Fork and Ignore: Fighting a GPL Violation By Coding Instead

Bradley M. Kuhn

Friday 22 August 2014

This is a text version of the slides. The "full experience" web version is available online at: http://ebb.org/bkuhn/talks/LinuxCon-North-America-2014/kallithea.html.

The source code for these slides is available at: https://gitorious.org/bkuhn/talks/source/master:LinuxCon-North-America-2014/.

GPL Violations Differ

- Most of my talks historically have focused on embedded Linux-based GPL violations.
  - These are the most prevalent & insidious.
- However, GPL violations come in all shapes and sizes.
- This is the story of a different sort of violation ...
  - ... which inspired a different sort of resolution.

What's a GPL Violation?

- This point may be too remedial for some of you.
  - If so, take a 2 minute email break.
- GPL (both v2 and v3) require:
  - The whole work licensed under GPL.
  - Complete, Corresponding Source (CCS) of that work provided, under GPL.

What's a GPL Violation?

- I could give a whole talk on any of these topics:
  - What constitutes the "whole work".
  - CCS requirements of GPL.
- If you're new to GPL enforcement, just assume for this talk:
  - There are requirements on the forms of source-code disclosures.
  - Everyone has an opinion on the "whole/combined work" question ...
  - ... but there is limited statutory guidance or Court precedent on the topic.

Fundamental Assumptions

- Generally speaking, everyone assumes:
  - proprietary software is more lucrative than Free Software.
- The veracity of this claim is immaterial.
  - only the perception that it's true matters.
- Companies therefore try to keep as much proprietary as they can.

Fundamental Assumptions

- Most developers, left to their own devices, share their code.
- They tend to share their code if it's convenient.
  - & even sometimes when it's not so convenient.
- That's not to say most developers are software freedom zealots like me.
- Rather, they err on the side of code-sharing.
  - In current times, that means developers by default release code early and often under a Free Software license.
- Thus, under these assumptions, so begins our story ...

A Developer Releases a Codebase

- The story starts like most Free Software stories:
  - In 2010-10, a developer saw some useful proprietary software ...
  - ... didn't like the existing Free Software alternatives ...
  - ... and started writing one.
- Specifically, in June 2010:
  - GitHub was (still is) a proprietary solution & only supports Git.
  - Phabricator wasn't released yet.
  - GitLab didn't exist yet (doesn't support Mercurial anyway).
- Thus, hg-app is announced on 2010-06-03 under an MIT-permissive license on Mercurial's mailing list.

There's Always A Flame War.

- A debate erupts about GPL compatible licensing.
- hg-app incorporates code from Mercurial:
  - Thus, in the view of most people, hg-app is based on Mercurial (in the copyright & GPL sense).
- Some point out that MIT-permissive license is GPL-compatible.
- The developer decides to avoid confusion and relicenses hg-app under GPL.

... & the Community Goes On.

- The project is renamed to RhodeCode.
- Developers from Mercurial community begin contributing under GPL.
- Independent contracts for software improvement are available for some.
- All is well as a traditional Free Software community, until ...

Altering The Deal

- RhodeCode's primary author forms a company, RhodeCode GmbH (The Company).
- The Company announces future versions of RhodeCode won't be GPL'd.
  - adds a "20 user max" code.
- Community backlash, and threats from all sides.
  - Author of the patch that removes the 20 user max is attacked.

Conservancy as Mediator

- Conservancy is contacted since Mercurial is a member project.
  - & RhodeCode's codebase is based on GPL'd Mercurial code.
- Some Mercurial developers and other community members seek aggressive enforcement action.
- Conservancy seeks calm conversation with the Company.
- The Company makes it clear that they believe it's 100% their copyright.
- Situation quickly becomes a stand-off.

Of Course, GPL Is Irrevocable!

- Of course, old code cannot be un-GPL'd.
  - AFAIK, no one has legitimately disputed that.
- The question is with regard to future copyrights generated by the Company:
  - Can they relicense their *future* copyrighted works under non-GPL terms?
- It's complicated:

> All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met.
>
> — GPLv3§2¶1

"Based on" the earlier work

- This codebase is, in Conservancy's view, GPL'd:
  - first, it's based Mercurial.
  - second, some third-party authors contributed patches under GPL.
- The Company just doesn't agree.
- Friendly negotiations to reach a fully GPL'd version going further failed.
- So what are the options?

How This Differs from Other Violations

- Some GPL enforcement efforts reach the point of fundamental legal disagreement about the requirements of GPL.
- In most cases, the stand-off is a true stand-off:
  - Free Software community doesn't have necessary CCS for the product.
  - A lawsuit eventually becomes only remaining way to compel CCS release.
- But we don't have a mundane violation here.

Early Friendliness Yields Better Options

- Ultimately, the Company was, at one point, a good actor:
  - They shared kindly their software under the irrevocable GPL.
- That code release remains GPL'd and useful.
- In the worst case, forking at that point is an option.
  - & avoids conflict.

Avoiding Conflict

- Conservancy has done GPL litigation before.
- Lawsuits are the last resort; they takes far too long.
- When licensing of the primary codebase remains uncertain during the case:
  - you've effectively repeated USL v. BSDi, and tied your codebase up in court for years.
- Conservancy had better options here, thanks to earlier GPL release ...
  - ... plus partial release of more recent code under GPL.

A Complex License

- The Company's final text of their license is clearly ambiguous (or worse):
- GPLv3, in fact, contemplated this problem:

RhodeCode system is split-licensed and comprised of two parts: (1) The Python code and integrated HTML are licensed under the GPLv3 license as is RhodeCode itself. ... (2) All other parts of RhodeCode including, but not limited to, CSS code, images, and design are licensed according to the license purchased.

Invalid Additional Term?

> All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term.
>
> — GPLv3§7¶4

Remember: No Conflict!

- Obviously, the Company would fight Conservancy's GPLv3§7¶4 claim.
- Instead, to make peace, Conservancy reads the license more conservatively than necessary.

RhodeCode system is split-licensed and comprised of two parts: (1) The Python code and integrated HTML are licensed under the GPLv3 license as is RhodeCode itself. ... (2) All other parts of RhodeCode including, but not limited to, CSS code, images, and design are licensed according to the license purchased.

Making the Fork

- Thus begins: the hard work of a four step process:
  - Find the last known version of the codebase without the new (invalid) text added.
  - Extract useful patches of only Python code and HTML files from post-license-change versions.
  - Rebrand to a new name.
  - Ensure "beyond reproach" compliance.

Step 0: Last Known GPLv3'd version

- First step was easier than it looked.
- Public development had stopped completely when the controversy started:
  - i.e., the project became "throw it over the wall"
- The last released Mercurial repository was basically what we sought.

Step 1: Extract New Python & HTML

- Even with hyper-conservative reading, Python code & HTML are clearly GPLv3'd.
- Careful extraction was required to include only those changes.
- We wrote a shell script to rerun mercurial commands and verified resulting repositories included only new changes to Python & HTML files.

Step 3: Rename

- Won't use the (potentially trademarked) brand name, "RhodeCode"

  - ... other than in those ways third parties are already permitted to do so.

- Picking names is always hard.
  - We decided on: "Kallithea"

  Kallithea, or Καλλιθεα, is the name of a locality on the island of Rhodes, in Greece, which itself means 'the best view'. Our Kallithea project helps developers get the best views of their project and its contributions so they can better collaborate together.

Step 3: Rename

- Problem: rhodecode_ was commonly used throughout the code.
- Hyper-conservative approach requires removal.
  - This is of course mostly functional use, so perhaps unnecessary.
- Obvious sed/perl scripts require care:
  - You have to make sure your replacements generate working code!

Step 4: Beyond Reproach

- Even if upstream violated the GPLv3, it doesn't mean we have permission to do so.
- Therefore, getting CCS right is a complicated issue.
- Why is getting the CCS right difficult in this case?

GPL'd Javascript

- Most people don't realize how similar Javascript programs are to mundane software distribution:
  - You publish a .js file on a URL.
  - Your users download it with an HTTP request.
  - You've *distributed* (in GPLv3 terms, *conveyed*) the software to them.
- Every GPL requirement, including those relating to CCS provisioning *apply* fully for most Javascript in a GPL'd system!

So, What Do I find?

So, there's Javascript in this thing?

```
$ hg clone -q https://kallithea-scm.org/repos/kallithea/
$ cd kallithea; hg update -C ffd45b185016
$ ls -1 rhodecode/public/js/
codemirror.js
codemirror_loadmode.js
excanvas.min.js
qgraph.js
jquery.1.10.1.min.js
mergerly.js
mode
native.history.js
pyroutes_map.js
rhodecode.js
yui.2.9.js
yui.flot.js
```

Verify Upstream licensed this code

```
$
$ hg -v log -r bb9ef0638069
changeset:   4120:bb9ef0638069
branch:      rhodecode-2.2.5-gpl
user:        Bradley M. Kuhn <bkuhn@sfconservancy.org>
date:        Fri May 16 15:54:24 2014 -0400
description:
  Update CodeMirror CSS and Javascript files to version 3.15, under MIT-permissive license.

  These files are exactly as they appear the upstream release 3.15 of
  Codemirror, which was released under an MIT-permissive license.  To extract
  these files, I did the following:

  I downloaded the following file:
       http://codemirror.net/codemirror-3.15.zip
  with sha256sum of:
  $ sha256sum codemirror-3.15.zip
  8cf3a512899852fd4e3833423ea98d34918cbf7ee0e4e0b13f8b5e7b083f21b9  codemirror-3.15.zip

  And extracted from it the Javascript and CSS files herein committed, which
  are licensed under the MIT-permissive license, placing them into their
  locations in: rhodecode/public/{css,js}/
```

```
Using the procedure above, the only difference found between these files in
RhodeCode 2.2.5 release and herein were a few comments and whitespace.

Note that the file .../public/js/mode/meta_ext.js does *not* appear to be
part of CodeMirror and therefore is not included in this commit.
```

Add Correct LICENSE notices

Update our LICENSE.md:

```
Codemirror
----------

Kallithea incorporates parts of the Javascript system called
[Codemirror](http://codemirror.net/), which is primarily:

Copyright &copy; 2013 by Marijn Haverbeke <marijnh@gmail.com>

and licensed under the MIT-permissive license, which is
[included in this distribution](MIT-Permissive-License.txt).

Additional files from upstream Codemirror are copyrighted by various authors
and licensed under other permissive licenses.  The sub-directories under
[.../public/js/mode/](kallithea/public/js/mode) include the copyright and
license notice and information as they appeared in Codemirror's upstream
release.
```

Most Javascript is Object Code

- That was easy compared to minified Javascript.
- Minified Javascript is Object Code.
- Must find appropriate CCS, & restart from scratch.

Most Javascript is Object Code

> "Object code" means any non-source form of a work.
>
> — GPLv3§1¶1
>
> Object code is not restricted to a narrow technical meaning and is understood broadly to include any form of the work other than

> the preferred form for making modifications to it. **Object code** therefore **includes** any kind of transformed version of source code, such as bytecode or **minified Javascript**.
>
> — FSF's GPLv3 First Rationale Document, 2006-01-16 (emphasis mine)

Why You?!? I (asked)!

- YUI 2.9 is Yahoo's (deprecated) Javascript interface library.
- Minified versions found on many websites ...
  - ... which isn't a violation ...
  - ... since YUI is 3-Clause BSD ...
  - ... but it's now part of a larger GPLv3'd work's CCS.

> The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities.
>
> — GPLv3§1

Handling Our Distribution

What about distributing minified Javascript in our repository?

> Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code.
>
> — GPLv3§6(c)

YUI Solution

From our LICENSE.md:

```
Kallithea incorporates parts of the Javascript system called
[YUI 2 - Yahoo! User Interface Library](http://yui.github.io/yui2/docs/yui_2.9.0_full/),
which is made available under the [BSD License](http://yuilibrary.com/license/):

Copyright &copy; 2013 Yahoo! Inc. All rights reserved.
[Text of 3-Clause BSD]

Kallithea includes a minified version of YUI 2.9. To build yui.2.9.js:
    git clone https://github.com/yui/builder
    git clone https://github.com/yui/yui2
    cd yui2/
    git checkout hudson-yui2-2800
    # work around inconsistent casing
    ln -sf JumpToPageDropDown.js src/paginator/js/JumpToPageDropdown.js
    rm -f tmp.js
    for m in yahoo event dom connection animation dragdrop \
             element datasource autocomplete container event-delegate \
             json datatable paginator; do
      rm -f build/\$m/\$m.js
      ( cd src/\$m && ant build deploybuild ) && \
        sed -e 's,@VERSION@,2.9.0,g' -e 's,@BUILD@,2800,g' build/\$m/\$m.js >> tmp.js
    done
    java -jar ../builder/componentbuild/lib/yuicompressor/yuicompressor-2.4.4.jar \
     tmp.js -o yui.2.9.js

In compliance with GPLv3 the Corresponding Source for this Object Code is made
available on [https://kallithea-scm.org/repos/mirror](https://kallithea-scm.org/repos/mirror
```

Lessons for New Communities

- For a web application, *don't* just copy Javascript (even Free Software stuff) into your repository.
- When you start contributing to a project, *ask* who holds domain name & trademark for the project.
  - If the answer is one person or I don't know, find a non-profit like Conservancy to help *right away*.
- Keep your own copyrights & make it clear you expect the license to be upheld.
  - & will do so yourself if needed.

More Info / Talk License

- URLs / Social Networking / Email:
  - Kallithea Project: https://kallithea-scm.org.
  - A Book I helped write: *Copyleft and the GNU General Public License: A Comprehensive Tutorial* is available.
  - Conservancy: sfconservancy.org & @conservancy
  - Me: faif.us & ebb.org/bkuhn
  - Slides: ebb.org/bkuhn/talks & gitorious.org/bkuhn/talks (source)
  - DONATE: https://sfconservancy.org/donate/
  - Sign Up Today as part of the *GPL Compliance Project for Linux Developers*.

Presentation and slides are: Copyright © 2014 Bradley M. Kuhn, and are licensed under the Creative Commons Attribution-Share Alike 4.0 International License.

Some images included herein are ©'ed by others. I believe my use of those images is fair use under USA © law. However, I suggest you remove such images if you redistribute these slides under CC-By-SA 4.0.