OSS Review Toolkit: Automating Open Source Compliance within CI/CD

Sebastian Schuberth, HERE Technologies

FOSS Backstage, 2018-Jun-14

About me

Who is this guy anyway?

Sebastian Schuberth, HERE Technologies, 6+ years

- Head of Open Source engineering
- Active Open Source contributor
- Background in mobile development and computer graphics

Favorite technologies (currently)

- Kotlin
- Gradle
- Git

Hobbies

- Coding, coding, coding
- Offroad RC car racing

About me

Who is this guy anyway?

Sebastian Schuberth, HERE Technologies, 6+ years

- Head of Open Source engineering
- Active Open Source contributor
- Background in mobile development and computer graphics

Favorite technologies (currently)

- Kotlin
- Gradle
- Git

Hobbies

- Coding, coding, coding
- Offroad RC car racing

About me

Who is this guy anyway?

Sebastian Schuberth, HERE Technologies, 6+ years

- Head of Open Source engineering
- Active Open Source contributor
- Background in mobile development and computer graphics

Favorite technologies (currently)

- Kotlin
- Gradle
- Git

Hobbies

- Coding, coding, coding
- Offroad RC car racing

About me

Who is this guy anyway?

Sebastian Schuberth, HERE Technologies, 6+ years

- Head of Open Source engineering
- Active Open Source contributor
- Background in mobile development and computer graphics

Favorite technologies (currently)

- Kotlin
- Gradle
- Git

Hobbies

- Coding, coding, coding
- Offroad RC car racing

Topics

This is going to be rather technical!

The problem

- What issue are we trying to solve?

Our requirements

- What do we need the tool to be able to do?
- Or: Why are we actually doing this?

OSS Review Toolkit (ORT)

- Overview of tools in the suite
- Running in CI at the example of Jenkins
- Roadmap

The problem

What issue are we trying to solve?

Review own products for license compliance

- Identify license incompatibilities in transitive dependency tree
- Ensure to follow license obligations / create NOTICE file
- Not necessarily limited to FOSS dependencies

Side benefits

- Overview of FOSS / technologies used in the company
- Identify “problematic” / not well maintained software packages
- Enable security vulnerability reporting
- Enforce best engineering practices (WRT the build system)

Our requirements

What do we need the tool to be able to do?

Inspect projects from the outside

- No changes to the project to analyze must be required
- Except if it does not follow best engineering practices (like if the build is not self-contained)

Support common package managers

- Capture meta-data (declared license etc.)
- Determine the *real* version of dependencies used
- Retrieve and scan the source code (must not rely on declared license)
- Allow to fixup broken meta-data (and allow to contribute it back upstream, e.g. via ClearlyDefined)
- Support mixed projects or multi-module projects

Support “unmanaged” projects (as good as possible)

- C/C++ projects, Makefile-based, embedded Linux

Our requirements, part 2

What do we need the tool also to be able to do?

Use standardized interchange formats

- Software Package Data Exchange (SPDX)
- AboutCode Data (ABCD)

Bring our own scanner (BYOS)

- Do not reinvent the wheel
- Use the license / copyright scanner that works best for *you*
- No vendor lock-in

Fast incremental scans

- Reuse existing results
- Delta-scans

Our requirements, part 3

What do we need the tool yet also to be able to do?

Customizable license compliance rules

- Apache-2.0 vs. GPL-3.0
- Take dependency scopes into account

Multiple result formats

- Graphical representation of dependency tree
- Legal people love Excel
- Need to generate NOTICE files

Reasonably easy to setup

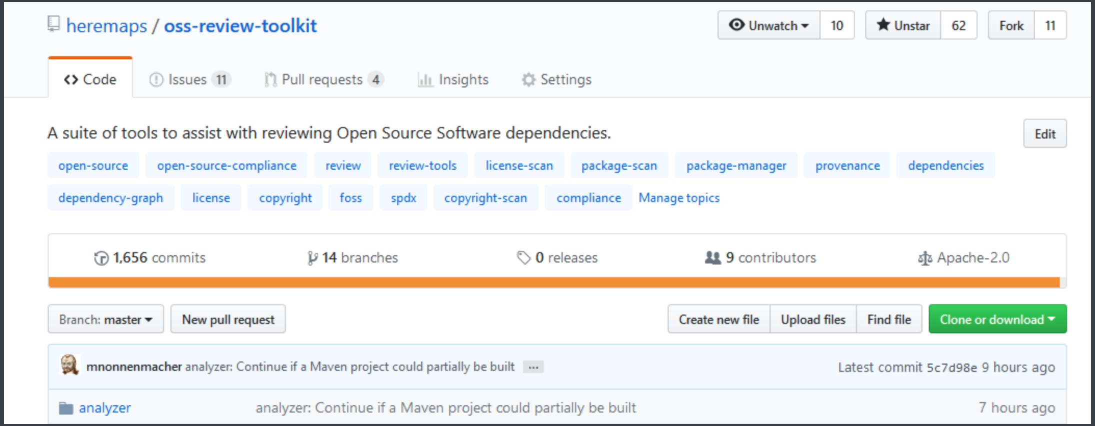
- Runs locally as well as on CI/CD

OSS Review Toolkit (ORT)

A suite of tools to assist with license reviews

Facts

- Open Source, Apache-2.0 licensed
- Written in Kotlin
- Libraries with a “Main“ entry point each
- In production use for 6 months



heremaps / oss-review-toolkit

A suite of tools to assist with reviewing Open Source Software dependencies.

open-source open-source-compliance review review-tools license-scan package-scan package-manager provenance dependencies dependency-graph license copyright foss spdx copyright-scan compliance Manage topics

1,656 commits | 14 branches | 0 releases | 9 contributors | Apache-2.0

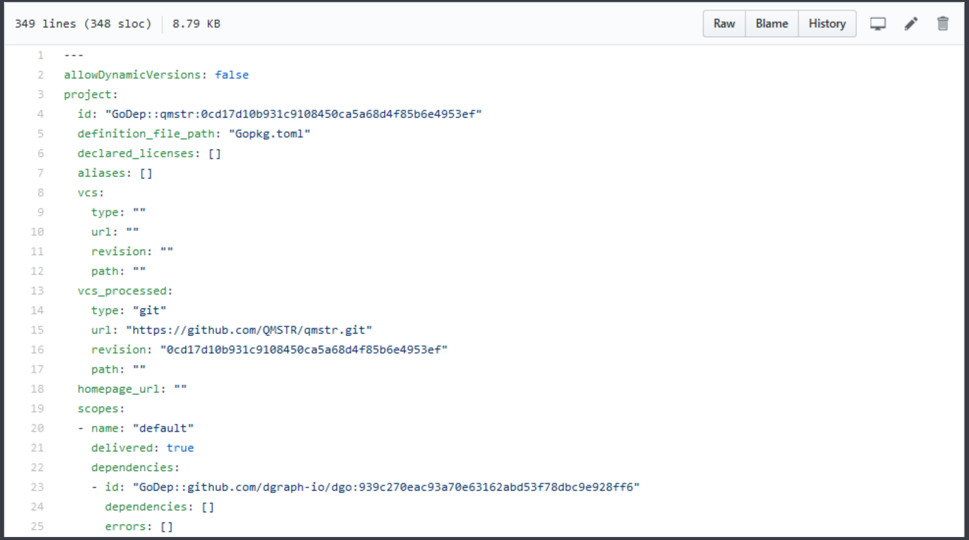
mnonnenmacher analyzer: Continue if a Maven project could partially be built ... Latest commit 5c7d98e 9 hours ago

analyzer analyzer: Continue if a Maven project could partially be built 7 hours ago

OSS Review Toolkit (ORT)

The Analyzer (OpenChain ”Identification” step)

- Input: Local directory with source code, and optional curations
- Action: Gather data about software dependencies (currently 11 supported package managers)
- Output: YAML / JSON file with dependency tree and meta-data about packages



```
---
allowDynamicVersions: false
project:
  id: "GoDep::qmstr:0cd17d10b931c9108450ca5a68d4f85b6e4953ef"
  definition_file_path: "Gopkg.toml"
  declared_licenses: []
  aliases: []
  vcs:
    type: ""
    url: ""
    revision: ""
    path: ""
  vcs_processed:
    type: "git"
    url: "https://github.com/QMSTR/qmstr.git"
    revision: "0cd17d10b931c9108450ca5a68d4f85b6e4953ef"
    path: ""
  homepage_url: ""
  scopes:
  - name: "default"
    delivered: true
    dependencies:
    - id: "GoDep::github.com/dgraph-io/dgo:939c270eac93a70e63162abd53f78dbc9e928ff6"
      dependencies: []
      errors: []
```

OSS Review Toolkit (ORT)

The Downloader

- Input: Analyzer file
- Action: Fetch source code (Git, Mercurial, Subversion, CVS, HTTP)
- Output: Local directories with source code

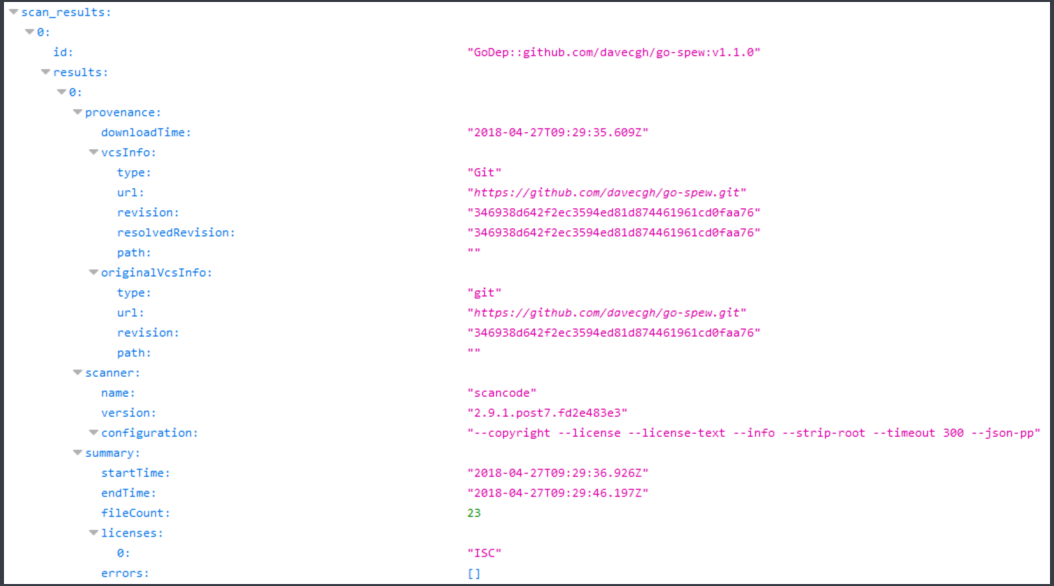
Hints

- Intermediate tool used internally by Scanner
- Can also be (mis-)used to download source code before Analyzer

OSS Review Toolkit (ORT)

The Scanner (OpenChain ”Audit” step)

- Input: Analyzer file, or local directory
- Action: Run configured license scanner (currently 4 supported scanners)
- Output: YAML / JSON file with scan results

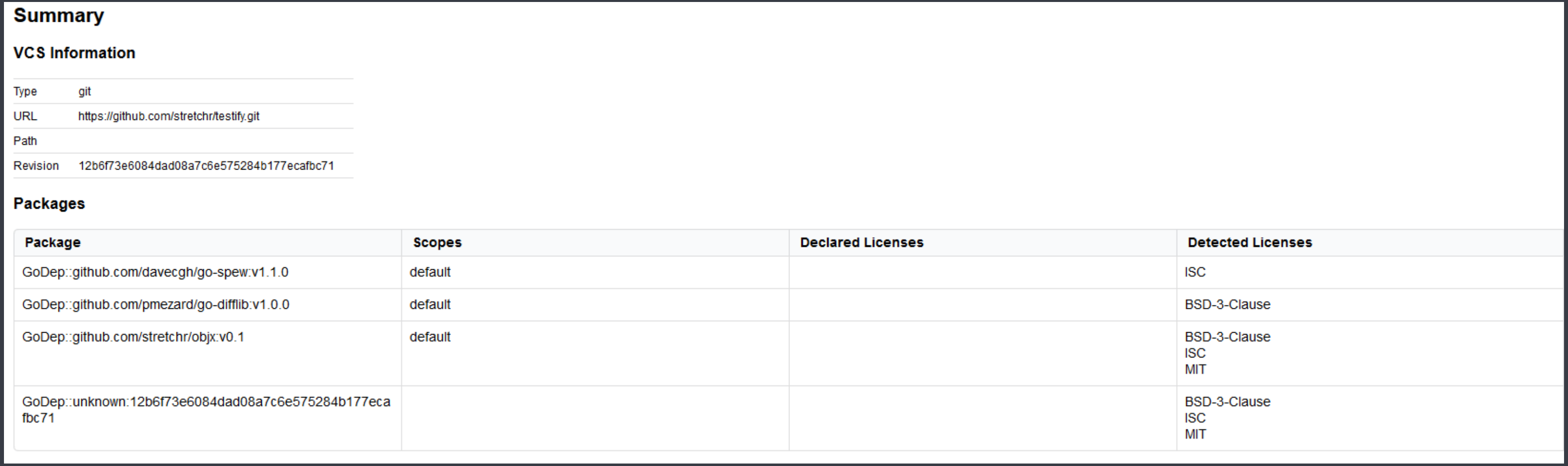


```
▼scan_results:
  ▼0:
      id:                                      "GoDep::github.com/davecgh/go-spew:v1.1.0"
    ▼results:
      ▼0:
        ▼provenance:
            downloadTime:                      "2018-04-27T09:29:35.609Z"
          ▼vcsInfo:
              type:                            "Git"
              url:                             "https://github.com/davecgh/go-spew.git"
              revision:                        "346938d642f2ec3594ed81d874461961cd0faa76"
              resolvedRevision:                "346938d642f2ec3594ed81d874461961cd0faa76"
              path:                            ""
          ▼originalVcsInfo:
              type:                            "git"
              url:                             "https://github.com/davecgh/go-spew.git"
              revision:                        "346938d642f2ec3594ed81d874461961cd0faa76"
              path:                            ""
        ▼scanner:
            name:                              "scancode"
            version:                           "2.9.1.post7.fd2e483e3"
          ▼configuration:                      "--copyright --license --license-text --info --strip-root --timeout 300 --json-pp"
        ▼summary:
            startTime:                         "2018-04-27T09:29:36.926Z"
            endTime:                           "2018-04-27T09:29:46.197Z"
            fileCount:                         23
          ▼licenses:
              0:                               "ISC"
            errors:                            []
```

OSS Review Toolkit (ORT)

The Reporter

- Input: Scanner file
- Action: Generate a custom report / visualization
- Output: A report file in some format



Summary

VCS Information

Type	git
URL	https://github.com/stretchr/testify.git
Path	
Revision	12b6f73e6084dad08a7c6e575284b177ecafbc71

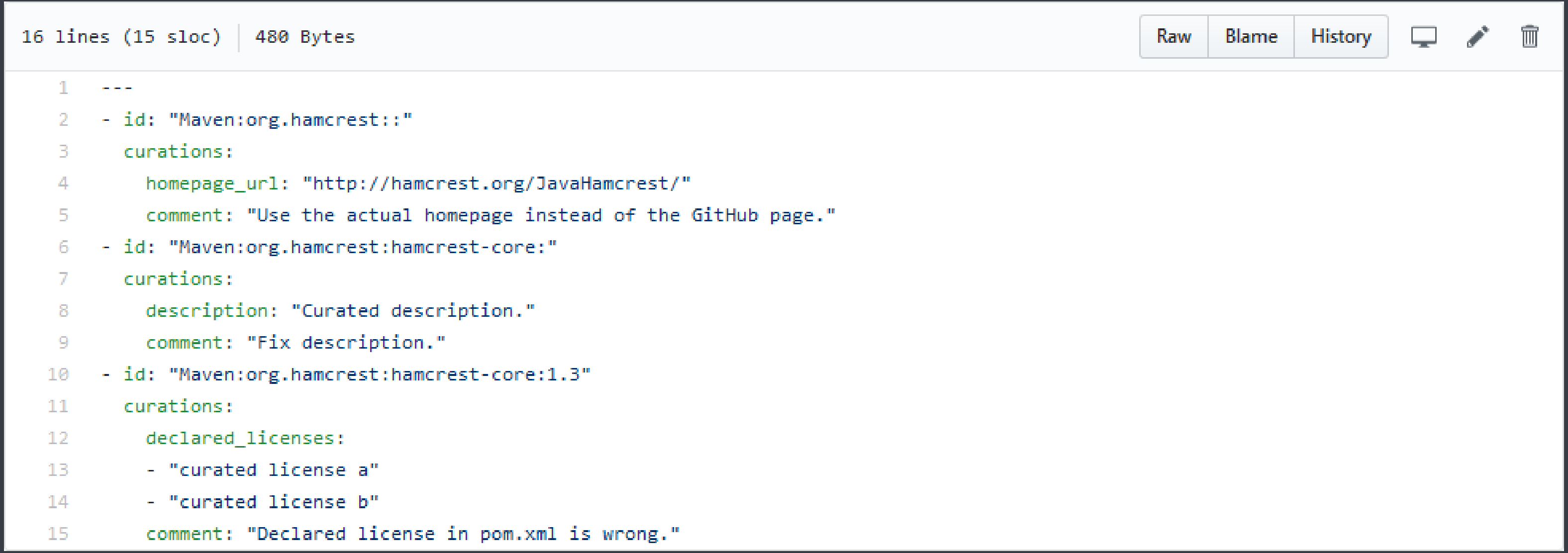
Packages

Package	Scopes	Declared Licenses	Detected Licenses
GoDep::github.com/davecgh/go-spew:v1.1.0	default		ISC
GoDep::github.com/pmezard/go-difflib:v1.0.0	default		BSD-3-Clause
GoDep::github.com/stretchr/objx:v0.1	default		BSD-3-Clause ISC MIT
GoDep::unknown:12b6f73e6084dad08a7c6e575284b177ecafbc71			BSD-3-Clause ISC MIT

OSS Review Toolkit (ORT)

Curations

- YAML / JSON file to “augment“ a package‘s meta-data
- To be shared with ClearlyDefined

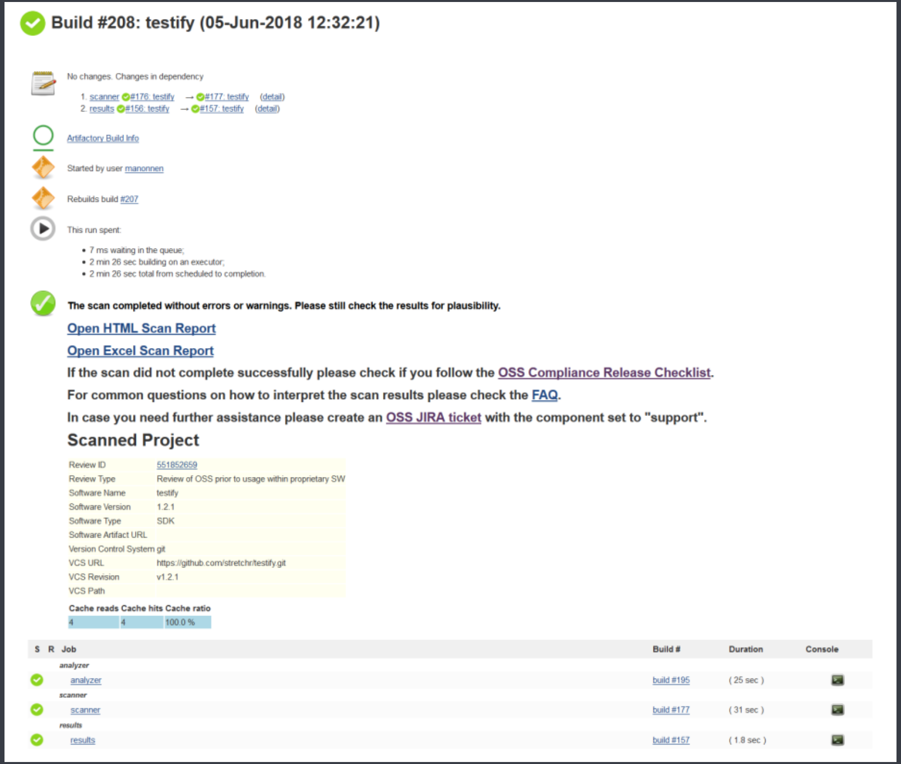


```yaml
---
- id: "Maven:org.hamcrest::"
  curations:
    homepage_url: "http://hamcrest.org/JavaHamcrest/"
    comment: "Use the actual homepage instead of the GitHub page."
- id: "Maven:org.hamcrest:hamcrest-core:"
  curations:
    description: "Curated description."
    comment: "Fix description."
- id: "Maven:org.hamcrest:hamcrest-core:1.3"
  curations:
    declared_licenses:
    - "curated license a"
    - "curated license b"
    comment: "Declared license in pom.xml is wrong."
```

Continuous Integration

Didn’t the title include “CI/CD”?

- Set up as Jenkins multi-job
- Easy configuration thanks to CLI
- Meant to be triggered by code changes or run on demand
- Non-blocking feedback to code review tool



Build #208: testify (05-Jun-2018 12:32:21)

No changes. Changes in dependency

1. scanner #176: testify → #177: testify (detail)
2. results #156: testify → #157: testify (detail)

Artifactory Build Info

Started by user manonnen

Rebuilds build #207

This run spent:

- 7 ms waiting in the queue;
- 2 min 26 sec building on an executor;
- 2 min 26 sec total from scheduled to completion.

The scan completed without errors or warnings. Please still check the results for plausibility.

Open HTML Scan Report

Open Excel Scan Report

If the scan did not complete successfully please check if you follow the OSS Compliance Release Checklist.

For common questions on how to interpret the scan results please check the FAQ.

In case you need further assistance please create an OSS JIRA ticket with the component set to "support".

Scanned Project

Review ID	551852659
Review Type	Review of OSS prior to usage within proprietary SW
Software Name	testify
Software Version	1.2.1
Software Type	SDK
Software Artifact URL	
Version Control System	git
VCS URL	https://github.com/stretchr/testify.git
VCS Revision	v1.2.1
VCS Path	

Cache reads	Cache hits	Cache ratio
4	4	100.0 %

S	R	Job	Build #	Duration	Console
		analyzer			
✓		analyzer	build #195	(25 sec)	
		scanner			
✓		scanner	build #177	(31 sec)	
		results			
✓		results	build #157	(1.8 sec)	

Roadmap

What additional tools will the future bring?

The Evaluator

- Evaluates the scan results as OK or NOT OK based on user specified approval / rejection ruleset.

The Advisor

- Retrieves security advisories based on Analyzer results.

The Documenter

- Generates documents about the outcome of the *whole* review process, like BOMs in SPDX format (with annotations).

Thank you!
Questions?

Links

- mailto:sebastian.schuberth@here.com
- https://github.com/heremaps/oss-review-toolkit
- https://clearlydefined.io
- https://github.com/nexB/scancode-toolkit