b246b0ab/
http://de.slideshare.net/kitenco

https://open.spotify.com/album/2tHppzsY0ZPb57Xa7PRkEX

Quelle aller Hintergrundgrafiken ist https://www.pixabay.com