Is the future of open source software at risk due to protestware?

February 23 2024, by Stuart Pallister

Credit: Pixabay/CC0 Public Domain

SMU Associate Professor Christoph Treude examines the foundations for studies on open-source software and protestware.

"Software developers don't develop everything from scratch," he says. "Just like car manufacturing, you rely on pieces that have been manufactured by others. So, it's the same with software developers, whether in the open source world or industry. They tend to re-use a lot of stuff that others have done."

Open source ecosystems can contain millions of individual items. So what happens if someone adds malware to their particular piece of software to protest, say, the war in Ukraine? Well, that has happened, with the result that some users in Russia and Belarus have had their computers hacked.

For instance, the developer behind software library node-ipc with its more than a million weekly downloads tried to replace all the files on the computers of users in Russia and Belarus with a heart emoji back in March 2022.

"Because of the interconnectedness of the software ecosystem, people who contribute or maintain just one piece of the gigantic puzzle can have quite a bit of power."

Sometimes, a maintainer, the main person driving an open source project, may make an honest mistake when developing software, Professor Treude says. "But more recently, with the war in Ukraine, if maintainers want to raise awareness about something specific, they turn their open source project into malware." In extreme cases, he says, "they've re-programmed the library purposefully to attack machines located in Russia and Belarus."

Others take less drastic action and merely introduce a message or document "urging support for whatever side they're on."

Identifying the main types of protestware

In a paper titled 'In War and Peace: The Impact of World Politics on Software Ecosystems', which was presented at a software engineering conference more than a year ago, Professor Treude and his co-researcher Raula Gaikovina Kula from Japan's Nara Institute on Science and Technology identified three main types of protestware:

1. Malignant protestware—software that intentionally damages or takes control of a user's computer without their knowledge or consent.

1. Benign protestware—software created to raise awareness of a political or social issue but does not take control of the user's device.

1. Developer sanctions which affect a software ecosystem more broadly. For instance, MongoDB decided not to sell its products to Russian users, and GitHub suspended Russian accounts.

'A loss of trust'

Professor Treude says the role of open source in software engineering has shifted over the past decade. In the early days, major corporations such as Microsoft were opposed to open source software "as they believed software should be sold for money and should not be available to everybody for free." However, Microsoft eventually became a major contributor to open source, maintaining its own libraries.

"Of course, open source doesn't require you to pay anything when you use it. That's the whole idea. But then maintainers realized that companies are making money based on the code they've written." In addition, he says, companies are also attaching business licenses to the open-source software "to ensure that others can't make money from it."

"It's just against all the fundamental ideas of open source, but maybe we're seeing a reconfiguration of how all this fits together, and a lot of it has to do with a loss of trust."

And, presumably, the emergence of protestware has also led to a loss of trust in open-source software.

"Absolutely. We're wondering if we're slowly seeing the signs of the heyday of open source ending," Professor Treude says. "Maybe not ending, but people are getting less excited about it because of incidents such as protestware."

Perhaps, then, some sort of certification process may be required to establish the provenance of open-source software.

"That's a really interesting question because, on the one hand, that goes against some of the core ideas of open source that everything is free and open to everybody. On the other hand, maybe we do need some sort of liability when things go wrong."

However, open source licenses do not "necessarily tell you if you're allowed to develop protestware or not," and there is no legal liability.

Another option is that because there is so much information about programmers available on the internet, end users should do some digging before downloading software.

"You can find out which libraries they maintain and have contributed to, how long they've been active, how reliable they are, and how quickly they've fixed security vulnerabilities and so on. There's a whole research area in my field called mining software repositories, which focuses on questions such as these."

"Because we do have the data available, we might be able to use that information to help establish trust. Just by looking at someone's history, you can't necessarily predict the future, but it helps."

Promoting ethical responsibility

Professor Treude's latest paper, "Ethical Considerations Towards Protestware", which has been accepted by a practitioner magazine, examines several different ethical frameworks in relation to protestware:

Duty Ethics—effectively a sense of duty (linked to Kant's Categorical Imperative) which would mean, by implication, that the injection of malware would not be ethical.

Consequential Ethics—you should take into account the consequences of your actions (and seek to maximize overall happiness). Although the damage caused by malware might be seen as short-term, its long-term effects would be much larger.

Principlism—taking a more pragmatic approach by following a fixed set of principles such as respect for autonomy and justice.

This paper, written with co-researchers Marc Cheong of The University of Melbourne and Raula Gaikovina Kula, explores various ethical frameworks, but Professor Treude says there is no right answer.

"There is the whole concept of 'do no evil' which stuck with me when we were writing this up. If you know that what you're doing will have negative consequences for a lot of people, I would personally think twice about doing it, but obviously, there are people who think the greater good is more important."

The role of education

Education could play a crucial role here, Professor Treude says. "A lot of computer science and software engineering programs, even at universities, barely cover ethics these days, and even if they do, they're often dealing with topics with no connection to computer science. But in protestware, we have something concrete that anybody with a software development background would be able to relate to. So, I think using examples like this in education would be useful."

"I was hired specifically to look at the human and social aspects of software engineering. Software development is changing quickly at the moment thanks to AI (artificial intelligence) programs like ChatGPT, and a lot of the progress has focused on the technical side. But very few look at how we can actually empower humans to write better software. How can we get that interface between humans and AI right? That's what I see myself working on for the next little while."

Both articles mentioned in this story are published on the *arXiv* preprint server.

More information: Raula Gaikovina Kula et al, In War and Peace: The Impact of World Politics on Software Ecosystems, *arXiv* (2022). DOI: 10.48550/arxiv.2208.01393

Marc Cheong et al, Ethical Considerations Towards Protestware, *arXiv* (2023). DOI: 10.48550/arxiv.2306.10019

Provided by Singapore Management University