UDC 004.41:004.056:3-053.2

O.O. Lukashuk
V. V. Chopliak

SECURE SOFTWARE DEVELOPMENT: PROTECTING USER DATA AND RESTRICTING HARMFUL CONTENT FOR CHILDREN

Vinnytsia National Technical University

Анотація

У статті розглядається проблематика безпечної розробки програмного забезпечення у контексті захисту персональних даних користувачів та обмеження шкідливого контенту для дітей. Обговорюються основні принципи безпечного програмування, механізми шифрування даних, автентифікації та системи батьківського контролю. Проаналізовано сучасні підходи до проектування програмного забезпечення з урахуванням вимог інформаційної безпеки та етичних стандартів на сучасному цифровому ринку.

Ключові слова: безпечна розробка ПЗ, захист даних, батьківський контроль, шифрування, інформаційна безпека.

Abstract

The growing dependence on digital services has made secure software development a critical discipline within software engineering. This paper examines the fundamental importance of building secure applications that protect user personal data and implement effective content restriction mechanisms for children. Key principles of the secure software development lifecycle (SSDLC) are analyzed, including threat modeling, input validation, encryption, and access control. The study also addresses child-safety features such as parental controls, age verification systems, and AI-driven content filtering. The findings demonstrate that embedding security and ethical design from the earliest stages of development significantly reduces vulnerabilities and fosters user trust.

Keywords: secure software development, data protection, parental controls, encryption, information security.

Introduction

Modern software systems handle vast amounts of sensitive personal data, ranging from medical records and financial transactions to private communications and browsing habits. As digital platforms become increasingly embedded in daily life, the consequences of inadequate software security grow ever more severe. Data breaches, identity theft, and unauthorized access to private information have become persistent threats that affect millions of users worldwide.

A parallel and equally important concern is the protection of young users. Children and adolescents are among the most active consumers of digital content, yet they are particularly vulnerable to exposure to inappropriate material, online predators, and addictive design patterns. Software engineers bear a direct responsibility to incorporate protective mechanisms that limit access to harmful content and safeguard young users' digital wellbeing. This paper explores the engineering practices and architectural decisions that underpin both user data protection and child content safety in modern software systems.

Research results

Secure software development begins at the design phase with threat modeling. Methodologies such as STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) allow development teams to systematically identify and mitigate risks before a single line of code is written. When threat modeling is integrated into the software development lifecycle alongside rigorous code reviews and automated static analysis tools, the overall attack surface of an application is dramatically reduced.

Data protection is another foundational pillar. End-to-end encryption, secure storage of credentials using modern hashing algorithms such as bcrypt or Argon2 and adherence to the principle of least privilege ensure that user data remains confidential even in the event of a partial breach. Compliance with international

regulations such as GDPR and COPPA further enforces engineering teams to adopt data minimization strategies, whereby only the data strictly necessary for service functionality is collected and retained.

Regarding child safety, contemporary software engineering increasingly relies on a layered approach. Age verification systems, though imperfect, serve as an initial gate to restrict access to adult content. Parental control dashboards allow guardians to configure usage time limits, block specific application categories, and monitor activity logs. More sophisticated platforms employ machine learning classifiers to automatically detect and filter harmful textual and visual content in real time, reducing the burden on manual moderation teams.

A critical engineering challenge lies in balancing strong security with seamless user experience. Overly restrictive systems may frustrate legitimate users and undermine adoption, while permissive systems leave data and young users exposed. Techniques such as progressive disclosure of permissions, transparent privacy policies, and user-friendly two-factor authentication represent practical solutions that achieve this balance without compromising protection.

Conclusion

Secure software development is not merely a technical requirement but an ethical imperative. As digital platforms process increasingly sensitive personal information and serve younger audiences, the responsibility of software engineers extends beyond functionality to encompass the safety and wellbeing of all users. Integrating security practices from the earliest stages of the development lifecycle, combined with dedicated child protection features, results in more resilient, trustworthy, and socially responsible software systems. Future work should continue to refine AI-based content moderation tools and advance standardized frameworks for child-safe software design across the industry.

REFERENCES

1. Umeugo, W. (2023). Secure software development lifecycle: A case for adoption in software SMEs. *International Journal of Advanced Research in Computer Science*, 14(1), 5–12.
2. Ta, V.-T. (2024). A safety risk assessment framework for children's online safety based on a novel safety weakness assessment approach. *arXiv preprint* arXiv:2401.14713. https://doi.org/10.48550/arXiv.2401.14713.
3. Jarvie, C., & Renaud, K. (2024). Online age verification: Government legislation, supplier responsibilization, and public perceptions. *Children*, 11(9), 1068. https://doi.org/10.3390/children11091068.
4. Kravchenko, K., Ketsyk-Zinchenko, U., Suduk, I., Nykyporets, S., & Cherednychenko, V. (2025). Effectiveness of online platforms in developing language skills of higher education students. *Revista Eduweb*, 19(3), 303-314. https://doi.org/10.46502/issn.1856-7576/2025.19.03.19.
5. Nykyporets, S. S., Kot, S. O., Boiko, Y. V., Melnyk, M. B., & Chopliak, V. V. (2024). Advanced integration of virtual information environments (VIEs) in contemporary educational methodologies. *Society and National Interests. Series "Education/Pedagogy"*, 4(4), 139-154. https://doi.org/10.52058/3041-1572-2024-4(4)-139-154.

Лукашук Оксана Олегівна – студентка групи 6ПІ-24б, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: oks.luka3.14@gmail.com.

Науковий керівник: ***Чопляк Вікторія Володимирівна*** – викладач англійської мови, кафедра іноземних мов, Вінницький національний технічний університет, м. Вінниця, e-mail: nikavnuchkova@gmail.com.

Oksana O. Lukashuk – a student of 6SE-24b group, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: oks.luka3.14@gmail.com.

Scientific Supervisor***: Victoriia V. Chopliak*** – teacher of English, Foreign Languages Department, Vinnytsia National Technical University, Vinnytsia, e-mail: nikavnuchkova@gmail.com.