

Singapore Management University

## Institutional Knowledge at Singapore Management University

---

Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

---

4-2024

### Going viral: Case studies on the impact of protestware

Youmei FAN

Dong WANG

Supastsara WATTANAKRIENGKRAI

Hathaichanok DAMRONGSIRI

Christoph TREUDE

Singapore Management University, [ctreude@smu.edu.sg](mailto:ctreude@smu.edu.sg)

*See next page for additional authors*

Follow this and additional works at: [https://ink.library.smu.edu.sg/sis\\_research](https://ink.library.smu.edu.sg/sis_research)



Part of the [Information Security Commons](#), and the [Software Engineering Commons](#)

---

#### Citation

FAN, Youmei; WANG, Dong; WATTANAKRIENGKRAI, Supastsara; DAMRONGSIRI, Hathaichanok; TREUDE, Christoph; HATA, Hideaki; and KULA, Raula Gaikovina. Going viral: Case studies on the impact of protestware. (2024). *ICSE-Companion '24: Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion: Lisbon, April 14-20*. 308-309.

Available at: [https://ink.library.smu.edu.sg/sis\\_research/8924](https://ink.library.smu.edu.sg/sis_research/8924)

This Conference Proceeding Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email [cherylids@smu.edu.sg](mailto:cherylids@smu.edu.sg).

---

**Author**

Youmei FAN, Dong WANG, Supastsara WATTANAKRIENGKRAI, Hathaichanok DAMRONGSIRI, Christoph TREUDE, Hideaki HATA, and Raula Gaikovina KULA



# Going Viral: Case Studies on the Impact of Protestware

Youmei Fan  
Nara Institute of Science and  
Technology, Japan  
fan.youmei.fs2@is.naist.jp

Dong Wang  
Tianjin University  
China  
d.wang@ait.kyushu-u.ac.jp

Supatsara Wattanakriengkrai  
Nara Institute of Science and  
Technology, Japan  
wattanakri.supatsara.ws3@is.naist.jp

Hathaichanok Damrongsiri  
Nara Institute of Science and  
Technology, Japan  
damrongsiri.hathaichanok.db5@is.naist.jp

Christoph Treude  
Singapore Management University  
Singapore  
ctreude@smu.edu.sg

Hideaki Hata  
Shinshu University  
Japan  
hata@shinshu-u.ac.jp

Raula Gaikovina Kula  
Nara Institute of Science and  
Technology, Japan  
raula-k@is.naist.jp

## ABSTRACT

Maintainers are now self-sabotaging their work in order to take political or economic stances, a practice referred to as “protestware”. In this poster, we present our approach to understand how the discourse about such an attack went viral, how it is received by the community, and whether developers respond to the attack in a timely manner. We study two notable protestware cases, i.e., Colors.js and es5-ext, comparing with discussions of a typical security vulnerability as a baseline, i.e., Ua-parser, and perform a thematic analysis of more than two thousand protest-related posts to extract the different narratives when discussing protestware.

*“Anyone who experienced actual significant disruption from this (protestware) brought it on themselves with their bad dev practices. No one forced anyone to install the latest version without actually verifying it at all. Didn't corrupt the version history so nope, people just letting their entitlement and lack of understanding of licenses show.”*

commentary on protestware

*“In dev-land, we don't stand on the shoulders of giants. We keep our life-rafts afloat by sticky-taping together skerricks of code that hopefully has more buoyancy than ballast. And sometime it just takes one person to take the whole ship down.”*

commentary on protestware

## KEYWORDS

Protestware, Software Ecosystems, Case Studies

### ACM Reference Format:

Youmei Fan, Dong Wang, Supatsara Wattanakriengkrai, Hathaichanok Damrongsiri, Christoph Treude, Hideaki Hata, and Raula Gaikovina Kula. 2024. Going Viral: Case Studies on the Impact of Protestware. In *2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion '24)*, April 14–20, 2024, Lisbon, Portugal. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3639478.3643086>

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

*ICSE-Companion '24, April 14–20, 2024, Lisbon, Portugal*

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0502-1/24/04...\$15.00

<https://doi.org/10.1145/3639478.3643086>

## 1 EXTENDED ABSTRACT

Open source software development has emerged as an unexpected platform for expressing social and political protests. In recent years, we have witnessed the emergence of “protestware”, defined as software that developers deliberately modify to express dissent and draw attention to issues they consider important [8]. Such protests can have significant social and technical impacts, especially given the interconnected nature of modern software systems.

Protestware takes on many forms. For example, the maintainer of ‘node-ipc’ used malicious code to target host machines with IP addresses in Russia or Belarus in response to the War in Ukraine [9]. GitHub declared this a critical vulnerability, known as CVE-2022-23812 [2]. This act underscores the political motivations that can inspire the creation of protestware. Other instances of protestware stem from industry concerns, such as the perceived exploitation of open-source labor by large corporations. For instance, the developer of the ‘faker’ library intentionally introduced an infinite loop, disrupting thousands of projects [3]. Today, ‘faker’ is a community-controlled project, maintained by a team of developers from various companies [1]. Such protests can have widespread consequences, particularly since modern software systems often rely heavily on third-party libraries, making them potential points of vulnerability [10]. To illustrate the scale of this interconnectedness, libraries

listed in the popular NPM registry, which hosts more than a million libraries, each depend on an average of five to six other libraries within the same ecosystem [4]. Beyond technical issues, the rise of protestware threatens the trust that is foundational to modern software ecosystems [5], creating disruptions within the software development community. Developers using a protesting library must decide whether to continue using it or take on the potentially complex task of finding alternatives [6] or reverting to older versions [7]. Societal reactions to protestware vary, with responses ranging from strong support to harsh criticism, and everything in between [9].

**Motivation.** The motivation that will be illustrated by this poster is to track and characterize disruptions caused by protestware and the immediate reactions it prompted. Concretely, we would like to understand the extent to which the software ecosystem is affected by discussions of the protestware, and what is the nature of the discussions.

**Approach.** In this study, we apply a comprehensive empirical study approach by combining quantitative analysis and later a systematic manual coding of the different protestware discussions. We conduct a large-scale empirical study by mining both the GitHub platform and social media (i.e., Twitter, Reddit, HackerNews) to extract discussions of the two case studies of protestware. We then conduct two analyses.

First, a quantitative analysis is conducted, focusing on the abandonment of protestware by developers. The approach involves identifying valid JavaScript repositories using protestware as a dependency, mining historical dependency changes, and retrieving popularity metrics using the GitHub API. The results reveal a lower likelihood that developers abandon protestware, with specific insights into the characteristics of influenced repositories and the distribution of the changes in dependencies. The spread of protestware discussions throughout the ecosystem is explored through two analyses, quantifying the speed of spread, and visualizing the evolution of disruption across GitHub.

For the second analysis, we extract characteristics of posts related to protestware using manual coding. We categorize posts into opinion-related and technical categories, classify opinion-related posts to identify the protestware stance, and examine technical posts related to mitigation. The overall methodology integrates both quantitative and qualitative approaches, providing a comprehensive understanding of the dynamics surrounding protestware and its impact on the developer community.

**Key results so far.** Our results indicate that both cases of protestware (Colors.js and es5-ext) have comparable disruption to that of a security vulnerability (i.e., Ua-parser). However, unlike a security vulnerability, developers are less likely to abandon the dependency. For example, of the 146 repositories that discussed the Colors.js issue, only 15% abandoned the dependency, compared to 82% by dependents of Ua-parser. Measuring the spread, we find that the news of protestware can be broken through social media outlets (i.e., Colors.js), and may have a greater reach in terms of developers and repositories (Colors.js) compared to Ua-parser, taking two days less to reach 2,000 posts, and almost 53 days less to reach 10,000 developers. Additionally, it should be noted that Twitter seemed to

break the news of the protest, even before the original issue was created on GitHub. For example, the first tweet,<sup>1</sup> directly asked the protester about the injected protest code.

By categorizing posts into opinions and technical discussions, we show that Colors.js demonstrated a mixture of opinions (50.3%) and technical discussions (44.6%). Developers impacted by es5-ext, on the other hand, predominantly engaged in expressing their opinions (77.3%) rather than focusing on technical aspects (22.6%). Furthermore, we identified four main themes in the opinions expressed during the discussions: 1. *OSS Philosophy*, 2. *Legal Issues and Rights*, 3. *Trust and Reputation*, and 4. *Supportive or Dogmatic responses*.

**Looking Ahead.** At this early stage, it is unsure whether or not protestware is here to stay in the open source software ecosystem. With the heavy reliance on the ecosystem and its supply chain, the mere presence of protestware is a reminder that software development is becoming increasingly intertwined with society and its issues, ranging from economic to political. In this first large-scale exploration into protestware, we present the different narratives that drive this emerging phenomenon. These future research opportunities not only include providing developers with insight into varying perspectives but also offer actionable guidance to foster constructive dialogue and reduce toxicity in these sensitive discussions. Such frameworks could transform protestware from a potential point of contention into an instrument for positive social change, aligning the open-source community with broader societal values and goals.

## ACKNOWLEDGEMENTS

This work was supported by JST SICORP Grant Number JPMJSC2206.

## REFERENCES

- [1] [n. d.]. An update from the Faker team | Faker – fakerjs.dev. <https://fakerjs.dev/about/announcements/2022-01-14.html>. [Accessed 20-Jul-2023].
- [2] 2022. *CVE-2022-23812*. <https://nvd.nist.gov/vuln/detail/cve-2022-23812>
- [3] Steven M Bellovin. 2022. Open Source and Trust. *IEEE Security & Privacy* 20, 02 (2022), 107–108.
- [4] Bodin Chinthanet, Brittany Reid, Christoph Treude, Markus Wagner, Raula Gaikovina Kula, Takashi Ishio, and Kenichi Matsumoto. 2021. What makes a good Node.js package? Investigating Users, Contributors, and Runnability. *arXiv preprint arXiv:2106.12239* (2021).
- [5] Javad Ghofrani, Paria Heravi, Kambiz A Babaei, and Mohammad D Soorati. 2022. Trust challenges in reusing open source software: An interview-based initial study. In *Proceedings of the 26th ACM International Systems and Software Product Line Conference-Volume B*. 110–116.
- [6] Hao He, Yulin Xu, Yixiao Ma, Yifei Xu, Guangtai Liang, and Minghui Zhou. 2021. A multi-metric ranking approach for library migration recommendations. In *2021 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 72–83.
- [7] Raula Gaikovina Kula, Daniel M German, Ali Ouni, Takashi Ishio, and Katsuro Inoue. 2018. Do developers update their library dependencies? An empirical study on the impact of security advisories on library migration. *Empirical Software Engineering* 23 (2018), 384–417.
- [8] Raula Gaikovina Kula and Christoph Treude. 2022. In war and peace: the impact of world politics on software ecosystems. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, 1600–1604.
- [9] Fabio Massacci, Antonino Sabetta, Jelena Mirkovic, Toby Murray, Hamed Okhravi, Mohammad Mannan, Anderson Rocha, Eric Bodden, and Daniel E Geer. 2022. “Freedom to Protest?” *IEEE Security & Privacy* 20, 5 (2022), 16–21.
- [10] Nusrat Zahan, Thomas Zimmermann, Patrice Godefroid, Brendan Murphy, Chandra Maddila, and Laurie Williams. 2022. What are weak links in the npm supply chain?. In *Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice*. 331–340.

<sup>1</sup><https://twitter.com/andrewmcodes/status/1479876979151106048>