Global Ethical Activity

James Burke, Michael Dandrea, Cade Gore and Damari Mazyck

Department of Computer Science, Coastal Carolina University

CSCI 490: Software Engineering II

Dr. Foultz

October 7, 2024

Navigating Complexities of Log Files and Data Privacy

In today's world, software applications profoundly influence society, transcending borders and affecting individuals on a global scale. As developers, we recognize that with the power to build such systems comes significant ethical responsibility. One of the main ethical challenges developers faces relates to the handling of log files, particularly when these files contain sensitive user information. Programmers often collect extensive data, sometimes unwittingly transforming users into products for corporate gain. This essay will explore the ethical implications surrounding log files, user data, and the associated dilemmas of monitoring, protection, and code design. We will discuss the ethical complexities of these issues, offering strategies to address them and examining how our own application would be affected. Finally, we will propose an ethical framework to navigate these dilemmas while maintaining user privacy, ensuring transparency, and balancing the need for functionality with respect for human rights.

The Ethical Dilemma of Log Files

Programmers are often compared to pack rats because of their tendency to store vast amounts of data in log files. These files are essential for debugging, performance monitoring, and ensuring application security. However, their very nature presents a significant ethical challenge. Logs may contain sensitive user data, such as IP addresses, session cookies, user preferences, or even interactions with the system. When mishandled, these logs can inadvertently expose personal information to unauthorized parties or become the target of cyberattacks. Furthermore, the practice of collecting extensive log data without user consent or awareness raises significant privacy concerns.

In the context of our application, log files will play a crucial role in tracking system performance and debugging. However, storing user data in these logs poses an ethical challenge. If mishandled, logs could inadvertently expose users to privacy violations. This is especially relevant when considering that in many cases, users are unaware of what information is being collected or how it is being stored (Floridi, 2013). To address this, we plan to adopt the principle of data minimization, collecting only the necessary information to maintain and optimize the application. We will avoid logging personal identifiers such as usernames or IP addresses unless absolutely necessary. Furthermore, we will implement anonymization techniques where possible and ensure that logs are securely encrypted and stored for a limited time before being deleted. By doing so, we hope to balance the need for system functionality with the ethical obligation to protect user privacy.

Transforming Users into Products: The Commercialization of Data

One of the most significant ethical concerns in modern software development is the transformation of users into products. Many companies collect user data not only to improve their services but also to monetize it by selling the information to third parties, often without the user's explicit consent. This business model has raised serious ethical questions regarding privacy, consent, and the commodification of human behavior.

In building our application, we will face the dilemma of whether or not to collect user data for commercial purposes. While the collection of non-identifiable user data can help improve application performance and user experience, selling this data would erode user trust and violate the principle of informed consent. Our approach would be to maintain full transparency with users, offering them the option to opt-in or opt-out

of data collection. By providing clear and accessible privacy policies and ensuring that user data is only used to improve the service, we will strive to build an ethical relationship with users based on trust.

Moreover, user data collected will never be sold to third parties or used for purposes unrelated to the core functionality of the application. This approach respects users' autonomy and privacy while still allowing for improvements based on aggregated, anonymized data (Davis, 2012).

How Bulletproof Should Code Really Be? The Ethics of Code Quality and Security

When developing software, the question of how "bulletproof" the code should be is another ethical consideration. On the one hand, writing robust and secure code is essential for preventing misuse or cyberattacks. On the other hand, spending excessive time bulletproofing code could lead to resource inefficiencies and diminish focus on other critical areas of development, such as user experience or accessibility.

For our application, the security of user data is a top priority. Given the potential for global usage, the application must be resilient to cyberattacks, particularly since log files may contain sensitive information. Ethically, it is our responsibility to protect users from security breaches that could expose their personal data. However, a balance must be struck between writing "perfect" code and meeting deadlines or delivering features. Ethically, we will prioritize essential security measures, such as regular code reviews, security audits, and adherence to secure coding practices. Moreover, we will use tools such as automated vulnerability scanners to identify and fix potential security gaps early in the development process (Verma & Henson, 2020). While no code can ever be

completely bulletproof, these steps will minimize the likelihood of exploitation and demonstrate a commitment to user safety.

References

Floridi, L. (2013). *The ethics of information.* Oxford University Press.

Davis, K. (2012). *Ethics of Big Data: Balancing risk and innovation*. O'Reilly Media.

Verma, R., & Henson, V. (2020). *Hands-On Security in DevOps*. O'Reilly Media.