

OpenChain-KWG Tooling SG 회의('23년 1월)

컴플라이언스 도구 분석

OSS Review Toolkit



2023.1.11.(수)
ETRI 오픈소스센터
박정숙

내용

- 개요
- 오픈소스 컴플라이언스 도구 조사
- ORT 사용법

참고자료

- 오픈소스 컴플라이언스 관련 도구 리스트: <https://oss-compliance-tooling.org/Tooling-Landscape/OSS-Based-License-Compliance-Tools/>
- ORT 소스코드: <https://github.com/oss-review-toolkit/ort>
- ORT 관련 동영상: “An Introduction to the OSS Review Toolkit”, Sebastian Schubert, 2022, https://www.youtube.com/watch?v=JaVpB_d0kWY
- ORT 설치 및 시험 방법 소개: [How to Run · OSS Review Toolkit \(lge-oss.github.io\)](https://lge-oss.github.io)
- ORT 설치 방법: Install OSS Review Toolkit (ORT) (canvasslabs.com), https://rivera.canvasslabs.com/help/install_ort#install_ort
- ORT 시작하기: [ort/getting-started.md at main · oss-review-toolkit/ort · GitHub](https://github.com/oss-review-toolkit/ort/blob/main/ort/getting-started.md)

개요

- **SW 공급망 관리의 중요성**

- 공급망에서 오픈소스 라이선스, 보안취약점 이슈 발생 가능

- **의존성 관리 도구 도입을 통해 공급망 관리 필요**

- 패키지 매니저 내용 분석

- Gradle, Maven, NPM, Pypi, Cocoapod 등

오픈소스 컴플라이언스 도구 유형

- **License scanning**
 - 라이선스, 라이선스 관련 문구, 저작권 문구, 저자 문구, ack 식별
- **Binary scanning**
 - SW 바이너리에서 사용된 SW 패키지를 식별하고 버전 확인
- **Source code scanning**
 - 소스 코드 및 기타 파일의 게시된 출처 식별
- **Component management**
 - 사용된 SW 구성 요소 및 제품 또는 프로젝트에서의 사용에 대한 정보를 중앙에서 수집, 재사용
- **Vulnerability scanning**
 - 보안취약점 정보 식별
- **Notice generation**
- **UI**

오픈소스 컴플라이언스 관련 도구들(1)

출처: <https://oss-compliance-tooling.org/Tooling-Landscape/OSS-Based-License-Compliance-Tools/>

No.	도구명	라이선스	특징	웹사이트
1	AboutCode Toolkit	Apache-2.0	- 프로젝트에서 사용하는 외부 SW 구성 요소에 대한 원본, 라이선스, 사용 및 기타 정보를 문서화하는 간단한 방법 제공 - 고지문을 생성하고 프로젝트에서 사용되는 재배포 가능한 소스코드 식별	https://www.aboutcode.org/
2	AboutCode Manager	Apache-2.0	- 시각적 UI 제공하여 ScanCode로 식별된 라이선스나 기타 알림을 평가하고 구성요소의 유효 라이선스에 대한 결론 기록 - Electron 기반, nexB의 AboutCode 도구를 사용하기 위한 기본 데스크톱/GUI 도구	https://www.aboutcode.org/
3	Apache Rat	Apache-2.0	- 라이선스에 중점을 둔 릴리즈 audit 도구(Apache Creadur 프로젝트의 일부) - Java로 코딩, Maven 및 Ant용 플러그인을 사용하여 명령줄에서 실행	https://creadur.apache.org/rat/
4	Apache Tentacles	Apache-2.0	- 릴리즈할 산출물이 포함된 저장소와의 인터랙션을 자동화하여 검토 작업 단순화 - Java로 코딩, CLI 실행	https://creadur.apache.org/tentacles/
5	Apache Whisker	Apache-2.0	- 조립된 응용이 올바른 법적 문서를 유지하도록 지원 - 복잡한 조립 응용 프로그램에 특히 유용	https://creadur.apache.org/whisker/
6	Bang	AGPL-3.0	- 바이너리 파일 분석 도구 - 바이너리 파일의 내용을 찾아 추출된 정보를 라이선스 준수, 보안 연구 또는 구성 분석과 같은 추가 분석에 사용할 수 있도록 만들 - 감지, 압축 풀기 및 레이블 지정이 가능한 약 130개의 다양한 파일 형식을 지원	https://github.com/armijnhemel/binaryanalysis-ng
7	Barista	Apache-2.0	- 오픈 소스 구성 요소, 라이선스 및 잠재적인 취약점을 감지하는 스캐닝 도구 - 단단계 종속성을 포함하여 오픈 소스 BOM을 자동으로 생성하고 유지 관리 - Barista 관리자는 감지된 각 라이선스와 관련된 의무사항을 결정하고 배포 모델, 적용 가능한 라이선스 및 감지된 종속성에 대한 문서화된 취약성을 기반으로 프로젝트 승인 상태 할당 - 클라우드 네이티브 아키텍처: 필요에 따라 호스팅 유연성과 확장성을 허용하는 클라우드 네이티브 배포 환경용으로 설계	https://optum.github.io/barista/
8	Bubby	MPL-2.0	- 호환 SW를 안심하고 릴리즈할 수 있도록 지원하는 릴리즈 준비 플랫폼 - 보고 및 분석을 통해 릴리즈 프로세스에 대한 가시성을 확보하여 위험을 낮추고 품질을 높이며 주기 시간을 단축하고 지속적인 개선	https://github.com/valocode/bubby/
9	CLA Assistant	Apache-2.0	- 기여자가 CLA에 서명할 수 있도록 하여 리포지토리에 대한 기여의 법적 측면을 처리하는 도구 - CLA는 GitHub Gist 파일로 저장 후 CLA Assistant의 저장소/조직과 연결 가능	https://github.com/cla-assistant/cla-assistant
10	Cregit	GPL-3.0	- 소스코드의 기여자 식별	https://github.com/cregit/cregit

오픈소스 컴플라이언스 도구들(2)

No.	도구명	라이선스	특징	웹사이트
11	Deltacode	Apache-2.0	<ul style="list-style-type: none"> 패키지, 구성 요소, 코드베이스(제품)의 두 버전에 대한 ScanCode 스캔을 쉽게 비교가능하므로 라이선스 변경 사항 식별에 중점을 두고 가능한 변경 사항을 빠르게 식별 가능 ScanCode와 함께 DeltaCode를 사용하여 릴리스 간 오픈소스, 타사 SW 패키지, 구성요소의 라이선스 및 관련 변경 사항을 식별/추적 	https://www.aboutcode.org/
12	Eclipse SW360	EPL-1.0	<ul style="list-style-type: none"> 조직에서 사용하는 SW 구성 요소에 대한 정보 공유를 위한 중앙 위치를 제공하도록 설계된 SW 카탈로그 응용 프로그램 고유한 작업을 위한 별도의 백엔드 서비스 및 이의 액세스를 위한 포틀릿(재사용 가능한 웹 구성요소) 제공으로 SW 산출물 및 프로젝트 관리용 인프라에 쉽게 통합 코드 스캔 도구와 같은 외부 시스템과 상호 작용하는 커넥터 제공 	https://projects.eclipse.org/projects/technology.sw360
13	Eclipse SW360antenna	EPL-2.0	<ul style="list-style-type: none"> 오픈소스 라이선스 준수 프로세스를 최대한 자동화하는 도구 1) 모든 규정 준수 관련 데이터 수집 2) 해당 데이터를 처리하고 라이선스 준수 관련 문제가 있을 시 경고 3) 컴플라이언스 산출물 세트 생성(소스 코드 번들, 공개 문서, 보고서) 	https://projects.eclipse.org/projects/technology.sw360.antenna
14	Fossology	GPL-2.0	<ul style="list-style-type: none"> 라이선스, 저작권 및 export 통제 스캔을 위한 스캔 도구. SW의 모든 저작권 고지가 포함된 ReadMe 또는 SPDX 파일 생성 컴플라이언스 워크플로를 위한 웹 UI 및 DB 제공. 스캔 패키지를 서버에 업로드 필요 (Monk, Nomos 및 Ninka) 검사된 패키지에 대한 버전 제어가 있으므로 이전 패키지의 최신 버전을 검사할 때 변경된 파일만 다시 검사 	https://www.fossology.org/
15	LDBCollector	BSD-3-Clause	<ul style="list-style-type: none"> OSS 라이선스 메타데이터를 수집하고 결합하는 응용 프로그램 	https://github.com/maxhbr/LDBCollector
16	License Compatibility Checker	MIT	<ul style="list-style-type: none"> SPDX 표준 기반으로 라이선스 호환성을 위한 NPM package.json 종속성 확인 라이선스가 얼마나 허용되는지에 대한 설명 및 패키지의 라이선스를 간단히 비교 (허용 > 약한 보호 > 강력한 보호 > 네트워크 보호). 	https://github.com/HansHammel/license-compatibility-checker#readme
17	Licensee.js	Apache-2.0	<ul style="list-style-type: none"> 규칙에 대해 npm 패키지 종속성 라이선스 메타데이터를 확인하는 CLI SPDX 라이선스 표현과 화이트리스트 데이터를 사용하여 화이트리스트와 다른 라이선스에 있는 패키지를 캡처 	https://github.com/jslicense/licensee.js
18	Ninka	GPL-2.0	<ul style="list-style-type: none"> 소스 코드용 경량 라이선스 식별 도구 문장 기반이고 소스코드 파일에서 오픈 소스 라이선스를 식별하는 간단한 방법 제공 	http://ninka.turingmachine.org/
19	Opossum Tool	Apache-2.0	<ul style="list-style-type: none"> 오픈소스 라이선스 준수를 위해 대규모 코드베이스를 감사하고 인벤토리하는 경량 앱 다양한 소스의 오픈 소스 컴플라이언스 데이터를 관리하고 결합하기 위한 도구 대규모 코드베이스에 대한 규정 준수 정보를 검토하기 위한 경량 앱 OpossumUI: 애플리케이션에 사용되는 오픈 소스 소프트웨어 검색, 오픈소스 코드 스캔에서 보고서 생성 	https://github.com/opossum-tool
20	OSS Attribution Builder	Apache-2.0	<ul style="list-style-type: none"> SW 제품에 대한 속성 문서를 만드는 데 도움이 되는 웹 사이트 	https://github.com/amzn/oss-attribution-builder

오픈소스 컴플라이언스 도구들(3)

No.	도구명	라이선스	특징	웹사이트
21	OSS Discovery	GPL-3.0	- 응용 프로그램에 내장되고 컴퓨터에 설치된 오픈소스 SW 검색하는 스캐닝 도구 - 저작권자: OpenLogic	https://osdiscovery.sourceforge.net/
22	OSS Review Toolkit	Apache-2.0	- 소스 코드 및 종속성을 확인하여 오픈 소스 SW 라이선스 준수 여부 확인 - 종속성에 대한 소스 코드를 분석하고, 라이선스 정보에 대한 모든 소스 코드를 스캔하고, 결과를 요약하는 방식으로 작동 - ORT를 구성하는 다양한 도구는 최소한의 명령줄 인터페이스(스크립트 사용)가 있는 라이브러리로 설계	https://github.com/oss-review-toolkit/ort
23	OSSPolice	GPL-3.0	- 앱에서 잠재적인 자유 소프트웨어 라이선스 위반 및 알려진 n-day 보안 취약점 을 신속하게 식별가능한 위험 평가 서비스	https://github.com/ossanitizer/osspolice
24	Quartermaster Project QMSTR	GPL-3.0	- FOSS 규정 준수 문서를 작성 하고 규정 준수 결정을 지원하기 위해 SW 빌드를 계획하는 명령줄 도구 모음 및 빌드 시스템 확장 - 마스터 프로세스는 빌드 중인 소프트웨어에 대한 정보 수집. 빌드 완료되면 여러 분석 도구를 실행하고 보고 - 모든 모듈은 마스터 컨텍스트에서 실행. 마스터는 빌드 클라이언트 파일 시스템을 영향을 주지 않고 모듈의 모든 종속성을 제공	https://qmstr.org/
25	ScanCode.io & ScanPipe	Apache-2.0	- ScanCode.io는 SCA 프로세스를 스크립팅 하고 자동화하여 응용의 코드베이스에서 모든 오픈소스 구성 요소 및 해당 라이선스 준수 데이터를 식별 하는 서버 - ScanPipe는 SW 분석가와 엔지니어가 실제 SW 구성 분석 프로젝트를 스크립팅된 파이프라인으로 구축하고 관리하는 데 도움	https://scancodeio.readthedocs.io/en/latest/introduction.html#
26	ScanCode Toolkit	Apache-2.0	- 라이선스, 저작권, 패키지 매니페스트, 직접 종속성 , 소스/바이너리 파일에서 발견된 출처 및 라이선스 정보 를 스캔하는 CLI 모음 - JSON, HTML, CSV 또는 SPDX로 저장할 수 있는 스캔 결과 제공 - ScanCode는 코드 분석 파이프라인, CI/CD와 쉽게 통합	https://www.aboutcode.org/
27	SCANOSS	GPL-2.0-or-later	- 오픈 소스 SCA 플랫폼이자 개방형 데이터 OSS 지식 기반 - SPDX 및 CycloneDX에서 SBOM 생성 을 수행하고 스니펫, 파일 및 구성 요소 수준에서 오픈소스의 존재 감지 - 중심 구성 요소는 OpenAPI 표준을 기반으로 하는 RESTful API - 구성 요소, 파일 및 스니펫을 모든 도구에 일치 가능. 공용 지식 베이스 OSSKB는 osskb.org에서 사용 가능. 스캐닝은 익명 수행	https://www.scanoss.com/
28	SPDX Tools	Apache-2.0	- 단일 다운로드에서 변환, 비교 및 검증 기능 제공하는 자바 CLI 도구 - 제공기능: TagToSpreadsheet, TagToRDF, RdfToTag, RdfToHtml, RdfToSpreadsheet, SpreadsheetToRDF, SpreadsheetToTag, SPDXViewer, CompareMultipleSpdxDocs, CompareSpdxDocs, GenerateVerificationCode	https://spdx.dev/resources/tools/
29	SPDX Maven Plugin	Apache-2.0	- POM 파일에 설명된 산출물에 대한 SPDX 문서를 생성 하는 Maven 플러그인	https://github.com/spdx/spdx-maven-plugin
30	TraceCode toolkit	Apache-2.0	- 제품에 대해 실제로 배포 또는 배포되는 구성 요소를 결정 하는 도구 - 많은 라이선스가 배포에 의해서만 트리거되므로 오픈소스 라이선스 의무를 결정하는 필수 정보 - 추적된 빌드 실행을 분석하는 도구로, 어떤 파일이 바이너리에 빌드되고 궁극적으로 분산 SW에 배포되는지 확인 가능	https://www.aboutcode.org/

오픈소스 컴플라이언스 도구들(4)

No.	도구명	라이선스	특징	웹사이트
31	Tern	BSD-2-Clause	<ul style="list-style-type: none"> - 컨테이너용 SW 패키지 검사 도구(Python). 컨테이너 이미지에 설치된 패키지의 메타데이터 검색 - Dockerfile에서 레이어를 생성하는 데 사용된 컴포넌트 정보 제공 - 작업 수행 단계 <ol style="list-style-type: none"> 1) overlayfs를 사용하여 컨테이너 이미지의 첫 번째 파일 시스템 계층을 마운트 2) chroot 환경의 명령 라이브러리에서 스크립트를 실행하여 해당 계층에 설치된 패키지에 대한 정보 수집 : 해당 정보를 기반으로 컨테이너 이미지의 나머지 레이어에 대해 1단계와 2단계를 계속 반복 3) 완료되면 다양한 형식의 보고서 생성. 기본 보고서는 어떤 계층이 어떤 SW 구성 요소를 가져왔는지에 대한 설명 	https://github.com/tern-tools/tern
32	Vulnerability Assessment Tool	Apache-2.0	<ul style="list-style-type: none"> - 응용 개발 중 오픈소스 구성 요소의 안전한 사용과 관련하여 SW 개발 조직 지원 - Java 및 Python 응용 분석하여 알려진 취약성이 있는 오픈소스 구성 요소에 의존하는지 여부를 감지하고, 지정된 응용 컨텍스트에서 취약한 코드 실행에 관한 증거를 수집 지원 - 데이터 유출의 근본 원인인 알려진 취약성이 있는 구성 요소 사용, OWASP 상위 10대 보안위험 A9 해결 	https://github.com/eclipse/steady
33	REUSE		<ul style="list-style-type: none"> - 프로젝트 라이선스를 쉽게 부여할 수 있는 권장 사항들 제공 	https://reuse.software
34	FOSSLight	AGPL-3.0	<ul style="list-style-type: none"> - 소스 코드 내에 저작권 및 라이선스 규칙을 준수했는지 확인하고 또 저작권 및 라이선스 정보를 쉽게 추가할 수 있도록 도와주는 도구 	https://fossilight.org
35	ClearlyDefined	CC0-1.0, MIT	<ul style="list-style-type: none"> - ScanCode, FOSSology 통하여 사용가능한 데이터를 수집하고 모호함이나 격차가 발생할 때 해당 정보의 큐레이션을 크라우드 소싱 - 목표는 업스트림 프로젝트에 새로 발견된 명확성을 제공하여 다음 릴리스에 업데이트를 포함하여 보다 명확하게 정의되도록 하는 것 	https://clearlydefined.io

컴플라이언스 도구 정리 * 표 추가 정리 필요(유형 및 분류)

No.	도구 유형	관련 도구
1	License scanning	Fossology, LDBCollector, Ninka, SCANCode Toolkit
2	Binary scanning	Bang
3	Source code scanning	Deltacode, SCANOSS, (Protex, BlackDuck Professional, FOSSID, White source)
4	Component management	SW360, SW360Antenna, Opossum, OSS Discovery, SCANOSS, Tern, ClearlyDefined, FOSSLight
5	Dependency scanning	License Compatibility Checker, Licensee.js, ORT
6	Vulnerability scanning	Barista, OSSPolice, Vulnerability assessment tool
7	SBOM/Notice	AboutCode toolkit, CLA Assistant, Cregit, OSS Attribution builder, Quartermaster Project QMSTR, SPDX toolkit, SPDX Maven Plugin, REUSE
8	UI	AboutCode Manager
9	Release	Apache Rat, Apache Tentacles, Apache Whisker, Bubby, ScanCode.io & ScanPipe, TraceCode toolkit

ORT 개요

- **ORT(OSS Review Toolkit)**

- LF의 프로젝트 중 하나
- <https://github.com/oss-review-toolkit/ort>
- 오픈소스 소프트웨어 종속성에 대해 라이선스 준수 검사 및 관련 수행 작업을 지원
- 사용자 정의 가능한 파이프라인을 조정하여 CI/CD에 통합 가능

- **구성요소**

- **Analyzer**: 프로젝트 및 해당 메타데이터의 종속성을 결정하여 실제로 사용 중인 패키지 관리자 또는 빌드 시스템을 추상화
- **Downloader**: 소스코드를 검색하는 데 사용되는 VCS 등을 추상화하여 프로젝트 및 해당 종속성의 모든 소스코드 가져옴
- **Scanner**: 구성된 소스코드 스캐너를 사용하여 라이선스/저작권 결과를 감지하고 스캐너 유형을 추상화
- **Advisor**: 구성된 취약성 데이터 서비스에서 사용된 종속성에 대한 보안 권고 검색
- **Evaluator**: 맞춤형 정책 규칙 및 라이선스 분류에 대해 라이선스/저작권 결과 평가
- **Reporter**: 종속성, 라이선스, 저작권 또는 정책 규칙 위반 식별을 위한 시각적 보고서, 알림, BOM 등으로 결과 표시
- **Notifier**: 다양한 채널(예: 이메일 및/또는 JIRA 티켓) 통해 결과 알림

빌드 도구

* Olive, FOSSLight 사이트 참고

<Olive>

No.	패키지 매니저	설정 파일
1	Gradle	Build.gradle
2	Maven	Pom.xml
3	NPM	Package.json
4	Cocoapod	Podfile
5	Swift	Package.swift, Projejt.pbxproj
6	Carthage	Cartfile
7	Python	Setup.py, Requirement.txt
8	Ruby	Gemfile
9	Golang	Godeps.json, gopkg.toml, gopkg.lock, go.mod
10	Chef	Berkshelf
11	Cmake	CMakeLists.txt
12	Git Module	.gimodules
13	Android NDK	Android.mk

<FOSSLight>

No.	패키지 매니저	의존성 분석 도구
1	NPM	NPM License Checker
2	Pypi	Pip-licenses
3	Gradle	License Gradle Plugin
4	Maven	License-maven-plugin
5	Pub	Flutter_oss_licenses
6	Android(gradle)	Android-dependency-scanning
7	바이너리 파일 분석	Dependency-check-py

cf) LF의 관련 WG 및 프로젝트들

- **ACT(Automated Compliance Tooling)**

- 라이선스 준수, 보안, export 통제, 혈통 및 출처 워크플로를 가능하게 하는 SBOM의 효율적이고 효과적인 교환을 위한 오픈 소스 도구 개발 지원

- **TODO Group**

- 성공적이고 효과적인 OSPO 또는 유사한 OSS 이니셔티브를 실행하기 위해 지식을 생성, 공유하고 관행, 도구 및 기타 방법에 대해 협력하는 것을 목표로 하는 실무자의 개방형 커뮤니티

- **OpenChain Project**

- 기업/기관이 오픈소스 컴플라이언스를 위해 준수해야 할 활동을 더 간단하고 일관성있게 만들어 SW 공급망 전체에 신뢰를 구축할 수 있도록 지침 제공

- **도구 프로젝트**

- ORT, SPDX, Fossology

ORT의 특징

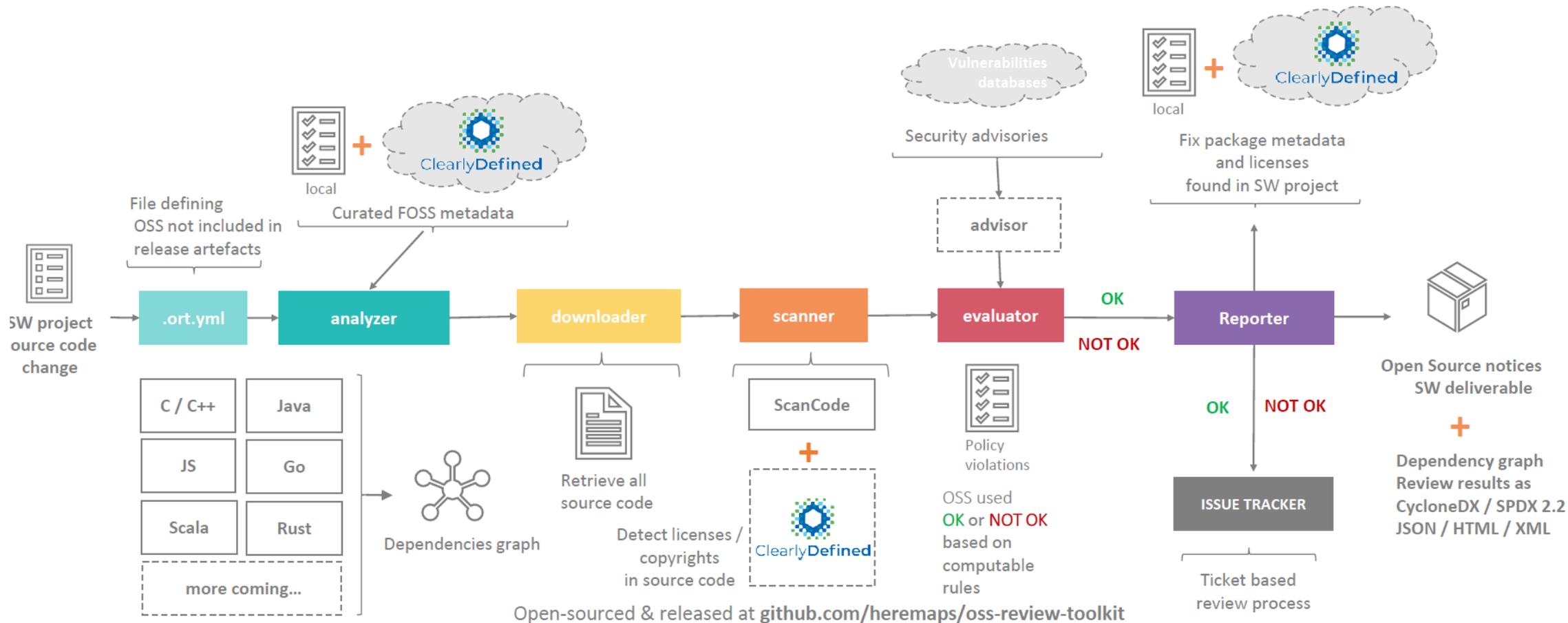
- **License scanning**
 - 기존 라이선스/저작권 스캐너(ScanCode 등)를 이용해 저작권/라이선스를 식별
- **Policy violations rule engine**
 - 스캐닝 결과에 대해 고객 맞춤형 정책 검사 실행
- **Software Bill of Materials / Notices**
 - CycloneDX, SPDX 2.2 파일 또는 오픈소스 고지문 생성
- **Source code scanning**
 - 소스코드나 다른 파일의 출처를 식별하기 위한 통합 작업을 개발하기 위해 벤더와 협업
- **Security scanning**
 - 다양한 벤더로부터 들어오는 OSS 보안취약성 데이터 통합
- **Dev Ops integration**
 - CI/CD를 위해 시작부터 설계

수집되는 정보

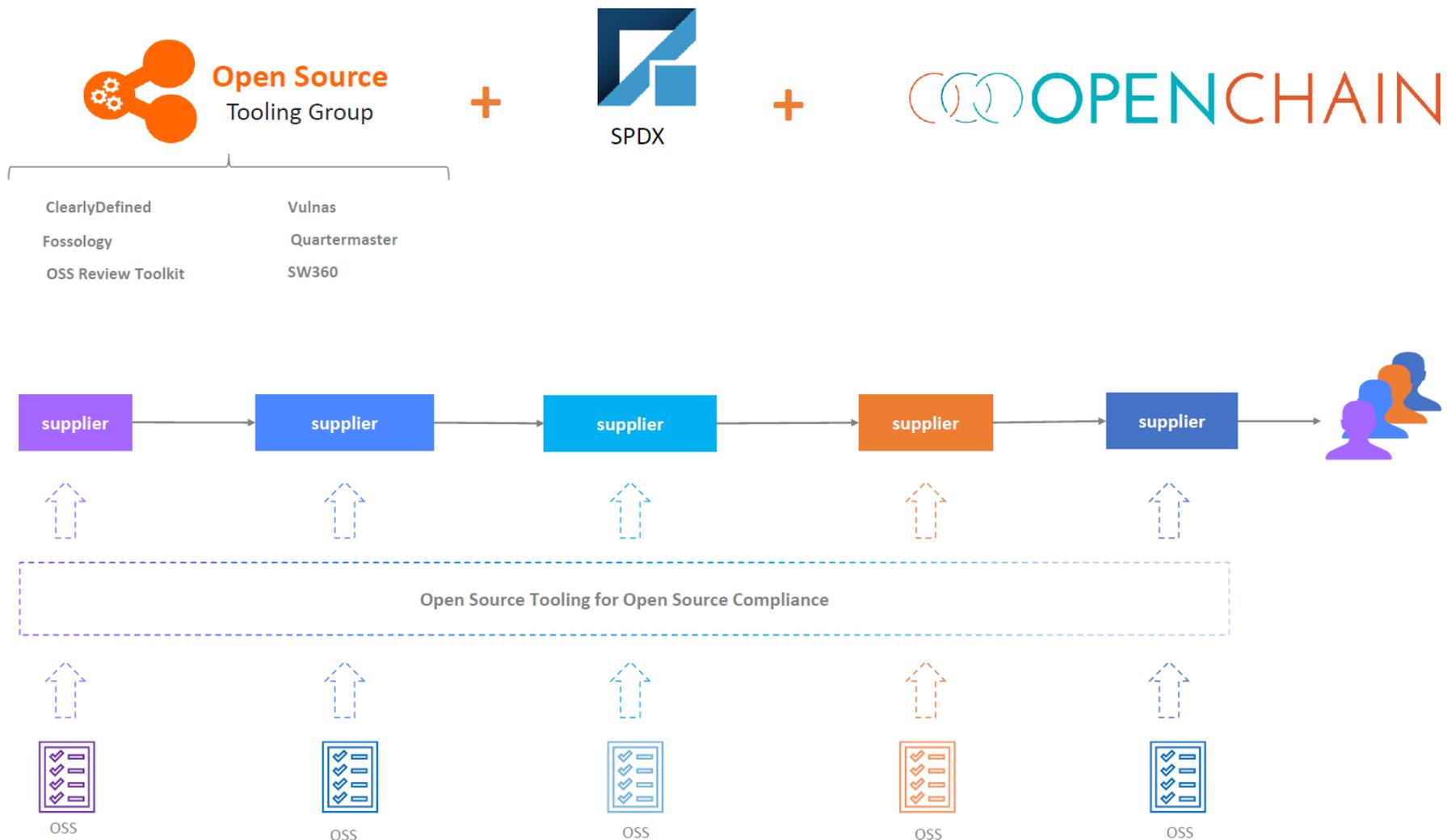
- 패키지명
- 버전
- 소스코드 저장소 URL
- 소스코드 및 바이너리 산출물
- 저작권자
- 라이선스 및 URL
- 귀속 및 다른 공지 정보 URL
- 의존성 리스트 및 트리

ORT 구조

* Jenkins, TEKTON (Google Kubernetes Engine(GKE))

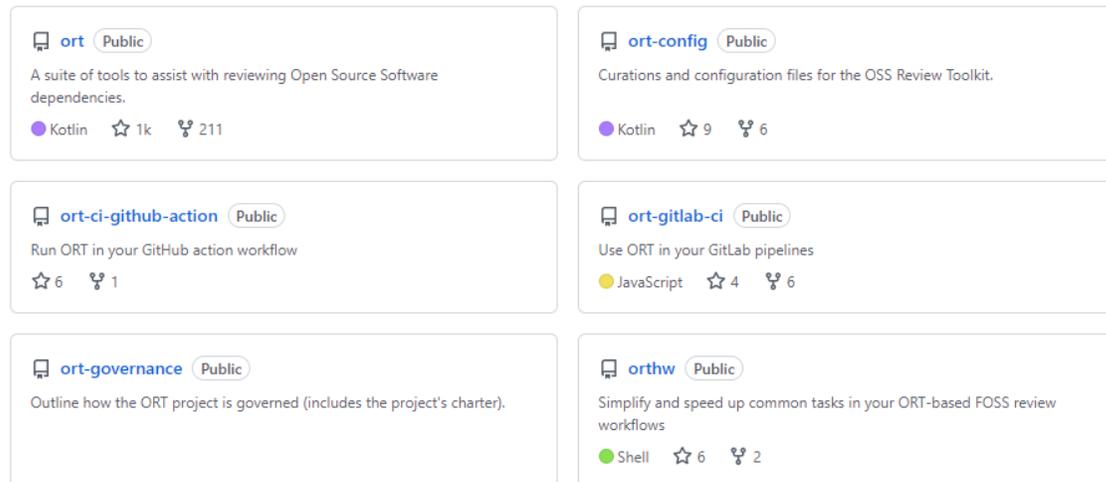


SW 공급망 관리 구조



OSS Review Toolkits의 구성 요소

No.	도구	기능
1	ORT	오픈 소스 소프트웨어 종속성에 대해 라이선스 준수 검사 및 관련 수행 작업 지원
2	ORT Config Repository	ORT 구성 파일이 있는 저장소 예시
3	ORT Workbench	ORT 결과 파일 뷰어. 보고서 생성 대신 사용
4	ORT GitHub Action	GitHub 리포지토리에 대해 ORT를 실행하는 GitHub 작업
5	ORT GitLab Pipeline	GitLab 리포지토리에 대해 ORT를 수행하는 GitLab 파이프라인
6	ORTHW	ORT 결과를 처리할 때 수행되는 작업들을 단순화하고 속도를 높이는데 도움이 되는 bash 스크립트



The screenshot displays six GitHub repository cards for OSS Review Toolkit components:

- ort** (Public): A suite of tools to assist with reviewing Open Source Software dependencies. 1k stars, 211 forks.
- ort-config** (Public): Curations and configuration files for the OSS Review Toolkit. 9 stars, 6 forks.
- ort-ci-github-action** (Public): Run ORT in your GitHub action workflow. 6 stars, 1 fork.
- ort-gitlab-ci** (Public): Use ORT in your GitLab pipelines. 4 stars, 6 forks.
- ort-governance** (Public): Outline how the ORT project is governed (includes the project's charter).
- orthw** (Public): Simplify and speed up common tasks in your ORT-based FOSS review workflows. 6 stars, 2 forks.

설치 방법

출처: <https://lge-oss.github.io/oss-review-toolkit-guide/>

• 바이너리로부터 설치

- JitPack 활용
- Web app 리포트는 동작 안함
- 사용법
 - ✓ `curl -o ort.jar https://jitpack.io/com/github/oss-review-toolkit/ort/cli/7aa61bf96b/cli-7aa61bf96b-all.jar`

• 소스코드로부터 설치

- 사전 설치 요구사항
 - ✓ Git
 - ❖ `Sudo apt-get install git`
- Docker를 사용한 설치
 - ✓ Docker 설치
 - ❖ `Sudo apt-get install docker`
 - ✓ 도커를 위한 buildkit 실행
- Build natively
 - ✓ JDK 11 이상. JAVA_HOME 환경변수 설정 필요
 - ❖ `Sudo apt-get install openjdk-11-jdk`
 - ✓ `./gradlew installDist`

```

root@dchecker:/home/jungsp/ort# ls
ADOPTERS.md  README.md  batect.yml  detekt-rules  examples  helper-cli  notifier  scanner
Dockerfile  advisor    build       docker         gradle    integrations  qodana.yml  scripts
LICENSE      analyzer  build.gradle.kts  docs          gradle.properties  logos       renovate.json  settings.gradle.kts
LICENSES     batect    cli         downloader     gradlew   mlc_config.json  reporter     utils
NOTICE       batect.cmd  clients    evaluator      gradlew.bat  model         reporter-web-app

root@dchecker:/home/jungsp/ort#

```

설정

- 환경 변수

- ort/model/build/resources/main/reference.yml을 ~/.ort/config/config.yml로 복사

- 설정 파일

- ORT configuration file
 - Copyright garbage file
 - Curations file
 - Custom license texts dir
 - How to fix text provider script
 - License classifications file
 - Resolution file
 - Repository configuration file
 - Package configuration file / directory
 - Policy rules file

사용법(1): Analyzer

- 지정된 입력 디렉토리 내에서 SW 프로젝트의 종속성을 결정하는 SCA 도구
- 감지된 패키지 관리자를 쿼리하여 수행
- 모든 패키지 관련 메타데이터의 상태를 문서화함
- 다른 도구로 전달하기 전에 추가 처리나 수동 편집 가능
- **사용법**
 - `./ort analyze -i [working-directory] -o [analyze-output-dir]`
- **결과**
 - `analyzer-result.yml`
 - 프로젝트의 의존성 정보와 메타데이터 출력

ORT에서 지원 중인 패키지 매니저

- C / C++
 - [Conan](#)
 - Also see: [SPDX documents](#)
- Dart / Flutter
 - [Pub](#)
- Go
 - [dep](#)
 - [Glide](#)
 - [Godep](#)
 - [GoMod](#)
- Haskell
 - [Stack](#)
- Java
 - [Gradle](#)
 - [Maven](#) (limitations: [default profile only](#))
- JavaScript / Node.js
 - [Bower](#)
 - [NPM](#) (limitations: [no peer dependencies](#))
 - [PNPM](#) (limitations: [no peer dependencies](#))
 - [Yarn 1](#)
 - [Yarn 2+](#)
- .NET
 - [DotNet](#) (limitations: [no floating versions / ranges, no target framework](#))
 - [NuGet](#) (limitations: [no floating versions / ranges, no target framework](#))
- Objective-C / Swift
 - [Carthage](#) (limitation: [no cartfile.private](#))
 - [CocoaPods](#) (limitations: [no custom source repositories](#))
- PHP
 - [Composer](#)
- Python
 - [PIP](#)
 - [Pipenv](#)
 - [Poetry](#)
- Ruby
 - [Bundler](#) (limitations: [restricted to the version available on the host](#))
- Rust
 - [Cargo](#)
- Scala
 - [SBT](#)
- Unmanaged
 - This is a special "package manager" that manages all files that cannot be associated to any of the other package managers.

실행 결과

```
(venv) root@dchecker:/home/jungsp/ort/cli/build/install/ort/bin# ls test
analyzer-result.yml  scan-result.yml
```

Key	Description
repository > vcs	Version Control System 기반 추출된 정보를 표시합니다.
repository > vcs_processed	Version Control System URL 또는 Fallback URL에서 추출된 정보를 표시합니다.
analyzer > result	dependency 관련 추출된 정보를 표시합니다.
analyzer > result > projects	상위 project의 정보가 출력됩니다.
analyzer > result > packages	Project별 dependency에 따른 package 정보가 출력됩니다.

```
root@dchecker:/home/jungsp/ort-src/ort# find . -name "pom.xml" -print
./analyzer/src/funTest/assets/projects/synthetic/maven-parent/pom.xml
./analyzer/src/funTest/assets/projects/synthetic/maven/pom.xml
./analyzer/src/funTest/assets/projects/synthetic/maven/app/pom.xml
./analyzer/src/funTest/assets/projects/synthetic/maven/lib/pom.xml
./analyzer/src/funTest/assets/projects/synthetic/maven-wagon/pom.xml
root@dchecker:/home/jungsp/ort-src/ort# find . -name "build.gradle" -print
./analyzer/src/funTest/assets/projects/synthetic/gradle-android/build.gradle
./analyzer/src/funTest/assets/projects/synthetic/gradle-android/app/build.gradle
./analyzer/src/funTest/assets/projects/synthetic/gradle-android/lib/build.gradle
./analyzer/src/funTest/assets/projects/synthetic/gradle-android-cyclic/build.gradle
./analyzer/src/funTest/assets/projects/synthetic/gradle-android-cyclic/app/build.gradle
./analyzer/src/funTest/assets/projects/synthetic/gradle-android-cyclic/lib/build.gradle
./analyzer/src/funTest/assets/projects/synthetic/gradle-library/build.gradle
./analyzer/src/funTest/assets/projects/synthetic/gradle-library/app/build.gradle
./analyzer/src/funTest/assets/projects/synthetic/gradle-library/lib/build.gradle
./analyzer/src/funTest/assets/projects/synthetic/pub/flutter-project-with-android-and-cocoapods/android/build.gradle
./analyzer/src/funTest/assets/projects/synthetic/pub/flutter-project-with-android-and-cocoapods/android/app/build.gradle
./analyzer/src/funTest/assets/projects/synthetic/gradle/build.gradle
./analyzer/src/funTest/assets/projects/synthetic/gradle/app/build.gradle
./analyzer/src/funTest/assets/projects/synthetic/gradle/lib/build.gradle
./analyzer/src/funTest/assets/projects/synthetic/gradle/lib-without-repo/build.gradle
./analyzer/src/funTest/assets/projects/synthetic/gradle-unsupported-version/build.gradle
./analyzer/src/funTest/assets/projects/synthetic/gradle-bom/build.gradle
root@dchecker:/home/jungsp/ort-src/ort#
```

```
- metadata:
  id: "Maven:org.apache.logging.log4j:log4j-to-slf4j:2.19.0"
  purl: "pkg:maven/org.apache.logging.log4j/log4j-to-slf4j@2.19.0"
  authors:
  - "Apple"
  - "Bruce Brouwer"
  - "Carter Kozak"
  - "Nextiva"
  - "Nick Williams"
  - "Piotr P. Karwasz"
  - "Remko Popma"
  - "Ron Grabowski"
  - "Scott Deboy"
  - "Spotify"
  - "The Apache Software Foundation"
  - "Volkan Yazıcı"
  declared_licenses:
  - "Apache License, Version 2.0"
  declared_licenses_processed:
    spdx_expression: "Apache-2.0"
    mapped:
      Apache License, Version 2.0: "Apache-2.0"
  description: "The Apache Log4j binding between Log4j 2 API and SLF4J."
  homepage_url: "https://logging.apache.org/log4j/2.x/log4j-to-slf4j/"
  binary_artifact:
    url: "https://repo.maven.apache.org/maven2/org/apache/logging/log4j/"
    hash:
      value: "30f4812e43172ecca5041da2cb6b965cc477c19"
      algorithm: "SHA-1"
  source_artifact:
    url: "https://repo.maven.apache.org/maven2/org/apache/logging/log4j/"
    hash:
      value: "6da7f8160a76c69a3f904272e31509d6ae45881e"
      algorithm: "SHA-1"
  vcs:
    type: "Git"
    url: "https://gitbox.apache.org/repos/asf/logging-log4j2.git"
    revision: "log4j-2.19.0-rc2"
    path: ""
  vcs_processed:
    type: "Git"
    url: "https://gitbox.apache.org/repos/asf/logging-log4j2.git"
    revision: "log4j-2.19.0-rc2"
    path: ""
  curations: []
```

사용법(2): Downloader

- 분석기 결과가 포함된 ORT 결과 파일을 입력(-i)으로 사용하면 downloader 는 포함된 모든 패키지의 소스코드를 지정된 출력 디렉토리로 검색
- 지원되는 버전 도구
 - CVS
 - Git
 - Git-Repo
 - Mercurial
 - Subversion

사용법(3): Scanner

- 분석기 결과와 함께 ORT 결과 파일을 전달하면 scanner는 자동으로 다운로더를 통해 종속성의 소스코드를 다운로드하여 스캔
- 기본 라이선스/저작권 스캐너를 공통 API로 wrapping
- 지원 도구
 - FOSSID
 - ScanCode
- 참조 구현
 - Askalono, Ic, Licensee, SCANOSS
- 사용법
 - `./ort scan -i [analyzer-output-file] -o [scanner-output-dir]`
- 결과
 - Scan-result.yml
 - 의존성 있는 패키지를 다운로드 후 라이선스 텍스트 분석 결과와 copyright 텍스트 출력

실행 결과

Output 정보

File or Directory	Description
downloads	Analyzer 결과에 따라 Dependency가 있는 Package를 다운로드 받은 폴더
native-scan-results	각 Package별 Scanner 분석 결과 파일이 위치한 폴더
scan-result.yml	Analyzer 결과에 Scanner 분석 결과가 추가된 파일. Scanner 분석 결과에는 검출된 License, Copyright text가 포함됩니다.

```

- id: "Gradle:oss-review-toolkit:detekt-rules:de9cf819e3"
  definition_file_path: "detekt-rules/build.gradle.kts"
  declared_licenses: []
  declared_licenses_processed: {}
  vcs:
    type: ""
    url: ""
    revision: ""
    path: ""
  vcs_processed:
    type: "Git"
    url: "https://github.com/oss-review-toolkit/ort.git"
    revision: "de9cf819e3b2277ebf6b93d513cf050ce80e5d58"
    path: "detekt-rules"
  homepage_url: ""
  scope_names:
  - "compileClasspath"
  - "compileOnlyDependenciesMetadata"
  - "detekt"
  - "detektPlugins"
  - "dokkaGfmPartialPlugin"
  - "dokkaGfmPartialRuntime"
  - "dokkaGfmPlugin"
  - "dokkaGfmRuntime"
  - "dokkaHtmlPartialPlugin"
  - "dokkaHtmlPartialRuntime"
  - "dokkaHtmlPlugin"
  - "dokkaHtmlRuntime"
  - "dokkaJavadocPartialPlugin"
  - "dokkaJavadocPartialRuntime"
  - "dokkaJavadocPlugin"
  - "dokkaJavadocRuntime"
  - "dokkaJekyllPartialPlugin"
  - "dokkaJekyllPartialRuntime"
  - "dokkaJekyllPlugin"
  - "dokkaJekyllRuntime"
  - "funTestCompileClasspath"
  - "funTestImplementationDependenciesMetadata"
  - "funTestRuntimeClasspath"
  - "implementationDependenciesMetadata"
  - "jacocoAgent"
  - "jacocoAnt"
  - "kotlinCompilerClasspath"
  - "kotlinCompilerPluginClasspathFunTest"
  - "kotlinCompilerPluginClasspathMain"
  - "kotlinCompilerPluginClasspathTest"
  - "kotlinKlibCommonizerClasspath"
  - "runtimeClasspath"
  - "testCompileClasspath"
  - "testImplementationDependenciesMetadata"
  - "testRuntimeClasspath"

```

사용법(4): Advisor

- 구성된 서비스에서 보안 권고 검색
- 분석기가 식별한 모든 패키지에 대해 알려진 보안취약점에 대해 정보 수집
- 공급자
 - NexusIQ
 - OSS Index
 - VulnerableCode
 - Google OSV
- 사용법
 - `./ort -c [ort.conf] advise -i [analyzer-output-file] -o [advisor-output-dir]`

사용법(5): Evaluator

- 스캔 결과에 대한 사용자 정의 라이선스 정책 확인을 수행하는데 사용
- 확인할 규칙은 전용 DSL이 있는 Kotlin 스크립트로 구현
- 예) `example.rules.kts`
- **사용법**
 - `./ort evaluate -package-curation-file [examples/curations.yml] -rules-file [examples/rules.kts] --license-configuration-file [examples/license-classifications.yml] -i [analyze-output-file] -o [evaluator-output-dir]`
- **결과**
 - `Evaluation-result.yml`
 - 라이선스별 카테고리 정의 파일과 분석할 규칙을 정의한 파일을 기반으로 이슈 출력

사용법(6): Reporter

- **ORT 결과 파일에서 다양한 형식의 문서 생성**
- **지원 포맷**
 - AsciiDoc Template
 - ctrlX AUTOMATION
 - CycloneDX BOM
 - Excel sheet
 - FOSSID report
 - GitLabLicenseModel
 - NOTICE file
 - Opossum input
 - SPDX document
 - Static HTML
 - Web App
- **사용법**
 - `./ort report -f NoticeTemplate,StaticHtml,WebApp -i [evaluator-output-file] -o [reporter-output-dir]`
- **결과**
 - NOTICE_default
 - Scan-report-web-app.html
 - Scan-report.html

CI 및 시각화 도구와 연동

- Jenkins와 연동
- Dependency graph
 - [dependency-graph · GitHub Topics · GitHub](#)

Dependency graph(1)

출처: <https://github.com/topics/dependency-graph>
<https://github.com/topics/dependencies>

No.	도구명	기능	다운로드
1	Dependency-cruiser	종속성을 검증하고 시각화(JavaScript, Typescript, Coffeescript, ES6, CommonJS, AMD)	
2	Pipdeptree	Python 패키지의 종속성 트리를 표시하는 명령어 유틸리티	
3	Objc-dependency-visualizer	Objective-C 및 Swift의 종속성 가상화. 프로젝트의 현재 상태 및 클래스 결합도 표현	
4	Dephell	Python 프로젝트 관리. 패키지들 관리(format, lock, install, resolve, isolate, test, build graph, outdated, audit)	
5	scancode-toolkit	스캐닝 코드로 라이선스, 저작권, 종속성을 감지해 코드에 사용된 오픈소스, 타사 패키지를 검색하고 목록화	
6	statoscope	웹팩 번들을 분석하고 검증	
7	scabbard	Dagger 2 의존성 그래프 시각화	
8	Do	Go 1.18+ Generics 기반 의존성 주입 툴킷	
9	Ngd	Angular 응용의 의존성 트리 시각화	
10	phpDependencyAnalysis	의존성 그래프에서 위법 사항 탐색하는 정적 코드 분석	

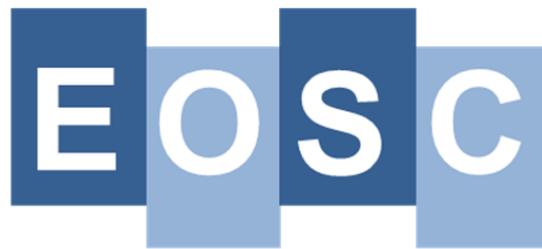
Dependency graph(2)

No.	도구명	기능	다운로드
11	Sidekiq-superworker	Sidekiq job들의 directed acyclic 그래프	
12	emerge	SW 프로젝트의 소스 코드 구조, 메트릭, 종속성 및 복잡성에 대한 통찰력을 수집하는 데 사용할 수 있는 소스 코드 및 종속성 시각화 도우미 소스코드를 스캔한 후 그래프 구조를 사용하여 프로젝트를 탐색/분석가능한 대화형 웹 인터페이스 제공	
13	cmake-scripts	코드 커버리지, sanitizer, 의존성 그래프 생성을 포함한 Cmake에서 사용되는 스크립트	
14	Dg	LLVM 비트코드의 의존성 그래프 및 프로그램 슬라이싱의 프로그램 분석 및 구성	
15	Swift-code-metrics	Swift 프로젝트를 위한 코드 메트릭 분석기	
16	Nix-tree	Nix 파생물의 종속성 그래프를 대화식으로 탐색	
17	protodot	.proto 파일을 .dot 파일로 변환	
18	Kotlin-MVVM-Architecture	Kotlin, MVVM, Hilt, LiveData, Room, MediatorLiveData, NetworkBoundResources, Retrofit, AndroidX, ViewModels, Dagger2를 사용한 종속성 주입, 리포지토리 패턴을 사용한 Android 아키텍처 디자인 패턴	
19	Cargo-supply-chain	종속성 그래프에서 crate에 대한 저자, 기여자 및 게시자 데이터를 수집	
20	ORT	오픈소스 SW 의존성 검토를 지원하는 도구 모음	

토의

- 의존성 검사의 depth 지원 방법
- Dependency graph 도구 논의
 - 각 도구가 특정 상황에 적용
- 의존성 검사 결과의 SBOM 출력

감사합니다



ETRI 오픈소스센터
Open Source Center