# Does IP matter with Open Source Software?

## Getting it wrong with open source

### Introduction

Hi, I'm Paul McAdam and in this module, we're going to be looking at the organizational impacts of getting things wrong with open source software.

In the previous module, we looked at open source software and the way you should manage it with good practice. In this module, we'll be looking at ways in which things can go wrong if you don't get that management quite correct.

The impacts are wide ranging. They can range from copyright infringement, facing security issues, being fined by regulators, facing some bad publicity or issues that can affect the company value - from perhaps struggling to raise funds or facing issues with trying to IPO the organisation. So, in the previous module, we covered off proprietary software and many of you will be aware of the order clauses which exist within those proprietary software contracts. Now the same construct doesn't exist within open source software. There's no big bad wolf organisation seeking to enforce those order contracts in quite the same way. We have, however, seen an increase in activity amongst intellectual property trolls – and what they do as individual copyright owners of components within something, for example, say Linux, they will go the courts in countries such as Germany, where is relatively cheap to obtain a cease and desist order in one of law courts. They will take then that cease and desist order to an organization who they believe is infringing their copyright, on say, a particular component. They then ask to apply the cease and desist order to that software publisher, who then obviously has to stop their activities… But not just them, but also their customers. The best way then for the customer to quickly get out of that cease and desist order is to negotiate with the copyright holder on that particular module. Unfortunately then, what often happens, is that the copyright holder will then seek to discuss a number of other modules in which they are similarly attributed as the copyright owner.

Within case law, in the US, there are a couple of very important cases which have been established. The first of those is Artifex and Hancom. Now Artifex were the copyright holder of an item called Ghostscript, which it felt Hancom had incorporated into their product without adhering to the terms and conditions of the GPL contract. In other words, Hancom hadn't been publishing their source code. The case made it to the law courts in the US and they managed to establish a couple of key principles – two very important principles.

First of all, they established that the GPL licence and not correctly adhering to the GPL licence was a breach of contract within the US. The second aspect which was established in the case was that of the size of the damages. What was a relatively simple piece of open source software, actually the damages which were incurred by Hancom was calculated on the basis of the alternative proprietary pricing and equated back to the open source software. SO what was

relatively simple in open source became a very expensive set of damages based on proprietary costs.

The second case in case law was quite a complex one, but quite entertaining. So, an organization called XimpleWare created a piece of open source software. Now, along came Versata – and Versata used the XimpleWare code in their product. But what they did is they stripped out and removed the copyright statements. Versata then sold their product to Ameriprise Financial. Ameriprise took the Versata code and distributed it out to thousands of their contractors. Ameriprise then at some point became unhappy with the costs that they were incurring paying for the Versata licensing and they decided to write their own code to replace Versata's code, feeling that it was just getting too expensive. Versata then sued Ameriprise as part of their attempts to [hold back their] move off their software and on to their own proprietary software. Now, then, during the legal discovery process, Ameriprise discovers that the XimpleWare software is used within the Versata code without any recognition or without any attribution or consistency with the copyright which belongs to XimpleWare. Thinking this is a great tactic, Ameriprise shares this discovery with XimpleWare, intending to, effectively gang up on Versata. XimpleWare, however, finds itself in an interesting position, because XimpleWare has the right to sue not only Versata but also any of Versata's customers – because none of them have correctly adhered to the copyright requirements. Included in those customers, of course, is Ameriprise. Versata alone is now facing possible damages of in excess of $150 million.

The next case is another interesting one. This is CoKinetic and Panasonic Avionics, two organizations who are involved in the distribution of those little video screen that you sit and watch when you're on a long haul flight. Now, this case is a GPL licence infringement. Both organizations were using Linux within the creation of their offerings and what CoKinetic has done, is to bring a $100 million case against Panasonic on the grounds of unfair business practices. They believe that they were basically absorbing technology from their open source market not correctly attributing that software and distributing, in this case Linux software, along with their solution offering and, as a result, were making great advanced and gaining great market share without correctly adhering to copyright and contracts.

Another example of a high-tech organization who has fallen foul of the headlines would be that of Symantec. Their high end router product uses OpenWRT, which is based on Linux, and it appears to be that they haven't correctly adhered to the GPL licence and released their source code and made it available to those that request it.

There is some good news in this sector though… and perhaps a sign that open source software is maturing. Some of the biggest organizations in the world, IBM, Microsoft, RedHat, etc. have been getting together to standardise an approach to dealing with these complex licensing cases, by giving effectively a little bit of time to get things right before they take legal action.

So that covers off copyright, let's have a look at some of the security issues you might come across as a result of the mismanagement of open source software. In the US, there exists a National Vulnerability Database, which collects the reported vulnerabilities across the different

software platforms. There's over 100,000 vulnerabilities recorded with open source software. There's appearing in excess of ten new vulnerabilities each day – and in 2016 it was 4,300 and higher again in 2017. What we've found throughout our work is that very very few organizations have a process of recording, tracking, identifying and remediating these issues. And down the right hand side there, you'll even see that some of these vulnerabilities have attracted their own icons: Heartbleed, Shellshock, Ghost and Poodle.

Probably the world's most notorious or infamous hack ever, is the result of mismanagement of open source software. We're talking of course about Equifax with the release of 14.5 million records of personal data being released onto the Internet. The issue was caused by mismanagement of a single component, Apache Struts. The vulnerability was identified in March, and unfortunately it took through until May of the same year for that issue to be fully rectified by the company. In the meantime, all that data's leaked out and been made available on the Internet. What happened subsequently is that the CEO had to appear in front of the Senate and explain the situation. The company has experienced an absolute outrage from its customers and a series of bad publicity – and the CEO, obviously at some point his position became untenable and he had to resign. Now, I'm sure that as he was signing his resignation letter, he probably wasn't thinking, "I really wish I had checked all those Apache Strut implementations that we had," he was actually looking at the development team and saying "Surely someone must have been responsible for this," and "I can't believe that we weren't checking this on a regular basis." But, of course, that is exactly what had happened. The issue being that the individual components just were not being managed by the organization.

Now what's interesting for me with Gloucester City Council is that the original vulnerability for Heartbleed was discovered in 2014. The actual hack and the release of information of Council Tax payers was in 2016. Between those two dates, there was over one hundred new vulnerabilities identified in OpenSSL and twelve new versions of the software released. Quite simply, what was happening, was that there was no patching and update of the OpenSSL component – and again, unfortunately, mismanagement by the City Council.

Now, as an organization, we don't tend to lead with GDPR, we leave that for absolutely everybody else to do – but I'd just like to draw your attention to Article 32. Article 32 of GDPR says that an organization is required to ensure a level of security appropriate to the risk, including establishing processes for regularly assessing and testing security practices. It goes on to say that organizations will be required to implement technical and organisational measures to ensure a level of security appropriate to the risk. Now, all the processes we have been advocating, such as software composition analysis, would help any organization to understand what they have at any point in time and where the vulnerabilities may lie. The key point being, you can only protect what you know you actually have.

And then, finally, just to draw your attention to how mismanagement of components can affect the company value. So we do quite a lot of work with organizations looking to attract investment or with investors looking to invest in high-tech or software companies. We often find organizations are a little bit ahead of themselves and not quite ready for that investment discussion. They haven't thought about penetration weaknesses, where they have security vulnerabilities. They haven't thought about the copyrighting of their own code and they haven't

thought about licencing and correctly attributing the code of other people's software which they have in use.

Thank you for watching this module. In the next module we're going to be looking at some definitions of open source software and explaining exactly what it is.