



SBOM-Generierung ohne Quellcode: Herausforderungen und Lösungen

#BinarySoftwareCompositionAnalysis

#SupplyChainSecurity

#SBOM

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung



Das Technologietransfer- &
Existenzgründungs-Center
der Universität Paderborn

Nutella Bill of Materials



Kakao, Lieferant X, Charge 9111342

Haselnüsse, Lieferant Y, Charge 597557

Zucker, Hersteller Y, Charge 4711123

Ein guter Ansatz



pom.xml



package.json



requirements.txt



Makefile, CmakeLists.txt,
vcpkg.json, conanfile.txt,

...

```
1 annotated-types==0.5.0
2 beanie==1.21.0
3 datasketch==1.5.9
4 dnspython==2.3.0
5 idna==3.4
6 loguru==0.7.2
7 motor==3.3.1
8 mypy==1.4.1
9 mypy-extensions==1.0.0
10 numpy==1.25.1
11 psycpg2-binary==2.9.6
12 pydantic==1.10.0
13 pydantic_core==2.1.2
14 pymongo==4.5.0
15 python-dotenv==1.0.0
16 python-magic==0.4.27
17 requests==2.31.0
18 scipy==1.11.1
19 semver==3.0.1
20 SQLAlchemy==1.4.48
21 types-requests==2.31.0.1
22 types-urllib3==1.26.25.13
23 typing_extensions==4.7.1
```

... aber nicht immer möglich

- Legacy Software, insb. C/C++
 - Keine Paketmanager
 - Komplexe Build-Prozesse
- Abschließende Qualitätskontrolle
- Zulieferungen

Lösungsansatz

Die gebaute Software als alleinige Informationsquelle



Hintergrund

Komponenten sind unterschiedlich leicht zu identifizieren

Komponenten sind dynamisch oder statisch gelinkt

- .so | .a
- .dll | .obj



... oder header-only

Erkennungsmarker

- (1) File Hashes & Meta Data
- (2) Strings & konstante Werte
- (3) Static Binary Code Bytes



File Hashes

- Erstellung von Paketmuseen
 - Binärpakete
 - Binärgleicher Nachbau



Code-Marker

File Hash

Meta Data

Code Segment

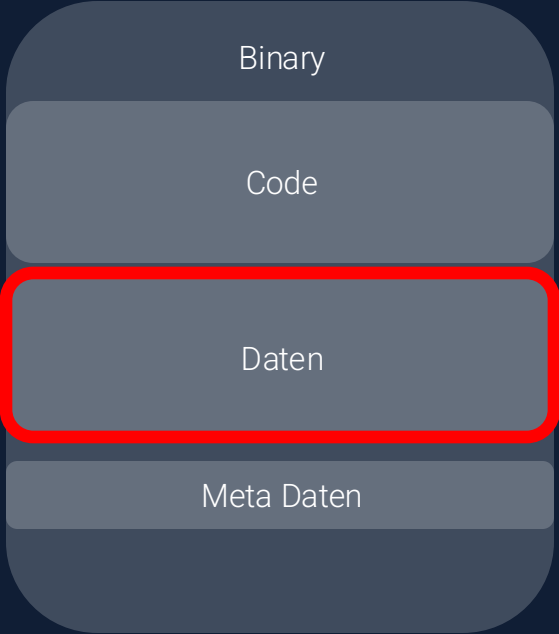
Data Segment

Kombinatorische Explosion

- Generierter Code variiert je nach
 - Compiler, Compiler-Version, Compiler-Flags
 - CPU-Arch., Endianness, Bitness
-  **LOTTO[®]6aus49** : 13.983.816 Möglichkeiten

Daten-Marker

00	FF	FF	FF	FF	FF	FF	FF	FF	FF	0A	FF	FF	FF	FF	FF	
FF	FF	FF	FF	14	15	16	FF	FF	FF	FF	FF	FF	1E	FF	FF	
FF	FF	FF	FF	FF	FF	FF	FF	28	FF	2A	2B	2C	2D	2E	2F	(.*+, -./
30	31	32	33	FF	FF	FF	FF	FF	FF	FF	FF	3C	FF	FF	FF	0123.....<...
FF	FF	FF	FF	FF	FF	46	47	FF	FF	FF	FF	FF	FF	FF	FF	FG.....
50	FF	FF	FF	FF	FF	FF	FF	FF	FF	5A	FF	FF	FF	FF	FF	P.....Z.....
FF	FF	FF	FF	64	00	73	73	6C	5F	6C	69	62	2E	63	00	d.ssl_lib.c.
41	4C	4C	3A	21	41	44	48	3A	2B	52	43	34	3A	40	53	ALL:!ADH:+RC4:@S
54	52	45	4E	47	54	48	00	73	73	6C	32	2D	6D	64	35	TRENGTH.ssl2-md5
00	73	73	6C	33	2D	6D	64	35	00	73	73	6C	33	2D	73	.ssl3-md5.ssl3-s
68	61	31	00	53	53	4C	76	32	00	54	4C	53	76	31	00	ha1.SSLv2.TLSv1.
4F	70	65	6E	53	53	4C	20	30	2E	39	2E	37	61	20	46	OpenSSL-0.9.7a-F
65	62	20	31	39	20	32	30	30	33	00	00	A4	F5	EB	FF	eb-19-2003.....
94	F5	EB	FF	94	F5	EB	FF	94	F5	EB	FF	94	F5	EB	FF	
94	F5	EB	FF	94	F5	EB	FF	94	F5	EB	FF	94	F5	EB	FF	



„OpenSSL 1.0.1e-fips 11 Feb 2013“

Keine kombinatorische Explosion, aber...

- Oftmals nur wenige Marker vorhanden
- False Positives („None“, „Error“, „Password“)
- Top 10?

Keine kombinatorische Explosion, aber...

- Oftmals nur wenige Marker vorhanden
- False Positives („None“, „Error“, „Password“)

Platz	String
1	none
2	true
3	false
4	
5	unknown
6	default
7	help
8	%s/%s
9	version
10	debug

Intelligentes Brute Force

- Erstellung von „Paketmuseen“
- Großskalige Generierung von Code-Markern mit der höchsten Trefferwahrscheinlichkeit
- Kombinierte Anwendung mit **bestimmten** Daten-Marken

Die nächsten Schritte

Fordern Sie unsere Technologie heraus!



Jan Stijohann
Gründer | CTO

stijohann@langlauf.io
www.langlauf.io

Halle 6 Startups@it-sa



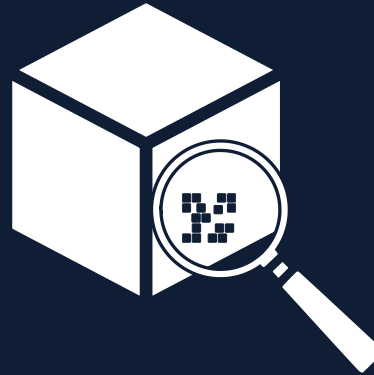
Anwendungsfälle

SBOM-Generierung ohne Quellcode

01 **Endkontrolle** der eigenen
(komplexen) Produkte

02 Compliance Prüfung von
Legacy Software

03 Compliance-Prüfung von
zugelieferten Komponenten



04 Generierung von fehlenden
SBOMs für **zugelieferte
Komponenten**

05 **Nachweis** von Patent- und
Lizenzverletzungen in
Drittkomponenten

06 **Abschätzung von
Compliance Risiken**, z.B. bei
M&As und VC-Beteiligungen

Backup Folien



Über uns



Start der Produktentwicklung in 2021 | Paderborn / Lippstadt



10+ Mitarbeiter | Ex-Siemens, Rohde & Schwarz, CGI, BMW

Anwendungsfälle unserer Technologie



Prüfungen zugelieferter Software auf Security-Compliance



Automatische Software Analyse für die Exportkontrolle



SBOM-Generierung – auch ohne Quellcode