Exploiting the SAT Revolution for Automated Software Verification: Report from an Industrial Case Study

Lucas C. Cordeiro
University of Manchester, UK
Department of Computer Science
lucas.cordeiro@manchester.ac.uk

***Abstract*—In the last three decades, Boolean Satisfiability (SAT) solvers experienced a dramatic performance revolution; they are now used as the backend of various industrial verification engines. SAT solvers can now check logical formulas that contain millions of propositional variables. In Satisfiability Modulo Theories (SMT) solvers, predicates from various theories are not encoded using propositional variables as in SAT but remain in the problem formulation. Thus, SMT solvers can be used as backends for solving the generated verification conditions to cope with increasing software complexity from industrial applications. This talk will overview automated software verification techniques that rely on sophisticated SMT solvers built over efficient SAT solvers. I will discuss challenges, problems, and recent advances to ensure safety and security in open-source and embedded software applications. I will describe novel algorithms that exploit fuzzing, explicit-state, and SMT-based symbolic model checking for verifying single- and multi-threaded software. These algorithms were the first to verify multi-threaded C/Posix software based on shared-memory synchronization and communication symbolically. They are implemented in industrial-strength software verification tools, now considered state-of-the-art in the software testing and verification community, receiving 28 medals at SV-COMP and Test-COMP. This achievement enabled industrial research collaborations with Intel and Nokia. Software engineers applied these tools to find real security vulnerabilities in large-scale software systems (e.g., memory safety in firmware for Intel and arithmetic overflow in telecommunication software for Nokia, neither of which had been found before).**

***Index Terms*—Model Checking, Software Verification, Satisfiability Modulo Theories, Security.**

I. Content of this Talk

Motivation. Memory errors in low-level systems software written in unsafe (industrial) programming languages such as C or C++ represent one of the main problems in computer security [1]. In particular, in the MITRE ranking [2], the top ten vulnerabilities include four types of memory errors (e.g., out of bounds and use after free). In addition, Microsoft reports that around 70% of all security updates in their products address memory issues [3], and Google reports a similar number regarding bugs in the Chrome Browser [4].

Research Questions. In this talk, I want to answer two main research questions based on my experience when applying software verification techniques and tools with industrial partners:

1) *Given a computer program and a specification, can we automatically verify that the program performs as specified?*
2) *Can we leverage program analysis/synthesis to discover more software vulnerabilities than existing state-of-the-art approaches in open-source applications?*

Objectives. This talk discusses the past, present, and future of software testing and verification based on SMT solving. To achieve this goal, I will consider my existing industrial collaborations, which helped shape my research in software verification. I will also consider the following observations to answer the above research questions:

- Bounded model checking (BMC) analyzes bounded program runs, thus achieving decidability.
- SAT solvers handle logical formulas with millions of propositional variables; they are now routinely used as the backend of various industrial verification engines.
- There exist better encodings using word-level theories, which can cope with increasing software complexity from industrial applications.
- Invariant inference and induction can help verify more programs than plain BMC even if we consider verifying security vulnerabilities in large-scale software systems.

II. Speaker Biography

Lucas C. Cordeiro is a Reader in the Department of Computer Science at the University of Manchester (UoM), where he leads the Systems and Software Security (S3) Research Group.[1] Dr. Cordeiro is the Arm Centre of Excellence Director at UoM; he also leads the Trusted Digital Systems cluster at the Centre for Digital Trust and Society.[2] He is also affiliated with the Formal Methods Group at UoM and the Post-Graduate Programs in Electrical Engineering and Informatics at the Federal University of Amazonas, Brazil. Dr. Cordeiro has implemented various tools used to verify safety and security properties in significant industrial programs written in Java, C/C++, and CUDA. Among those are ESBMC [5]–[8], an SMT-based model checker for C/C++/Qt and CUDA,

[1] https://www.cs.manchester.ac.uk/research/expertise/systems-and-software-security/

[2] https://www.socialsciences.manchester.ac.uk/dts/research/clusters/trusted-digital-systems/

and JBMC [9], [10], an SAT-based model checker for Java bytecode. In the last ten years, he also won 28 awards from international software verification and testing competitions held as part of ETAPS at TACAS 2012-2021 and FASE 2020-2021. Dr. Cordeiro's industrial research collaborators include Samsung, Nokia, Motorola, TP Vision, Intel, and ARM. His tools have been applied to find real security vulnerabilities in large-scale software systems. He has a proven track record of securing research funding from Samsung, Nokia Institute of Technology, Motorola, CAPES, CNPq, FAPEAM, British Council, EPSRC, and Royal Society. He leads one large EPSRC project concerning verifiable and explainable secure AI. He is Co-I on three others about software security and automated reasoning, with a portfolio of approximately 5.4m GBP.

Acknowledgment

The work presented in this invited talk is partially funded by the EPSRC grants EP/T026995/1, EP/V000497/1, EU H2020 ELEGANT 957286, and Soteria project awarded by the UK Research and Innovation for the Digital Security by Design (DSbD) Programme.

References

[1] L. Szekeres, M. Payer, T. Wei, and D. Song, "Sok: Eternal war in memory," in *2013 IEEE Symposium on Security and Privacy*. IEEE, 2013, pp. 48–62.

[2] MITRE, "Mitre's top 25 cwe," 2020, https://cwe.mitre.org/top25/archive/2020/2020_cwe_top25.html.

[3] C. Cimpanu, "Microsoft: 70 percent of all security bugs are memory safety issues," 2019, https://www.zdnet.com/article/microsoft-70-percent-of-all-security-bugs-are-memory-safety-issues/.

[4] Google, "https://www.chromium.org/home/chromium-security/memory-safety," 2020, https://www.chromium.org/Home/chromium-security/memory-safety.

[5] L. C. Cordeiro and B. Fischer, "Verifying multi-threaded software using smt-based context-bounded model checking," in *Proceedings of the 33rd International Conference on Software Engineering, ICSE 2011, Waikiki, Honolulu , HI, USA, May 21-28, 2011*. ACM, 2011, pp. 331–340. [Online]. Available: https://doi.org/10.1145/1985793.1985839

[6] L. C. Cordeiro, B. Fischer, and J. Marques-Silva, "Smt-based bounded model checking for embedded ANSI-C software," *IEEE Trans. Software Eng.*, vol. 38, no. 4, pp. 957–974, 2012. [Online]. Available: https://doi.org/10.1109/TSE.2011.59

[7] P. A. Pereira, H. F. Albuquerque, I. da Silva, H. Marques, F. R. Monteiro, R. Ferreira, and L. C. Cordeiro, "Smt-based context-bounded model checking for CUDA programs," *Concurr. Comput. Pract. Exp.*, vol. 29, no. 22, 2017. [Online]. Available: https://doi.org/10.1002/cpe.3934

[8] F. R. Monteiro, M. R. Gadelha, and L. C. Cordeiro, "Model checking C++ programs," *Journal of Software: Testing, Verification and Reliability*, vol. 31, 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/epdf/10.1002/stvr.1793

[9] L. C. Cordeiro, P. Kesseli, D. Kroening, P. Schrammel, and M. Trtík, "JBMC: A bounded model checking tool for verifying java bytecode," in *Computer Aided Verification - 30th International Conference, CAV 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, Part I*, ser. Lecture Notes in Computer Science, H. Chockler and G. Weissenbacher, Eds., vol. 10981. Springer, 2018, pp. 183–190. [Online]. Available: https://doi.org/10.1007/978-3-319-96145-3_10

[10] L. C. Cordeiro, D. Kroening, and P. Schrammel, "JBMC: bounded model checking for java bytecode - (competition contribution)," in *Tools and Algorithms for the Construction and Analysis of Systems - 25 Years of TACAS: TOOLympics, Held as Part of ETAPS 2019, Prague, Czech Republic, April 6-11, 2019, Proceedings, Part III*, ser. Lecture Notes in Computer Science, vol. 11429. Springer, 2019, pp. 219–223. [Online]. Available: https://doi.org/10.1007/978-3-030-17502-3_17