Dynamic License Scanning

Alexander Mazuruk (Samsung), Morgan Richomme (Orange)

February 2021

License scanning in LFN projects

Regular codebase license scan reported by LF

[onap-ptl] ONAP codebase license scan, Sept. 2020

onap-ptl@lists.onap.org de la part de Steve Winslow [swinslow@linuxfoundation.org]

À : onap-ptl@lists.onap.org

Cc : Lefevre, Catherine [catherine.lefevre@intl.att.com]; Close, Pierre [pierre.close@intl.att.com]; MCCRAY, CHRISTOPHER [cm6826@att.com]; ittay.stern@att.com; Kenny Paul [kpaul@linuxfoundation.org]; David McBride [dmcbride@linuxfoundation.org]

Pièces jointes : onap-2020-09-10-subproject~1.txt (12 Ko)

2020-09

Hello ONAP PTLs,

I am attaching links to the subprojects results of the most recent ONAP codebase license scans. These are based on a scan of a repo snapshot as of Sep

The key findings, as well as the overall license summary, can be found at the following address:

https://lfscanning.org/reports/onap/onap-2020-09-9fdc925e-0f05-4997-85dc-5ffdb5c2db54.html

- There is a high-priority finding for **externalapi-nbi**, regarding a GPL-3.0 file which should be removed.
- There is also a high-priority finding for **aai-traversal** (noted in last month's scans also), relating to the inclusion of a .jar file containing binary code ir

The full spreadsheet with a list of all licenses and files can be found at:

https://lfscanning.org/reports/onap/onap-2020-09-9fdc925e-0f05-4997-85dc-5ffdb5c2db54.xlsx

Although these links and its contents are not confidential, they may be considered sensitive and should not be generally publicized / uploaded to public wi

Please take a look at the findings and recommendations available at the first URL. There are also separate reports for each subproject, and the URLs to t

These reports cover license notices contained in the ONAP codebases themselves. I will also continue to update the build-time dependency license resul

Updated SPDX files for the scan results from each subproject's repos can be found at https://github.com/lfscanning/spdx-onap.

Please feel free to let me know if you have any questions. Best,
Steve

--
Steve Winslow
Director of Strategic Programs
The Linux Foundation
swinslow@linuxfoundation.org

Linux Foundation License Scan report

Project: **onap**
Subproject: **(all subprojects)**
Snapshot on: **2020-09** (show repos)

Key findings:

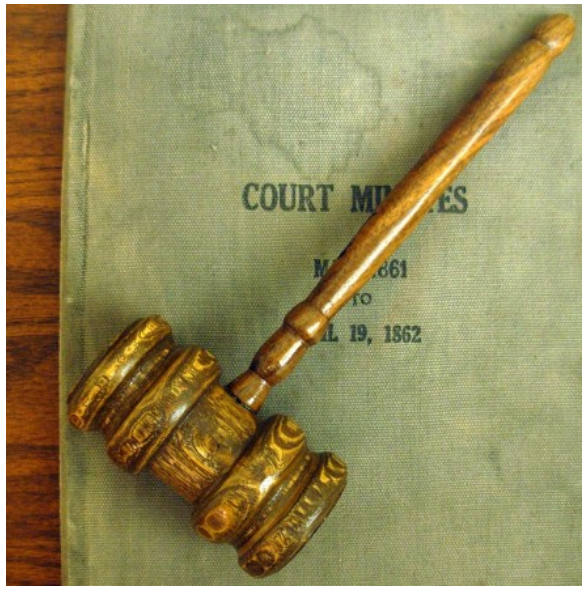
Finding #1
Priority: **Very High**
This file is under GPL-3.0 (with an OpenSSL-related exception), which is typically seen as a strong copyleft license. This file should likely be removed from the repo.
1 file (show files)

Finding #2
Priority: **High**
This repo contains one or more .jar files with compiled binary / object code. For .jar files that are upstream dependencies, we would strongly recommend pulling those in at build-time rather than distributing them in the source repos. Or, if they contain the project's own compiled binary / object code, we would not recommend distributing them within the source code repo itself, and instead configuring to compile it at build time.
559 files (show files)

License summary:

Project licenses:	
Apache-2.0	47015
Apache-2.0 (ASF license header)	381
Apache-2.0 (list of dependencies)	2
Apache-2.0 AND CC-BY-4.0	4041
Apache-2.0 OR EPL-1.0	31
CC-BY-4.0	1280
Needs review:	
Binary file	553
Binary file - Apache-2.0	6
Copyleft:	
GPL-3.0-or-later WITH Metaswitch OpenSSL Exception	1
Advertising clause:	
OpenSSL AND BSD-2-Clause	1
Attribution:	
Apache (version unspecified, from upstream)	1
Apache-2.0 AND (MIT OR GPL (version unspecified))	1
Apache-2.0 AND (MIT OR GPL-3.0-only)	1
Apache-2.0 AND BSD (version unspecified) and Public domain statement	1

Why it is import to scan licenses...



Because it is the law...

Because breaking the law could be very costly for companies/communities

Because it is important to keep open-source credible, legitimate and usable- projects with record of licensing issues create bad reputation for open-source in general. Some corporations even prohibit use open-source due to potential licensing issues.

Codebase scanning does not see everything...

Codebase scanning usually sees only explicit references to licence in the codebase

BUT

We distribute lots of dockers (from the Nexus) built with LOTS of upstream components

Most of the time the teams use baseline images without checking all the components in it

These upstream components may include LOTS of licences not seen by a static scanning

We Need a dynamic scanning to be sure that what we distribute conforms to licensing requirements

For mature projects, static codebase scan is not enough

https://www.linuxfoundation.org/wp-content/uploads/Docker-Containers-for-Legal-Professionals-Whitepaper_042420.pdf

Dynamic scanning with Tern and ScanCode toolkit

https://github.com/tern-tools/tern

Dynamic scan of the images
Stand-alone version queries
package manager for licences
of installed software

https://github.com/nexB/scancode-toolkit

Used as extension for Tern.
Analysis on file-by-file basis of image contents

Dynamic scanning with Tern and ScanCode toolkit

TERN

Tern is an inspection tool to find the metadata of the packages installed in a container image.

It unpacks each layer, and mounts them one-by-one using overlayfs for analysis.

Queries package manager for installed packages and licences. Relies on correctness of how packages were marked in package manager.

Relatively quick – a large image should be scanned in less than 1h.

We recommend setting the format to html or yaml/json for parsing as those include Relation between packages and licenses

ScanCode

ScanCode detects licenses, copyrights, package manifests, direct dependencies, and more both in source code and binaries.

Does diff comparison between a database of license texts and code instead of relying only on approximate regex patterns
or probabilistic search, edit distance or machine learning.

It takes a long time to perform a scan – in most cases counted in hours for a Docker image.

It has its own reporting tools, including dedicated GUI app.

Used by Eclipse Foundation, OpenEmbedded, Free Software Foundation and many more

From Static to Dynamic scanning in ONAP

TERN

Tern at Version: 2.2.0

The following report was generated for "nexus3.onap.org:" image.

▼ Summary of Licenses Found

GPL-2.0 GPL-2.0-or-later LGPL-2.0-or-later BSD Public-Domain
GPL
BSD
MPL GPL
Public-Domain
GPL-2.0-or-later LGPL-2.0-or-later
Custom
ISC
zlib
FTL GPL-2.0-or-later
OpenSSL
LGPL-2.1-or-later
Libpng
custom:XFREE86
custom
GPL-2.0-or-later LGPL-2.0 BSD-3-Clause MIT
GPL-3.0-or-later
MIT
GPL-3.0 LGPL
GPL-2.0
MPL-2.0 GPL-2.0-or-later
GPL-2.0 and GPL-2.0-or-later and LGPL-2.0 and MIT
GPL LGPL
MPL-1.1 GPL-2.0 LGPL-2.1
GPL2
MIT BSD GPL2+
IJG
GPL-2.0-or-later Public-Domain

TERN

Tern at Version: 2.2.0

The following report was generated for "nexus3.onap.org:" image.

▶ Summary of Licenses Found

▼ REPORT DETAILS

▼ images : [1]

▼ nexus3.onap.org :

repotag : nexus3.onap.org:10001/onap/aai-schema-service:1.7.9

name : nexus3.onap.org

tag :

▶ manifest : [1]

▶ config :

▼ layers : [9]

▼ b7e513f178 :

diff_id : f1b5933fe4b5f49bbe8258745cf396afe07e625bdab3168e364daf7c956b6b81

fs_hash : 1e0075f82ff52fdb35364de288bcd13cba6632e9b7a568642be31cda24fdfaee

tar_file : b7e513f1782880dddf7b47963f82673b3dbd5c2eeb337d0c96e1ab6d9f3b76bd/layer.tar

created_by : /bin/sh -c #(nop) ADD file:a86aea1f3a7d68f6ae03397b99ea77f2e9ee901c5c59e59f76f93adbb4035913 in /

▼ packages : [14]

▼ musl :

name : musl

version : 1.1.20-r4

pkg_license : MIT

copyright :

proj_url : http://www.musl-libc.org/

download_url :

checksum :

▶ origins : [1]

▶ files : [0]

▶ pkg_licenses : [0]

▼ busybox :

name : busybox

version : 1.29.3-r10

pkg_license : GPL-2.0

copyright :

proj_url : http://busybox.net

Layer 1

Pkg 1 of Layer 1

License of Pkg 1 of Layer 1

Pkg 2 of Layer 1

License of Pkg 2 of Layer 1

From Static to Dynamic scanning in ONAP

1 or 2 warning reported

Usually simple to fix :
- remove a file/directory
- complete the licence description

76 % of the Docker images we are building contain GPLv3 libraries/packages…

* : Manual tests done with Tern only on subset of 163/184 of ONAP Docker images

We need to have an automated way to report licensing issues as early as possible to the PTLs

Automation of Dynamic scanning: PoC part I – weekly run

ONAP solution consuming the dockers

ONAP registries where we push the dockers we build

1 : Run Tern on weekly master (manually then automatically as part of weekly tests)
Push results as part of weekly tests
Share results with the PTL
Create JIRA (as we do for security issues)

Docker files
Hosted in ONAP repo
Reviewed in Gerrit

LF docker build chain

Automation of Dynamic scanning POC part II – part of build verif job

ONAP solution consuming the dockers

ONAP registries where we push the dockers we build

2. Add scancode and include verification as closely as possible to the docker build

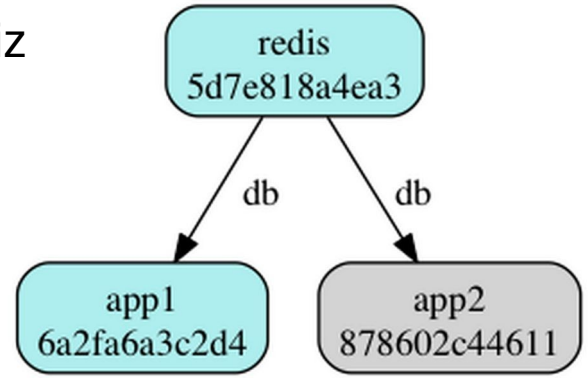
Add a Tern + Scancode processing in docker build chain

Docker files
Hosted in ONAP repo
Reviewed in Gerrit

LF docker build chain

Other useful tools

Dockviz



https://github.com/justone/dockviz

Generation of Docker's layer-by-layer dependancy graph
Useful in multiple image scenario (e.g. kubernetes), to find which fixes may fix most dependant images

https://gerrit.onap.org/r/gitweb?p=integration.git;a=tree;f=test/legal/docker_license_analysis

Vagrantfile with tern & scancode installed + usage instructions.
It is virtualized rather than containerized for CI usage due to need of fuse device access & docker.sock acces on host if dockerized.

Conclusions

We will not fix everything at once… But slow and steady wins the race.

Tern + Scancode processing takes time and resources – discussion with Tern community to adjust the configuration to improve performance.

Reuse of Baseline images (java & python produced by Integration team) reduce the risk and Integration is responsible of these images. If you prefer to use base images of your choice, you are free to do so but then you are responsible for licensing issues.
Usage of official Baseline images should be considered as a best practice and adopted by new projects.

Automate (where feasible) **generation of Compliance documents for Docker images** (needs some hosting from LF for source code of packages, etc)

It might be impossible to rid ourselves of GPLv3 packages entirely. I propose we avoid them as much as possible and ask for waiver when required (e.g. onap-python baseline image) & provide compliance for the packages + link to it in the depending images.

Thank you

ONAP Dynamic scanning in weekly master CI/CD chain

https://logs.onap.org/onap-integration/weekly/onap_weekly_pod4_master/01-14-2021_00-01/tern-reports/

164 images analyzed (on 183 images detected in the ONAP cluster)
125 on 164 (76 %) includes GPLv3 components

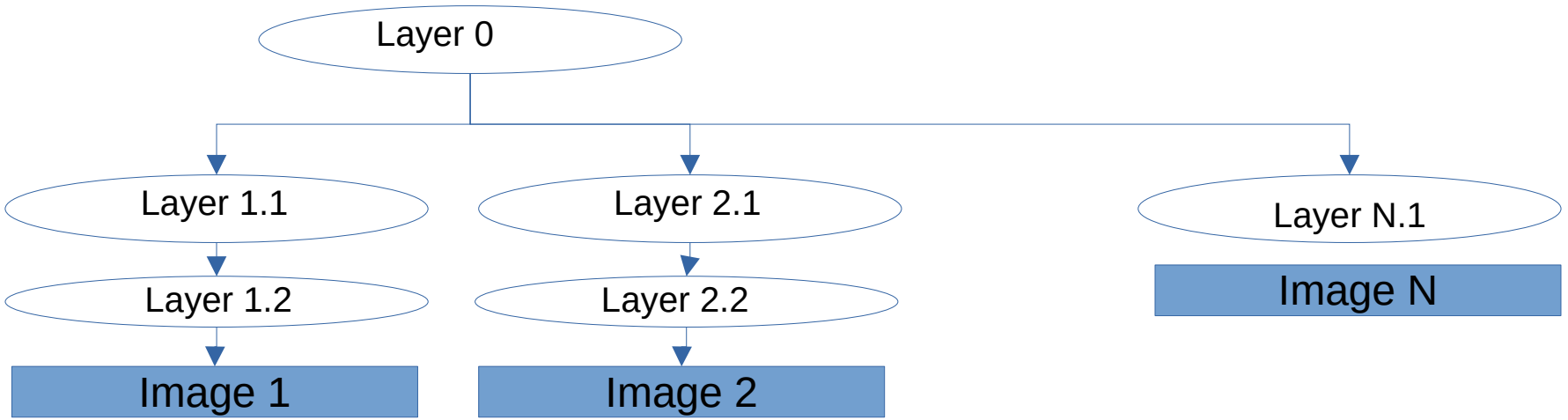
Upstream components (dockerhub) include GPLv3 components

The python baseline image contains 2 python lib released in GPLv3 (gdbm and readline)

Libraries are indicated in the report, but postprocessing needed to extract main GPLv3 components used in the dockers

ONAP Dynamic scanning in weekly master CI/CD chain

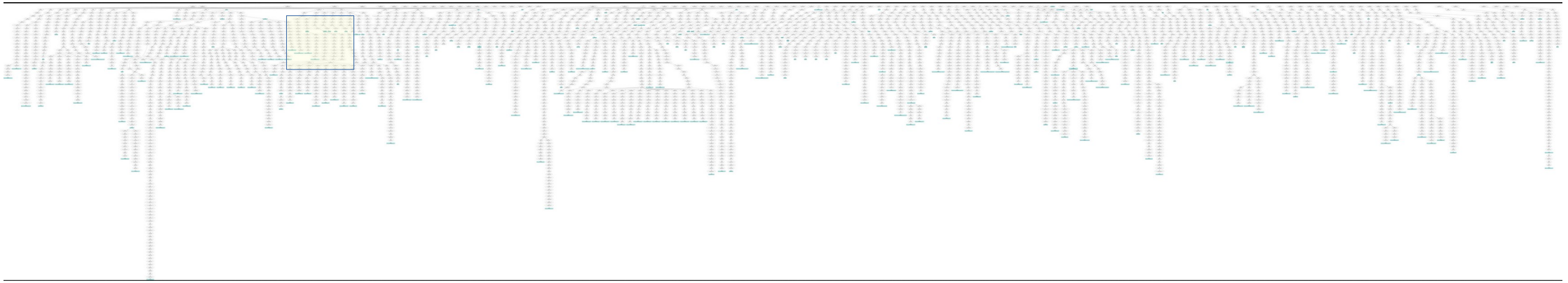
It is possible to build a complete (and complex) dependency graph of the layers of all the dirstributed dockers.
Any fix in a common layer can fix several problems

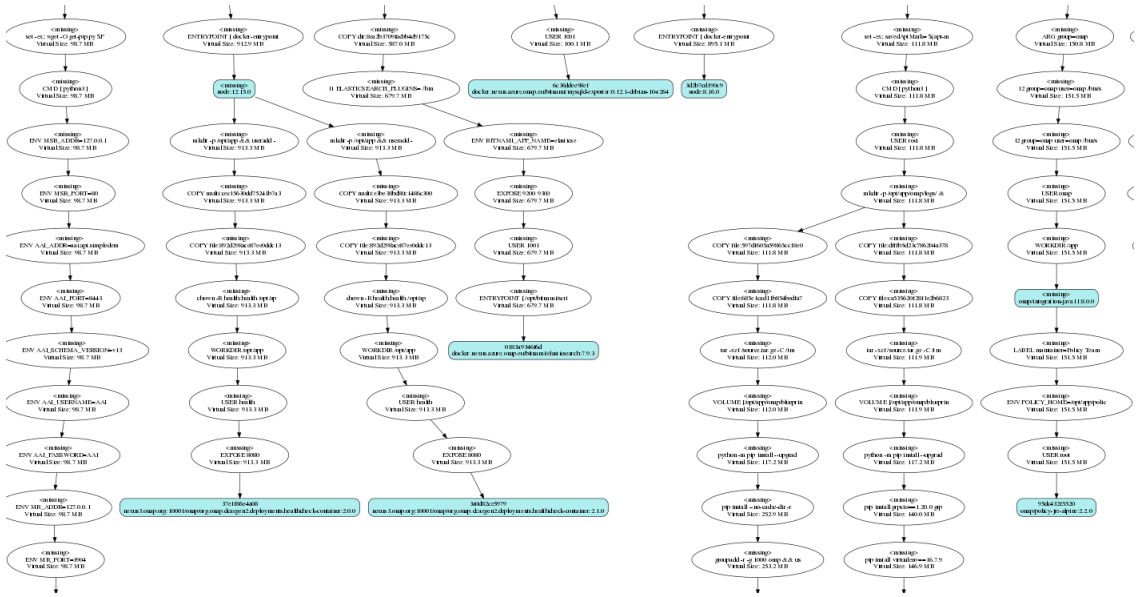


Simplified dependency view

ONAP Dynamic scanning in weekly master CI/CD chain

Full view : https://logs.onap.org/onap-integration/weekly/onap_weekly_pod4_master/01-14-2021_00-01/tern-reports/images-created-by2.png





Credits

Photos

- Philosophers of law ask "what is law, and what should it be? By Jonathunder – wikipedia - CC BY-SA 4.0
- United States one dollar bill. - wikipedia - Public Domain
- Professional reputation, red and green buttons with hand gestures – pixy.org - CC BY-NC-ND 4.0
- Warning Sign – Openclipart – Public Domain