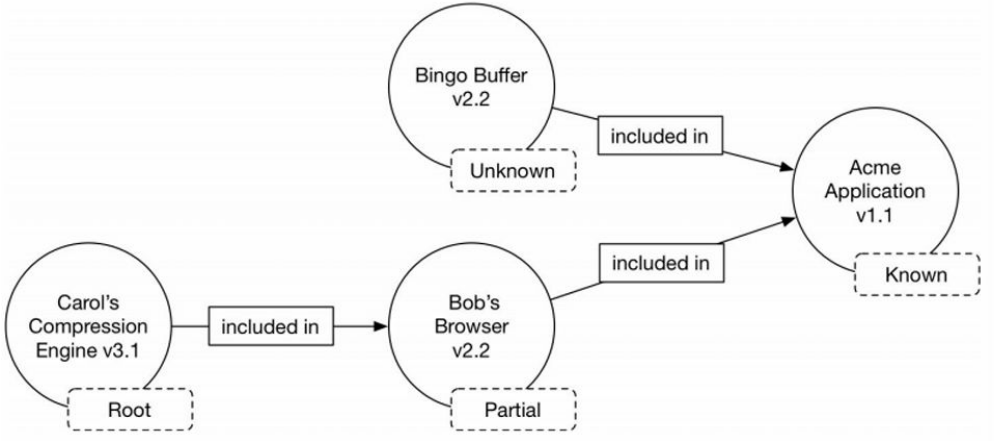
SBOM Visualization and Exploration Toolkit

Thomas Eby, Pranav Gonepalli, Manel Leong,
Camron Rule, Skyler Walker, Duohan Xu, Rachel Zheng

College of William & Mary
CSCI 435: Software Engineering
December 4, 2024

What is an SBOM?

- Software Bill of Materials (SBOM) = A formal document that is machine-readable and describes the components that make up a software product, their relationships, and their dependencies.



Relevance of SBOMs

- Executive Order 14028 (5/2021) for Cybersecurity
  - Requires all federal contractors to share threats, vulnerabilities, and cyber incidents to the federal government.
  - Requires the creation of SBOMs

BRIEFING ROOM

Executive Order on Improving the Nation's Cybersecurity

MAY 12, 2021 • PRESIDENTIAL ACTIONS

By the authority vested in me as President by the Constitution and the laws of the United States of America, it is hereby ordered as follows:

Relevance of SBOMs

- Implications: SBOM popularity will increase for private sector to comply with federal regulations.
- License and vulnerability information
- Dependency hierarchy

Motivation

- Difficulties with SBOMs:
- Difficult for humans to read
- Important information must be extracted and visualized
- SBOM generation tools exist, but few tools for visualization exist


```
{
  "bomFormat": "CycloneDX",
  "specVersion": "1.2",
  "serialNumber": "urn:uuid:371ffb8c-c11e-42b5-b5b9-9280fc62783e",
  "version": 1,
  "metadata": {
    "timestamp": "2020-08-03T08:53:09.834Z",
    "tools": [
      {
        "vendor": "CycloneDX",
        "name": "Node.js module",
        "version": "2.0.0"
      }
    ],
    "component": {
      "type": "library",
      "bom-ref": "pkg:npm/protonmail-web@4.0.0-beta.20",
      "name": "protonmail-web",
      "version": "4.0.0-beta.20",
      "description": "Angular frontend for protonmail.com",
      "licenses": [
        {
          "license": {
            "id": "MIT"
          }
        }
      ],
      "purl": "pkg:npm/protonmail-web@4.0.0-beta.20",
      "externalReferences": [
        {
          "type": "website",
          "url": "https://github.com/ProtonMail/WebClient#readme"
        },
        {
          "type": "issue-tracker",
          "url": "https://github.com/ProtonMail/WebClient/issues"
        },
        {
          "type": "vcs",
          "url": "git+https://github.com/ProtonMail/WebClient.git"
        }
      ]
    }
  },
  "components": [
```

Solution

- Open-source, Django-based web application that extracts and visualizes the most important information in an SBOM:
  - Software dependency tree
  - License distribution
  - Top vulnerability identification and vulnerability distribution

Overview of Accomplishments

- Parsing
  - SPDX 2.2, 2.3, 3.0
  - CycloneDX
- Tree Visualization
- Analysis
  - Licenses
  - Vulnerabilities

Demo

Testing and Validation

- Mostly manual testing
  - We have a set of ~15 SBOMs that we used to test
  - These were taken from public GitHub repos
- Security Testing
  - Wrote one test class based on a specific SBOM file
  - Sanity check whenever code was modified

Limitations and Improvements

- More in depth license analysis
- Representation of more info from CycloneDX sboms
    - Additional pages only visible when processing a CycloneDX file
- Security analysis may not work on some SBOMs and the quantity/quality of information shown is highly dependent on the SBOM

Lessons from Project

- There are many factors to consider when incorporating 3rd party software/ packages (eg. installation, compatibility, updated versions)
- Consistent communication is crucial to ensure that interfaces between different components (e.g., frontend and backend) work as expected and that teams are on the same page.
- Unexpected preconditions or requirements may come up during the software process that require creativity to solve and redesign, which is one instance where agile is most useful (e.g., SBOMs being stored as graphs rather than trees).

Contributions

Skyler:

- Coordinated and organized project meetings.
- Debugged tree visualization and implemented license processing and display.
- Created the PDF preview and set up page transitions.

Camron:

- Wrote documentation
- Set up and helped to improve initial tree visualization
- Created frontend elements on license and security pages

Pranav:

- Front-end work and design
- Special focus on the component info sidebar and relationships
- Bug fixes and quality of life changes

Contributions

Manel:

- Created Tree Builder
- Created Relationship Map Builder
- Created SPDX 2 xml parsers
- Aided in backend refactoring and planning

Thomas:

- Created initial tree-building algorithm
- Created SPDX 2 json parser
- Created CycloneDX json and xml parsers

Rachel:

- Security and license analysis of SBOM's

Duohan:

- Created SPDX 3 json and xml parsers

References

- “Improving the nation’s cybersecurity,” *Federal Register*, May 17, 2021. https://www.federalregister.gov/documents/2021/05/17/2021-10460/improving-the-nations-cybersecurity