The 19th Workshop on Programming Languages and Analysis for Security (PLAS 2024)

Lesly-Ann Daniel
DistriNet, KU Leuven
Belgium
lesly-ann.daniel@kuleuven.be

Vineet Rajani
University of Kent
United Kingdom
V.Rajani@kent.ac.uk

ABSTRACT

PLAS provides a forum for exploring and evaluating the use of programming language and program analysis techniques for promoting security in the complete range of software systems, from compilers to machine-learned models and smart contracts. The workshop encourages proposals of new, speculative ideas, evaluations of new or known techniques in practical settings, and discussions of emerging threats and problems. It also hosts position papers that are radical, forward-looking, and lead to lively and insightful discussions influential to the future research at the intersection of programming languages and security.

This year will mark the 19th edition of PLAS, which was first held in 2007 in San Diego. The workshop will host 2 keynote talks, by Natasha Fernandes and Binoy Ravindran, and 8 paper presentations.

CCS CONCEPTS

• **Security and privacy → Formal methods and theory of security**.

ACM Reference Format:
Lesly-Ann Daniel and Vineet Rajani. 2024. The 19th Workshop on Programming Languages and Analysis for Security (PLAS 2024). In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (CCS '24), October 14–18, 2024, Salt Lake City, UT, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3658644.3691336

1 TOPICS OF INTEREST

The scope of PLAS includes, but is not limited to:

- Language-based techniques for detecting and eliminating side-channel vulnerabilities
- Programming language techniques and verification applied to security in other domains (e.g. adversarial learning and smart contracts)
- Software isolation techniques (e.g., SFI and sandboxing) and compiler-based hardening techniques (e.g, secure compilation)
- Compiler-based security mechanisms (e.g. security type systems) or runtime-based security mechanisms (e.g. inline reference monitors)
- Techniques for discovering and detecting security vulnerabilities, including program (binary) analysis and fuzzing
- Automated introduction and/or verification of security enforcement mechanisms
- Language-based verification of security properties in software, including verification of cryptographic protocols
- Specifying and enforcing security policies for information flow and access control
- Model-driven approaches to security
- Security concerns for Web programming languages
- Language design for security in new domains such as cloud computing and IoT
- Applications, case studies, and implementations of these techniques

CCS '24, October 14–18, 2024, Salt Lake City, UT, USA

ACM ISBN 979-8-4007-0636-3/24/10.
https://doi.org/10.1145/3658644.3691336

2 WORKSHOP FORMAT

2.1 Invited Keynote Speakers

Natasha Fernandes (Macquarie University): *Quantitative information flow reasoning for differential privacy.*
Abstract. Quantitative information flow (QIF) is an information-theoretic framework for analysing information leaks from secure systems. In its current form, QIF derives from the g-leakage framework proposed by Geoffrey Smith (FOSSACS, 2009) and has developed into a powerful mathematical toolkit for the analysis of probabilistic systems. Differential privacy (DP) is the currently-accepted standard for privacy protection in the academic community, and has been adopted by industry giants such as Apple and Google. Despite intense research interest, there remain questions surrounding DP's interpretability and how to manage the fine-tuning of privacy so as to maximise utility, commonly called the *privacy-utility trade-off*.

In this talk I will discuss some recent work on how QIF has helped to shed light on these questions, and present some interesting challenges which remain to be addressed.

Binoy Ravindran (Virginia Tech): *A Step Toward Trustworthy Binary Verification.*
Abstract. Many production software systems are available only in binary form. This is due to several reasons including intellectual property and proprietary issues, outdated and decaying build processes and environments, and third-party libraries and tools that are no longer available or backwards compatible. Security vulnerability analysis of such software is still a necessary task due to the need to rapidly patch program errors, especially those that can be used to create security exploits, i.e., unintended, or malicious behaviors or leak sensitive data. A large body of work has focused on this problem space including disassembly, decompilation, and binary verification, among others. A common denominator of these

approaches is binary lifting: raw unstructured data is lifted to a form for reasoning over behavior and semantics. Majority of existing binary lifting approaches are untrustworthy (e.g., misses jump targets, misidentifies code as data), which negatively affects the trust base of techniques that rely on them.

In this talk, I will present an approach for binary lifting that simultaneously disassembles, recovers control flow, and generates formal proofs on the correctness of the lifted representation and a class of sanity properties including return address integrity, bounded control flow, and calling convention adherence. Establishing these properties allows the binary to be lifted to a representation that contains an over-approximation of all possible execution paths of the binary. The lifted representation contains proof obligations that are sufficient to formally prove—by exporting to a theorem-prover—the sanity properties and correctness of the lifted representation, which removes the lifting algorithm and its implementation from the trust base. We apply this approach to Linux Foundation's Xen Hypervisor covering about 400K instructions, providing evidence of its effectiveness and scalability for trustworthy binary lifting of off-the-shelf production software. I will argue that such a verified lifted representation not only reduces the trust base of downstream techniques (e.g., binary verification), but also exposes novel ways for reasoning about related problems (e.g., binary patching).

2.2 Accepted Papers

The workshop invites two types of submissions: short papers and long papers. For this edition, 5 long paper and 3 short papers have been accepted.

- **Long papers:** Papers in this category are expected to have relatively mature content. Papers that present promising preliminary and exploratory work, or recently published work are particularly welcome in this category. Long papers have a 30 min talk slots.
- **Short papers:** Papers that present radical, open-ended and forward-looking ideas are particularly welcome in this category. Long papers have a 20 min talk slots.

The workshop has no published workshop proceedings. Presenting a paper (either short or long) at the workshop does not preclude submission to or publication in other venues that are before, concurrent, or after the workshop. Papers presented at the workshop will be made available to workshop participants only.

3 WORKSHOP ORGANIZERS

3.1 Program Committee

- Mohammad M. Ahmadpanah (Chalmers University)
- Ethan Cecchetti (University of Wisconsin)
- Lesly-Ann Daniel (KU Leuven)
- Andrew Hirsch (University at Buffalo)
- Marco Patrignani (University of Trento)
- Vineet Rajani (University of Kent)
- Robert Sison (University of New South Wales)
- Pierre Wilke (CentraleSupélec, Rennes)
- Nisansala Yatapanage (Australian National University)

3.2 Workshop Chairs

Lesly-Ann Daniel is a postdoctoral researcher in the DistriNet group, at KU Leuven. She currently works on hardware-software co-designs for security. More generally, she is interested in the application of formal methods for low-level software and hardware security, including program analysis, secure compilation, hardware security extensions, etc. In recent years, Lesly-Ann has published in prestigious security conferences (CCS, Usenix Security, S&P, NDSS). She has served on the program committees of various security conferences (CSF, EuroS&P, DIMVA), and various security and programming language workshops, i.e. Principles of Secure Compilation (PriSC), Programming Languages and Analysis for Security (PLAS), Security of Software/Hardware Interfaces (SILM). In 2024, she was also a co-organizer of the "Annual Meeting of the Working Group on Formal Methods for Security of the GDR Sécurité Informatique", a 3-day seminar gathering researchers working at the intersection of formal methods and security, where she was (jointly) in charge of both the scientific and administrative aspects.

Vineet Rajani is a lecturer (Assistant professor) in the School of Computing at the University of Kent. At Kent he is a member of the Programming languages and Systems group, the Cyber Security group and the Institute of Cyber Security for Society. Vineet's interest lies in the intersection of formal verification and security analysis. Before joining Kent, Vineet was a post-doctoral researcher at the Max Planck Institute for Security and Privacy, and before that he earned his PhD from the Max Planck Institute for Software Systems. Vineet's work has won distinguished paper awards at premier venues of both security (CSF) and programming languages research (POPL). In the recent years, Vineet has co-chaired the poster track of ICICS, has been on the program committee of FCS and has served as a reviewer for JCS.

ACKNOWLEDGEMENTS

We would like to thank our reviewers for taking the time to provide timely and constructive reviews. Lesly-Ann Daniel is supported by the Research Foundation – Flanders (FWO) under grant number 12B2A24N and by the Flemish Research Programme Cybersecurity. Vineet Rajani is supported in part by the EPSRC grant EP/X015076/1.