MSR4P&S 2022 Workshop Summary

Melina Vidoni
Australian National University
Australia
melina.vidoni@anu.edu.au

Nicolás E. Díaz Ferreyra
Hamburg University of Technology
Germany
nicolas.diaz-ferreyra@tuhh.de

Zadia Codabux
University of Saskatchewan
Canada
zadiacodabux@ieee.org

DOI: 10.1145/3573074.3573100
http://doi.org/10.1145/3573074.3573100

ABSTRACT

The 1st edition of the workshop on Mining Software Repositories for Privacy and Security (MSR4P&S 2022) was held virtually on November 18th, 2022, co-located with the 30th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2022), which took place in Singapore. MSR4P&S received a total of five submissions from diverse geographic locations, which were all included in the program after a rigorous peer-review process. The program also featured a keynote by M. Ali Babar on the quality of data mined for security research. This report summarises the event and insights stemming from the keynote and presentations in the workshop's two sessions.

1. INTRODUCTION

The last decades have put Privacy and Security (P&S) in the spotlight of information technology as data breaches, and cyberattacks have spiked globally. Still, P&S are often afterthoughts in software development as their benefits are sometimes difficult to demonstrate and their costs hard to justify [9, 13]. Such an issue is becoming hard to sustain as new legal frameworks such as the EU General Data Protection Regulation (GDPR) demand companies to incorporate P&S features (e.g., transparency, anonymity, and informed consent) at the core of their products [9]. Hence, there is an urgent call for tools and methods supporting the elicitation and deployment of P&S requirements in a *by-design* approach.

P&S are multifaceted, complex research areas spanning different knowledge domains (e.g., engineering, psychology) [2, 6]. Challenges in P&S cannot be solely addressed from a single discipline as they involve human factors, technological artefacts, and regulatory/legal frameworks [4, 16]. Remarkably, the quest for P&S solutions requires in-deep knowledge and actionable information about its users/stakeholders, vulnerabilities/flaws, and potential attackers [5, 6]. Mining Software Repositories (MSR) techniques can support this quest by providing means to understand the P&S dimensions of information systems, thus helping shape privacy- and security-friendly software. This workshop aims to explore the application of MSR at the different stages of P&S engineering [3, 10].

MSR4P&S aims to provide a forum for researchers and practitioners to present and discuss new ideas, trends, and results regarding MSR applications for cybersecurity research, including empirical and mixed-method approaches, as well as datasets and tools. In the remainder of this report, we start by briefly describing the workshop format in Section 2, followed by the highlights of the keynote in Section 3, and the insights from the presentation sessions in Section 4. Finally, we conclude the report in Section 5 with the takeaway messages and future plans.

2. WORKSHOP FORMAT

MSR4P&S received five submissions (two 8-page full papers and three 4-page short papers) from Canada, the United States, the UK, France, and Norway. After a thorough peer-review process, all five papers were accepted and given an additional page to incorporate the changes suggested by the reviewers. At least three members of the Program Committee reviewed each submission. The accepted papers cover various topics and are divided into two main sessions: *assessing privacy through mining software repositories* and *using mining software repositories to detect or analyse vulnerabilities*.

The workshop proceedings are published by ACM [17] as a co-located event of the 30th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2022). The workshop was held on November 18th, 2022 and had 45 participants. Due to the SARS-CoV-2 pandemic, the event took place virtually and was organised with this aspect in mind. The program comprised the opening, one keynote, two presentation sessions, and the closing. We planned two thematic sessions, "Assessing Privacy" and "Vulnerabilities", based on the topics addressed by the accepted papers. Each talk of the thematic sessions had fifteen minutes allocated for the presentation and five minutes for discussions.

3. KEYNOTE TAKEAWAY

M. Ali Babar is the director of the Centre for Research on Engineering Software Technologies (CREST) at the University of Adelaide (Australia). He delivered the keynote entitled "Mining Software Repositories for Security: Data Quality Issues Lessons from Trenches" [1].

Approximately 90% of successful cyber incidents are caused by vulnerabilities rooted in software. MSRs are essential to investigate P&S concerns in multiple domains and combine a myriad of sources. In particular, they can be used to study *malware and malicious code* (through source code, packages, and commits), *vulnerabilities and bugs* (through source code, issues and bug reports, social media posts, and discussions), *software supply chain security* (with package managers and pre-trained toolkits), *licence compliance* (with licence agreements and source code), *privacy*, and *secure coding detection* (through source code and commits). Moreover, MSR-based P&S research can use a single data source (e.g., by extracting the data from a single source, like StackOverflow) or by combining multiple data sources (e.g., combining data from a vulnerability database and GitHub).

Nevertheless, the prevalence of vulnerabilities and disclosures in open-source systems enables MSRs and creates a number of challenges, all with unique solutions or workarounds. Regarding

the *data extraction phase*, there is limited support for automated extraction and keeping the scraper and the extracted data up-to-date. Meanwhile, during *data pre-processing*, the main challenges are handling different version control systems and a pervasive lack of standards in tools for bug reporting and data collection. Finally, during the *data integration* phase, correlating the data points across technical domains is an issue that directly impacts the quality of sources.

Given that reproducibility and the inaccessibility of data are essential to MSR-based P&S research, the talk centred on these issues, which were further divided into ten challenges:

1. **Data Scarcity** may be caused by lack of explicit labelling/understanding of security issues, imperfect data collection, or rare occurrence of certain types of security issues, leading to imbalanced data. Nevertheless, it was highlighted that having more data is preferable over a 'cleverer' (i.e., more advanced) algorithm.
2. **Data (In)Accuracy** is a problem since critical vulnerabilities may go unfixed due to being incorrectly labelled as false negatives. This is due to the "unknown unknowns"–in M. Ali Babar's words, "if you don't know about it, you can't fix it." This "unknown unknown" may happen due to a lack of reporting or silent patches and even tangled changes that fix non-security and security issues. Eventually, these problems can lead to Machine or Deep Learning (ML/DL) classifiers learning the wrong patterns, effectively introducing backdoors.
3. **Data (Ir)Relevance** happens because not all input helps predicting security issues and may lead to overfitting. This reduced usefulness is often the consequence of not having thoroughly explored the issue due to time constraints or the lack of an exploratory analysis. Designing an investigation while considering its construct validity is crucial to avoid this issue.
4. **Data Redundancy** can have multiple causes, for example, attempting to increase the number of security issues by oversampling, crossing from multiple sources, and a poor MSR process. Regarding the latter, when using version control systems as the source, *data redundancy* may occur due to mining cloned projects that carry unfixed vulnerabilities, code that is merged into the master branch, renamed elements, and cosmetic-only changes. Often, *data redundancy* diminishes the capability of the ML models, leading to bias and overfitting, thus inflating the models' performance.
5. **Data (Mis)Coverage** refers to security issues spanning multiple lines, functions, files, and modules generally considered only partially, for example, by focusing on a single function or file. The challenge for ML/DL-based approaches is known as 'coverage vs. size'; namely, more granular approaches acquire more samples at the cost of losing the overall view. Cases like this are trading coverage for convenience by reducing the inspection effort. The most corning impact is that if lacking the context for training the model, the performance obtained (namely, the F1 values) will not be accurate.
6. **Data (Non-)Representativeness** happens because real-world security vary vastly from synthetically-generated issues. Real-world data is generally interdependent and complex, taking into account the different features and nature of the apps. Synthetic data lack generalisability and transferability, producing models that cannot be easily extended.
7. **Data Drift** happens due to the unending 'battle' between attackers and defenders. This leads to "evolving" threat landscapes with characteristics that change over time since there is always a difference between the stage in which the data is collected and when it is analysed. For example, out-of-vocabulary words may degrade the performance, leading to data leakage and cases of unrealistic performance with severe overfitting.
8. **Data (In)Accessibility** happens because security data is not always shared, and reconstructed data may be different from the original. More importantly, this challenge also affects the reviewing process of academic works. *Data Inaccessibility* may be rooted in non-disclosure agreements with industry partners, datasets that are too large for storage, and data values that change over time. The impact of this challenge is the limited reproducibility of existing results and the limited generalisability of ML models.
9. **Data (Re-)Use** sometimes is better than updating or collecting new data because pre-existing datasets are often more trustworthy and reliable than new ones. However, those existing datasets can be severely outdated and suffer other challenges, such as redundancy, inaccuracy, and irrelevancy. ML/DL-based approaches trained with existing data may become obsolete and less generalisable.
10. **Data Maliciousness** refers more to ethical research, as data regarding threat and security is itself a threat (namely, a new vulnerability). Moreover, using and sharing this data without precautions can create backdoors due to simple oversights. Open science should also be responsible science–and informing the indirect participants and the reviewers is vital.

Based on his experience, M. Ali Babar considered data (ir)relevance and (in)accuracy as the most frequent challenges in MSR research applied to P&S. He also gave several recommendations to deal with these challenges. In particular, to 1) consider the labels' noise in the negative classes (e.g., confusing words, or raters' mistakes), 2) consider the timeliness (i.e., to preserve data sequence for training), 3) use data visualisation to improve understandability for non-scientists, 4) create and use diverse language datasets, 5) use data-quality assessments with clear criteria, specific to each set, and 6) to improve better data sharing and governance, by providing the exact details and processes of data preparation.

4. INSIGHTS FROM SECTIONS

This section elaborates on the outcome of the two presentation sessions: "Assessing Privacy" and "Vulnerabilities".

4.1 Assessing Privacy

The first session addressed MSR-driven studies aiming to gather real-world evidence to investigate privacy and security issues. It included the following two presentations:

Shimmi and Rahimi [12] proposed mining existing software repositories to document patterns of co-evolution to determine the probability of attackers' responsive actions. They addressed two main challenges: 1) creating a diverse, large dataset to train a deep learning model to identify semantic properties, and 2)

extending the dataset to include a wide range of various-type artefacts in heterogeneous formats. Their preliminary results indicate that the patterns can be defined within three categories: (i) consistency, (ii) adaptation, and (iii) optimisation co-evolution, each with semantic and syntactical properties.

Tang and Østvold [15] developed an automated software analysis technique to assist developers and non-technical people to document software privacy and data protection behaviour. This was done using the GDPR as the main guideline. Their method was tested in apps accepting raw, sensitive user data while also entailing data transmission. They considered Signal and NexCloud, two messaging apps. These cases demonstrated that their technique effectively detects privacy source and sink methods in software bytecode, generating privacy flow graphs.

4.2 Vulnerabilities

The second session was dedicated to research efforts about software vulnerabilities. It included three presentations.

[7] presented a preliminary study exploring the relationship between smells, design issues, and software vulnerabilities. They studied nine open-source projects from the Apache foundation (including Kafka, Camel, Solr, and Tomcat), leveraging their fix-commits, the vulnerability reports, and results of a code inspection performed with GetSmell, a code smell detection tool. Their results show that although some smells and design issues are significantly related to vulnerabilities, a manual analysis did not directly indicate that smells or design issues induce vulnerabilities. They also uncovered that smells and design issues are still present in the classes, even after fixing the vulnerabilities.

[11] presented a preliminary investigation regarding counterfeit object-oriented programming, where attackers hijack objects in the program to create a sequence of method calls that introduce malicious behaviour. They mined vulnerability reports from the National Vulnerability Database, currently tracking over 191000 vulnerabilities, filtering them by open/closed source and CWE (Common Vulnerabilities and Exposure) tags related to specific concerns. They randomly sampled the resulting set and manually analysed the cases. Overall their results show that this type of attack can lead to severe vulnerabilities, that developers may use inherently flawed/improper mitigation techniques, and that there are trade-offs regarding choosing a specific mitigation strategy.

[14] presented a de-identified dataset (named, SecurityEval) to assess code generation that results in vulnerable code. Currently, it contains 130 Python samples of 75 vulnerability types mapped to the CWE. This dataset was generated by mining vulnerability examples from CodeQL, CWE, Sonar Rules, and prior works. Moreover, the authors demonstrated that SecurityEval could be used in automated and manual analyses and conducted example assessments using InCoder and GitHub Copilot. Among their future works, the authors consider adding more samples, covering other CWEs, and extending the dataset to other languages.

5. CONCLUSIONS

By retrospectively analysing the MSR4P&S workshop and events in related topics, we notice that MSRs are becoming an essential methodology for evidence-based studies, while P&S are spreading into multiple areas providing highly diverse, inter-disciplinary works. This observation is supported by the continuously growing number of studies published in journals and conferences, the inclusion of MSR in the ACM Empirical Standards [8], and the diversity of studies assessing P&S. Moreover, multiple specific conferences, such as the International Conference on Mining Software Repositories and the IEEE Symposium on Security and Privacy, tackle these issues individually. MSR4P&S indicates the need to coordinate the efforts for providing a lasting forum at the intersection of these areas to exchange ideas and present insightful results.

As we move forward, it is essential that this community continues to be supported while providing a venue to consolidate the related body of knowledge. For that reason, we also consider organising follow-up editions of MSR4P&S for several years and eventually join efforts with the organisers of related workshops to transform MSR4P&S into a working conference with a well-defined format and organisation. Such action would reinforce the existing vibrant community of researchers and provide a connection with related venues.

References

[1] Muhammad Ali Babar. 2022. Mining Software Repositories for Security: Data Quality Issues Lessons from Trenches (Keynote). In *Proceedings of the 1st International Workshop on Mining Software Repositories Applications for Privacy and Security* (Singapore, Singapore) *(MSR4P&S 2022)*. Association for Computing Machinery, New York, NY, USA, 1. https://doi.org/10.1145/3549035.3570192

[2] Kathrin Bednar, Sarah Spiekermann, and Marc Langheinrich. 2019. Engineering Privacy by Design: Are engineers ready to live up to the challenge? *The Information Society* 35, 3 (2019), 122–142.

[3] Seyed Mohammad Ghaffarian and Hamid Reza Shahriari. 2017. Software vulnerability analysis and discovery using machine-learning and data-mining techniques: A survey. *ACM Computing Surveys (CSUR)* 50, 4 (2017), 1–36.

[4] Seda Gürses and Jose M Del Alamo. 2016. Privacy engineering: Shaping an emerging field of research and practice. *IEEE Security & Privacy* 14, 2 (2016), 40–46.

[5] Phu X Mai, Arda Goknil, Lwin Khin Shar, Fabrizio Pastore, Lionel C Briand, and Shaban Shaame. 2018. Modeling security and privacy requirements: a use case-driven approach. *Information and Software Technology* 100 (2018), 165–182.

[6] Yod-Samuel Martin and Antonio Kung. 2018. Methods and tools for GDPR compliance through privacy and data protection engineering. In *IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, London, UK, 108–111.

[7] Sahrima Jannat Oishwee, Zadia Codabux, and Natalia Stakhanova. 2022. An Exploratory Study on the Relationship of Smells and Design Issues with Software Vulnerabilities. In *Proceedings of the 1st International Workshop on Mining Software Repositories Applications for Privacy and Security* (Singapore, Singapore) *(MSR4P&S 2022)*. Association for Computing Machinery, New York, NY, USA, 16–20. https://doi.org/10.1145/3549035.3561182

[8] Paul Ralph. 2021. ACM SIGSOFT Empirical Standards Released. *SIGSOFT Softw. Eng. Notes* 46, 1 (jan 2021), 19. https://doi.org/10.1145/3437479.3437483

[9] Kalle Rindell, Karin Bernsmed, and Martin Gilje Jaatun. 2019. Managing security in software: Or: How i learned to stop worrying and manage the security technical debt. In *14th International Conference on Availability, Reliability and Security*. ACM, UK, 1–8.

[10] Alireza Sadeghi, Naeem Esfahani, and Sam Malek. 2014. Mining the categorized software repositories to improve the analysis of security vulnerabilities. In *International conference on fundamental approaches to software engineering*. Springer, Berlin, Germany, 155–169.

[11] Joanna C. S. Santos, Xueling Zhang, and Mehdi Mirakhorli. 2022. Counterfeit Object-Oriented Programming Vulnerabilities: An Empirical Study in Java. In *Proceedings of the 1st International Workshop on Mining Software Repositories Applications for Privacy and Security* (Singapore, Singapore) *(MSR4P&S 2022)*. Association for Computing Machinery, New York, NY, USA, 21–28. https://doi.org/10.1145/3549035.3561183

[12] Samiha Shimmi and Mona Rahimi. 2022. Mining Software Repositories for Patternizing Attack-and-Defense Co-Evolution. In *Proceedings of the 1st International Workshop on Mining Software Repositories Applications for Privacy and Security* (Singapore, Singapore) *(MSR4P&S 2022)*. Association for Computing Machinery, New York, NY, USA, 2–6. https://doi.org/10.1145/3549035.3561181

[13] Miltiadis Siavvas, Dimitrios Tsoukalas, Marija Jankovic, Dionysios Kehagias, Alexander Chatzigeorgiou, Dimitrios Tzovaras, Nenad Anicic, and Erol Gelenbe. 2019. An empirical evaluation of the relationship between technical debt and software security. In *9th International Conference on Information society and technology (ICIST)*, Vol. 2019. Commission of the European Communities, Europe, 1–10.

[14] Mohammed Latif Siddiq and Joanna C. S. Santos. 2022. SecurityEval Dataset: Mining Vulnerability Examples to Evaluate Machine Learning-Based Code Generation Techniques. In *Proceedings of the 1st International Workshop on Mining Software Repositories Applications for Privacy and Security* (Singapore, Singapore) *(MSR4P&S 2022)*. Association for Computing Machinery, New York, NY, USA, 29–33. https://doi.org/10.1145/3549035.3561184

[15] Feiyang Tang and Bjarte M. Østvold. 2022. Assessing Software Privacy Using the Privacy Flow-Graph. In *Proceedings of the 1st International Workshop on Mining Software Repositories Applications for Privacy and Security* (Singapore, Singapore) *(MSR4P&S 2022)*. Association for Computing Machinery, New York, NY, USA, 7–15. https://doi.org/10.1145/3549035.3561185

[16] Sven Türpe. 2017. The trouble with security requirements. In *25th International Requirements Engineering Conference (RE)*. IEEE, Lisbon, Portugal, 122–133.

[17] Melina Vidoni, Nicolás Díaz-Ferreyra, and Zadia Codabux (Eds.). 2022. *MSR4P&2022: Proceedings of the 1st International Workshop on Mining Software Repositories Applications for Privacy and Security* (Singapore, Singapore). Association for Computing Machinery, New York, NY, USA.