'Protestware' is on the rise, with programmers self-sabotaging their own code. Should we be worried?

September 28 2022, by Christoph Treude

Credit: Alexander Sinn/Unsplash

In March 2022, the author of node-ipc, a software library with over a million weekly downloads, deliberately broke their code. If the code discovers it is running within Russia or Belarus, it attempts to replace the contents of every file on the user's computer with a heart emoji.

A software library is a collection of code other programmers can use for

their purposes. The library node-ipc is used by Vue.js, a framework that powers millions of websites for businesses such as Google, Facebook, and Netflix.

This critical security vulnerability is just one example of a growing trend of programmers self-sabotaging their own code for political purposes. When programmers protest through their code—a phenomenon known as "protestware"—it can have consequences for the people and businesses who rely on the code they create.

Different forms of protest

My colleague Raula Gaikovina Kula and I have identified three main types of protestware.

- **Malignant protestware** is software that intentionally damages or takes control of a user's device without their knowledge or consent.
- **Benign protestware** is software created to raise awareness about a social or political issue, but does not damage or take control of a user's device.
- **Developer sanctions** are instances of programmers' accounts being suspended by the internet hosting service that provides them with a space to store their code and collaborate with others.

Modern software systems are prone to vulnerabilities because they rely on third-party libraries. These libraries are made of code that performs particular functions, created by someone else. Using this code lets programmers add existing functions into their own software without having to "reinvent the wheel."

The use of third-party libraries is common among programmers—it speeds up the development process and reduces costs. For example,

libraries listed in the popular NPM registry, which contains more than 1 million libraries, rely on an average of five to six other libraries from the same ecosystem. It's like a car manufacturer who uses parts from other manufacturers to complete their vehicles.

These libraries are typically maintained by one or a handful of volunteers and made available to other programmers for free under an open-source software license.

The success of a third-party library is based on its reputation among programmers. A library builds its reputation over time, as programmers gain trust in its capabilities and the responsiveness of its maintainers to reported defects and feature requests.

If third-party library weaknesses are exploited, it could give attackers access to a software system. For example, a critical security vulnerability was recently discovered in the popular Log4j library. This flaw could allow a remote attacker to access sensitive information that was logged by applications using Log4j—such as passwords or other sensitive data.

What if vulnerabilities are not created by an attacker looking for passwords, but by the programmer themselves with the intention to make users of their library aware of a political opinion? The emergence of protestware is giving rise to such questions, and responses are mixed.

Ethical questions abound

A blog post on the Open Source Initiative site responds to the rise of protestware stating "protest is an important element of free speech that should be protected" but concludes with a warning: "The downsides of vandalizing open source projects far outweigh any possible benefit, and the blowback will ultimately damage the projects and contributors responsible."

What is the main ethical question behind protestware? Is it ethical to make something worse in order to make a point? The answer to this question largely depends on the individual's personal ethical beliefs.

Some people may see the impact of the software on its users and argue protestware is unethical if it's designed to make life more difficult for them. Others may argue that if the software is designed to make a point or raise awareness about an issue, it may be seen as more ethically acceptable.

From a utilitarian perspective, one might argue that if a form of protestware is effective in bringing about a greater good (such as political change), then it can be morally justified.

From a technical standpoint, we are developing ways to automatically detect and counteract protestware. Protestware would be an unusual or surprising event in the change history of a third-party library. Mitigation is possible through redundancies—for example, code that is similar or identical to other code in the same or different libraries.

The rise of protestware is a symptom of a larger social problem. When people feel they are not being heard, they may resort to different measures to get their message across. In the case of programmers, they have the unique ability to protest through their code.

While protestware may be a new phenomenon, it is likely here to stay. We need to be aware of the ethical implications of this trend and take steps to ensure software development remains a stable and secure field.

We rely on software to run our businesses and our lives. But every time we use software, we're putting our trust in the people who wrote it. The emergence of protestware threatens to destabilize this trust if we don't take action.

Provided by The Conversation