SB💣💣M: Making SBOMs Play Together

Jacopo Bufalino, CNAM &
Agathe Blaise, Thales

#whoarewe

KubeCon CloudNativeCon Europe 2026

Jacopo Bufalino

Ph.D. Candidate

Cnam, Paris & Aalto University, Espoo

Agathe Blaise

Research engineer

Thales SIX GTS France

European project funding

SEC 4AI4 SEC

Secure AI-enhanced systems and
AI-enhanced Systems for Security
(**Sec4AI4Sec**)

https://www.sec4ai4sec-project.eu/
Available here →

Where we left...

KubeCon CloudNativeCon Europe 2026

KubeCon Europe 2025: studied the impact of container obfuscation on SBOM generation

Enhancing Software Composition Analysis Resilience Against Container Image Obfuscation

Agathe Blaise, Thales &
Jacopo Bufalino, CNAM

- Showed the limitations of SBOM generation tools in handling images in which important metadata are deleted

Where we left...

KubeCon CloudNativeCon Europe 2026

Technique(s)	Trivy		Syft		Syft (All)		Scout		Microsoft		Gcloud		Amazon	
	V	P	V	P	V	P	V	P	V	P	V	P	V	P
BASE (no obscuration)	1164	441	625	448	625	448	123	585	154	429	722	441	471	587
OSPKG	6	11	25	25	625	448	10	23	154	429	6	12	N/A	0
URL	1164	441	625	448	625	448	123	585	154	429	722	441	471	587
LINK	1164	441	625	448	625	448	123	585	154	429	722	441	471	587
DEP	1164	441	625	448	625	448	123	585	154	429	722	441	471	579
PKG														576
ALIAS														587
PACK														587
OS														587
OS + OSPKG														0
DEP + PKG														568
OS + DEP														579
OS + PKG														576
OS + OSPKG +														0
OS + OSPKG +														0
OS + OSPKG + DEP	6	11	27	25	625	448	6	23	154	429	6	12	N/A	0
OS + OSPKG + DEP + LINK	6	11	27	25	625	448	6	23	154	429	6	12	N/A	0
OS + OSPKG + DEP + PKG	0	0	21	13	625	448	0	11	148	429	N/A	0	N/A	0
OS + OSPKG + DEP + PACK	6	12	27	25	27	25	6	23	0	0	6	12	N/A	0
OS + OSPKG + DEP + ALIAS	6	11	25	24	625	448	6	22	154	429	6	12	N/A	0
OS + OSPKG + DEP + ALIAS + PACK	6	12	25	24	27	25	6	22	0	0	6	12	N/A	0
OS + OSPKG + DEP + ALIAS + PKG	0	0	19	12	625	448	0	10	148	429	N/A	0	N/A	0

→ No one noticed that each tool has different numbers of **packages** and **vulnerabilities** as ground truth!

Where we left...

Technique(s)	Trivy		Syft		Syft (All)		Scout		Microsoft		Gcloud		Amazon	
	V	P	V	P	V	P	V	P	V	P	V	P	V	P
BASE (no obscuration)	1164	441	625	448	625	448	123	585	154	429	722	441	471	587
OSPKG	6	11	25	25	625	448	10	23	154	429	6	12	N/A	0
URL	1164	441	625	448	625	448	123	585	154	429	722	441	471	587
LINK	1164	441	625	448	625	448	123	585	154	429	722	441	471	587
DEP	1164	441	625	448	625	448	123	585	154	429	722	441	471	579
PKG													469	576
ALIAS													471	587
PACK													471	587
OS													471	587
OS + OSPKG													N/A	0
DEP + PKG													465	568
OS + DEP													471	579
OS + PKG													469	576
OS + OSPKG + PACK													N/A	0
OS + OSPKG + PKG													N/A	0
OS + OSPKG + DEP	6	11	27	25	625	448	6	23	154	429	6	12	N/A	0
OS + OSPKG + DEP + LINK	6	11	27	25	625	448	6	23	154	429	6	12	N/A	0
OS + OSPKG + DEP + PKG	0	0	21	13	625	448	0	11	148	429	N/A	0	N/A	0
OS + OSPKG + DEP + PACK	6	12	27	25	27	25	6	23	0	0	6	12	N/A	0
OS + OSPKG + DEP + ALIAS	6	11	25	24	625	448	6	22	154	429	6	12	N/A	0
OS + OSPKG + DEP + ALIAS + PACK	6	12	25	24	27	25	6	22	0	0	6	12	N/A	0
OS + OSPKG + DEP + ALIAS + PKG	0	0	19	12	625	448	0	10	148	429	N/A	0	N/A	0

Difference mostly on OS packages
Worrying because base layers are reshared a lot

Where we left...

KubeCon CloudNativeCon Europe 2026

Technique(s)	Trivy		Syft		Syft (All)		Scout		Microsoft		Gcloud		Amazon	
	V	P	V	P	V	P	V	P	V	P	V	P	V	P
BASE (no obscuration)	1164	441	625	448	625	448	123	585	154	429	722	441	471	587
OSPKG	6	11	25	25	625	448	10	23	154	429	6	12	N/A	0
URL	1164	441	625	448	625	448	123	585	154	429	722	441	471	587
LINK	1164	441	625	448	625	448	123	585	154	429	722	441	471	587
DEP	1164	441	625	448	625	448	123	585	154	429	722	441	471	579
PKG													469	576
ALIAS													471	587
PACK													471	587
OS													471	587
OS + OSPKG													N/A	0
DEP + PKG													465	568
OS + DEP	1164	441	9	448	625	448	6	18	154	429	6	12	471	579
OS + PKG	1158	430	3	436	625	448	0	6	148	429	N/A	0	469	576
OS + OSPKG + PACK	6	12	27	25	27	25	6	23	0	0	6	12	N/A	0
OS + OSPKG + PKG	0	0	21	13	625	448	0	11	148	429	N/A	0	N/A	0
OS + OSPKG + DEP	6	11	27	25	625	448	6	23	154	429	6	12	N/A	0
OS + OSPKG + DEP + LINK	6	11	27	25	625	448	6	23	154	429	6	12	N/A	0
OS + OSPKG + DEP + PKG	0	0	21	13	625	448	0	11	148	429	N/A	0	N/A	0
OS + OSPKG + DEP + PACK	6	12	27	25	27	25	6	23	0	0	6	12	N/A	0
OS + OSPKG + DEP + ALIAS	6	11	25	24	625	448	6	22	154	429	6	12	N/A	0
OS + OSPKG + DEP + ALIAS + PACK	6	12	25	24	27	25	6	22	0	0	6	12	N/A	0
OS + OSPKG + DEP + ALIAS + PKG	0	0	19	12	625	448	0	10	148	429	N/A	0	N/A	0

Our focus: SBOM interoperability across different tools

KubeCon CloudNativeCon Europe 2026

Our objectives

Our objectives

We will cover **3 main objectives** during this presentation:

- Why **different packages** are reported for the same app?
- What is the impact on **vulnerability detection** accuracy?
- Is there a way to **automatically mitigate** inconsistent SBOM outputs?

KubeCon CloudNativeCon Europe 2026

Background

SBOMs

A **SBOM** is a structured inventory that **lists all components**, dependencies, and metadata of a digital product.

Similarly, an **AIBOM** is a comprehensive list that shows all the **AI models**, tools, and datasets used in a software application.

Package identification

Package URLs (pURL) are (ECMA) **standardized** strings that uniquely identify software packages across package managers and repositories.

Format:

```
pkg:type/namespace/name@version?qualifiers#subpath
```

optional

optional

Examples:

```
pkg:npm/animation@12.0.0
pkg:npm/@angular/animation@12.0.0
pkg:npm/@angular/animation@12.0.0?platform=linux
```

What do we do with SBOMs?

Vulnerability management

- Read the package identifiers of the SBOM
- Compare them against a database of vulnerabilities
- Either on our own or on third party SBOMs

Compliance check

- Upcoming **Cyber Resilience Act (CRA)** and **Digital Networks Act**
- **AI Act** regarding the AIBOM

KubeCon CloudNativeCon Europe 2026

Let’s plan our research

Steps

1. Select **tools**
2. Select containerized **applications**
3. Compute the **SBOM** of each application using each tool
4. Use each tool to find **vulnerabilities** using the SBOM generated by every other tool (test matrix)
5. Study and analyze the **differences**

SBOM generation tools

Tool	*Company*	*Open-source*	*SBOM Output*	*Vulnerability scan*	*SBOM Input*
Amazon Inspector	Amazon	✓	✓	✓	~
Syft	Anchore	✓	✓	✘	N/A
Grype	Anchore	✓	N/A	✓	✓
Docker Scout	Docker	✘ (free)	✓	✓	✓
Artifact Analysis	Google	✘	✓	✓	✘
sbom-tool	Microsoft	✓	✓	✘	✘
Defender for Cloud	Microsoft	✘	✘	✓	✘
Trivy	Trivy	✓	✓	✓	✓

Dataset

We selected the **20 most downloaded** container images in DockerHub with support for both Debian and Alpine

Dataset	***Total # of packages***	***Distinct # of packages***	***# of vulnerabilities***
Top 20 Debian	3,651	747	261
Top 20 Alpine	741	442	78

KubeCon CloudNativeCon Europe 2026

Our findings

KubeCon CloudNativeCon Europe 2026

SBOM analysis

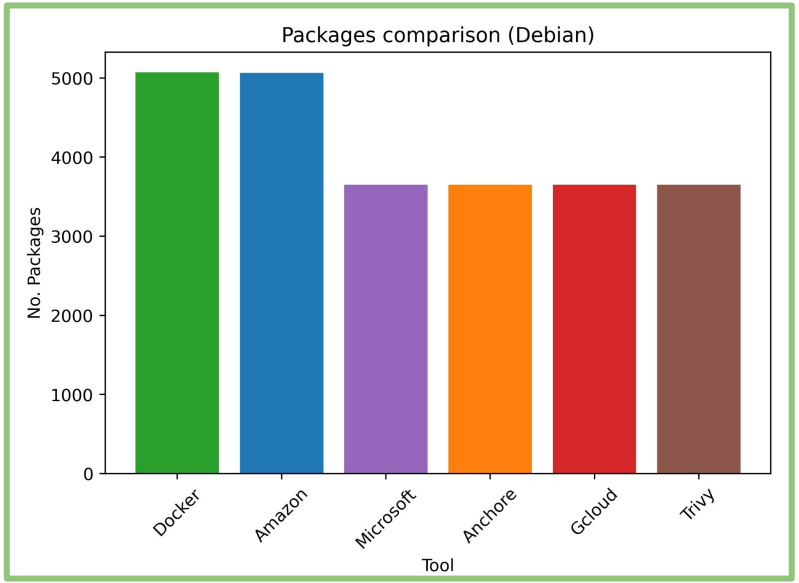
Look at the packages

Tools report different number of packages

Are they really that different?

Binary vs source package:

- `shadow:`
  - `login`
  - `passwd`
  - `libsubid`
  - `uidmap`



Misc packages:

- Some tools create additional packages to record information in SBOMs

Compare pURLS

Tools report different number of packages

Are they really that different?

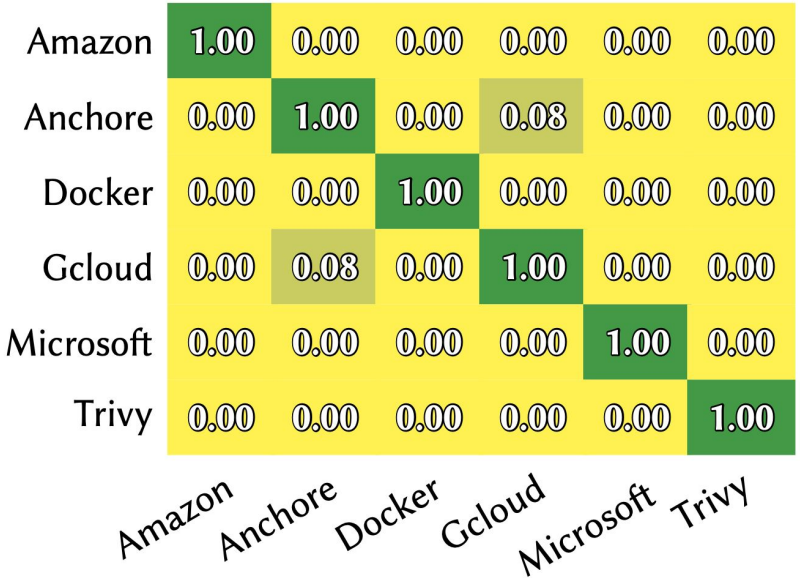
Compared pURLs:

1. Using name and version only
2. As they are

Outcome:

- Pac
  but

WHY???



Compare pURLS

KubeCon CloudNativeCon Europe 2026

Example for the same package: debian `python3-magics++`

Tool	pURL
Standard	pkg:deb/debian/python3-magics%2B%2B@2:1.5.8-1? arch=amd64&distro=bookworm
Amazon	**pkg:dpkg/python3-magics++**@**1.5.8-1**?arch=AMD64&epoch=1&upstream=python3-magics++-1.5.8-1.src.dpkg
Anchore	pkg:deb/debian/python3-magics%2B%2B@2:1.5.8-1?arch=amd64&upstream=magics-python&distro=debian-12
Google	pkg:deb/debian/python3-magics%2B%2B@**2%3A1.5.8-1**?arch=amd64&distro=debian-12&upstream=magics-python
Microsoft	**pkg:deb/debian/python3-magics++**@2:1.5.8-1
Docker	pkg:deb/debian/python3-magics%2B%2B@2:1.5.8-1?os_version=12&os_name=debian&os_distro=bookworm
Trivy	pkg:deb/debian/python3-magics%2B%2B@**1.5.8-1**?arch=amd64&distro=debian-12.11&epoch=2

KubeCon CloudNativeCon Europe 2026

CVE analysis

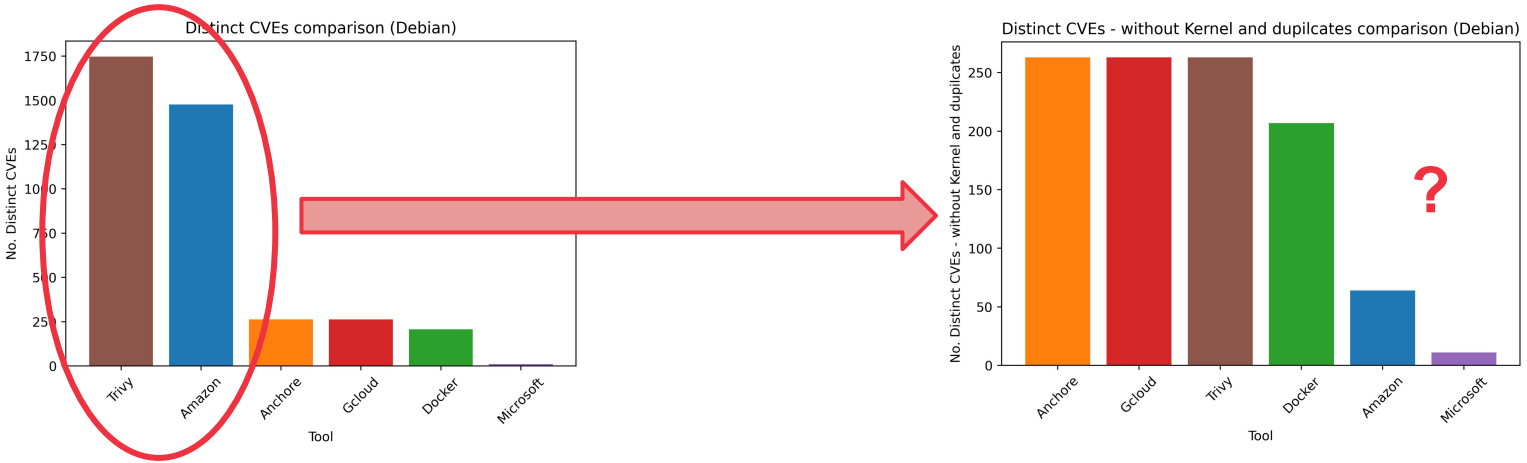
Let’s look at the vulnerabilities

Can we cluster the tools with similar CVEs?

Look at the type of CVEs

Some tools report kernel vulnerabilities

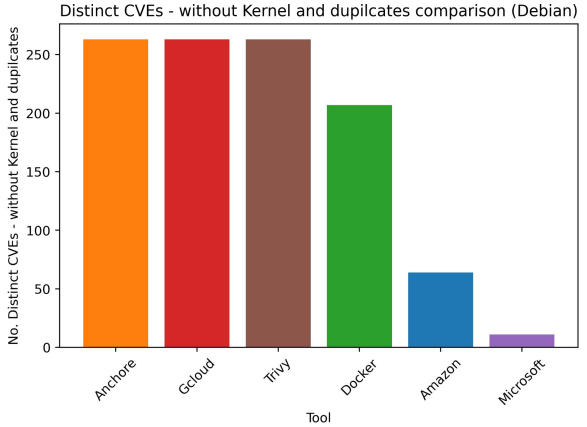
→ ***Containers share kernel with host!***



Let’s look at the vulnerabilities

Why do we see different CVEs?

- **Irrelevant CVEs**: tools report CVEs that do not affect the analyzed environments
- **pURL parsing**: not enough info to identify a package
- **Precision**: vulnerabilities are too old or deprecated



Let’s look at the vulnerabilities

Why do we see different CVEs?

Notes

NOT-FOR-US: Apple

Notes

Crash in CLI tool, no security impact
https://www.zerodayinitiative.com/advisories/ZDI-24-1606/
https://bushido-sec.com/index.php/2024/11/22/2ourc3-vulnerabiltiy-7zip-fuzzing/
Since p7zip/16.02+transitional.1 src:p7zip is only a empty source package
depending on 7zip. Mark this version as fixed version.

Notes

[buster] - shadow <no-dsa> (Minor issue)
https://github.com/shadow-maint/shadow/pull/687
Fixed by: https://github.com/shadow-maint/shadow/commit/e5905c4b84d4fb90aefcd96ee618411ebfac663d (4.14.0-rc1)
Regression fix: https://github.com/shadow-maint/shadow/commit/2eaea70111f65b16d55998386e4ceb4273c19eb4 (4.14.0-rc1)
https://www.trustwave.com/en-us/resources/security-resources/security-advisories/?fid=31797
https://www.trustwave.com/en-us/resources/blogs/spiderlabs-blog/cve-2023-29383-abusing-linux-chfn-to-misrepresent-etc-passwd/

Experiment

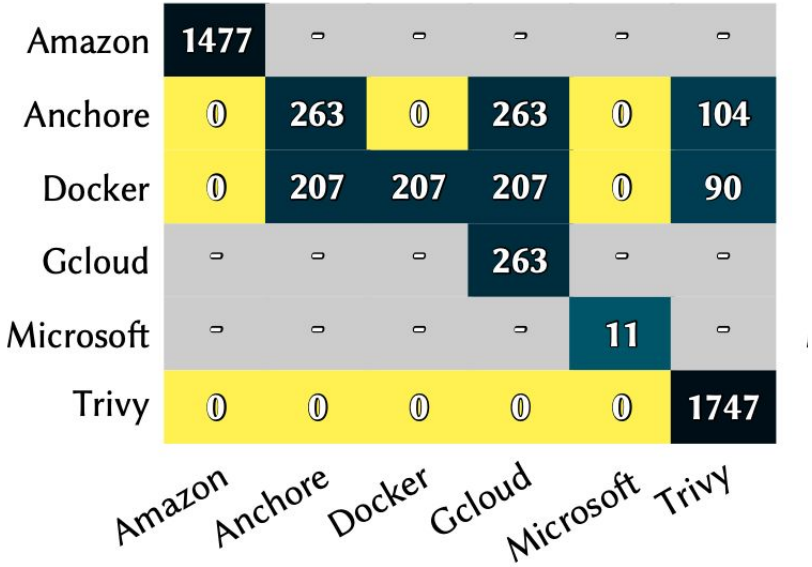
Simulate a potential scenario

A vendor gives us a SBOM, what do we do with it?

We scan it for CVEs (CI)

Experiment

Simulate a potential scenario



Experiment

Can we do better?

Amazon	1477	-	-	-	-	-
Anchore	1765	263	1765	263	104	263
Docker	207	207	207	207	90	207
Gcloud	-	-	-	263	-	-

It is possible to “translate SBOMs” but tools are not compliant with the standard.

Summary

- Why **different packages** are reported for the same container?
  - **Different ways of looking at the container filesystem**
  - **Duplicated and/or unrelated packages**
- What is the impact on **vulnerability detection** accuracy?
  - **Tools are not interoperable**
  - **Can cause missing important vulnerabilities**
- Is there a way to **automatically mitigate** inconsistent SBOM outputs?
  - **We can map some packages automatically**
  - **SBOM generators should agree on the pURL standard**

Making this research visible

We open source!

- Tool to translate SBOMs of different “dialects”
- UI to show differences between SBOMs and CVEs

KubeCon CloudNativeCon Europe 2026

Live demo

Takeaways

- More vulnerabilities detected != better tool
- Fragmented SBOM ecosystem

Short term

- Inform your customers on the tools to use

Long term

- SBOM vendors come together and define common interfaces

We will keep an eye on the compatibility

KubeCon CloudNativeCon Europe 2026

Thank you!

Jacopo Bufalino
jacopo.bufalino@lecnam.net

Agathe Blaise
agathe.blaise@thalesgroup.com