Mojaloop
Platform Quality & Security

PI-25 product council update, 29th October 2024

Sam Kummary MLF, Shuchita Prakash Community member, DMP 2024 mentee

Workstream Update

Workstream Name:	Mojaloop platform quality & security (PQS)
Roadmap Pillar:	Quality Product (Foundation)
Lead:	Sam Kummary
Workstream Objectives:	The focus for PI-25 will be on 1) GitHub maintenance of Mojaloop repositories (used in the platform), continuation of 2) Hardening Mojaloop core and addressing any known and reported quality, security issues. 3) We also want to start the activity of identifying all the third party dependencies (SBOMs) in Mojaloop to better manage situations when issues arise with any of them or as some of them are not maintained. 4) Bonus: Another objective this PI is to look into achieving OpenSSF badge for Mojaloop and blog posts.
Progress Against Objectives:	1. Mojaloop GitHub repos maintenance - tightening up on branch protection rules 2. Dependabot alerts 3. SBOM generation and automation - done 4. License issues. Vulnerability management - docs in progress
Anticipated Progress by PI End:	1. GitHub repositories that are all part of Mojaloop platform to follow best practices 2. Mojaloop release made with all high, critical dependabot security alerts addressed 3. SBOMs generated monthly and stored with timestamps
Roadblocks:	
Support Needed:	1. No dedicated team for the PI, thankful for the contributions so far. 2. Further contributions welcome.

Mojaloop PQS: SBOM for Mojaloop

Goals

1. Generate and document SBOMs for each repository/service.
2. Automate monthly publication of metrics and SBOMs.
3. Flag anomalies and discrepancies in SBOMs.
4. Maintain a list of open security alerts for core platform repositories.

What is an SBOM?

A Software Bill of Materials(SBOM) is a machine and human-readable list of a project's entire software inventory.

Why do organizations need an SBOM?

1. Transparency
2. Security
3. Compliance
4. Maintenance

Formats - generated/converted

xml - json - csv - html

What all does SBOM contain?

1. Open source components
2. Third-party components
3. Licenses
4. Versions of components
5. Patch status of components
6. Open source vulnerabilities
7. Package names
8. Package versions

CycloneDX Generator

1. It is the official OWASP SBOM tool
2. Supports many languages-C, C++, JavaScript, Java, Python
3. Comes with a CLI that can scan locally or as part of a CI/CD pipeline and an API server
4. Output format is CycloneDX.

Links

1. https://github.com/mojaloop/community-tools/tree/master/oss-stats/sbom
2. https://www.npmjs.com/package/%40cyclonedx/cyclonedx-npm
3. https://github.com/mojaloop/ml-api-adapter/pull/530

Mojaloop PQS: SBOM for Mojaloop

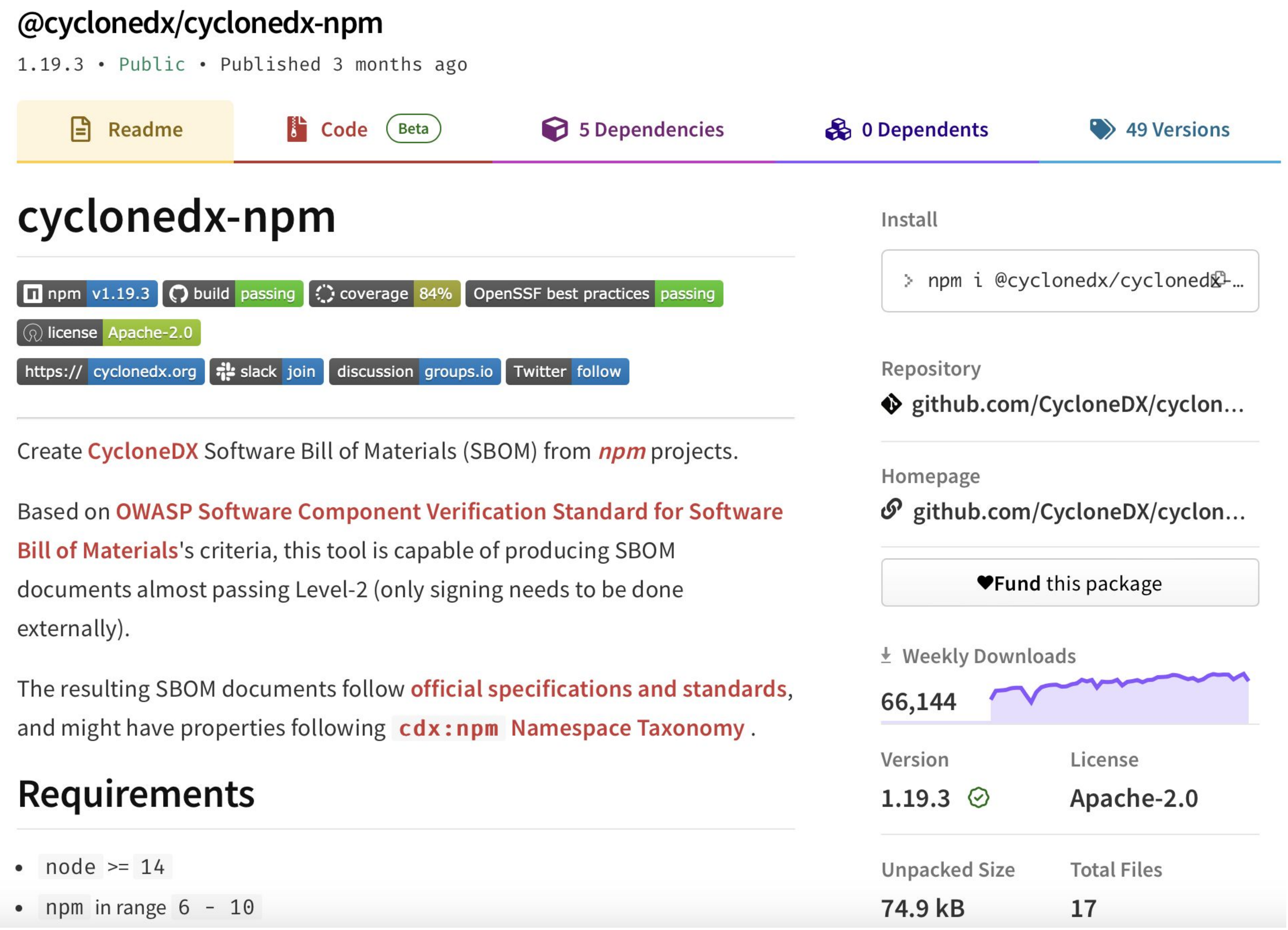
SBOM of all services included in Mojaloop Platform’s core

1. Tooling - CycloneDX and SPDX
2. Individual SBOMs for core repositories
3. Centralized list for all core / critical repositories
4. Automate generation

Package and Generation and Formats

SBOMs are generated for Node.js environment using the commands

- Installing CycloneDX generator
  - npm install --global @cyclonedx/cyclonedx-npm
- Generating sbom
  - cyclonedx-npm --output-file outputfile.json
  - cyclonedx-npm --output-format "XML" --output-file outputfile.xml
- Formats
  - XML - generation
  - JSON - generation
  - CSV - parsing, processing
  - HTML - visibility, presentation



@cyclonedx/cyclonedx-npm

1.19.3 • Public • Published 3 months ago

cyclonedx-npm

Create CycloneDX Software Bill of Materials (SBOM) from *npm* projects.

Based on OWASP Software Component Verification Standard for Software Bill of Materials's criteria, this tool is capable of producing SBOM documents almost passing Level-2 (only signing needs to be done externally).

The resulting SBOM documents follow official specifications and standards, and might have properties following `cdx:npm` Namespace Taxonomy .

Requirements

- `node` >= `14`
- `npm` in range `6 - 10`

Install

```
> npm i @cyclonedx/cyclonedx-...
```

Repository

github.com/CycloneDX/cyclon...

Homepage

github.com/CycloneDX/cyclon...

♥Fund this package

Weekly Downloads

66,144

Version	License
1.19.3 ✓	Apache-2.0

Unpacked Size	Total Files
74.9 kB	17

https://www.npmjs.com/package/@cyclonedx/cyclonedx-npm

Dependency Analysis

SBOM Process

1. **SBOM Analysis Repository-Wise**

   The repository is cloned, and the SBOM is generated in either XML or JSON format. The SBOM is then converted to CSV for parsing and finally to HTML for better visualisation.

2. **Collecting and Mapping Dependencies**

   All unique dependencies across the repositories are scanned, and a CSV is generated, mapping each component to the repositories that depend on it. This CSV also includes key data such as version, license type, and maintainer information.

3. **Databases and Queries**

   A CSV containing all components across repositories, along with their related data, is generated and uploaded to a database. This allows SQL queries to be run for efficient searching and analysis of dependencies and their attributes.

Mojaloop PQS: Mojaloop releases

Review core and related repositories released with Mojaloop for

1. Snyk alerts for Central-ledger repository are addressed
2. Dependabot alerts for Central-ledger repository are addressed (moderate, high and critical)
3. Ensure main branch is protected and collaborator list is up-to-date
4. Ensure open PRs are addressed, closing stale PRs
5. Update audit exceptions json file to remove exceptions added that are not necessary anymore
6. Close issues on the repository that are fixed / out-of-date
7. Ensure codeowners file is current
8. *SBOMs for all repositories published with a Mojaloop release - PI25*
9. *Address deprecated dependencies - PI25*

Reviewing updates and vulnerabilities (GitHub, NVD, other related media) constantly

1. Take necessary action for Mojaloop services as required
2. Provide guidance and mitigate exposure to Mojaloop following the Mojaloop CVD and Cybersecurity policies

Mojaloop PQS: PI-25 pipeline

1. SBOM for component and Mojaloop platform
2. Hardening of Mojaloop releases - continuation
3. SonarCloud - followup issues
4. Improve tooling used in CI/CD - ci tools, license checkers
5. Blog posts and guidance to implementers and improve documentation - in progress
6. OpenSSF checklist and compliance - next PI
7. Stress / Load testing of Mojaloop - next PI

PQS Workstream Mojaloop slack channel: https://mojaloop.slack.com/archives/C06UW0E2KBN

Mojaloop PQS: Objectives

1. Hardening of Mojaloop releases
2. Security testing of Mojaloop
   - Assess feedback from testing done by security experts, internally
   - Prioritize issues based on criticality and severity ✓
   - Update backlog with stories to address issues ✓
3. Review core and related repositories released with Mojaloop for
   - Ensure dependabot alerts are addressed
   - Ensure Snyk issues are addressed ✓
4. Review any tasks “to-do” in the codebase and related quality issues
5. Review PRs in critical repositories and address (close, approve / reject)
6. Update nodejs version for any services not yet upgraded ✓
7. Update maintenance tooling (license-scanner and such) ✓
8. GitHub maintenance of Mojaloop repositories (used in the platform) (continuation)
9. Hardening Mojaloop core and addressing any known and reported quality, security issues
10. SBOM of dependencies in Mojaloop to better manage situations ✓
11. Provenance of Mojaloop artefacts - PI26 candidate