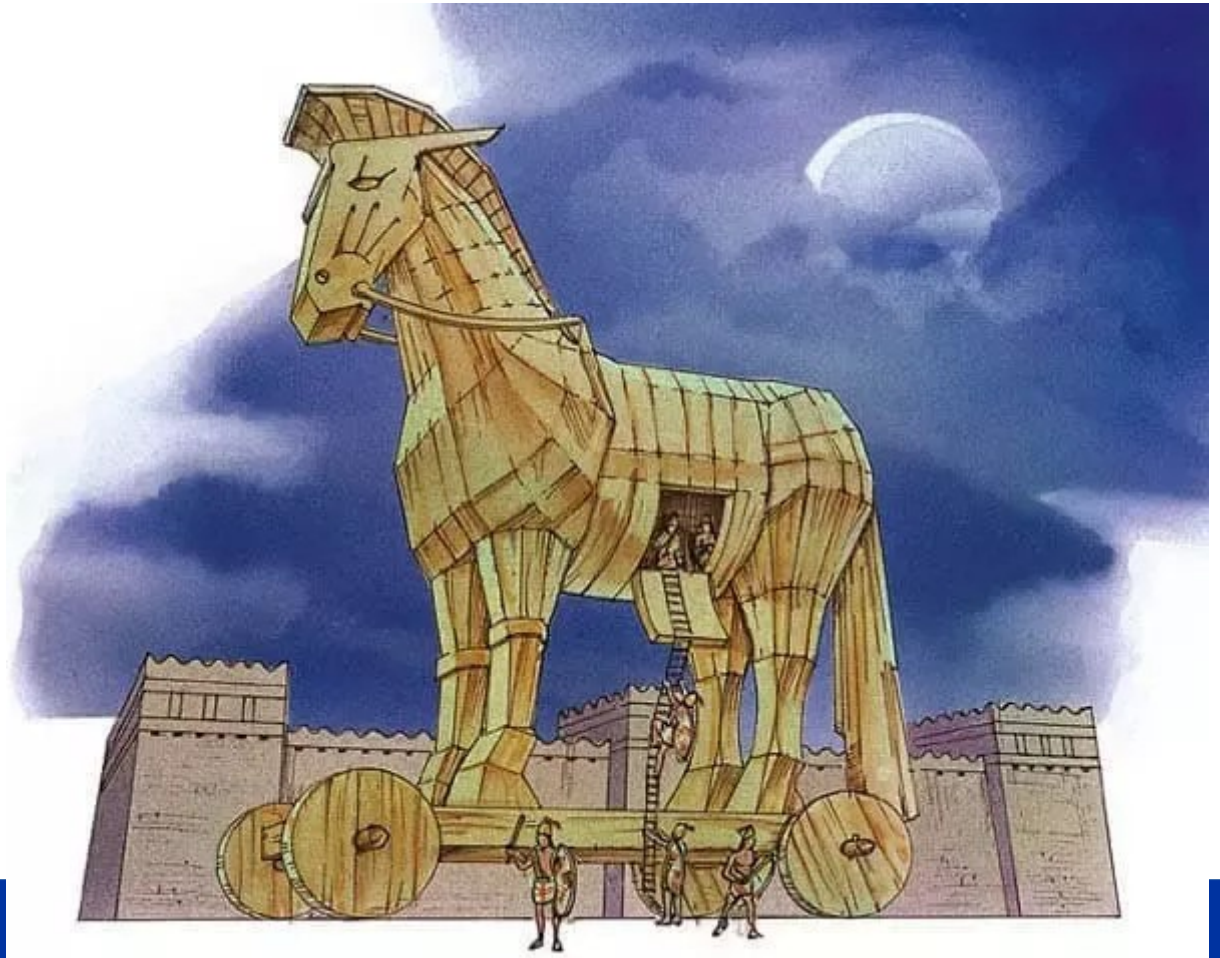
Protecting your controls infrastructure supply chain

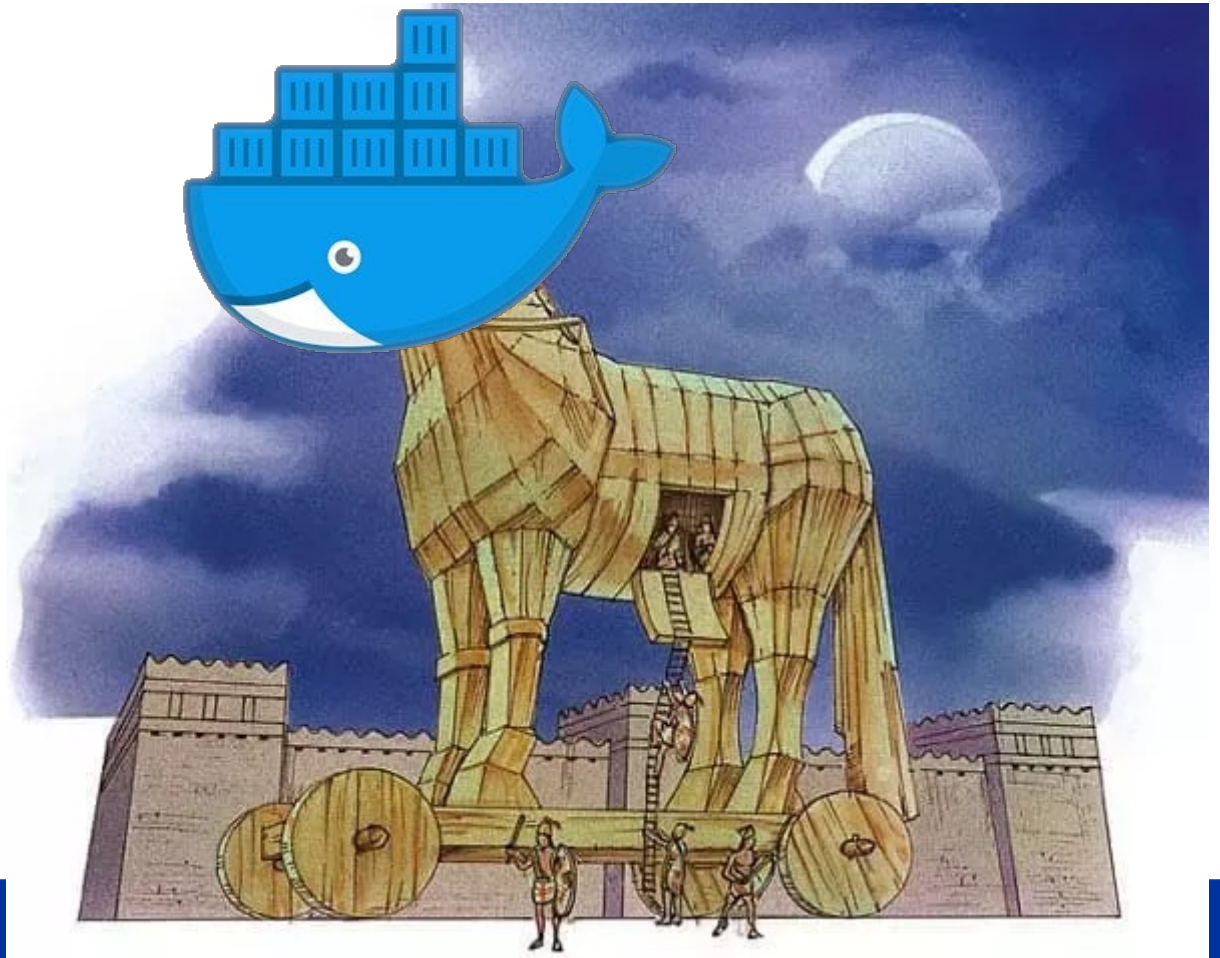
Brice Copy – CERN Beams Department

ICALEPCS October 2023

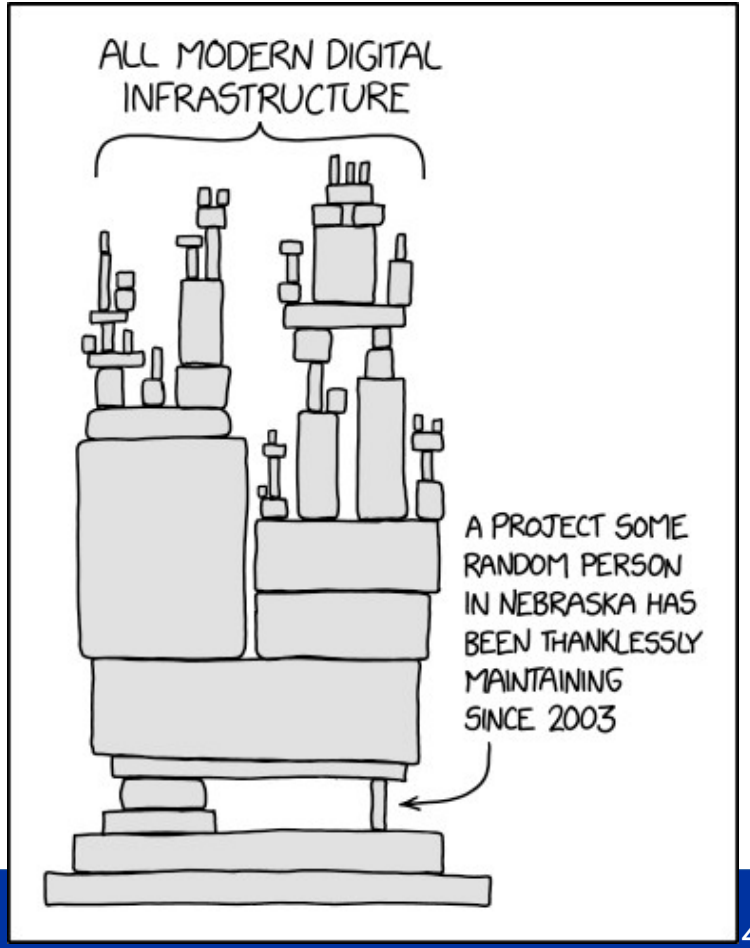
Your infrastructure under attack



Your infrastructure under attack



Modern software architecture



Dependency on open-source in 2022

- **Average software project has 595 dependencies**
  - 200% growth over the past four years
  - 48% of projects carry active documented exploits or remote code execution (RCE) opportunities
- **Open source maintenance on the decline**
  - Percentage of open source projects with dev activity in the past 24 months :
    - 15% in 2018 vs 10% in 2022

Source : 2023 Open source security and risk analysis report, Synopsis inc.

Dependency on open-source in 2022

- **High risk Vulnerabilities not being fixed sufficiently**
  - On average, 42% more high-risk vulnerabilities than in 2018
  - Some industries more severely hit (e.g. E-commerce = +557%)
- **Most cited reason for not fixing vulnerabilities**
  - Lack of visibility, lack of recognition
  - Lack of agility with regards to production deployment

Source : 2023 Open source security and risk analysis report, Synopsis inc.

More transparency across languages

CycloneDX

SPDX

NIST NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY U.S. DEPARTMENT OF COMMERCE

SWID

Software Bill of Material (SBOM)

CycloneDX

```
<bom>
    <metadata>
        (...)
    </metadata>
    <components>
        <component type="library" bom-ref="8a30a40b-1f73dbe7">
            (...)
        </component>
    </components>
    <dependencies>
            (...)
    </dependencies>
</bom>
```

Software Bill of Material (SBOM)

CycloneDX

```
<bom>
  <metadata>
    <timestamp>2023-09-22T12:04:11Z</timestamp>
    <component
          type="library"
          bom-ref="05740efc-a60...b338e">
      <name>name-of-the-library</name>
      <version>6.2.41</version>
    </component>
  </metadata>
```

Software Bill of Material (SBOM)

CycloneDX

```
<bom>
  <components>
    <component type="library" bom-ref="290b...b-5647f300">
      <group>com.netflix.ribbon</group>
      <name>ribbon-core</name>
      <version>2.7.18</version>
      <hashes>
        <hash alg="MD5">ca095fe37….369bb85cef6</hash>
        <hash alg="SHA-256">4557efbb….9bba705</hash>
        (...)
      </hashes>
```

Software Bill of Material (SBOM)

CycloneDX

```
<component ...>
    <licenses>
      <license><id>Apache-2.0</id></license>
    </licenses>
    <purl>pkg:maven/com.netflix.ribbon/ribbon-core@2.7.18?type=jar</purl>
      <externalReferences>
          <reference type="vcs">
            <url>scm:https://github.com/Netflix/ribbon.git</url>
          </reference>
      </externalReferences>
</component>
```

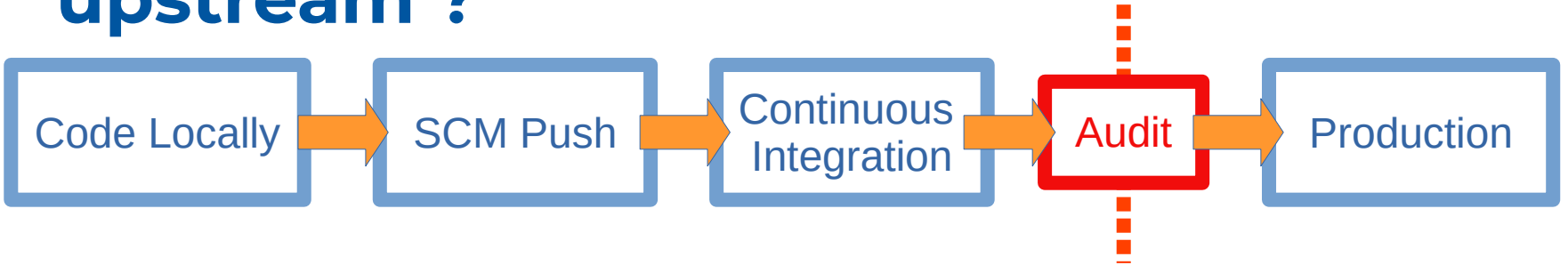
Software Bill of Material (SBOM)

CycloneDX

```
<dependencies>
    <dependency ref="05740efc...-a60ccfeb338e" />
     <dependency ref="8a30a40b-…1f73d63a8be7">
        <dependency ref="17029bb….-1f5553170" />
        (...)
      </dependency>
      <dependency ref="21326a45ad634a9f" />
      (...)
</dependencies>
```

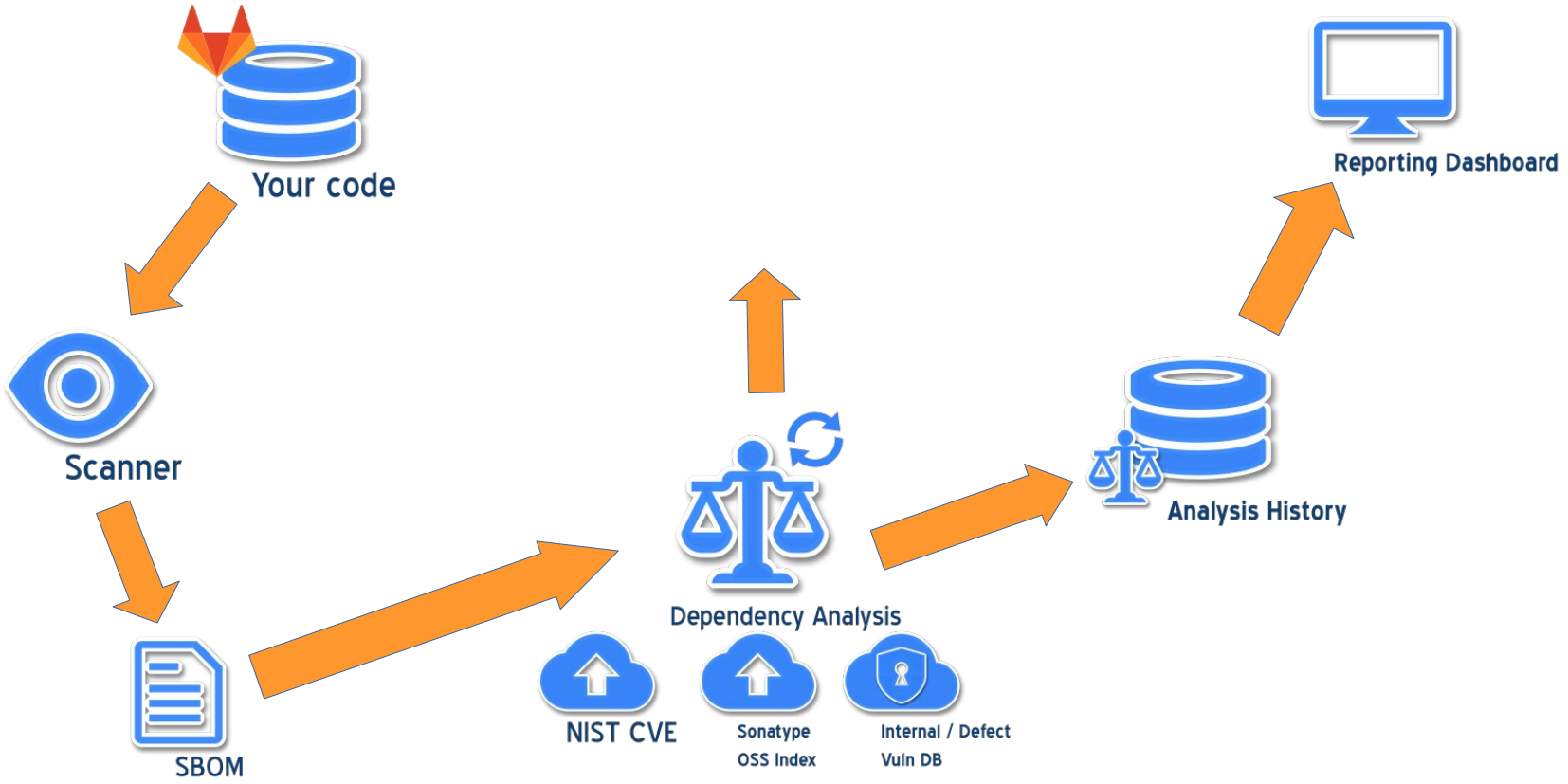
Dependency on open-source in 2022

- **Most cited role responsible for addressing vulnerabilities (82% of respondents) :**
  - Application Developers
- **How do we shift the onus of secure code upstream ?**

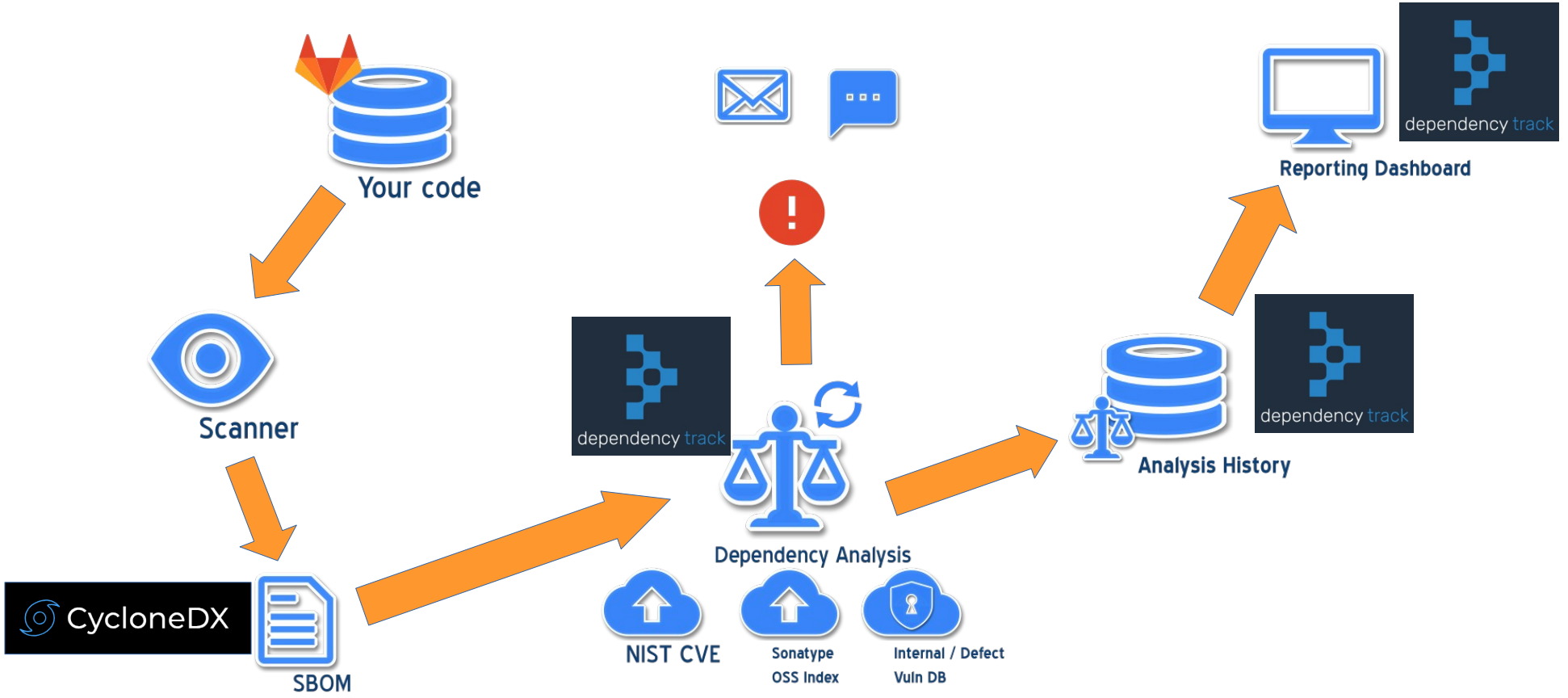


Source : Snyk DevSecops insights survey 2022

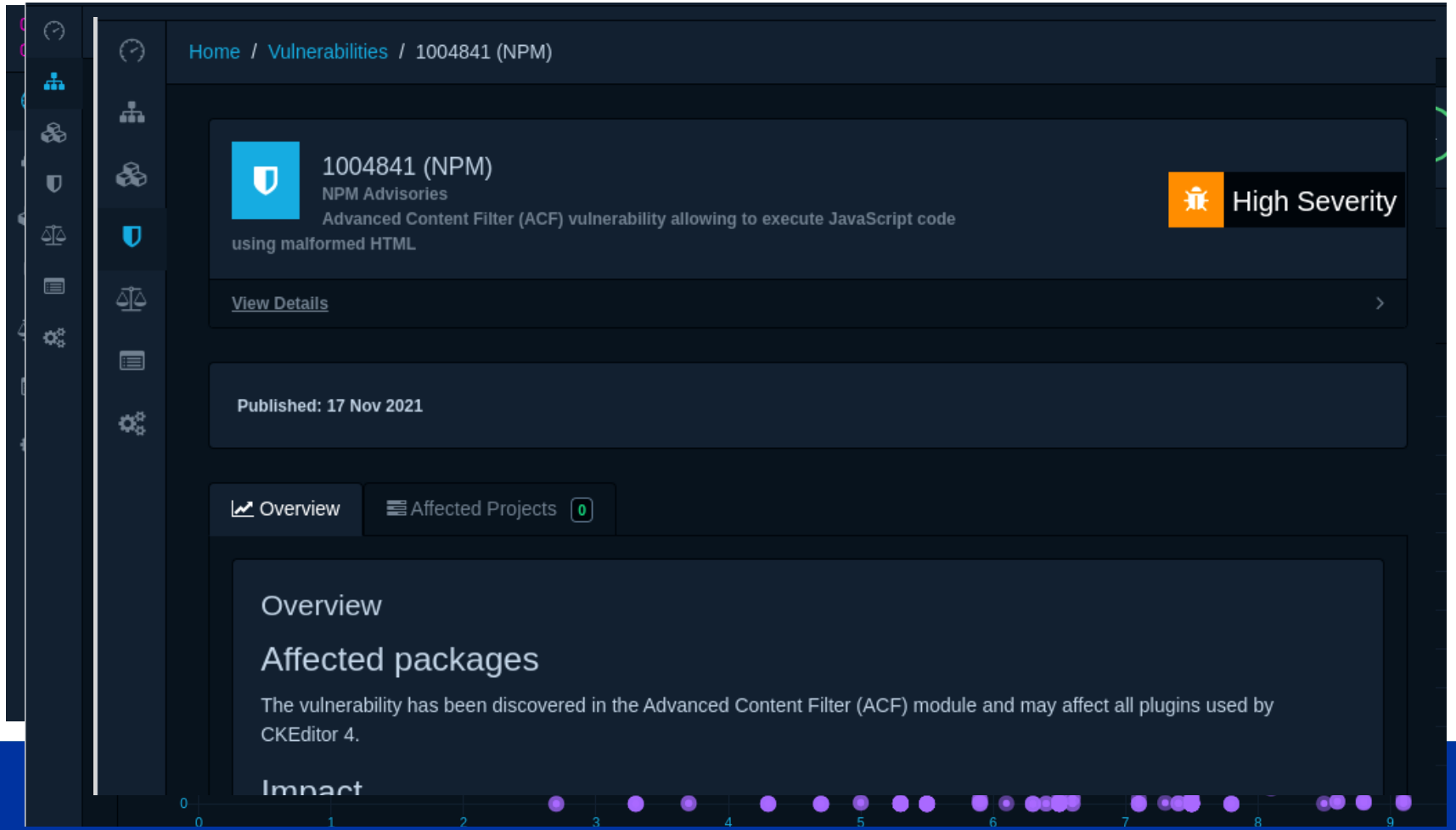
SBOM life cycle



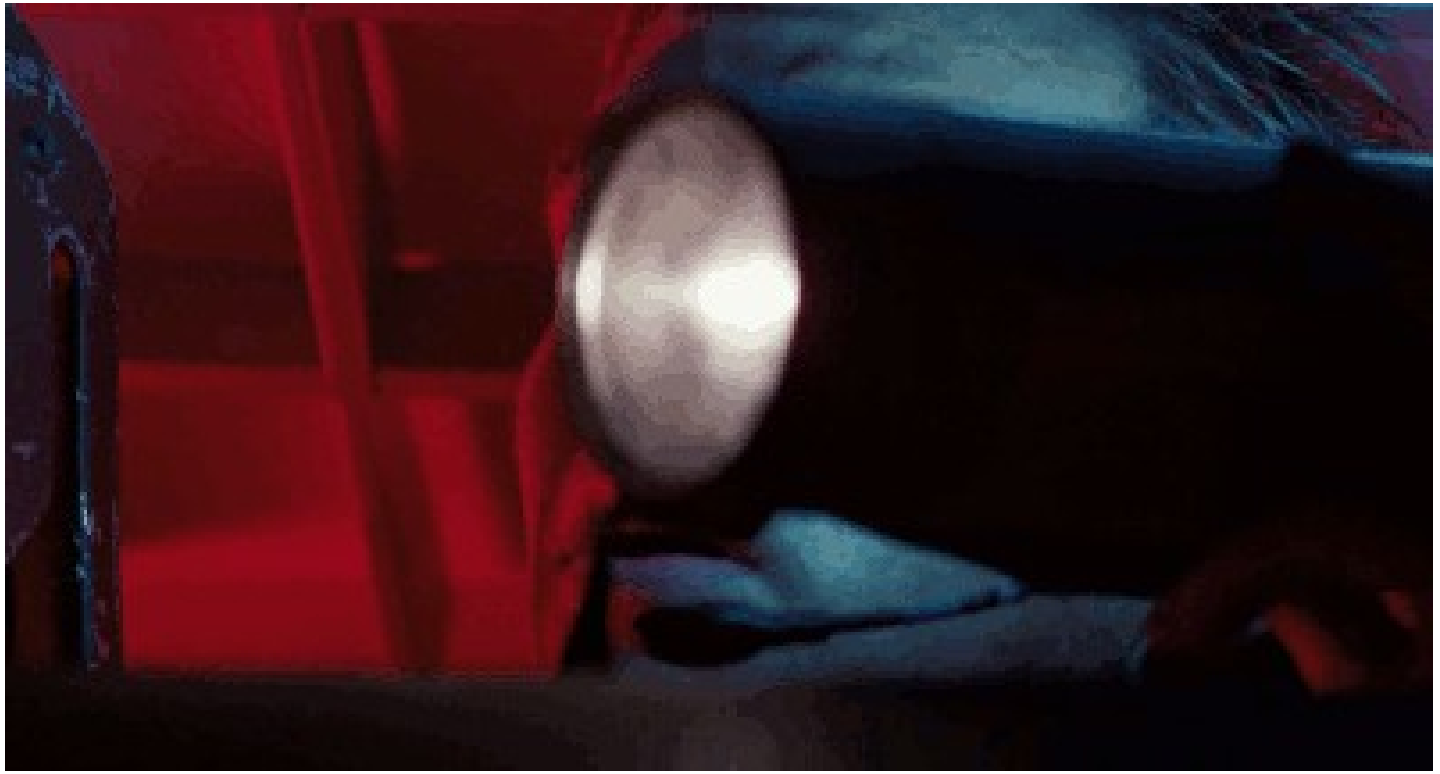
SBOM life cycle – in CERN Acc Controls



Dependency Track



Sample CERN Acc controls portfolio



Sample CERN Acc controls portfolio

- **6 representative accelerator controls projects**
  - **84** critical vulnerabilities over **1841** dependencies
  - A maximum CVSS of 10 (RCE, DOS, privilege escalation)
  - **22%** of vulnerable dependencies (only)
  - Serialization, authentication are pain points
- **Mitigations**
  - **100%** of critical vulnerabilities could be solved with a version upgrade (not always trivial, not always the case)

How to improve our situation ?

- **Better inventory and coordination**
  - “Who’s going to fix openssl ?”
  - “Who’s responsible for upgrading library XYZ ?”
- **Better communication**
  - “Where are we with the patching of log4j ?”
  - How can we keep track of vulnerabilities that :
    - Do not affect us at all
    - Are not **currently** affecting us

Perspectives at CERN

- **Automate and integrate SBOM into container life cycle**
  - Gitlab CI templates
  - Merge Request triggers and policy enforcement
- **Provide developer portals to tie our inventory to SBOM and vulnerability metrics**
  - Better accountability and visibility

So go ahead !

- **Open source tools are readily available**
  - Simple and free to deploy → run your projects through
  - Centralize SBOM documents and monitor them
- **SBOM Standards can handle more than dependencies**
  - Licensing
  - Digital signatures, Service dependency
  - Vulnerability information exchange etc…


CERN

Vulnerability audit cycle

Contents credits

- Slide 1 : DHL Stadium, Cape Town
- Slide 4 : XKCD : Dependency https://xkcd.com/2347/
- Other diagrams under CERN Copyright 2023