Does IP matter with Open Source Software?

Licensing and the open source philosophy

Introduction

Hello, my name is Martin Callinan, director at Source Code Control. In this module, we are going to define what we mean by open source software and discuss the different types of open source licences, their obligations and the potential impact on businesses.

When people talk about open source software, they are either referring to the software development model, where source code is freely available, can be copied, modified and reutilised in applications; or they are talking about the commercial model and licensing of open source software. It's the latter that we are going to discuss in this module.

Now there are two organizations that have become the definers of open source software. The first is the Free Software Foundation, which was formed as a non-profit in 1985 by Richard Storman. Then there is the Open Source Initiative, which was formed in 1998 and they are the definers of the open source definition and have become the approvers of open source licences that conform with that definition.

The Free Software Foundation has come up with what's known as the four freedoms. It's numbered 0 to 3, as it was created by software developers and software code starts at number zero. Now the freedoms that are talked about are all about freedoms around how software works – the freedom to be able to distribute that software, the freedom to be able to modify the software and redistribute that software. Although the term free is used, it doesn't necessarily mean that the software is free of cost. In fact, one of the freedoms is, you can charge for your software or distribute it free of charge.

So the open source initiative and the open source definition has ten statements that make up that definition. It includes the four freedoms discussed by the Free Software Foundation, but it goes into a bit more detail about defining what is open source software and it includes things like you cannot restrict fields of endeavour. So for instance, there are open source licences that restrict usage in certain industries, such as if animal testing is involved or in defence and military. That will be restricting a field of endeavour and therefore would not conform with the open source definition.

Now there are over two thousand documented open source licences, each with different obligations and terms, which obviously will be difficult to manage. Fortunately, there is only a handful of licences that are very popular, but you do need to be mindful if all these other licences that could make their way into your source code. The most popular licences that you are likely to come across are things like the Apache licence, the GPL licence, which is the licence of the Linux kernel and things like BSD and MIT.

By definition, open source software licences make the source code available to be viewed, modified and re-distributed. You need to be aware that there are further obligations within these licences, such as the need to provide attribution to the original developers, potentially to distribute the source code of your software, if you use that software within your software and things like copyright notices.

I mentioned earlier that there are over 2000 open source licences with different obligations. There’s basically a spectrum of open source licences. At the easier end, there’s what’s known as permissive licences, so licences like the Apache licence, the BSD licence, MIT – these have very few obligations and are relatively easy to comply with. At the other extreme, you’ve got licences like the family of GPL licences, which is the licence of the Linux kernel, which are far more restrictive and impact commercially larger organizations.

So the term permissive really means minimally restrictive licence – so as an example, if we look at the BSD licence (there are different editions of the BSD licence) – basically with the BSD licence, you can use the software within your software and the obligations you have are to provide attribution to the previous developers of the software, copyright notices (so include the statement that it’s licensed under the BSD licence), and there is a clause (sometimes referred to as the advertising clause), where you can’t promote your software based on the original developers. So, for instance, if I take a piece of software by a brand named company and incorporate it in my software, I can’t advertise my software as being associated with that organization. With the BSD licence, there’s no obligation to share your source code where you’ve used the components within the software. Other examples of permissive licences are the Apache licence and the MIT licence.

So if we look at the structure of an open source licence, again looking at the BSD licence if you could compare this licence to a proprietary licence which sometimes goes on for pages, it’s a relatively straightforward licence, literally just one page. It has a grant broad of rights defining how you can use the software, the obligation to include attribution and copyright notices, a non-endorsement obligation (that is the advertising clause I mentioned previously), and also a disclaimer of liability (i.e. there are no warranties with this software).

Let’s look at another permissive licence – the Apache licence – which is a very popular open source licence. Starting with a broad grant of rights, which defines how you can use the software, it also explicitly calls out that you can license your software under a different licence but still use that Apache software within your software. It also defines how you can distribute the software and what exactly is a derivative piece of software.

So let’s look at how organizations comply with these obligations. Here we’re looking at the Facebook application on an Android device. If I go into the ‘About’ section, I will see a list of all the third party open source components that Facebook has used to build their application. It will include a list of all the third party components that have been used, their licences, the text of the licence and any attributions associated with those licences.

Here’s another example, this time on Apple iOS. If we go into the setting of the operating system, we’ll find a section listing all the third party components Apple has used to be able to build the operating system. In this example, we're looking at a piece of code licensed under the MIT and GPL, i.e. a dual licensed piece of code. Every component that's used to build an application needs to be listed, including their licences and copyright notices and attribution notices. In this example, I'm looking at Chrome browser. Again, if I go into the settings, I will see a list of all the components, how they're licensed and any copyright and attribution associated with those licences.

So, that was permissive licences, which I said at the beginning were relatively easy to comply with. Now, we're going to talk about the more complex licences, more restrictive licences, sometimes referred to as copyleft or reciprocal licences.

So the obligations that make this difficult for companies to comply with – there are two main obligations – First of all, there is an obligation, if I use one component licensed under a restrictive licence, I have to licence my software under the same licence terms. Secondly, there is an obligation to make the source code of my software available to any users of the software. Now if the IP value of my company is based on the value of the software, naturally I don't want to share the source code. So organizations that have no desire to share their source code should not be using copyleft or reciprocal licences within their code.

The term copyleft came about because the authors of the licence used copyright law to add terms of distribution which enforce the requirements to make the source code always available and also to reciprocate the licence. Now if I refer back to the Free Software Foundation, their desire that software should always be open, they clearly advocate copyleft licences.

So let's have a look at how organizations comply with restrictive licences. Here, I'm looking at the VLC Media Player, which is a very popular application, I'm looking at the Android app. Again, if I go into the settings, I can see a list of all the third party components, how they're licensed and any attribution and copyright notices. I can also see that there is a link to the source code of the software. That means that I can go to that link, I can view the source code, I can modify the source code and I can redistribute that source code. Any re-distribution that I make of any modifications, I also need to license under the same licence, which is the GPL and also I have to make any modifications that I make available on request.

A further complication with open source licences, is that some licences allow you to add additional terms to the licence. So these are bespoke terms to your organization. Now there are tools that scan source code and identify licences – and typically they would only identify licences like the GPL and the Apache licence. If there are additional terms, they typically will not be discovered by a tool and therefore you could be distributing software with terms that could be an issue to your organization.

So how would you comply if you have got a physical device, say with embedded software, how do you comply with the obligations? Typically, organizations would have a section on their website which lists the third party components. If there is an obligation to share the source code, there will be a form, where you complete the details and the source code will be shipped to you, either as a download from the website, or in physical format on a DVD.

So let’s have a look at how these obligations can cause issues for organizations. In 2016, a company called Jide were launching a consumer operating system called Remix, which is based on the Chrome operating system. So, in early January, they started doing their PR, the product was in beta and it was getting great reviews. Software developers started requesting a copy of the source code for the Remix operating system on the support forum on their public website. There were clear responses from Jide stating that this is not an open source project and, unless you were a partner, you can’t have access to the source code.

However, when installing the beta, there’s clear reference to the GPL licence. The GPL licence we covered earlier is one of these restrictive licences with an obligation to make the source code available. So there literally could be just one component in Installer which creates this obligation for Jide to share the Remix operating system source code.

Very soon, journalists started getting wind of this issue and there were headlines such as ‘Are Jide with their Remix Operating System violating the GPL licence?’ which clearly aren’t great prior to launch of a piece of software. Two days later, there was a press release from Jide, stating that they take software licensing seriously and how to access a copy of the source code.

Now the speed with which they reacted could cause an issue in its own right. So, for instance, in a matter of days, there would not be time to go through the source code looking for things like derogatory comments in the source code put there by developers, looking at security vulnerabilities in the code, and any other particular issues that might cause embarrassment to the company.

Another example is the BMW i3. Now the average car today has over 100,000 lines of code – most of that code is open source software. Now if you go into the settings of the media player of the i3, there is a clear reference to the GPL licence and also how you request the source code. So a developer walked into a BMW dealership in Australia and requested the source code. And clearly he got blank looks from the salespeople in the dealership, and eventually it was escalated to BMW’s head office. Not before there was a lot of noise on the Internet about BMW not complying with the GPL licence. Eventually BMW shipped a DVD with the software for the media player of the i3 and it included components from Nvidia, Wind River and other software vendors. That code has subsequently been uploaded to the GitHub website, so anybody can go onto GitHub and view that source code, modify that source code and re-distribute that source code.

So that concludes the module introducing what is open source software and the variety of open source software licences. In the next module, we’ll delve deeper into open source software licensing.