OPEN SOURCE SOFTWARE ISSUES REMEDIATION

SOME RECOMMENDATION HOW TO REMEDIATE OPEN SOURCE ISSUES

Jari Koivisto
FLIGHT EU 2021
2021-04-20

DISCLAIMER

- I am not a lawyer
  - If in any doubt, consult your legal department or seek help from outside legal counsels
- These are just recommendations
  - For most of the issues there can be several ways to remediate

SOME ASSUMPTIONS

- Everyone on this call knows the basics of open source licensing
  - E.g. what is copyleft, strong / weak copyleft, etc.
- Everyone knows basics of software products, different deployment models, etc.
  - Shipping product vs. SaaS
- The coming examples are mainly for products that are shipped
  - SaaS products are easier and AGPL and SSPL licensed components are typically the ones one needs to be more careful, also non-commercial and components without any license need attention

AGPL – Affero GPL
SSPL – Server Side Public License

ACTIONS NEEDED ONCE CODE AUDIT DONE

- Code audit done for internal or external (typically for M&A) codebase
- A number of issues found
  - Depending of the size of the audit the audit report may have thousands of line items
- Analyze the audit report and decide, what are the issues that need attention
  - Questionnaires, find out the use of the components, etc.
  - Concentrate on components that are under copyleft licenses or components without any license

STRONG COPYLEFT COMPONENTS

EMBEDDED GPL AND SIMILAR STRONG COPYLEFT SNIPPETS

- Remediation ideas:
  - Can the component be open sourced?
    - If yes → Package the component and make ready for release
    - If no → Remove and rewrite (this is the typical remediation recommendation)

GPL AND OTHER STRONG COPYLEFT LICENSED COMPONENTS

- Some things to find out:
  1. Is the component running in a separate process?
  2. How is the component accessed?
     - Inter-process communication (IPC), e.g. sockets, pipes typically o.k.
  3. Is intimate knowledge needed about the GPL component?
  4. Are there modifications?
- Some remediation thoughts:
  - Make sure that the GPL component runs in a separate process
  - Communication between proprietary code and GPL through sockets, pipes, etc.
  - Find out if the same functionality can be found using a component that is under a permissive license
  - If needed, consider if the proprietary code can be released

CC BY-SA SNIPPETS

- Typical source is StackOverflow, other typical source is Wikipedia
- A couple of options for remediation:
  1. Is code copy-pasted or is the source used as an inspiration?
  2. Is the code very generic → if yes, normally o.k.
  3. Find out if StackOverflow is **not** the original source of the snippet. If snippet is originally coming from a source that has a permissive license → typically o.k.
  4. Is the component something that can be open sourced?
     - If yes → package the component and make ready for release
     - If no → Remove and rewrite (common remediation action)

LGPL

- Questions to ask:
  - Are there any modifications in the component?
  - How is the component linked?
    - Static / dynamic linking
- If there are modifications one needs to package the code and make it available under LGPL if requested
- If static linking, can dynamic linking be used?

GPLv3, LGPLv3

- The goal of v3 license is to prevent Tivoization
  - This is a problem, especially if the product is a consumer premises product
- If a consumer premises product (such as STB / DVR / smart watch / etc.):
  - Find out if the same functionality can be found using components that are not under v3 license
- If a product is **not** shipped to a consumer but to another business
  - Normally v3 Tivoization prevention clause is not a problem, but normal GPL and LGPL issues need to be taken into account

https://en.wikipedia.org/wiki/Tivoization

Tivoization is the creation of a system that incorporates software under the terms of a copyleft software license (like the GPL), but uses hardware restrictions or digital rights management to prevent users from running modified versions of the software on that hardware.

NON-COMMERCIAL USE LICENSE

SHIPPING OR SAAS PRODUCTS

COMPONENTS WITH NON-COMMERCIAL LICENSE

- E.g. CC BY-NC or sometimes someone adds non-commercial addendum to MIT, BSD, etc. licenses
- Some remediation options:
  1. Find out if there is a commercial version available
  2. Find out if there are other components with similar functionality and that can be used for commercial products
  3. Remove and write the functionality in-house

NO LICENSE COMPONENTS

SHIPPING AND SAAS PRODUCTS

COMPONENTS WITHOUT A LICENSE
NO LICENSE → NO RIGHT TO USE

- Typical sources: blog posts, GitHub
- Some remediation options:
  1. Ask the copyright owner if she/he can add a license
     - Often times the idea was that the code is under some no copyleft license or public domain, but the one who wrote the code forgot the license or statement that it is public domain
  2. Ask copyright owner if it is o.k. to use the code in commercial product
     - Get this in written (email normally o.k.)
  3. Remove and rewrite

No license ≠ public domain

SOME SPECIAL CASES

MANY LICENSE OPTIONS

- If a component has several (2 or more) license options:
  - Typically one selects the more permissive license, but not always, e.g. because of license compatibility

DIFFERENT VERSIONS HAVE DIFFERENT LICENSES

- Same component, different versions have different licenses
  - Newer version may have more or less permissive license
    - Newer version more permissive license (e.g. old GPL, new MIT) → update the component to the latest version (MIT)
    - Newer version less permissive license: need to be careful with component updates

KERNEL LOADABLE MODULES (KLMs) / LOADABLE KERNEL MODULES (LKMs)

- KLM:
  - Are they GPLv2 or not?
  - Safe bet is to say that KLMs are in general GPLv2
    - afaik there is no high court decision on this and therefore it is still a gray area

CONTACTS

https://www.linkedin.com/in/jarikoivisto/
jari.p.koivisto@iki.fi
+41 78 7479791 (Central Europe Time zone)