Identifying licensing issues

Armijn Hemel

April 11, 2008

About me

Professional:

- 1996-2006: computer science at Utrecht University
- 2005-present: core team `gpl-violations.org`
- 2006-present: board member of NLUUG (`http://www.nluug.nl/`)
- 2006-present: Chief Random Projects at Loohuis Consulting

A word from our sponsors: Loohuis Consulting

- specialized hosting
- web development (AJAX, other buzzwords)
- GPL license compliance
- UPnP security
- router/embedded security advice

More info: `http://www.loohuis-consulting.nl/`

Today's topics and goals

- gpl-violations.org history
- problems observed in the marketplace
- how to identify licensing issues
- cause of errors + fixes

gpl-violations.org history

Started by Harald Welte in 2003. The project currently has a strong focus on the embedded market, because:

- rapidly growing market (for Linux and in general)
- traditionally closed market
- personal interests in embedded devices

gpl-violations.org has had quite some results:

- first court cases about GPL ever
- many out of court settlements
- even more cases fixed by just frowning

gpl-violations.org goals

1. Raise public awareness of the infringing use of free software, and put pressure on the infringers.
2. Give users who detect or assume GPL-licensed software is being misused a way to report them to the copyright holders.
3. Assist copyright holders in any action against GPL infringing organizations.
4. Distribute information on how a commercial entity using GPL licensed software in its products can comply with the license.

Our goal is not to sue, at least not initially.

gpl-violations.org: other facts

- it is a volunteer project
- we don't make any money, even lose on it (mostly time)
- people expect us to make miracles happen (source releases within a day)
- others think we are way too strict ("GPL nazis")

It's a dirty job, but someone has to do it.

Problems observed in the marketplace

1. no license texts shipped with the product
2. incompatible licenses and EULA
3. missing code
4. no code at all
5. combinations of the above

What actually is the problem?

Some facts about Free Software:

- Free Software is not public domain software
- Free Software has one, some or many authors
- some Free Software is written by very large corporations (look around you for some examples)
- Free Software not free to grab and just use as if it were yours

Licenses are very important in Free Software. Licenses are also very tricky and not very well understood by most people.

Time to go technical

1. some technical terminology used in licenses
2. how do we determine a violation?
3. what can you do to detect violations?

Some technical details about software (1)

Licenses talk about binaries, source code and object files. What are these?

Most programs (that we care about) are written by programmers in "human readable code":

```
#include <stdio.h>

int main(int argc, char* argv) {
        printf("hello world!\n");
}
```

This so called "source code" is translated into a program a computer can run. This process is called "compilation".

Note: The code can be the first source of violations when you (or a programmer) copy/pastes other code.

Some technical details about software (2)

“Compilation” has various phases (very very simplified):

1. parsing
2. code generation (make “object files”)
3. linking

The end result is either a program, or a piece of binary code which can be reused by other programs (often called “library”).

Some technical details about software (3)

“Linking” is combining object files and libraries (pieces of code for particular functionality) into a program.

Licenses come into play here:

- your code has a license
- an object file or library you link with has a license

It depends on the licenses of your code and the object file or library what you are allowed to do here.

Questions?

Any technical questions so far?

GPL forensics: embedded Linux device

Software is often distributed as firmware:

- bootloader (not always in the firmware, but often on the device)
- kernel
- file systems with software

Firmware is often downloadable, or preloaded on a device.

We pick apart components in the firmware and try to determine what is in there using:

- network scans
- output strings in the binaries
- debug code in the binaries
- file names + version numbers of the binaries
- previous cases (different companies often use the same suppliers)
- physical modifications to the hardware (serial port, JTAG)

This is a lot of specialized tricky mind numbing work.

Violation prevention

What you can do:

- tracking where sources come from
- forcing upstream vendors to be compliant (contract law)
- have a policy in place to deal with violations
- keep checklists/documentation on previous cases plus their resolution (issues tend to resurface a lot)
- train an engineer to at least catch the obvious mistakes

Code checking tools can help, but are not a panacea. It does not help when you only get binaries from upstream.

Circumvention

Making it more difficult for us is possible, but:

- much more expensive to do and maintain
- court might see it as deliberate circumvention
- you can't afford a single mistake

Better try to work with the copyright holders and spend your money on adding value to your product instead of fighting us.

Conclusion

Please work with us!

More information:

- http://www.gpl-violations.org/
- {laforge,armijn}@gpl-violations.org