MATERIALIST: A Web Platform for Guiding Privacy Design Pattern Selection in Software Development

Giuseppe Desolda[1], Andrea Esposito[1], Francesco Greco[1], Rosa Lanzilotti[1] and Marco Saltarella[2]

[1]Computer Science Department, University of Bari Aldo Moro, Via E. Orabona 4, 70125 Bari (BA), Italy
[2]FINCONS S.p.A., Via Orfeo Mazzitelli, 258/E, 70124 Bari (BA), Italy

Abstract

As software production evolves, privacy is becoming an increasingly important consideration. This is especially true as national and supranational regulations, such as GDPR, require privacy as a mandatory aspect of software development. However, challenges such as a lack of knowledge about privacy and data protection regulations by developers hinder the adoption of effective privacy implementation mechanisms. To address this issue, this paper presents MATERIALIST, a Web platform designed to assist developers in using proper privacy design patterns during software development. The main focus is to suggest privacy design patterns starting from GDPR requirements, code vulnerabilities, or software lifecycle phases, providing a practical solution that developers can apply in their work. MATERIALIST aims to facilitate the adoption of appropriate privacy implementation mechanisms in the software development lifecycle, thereby improving software quality.

Keywords

Privacy design patterns, GDPR, ISO 9241-210, code vulnerabilities

1. Introduction

Although privacy is a critical aspect of software systems and is required by privacy regulations such as the European Union's General Data Protection Regulation (GDPR), several factors limit its proper implementation. Firstly, designers and developers have different views on privacy, leading to different approaches and solutions [1]. Secondly, privacy regulations only provide legal guidance without specific technical instructions for developers. Thirdly, although privacy regulations require user involvement, there is a lack of user-centric implementations [2]. Finally, developers and engineers often lack privacy and security skills, leading to vulnerable and insecure privacy solutions [3]. Thus, there is a need for more appropriate methodologies to integrate privacy aspects into the software development process without ambiguity and from an end-user perspective.

This paper presents a web platform, called MATERIALIST (Mapping dATa rEgulation softwaRe lIfecycle And vuLnerabilitIeS paTterns) whose aim is to assist developers, designers, and

IS-EUD 2023: 9th International Symposium on End-User Development, 6-8 June 2023, Cagliari, Italy
giuseppe.desolda@uniba.it (G. Desolda); andrea.esposito@uniba.it (A. Esposito); francesco.greco@uniba.it (F. Greco); rosa.lanzilotti@uniba.it (R. Lanzilotti); marco.saltarella@finconsgroup.com (M. Saltarella)
0000-0001-9894-2116 (G. Desolda); 0000-0002-9536-3087 (A. Esposito); 0000-0003-2730-7697 (F. Greco); 0000-0002-2039-8162 (R. Lanzilotti); 0000-0002-0021-9972 (M. Saltarella)

engineers in integrating privacy aspects during software development. This platform can be considered a CASE (Computer-Aided Software Engineering) tool, i.e., software used to facilitate and support Software Engineering processes, such as UML tools, process modeling tools, project management tools, documentation tools, IDEs, and compilers. It guides the selection of privacy design patterns (PDPs) from three starting points: GDPR articles, phases of the ISO 9241-210 software development lifecycle, and vulnerabilities discovered during static code analysis. The first two entry points allow the incorporation of PDPs at the beginning of the development process (forward engineering), while the third entry point supports the re-engineering of software systems (backward engineering). This platform provides PDPs at both architectural and user interface levels to make the adoption of selected PDPs more concrete. In the following, we report on the details of MATERIALIST.

2. The MATERIALIST Platform

This platform implements a framework called MATERIALIST [4], designed by some authors of this paper. The heart of this framework is a collection of 72 privacy design patterns found on privacypatterns.org. To make the adoption of these patterns more practical, we extended them with new architectural and user interface patterns that help developers and designers create privacy-related code and interfaces more clearly and unambiguously. Throughout the paper, we refer to the extended set of privacy design patterns as PDPs for simplicity. The idea is to use these PDPs to guide stakeholders in selecting privacy solutions from three different entry points: GDPR articles, ISO 9241-210 phases, and privacy vulnerabilities found during static code analysis. The use of PDPs addresses two of the issues mentioned in the introduction – differing views on privacy and a lack of privacy and security knowledge.

PDPs provide a standardized language for privacy and offer robust and practical solutions created by privacy experts. The MATERIALIST framework considers the GDPR and ISO 9241-210 because they are the main privacy regulation and human-centered software development process in the EU, respectively. Using these standards mitigates the issues of non-compliance and non-user-centric implementations. We also use the OWASP Top 10 2021, de-facto standard, to identify vulnerabilities. One unique aspect of this framework is the ability to traverse between entry points using privacy patterns, allowing stakeholders to easily identify PDPs that apply to different phases of development or specific vulnerabilities. In forward engineering, PDPs are proposed to comply with GDPR articles, as well as for each ISO 9241-210 phase; in backward engineering, PDPs are proposed for each vulnerability found during the static analysis of existing code.

To guide stakeholders in selecting the right PDPs for GDPR compliance, the authors of [4] systematically mapped the 72 privacy patterns and the GDPR articles. Two researchers, who are experts in both GDPR and PDPs, independently conducted the mapping phase, which took around 40 hours each. To increase robustness and reduce biases, the mapping was performed starting from each PDP towards the GDPR articles and vice versa. The researchers then compared their results and reached full agreement on the remaining mappings, resulting in different relationships between each privacy pattern and 14 different articles of the GDPR. The mapping result is reported as a Web page available at http://90.147.170.155/mapping.html.

The mapping between vulnerabilities and PDPs aligns with the study reported by Baldassarre et al. [5]. We performed the same mapping but considered the updated OWASP Top 10 2021 instead of the OWASP Top 10 2017 used in the existing mapping. The mapping is achieved by first associating the vulnerabilities with the Privacy by Design (PbD) principles, which are then mapped with privacy design strategy, and finally with the PDPs.

Finally, a mapping between the ISO 9241-210 process phases and each PDP is also performed to identify when stakeholders should consider implementing a particular pattern during the development process [6, 7]. This helps ensure that security is considered a process throughout the whole software development life cycle (SDLC). The ISO 9241-210 process is iterative and divided into five phases, and the mapping helps to identify which PDPs should be implemented during each phase. This helps to ease the secure implementation of the system.

All the resulting PDPs and the above-mentioned mappings have been implemented in a web platform that simplifies the selection of appropriate PDPs in the software development lifecycle, thereby improving software quality. The design of the platform followed a Human-Centered Design process. The requirements were collected during an elicitation study performed in the forms of interviews and involving 3 project managers, 1 analyst, and 5 developers. The resulting requirements drove the design of the web platform, which was then evaluated during a user test. This paper focuses on the details of the platform while further details of the HCD process and of all the mappings will be reported in a longer version of this study and presented during the workshop.

The platform comprises two principal sections: the Knowledge Base section and the Project section. The former allows users to explore the various entry points of the framework and their corresponding mappings. The latter allows users to select and track the PDPs to be implemented and provides an overview of each phase of the ISO 9241-210 lifecycle and the GDPR articles to which the implementation of the PDPs corresponds. Additionally, an example project is available to each user to familiarize them with the platform's functionalities.

When a new project is created, the user is prompted to complete a questionnaire to assist them in selecting the PDPs to be implemented based on the requirements of the software being analyzed, as shown in Figure 1. After completing the questionnaire, the user is presented with an overview tab on the left side displaying the different PDPs tracked in the project and their implementation details, including the practical application of the patterns (see Figure 2). An info tooltip explains to the user why a PDP was added to the project, such as a positive response to a question or manual addition. A "GDPR" tab highlights the GDPR articles covered by the implementation of each PDP, while the "ISO9241-210" tab indicates the phase in which each PDP should be considered and implemented. Users can add or remove any PDP to the project by searching the knowledge base for patterns that match the system requirements.

3. Conclusion and Future Work

In this paper, we presented MATERIALIST, a web platform designed to support user-centric, secure, and privacy-aware software engineering processes both forward and backward. This tool facilitates the selection of Privacy Design Patterns (PDPs) using different starting points, such as GDPR articles, code vulnerabilities, or ISO 9241-210 phases.

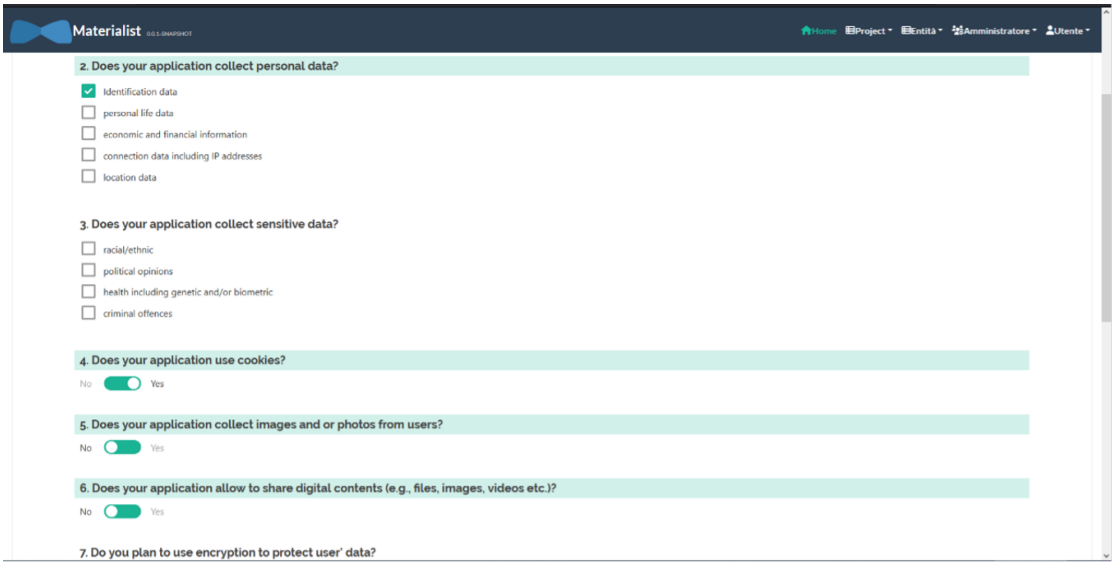

Figure 1: The questionnaire proposed by the platform

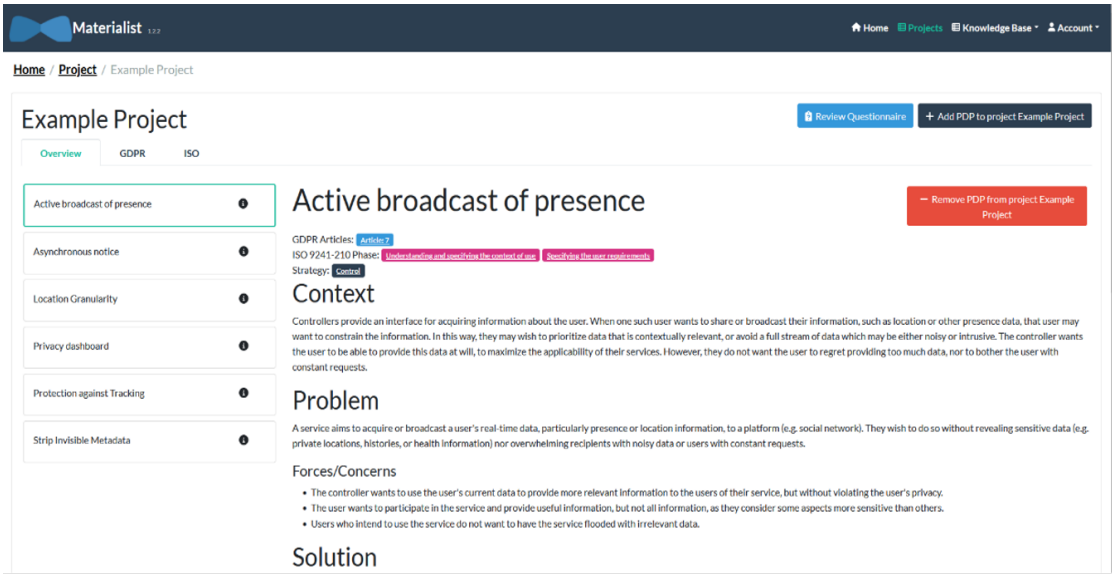

Figure 2: Project overview

To enhance the selection of PDP, we are also working on defining intermediate layers that will help guide PDP selection from any entry point. We are also exploring heuristic and metric-based approaches to recommend appropriate PDPs based on contextual factors. Furthermore, we plan to further expand the PDPs catalog by defining architectural patterns and User Interface patterns for each one to better assist designers and developers during software development activities such as coding and design.

Acknowledgments

This work is partially supported by the co-funding of the European union - Next Generation EU: NRRP Initiative, Mission 4, Component 2, Investment 1.3 – Partnerships extended to universities, research centres, companies and research D.D. MUR n. 341 del 5.03.2022 – Next Generation EU (PE0000014 - "Security and Rights In the CyberSpace - SERICS" - CUP: H93C22000620001).

The research of Andrea Esposito is funded by a Ph.D. fellowship within the framework of the Italian "D.M. n. 352, April 9, 2022"- under the National Recovery and Resilience Plan, Mission 4, Component 2, Investment 3.3 – Ph.D. Project "Human-Centered Artificial Intelligence (HCAI) techniques for supporting end users interacting with AI systems", co-supported by "Eusoft S.r.l." (CUP H91I22000410007).

The research of Francesco Greco is funded by a PhD fellowship within the framework of the Italian "D.M. n. 352, April 9, 2022"- under the National Recovery and Resilience Plan, Mission 4, Component 2, Investment 3.3 - PhD Project "Investigating XAI techniques to help user defend from phishing attacks", co-supported by "Auriga S.p.A." (CUP H91I22000410007).

References

[1] Y.-S. Martin, A. Kung, Methods and tools for GDPR compliance through privacy and data protection engineering, in: 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE, 2018, pp. 108–111. URL: https://doi.org/10.1109/eurospw.2018.00021. doi:10.1109/eurospw.2018.00021.

[2] M. Sobolewski, J. Mazur, M. Paliński, Gdpr: A step towards a user-centric internet?, Intereconomics 52 (2017) 207–213. URL: https://doi.org/10.1007/s10272-017-0676-5. doi:10.1007/s10272-017-0676-5.

[3] K. Hjerppe, J. Ruohonen, V. Leppanen, The general data protection regulation: Requirements, architectures, and constraints, in: 2019 IEEE 27th International Requirements Engineering Conference (RE), IEEE, 2019, pp. 265–275. URL: https://doi.org/10.1109/re.2019.00036. doi:10.1109/re.2019.00036.

[4] V. Barletta, G. Desolda, D. Gigante, R. Lanzilotti, M. Saltarella, From gdpr to privacy design patterns: The materialist framework, in: Proceedings of the 19th International Conference on Security and Cryptography - Volume 1: SECRYPT,, INSTICC, SciTePress, 2022, pp. 642–648. doi:10.5220/0011305900003283.

[5] M. T. Baldassarre, V. S. Barletta, D. Caivano, M. Scalera, Privacy oriented software development, in: M. Piattini, P. Rupino da Cunha, I. García Rodríguez de Guzmán, R. Pérez-Castillo (Eds.), Quality of Information and Communications Technology, Springer International Publishing, Cham, 2019, pp. 18–32.

[6] S. Alpers, A. Oberweis, M. Pieper, S. Betz, A. Fritsch, G. Schiefer, M. Wagner, PRIVACY-AVARE: An approach to manage and distribute privacy settings, in: 2017 3rd IEEE International Conference on Computer and Communications (ICCC), IEEE, 2017, pp. 1460–1468. doi:10.1109/compcomm.2017.8322784.

[7] T. Jakobi, S. Patil, D. Randall, G. Stevens, V. Wulf, It is about what they could do with the

data: A user perspective on privacy in smart metering, ACM Trans. Comput.-Hum. Interact. 26 (2019). URL: https://doi.org/10.1145/3281444. doi:10.1145/3281444.