Activity Five

License Compliance and Auditing

Beth ‘pidge’ Flanagan and Paul Barker
Togán Labs Ltd.

License Compliance and Auditing

Togán Labs Ltd

- Ireland/UK based Embedded Consultancy
- Oryx Linux and Oryx Linux Plus
- OpenChain Partner
- Made up of OpenEmbedded/Yocto Project Developers
- We REALLY like License Compliance

TogánLabs

License Compliance and Auditing - Overview

Topics

- meta-wrong
  - all the horrible in one lovely layer
- meta-spdxscanner
  - temp fork of mainline

meta-wrong recipes

- bad-chksum
- bad-license-mix
- closed-app
- hello-lib
- mit-app
- shotgun-lic
- use-hello-lib

bad-chksum

```
bitbake bad-chksum -f -c cleanall
```

```
bitbake bad-chksum
```

bad-chksum

Two issues (one, not so obvious)

bad-chksum

Two issues (one, not so obvious)

- Bad checksum
- more ../conf/distro/wrong.conf
  - license-checksum in WARN_QA
  - devs tend to ignore bb.warns

closed-app

```
bitbake closed-app -f -c cleanall
```

```
bitbake closed-app
```

closed-app

Again, two issues (one, not so obvious)

closed-app

Again, two issues (one, not so obvious)

- build/tmp/work/armv5e-poky-linux-gnueabi/closed-app/1.0.0-r0/closed-app-1.0.0/app.py
    - wrong license
- look at recipe
    - specifically the LIC_FILES_CHKSUM
    - CLOSED ignores checksum

closed-app

A short diversion….

- CLOSED is not a license
- it’s being used as a lazy way to subvert some QA warnings
- Use at your peril

bad-license-mix

```
more bad-license-mix/bad-license-mix_1.0.0.bb
```

bad-license-mix

LICENSE = "CLOSED & GPLv2"

- theoretically possible
- but we need to look at the code

```
more tmp/work/armv5e-poky-linux-gnueabi/bad-license-mix/1.0.0-r0/bad-license-mix-1.0.0/app-closed.py
```

```
more tmp/work/armv5e-poky-linux-gnueabi/bad-license-mix/1.0.0-r0/bad-license-mix-1.0.0/app.py
```

bad-license-mix

Solution here?

- Developer open source training
- This can sometimes be difficult to catch with copy-paste code

shotgun-lic

```
more shotgun-lic/shotgun-lic_1.0.0.bb
```

shotgun-lic

more shotgun-lic/shotgun-lic_1.0.0.bb

- LICENSE is theoretically valid
- gold star for
  - LICENSE_PATH += "${LAYERDIR}/files/licenses" in layer.conf
  - Not using CLOSED for MyWeirdProprietaryLicense

Let’s look at the source!

shotgun-lic

tmp/work/armv5e-poky-linux-gnueabi/shotgun-lic/1.0.0-r0/shotgun-lic-1.0.0

- Two license files
  - COPYING
  - MyWeirdProprietaryLicense
- Let's look at the code in random_lib and another_random_lib

shotgun-lic

Uhhh….

- Which files are which license?
- Why not use DEPENDS?
  - Sometimes valid reasons why you don’t
    - **don’t control upstream source**
    - **but this is non-distributable**

mit-app

```
bitbake mit-app -f -c cleanall

bitbake mit-app
```

mit-app

No errors!

But does this mean nothing is wrong…?

This is where license scanning helps you!

mit-app

Two files:

- app.py
  - LIC_FILES_CHKSUM looks at this
  - License is correct
- local_lib.py
  - Not covered by LIC_FILES_CHKSUM
  - Contains a GPLv2 header

The application needs fixing!

hello-lib & use-hello-lib

```
bitbake hello-lib -f -c cleanall

bitbake use-hello-lib -f -c cleanall

bitbake use-hello-lib
```

hello-lib & use-hello-lib

No errors again!

But let’s look closer…

hello-lib & use-hello-lib

Licenses:

- hello-lib: LGPLv2
  - contains hello_lib.py

- use-hello-lib: CLOSED
  - imports hello_lib

- Valid usage of an LGPL library

hello-lib & use-hello-lib

Let’s look deeper:

- hello-lib contains hello_lib.py
  - License header is GPLv2 not LGPLv2
  - This is the sort of issue license scanning will detect
- So let’s fix hello-lib_1.0.0.bb:
  - LICENSE = “GPLv2”

bitbake use-hello-lib (again)

hello-lib & use-hello-lib

Still no errors…

- But using GPLv2 library from a closed app is not valid
- License scanning tools won’t catch this
- This is where you need to use judgement or legal advice

meta-spdxscanner

- Using the Togán Labs fork of meta-spdxscanner
  - **https://gitlab.com/toganlabs/meta-spdxscanner**
  - **requires meta-gplv2**
- Not a fan of DoSOCSv2, looking at moving
  - **scancode**
  - **fossology**
- Want to help? pidge@toganlabs.com

meta-spdxscanner

- spdx-runs/gobject-introspection.spdx
  - **find PackageLicenseInfoFromFiles**
- the license of source and the license of package is usually different
  - **This is ok**
  - **Things we don’t ship (setup.py)**
  - **But we need to compare LICENSE to what we see here.**

meta-spdxscanner

- recipe states
  - **LICENSE = "LGPLv2+ & GPLv2+"**
- scan states

meta-spdxscanner

- recipe states
  - LICENSE = "LGPLv2+ & GPLv2+"
- scan states
  - GPL-3.0+ & LicenseRef-Freeware & LicenseRef-MIT-style & LicenseRef-Public-domain & LicenseRef-See-file & X11 & GPL-2.0 & GPL-2.0-with-autoconf-exception & LGPL-2.0 & LGPL-2.1+ & LicenseRef-GPL-3.0+-with-bison-exception & MIT & BSD-2-Clause & LicenseRef-See-doc.OTHER & LicenseRef-GPL-exception & GPL-2.0+ & LicenseRef-FSF & LGPL-2.0+

meta-spdxscanner

- Find the GPL files!
  - **What is scannerparser.c**
- Look in the source, see if it’s something we distribute
  - **if so, we need to fix the LICENSE**
  - **maybe on a package layer**
    - **LICENSE_${PN}-dbg**

License Auditing and Compliance

Q&A