Published on *The National Law Review* https://natlawreview.com

Is The Use of Open Source Software Putting Your Business at Risk?

Article By:

Samuel T. Tibbetts

Stuart James

A brief overview

La propriété, c'est le vol! (roughly translated as "property is theft!"). Perhaps the most famous assertion of Pierre-Joseph Proudhon, the French philosopher, considered by some to be the father of anarchy.

A contemporary of Karl Marx, Proudhon's focus was on physical property. However, this resonates with the early libertarian open source software philosophy, which encouraged the free distribution of software for the purposes of learning and evolving the field of computing.

Much has changed since a senior Microsoft executive once exclaimed that they could not imagine anything worse than open source software for the software and intellectual property business. What was once limited to a few discrete programmes shared between academics has now become a key resource for multi-national PLCs and start-ups, enabling businesses to react quickly and innovate using freely available flexible technology, as well as making significant cost savings for business and consumers.

With no small degree of irony, in 2019 Microsoft counted itself as the world's biggest open source contributor, paying $7.5 billion to acquire the software development platform GitHub in 2018—GitHub being one of the original promoters of open source software.

Some notable examples of open source software include:

- **Linux Operating System**: Google's Android, the world's most popular mobile device operating system, is based on the open source Linux operating system, which is developed and maintained by more than 15,000 developers around the world.

- **WordPress**: The popular website content management system is used by a wide range of organisations ranging from The Walt Disney Company, Capgemini, Unicef to even The White House.

- **TensorFlow**: Google open sourced its artificial intelligence engine to enable programmers to develop and enhance AI applications.

- **MySQL**: MySQL is a relational database management system *used*to create and manage *databases*. The application is used for a range of purposes, including data warehousing, e-commerce, and logging applications.

Open source software sounds great so why should I worry?

Using open source comes with some very significant catches. Two of the most notable being:

- **Vulnerabilities in open source software**. Vulnerabilities in open source software can expose a business to data breaches.

- **Requirement to redistribute source code**. In certain circumstances, open source licence terms may require the user to distribute the source code of modified versions of the particular open source software or programs based on that software. This can extend beyond the open source software itself to also require disclosure of the source code of valuable proprietary software incorporating the open source software.

Vulnerabilities

It would certainly not be fair to claim the issue of vulnerabilities is exclusive to open source software. Far from it. Even the most arrogant of coders is unlikely to be able or wish to claim their code was perfect with absolute confidence and a straight face. However, the National Vulnerability Database (NVD)—the U.S. government repository of standards based vulnerability management data—has identified more than 100,000 vulnerabilities in open source software with 5 new vulnerabilities being discovered every day. This also naturally begs the question regarding the number of known (and unknown) vulnerabilities skulking in the shadows that are not reported to the NVD.

According to a recent report undertaken by leading open source consultants WhiteSource, the number of reported vulnerabilities in open source projects rose by an eye-watering 50 per cent in 2019. This is further evidence of the proliferation of the use of open source software by commercial

enterprise. The rise is not necessarily all bad news though. The optimists would say that the fact that the vulnerabilities have been reported indicates a raised awareness of the potential pitfalls of using openly available source code—effectively, people are now looking for them!

However, although all software contains vulnerabilities, the key difference with open source software is that the underlying source code is widely available. Therefore, vulnerabilities in such code are easier to determine and, for the nefarious parts of the community, exploit. Furthermore, once a vulnerability is reported, that vulnerability becomes public knowledge giving such ne'er-do-wells' access to the backdoor keys left under the plant pot. It is therefore crucial that any known vulnerabilities in open source software are fixed quickly before that brand new 8k Ultra HD TV is stolen (well, that or a lot of very sensitive commercial information).

The pace of development and requirement to exploit new market opportunities means that many businesses will have been caught unaware, with few having developed detailed policies and procedures for tracking, identifying and remediating open source software issues. This is also far from an issue that only IT companies need to worry about as every company worth its salt uses computing software or technology, a portion of which may include open source software. There is a long list of recent and painful examples of hackers exploiting vulnerabilities in open source software used by companies in a range of sectors, including:

- **Aptoide (April 2020):** on April 17th, ZNet reported the theft and leak of personal data of more than 20 million users of Aptoide, a third-party store for Android applications. Aptoide is an open source platform with users able to download the code for the purpose of creating their own individual app stores within the platform (unlike e.g. the Android Google Play Store, which operates a single central marketplace).

- **Tupperware (March 2020):** cybersecurity company Malwarebytes recently claimed to have discovered a card skimmer on the Tupperware.com website, which it is believed arose due to Tupperware using an unpatched version of the open source Magento e-commerce software—see also Magento update below. Feel free to come up with your own puns….

- **Scotiabank (September 2019):** The bank stored highly sensitive data in publicly open and accessible GitHub repositories, exposing its internal source code, login credentials, and confidential access keys (although the bank claimed that no information affecting customers or employees was exposed).

- **Fortnite (August 2019):** Fortnite users were targeted by ransomware Syrk**, based on the open source Hidden-Cry malware. Syrk** masqueraded as a game hack tool for Fortnite but, on being downloaded, locked up the user's computer and demanded a ransom.

- **io (March 2019):** the email validation service used an unprotected open-source database to store its data, MongoDB, resulting in 809 million email addresses and phone numbers being exposed.

- **Magento (February 2019)**: the popular open-source e-commerce platform, Magento was targeted by criminals looking to steal personal and payment information. At least 1,000 websites running on the platform were targeted via a skimmer—a piece of code that is either directly injected into a hacked site or referenced externally and which tracks user input such as credit card numbers and passwords.

Other infamous examples:

- **The Panama Papers (early 2016):** Mossack Fonseca's client portal used Drupal, an open source content management system, which had at least 25 known vulnerabilities that were exploited by hackers. The firm's client login portal, which ran on Drupal, had not been updated since 2013.

- **Heartbleed Bug (July 2014):** The Heartbleed Bug is a serious vulnerability in the popular OpenSSL cryptographic software library. This Heartbleed Bug allows anyone on the Internet to read the memory of the systems protected by the vulnerable versions of the OpenSSL software. Gloucester Council was fined £100,000 by the ICO for failing to patch the bug which had been widely reported on months prior to the attack.

Requirement to redistribute source code of modified versions

Open source software by its nature includes protectable intellectual property rights, most notably copyright. Therefore, if a person wishes to use any open source software, that person will be obliged to comply with the associated licence terms.

The Open Source Initiative (OSI), founded in 1998, is the self-titled steward of the open source definition and the community-recognised body for reviewing and approving open source licences. The OSI has approved over 90 licences, with open source software also being made available on countless other licences. One of the most unusual of the non-approved licences being the chicken dance licence (requiring the user to video themselves performing a chicken dance for every 20,000 units distributed).

Chicken dance licence aside, a detailed analysis of open source software licencing provisions would be extremely lengthy (including source code availability, attribution and notices, reciprocity of treatment and other terms) and therefore not appropriate for this note. Instead, a few key points below.

Open source software licences commonly fall into, or between, two camps:

- **Copyleft**

Copyleft is the practice of offering people the right to freely distribute copies and modified versions of a work but with the stipulation that the same rights be preserved in derivative works down the line. Such licences typically require a user to make the source code of the entire programme available to such customers.

This is not necessarily limited to just the source code of the specific open source component but also potentially any work based on / incorporating that open source component. A company could therefore be required to also disclose certain of its own proprietary source code. Examples of copyleft licences include: GNU GPL 3.0 and GNU Affero GPL. The latter GNU Affero GLP was introduced to cover the use of open source software in software provided as a service; such use was not caught by the more traditional licences as they required a form of distribution for the copyleft licence terms to bite. Google is an example of a major company that has banned employees from using any open source software licenced on GNU Affero GPL due to the additional reach of these licence terms.

- **Permissive**

Permissive licences contain minimal requirements about how the software can be redistributed. Examples include: Apache (Google's preferred choice), MIT and BSD.

Some licences sit in the middle such as LGPL and Mozilla and it should also be noted that there can be multiple licences that apply to one piece of software.

The various licences use different terminology and are by no means clear. However, in the most basic terms, if a business incorporates open source software into any programme that it makes available to customers, it must comply with the licence terms and, in certain cases, may be required to disclose a lot more than it had bargained for when using that software.

What should we be doing about all this?

Clearly, the use of open source software is only going to grow. There are evident commercial benefits in terms of cost savings, driving innovation as well as the broader societal advantages of sharing knowledge and materials amongst a wider community. Therefore, what should we be doing about the use of open source software (OSS)?

A few points to consider:

- **You most likely use OSS**: this is not solely an issue for IT companies. It is extremely likely that your company uses a form of OSS be it your website, database management system, communications systems and/or through use of third party systems. In fact, companies not directly involved in the IT sector are potentially the most at risk having less knowledge of the potential threats—hackers will be more than aware of this vulnerability.

- **Identification of OSS**: Ignorance of the use of OSS will rarely result in continued bliss. It is therefore very important to ensure that a business has clear procedures in place regarding the use of open source software, including in relation to any systems which store, process, and/or transmit cardholder data (see also our blog on PCI DSS compliance). As a minimum,

these procedures should include a process for:

- Identifying and vetting the use of OSS by the company. This will include:
  - implementing a vetting and approval process, including the development of internal rules for approval and use of OSS;
  - identifying the licence terms that apply to any OSS. In particular, if any OSS may be made available to third parties, it is important to understand the licence terms that apply and whether the company is willing to comply with such terms. E.g. may the use of the OSS require distribution of any other proprietary code and, if so, is the company comfortable with that requirement?
  - training of staff in relation to issues regarding the use of OSS, particularly developers and individuals involved in the procurement of IT systems
  - incorporate OSS best practices into the software development/build processes for IT companies, including ensuring the use of the most recent versions of OSS libraries that contain the most-up to date vulnerability patches.
- monitoring and patching any known vulnerabilities in the existing OSS used by such company.
- continual OSS license compliance reviews to identify conflicting OSS license terms and regular OSS audits.

For IT companies early identification is crucial as the costs and time required to resolve issues escalates dramatically between the initial coding phase and after release. The cost to fix the same defect can vary from tens of pounds / dollars to tens of thousands of pounds / dollars depending on when an issue is addressed in the timeline!

For non IT Companies, in addition to the points identified above regarding vulnerabilities and distribution of source code, it is also important to note that OSS is typically provided on an “as is” basis, without any commitments regarding quality and performance or that the OSS does not infringe any third party rights. Any use is therefore very much at the user’s own risk.

- **Acquisition of IT Systems**: to the extent possible, seek to address OSS in all relevant contracts for the acquisition of IT systems, including warranties / commitments that:
  - the systems made available by the supplier will not contain any OSS unless otherwise specifically agreed. This may be harder to obtain if the system is cloud based.
  - if it is agreed that OSS will form part of the systems: (i) the particular OSS libraries (including specific versions) is clearly identified; (ii) no OSS shall be used if it is made available on the basis of “copyleft” licence terms; and (ii) any additional terms that apply have been identified and are accepted.
  - the systems will be checked for and do not contain any reported vulnerabilities or conflicting OSS license terms, together with a commitment to fix any new reported

vulnerabilities or resolve any such conflicting terms.

- **Corporate Acquisitions**: to the extent possible, seek to address OSS in corporate acquisitions, including appropriate warranties and indemnities. In particular:
    - in an IT related acquisition, the use of OSS could potentially have a very material impact on the underlying value of the transaction.
    - vulnerabilities in core systems used by both tech and non-tech businesses could result in significant exposure to hacks, including resulting losses and costs of remediation (not only resolving the data breach but also resolving issues with the system).
    - A purchaser may also wish to consider engaging a specialist consultancy to undertake a detailed analysis of OSS used by the target business as part of the due diligence exercise.

National Law Review, Volumess X, Number 127