Consumer Perception of Open Source Software Security and Privacy

Christopher Ferrer

ENC2135: Peyton Wahl

February 2, 2024

Open source software is a type of software that has publicly available code and allows users of that software to freely download, use, modify, and redistribute their personal copy of the software, assuming it is published under the same copyright license. All open source software that has been created since the early 1990s has been published with a GNU General Public License copyright license: a copyright license created by programmer Richard Stallman that has led to the modern definition of open source software, allowing users to freely use, modify, and share the software. One of the most notable examples of open source software is the operating system Linux. Linux is a direct competitor to Windows and MacOS, allowing users to customize their personal experience however the user desires and avoid 'software bloat' — a term in computer science that refers to unnecessary and/or unremovable programs or code that take up space on a device and encumber its performance. For example, iPhones used to be unable to uninstall the iTunes, Apple Store, Podcasts, and Stocks apps that come with every version of iOS. However, with the release of iOS 10, Apple started to allow users to delete these apps that many claimed bloated their iPhones (Apple put out a statement with the release of iOS 10 claiming that the built-in, preloaded apps are "very space efficient", but many users still consider them a nuisance). [1] [2] While customization in open source software is one of the most prevalent reasons for its use, especially among individuals with technological backgrounds, it still has major benefits for those who may know very little about tech, such as transparent business practices and the perception of increased security. While consumers may perceive open source software as more secure, for various valid reasons, peer reviewed research has seemingly come up with no significant difference between the security practices of open source software

[1] Chris Welch, "IOS 10 Will Let You Uninstall the Apple Apps You Never Use," The Verge, last modified June 13, 2016, https://www.theverge.com/2016/6/13/11923112/apple-ios-10-delete-stock-apps.
[2] Apple, "Remove Built-in Apple Apps from the Home Screen on Your IOS 10 Device or Apple Watch," Apple Support, n.d. https://support.apple.com/en-us/100567.

compared to proprietary software (also referred to as closed source software); nonetheless, this beneficial perception of open source software does have merit to it, including protecting the user of the software from the software and its developers.

The first main point of contention for open source software users is security. As noted before, peer reviewed research has typically concluded that there is no significant difference in the security between open source and closed source software, although there is a major difference in design philosophy when it comes to security. It is incredibly important to note that recent research on this subject is lacking; most research on the average security of open and closed source software is at least a decade old, which is incredibly outdated for modern technological standards. This lack of research is due, in part, to modern day threat analysis being focused on specific programs/algorithms, instead of entire categories of programs.[3] Despite this lack of modern research, the design principles themselves have not changed. Open source software relies on the vast population it can reach to check its code and ensure that it is secure and up to standard. The underlying concept behind this is that errors are more likely to get fixed (if they exist) if there are more checks in place for that error to be found. One can relate this to the typical separation of powers in the American governmental system. On the other hand, closed source software relies on a design philosophy widely referred to as “security through obscurity”. This design philosophy is focused on the idea that it is hard to find flaws or vulnerabilities in code if a would-be attacker cannot see the code in the first place. In practice, both design philosophies are widely used, meaning that a consumer may be led to believe that they both are equally secure. However, they both also have critical flaws. Open source software could easily be incredibly secure, however it relies on the contentious idea that humans are naturally good

[3] Russell Clarke, David Dorwin, and Rob Nash, "Is Open Source Software More Secure?," Courses.cs.washington.edu, n.d. https://courses.cs.washington.edu/courses/csep590/05au/whitepaper_turnin/oss(10).pdf.

and will report any problems they find instead of abusing them for their own personal gain. Obviously, not all individuals have this good-mannered inclination and may abuse any issues they find instead of promoting the common good. This is the first main argument against open source software, and a valid argument at that. Closed source software, alternatively, takes a near polar opposite contentious idea and assumes that anyone who does want to see the source code will either expose bad practices or abuse vulnerabilities they discover. This can typically be effective, but the main issue with closed source software's design philosophy is the core belief that it is hard, if not impossible, to find vulnerabilities without seeing the source code. Any experienced programmer will be able to explain that while it may be hard, it is most definitely not impossible to find and exploit vulnerabilities without knowing intricate details of any system.

A highly simplified, yet apt comparison to closed source software would be a magician performing a card trick to an audience. If the audience is completely unaware of how magicians typically perform card tricks, or even just unaware of how this particular card trick is typically performed, they are likely to be fooled. However, if the audience happens to contain another magician who is experienced with card tricks, they may be able to pinpoint where the misdirection or sleight of hand occurred during the card trick. The same core principles apply to experienced programmers with closed source software. Thus, while not being able to see the 'magic', the programmer can still assume certain major processes that are taking place and semi-accurately pinpoint where in the process a vulnerability may occur. The accuracy of the experienced programmer's guess can improve if they know basic properties of the software, such as the programming language involved, any outside resources the program is dependent on, etc.

Now that the design philosophies have been identified, the identification and repair of vulnerabilities in software is the next major concern. While there is a lot of complicated mathematics, technology, and research behind identifying vulnerabilities and weak points in open source software, fixing those vulnerabilities typically requires understanding why they are occurring, what is causing them, and understanding the ramifications of any potential fixes. A common rule of thumb in programming is called “Kernighan's Law” — “Everyone knows that debugging is twice as hard as writing a program in the first place. So if you're as clever as you can be when you write it, how will you ever debug it?”[4] This comes into play heavily in closed-source software, as programmers are often organized into small teams. If the team cannot catch a vulnerability before it is abused, it will likely take a long time for them to fix the vulnerability, or they may have to recruit outside help if necessary. In open source software, this issue is typically avoided by users of the software being able to suggest edits to the official releases of the software; someone more qualified to debug the code written as cleverly as possible by the original developers may see the project and want to help out. This may also lead to a possible downside however, as those *unqualified* to fix a certain error may suggest an edit that is then erroneously accepted. Even so, it is likely that the design philosophy of open source comes back into play in this circumstance, and other users correct the suggested edit, claiming that it is worse or less secure in some way.

While security from attackers or potential hackers is important, an equally important form of security to take into consideration is the security and privacy of the user and the user’s data from the developers of any given program. Google, being one of the largest search engines and browsers while simultaneously being one of the largest advertising companies in the world, is no stranger to ethically questionable and potentially privacy-violating practices. These practices have recently landed them in legal peril over Google Chrome’s “incognito mode” tabs

[4] Brian W. Kernighan and P. J. Plauger, The Elements of Programming Style (New York: McGraw-Hill Companies, 1974), page 10.

— tabs that allowed users to avoid having their search history, cookies, and browsing data saved to their device, while Google quietly collected this data regardless. After losing a $5 billion civil lawsuit, Google subsequently updated their incognito mode description, appending the statement “This won’t change how data is collected by websites you visit and the services they use, including Google.”[5] Google’s incognito tab data collection has not changed, still taking user’s browsing data using that information for targeted advertising and data brokering, leading to continued ethical debate about the anonymity expected while using any form of private browsing. Had Google Chrome been open source, it is possible that this long standing issue would have been discovered much earlier, leading to quicker legal action, backlash from the user base, or more clarification from Google earlier on. Unfortunately, software that practice ‘security through obscurity’ often have ulterior motives, such as data collection from users, or hidden power consumption from cryptocurrency miners (a newer trend in viruses known as cryptojackers, typically installed with software from questionable sources). While open source software is not immune to being infused with shady or questionable data collection practices, the ability for users to go into the program’s code and figure out exactly what is being collected is a much more reliable method than a privacy policy or terms of service document that may be misrepresenting the developers’ true goals.

A near perfect example of this scenario is TikTok. Recent legal issues surrounding TikTok and its parent company ByteDance have led the US to pass a sweeping bill granting aid to foreign allies while simultaneously giving ByteDance an ultimatum: divest itself from TikTok by selling it to a company based in the US, or the social media will be completely banned from any devices on US soil (the exact wording of the bill states “It shall be unlawful for an entity to

[5] Venkat Eswarlu, "Google Updates Chrome Incognito Disclaimer Amid $5 Billion Lawsuit Settlement," MSPoweruser, last modified January 16, 2024, https://mspoweruser.com/google-updates-chrome-incognito-disclaimer-amid-5-billion-lawsuit-settlement/.

distribute, maintain, or update (or enable the distribution, maintenance, or updating of) a foreign adversary controlled application by … Providing services to distribute, maintain, or update such foreign adversary controlled application … [or] Providing internet hosting services to enable the distribution, maintenance, or updating of such foreign adversary controlled application for users within the land or maritime borders of the United States.”)[6] While this may seem disconnected, the main reason that many US congresspeople supported this bill was due to TikTok’s proprietary, obscured personalization algorithm that is used for each user’s automatically generated recommendations in tandem with TikTok’s very opaque data collection specifications. If TikTok had chosen to open source its recommendation algorithm and its data collection methods, the bill would have likely faced some blowback from both inside Congress, and an immediate reaction from ByteDance, as opposed to ByteDance currently preparing to bring this fight through the judiciary system.

While transparency is incredibly useful, it is undeniably flawed, as only those who understand the code can utilize the benefits associated with the transparency afforded by open source software. In practice, it is likely that even if TikTok had an open source algorithm for its recommendation system or data collection methods, the congresspeople who brought the initial divest or ban bill would likely have still drafted and supported it. This often leads to those who have technological backgrounds or are more technologically inclined to using open source software. Consequently, as the user base of a specific open source project becomes more technologically inclined, the overall usefulness of the program for the average user may decrease. A major concern of non-Linux users wanting to transfer into Linux, for example, is the lack of accessibility. DistroWatch, a popular site known for monitoring open source software

[6] H.R.7521 - 118th Congress (2023-2024): Protecting Americans from Foreign Adversary Controlled Applications Act. (2024, March 14). https://www.congress.gov/bill/118th-congress/house-bill/7521/text#HC32FAC4BD38647D7BBFE9B480E77B096

releases/news, lists a total of 908 different distributions (a distribution is a different version of a more generalized software, owned by a different entity than the original creators; you can think of this as flavors of ice cream, except each ice cream flavor has a completely separate factory and brand name associated with it) that have been released, with 250 of those being actively maintained via semi-regular updates[7]. Each distribution has a different visual layout, hardware requirements, and, most importantly, must be installed by the user. Unlike Windows or MacOS, Linux has no ‘official’ version, and therefore does not come pre-installed on any computers or laptops a potential user can buy straight from a store.

This has become a major criticism of Linux as a whole, as often the installation and use of these distributions can be quite technical, and are not easy to perform without previous knowledge or a step by step guide (additionally, each individual distribution of Linux may be installed slightly differently than others, depending on requirements set by the developer). In the modern day, the natural instinct to get help for this issue would be asking random internet strangers who are more qualified than you to help you. However, even that has a downside, as TechEyes’s Nick Farrell so infamously said: “The biggest killer of putting penguin software on the desktop was the Linux community. If you think the Apple fanboys are completely barking, they are role models of sanity to the loudmouthed Open Sauce religious loonies who are out there. Like many fundamentalists they are totally inflexible — waving a GNU as if it were handed down by God to Richard Stallman.”[8] To summarize: if you ask for help about distribution A, everyone will ask why you don’t just use distribution B instead, and vice versa.

---

[7] DistroWatch, "DistroWatch: Search distributions," DistroWatch.com, accessed February 21, 2024, https://distrowatch.com/search.php.

[8] Nick Farrell, "Linux's Chance Has Gone," TechEye, last modified October 18, 2010, https://web.archive.org/web/20131222072445/news.techeye.net/software/linuxs-chance-has-gone.

Despite that abysmal review of Linux as a whole, open source software, for the most part, has positive connotations with it. Forcing a business to be transparent in its business practices is almost unanimously considered a social positive, despite the considerable lack of regulation surrounding this exact topic. Open source software has even given us modern marvels — the encryption that is used to ensure you have a safe and secure connection to a website runs on something called an SSL certificate: a stamp of approval from a third party that checks if each website is implementing safe, industry standard, open source encryption. The encryption that is used to ensure end to end encryption in WhatsApp uses SHA-256, an open source encryption method that was developed by the US government. Open source software is an incredibly valuable asset to society and can fill niches that cannot be satisfied by proprietary software. Even with negative reviews of certain open source software, and possible downsides of general open source software, it is ultimately a net positive in a variety of situations, allowing for increased security for the users.

Bibliography

Apple. "Remove Built-in Apple Apps from the Home Screen on Your IOS 10 Device or Apple Watch." Apple Support. n.d. https://support.apple.com/en-us/100567.

Clarke, Russell, David Dorwin, and Rob Nash. "Is Open Source Software More Secure?" Courses.cs.washington.edu. n.d. https://courses.cs.washington.edu/courses/csep590/05au/whitepaper_turnin/oss(10).pdf.

DistroWatch. "DistroWatch: Search distributions." DistroWatch.com. Accessed February 21, 2024. https://distrowatch.com/search.php.

Eswarlu, Venkat. "Google Updates Chrome Incognito Disclaimer Amid $5 Billion Lawsuit Settlement." MSPoweruser. Last modified January 16, 2024. https://mspoweruser.com/google-updates-chrome-incognito-disclaimer-amid-5-billion-lawsuit-settlement/.

Farrell, Nick. "Linux's Chance Has Gone." TechEye. Last modified October 18, 2010. https://web.archive.org/web/20131222072445/news.techeye.net/software/linuxs-chance-has-gone.

H.R.7521 - 118th Congress (2023-2024): Protecting Americans from Foreign Adversary Controlled Applications Act. (2024, March 14). https://www.congress.gov/bill/118th-congress/house-bill/7521/text#HC32FAC4BD38647D7BBFE9B480E77B096

IBM. "What is Open Source Software?" IBM.com. n.d. https://www.ibm.com/topics/open-source.

Imtiaz, Nasif, Aniqa Khanom, and Laurie Williams. “Open or Sneaky? Fast or Slow? Light or Heavy?: Investigating Security Releases of Open Source Packages.” IEEE Transactions on Software Engineering 49, no. 4 (2023): 1540–60. https://doi.org/10.1109/TSE.2022.3181010.

Kernighan, Brian W., and P. J. Plauger. The Elements of Programming Style. New York: McGraw-Hill Companies, 1974. ICh.

Li, Yuancheng, Longqiang Ma, Liang Shen, Junfeng Lv, and Pan Zhang. "Open Source Software Security Vulnerability Detection Based on Dynamic Behavior Features." PloS One 14, no. 8 (2019): e0221530–e0221530. https://doi.org/10.1371/journal.pone.0221530.

Suman, Rajiv Ranjan, Bhaskar Mondal, and Tarni Mandal. "A Secure Encryption Scheme Using a Composite Logistic Sine Map (CLSM) and SHA-256." Multimedia Tools and Applications 81, no. 19 (2022): 27089–110. https://doi.org/10.1007/s11042-021-11460-4.

Welch, Chris. "IOS 10 Will Let You Uninstall the Apple Apps You Never Use." The Verge. Last modified June 13, 2016. https://www.theverge.com/2016/6/13/11923112/apple-ios-10-delete-stock-apps.