SPDX: The Lingua Franca of Open Source Governance

Gary O’Neall, Source Auditor

Tim Mackey
Black Duck by Synopsys
@TimInTech
License (mis-)management

Stories from 15 years of Open Source analysis
Microsoft Acquisition of a SaaS Company

CTO Credibility

“Here’s the Open Source Disclosure” – based on grep’ing for licenses

“Here’s some more we missed” – based on engineer’s observation

“Here’s some more” after surveying all engineers

“Oops – forgot this one”
Large Software Supplier using Apache MQ
Large Software Supplier using Apache MQ

- LGPL Library inside another open source package inside a large app
- Found and fixed by Apache, but already out there
- The original source was removed by Apache – makes it hard to meet the source distribution obligations
- Would have been easy to update the versions if they knew of the issue
- Apache could have probably avoided the issue if they had tooling in place to maintain the embedded licenses (partially addressed by RAT)
Audits for Inbound Software

- Large corporation which embeds software in devices
- Very concerned about compliance
- Most inbound software suppliers’ disclosure is incorrect
- Hires external software auditors
  - Cost of audits
  - Concerns about confidentiality
  - Just doing a 3 way NDA is a challenge
GhostScript and iText – version caution!

- Depending on version and which fork, Ghostscript may be under GPL, Aladdin Free Public License (which forbids commercial distribution), or AGPL
  - Recently, a Ghostscript litigation tested the enforceability of open source licenses (reference https://qz.com/981029/a-federal-court-has-ruled-that-an-open-source-license-is-an-enforceable-contract/)

- Versions of iText prior to 5.0 use a choice of Mozilla Public License or the GPL license. Versions 5.0 and later use the AGPL license.
Unnecessary scares

• GPL in contrib directories – zLib contrib/ada/zlib.ads “…under the terms of the GNU General Public License …”
• GPL build tools
• Lawyers looking at the list of all identified licenses without additional info can get quite (unnecessarily) concerned
• Takes some time during analysis to determine how the GPL code is used
Did we really distribute this?

- Leaking tools as part of the distribution
  - Testing tools – some GPL with redistribution requirements
  - Build environment tooling

Image by Hossam M. Omar under Pexels’ license
So what’s this SPDX thing?
What’s in your software?

- What are the ingredients?
- How is each ingredient used?
  - License
  - Relationship to product
- What do we know about each ingredient?
SPDX for Governance

- Generate
- Store
- Aggregate
- Query
Governance Today

- Code
  - BOM
- Deployments
  - BOM
- Audit
  - BOM
Governance Challenges

Requires Manual Labor
- Keeping Spreadsheet updated

Requires Compliance
- Reporting usage
- Adherence to Policy

Hard to standardize tooling
- Require aggregation of diverse tool outputs

Governance Goals

Automate building a master BOM

Automate Reporting

Produce single aggregable output
Governance with SPDX

spanql protocol and RDF query language
Governance with SPDX
Demo

Apache Jena
Fuseki

SPARQL

https://gitlab.com/yevster/spdx-server
prefix spdx: <http://spdx.org/rdf/terms#>
prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

select distinct ?name ?licenseConcluded ?licenseDeclared
{
    ?pkg rdf:type spdx:Package ;
        spdx:name ?name .
    FILTER regex(str(?pkg), "1.0.23")
}
List Packages With No License Declared

prefix spdx: <http://spdx.org/rdf/terms#>
select distinct ?item ?itemName
{
  { ?item spdx:licenseDeclared ?license} .
  OPTIONAL {?item spdx:name ?itemName} .
  FILTER (?license in (spdx:noassertion, spdx:none))
}

Enforcing Licenses with SPARQL
List Packages With No License Declared – Filtered For Our Version

```
prefix spdx: <http://spdx.org/rdf/terms#>
select distinct ?item ?itemName
{
  { ?item spdx:licenseDeclared ?license} .
  OPTIONAL {?item spdx:name ?itemName} .
  FILTER (?license in (spdx:noassertion, spdx:none))
  .FILTER contains(str(?item), "1.0.23")
}
```
prefix spdx: <http://spdx.org/rdf/terms#>
select distinct ?item ?p ?o
{
    ?item spdx:name 'jep'.
    {?item ?p ?o
     .FILTER regex(str(?item), "1.0.23")
    }
}
List Packages With Sensitive Licenses

prefix spdx: <http://spdx.org/rdf/terms#>

select distinct ?item ?itemName ?license
{
  {?
    ?item spdx:licenseDeclared ?license}
  UNION
  {?
    ?item spdx:licenseConcluded ?license}
  OPTIONAL {?
    ?item spdx:name ?itemName}
  FILTER (strstarts(str(?license), str(licenseList:AGPL-3.0)))
  .FILTER regex(str(?item), "1.0.23")
}
Questions?