

Risiken in der Software Supply Chain identifizieren, mit OWASP Tools

Was haben wir aus Log4j gelernt?

Stephan Kaps

Leiter Softwareentwicklung Bundesamt für Soziale Sicherung

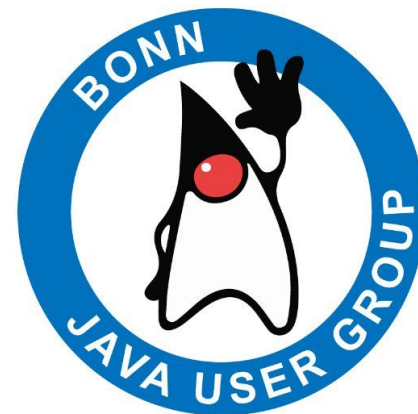
Gründer Java User Group Bonn

Java Entwickler & Architekt seit 2002

Weitere Schwerpunkte:

- Softwareentwicklungsprozesse
- BizDevSecOps
- OpenSource
- Speaker & Autor

loizodevops.org



Wer sind wir uns was machen wir?

(Aufsichts-) Behörde mit ca. 700 Mitarbeiter

ca. 60 Fachanwendungen (Produkte)

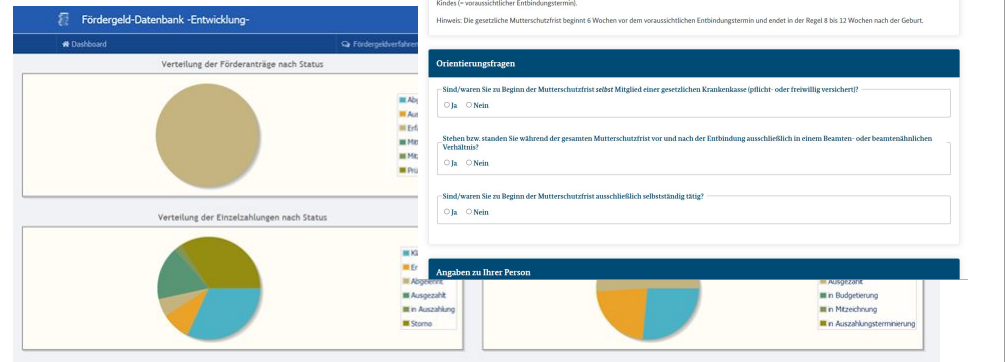
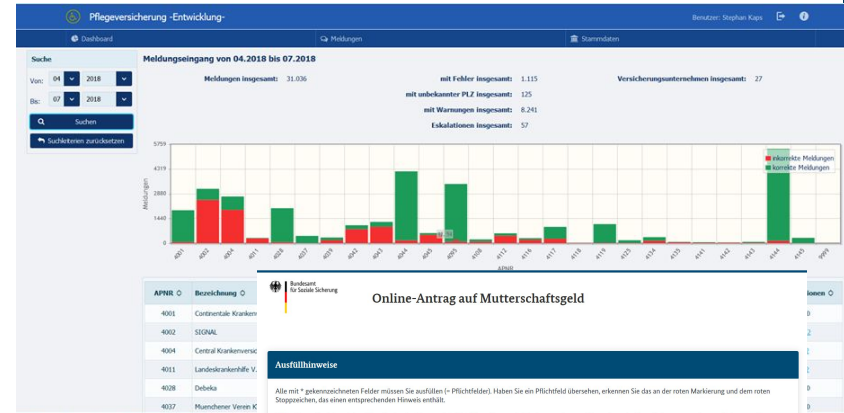
für interne Fachbereiche & Landesprüfdienste

seit 2016 agile Produktentwicklung (Scrum)

10 Entwickler und 2 Product Owner

Private-Cloud mit Containertechnologien

Sehr hohe Mitarbeiterzufriedenheit



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?

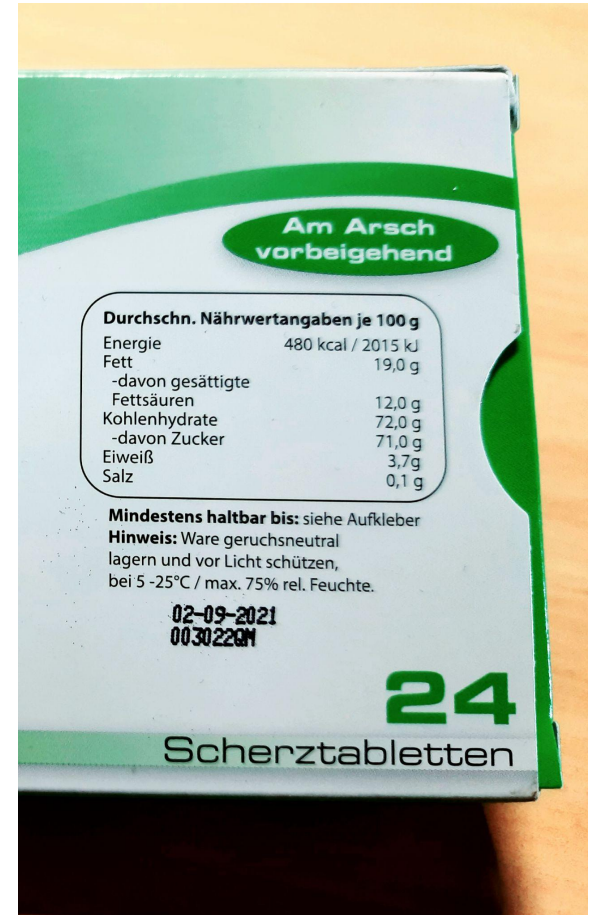
Haben wir einen Überblick über die
gesamte Anwendungslandschaft?



SBOM

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.

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

@kitencol

Why Do Developers Implementing State Management in React

Ever since Redux was released, we've seen a steady stream of tooling and frameworks that manage state in different ways, each with a different set of trade-offs. We don't know how to churn the JavaScript ecosystem to promote?

our experience, it is not based on a comprehensive market analysis.

Download

Adopt



No change

to see?

lecting what we came might have covered ar already. We re too many to talk e the Radar reflects

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

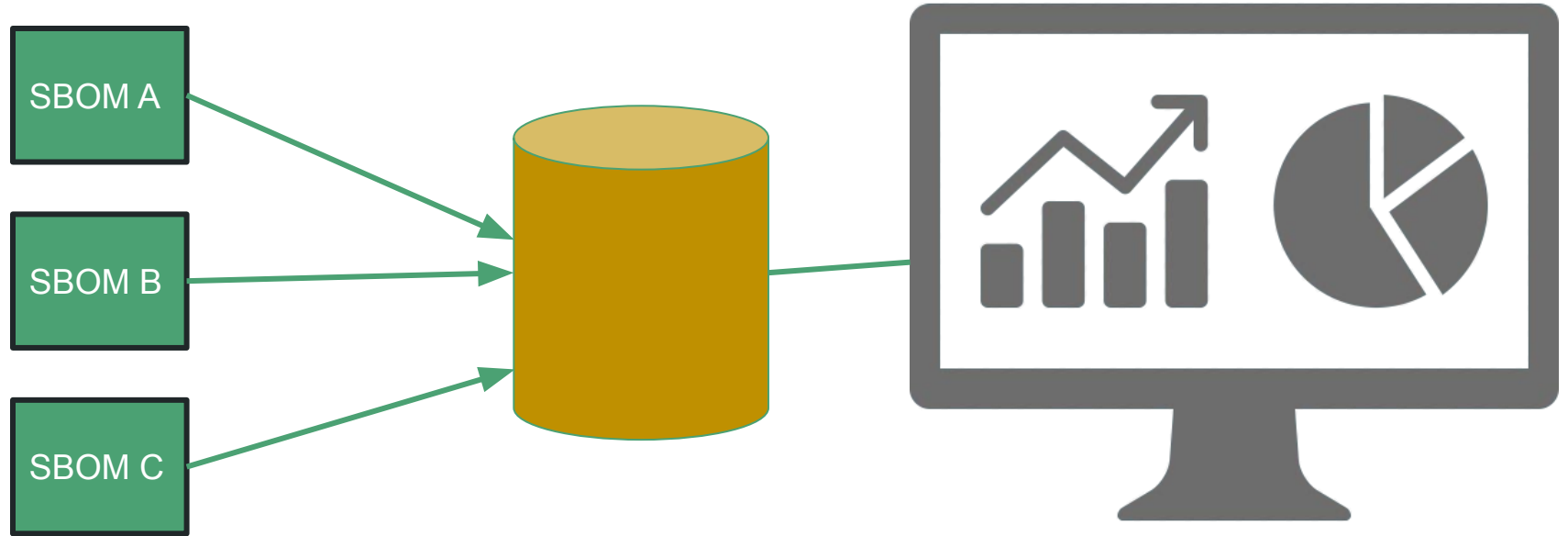
proprietary build-integration analysis github-ac...

opensource analysis

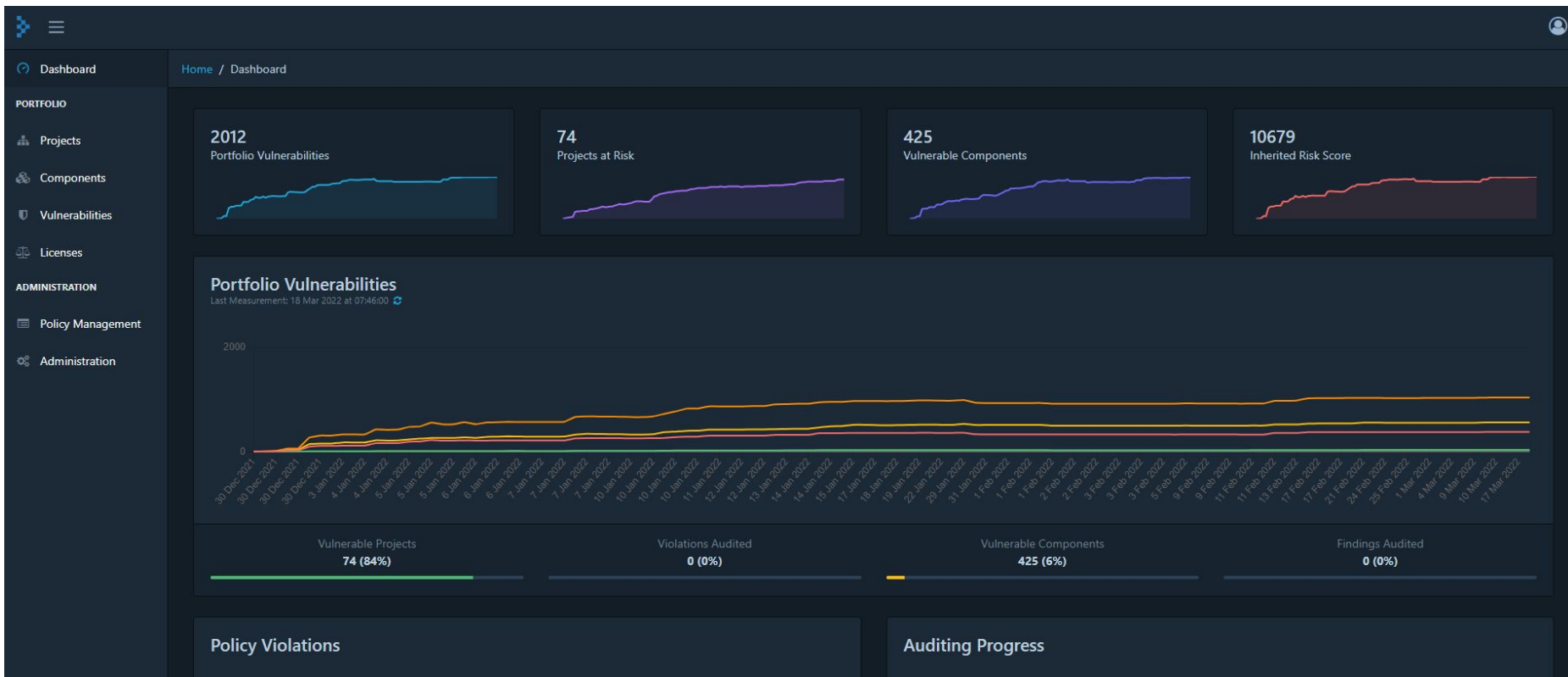
```
<component type="library" bom-ref="pkg:maven/org.apache.poi/poi@3.17?type=jar">
  <publisher>Apache Software Foundation</publisher>
  <group>org.apache.poi</group>
  <name>poi</name>
  <version>3.17</version>
  <description>Apache POI - Java API To Access Microsoft Format Files</description>
  <scope>optional</scope>
  <hashes>
    <hash alg="MD5">243bc3d431e4fadb79738719504c64f7</hash>
    <hash alg="SHA-1">0ae92292a2043888b40d418da97dc0b669fde326</hash>
    <hash alg="SHA-256">30181821dd2e849727b638b9e329aef4a64f3445c4142b13cf7a18bb3552edd</hash>
    <hash alg="SHA-384">f9a4db2b945cdba835b3c076c3f2bfa72767a5c02f81a57ffb4a5782651d5da66453c03e5c1be0505d1edad8a35c6f82</hash>
    <hash alg="SHA-512">a0c018d899935600b1077b343aac1a2e9050f84e3097e28b77869be9fc23475d4ac59bf375a2c96b6d824f06ba2d1873ee8d4495fb9bb412f848379ac7d11fb</hash>
    <hash alg="SHA3-256">84ba8a8001b44c04048d5ab1208640f0f5d54183231962cb180de7d7cfa7a386</hash>
    <hash alg="SHA3-384">2101589e294edd8d55f29be193e74c84afb8db754566b08b369c6c235e969af25ee8962e70f7c3dc262f7cbdebe3d57d</hash>
    <hash alg="SHA3-512">cc8b929d7f54aa98c606977bab56fcfe198e0bb21403b7b2a1bfff6bf2940901aee9d47bde493a98af376b4b1e75d110284023bd174b7774d454fac6e14b0606c</hash>
  </hashes>
  <licenses>
    <license>
      <id>Apache-2.0</id>
    </license>
  </licenses>
  <url>pkg:maven/org.apache.poi/poi@3.17?type=jar</url>
  <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>
```



Zusammenführen



OWASP Dependency Track



Continuous Security Monitoring

Operationalize Software Bill of Materials





Dashboard

Home / Projects

PORTFOLIO

Projects

Components

Vulnerabilities

Licenses

ADMINISTRATION

Policy Management

Administration



[+ 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	<input checked="" type="checkbox"/>	<div style="width: 100%;"><div style="width: 30%;"></div></div> 3 / 3
wms	0.1	18 Mar 2022 at 03:44:06	CycloneDX 1.3	13	<input checked="" type="checkbox"/>	<div style="width: 100%;"><div style="width: 60%;"></div></div> 2 / 1
rbus	0.6.0	18 Jan 2022 at 03:15:27	CycloneDX 1.3	69	<input checked="" type="checkbox"/>	<div style="width: 100%;"><div style="width: 70%;"></div></div> 1 / 7 / 6
rbus	0.7.0	18 Mar 2022 at 03:16:00	CycloneDX 1.3	24	<input checked="" type="checkbox"/>	<div style="width: 100%;"><div style="width: 30%;"></div></div> 3 / 3
zus-portal	1.0.0	18 Mar 2022 at 03:05:15	CycloneDX 1.3	3	<input checked="" type="checkbox"/>	<div style="width: 100%;"><div style="width: 10%;"></div></div> 1
ema-portal	1.0.0	18 Mar 2022 at 03:02:17	CycloneDX 1.3	0	<input checked="" type="checkbox"/>	<div style="width: 100%;"><div style="width: 0%;"></div></div> 0
para83a-online	1.0.0	18 Mar 2022 at 03:12:04	CycloneDX 1.3	50	<input checked="" type="checkbox"/>	<div style="width: 100%;"><div style="width: 50%;"></div></div> 1 / 5 / 5
mgs-agb-online	1.0.0	18 Mar 2022 at 03:24:55	CycloneDX 1.3	0	<input checked="" type="checkbox"/>	<div style="width: 100%;"><div style="width: 0%;"></div></div> 0
para80-online	1.0.0	18 Mar 2022 at 01:07:06	CycloneDX 1.3	50	<input checked="" type="checkbox"/>	<div style="width: 100%;"><div style="width: 50%;"></div></div> 1 / 5 / 5
imorsaftplogimporter	1.0.0	18 Mar 2022 at 02:19:28	CycloneDX 1.3	3	<input checked="" type="checkbox"/>	<div style="width: 100%;"><div style="width: 10%;"></div></div> 1

Showing 1 to 10 of 88 rows rows per page

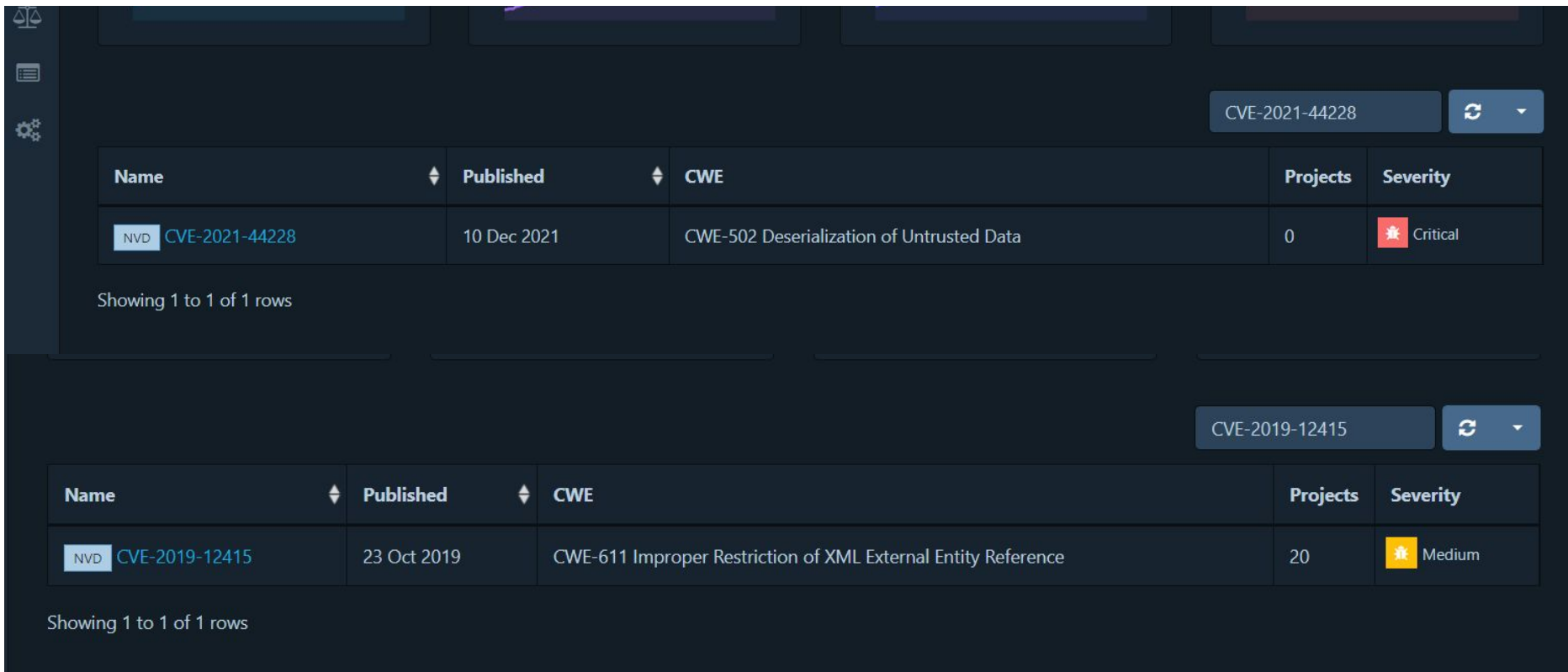
1 2 3 4 5 ... 9

Suche nach Komponenten

The screenshot displays a search interface for software components. At the top, there are four line charts: 'Portfolio Vulnerabilities', 'Projects at Risk', 'Vulnerable Components', and 'Inherited Risk Score'. Below these is a search bar with the following filters: 'Coordinates' (checked), 'Group' (log4j), and 'Version'. A 'Search' button is on the right. The main area contains a table with the following columns: Component, Version, Group, Package URL (PURL), CPE, SWID Tag ID, and Project Name.

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



The screenshot displays a search interface for vulnerabilities. It features a search bar at the top right with the text 'CVE-2021-44228' and a refresh button. Below the search bar is a table with the following columns: Name, Published, CWE, Projects, and Severity. The first row shows the vulnerability 'CVE-2021-44228' (NVD), published on '10 Dec 2021', with the CWE 'CWE-502 Deserialization of Untrusted Data', 0 projects, and a 'Critical' severity level. Below the table, it says 'Showing 1 to 1 of 1 rows'. A second search bar at the bottom right contains 'CVE-2019-12415' and a refresh button. Below this is another table with the same columns. The second row shows 'CVE-2019-12415' (NVD), published on '23 Oct 2019', with the CWE 'CWE-611 Improper Restriction of XML External Entity Reference', 20 projects, and a 'Medium' severity level. Below this table, it also says 'Showing 1 to 1 of 1 rows'.

Name	Published	CWE	Projects	Severity
NVD CVE-2021-44228	10 Dec 2021	CWE-502 Deserialization of Untrusted Data	0	Critical

Showing 1 to 1 of 1 rows

Name	Published	CWE	Projects	Severity
NVD CVE-2019-12415	23 Oct 2019	CWE-611 Improper Restriction of XML External Entity Reference	20	Medium

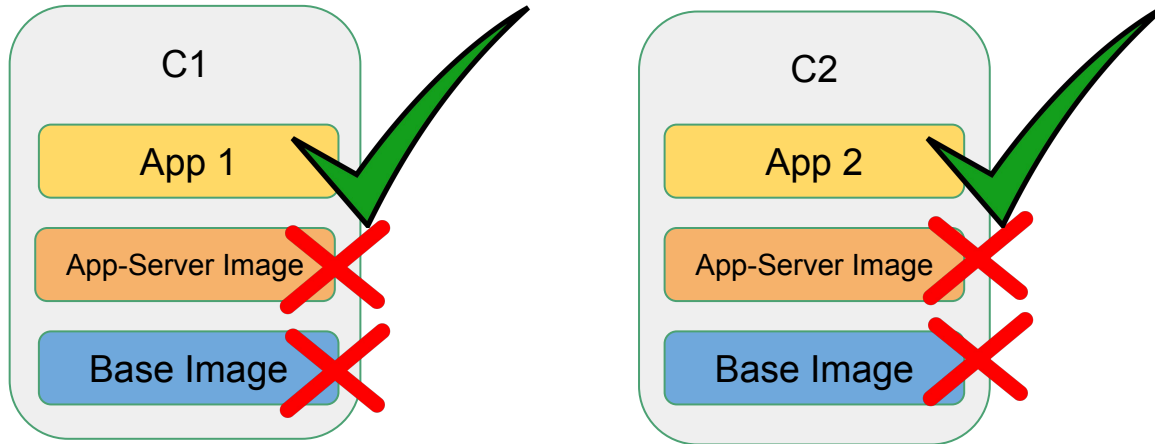
Showing 1 to 1 of 1 rows

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



Server / Hypervisor / OS Layer / Container Engine

SBOMs für Container Images und Filesystem



syft

<https://github.com/anchore/syft>

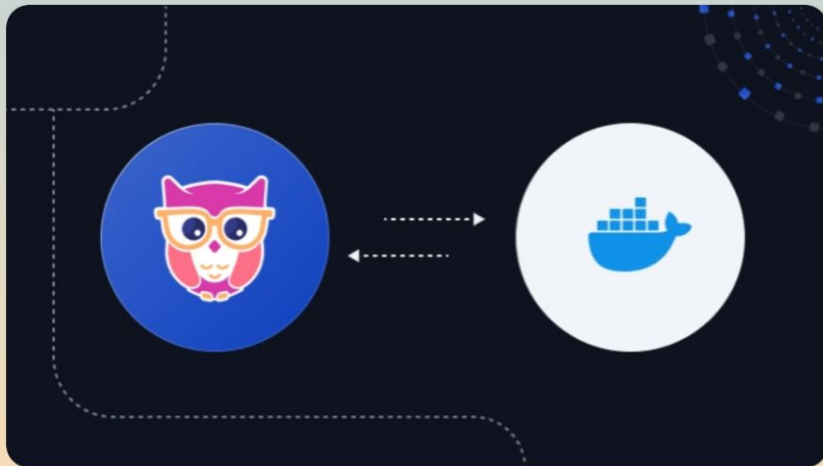


grype

<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



in

f

👍

🐦

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.

@kitencol

CLI Aufruf und direkte Auswertung

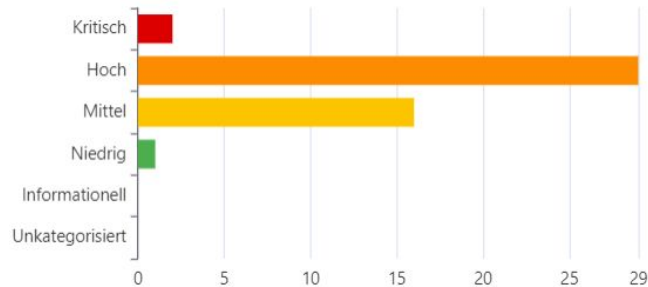
```
stage('Analyze Dependencies of Docker Image'){
  steps{
    script {
      dir ('out') {
        sh "syft -o cyclonedx-xml=sbom.xml ${env.IMAGE}:${env.TAG}"
        withCredentials([string(credentialsId: 'deptrack-apikey-text', variable: 'API_KEY')]) {
          dependencyTrackPublisher artifact: '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 ...

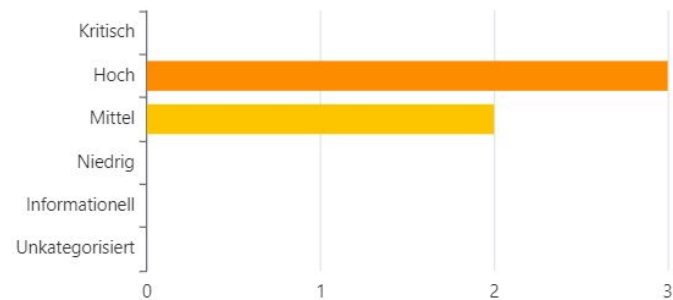
Project Name	Version	Classifier	Last BOM Import	BOM Format	Risk Score	Active	Policy Violations	Vulnerabilities
mgs-keycloak	19.0.1	Library	14 Sep 2022 at 01:33:29	CycloneDX 1.4	21	<input checked="" type="checkbox"/>	<div style="width: 100%;"><div style="width: 100%; background-color: #f08080;">11</div></div>	<div style="width: 100%;"><div style="width: 100%; background-color: #ffa500;">3</div><div style="width: 100%; background-color: #ffcc00;">2</div></div>
etw-docker-03.bvaetw.de/mgs/mgs-keycloak	19.0.1-19	Container	18 Aug 2022 at 17:45:40	CycloneDX 1.4	246	<input checked="" type="checkbox"/>	<div style="width: 100%;"><div style="width: 100%; background-color: #f08080;">3</div></div>	<div style="width: 100%;"><div style="width: 100%; background-color: #ffa500;">33</div><div style="width: 100%; background-color: #ffcc00;">20</div></div>



Zusammenfassung der Abhängigkeitsanalyse



Zusammenfassung der Abhängigkeitsanalyse





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

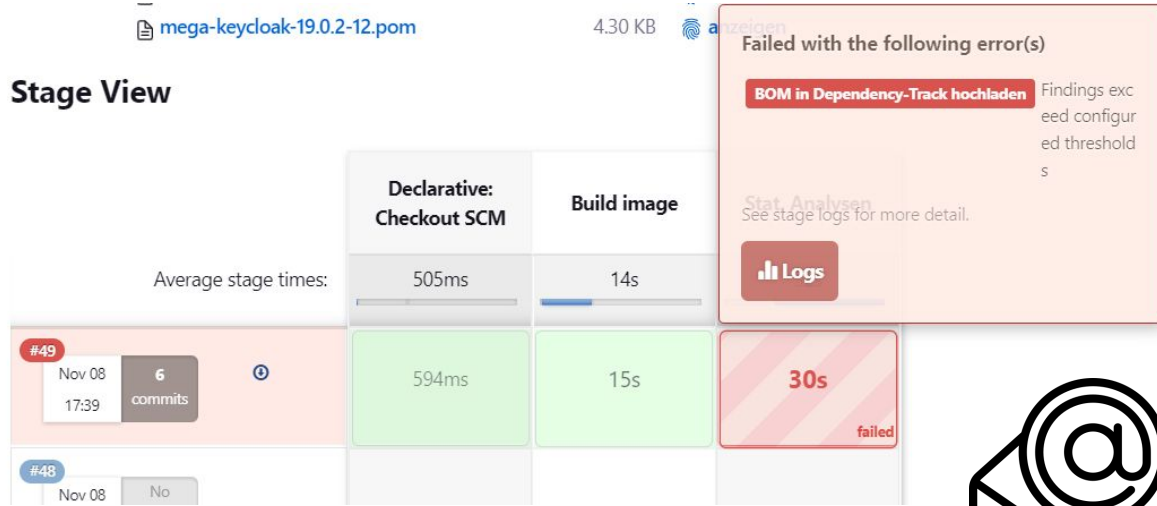


Component	Version	Group	Internal	License	Risk Score	Vulnerabilities
openldap	2.4.46-18.el8				96	1 16 2
libarchive	3.3.3-3.el8_5				40	5 5
freetype	2.9.1-4.el8_3.1				23	1 2 1
keycloak-services	19.0.1	org.keycloak			15	1 3 1
pip	9.0.3			MIT	13	2 1
keycloak-core	19.0.1	org.keycloak			13	2 1
libjpeg-turbo	1.5.3-12.el8				11	1 2
rhel	8.6				8	1 1
snakeyaml	1.30	org.yaml			8	1 1
okhttp	3.14.9	com.squareup.okhttp3			5	1

Showing 1 to 10 of 1208 rows 10 rows per page

Was jetzt?

Organisatorisch



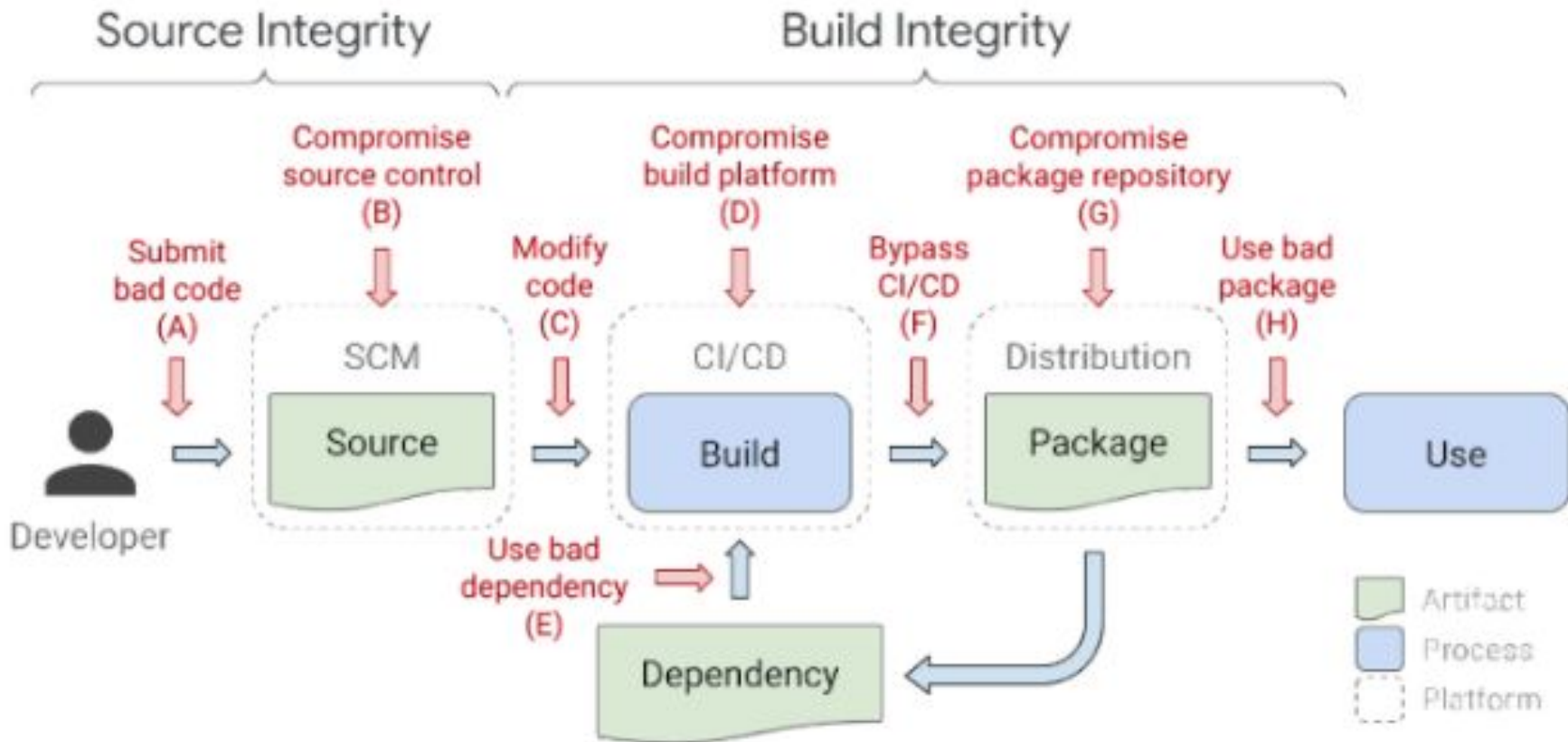
~~CVE-2022-12345~~



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

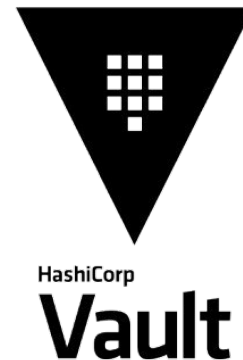
Github Branch Protection Rules



Secrets

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

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

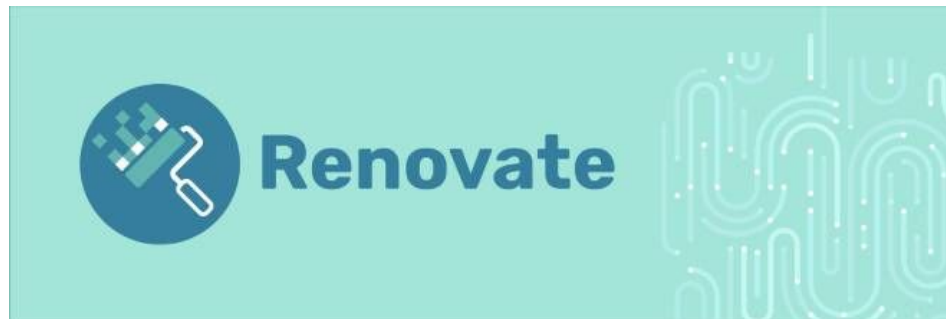


Automatische Updates für Dependencies

- Dependabot, Renovate und Ähnliches



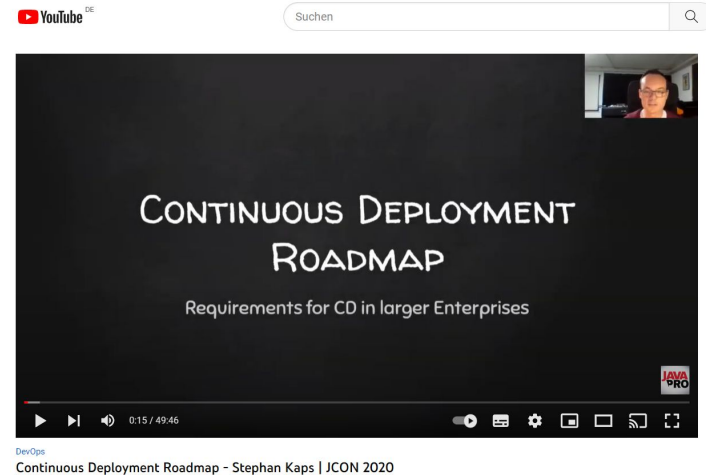
<https://github.com/dependabot>



<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

Wie man beginnt

Die Anforderungen an die IT-Sicherheit sind in den letzten Jahren erheblich gewachsen. Heute können Unternehmen, Behörden und Organisationen, die sich mit dem Internet beschäftigen, nicht mehr ohne IT-Sicherheit auskommen. Die Anforderungen an die IT-Sicherheit sind in den letzten Jahren erheblich gewachsen. Heute können Unternehmen, Behörden und Organisationen, die sich mit dem Internet beschäftigen, nicht mehr ohne IT-Sicherheit auskommen. Die Anforderungen an die IT-Sicherheit sind in den letzten Jahren erheblich gewachsen. Heute können Unternehmen, Behörden und Organisationen, die sich mit dem Internet beschäftigen, nicht mehr ohne IT-Sicherheit auskommen.

Mögliche Lösung:

Continuous Security Testing

Abb. 1: Screenshot des Buches



FROSCON 2016
Stephan Kaps: Sichere Softwareentwicklung

Container Security

- Angriffsfläche verkleinern
- Privilegien minimieren
- Docker Engine härten

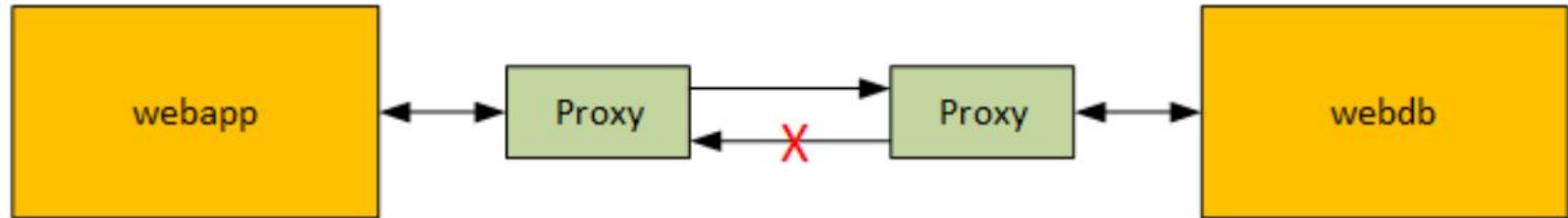


DevOps
In den sicheren Hafen: Einstieg in Container Security - Stephan Kaps | JCON 2020

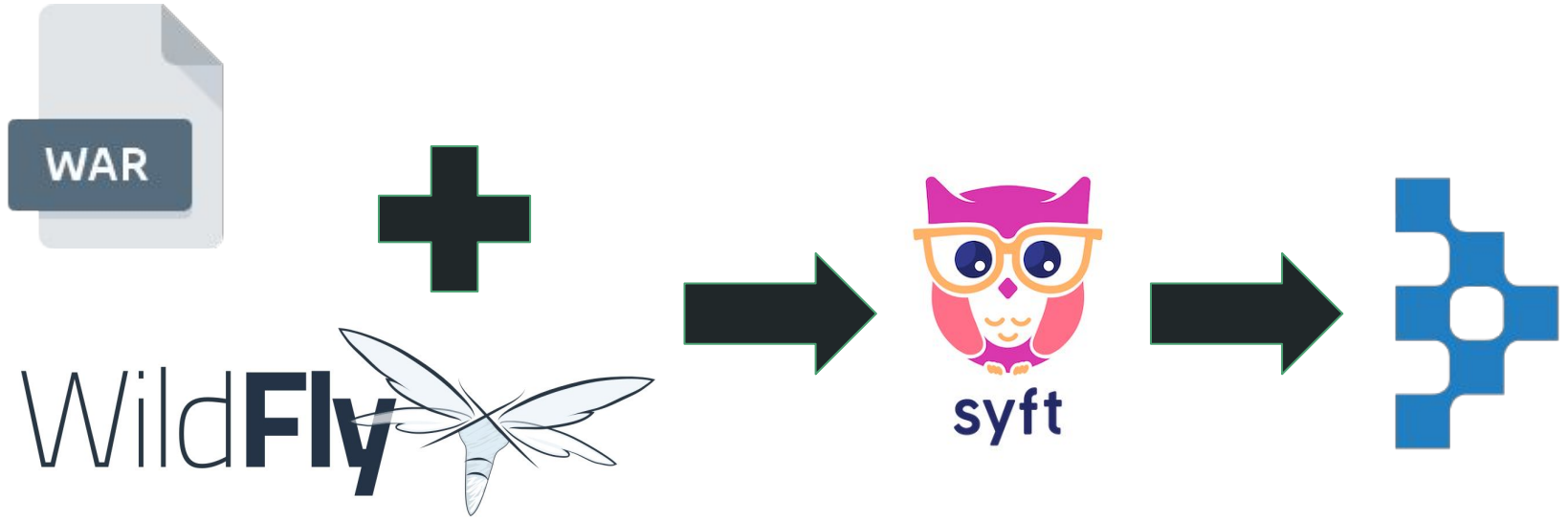


Isolierung

- 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 schadhafes 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>

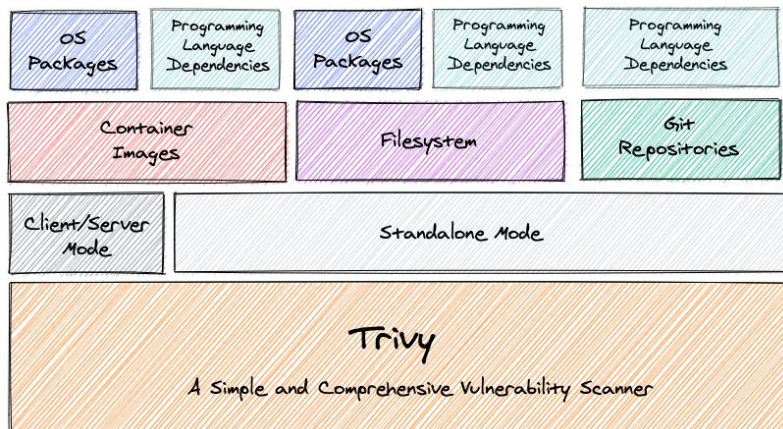
Secure Container Registry

- Vulnerability Scans
- Image Signing
- Pull Policies



HARBOR

<https://goharbor.io/>



Targets

Artifacts

Modes



<https://github.com/aquasecurity/trivy>

@kitencol

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

Info Artefakte

SCAN SCAN STOPPEN AKTIONEN 🔍 🔄

<input type="checkbox"/>	Artefakte	Pull Befehl	Tags	Signiert mit Cosign	Größe	Schwachstellen	Annotationen	Label	Push Zeit	Pull Zeit
<input type="checkbox"/>	 sha256:add50289		latest		86.14MiB	Keine Schwachstelle			04.11.22, 02:31	07.11.22, 01:00
<input type="checkbox"/>	  sha256:e7d34034				86.14MiB	Keine Schwachstelle			27.10.22, 14:39	07.11.22, 01:01
<input type="checkbox"/>	  sha256:0edb0aea				88.11MiB	H 1 Gesamt - 1 Fixable			04.10.22, 22:42	07.11.22, 01:01

Einträge pro Seite 15 1 - 3 von 3 Einträge

Pull Policy

The screenshot shows the Harbor web interface. The top navigation bar includes the Harbor logo and a search field. The left sidebar contains a menu with items like 'Projekte', 'Logs', 'Administration', 'Nutzer', 'Robot-Zugänge', 'Gruppen', 'Registries', 'Replikationen', 'P2P-Verteilung', 'Label', 'Projekt-Begrenzungen', 'Schwachstellen Scan', 'Clean Up', and 'Konfiguration'. The main content area is titled 'Konfiguration' and contains several sections: 'Projekt Registry' with an 'Öffentlich' checkbox; 'Deployment Sicherheit' with 'Cosign' and 'Verhindere den Download von Images mit Schwachstellen' checked; 'Scannen auf Schwachstellen' with 'Images automatisch beim Hochladen scannen' checked; and 'CVE Allowliste' with 'System-Allowliste' selected. At the bottom, there is a 'HINZUFÜGEN' button, a 'VON SYSTEM KOPIEREN' button, and a 'Läuft ab am' field with a dropdown menu set to 'Läuft niemals ab' and a checked 'Läuft niemals ab' checkbox. A text input field contains 'CVE-2021-43816'.

Harbor

Suche Harbor...

Zusammenfassung Repositories Mitglieder Labels Scanner P2P-Verteilung Policy Robot-Zugänge Webhooks Logs **Konfiguration**

Projekte

Logs

Administration

Nutzer

Robot-Zugänge

Gruppen

Registries

Replikationen

P2P-Verteilung

Label

Projekt-Begrenzungen

Schwachstellen Scan

Clean Up

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

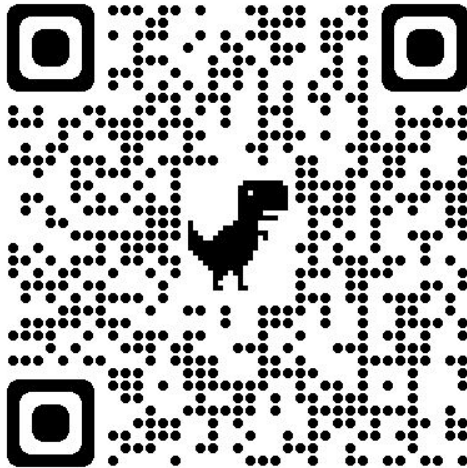
CVE-2021-43816

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 Beispackzettels (SBOM) für Ihre Software

WARUM?

Um Risiken in der Software-Supply-Chain frühzeitig zu identifizieren

WAS BEDEUTET DAS?

Es war NEM, das erste mal, das eine Schwachstelle in einer Open-Source-Bibliothek für Überstunden in 17 Millionen genutzt hat. Doch im Dezember 2022 war die CVE-2022-44228 zu Longi bzw. Logshell wegen in die Speicher und die Spring-Code des Betriebes.

Doch was haben wir daraus gelernt?

Wären wir informiert gewesen oder Maßnahmen ergriffen, um bei der nächsten Open-Source-Bibliothek offene Angewandte zu können? Können wir z.B. schneller herausfinden, in welchen Produkten eine Bibliothek in einer bestimmten Version verwendet wird?

Hätten wir aktuelle Beispackzettel zu jeder Software, sogenannte SBOM, wären wir dazu in der Lage.

Was ist eine SBOM?



SBOM steht für Software Bill of Materials und ist eine Tabelle in Form einer maschinenlesbaren Struktur, die enthält eine Auflistung aller verwendeten Bibliotheken bzw. Komponenten, sowie deren direkt und indirekte Abhängigkeiten und deren zugehörige Versionen. Solche Daten können viele weitere Informationen enthalten wie z.B. Lizenzinformationen.

Ein weiteres Beispiel sind SBOM (Standard Software Package Data Exchange), die sich als SBOM: SBOM:2021 generieren. Ein Open-Source-Tool wie CycloneDX von OWASP oder Syft von Aquent können SBOM generieren.

Log4Shell oder SolarWinds

Was haben wir daraus gelernt?



PRODUKT-INHALT IST UNBEKANNT

Alles ist bekannt ist Software, was wir wir wissen, was im Code ist, Kaufverfahren für Komponenten wie Binaries, Images ist auch bekannt, ist dieses Produkt von einer anderen Software benutzt ist.



RECHERCHEN SIND AUFWENDIG

Welches Open-Source-Software-Komponente eine spezifische, kritische Komponenten auf Web, Github oder andere oder in Tools per CI/CD oder per manuell bei jedem einzelnen Release überprüft werden.



WIR SIND EVENTUELL ANGREIFBAR

Indirekt wird ein Angriff, der ein Produkt angreift und ein Kommando führt, muss viele Angreifer der Hersteller zu erhalten. Angreiferische können Systeme angreifen werden.

Was können wir selbst tun?



Regelmäßige Analyse auf Sicherheitslücken

Für open-source Software bzw. von einer SBOM zur Verfügung steht, ist die Analyse der Sicherheitslücken in verschiedenen Bibliotheken rechtlich möglich. Durch geeignete Tools ist auch ein Überblick über das gesamte Anwendungsspektrum aufwandarm erstellbar.

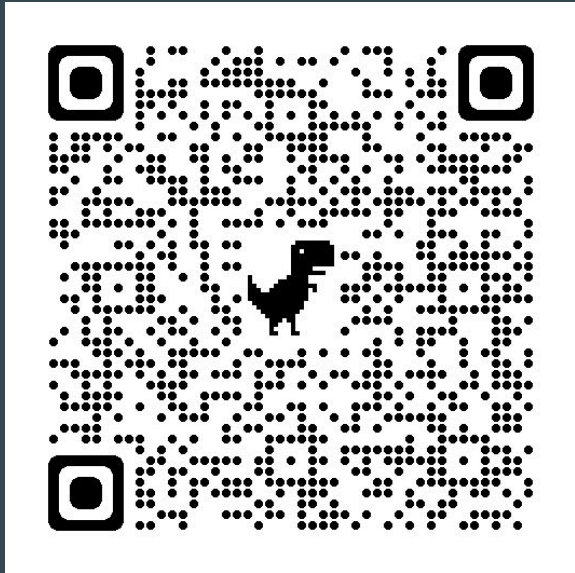


Aufnahme der Verpflichtung in Ausschreibungen

In künftigen Ausschreibungen sollte aufgenommen werden, dass Hersteller von Software verpflichtet sind, zu jeder angeforderten Version auch eine SBOM bereitzustellen. Zusätzlich sollten SLAs aufgenommen werden, die eine zeitliche Definition von Sicherheitslücken festlegen.

© Kitenco AG (E.ON Energy Research Center) | Cyber-Sicherheitsentwicklung im Rahmen der Secure Engineering

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



bis 75.000 €
Standort: Bonn
bis zu 100% remote

New Work



Vielen Dank!

Kontakt:

Stephan Kaps
info@kitenco.de

www.kitenco.de



@kitencol



https://www.xing.com/profile/Stephan_Kaps2



<https://www.linkedin.com/in/stephan-kaps-b246b0ab/>



<http://de.slideshare.net/kitenco>

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

@kitencol

MC Flavour

Der Nächste Request



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