SBOM Sprint #5

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

College of William & Mary
CSCI 435: Software Engineering
November 25, 2024

Sprint Goal

- Add final major software functionality
  - More SBOM versions and vulnerabilities
  - Integrate license and vulnerability analysis into frontend

Sprint Backlog - Front end

- New Feature Additions
  - License data pipeline to frontend (transmission, processing, display)
  - Vulnerability analysis display in frontend
  - Print PDF button on PDF Preview page
  - Add component details to sidebar in tree visualization page
- Quality of Life Changes

Sprint Backlog - Backend

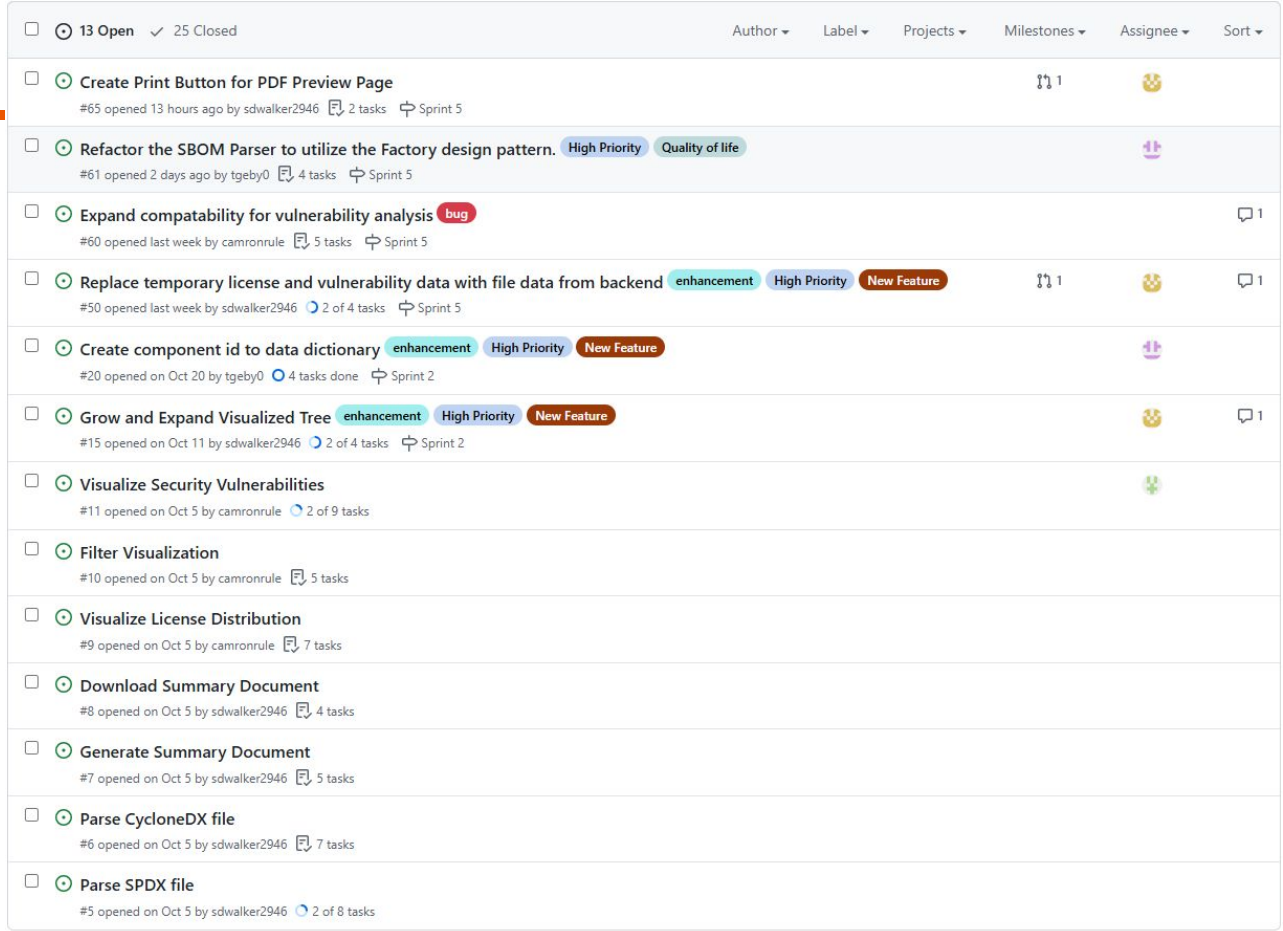
- Refactor of parsers to use a factory design pattern
  - Split parsers into 4 classes
    - CycloneDx xml, CycloneDx Json, SPDX xml, SPDX Json
  - Improves readability
- Implemented and integrated CycloneDx Json and XML parsers
- Security vulnerabilities endpoint
- License endpoint
- Create RelationshipMap builder and endpoint
- Bug fixes for TreeBuilder

Sprint Backlog - Backend

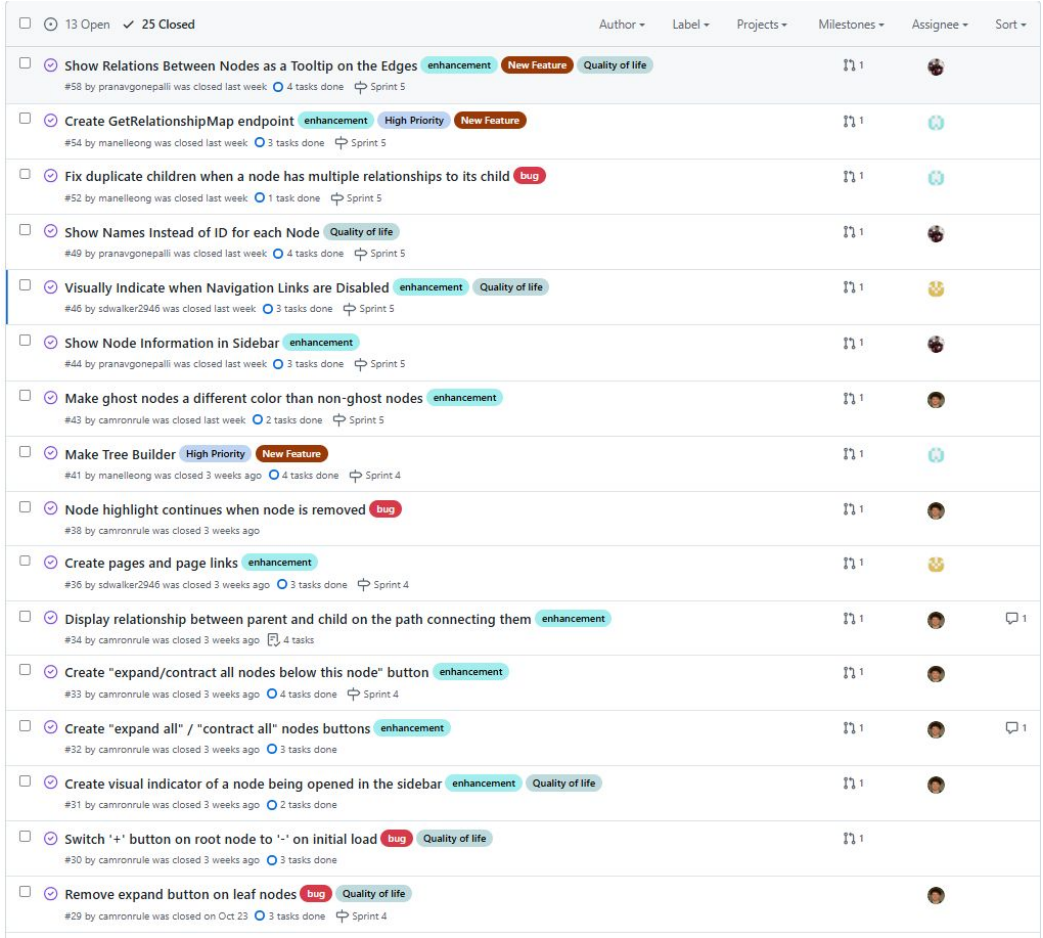
Works in progress

- Integrating SPDX 3.0 Json parsing
- Implementing SPDX 3.0 xml parsing
- SPDX 2.2, 2.3 xml parsing

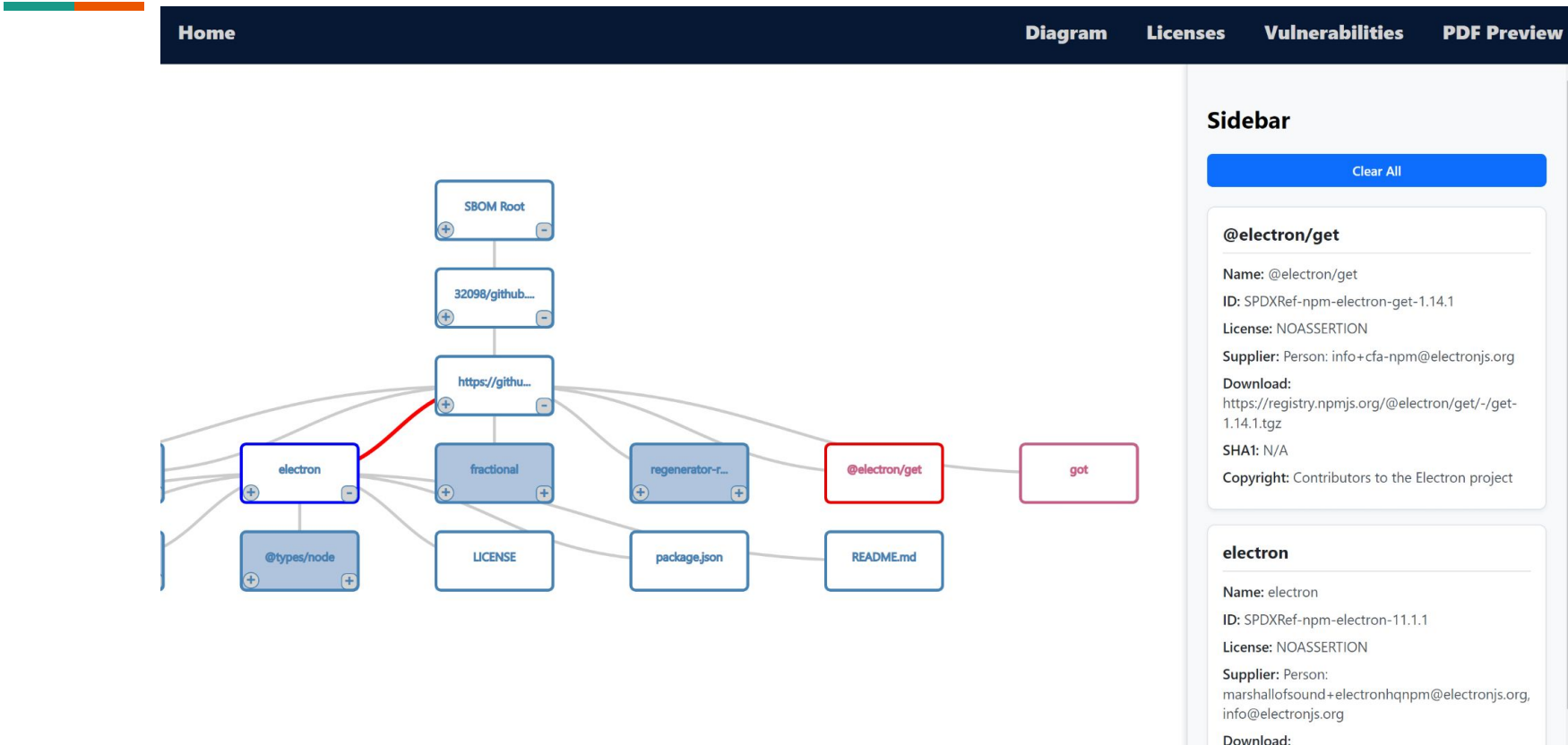
Issue Tracker



Issue Tracker (Continued)



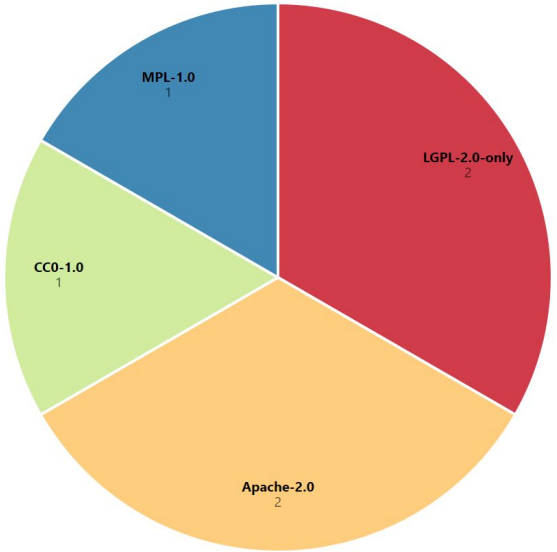
Tree Visualization Page



Licenses Page

Number of Distinct Licenses: 4

Distribution



10 Most Frequent License Types

LICENSE NAME	COUNT	COMPOSITION
LGPL-2.0-only	2	18.2%
Apache-2.0	2	18.2%
CC0-1.0	1	9.1%
MPL-1.0	1	9.1%

Learn more about these licenses

Vulnerabilities Page

CVSS Severity Counts

Critical: 1 (7.1%)

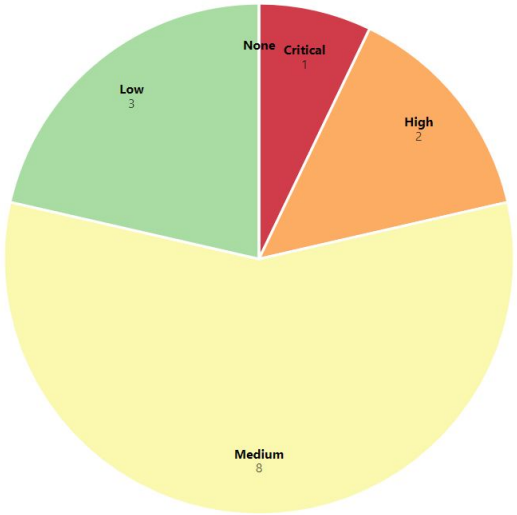
High: 2 (14.3%)

Medium: 8 (57.1%)

Low: 3 (21.4%)

None: 0 (0.0%)

Distribution



Top 10 Vulnerabilities by CVSS Score

COMPONENT_NAME	CVE_ID	SCORE	SEVERITY	DESCRIPTION
SPDXRef-npm-minimist-1.2.5	CVE-2021-44906	9.8	CRITICAL	Minimist <=1.2.5 is vulnerable to Prototype Pollution via file index.js, function setKey() (lines 69-95).
SPDXRef-npm-electron-11.1.1	CVE-2023-5217	8.8	HIGH	Heap buffer overflow in vp8 encoding in libvpx in Google Chrome prior to 117.0.5938.132 and libvpx 1.13.1 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page. (Chromium security severity: High)
SPDXRef-npm-lodash-4.17.20	CVE-2021-23337	7.2	HIGH	Lodash versions prior to 4.17.21 are vulnerable to Command Injection via the template function.
SPDXRef-npm-electron-11.1.1	CVE-2021-39184	6.8	MEDIUM	Electron is a framework for writing cross-platform desktop applications using JavaScript, HTML and CSS. A vulnerability in versions prior to 11.5.0, 12.1.0, and 13.3.0 allows a sandboxed renderer to request a "thumbnail" image of an arbitrary file on the user's system. The thumbnail can potentially include significant parts of the original file, including textual data in many cases. Versions 15.0.0-alpha.10, 14.0.0, 13.3.0, 12.1.0, and 11.5.0 all contain a fix for the vulnerability. Two workarounds aside from upgrading are available. One may make the vulnerability significantly more difficult for an attacker to exploit by enabling `contextIsolation` in one's app. One may also disable the functionality of the `createThumbnailFromPath` API if one does not need it.

Demo

Scanning for Security Vulnerabilities

- The security scan fails on some sboms likely because of no purl component
- To do:
  - Messaging for the user that security scanning failed/may not be as detailed
  - Test more sboms to improve robustness
  - Look into threads so security scan starts in the background
  - Write installation of security tools into documentation and explain clearly how security is evaluated

More in Depth License Analysis

- Trivy categorizes the licenses into different categories based on restrictiveness and we can incorporate this information into the analysis we display in the frontend

Classification	Severity
Forbidden	CRITICAL
Restricted	HIGH
Reciprocal	MEDIUM
Notice	LOW
Permissive	LOW
Unencumbered	LOW
Unknown	UNKNOWN

Next Sprint Backlog

- Improve compatibility of vulnerability analysis
- Integrate SPDX 3.0 Json
    - Implement and integrate SPDX 3.0 XML
    - Finish and integrate SPDX 2.2, 2.3 XML
- Make uml diagrams

Lessons

- Backend tasks becoming numerous, need to redistribute labor to backend team.

Contributions

Manel: Created RelationshipMapBuilder and getRelationshipMap endpoint. Gave the idea to use a factory design pattern for building the parser. Worked on SPDX 2.2, 2.3 XML Parser.

Skyler: Integrated backend license data into frontend and implemented license processor, print preview button, and QoL changes to homepage

Thomas: Refactored app to use a factory design pattern when making the parser. Implemented Cyclone Dx Json and Xml parsers. Added license info endpoint.

Pranav: Made changes on the frontend, showing node information in the sidebar, and relationship between nodes as a tooltip on the edges.

Camron: Connected backend vulnerability output to the vulnerability analysis page, colored ghost nodes, improved documentation

Rachel: Improve error handling and compatibility for vulnerability scanning and looked into providing more information on licenses

Evan:.

Duohan: Implementing past mechanisms in parsers to refactored parsers

Everyone: