Building C/C++ SBOMs with pkgconf

FOSDEM 2026

About me

Research psychologist turned security engineer and SRE (2006-present)

Alpine Linux, late 2009 to present

- x86-64, loongarch, GNU toolchain, ifupdown-ng, libucontext, gcompat, ...

Alpine's security team 2021 to present

Focus on long-term sustainability in Alpine

Socials:

- Mastodon: ariadne@treehouse.systems
- Bluesky: @ariadne.space

Why are we here?

Last year at FOSDEM, Chris Swan gave a talk about his struggles with C SBOMs

- Meanwhile in pkgconf, we have been building tooling for C SBOMs since 2021
- ...so this talk is kind of a response to his.

C, C++ and the SBOM model

SBOM models are largely focused on applications built with newer languages where dependencies are first-class objects

- This leads to a package-oriented model as seen in Go, Rust, Java, etc.
- Point your SBOM generator at a lockfile and get a build-time SBOM!

While C and C++ have modules now, nobody is really using them

- Libraries and SDKs maintain dominance: no packages, no lockfiles
- Dependency management therefore exists elsewhere in the stack

Enter pkgconf

There are two major SDK managers for C/C++

CMake – largely used for Windows projects, but also on Unix-likes

- Also a build system in its own right
- (started life as a project at US Department of Energy, and you can tell in its design)

pkg-config – largely used on Unix systems, but also on Windows

- (started life as a GNOME project, then I rewrote it from scratch in 2011 as pkgconf)

pkgconf in one slide

Maintains a database of information about dependencies (SDKs)

Queries are represented as a graph of constraints which must be satisfied, solutions are also graphs

Supported by every Unix build system

- GNU Automake, CMake, Meson, ...

Ask it for compiler and linker flags, e.g.

```
% pkg-config foo --cflags
-I/usr/include/foo
% pkg-config foo --libs
-lfoo
% pkg-config foo --cflags --libs
-I/usr/include/foo -lfoo
```

Useful pkgconf SBOM tools

```
% pkgconf --digraph <query> | dot
```

Visualize a query's solution graph with GraphViz

```
% bomtool <query> > query.spdx.txt
```

Generates an SPDX 2.3 Textual SBOM for a given query

```
% spdxtool <query> > query.spdx3.json
```

Generates an SPDX 3.0 Lite JSON-LD SBOM for a given query

Basic theory

Build a pc(5) for your application as a build artifact

- … this has a similar shape to a lockfile in package-based languages!

Analyze that pc(5) file with the pkgconf tools

- … voila: C SBOMs :)

Demo

Play along if you want:

https://codeberg.org/kaniini/fosdem-sbom-demo

Things left to do...

- Quality of life improvements
  - Proper build system integration for Meson and Automake
- Adding more SPDX annotations to pkg-config files
  - ... for higher quality SBOM data!
- CycloneDX? Package-URLs?

Build system integration

The best way to proliferate SBOMs is to get them by default

- This requires Meson and Automake to support them
- CMake already has a solution here

Proposed Automake integration

`AM_SBOM_PACKAGES = foo.pc`

Generate a foo.spdx.txt SBOM from foo.pc

Patch nearly ready for submission to GNU!

Proposed Meson integration

Rework pkgconfig module to return a pkgconfig object:

```
pkg = import('pkgconfig')
pc = pkg.generate(…)
sbom = pc.generate_sbom(type: 'spdx')
```

Present discussion ongoing in #meson-build IRC!

SPDX metadata

Several new fields have been added to pc(5) files for SBOMs:

- License: the SPDX license expression
- License.file: an optional URI to the license file text
- Maintainer: an optional contact for the maintainer
- Source: an optional URI to the source code

Use them to improve C SBOMs!

Meson support for these fields hopefully soon!

Package URLs

- PURLs provide stable URIs to identify packages
  - C/C++ SDKs are not shaped like packages
  - … but stable URIs to identify SDKs is still useful
  - … so some work will be needed to bridge the gap
  - … also, we need to identify who supplied the SDK as part of the URI
- I have thoughts, lets chat in the hallway track :)

CycloneDX

Most of the spdxtool code can be recycled to generate CycloneDX SBOMs

- Someone just needs to do the work...

Thanks to...

Chainguard, Edera

- Allowing me to work on this stuff since 2021

The FreeBSD Foundation, Quansight

- Sponsoring others to work on SBOM support

Tuukka Pasanen

- Writing the spdxtool prototype!

You!

- Making use of these new features to improve C SBOMs!

Questions?

IRC: irc.oftc.net #pkgconf
GitHub: https://github.com/pkgconf/pkgconf
Socials:
- Mastodon: ariadne@treehouse.systems
- Bluesky: @ariadne.space