SBOM with the Yocto Project for Automotive Grade Linux

Intro and Lessons Learned

Jan-Simon Möller, FOSDEM 2023

Intro

Jan-Simon Möller

AGL Release Manager

YP Board Member for AGL

jsmoeller@linuxfoundation.org

Topics

- Automotive Grade Linux in a nutshell
- SBOM Tools we evaluate(d)
  - meta-spdxscanner
  - create-spdx
- What we've learned

Automotive Grade Linux in a nutshell

Automotive Grade Linux

- Open Source Linux Platform for different use cases in the car
  - Infotainment , Instrument Cluster, HUD
  - Telematics, Software Defined Vehicule (SDV)
- Code first
- AGL "Unified Code Base" is the Software Platform
  - It is built using The Yocto Project

Tools we evaluate(d)

meta-spdxscanner

- Within AGL one of the member companies started to work on license compliance and evaluated multiple solutions, developed own connectors. We helped and encouraged them to work upstream.
- That work is available on git.yoctoproject.org as "meta-spdxscanner".
- Essentially this is a "Post-Mortem"-SBOM approach.
- Some of it is still useful, some maybe not.
- It pre-dates the now built-in create-spdx.bbclass.

meta-spdxscanner continued

- It requires a fossology instance to upload an run scanners against the source code. Other engines are also supported.
- It then presents the results for human review and correction/approval
- can output the curated data as SPDX
- and be paired with SW360

create-spdx.bbclass

- Built-in support for spdx files was added to upstream Yocto Project.
    - Tnx Joshua et al.
- It does **not** require an external server and uses the already available metadata.
- Runs during the build phase and files are part of the output folder

create-spdx.bbclass continued

- Enabled by default now for our releases
- E.g.: https://download.automotivelinux.org/AGL/release/needlefish/14.0.3/qemux86-64/deploy/images/qemux86-64/agl-demo-platform-crosssdk-qemux86-64-20230123140226.spdx.json

What we've learned

Lessons learned - I

- Post-mortem analysis requires
  - more CPU
  - more eyes
  - more coordination (what scan is what and where)
- Level of trust ?!
  - Maybe the extra eyes is exactly what you want or are required to do !

Lessons learned - II

- Analysis during the build is faster and needs less additional resources
- At this stage we know what goes into the packages and into the images. If I just review a scan of a tarball, we do not know.
- Metadata vs. scan+human review.

Lessons Learned - III

- We do use create-spdx now by default !
- So, we can output spdx files - great … what now ?
- TLDR:
  - work to do on:
    - tooling
    - interaction
    - presentation/visualization

Looking forward to the presentations and discussions here in the devroom !

Q/A

Thank You!