헤멜 아마인
Armijn Hemel, MSc
Tjaldur Software Governance Solutions

November 5, 2014

About Armijn

- using Open Source software since 1994
- MSc Computer Science from Utrecht University (The Netherlands)
- core team `gpl-violations.org` from 2005 - May 2012, helping solve hundreds of incompliance cases
- owner Tjaldur Software Governance Solutions, helping companies with compliance issues
- creator of the Binary Analysis Tool for analysing binary artefacts for license and security issues
- European coordinator Linux Defenders at Open Invention Network

Today's topic

Today I will talk about:

- why using open source makes business sense
- enforcement of one open source license, namely GNU General Public License (GPL)
- how to avoid getting in trouble with GPL enforcers

I am not a lawyer and nothing I say in this talk is legal advise. When in doubt, ask a competent lawyer.

Nothing I am saying today is rocket science, and everything is "common sense". This is because open source license trouble usually is not a technical problem but a *process* problem.

Software reuse is smart

Software is increasingly commoditized as open source software. In the late 1980s it started with single software components, now entire software stacks are a commodity, like:

- OpenStack and Apache Cloudstack (server)
- Android (mobile)
- OpenWrt and Yocto (networking)

Not reusing software, but developing everything from scratch is becoming a *stupid business decision*.

Open Source

Many components are available as “Open Source”.

The source code for these components is available and the licenses allow you to reuse and redistribute, at (in practice) no cost. Many of these components are high quality (note: there are also many bad quality open source components!).

Needless to say that a lot of open source is used by many vendors: a few years ago Gartner predicted (some form of) open source would used in *every* device or program in 2012.

The various open source licenses

There is no single “Open Source”. There are about 60 open source licenses that are frequently used. They differ in various ways:

- distribution under the same (or stricter) license of “derived software”, or offer option to make code “closed” again
- (no) patent language
- (no) attribution needed
- applies for network distribution (or not)

Not all the licenses are compatible and you need to take care of following the license terms correctly. This is not always trivial and you might not be able to combine software packages *legally*. In practice it is straightforward for most software packages.

Today I will (mostly) focus on the GNU licenses.

GNU license family

One of the most popular open source license families is the GNU license family. It contains:

- General Public License (GPL)
- Lesser General Public License (LGPL) – originally intended for shared libraries
- Affero General Public License (AGPL) – networked services

Not every version of every GNU license is compatible with other licenses. Example: GPL 2 and GPL 3 are not compatible.

GNU General Public License

There are various versions of the GNU General Public License, with GPL 3 being the latest. Some widely used components licensed under GPL are:

- Linux kernel: GPL 2
- BusyBox: GPL 2
- Samba (before 3.2): GPL 2 or later
- Samba (3.2 and later): GPL 3 or later

GPL grants you a lot of freedom, in exchange for keeping the code and modifications open under the same license.

Failure to comply with the license conditions of the GPL license renders the license void. You could then be sued by copyright holders for a *copyright violation*.

GPL enforcement around the world

The GPL license has been enforced in several countries:

- Germany (several law suits, hundreds of cases settled before going to court)
- USA (settled before going to court)
- France (settled before going to court)

During a case you might not be able to sell your product in a particular country for a significant period of time.

I have been involved in cases in Germany and the US, on both sides.

Known license enforcement

Some of the packages I know the license was enforced for:

- Linux kernel
- BusyBox
- XviD
- U-Boot
- GNU utilities
- Samba
- several more programs

None of the cases so far were primarily about *money* but about *license compliance*. Not solving issues can be costly though.

GPL license violations are a business risk

The risk of getting sued by copyright holders exists, but might not be very high. You might get away for quite a long time. But GPL license violations are still a business risk!

If your exit strategy is to be bought by a big company remember they *will* (or *should*) scan everything. If you have license problems you are a lot less attractive for them: no one wants to buy a liability.

Preventing (GPL) license violations: 어떻게?

- know your limitations
- know what the GPL license demands
- know what you use
- document what you use
- document your changes
- rebuild your software in a clean environment
- upstream changes as much as possible

Know your limitations

Many engineers approach licenses as if they were mathematics. But licenses are legal documents and law is not math!

Legal documents describe interactions between humans and codify *expectations*. Software licenses are therefore inherently human. Humans are not math! There is always a gray area, with uncertainty.

Know what the GPL license demands

You should read the actual license text before drawing any conclusions. When in doubt talk to a knowledgeable lawyer.

- complete & corresponding source code with the product, or
- written offer for the complete & corresponding source code, valid for three years

Complete & corresponding source code: everything needed to recreate the binary file, and licensed under GPL or a GPL compatible license.

Know what you use

We are living in the “golden age of plagiarism”!

You need to know what you use and how you integrate it. Scan incoming components for possible license issues:

- closed source tooling (licenses/copyright/code copying)
- FOSSology (licenses/copyright)
- Ninka (licenses)
- Binary Analysis Tool (for binary only components from suppliers)
- or a combination of these tools

Be careful with copy/paste from unknown sources (webpage, discussion forum, etc.)

Also discuss with your colleagues what you import and why.

Document what you use

The GPL license requires “an appropriate copyright notice”. It is not exactly clear what this means.

At least one copyright holder doing enforcement actively checks if there is a list of components and if it is complete as part of his compliance checks.

Document clearly what you use. Adapt your build system so it generates SPDX documents.

SPDX: Software Package Data eXchange, a format to specify a “software bill of materials”

Document your changes

The GPL license says that “modified files must carry prominent notices stating that you changed the files and the date of any change”. This is often neglected.

Keep track what you change. If not in the file, then at least use meaningful commit messages for version control!

Rebuild your software in a clean environment

Many companies do not rebuild their software in a clean environment. People who enforce the license typically *do*.

Often there are undocumented dependencies or configuration options. A rebuild in a different environment often fails.

By testing if your software builds in a *clean* environment according to the instructions you provide you can catch many problems before others do.

If rebuilding requires information from the build system/scripts, then you should add those too as part of "complete & corresponding source code".

You really should consider using standardized open source build systems (OpenWrt, Yocto, Android, etcetera)

Upstream your changes

One reason that many GPL violations get noticed is because products have additional functionality (like hardware support) that is not in the "upstream" version.

People want to study, share and improve the *new* functionality, but if there is no source code they get mad. Upstreaming makes you a better "open source citizen". It also makes it easier to contact you in case there are issues.

Dealing with (GPL) license violations: 어떻게?

- communicate
- participate in the global discussion about compliance
- learn & teach

Communicate

Some people doing enforcement get very irritated if companies do not respond or do not communicate well. They get emotional and irrational and assume you are violating the license on purpose and will try to catch you for even the smallest things because you are “bad”.

Communication (coordinate with your legal team!) is key.

Participate in the global discussion about compliance

When dealing with compliance you are not alone. There are several groups discussing licensing issues and you are welcome to participate:

- Free Software Foundation Europe runs the “legal network” with hundreds of experts worldwide and organises an (invite-only) legal conference each April (mid-April 2015, Barcelona)
- Linux Foundation frequently has legal talks at its conferences
- Linux Foundation Japan organises an (invite-only) legal conference each October or November (13/14 November 2014, Yokohama)
- KOSS Law Center organises FOSSCon Korea each year (4/5 December 2014, Seoul)

but also conferences like today.

Learn & teach

Spreading knowledge inside your company is very important: you don't want a "single point of failure" for compliance issues.

Talk to people, teach (new) colleagues, build up local expertise.

Ideally this should actually start before people start to work at your company.

Push for basic copyright classes at college/university

Open source ideas are moving into other domains, such as data, design, art, biochemistry and medicine. Copyright licensing will become much more important.

My request to NIPA: please push for having a mandatory basic copyright course at college/university. This would get Korea ahead of other countries. In Europe these are often not given, or optional as part of a master's degree.

Questions?

Contact

- `armijn@tjaldur.nl`
- `http://www.tjaldur.nl/`
- Binary Analysis Tool: `http://www.binaryanalysis.org/`