SBOM Sprint #4

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 6, 2024

Sprint Goal

- Implement templates for other pages and repair the backend to correctly parse all SBOMs.

Sprint Backlog - Front end

- Create page navigation
- Create content on each page (functionality will be added later)
- Quality of life and usability changes for tree visualization
  - Expand/collapse all levels
  - Highlight for selected nodes
- Improve documentation

Sprint Backlog - Backend

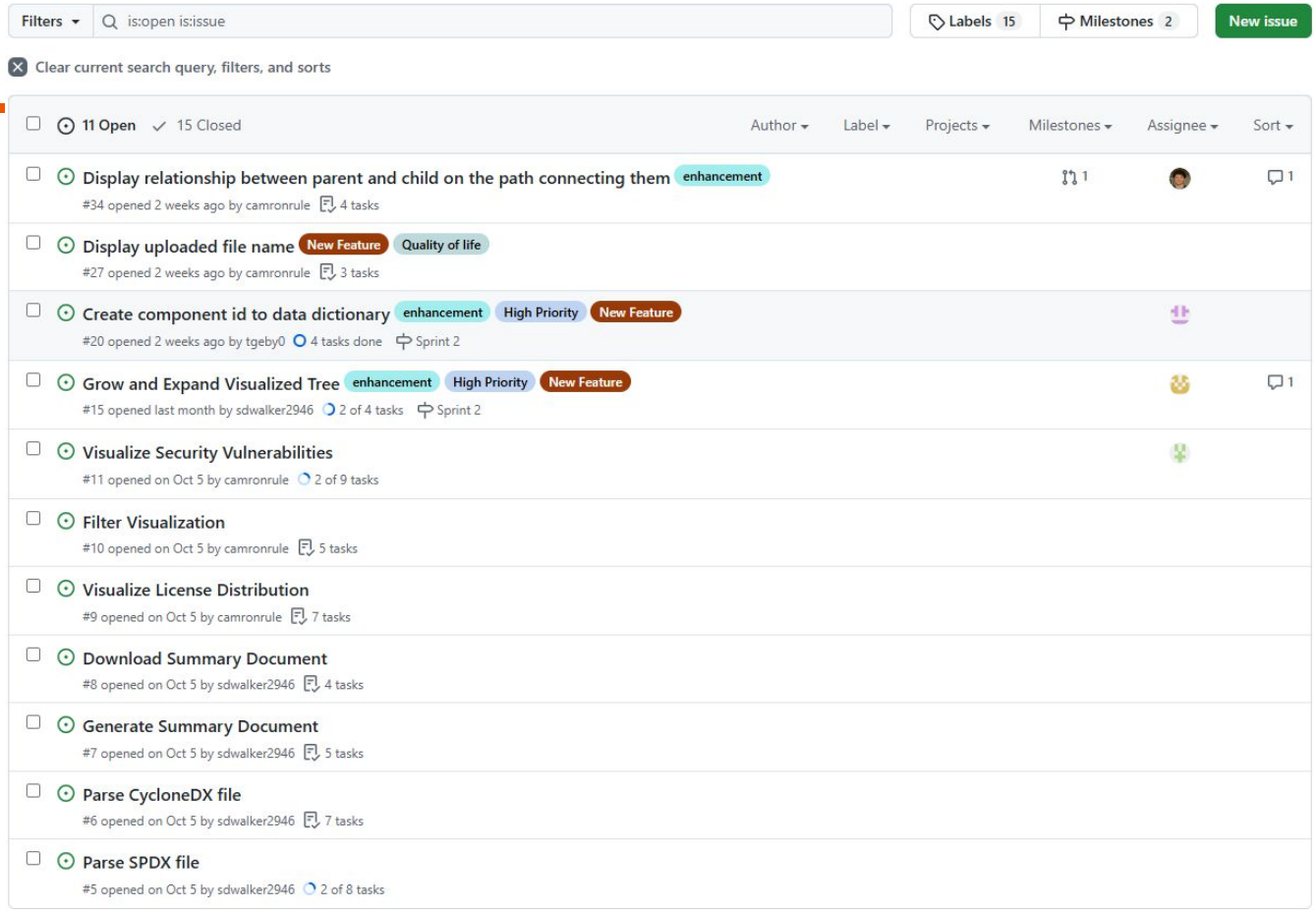
- Create SPDX 2.2, 2.3 Json Parser
- Create TreeBuilder and backend Tree Object
- Create id-data-map endpoint
- Get tree endpoint working
- Added feature flags
- Identified security vulnerabilities in a SBOM, severity distribution, and top 10 vulnerabilities

Sprint Backlog - Backend

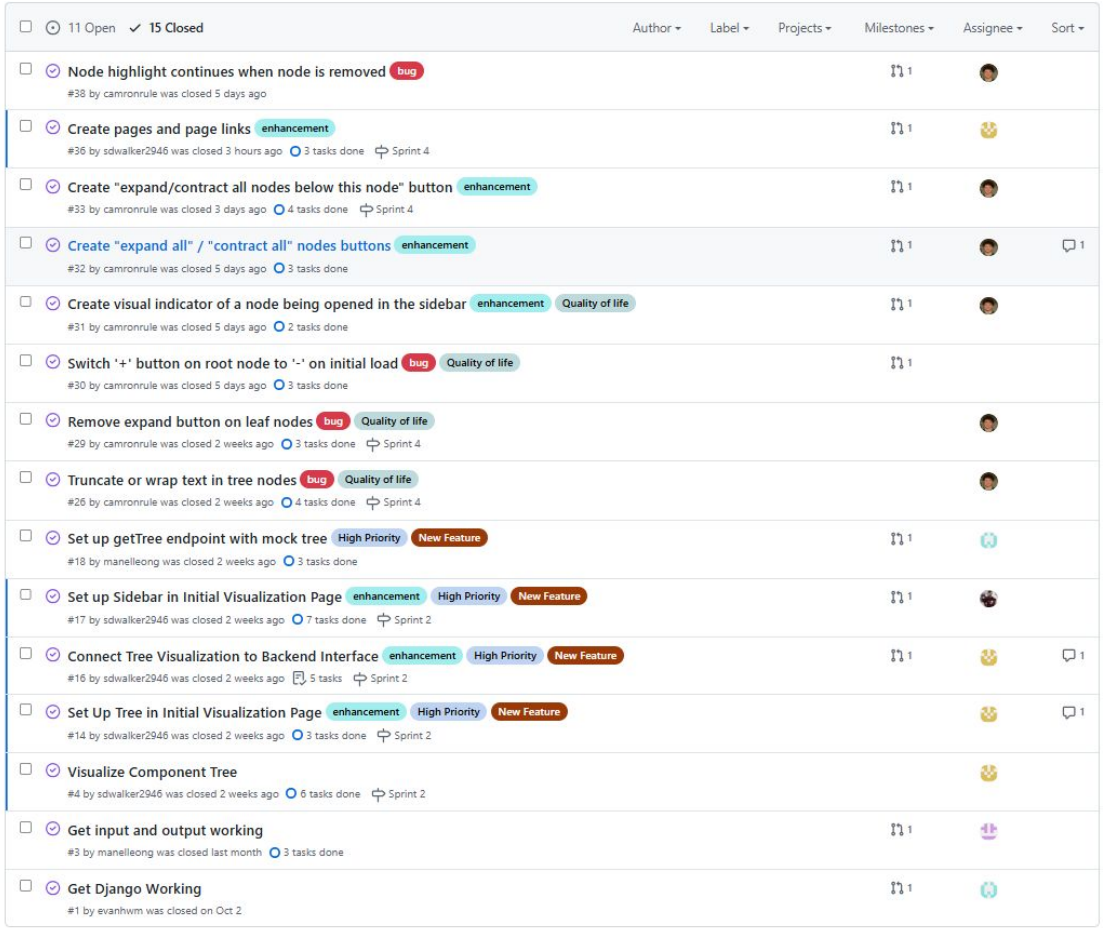
Works in progress

- SPDX 3.0 Json parsing
- SPDX License data parsing
  - Implemented in the 2.2 and 2.3 Json versions
    - Almost complete in 3.0 Json
  - Licence frequency map & license to component id map
  - No endpoint set up for it yet
- Connect security info with the rest of the backend

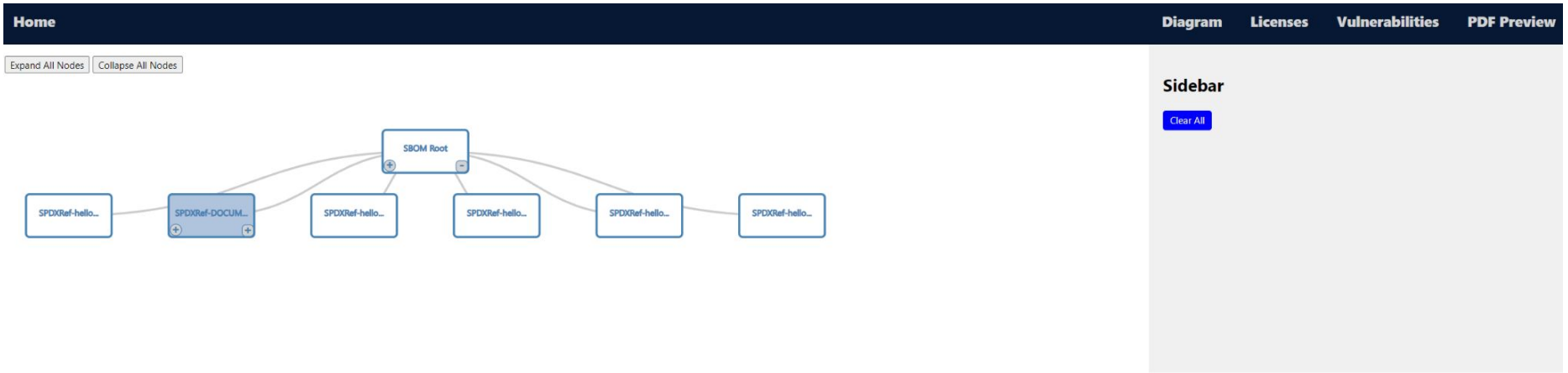
Issue Tracker



Issue Tracker (Continued)



Tree Visualization Page

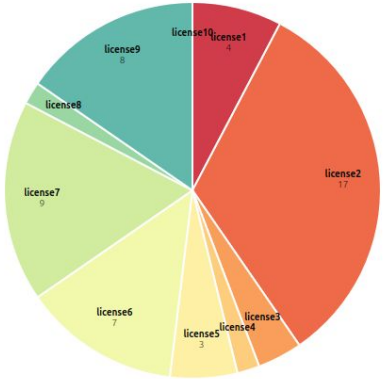


Licenses and Vulnerabilities Pages

Licenses

Number of Distinct Licenses: 10

Distribution



10 Most Frequent License Types

LICENSE NAME	VERSION	COUNT	COMPOSITION
license2	1.0	17	27.4%
license7	1.0	9	14.5%
license9	1.0	8	12.9%
license6	1.0	7	11.3%
license1	1.0	4	6.5%
license5	1.0	3	4.8%
license3	1.0	2	3.2%
license4	1.0	1	1.6%
license8	1.0	1	1.6%
license10	1.0	0	0.0%

Learn more about these licenses

Vulnerabilities

CVSS Severity Counts

Critical: 10 (6.2%)

Critical-High: 15 (9.3%)

High: 20 (12.3%)

High-Medium: 25 (15.4%)

Medium: 10 (6.2%)

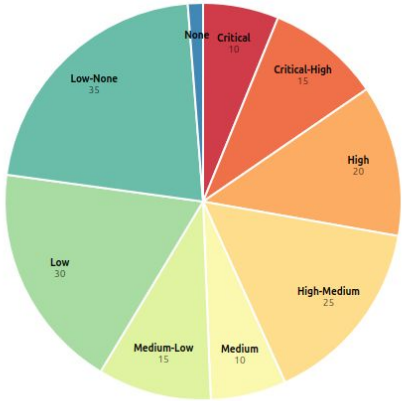
Medium-Low: 15 (9.3%)

Low: 30 (18.5%)

Low-None: 35 (21.6%)

None: 2 (1.2%)

Distribution



Top 10 Vulnerabilities by CVSS Score

COMPONENT NAME	VERSION	SCORE
software_component11	1.0	9
software_component9	1.0	8
software_component3	1.0	7
software_component5	1.0	6
software_component6	1.0	6
software_component2	1.0	5
software_component1	1.0	4
software_component8	1.0	2
software_component7	1.0	1
software_component4	1.0	0

Demo

Scanning for Security Vulnerabilities

- Found a new tool: trivy
  - Pros: returns more information for vulnerabilities
  - Cons: seems to have more trouble working with some versions of sboms or if sboms are not in a specific format
  - Idea: use trivy first and if that fails use bomber (would have to find package/api that can get more info)
- Wrote a script to reformat output from a trivy scan
  - Includes summary and top 10 info


{
"Summary": {
"SeverityDistr": {
"CRITICAL": 1,
"HIGH": 2,
"MEDIUM": 8,
"LOW": 3,
"NONE": 0
},
"Top_10": {
"CVE-2021-44906": {
"SBOM_ID": "SPDXRef-npm-minimist-1.2.5",
"SeveritySource": "ghsa",
"Title": "minimist: prototype pollution",
"Description": "Minimist <=1.2.5 is vulnerab
"Severity": "CRITICAL",
"CWE_IDs": [
"CWE-1321"
],
"Displayed_CVSS": 9.8
},
"CVE-2023-5217": {
"SBOM_ID": "SPDXRef-npm-electron-11.1.1",
"SeveritySource": "ghsa",
"Title": "libvpx: Heap buffer overflow in vp
"Description": "Heap buffer overflow in vp8
"Severity": "HIGH",
"CWE_IDs": [
"CWE-787"
],
"Displayed_CVSS": 8.8


"SPDXRef-npm-electron-11.1.1": {
"PURL": "pkg:npm/electron@11.1.1",
"Dependents": [],
"InstalledVersion": "11.1.1",
"Vulnerabilities": [
{
"CVE_ID": "CVE-2023-5217",
"SeveritySource": "ghsa",
"Status": "fixed",
"Title": "libvpx: Heap buffer overflow in vp8 encoding in libvpx",
"Description": "Heap buffer overflow in vp8 encoding in libvpx in Google
"Severity": "HIGH",
"CWE_IDs": [
"CWE-787"
],
"CVSS": {
"ghsa": {
"V3Vector": "CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H",
"V3Score": 8.8
},
"nvd": {
"V3Vector": "CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H",
"V3Score": 8.8
},
"redhat": {
"V3Vector": "CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H",
"V3Score": 8.8
}
},
"Displayed_CVSS": 8.8,
"References": [],
"PublishedDate": "2023-09-28T16:15:10.98Z",
"LastModifiedDate": "2024-02-15T02:00:01.65Z"
},

Next Sprint Backlog

- Integrate vulnerability analysis into backend and display in frontend.
- Implement license analysis in backend to display in frontend.
- Figure out how to generate and display a PDF.
- Add component details to sidebar in tree visualization page.

Lessons

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

Contributions

Manel: Made TreeBuilder, Tree object, and got getTree endpoint working with parser.

Skyler: Added remaining pages, page transition functionality. Added content to licenses and PDF overview pages.

Thomas: Implemented SPDX 2.2 and 2.3 JSON parser

Pranav: Was really busy this sprint and wasn't able to make much progress on my end.

Camron: Implemented many QOL and bug fixes for the tree. Helped put together and style new pages.

Rachel: Worked on identifying security vulnerabilities, summarizing results, and formatting output from vulnerability scanners

Evan: Was also busy and did not get much work done.

Duohan: Implementing SPDX 3.0 JSON parse, preparing tests for CycloneDX formats

Everyone: