bell//soft

FROM KITCHEN TO TABLE

A Safe Software Journey with SBOMs

Dmitry Chuyko

bell-sw.com | 2026

bellsoft

WHO WE ARE

Dmitry Chuyko
https://openjdk.org/census#dchuyko

Who we are

bellsoft

BellSoft was founded in 2017 by Java and Linux experts with 15+ years of experience working in Sun/Oracle.
Headquarters in San Jose, California.

Members of:

- JCP Executive Committee
- OpenJDK Vulnerability Group
- GraalVM Advisory Board
- Linux Foundation
- Cloud Native Computing Foundation

Our products

bellsoft

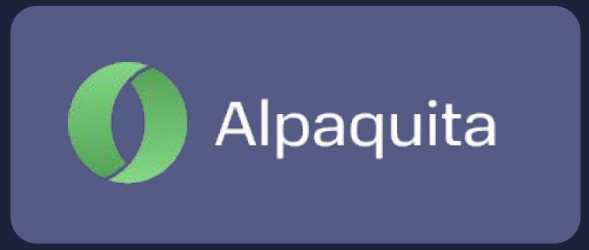
Java

Liberica JDK is a 100% open-source vanilla Java 8, 11, 17, 21 and 25 implementation.

Liberica Native Image Kit JDK is a GraalVM based tool for creating performant native images.

Release schedule for all the products conforms to the LTS roadmap.
All products are available for a large number of platforms.

Linux



Alpaquita Linux is 100% Alpine compatible, secure, and optimized for Java.

BellSoft Hardened Images

SCORE YOURSELF HONESTLY

TAKE THE TEST: HOW SECURE IS YOUR SPRING BOOT CONTAINER?

#	QUESTION	YES	NO
1	Does your container run as a non-root user with privilege escalation blocked at runtime?	☐	☐
2	Have you dropped all Linux capabilities your application doesn't explicitly need?	☐	☐
3	Does your production image use a JRE rather than a full JDK?	☐	☐
4	Is your base OS under 10 MB with no package manager in the runtime layer?	☐	☐
5	Have you scanned your images for CVEs in the last 30 days?	☐	☐
6	Do you go beyond CVSS scores, checking whether vulnerable code is actually reachable in your workload?	☐	☐
7	Do you have a Software Bill of Materials (SBOM) for your container image?	☐	☐
8	Is your base image from a vendor committed to patching CVEs on a defined SLA?	☐	☐
9	Are your base images pinned by digest rather than tag?	☐	☐
10	Can you verify your base image came from the expected publisher with signed attestation?	☐	☐

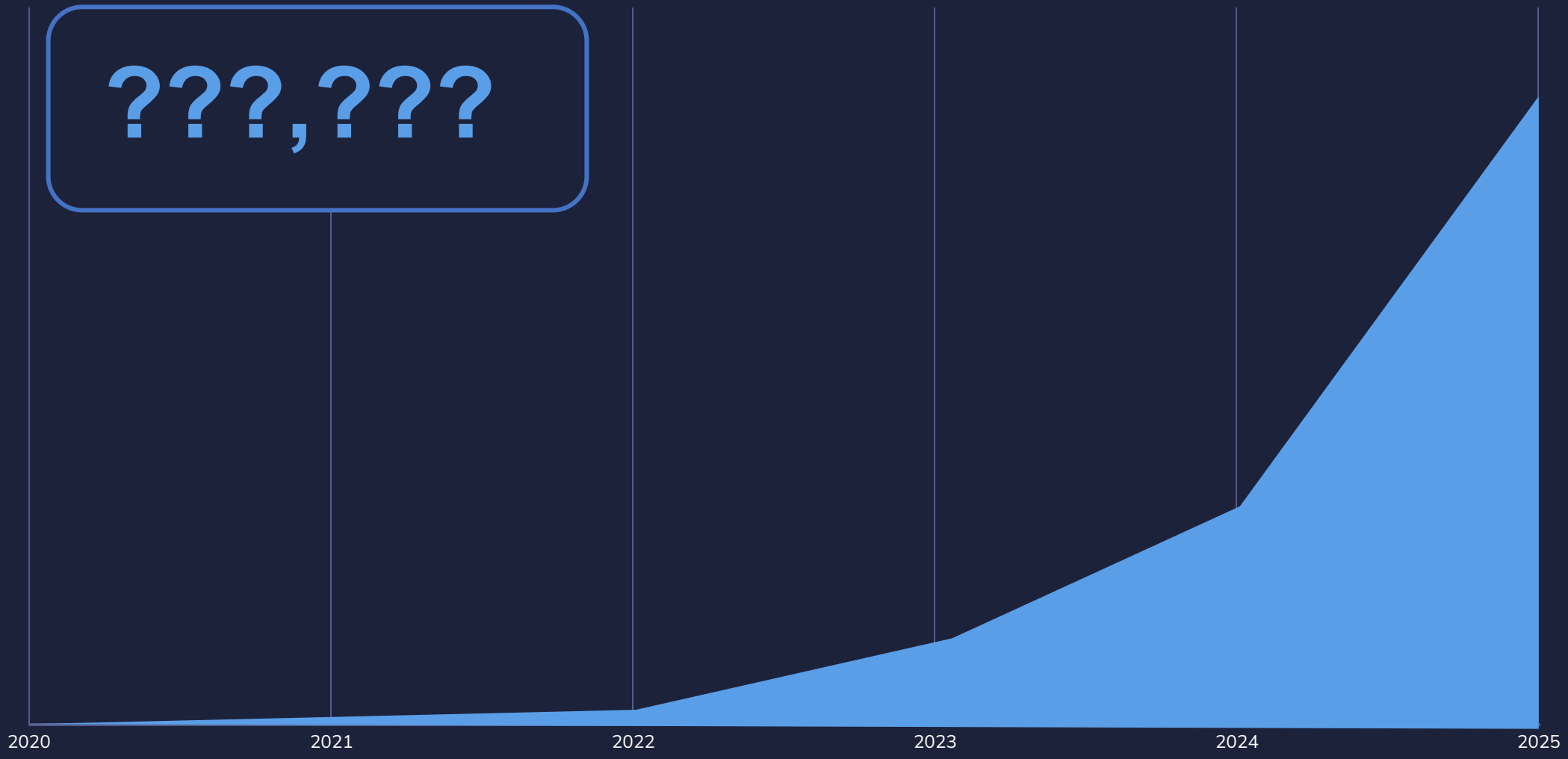
8-10 Yes
Good secure posture.
Use hardened images to maintain it without overhead.

5-7 Yes
At risk, gaps exist.
You have exploitable gaps that attackers actively look for.

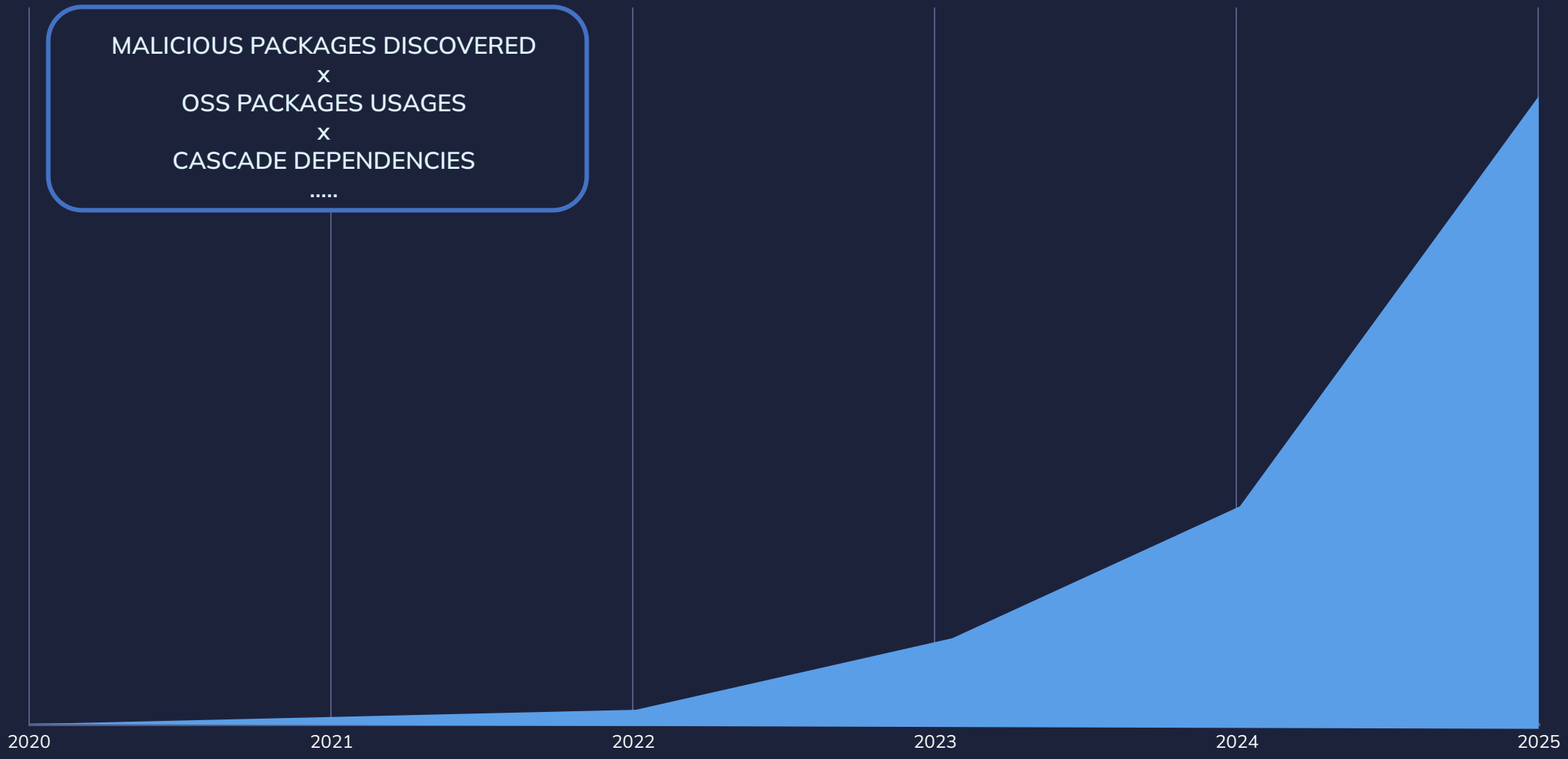
4 or fewer Yes
Exposed.
Your containers are a significant production risk. Action required.

THE RISKS HIDING IN EVERY "NO" ANSWER

Silent compromise	CVEs buried in your dependencies are invisible until exploited. One vulnerability in a transitive package gives an attacker a foothold your scanner may never flag.
Full system takeover	A container running as root isn't just a misconfiguration. Combined with a kernel vulnerability or Docker misconfiguration, it may hands an attacker root on the host, compromising every container on it.
	Without an SBOM and image signing, you cannot verify what you're shipping.

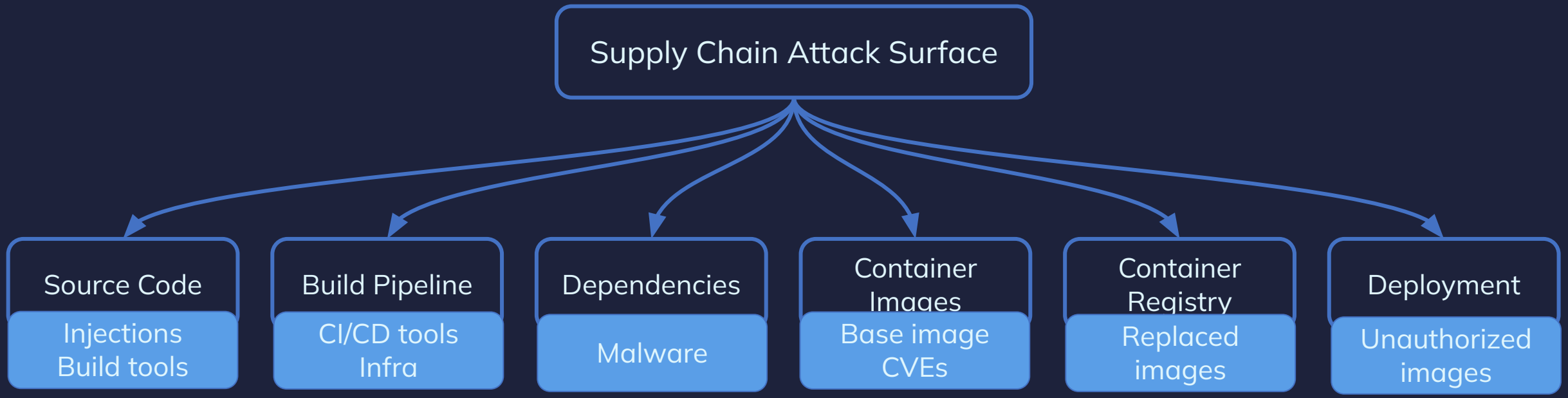

???,???
2020
2021
2022
2023
2024
2025

Risks


MALICIOUS PACKAGES DISCOVERED
x
OSS PACKAGES USAGES
x
CASCADE DEPENDENCIES
.....
2020
2021
2022
2023
2024
2025

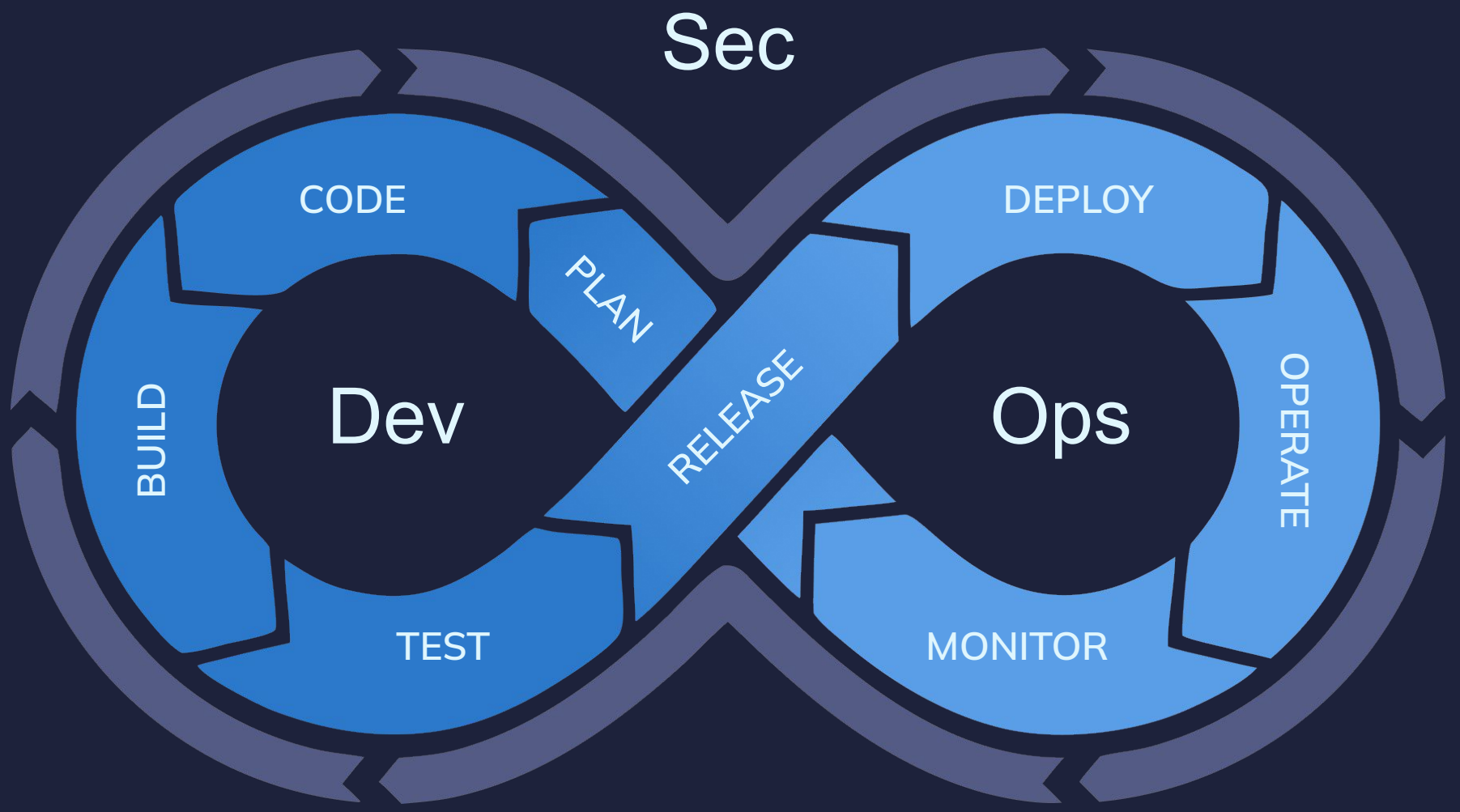
From kitchen to table

Daily nutrition

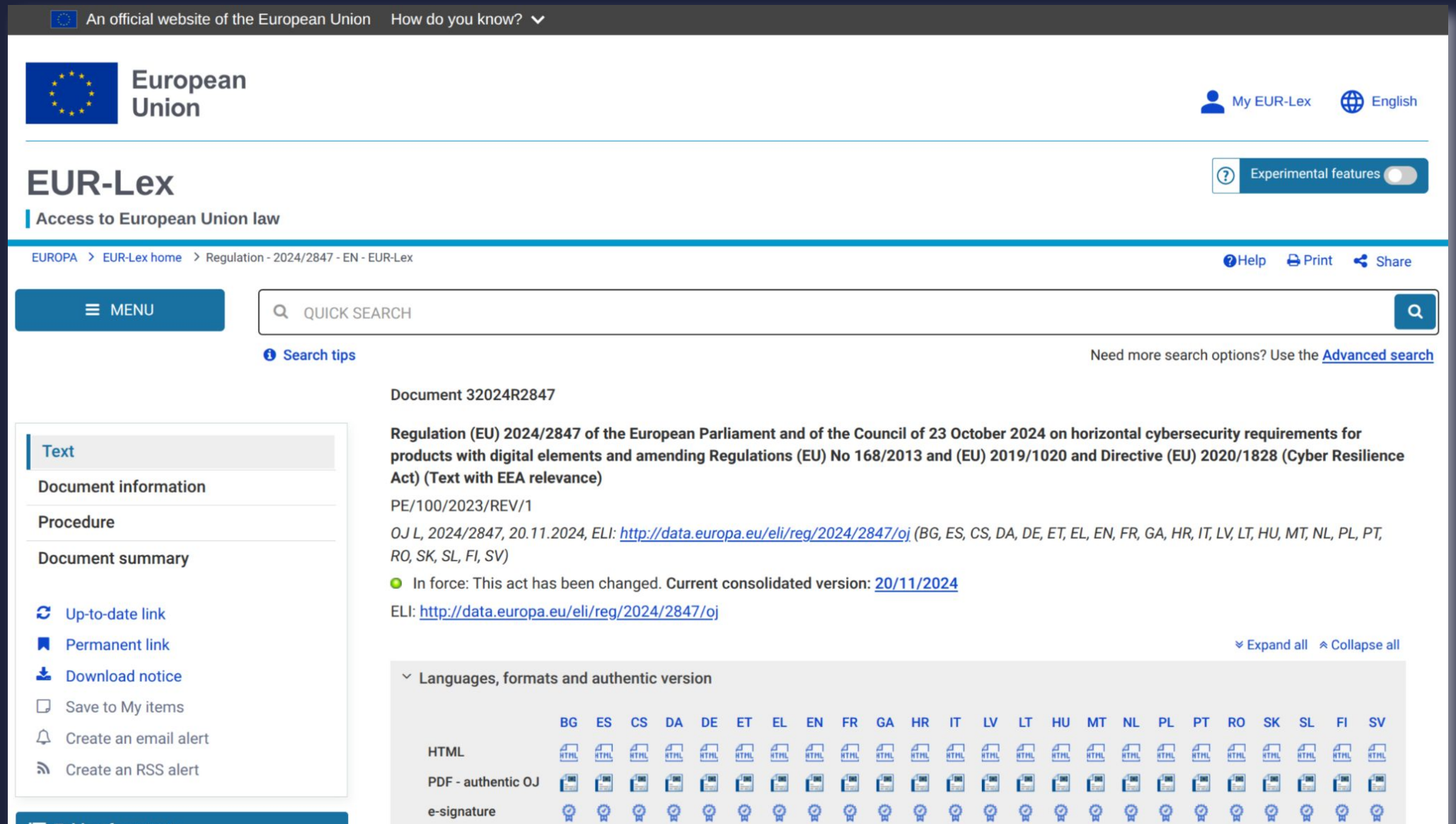


Uroboros to the rescue

Don’t forget to shift left...



CRA (Cyber Resilience Act)



An official website of the European Union How do you know?

European Union

My EUR-Lex English

EUR-Lex

Access to European Union law

Experimental features

EUROPA > EUR-Lex home > Regulation - 2024/2847 - EN - EUR-Lex

Help Print Share

MENU

QUICK SEARCH

Search tips

Need more search options? Use the Advanced search

Text

Document information

Procedure

Document summary

Up-to-date link

Permanent link

Download notice

Save to My items

Create an email alert

Create an RSS alert

Document 32024R2847

Regulation (EU) 2024/2847 of the European Parliament and of the Council of 23 October 2024 on horizontal cybersecurity requirements for products with digital elements and amending Regulations (EU) No 168/2013 and (EU) 2019/1020 and Directive (EU) 2020/1828 (Cyber Resilience Act) (Text with EEA relevance)

PE/100/2023/REV/1

OJ L, 2024/2847, 20.11.2024, ELI: http://data.europa.eu/eli/reg/2024/2847/oj (BG, ES, CS, DA, DE, ET, EL, EN, FR, GA, HR, IT, LV, LT, HU, MT, NL, PL, PT, RO, SK, SL, FI, SV)

In force: This act has been changed. **Current consolidated version: 20/11/2024**

ELI: http://data.europa.eu/eli/reg/2024/2847/oj

Expand all Collapse all

Languages, formats and authentic version

	BG	ES	CS	DA	DE	ET	EL	EN	FR	GA	HR	IT	LV	LT	HU	MT	NL	PL	PT	RO	SK	SL	FI	SV
HTML																								
PDF - authentic OJ																								
e-signature																								

Capability Maturity Model Integration

Vulnerability management at different levels

Level	Features
1. Initial	Prepare.
2. Managed	Ad-hoc scanning, criticals tracking.
3. Defined	Regular scans, inventory, reporting, red teams, triage, patching.
4. Quantitatively Managed	Task tracker automation, Security Operation Center (SOC), Known Exploited Vulnerabilities (KEV), mandatory scanning.
5. Optimizing	KPI.

Software Bill of Materials (SBOM)

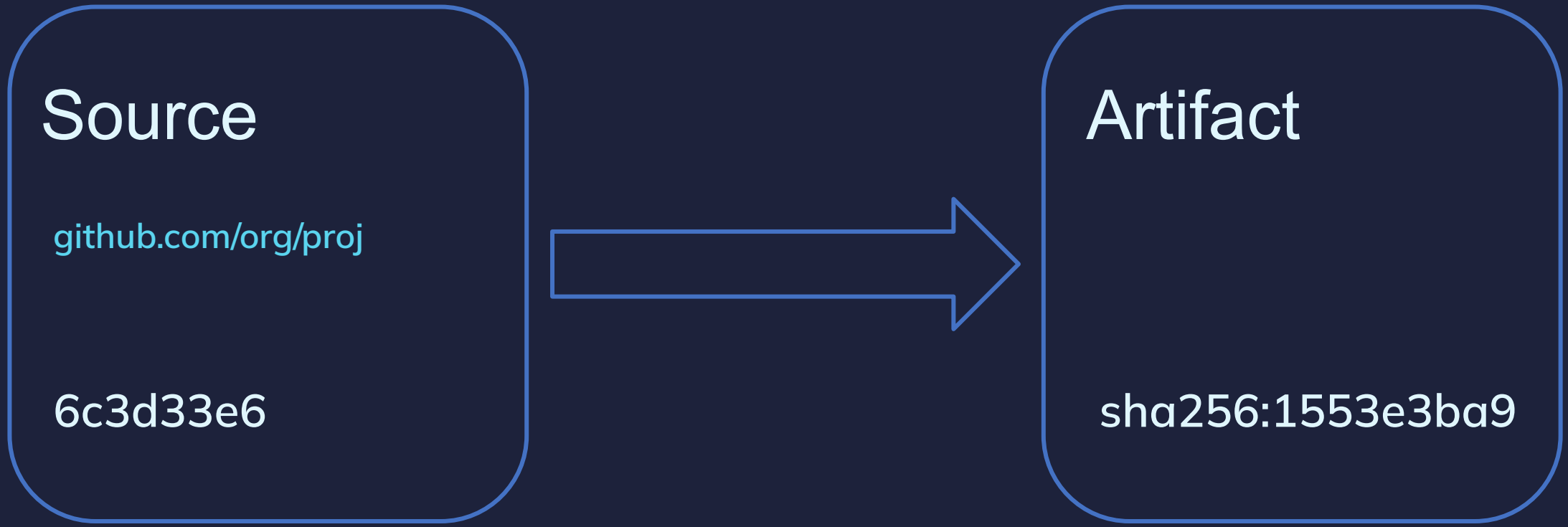
Track ingredients

SBOM types
- Design, Source, ***Build***, Analysed, Deployed, Runtime

Standard formats
- CycloneDX (CDX), SDPX
- Others

Signatures

Certify a recipe

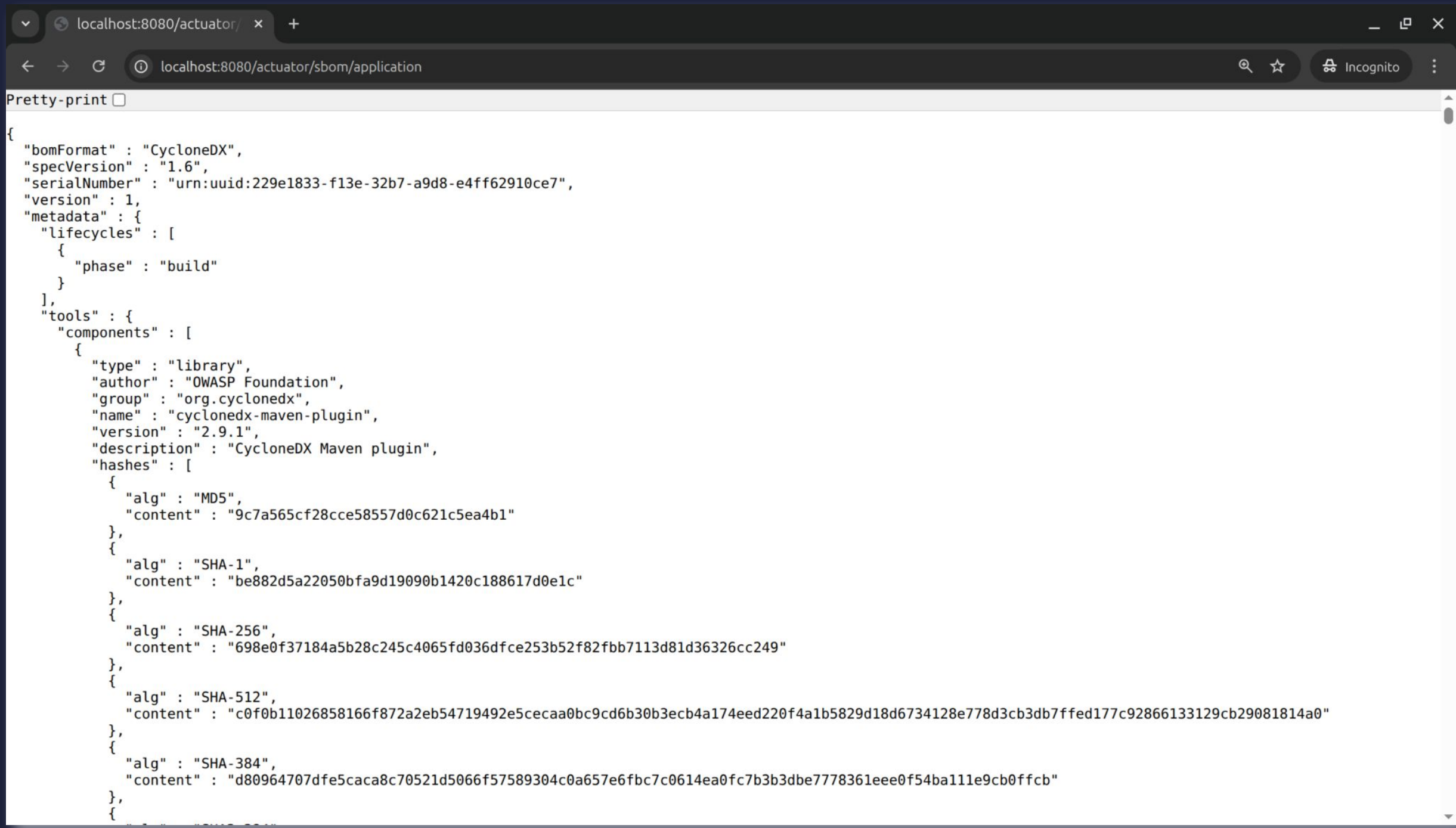


Verification after trusted signing by the signature owner

Meet SBOM

And some dependency downloads

```
$ mvn install
...
Downloaded from central:
https://repo1.maven.org/maven2/org/apache/logging/log4j/log4j-api/2.25.3/log4j-api-2.25.3.jar (351 kB at 2.0
MB/s)
...
[INFO] Installing <...>/target/spring-petclinic-4.0.0-SNAPSHOT.jar to
/home/tp/.m2/repository/org/springframework/samples/spring-petclinic/4.0.0-SNAPSHOT/spring-petclinic-4.0.0-S
NAPSHOT.jar
[INFO] Installing <...>/target/classes/META-INF/sbom/application.cdx.json to
/home/tp/.m2/repository/org/springframework/samples/spring-petclinic/4.0.0-SNAPSHOT/spring-petclinic-4.0.0-S
NAPSHOT-cyclonedx.json
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
```




```
{
  "bomFormat" : "CycloneDX",
  "specVersion" : "1.6",
  "serialNumber" : "urn:uuid:229e1833-f13e-32b7-a9d8-e4ff62910ce7",
  "version" : 1,
  "metadata" : {
    "lifecycles" : [
      {
        "phase" : "build"
      }
    ],
    "tools" : {
      "components" : [
        {
          "type" : "library",
          "author" : "OWASP Foundation",
          "group" : "org.cyclonedx",
          "name" : "cyclonedx-maven-plugin",
          "version" : "2.9.1",
          "description" : "CycloneDX Maven plugin",
          "hashes" : [
            {
              "alg" : "MD5",
              "content" : "9c7a565cf28cce58557d0c621c5ea4b1"
            },
            {
              "alg" : "SHA-1",
              "content" : "be882d5a22050bfa9d19090b1420c188617d0e1c"
            },
            {
              "alg" : "SHA-256",
              "content" : "698e0f37184a5b28c245c4065fd036dfce253b52f82fbb7113d81d36326cc249"
            },
            {
              "alg" : "SHA-512",
              "content" : "c0f0b11026858166f872a2eb54719492e5cecaa0bc9cd6b30b3ecb4a174eed220f4a1b5829d18d6734128e778d3cb3db7ffed177c92866133129cb29081814a0"
            },
            {
              "alg" : "SHA-384",
              "content" : "d80964707dfe5caca8c70521d5066f57589304c0a657e6fbc7c0614ea0fc7b3b3dbe7778361eee0f54ba111e9cb0ffcb"
            },
            {
```

Meet SBOM

Did we verify the libraries?

```
$ mvn org.simplify4u.plugins:pgpverify-maven-plugin:show \
       -Dartifact=org.apache.logging.log4j:log4j-api:2.25.2

Downloaded from central:
https://repo1.maven.org/maven2/org/apache/logging/log4j/log4j-api/2.25.2/log4j-api-2.25.2.jar.asc (856 B at
10 kB/s)
...
Artifact:
    groupId:     org.apache.logging.log4j
    artifactId:  log4j-api
    type:        jar
    version:     2.25.2

PGP signature:
    version:     4
    algorithm:   SHA512 with RSA (Encrypt or Sign)
    keyId:       0x077E8893A6DCC33DD4A4D5B256E73BA9A0B592D0
    create date: Thu Sep 18 23:35:37 GMT+05:00 2025
    status:      valid
```

Meet SBOM

The signing keys we trust

...

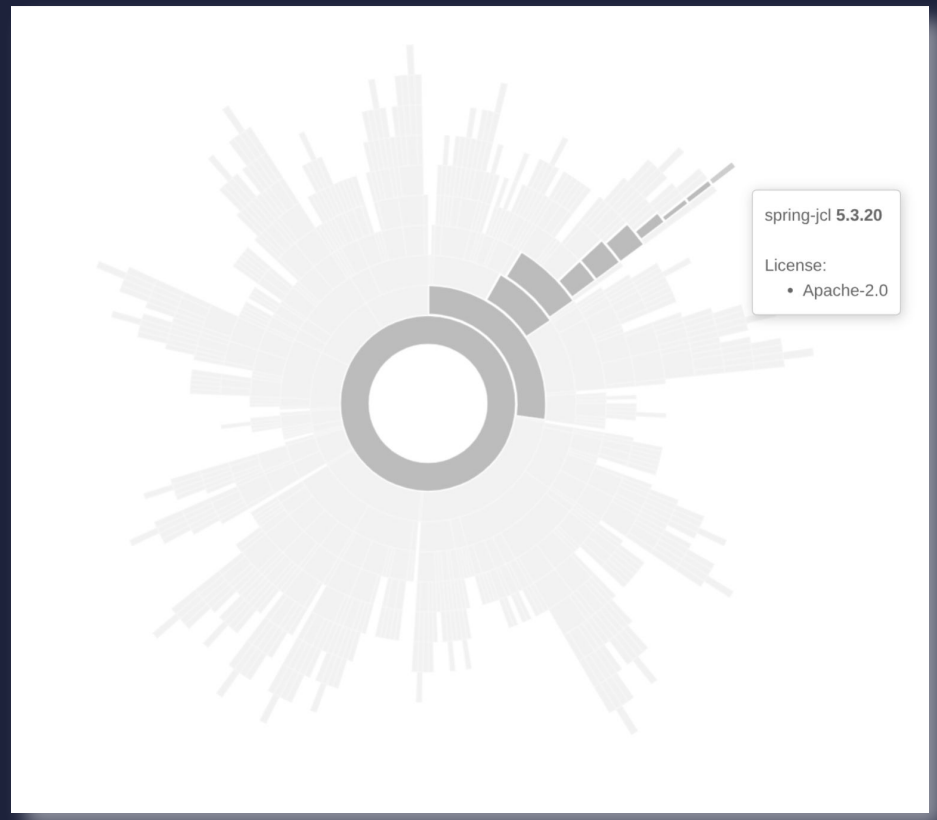
```
PGP key:
    version:     4
    algorithm:   RSA (Encrypt or Sign)
    bits:        4096
    fingerprint: 0x077E8893A6DCC33DD4A4D5B256E73BA9A0B592D0
    create date: Tue Jan 10 14:25:28 GMT+05:00 2023
    uids:        [ASF Logging Services RM <private@logging.apache.org>]
```

Meet SBOM

Different view

```
$ syft target/app.jar --output cyclonedx-json=app.cdx.json
```

https://github.com/CycloneDX/Sunshine/



Meet SBOM

Scan it finally

https://github.com/google/osv-scanner

```
$ osv-scanner --sbom=petclinic2.cdx.json # enabled cyclonedx-maven-plugin manually
Total 20 packages affected by 77 known vulnerabilities (8 Critical, 36 High, 26 Medium, 7 Low, 0 Unknown)
from 1 ecosystem.
76 vulnerabilities can be fixed.

$ osv-scanner --sbom=petclinic2-syft.cdx.json
Total 18 packages affected by 75 known vulnerabilities (8 Critical, 34 High, 26 Medium, 7 Low, 0 Unknown)
from 1 ecosystem.
75 vulnerabilities can be fixed.
```

Meet SBOM

Scan it finally

https://github.com/google/osv-scanner

```
$ osv-scanner --sbom=petclinic2.cdx.json
Total 20 packages affected by 77 known vulnerabilities (8 Critical, 36 High, 26 Medium, 7 Low, 0 Unknown)
from 1 ecosystem.
76 vulnerabilities can be fixed.

$ osv-scanner --sbom=petclinic3.cdx.json
Total 9 packages affected by 42 known vulnerabilities (5 Critical, 15 High, 14 Medium, 8 Low, 0 Unknown)
from 1 ecosystem.
42 vulnerabilities can be fixed

$ osv-scanner --sbom=petclinic4.cdx.json
Total 5 packages affected by 17 known vulnerabilities (1 Critical, 9 High, 5 Medium, 2 Low, 0 Unknown) from
1 ecosystem.
17 vulnerabilities can be fixed.
```

Verify and scan base image

Why not “FROM openjdk”?

```
$ docker trust inspect eclipse-temurin:25-jre
[
    {
        "Name": "eclipse-temurin:25-jre",
        "SignedTags": [], # 24_36 is the latest signed one now
        "Signers": [],
        "AdministrativeKeys": [
            {
                "Name": "Root",
                "Keys": [
                    {
                        "ID": "3091cac26efdcfd46978d99f4cdafb8687e8ddf55526c17491efec808113f404"
                    }
                ]
            },
            {
                "Name": "Repository",
                "Keys": [
                    {
```

Verify and scan base image

```
$ docker trust inspect bellsoft/hardened-liberica-runtime-container:jre-25-musl
[]
no signatures or cannot access bellsoft/hardened-liberica-runtime-container:jre-25-musl
```

Verify and scan base image

https://github.com/sigstore/cosign
https://docs.bell-sw.com/alpaquita-linux/latest/how-to/cosign-guide/
https://download.bell-sw.com/pki/cosign-bellsoft.pub

```
$ cosign verify --key cosign-bellsoft.pub
index.docker.io/bellsoft/hardened-liberica-runtime-container:jre-25-musl@sha256:76fe23eb93cbfaff384090e6458b
2a7876ba31e5ec69fa5367c720f24f6a0142

Verification for
index.docker.io/bellsoft/hardened-liberica-runtime-container@sha256:76fe23eb93cbfaff384090e6458b2a7876ba31e5
ec69fa5367c720f24f6a0142 --
The following checks were performed on each of these signatures:
  - The cosign claims were validated
  - Existence of the claims in the transparency log was verified offline
  - The signatures were verified against the specified public key

[{"critical":{"identity":{"docker-reference":"index.docker.io/bellsoft/hardened-liberica-runtime-container@s
ha256:76fe23eb93cbfaff384090e6458b2a7876ba31e5ec69fa5367c720f24f6a0142"},"image":{"docker-manifest-digest":"
sha256:76fe23eb93cbfaff384090e6458b2a7876ba31e5ec69fa5367c720f24f6a0142"},"type":"https://spdx.dev/Document"
},"optional":null},{"critical":{"identity":{"docker-reference":"index.docker.io/bellsoft/hardened-liberica-r
untime-container@sha256:76fe23eb93cbfaff384090e6458b2a7876ba31e5ec69fa5367c720f24f6a0142"},"image":{"docker-
```

Get SBOMs included

Side dish or the main course?

```
$ IMG='docker.io/bellsoft/hardened-base:musl' # here and after assume digests
$ cosign verify-attestation \
    --key cosign-bellsoft.pub \
    --type cyclonedx \
    $IMG | jq -r '.payload' | base64 -d | jq '.predicate'
```

Verify and scan base image

Scan it finally

```
$ osv-scanner scan image bellsoft/hardened-liberica-runtime-container:jre-25-musl
...
Starting filesystem walk for root:
End status: 132 dirs visited, 751 inodes visited, 116 Extract calls, 13.850781ms elapsed, 13.8509ms wall time

No issues found
```

Verify and scan base image

Different view

```
$ osv-scanner scan image alpine:latest --serve
...
Starting filesystem walk for root:
End status: 98 dirs visited, 516 inodes visited, 47 Extract calls, 5.694407ms elapsed, 5.694604ms wall time
Starting filesystem walk for root:
End status: 0 dirs visited, 1 inodes visited, 1 Extract calls, 425.112µs elapsed, 425.388µs wall time
Serving HTML report at http://localhost:8000
```

OSV | Open Source Vulnerabilities

Feedback

Scanning summary

Layer

All layers (12)

Filters

Default (12/12)

Overall Severity

0 Critical | 5 High | 7 Medium | 0 Low | 0 Unknown

OS version: **Alpine Linux v3.16**

▶ Base image 2: **alpine** (12/12 vulnerabilities)

▶ Base image 1: **alpine** (0/12 vulnerabilities)

Your image: (0/12 vulnerabilities)

Search vulnerability ID...

Alpine:v3.16

Source: os:/lib/apk/db/installed

	Source Package	Installed version	Fix available	Vulnerability count
▶	busybox	1.35.0-r17	Fix available	0 0 1 0 0
▶	musl	1.2.3-r0	Fix available	0 1 0 0 0
▶	openssl	1.1.1q-r0	Fix available	0 4 6 0 0

BellSoft Hardened Image

bell-sw.com/bellsoft-hardened-images/

Incognito

bellsoft Search

Downloads Products Resources Support About us Contact us

BellSoft

bellsoft/hardened-liberica-runtime-container:jre-25-musl

Total	Critical	High	Medium	Low	Unknown
0 vs 0 0%	0 vs 0 0%	0 vs 0 0%	0 vs 0 0%	0 vs 0 0%	0 vs 0 0%

Alternative

eclipse-temurin:25

Total	Critical	High	Medium	Low	Unknown
0 vs 0 0%	0 vs 0 0%	0 vs 0 0%	0 vs 0 0%	0 vs 0 0%	0 vs 0 0%

Sign and verify application image

Dockerfile (simplified)

```
# bellsoft/hardened-liberica-runtime-container:jre-25-musl
FROM
bellsoft/hardened-liberica-runtime-container@sha256:76fe23eb93cbfaff384090e6458b2a7876ba31e5ec69fa5367c720f2
4f6a0142

WORKDIR /app
ADD target/spring-petclinic-4.0.0-SNAPSHOT.jar /app/app.jar
EXPOSE 8080
ENTRYPOINT ["java", "-jar", "app.jar"]
```

Sign and verify application image

```
$ cosign sign --key my.key
repo:5000/petclinic4@sha256:1553e3ba9cc6b4278e0f57f053b664ef2c712c42df62e782ba129200992ed17b

$ cosign verify --key my.pub
repo:5000/petclinic4@sha256:1553e3ba9cc6b4278e0f57f053b664ef2c712c42df62e782ba129200992ed17b

Verification for
repo:5000/petclinic4@sha256:1553e3ba9cc6b4278e0f57f053b664ef2c712c42df62e782ba129200992ed17b --
The following checks were performed on each of these signatures:
  - The cosign claims were validated
  - Existence of the claims in the transparency log was verified offline
  - The signatures were verified against the specified public key

[{"critical":{"identity":{"docker-reference":"repo:5000/petclinic4@sha256:1553e3ba9cc6b4278e0f57f053b664ef2c
712c42df62e782ba129200992ed17b"},"image":{"docker-manifest-digest":"sha256:1553e3ba9cc6b4278e0f57f053b664ef2
c712c42df62e782ba129200992ed17b"},"type":"https://sigstore.dev/cosign/sign/v1"},"optional":null},{"critical"
:{"identity":{"docker-reference":"repo:5000/petclinic4@sha256:1553e3ba9cc6b4278e0f57f053b664ef2c712c42df62e7
82ba129200992ed17b"},"image":{"docker-manifest-digest":"sha256:1553e3ba9cc6b4278e0f57f053b664ef2c712c42df62e
782ba129200992ed17b"},"type":"https://sigstore.dev/cosign/sign/v1"},"optional":null}]
```

Sign and verify application image

Buildpack

```
$ mvn spring-boot:build-image
```

pom.xml

```
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-maven-plugin</artifactId>
        <configuration>
          <image>
            <builder>bellsoft/buildpacks.builder:musl</builder>
          </image>
        </configuration>
```

Sign and verify application image

Scan buildpack image

```
$ osv-scanner scan image spring-petclinic:4.0.0-bp-hardened
...
Starting filesystem walk for root:
End status: 239 dirs visited, 601 inodes visited, 25 Extract calls, 1.001505ms elapsed, 1.001561ms wall time
Starting filesystem walk for root:
End status: 0 dirs visited, 1 inodes visited, 1 Extract calls, 74.144µs elapsed, 74.212µs wall time
Starting filesystem walk for root:
End status: 0 dirs visited, 1 inodes visited, 1 Extract calls, 59.538µs elapsed, 59.614µs wall time

No issues found
```

Sign and verify application image

Scan buildpack image

```
$ osv-scanner scan image spring-petclinic:4.0.0-bp-default
...
End status: 247 dirs visited, 1454 inodes visited, 594 Extract calls, 16.650067ms elapsed, 16.650121ms wall time
...
```

Container Scanning Result (Paketo Buildpacks Tiny Noble):
Total 4 packages affected by 8 known vulnerabilities (0 Critical, 4 High, 4 Medium, 0 Low, 0 Unknown) from 1 ecosystem.
0 vulnerabilities can be fixed.

Ubuntu:24.04

Source:os:/var/lib/dpkg/status.d/libc6

SOURCE PACKAGE	INSTALLED VERSION	FIX AVAILABLE	VULN COUNT	BINARY PACKAGES (COUNT)	INTRODUCED LAYER	IN BASE IMAGE
glibc	2.39-0ubuntu8.7	No fix available	3	libc6	# 0 Layer	--

Source:os:/var/lib/dpkg/status.d/libssl3t64

SOURCE PACKAGE	INSTALLED VERSION	FIX AVAILABLE	VULN COUNT	BINARY PACKAGES (COUNT)	INTRODUCED LAYER	IN BASE IMAGE
openssl	3.0.13-0ubuntu3.9	No fix available	2	libssl3t64	# 0 Layer	--

Source:os:/var/lib/dpkg/status.d/openssl

SOURCE PACKAGE	INSTALLED VERSION	FIX AVAILABLE	VULN COUNT	BINARY PACKAGES (COUNT)	INTRODUCED LAYER	IN BASE IMAGE
openssl	3.0.13-0ubuntu3.9	No fix available	2	openssl	# 0 Layer	--

Source:os:/var/lib/dpkg/status.d/zlib1g

SOURCE PACKAGE	INSTALLED VERSION	FIX AVAILABLE	VULN COUNT	BINARY PACKAGES (COUNT)	INTRODUCED LAYER	IN BASE IMAGE
zlib	1:1.3.dfsg-3.1ubuntu2.1	No fix available	1	zlib1g	# 0 Layer	--

Hiding 1 number of vulnerabilities deemed unimportant, use --all-vulns to show them.
For the most comprehensive scan results, we recommend using the HTML output: `osv-scanner scan image --serve <image_name>`.
You can also view the full vulnerability list in your terminal with: `osv-scanner scan image --format vertical <image_name>`.

Image SBOMs and scanning in CI/CD

- Verify incoming artifacts
- Build image
- Generate or reuse SBOMs
- Supply SBOMs, sign image
- Scan image/SBOMs/projects
  - CI integration plugins / actions
  - Vulnerability reports
- [Fail pipeline on new CVEs above threshold]
- Upload SBOM
  - Bind to git sha
  - Bind to image/timestamp
  - Upload to SCA system (Software Composition Analysis)
  - Dependency-Track
  - CI integration plugins

SBOMs and policies in CI/CD

- Control licenses and component set
- jq helps to automate JSON processing
- Python AST libraries may break
- Advanced predicates with buildx and Rego
  - step towards PaC (Policy as a Code)

SBOMs and policies in K8s

- Regular image scans and SBOM generation for running Pods
  - E.g. Cron-based
  - CVE and License reports
- Kyverno cluster policies
  - Require signed images for Pods
  - Require SBOMs for images for Pods
- Scan images at deploy time
  - Cluster admission gatekeeper
  - Ratify Verifiers
- Vulnerability alerts
  - Prometheus

Many many SBOMs

Save some time for eating

100+ cves are published daily

SecOps

- Check SBOM
- Analyse the CVE
- Trace dependencies
- Exploitability at every step (different apps, different versions)
- Repeat

Vulnerability Exploitability eXchange (VEX)

Save some time for eating

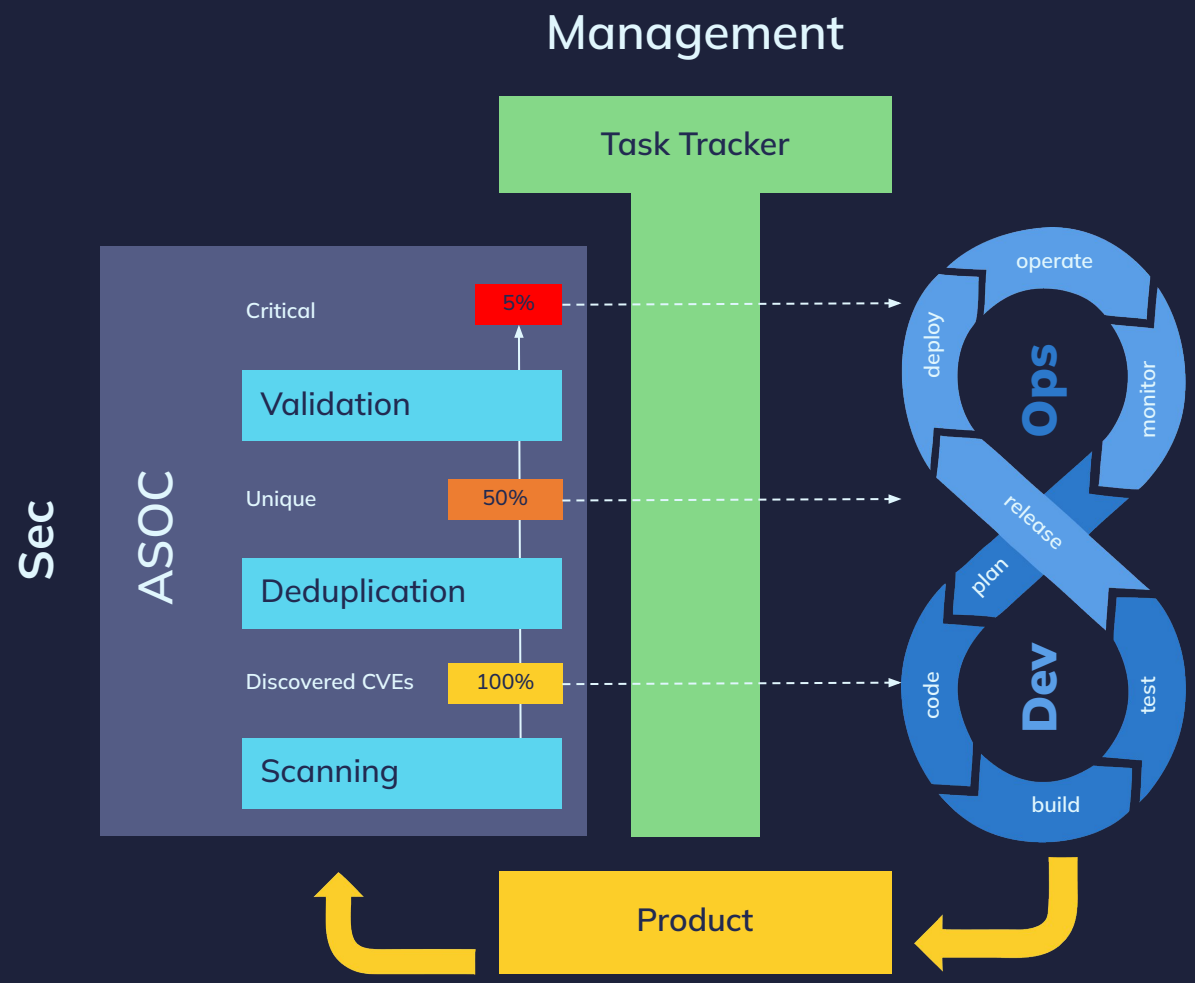
- A→B→C, D→E... (hundreds of thousands at scale)
- C has CVE-2026-XXXX
- Is B affected? Is A affected?
- Are vulnerable parts used by user?

SHOULD A USER REACT?

Somewhere at Level 5

Application Security Orchestration and Correlation

SAST, DAST, SCA



SBOMs & Signatures

- Keep poisoned stuff out
- Available today
- Sometimes mandatory
- Used in CI/CD
- Used in K8s

bell//soft

Thank you for your attention!

bell-sw.com/blog

@dchuyko