



Building Secure Software: The Future of Security, Privacy, and Compliance

Miroslaw Staron^{ID} and Silvia Abrahão^{ID}

IN TODAY'S RAPIDLY evolving software development landscape, security has become more crucial than ever, with new libraries and frameworks emerging at an unprecedented pace. As software engineers, we must pay more attention to the latest attack surfaces and vulnerabilities. However, security engineering and software engineering are often taught as separate disciplines, which can lead to gaps in knowledge. To help software practitioners stay informed on these topics, this column highlights key security concepts and approaches discussed at the 32nd ACM International Conference on the Foundations of Software Engineering (FSE 2024), held in Porto de Galinhas, Brazil, in July 2024. We welcome your feedback and suggestions on the topics covered. Additionally, if you experiment with or adopt any of the practices featured here, please share your experiences with us and the authors of the respective papers.

Security and Privacy: A Limited Focus?

Figure 1 shows the number of conference papers addressing security

and privacy issues, categorized by the domain they address. A keyword search across the proceedings revealed that 86 of the 121 papers included terms such as “security,” “privacy,” or “vulnerability.” However, only nine explicitly discussed topics related to security and privacy, while others merely explicitly referred to papers from security-related venues or used security concerns to justify their problems. Among the nine papers, three addressed software in mobile phones (specifically Android OS), and three focused on smart contract vulnerabilities.

Guidelines and Recommendations for the Internet of Things

As the Internet of Things (IoT) continues to expand, security remains a pressing concern. With an increasing number of smart devices connecting to the Internet, safeguarding them is more critical than ever. In their paper “On the Contents and Utility of IoT Cybersecurity Guidelines,” Jesse Chen, Dharun Anandayavaraj, James C. Davis, and Sazzadur Rahman collected 142 IoT cybersecurity guidelines and analyzed 958 unique security recommendations.

Using grounded theory, they developed a taxonomy to classify these recommendations and assessed their quality. Their analysis revealed that 64% of the recommendations focused on organizational processes, such as the IoT infrastructure or administration, while 36% addressed device capabilities and requirements, such as resource protection with software (10%) or interfaces to other components. To enhance their practical value, the authors categorized the recommendations by target audience—identifying that 65.5% were relevant to manufacturers and 52.5% to organizational users.

Comparing across different guidelines, the authors identified gaps in topic coverage and comprehensiveness. By manually reviewing major IoT failures reported in news media and cybersecurity reports, they found that 1) the combined set of guidelines mitigated all 17 IoT failures identified in their news corpus, 2) 21% of the Common Vulnerabilities and Exposures (CVEs) evade the existing guidelines, and 3) no individual guideline performed particularly well. In summary, while each guideline had significant shortcomings in depth and breadth, collectively,

they can address major security issues. This research provides valuable insights for software engineers and security professionals, helping them navigate the dynamic landscape of security guidelines. The paper is available at <https://tinyurl.com/yc33kbb9>.

Addressing AI-Driven Security Threats

Deep learning (DL) and generative AI are here to stay, whether we like them or not. They will continue improving our society, but their complexity opens up new attack vectors. One such threat is patch adversarial attacks, in which perpetrators manipulate pixels in a specific region of an image to deceive DL models, either during training or inference. In their paper “CrossCert: A Cross-Checking Detection Approach to Patch Robustness Certification for Deep Learning Models,” Qilin Zhou, Zhengyuan Wei, Haipeng Wang, Bo Jiang, and Wing-Kwong Chan present CrossCert, a new tool that can detect patch adversarial attacks so that software engineers can certify their software as secure. The novelty of this tool lies in its ability to provide an unwavering certificate, that is, a guarantee that no warnings will be issued to any malicious version of a given certifiably recoverable sample. According to the paper, a benign sample is called certifiably unwavering for a certified defender if the defender can always assign the same benign label to the benign sample and all its malicious versions without issuing any warnings for the malicious samples. The tool was validated on a set of commonly used datasets, like CIFAR100 or CIFAR10, with very promising results. The paper is available at <https://tinyurl.com/2enhxrbd>.

The Hidden Cost of Dependency Bloat

Modern software relies heavily on frameworks, libraries, and reusable components. This is a good trend to some extent, but it has a dark side.

paper titled “Bloat Beneath Python’s Scales: A Fine-Grained Inter-Project Dependency Analysis.” The authors analyze more than 1,000 popular Python projects and find that more than 50% of them suffer from the

The authors analyze more than 1,000 popular Python projects and find that more than 50% of them suffer from the problem of bloated dependencies.

Our apps and software get unnecessarily big for no reason. It’s like eating an entire meal if we only need chewing gum. This phenomenon is called *dependency bloat*, and it refers to unneeded and excessive code incorporated into a project through reused libraries. This is also the problem addressed by Georgios-Petros Drosos, Thodoris Sotiropoulos, Diomidis Spinellis, and Dimitris Mitropoulos in their

problem of bloated dependencies, which means that a good diet or a gym would not hurt—well, maybe a good diet for the software and a serious software engineering gym time for the developers. They have also found that these bloated dependencies contain security vulnerabilities that are often very difficult to find in the development process. If we zoom in on these vulnerabilities, the authors found that “in nearly

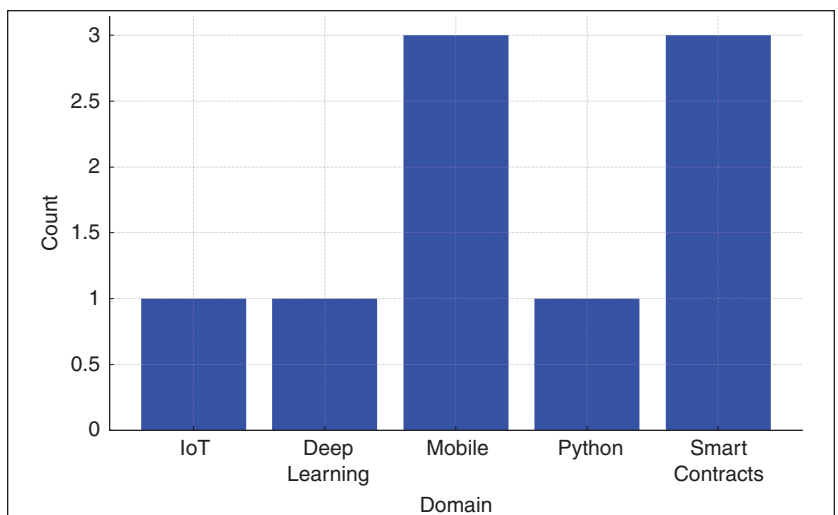


FIGURE 1. Number of security and privacy papers per domain. IoT: Internet of Things.

three-quarters of the cases (606/816), a project depends on a vulnerable package that it does not actually use.” As the authors point out, these dependencies are *time bombs* because an inactive vulnerability can be triggered when a developer changes (or adds) a method call to one of the vulnerable methods. This calls for better tools and increased awareness of secure programming in industry and university education. The paper is available at <https://tinyurl.com/277rutve>.

Cross-Layer Vulnerabilities in Mobile Apps

Dependencies are not the only way vulnerabilities creep into software products. In the world of mobile application development, cross-layer vulnerabilities are also dangerous, yet they are far from well studied. A cross-layer vulnerability means that attackers can indirectly manipulate the internal functions of mobile applications by contaminating their dependent data, including the databases, the shared preferences, or the files. Keke Lian, Lei Zhang, Guangliang Yang, Shuo Mao, Xinjie Wang, Yuan Zhang, and Min Yang, in their paper titled “Component Security Ten Years Later: An Empirical Study of Cross-Layer Threats in Real-World Mobile Applications,” developed CLDroid, an automatic tool that is the first to detect these emerging cross-layer threats. With the aid of CLDroid, the authors conducted a large-scale empirical study over 1,200 popular apps, installed more than 18 billion times, to find 56 new CVEs, 22 of them of high or critical severity. The general idea of CLDroid is to track injected data items and monitor whether they are used in security-critical scenarios. First, CLDroid identifies the data

that external apps may inject via exported components and then uses a data identifier to track the data flow. It then learns how app-specific data are used to assess security risks. A closer examination of the results reveals that the research team found some harmful behaviors that were previously unknown. For example, some data corruption in the database causes a denial-of-service attack that can persist even if an app is completely reinstalled or uninstalled. CLDroid also enables one to find other threatening new attack surfaces where the attacker can replace app font files with malicious files or change the app’s lock password and protected app list. Thanks to this tool, we can now recognize the cross-layer threats and ensure that these vulnerabilities do not become problems. Besides, the authors went one step beyond the research results and reported the CVEs to the app developers. The paper is available at <https://tinyurl.com/2fu85zx4>.

Privacy Challenges in a Rapidly Evolving Android Ecosystem

On a dynamically evolving platform like Android, updates are a means to keep the operating system safe, secure, and equipped with interesting features for developers. This is the bright side of the story, but there is also a dark side. The increased frequency of updates creates the need to update apps, programs, and frameworks. One of the challenging aspects is the privacy of the applications. Chuan Yan, Mark Huasong Meng, Fuman Xie, and Guangdong Bai address this problem in their paper titled “Investigating Document Privacy Changes in Android OS.” The authors investigate how documented privacy changes (DPCs)

impact applications and frameworks used with Android OS. The paper presents a new tool, DopCheck, a framework that parses natural language descriptions of DPCs to generate test cases, enabling the detection of inconsistencies between documentation and actual OS behavior. Combining good, old-fashioned programming using templates for apps with prompt engineering of the GPT-4 model, DopCheck can generate test cases that identify issues with privacy settings, parameters, and related application programming interface changes. The evaluation of the tool shows that it can achieve an overall success rate of 82.88%, with the best success rate in test case generation for static methods (100%) and the lowest rate for generating test cases for attributes (50%). After analyzing many apps, the study found 19 new defects in Android 13 and Android 10. The paper is available at <https://tinyurl.com/3ju3etvt>.

Securing Cross-Chain Transactions in Blockchain

The last category of security papers at the FSE conference is those discussing smart contracts. Smart contracts are the heart of distributed blockchains, to which the financial sector is paying increasing attention. As blockchain adoption grows, so do security threats. One of the challenges facing security research in this area is the protection of transactions that take place across two blockchains: cross-chain vulnerabilities (CCVs). The paper “SmartAxe: Detecting Cross-Chain Vulnerabilities in Bridge Smart Contracts via Fine-Grained Static Analysis,” by Zequin Liao, Yuhong Nan, Henglong Liang, Sicheng Hao, Juan Zhai, JiaJing Wu, and Zibin Zheng, introduces SmartAxe, a tool to detect cross-chain

bridges in smart contracts. Smart-Axe creates cross-chain control flow and data flow graphs to find semantic inconsistencies between smart contracts on both chains. The validation of the tool is based on the 88 CCVs collected by the authors. By applying the tool to 1,703 smart contracts, the authors find 232 new/unknown CCVs. The paper is available at <https://tinyurl.com/ytjyr2ck>.

Legal Compliance in Software Development

Security isn't the only concern—legal compliance is also crucial. Although security is important, and many authors are increasingly focusing on it, FSE contains many more papers from other areas. Therefore, let's look at one of the papers that

does not deal with security but secures business models from another perspective: the legal one. The paper “‘The Law Doesn't Work Like a Computer’: Exploring Software Licensing Issues Faced by Legal Practitioners,” by Nathan Wintersgill, Trevor Stalnaker, Laura A. Heyman, Oscar Chaparro, and Denys Poshyvanyk, presents a survey of 30 legal practitioners (and interviews with 16 of them) to identify conformance issues in open source licensing. The authors identified 14 challenges, for example, “Finding 2: Obtaining license exceptions is not uncommon but can be challenging. It can be difficult or impossible even to start the process if project authors cannot be identified or reached.” While software developers tend to

think of licenses in binary terms—one may either adhere to the given terms or not—lawyers may be able to help secure agreements to operate using a different model than the one the original license permits. The findings stress that compliance is a team effort among lawyers, developers, and other role players, which is something we often forget when developing software. To conclude, the authors identify the importance of community norms and the state of license enforcement as well as what is needed to make the compliance assessment process smoother: robust tooling, effective communication between lawyers and developers, and integrated and continuous license compliance. The paper is available at <https://tinyurl.com/32fdezus>.

Table 1. Takeaways from security and privacy approaches presented at the FSE 2024 conference.

Research focus	Key insights	Takeaways	Suggested applications
IoT security guidelines	IoT security recommendations are often contradictory; most focus on organizational processes rather than technical solutions.	Carefully evaluate and select security guidelines based on specific use cases.	Develop internal security policies tailored to IoT implementations.
AI and adversarial attacks	Patch adversarial attacks exploit deep learning weaknesses. <i>CrossCert</i> helps detect and prevent them.	AI-driven applications need robust security validation before deployment.	Integrate AI model security testing into machine learning workflows.
Dependency bloat and security risks	More than 50% of Python projects suffer from dependency bloat, introducing hidden security vulnerabilities.	Regularly audit dependencies to eliminate unnecessary and insecure components.	Use dependency analysis tools to reduce attack surfaces.
Cross-layer vulnerabilities in mobile apps	Attackers exploit cross-layer interactions to corrupt app data and introduce persistent threats.	Perform security assessments across all layers of mobile app development.	Use tools like <i>CLDroid</i> to detect hidden vulnerabilities in mobile applications.
Privacy challenges in Android OS updates	Frequent Android updates impact privacy settings and app compliance. <i>DopCheck</i> helps identify these issues.	Test privacy compliance with each major OS update.	Automate privacy impact assessments for app updates.
Cross-chain vulnerabilities in blockchain	Smart contract vulnerabilities in cross-chain transactions pose major security risks. <i>SmartAxe</i> identifies these issues.	Validate cross-chain transactions with security tools before deployment.	Implement automated security checks for blockchain bridges.
Legal compliance in open source licensing	Legal practitioners face 14 common challenges with open source licensing, including difficulties in obtaining exceptions.	Work closely with legal teams to ensure compliance with software licenses.	Develop internal tools to track and manage software licensing requirements.

ABOUT THE AUTHORS



MIROSLAW STARON is a full professor in the Interaction Design and Software Engineering division in the Computer Science and Engineering Department, Chalmers University of Technology and the University of Gothenburg, SE-412 96 Gothenburg, Sweden. Contact him at <https://www.staron.nu> or mirosław.staron@cse.gu.se.



SILVIA ABRAHÃO is a full professor of software engineering in the Department of Computer Systems and Computation, Universitat Politècnica de València, 46022 Valencia, Spain. Contact her at <https://sabrahao.wixsite.com/dsic-upv> or sabrahao@dsic.upv.es.

Software security, privacy, and legal compliance are becoming increasingly intertwined. As technology evolves, so must our approach to safeguarding software systems. The research presented at FSE 2024 underscores the need for stronger security measures, more efficient dependency management, and greater collaboration between developers and legal professionals. By embracing these challenges, we can ensure that future software is not just innovative but also secure, private, and legally compliant. Table 1 summarizes the key insights from the reviewed papers, highlighting several takeaways for practitioners and suggesting some applications in their contexts. 

Call for Articles

IEEE Pervasive Computing

seeks accessible, useful papers on the latest peer-reviewed developments in pervasive, mobile, and ubiquitous computing. Topics include hardware technology, software infrastructure, real-world sensing and interaction, human-computer interaction, and systems considerations, including deployment, scalability, security, and privacy.

Author guidelines:

www.computer.org/mc/pervasive/author.htm

Further details:

pervasive@computer.org
www.computer.org/pervasive

