Software Bill of Materials

Laurie Williams

Laurie_williams@ncsu.edu

https://ntia.gov/files/ntia/publications/ntia_sbom_energy_jan2021overview_0.pdf

Supply Chain as an (inter)national priority

Executive Order on Improving the Nation's Cybersecurity

MAY 12, 2021 • PRESIDENTIAL ACTIONS

(Section 4e) Within 90 days of publication of the preliminary guidelines pursuant to subsection (c) of this section, the Secretary of Commerce acting through the Director of NIST, in consultation with the heads of such agencies as the Director of NIST deems appropriate, shall issue guidance identifying practices that enhance the security of the software supply chain. Such guidance may incorporate the guidelines published pursuant to subsections (c) and (i) of this section.

→ Software Bill of Materials (SBOM) Generation
Self-Attestation of Secure Development Practices

What is a Software Bill of Materials (SBOM)

- An SBOM is a formal, machine-readable inventory of software components and dependencies, information about those components, and their hierarchical relationships ➔ transparency!
- SBOMs have been worked on for a decade – just prominent now
- Central body: US National Telecommunications and Information Administration (NTIA) – US Dept of Commerce
- Standards:
  - Linux Foundation Software Package Data Exchange® (SPDX®): https://spdx.github.io/spdx-spec/
  - OWASP CycloneDX: https://cyclonedx.org
  - ISO Software Identification (SWID) Tags: ISO/IEC 19770-2:2015

JuiceBox SBOM

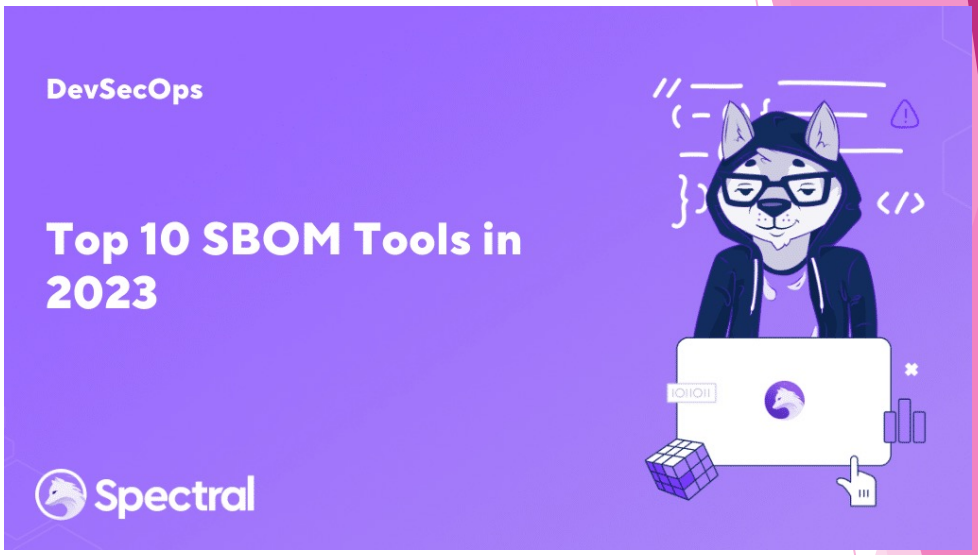

```
    "type": "library",
    "bom-ref": "pkg:npm/content-type@1.0.4",
    "name": "content-type",
    "version": "1.0.4",
    "description": "Create and parse HTTP Content-Type header",
    "hashes": [
      {
        "alg": "SHA-1",
        "content": "e138cc75e040c727b1966fe5e5f8c9aee256fe3b"
      }
    ],
    "licenses": [
      {
        "license": {
          "id": "MIT"
        }
      }
    ],
    "purl": "pkg:npm/content-type@1.0.4",
    "externalReferences": [
      {
        "type": "website",
        "url": "https://github.com/jshttp/content-type#readme"
      },
      {
        "type": "issue-tracker",
        "url": "https://github.com/jshttp/content-type/issues"
      },
      {
        "type": "vcs",
        "url": "git+https://github.com/jshttp/content-type.git"
      }
    ]
  },
  {
    "type": "library",
    "bom-ref": "pkg:npm/debug@2.6.9",
    "name": "debug",
    "version": "2.6.9",
```

Exercise

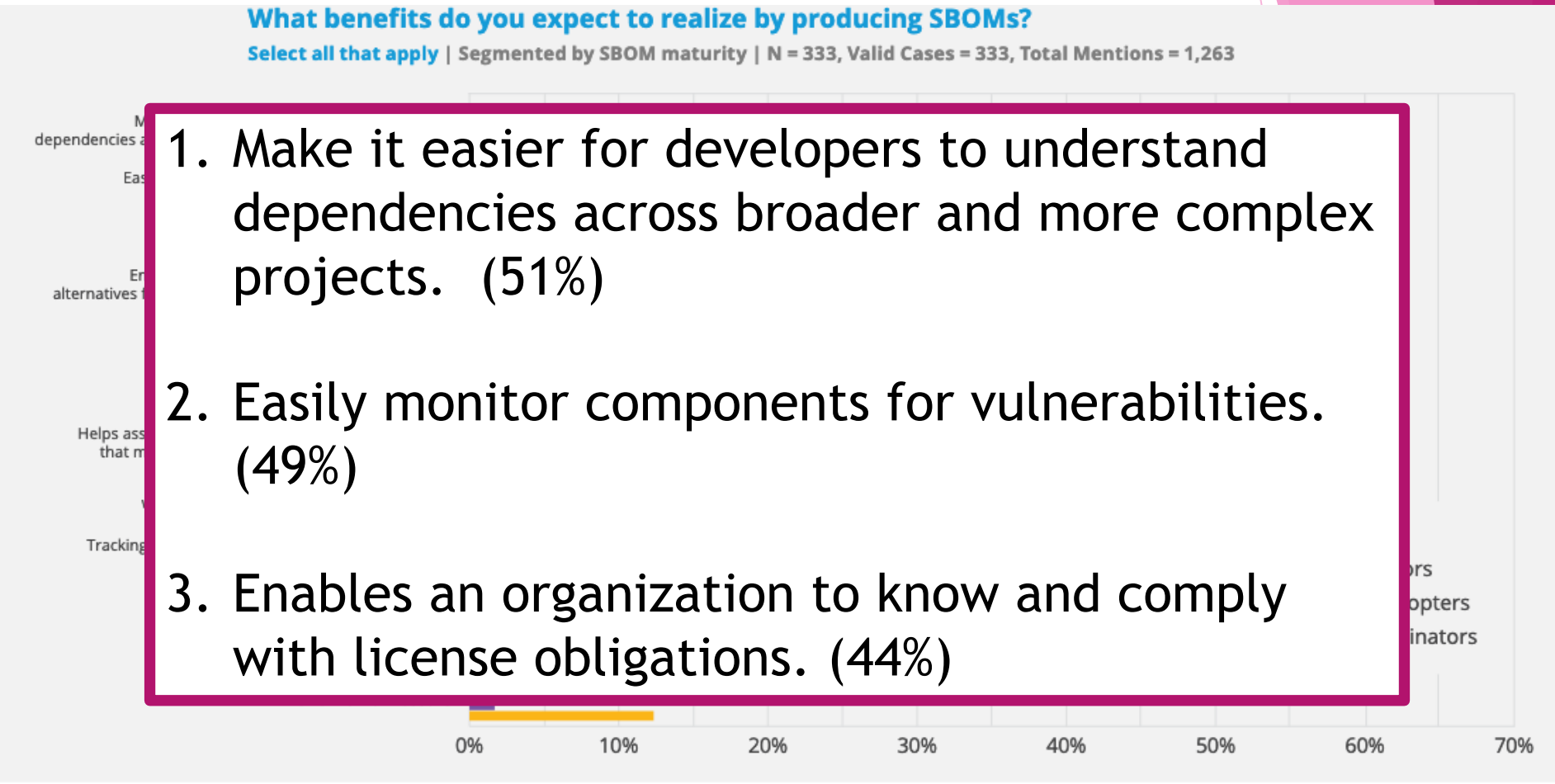
- Go to JuiceBox SBOM
  - https://github.com/CycloneDX/bom-examples/tree/master/SBOM/juice-shop/v11.1.2
  - https://tinyurl.com/bda62uze
- Find the component you worked with for the Dependency Check exercise. Look around at the fields.

SBOM Generation Tools

1. FOSSA
2. Spectral
3. Jit
4. Codenotary
5. Jfrog
6. Anchore
7. Cybeats
8. Endor Labs
9. Rezilion
10. SPDX SBOM Generator (GitHub open source)



Producing SBOM (January 2022)



1. Make it easier for developers to understand dependencies across broader and more complex projects. (51%)

2. Easily monitor components for vulnerabilities. (49%)

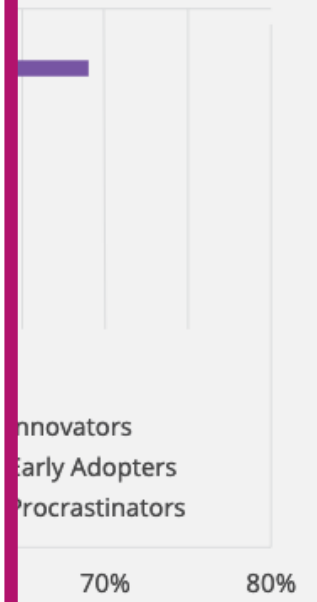
3. Enables an organization to know and comply with license obligations. (44%)

Consuming SBOMs (January 2022)

1. We are unclear that the industry is committed to requiring SBOMs, or whether it is optional. (49%)

2. We are uncertain as to whether there are tools available to automating the consumption of SBOMs. (48%)

3. We are unclear as to whether there is industry consensus on what an SBOM should contain. (44%)



Vulnerability Exploitability eXchange (VEX)

- VEX is a machine-readable artifact to allow a software supplier or other parties to assert the status of specific vulnerabilities in a particular product (similar to a security advisory)


```
"vulnerabilities": [
  {
    "id": "CVE-2021-44228",
    "source": {
      "name": "NVD",
      "url": "https://nvd.nist.gov/vuln/detail/CVE-2021-44228"
    },
    "analysis": {
      "state": "not_affected",
      "justification": "code_not_present",
      "response": ["will_not_fix"],
      "detail": "This version of Product ABC is not affected by the vulnerability. Class with vulnerable code was removed before shipping.
    },
    "affects": [
      {
        "ref": "product-ABC"
      }
    ]
  }
```

Summary

- Lots of focus on generating SBOM, tools emerging, SCA tools provide a lot of information
- Exploitability/reachability can be questioned … do I really need to update?
  - VEX is intended to help handle this
  - This is a good research area!
- SBOM production is getting better and better
  - Though comparison been tools can lead to questions
  - Some handle transitive/indirect dependencies different or not at all
- SBOM sharing, SBOM consumption are pretty immature