Mohannad Alhanahnah – Research Statement

Vision. My areas of interest are software engineering, programming languages, and systems security. My research to date has focused on developing automated programming systems that combine program analysis, cyber-security techniques, and machine learning to improve security [5, 13], safety [10], privacy [1], and robustness [15]. Consequently, software engineering and program analysis serve as the primary means to fulfill the objectives of my research vision.

Impact. The practical significance of my research findings is confirmed by their acceptance in top-tier software engineering and security venues such as ISSTA, TSE, USENIX Security, EuroS&P, TIFS, and NeurIPS. Beyond academic recognition, I strongly believe in the importance of technology transfer and commercialization as means to maximize societal benefits. Reflecting this belief, the technology emerging from my research has been approved for technical transfer, and a startup has been initiated with \$500K in pre-seed funding. Recently, I was awarded a grant (\$1M) to support our research project focused on utilizing Large Language Models (LLMs) for software debloating. In this project, I will serve as a Co-Principal Investigator. Following, I will delve into the details of my research and briefly outline my future research vision.

1 Research Themes

I will discuss recent research results that focus on *enforcing minimization concepts*. The interested reader may refer to the bibliography for pointers to my work in cloud trust [3, 4, 6], formal verification [2], provenance analysis [8], secure API usage [12], and IoT malware detection [7].

Software Minimization. In this research thrust, I leveraged program analysis and software engineering techniques to reduce the program's dependencies at the source code level and at the container level. Specifically, I use dynamic analysis to run the application based on a predefined set of workloads/configurations and then I use static analysis to remove unneeded components and dependencies (from program code or containerized programs). Following is a description of these tools that I developed and contributed to:

• LMCAS (EuroS&P'22 [5]): a code debloating tool that minimizes the application's footprint. It applies the concept of partial evaluation to generate specialized applications based on the supplied inputs by the user (specifies the required functionalities). LMCAS relies on the observation of the existence of a boundary between configuration logic and main logic in a given program. It then executes the program up to the boundary to capture the state of the program based on the supplied inputs. LMCAS finally applies a set of customized compiler optimizations to remove unused components and dependencies. LMCAS has been accepted for the Tech Transfer program funded by the Office of Naval Research (ONR). ONR also invited us to deliver LMCAS demo at the second Software Security School. Finally, a patent was filed for LMCAS technology.

• **BLAFS** [17] a *container debloating* tool that removes unneeded dependencies from containerized applications. In this study, we maintain the essential functionality but also introduce the ability to call on-the-fly dependencies from the original container. This ensures that the debloated container's functionality remains unbroken. Furthermore, in a previous study [16], we explored the bloat in machine learning containers and evaluated its effects on both security and performance.

Data Minimization. This thrust aims to protect users' privacy by eliminating the sharing of unneeded user data in Trigger-Action Platforms (TAPs).

• minTAP (USENIX'22 [1]): improves the privacy of TAPs by using languagebased data minimization. minTAP generates specialized minimizers that enforce the principle of least privilege by releasing only the necessary user data attributes to TAPs and fending off unrelated API access. The minimizers can be generated statically or dynamically. The integrity of the minimizers is protected by maintaining their digital signatures. We deploy minTAP on IFTTT, showing how to minimize trigger data before they are sent, thus boosting privacy while preserving the functionality. A patent has issued corresponding to minTAP technology.

2 Future Research Agenda

I would like to direct, but not limit, my future research efforts to the following research problems:

- Software Supply Chain Security: The modern software supply chain, however, includes additional dependencies beyond the applications' code. These dependencies extend to areas like Continuous Integration (CI) and Continuous Development (CD), which involve workflow automation and containerization. Therefore, it is imperative to consider these aspects in any comprehensive solution, calling for the development of a holistic dependency tree that traces the chain of dependencies across all necessary and unnecessary components. I plan to leverage my previous work in software debloating [5, 9] and ongoing collaborations from Chalmers University on container debloating [16, 17] to construct this tree, which will prove invaluable for various types of analysis, including threat and security evaluations. I also intend to establish research in the realm of software bots. My particular emphasis will be on evaluating the safety and security aspects of software bots.
- LLM Applications Safety: Drawing on my prior experience in investigating interactions on Android [13, 14] and IoT platforms [10, 11, 2], this project proposes to identify the safety and security challenges prevalent in the Large Language Models applications ecosystem. The initial phase of the research will be dedicated to pinpointing the necessary safety and security characteristics that should be sustained in these systems. Subsequent to this identification process, the proposal aims to devise an enforcement mechanism that mandates the incorporation of these properties, ensuring the effective and secure operation of the LLMs in diverse applications.
- Green Coding: This concept encourages, among other principles, the removal of unused software features, a move that improves energy efficiency and ease of

maintenance. It is a necessary step as around 90% of software developments utilize open-source code, which often includes redundant sections that consume excess processing power and emit unnecessary carbon. My research in this domain envisions sustainability challenges as a responsibility to reduce carbon footprint. Therefore, I will explore the use of software debloating and compiler optimization techniques to enhance the efficacy of such in reducing hardware requirements and, subsequently, energy consumption and carbon emissions. To realize this, I intend to apply for research grants from programs committed to environmental sustainability and addressing climate change issues.

References

- Y. Chen, M. Alhanahnah, A. Sabelfeld, R. Chatterjee, and E. Fernandes. Practical Data Access Minimization in Trigger-Action Platforms. In USENIX Security Symposium, 2022.
- [2] C. Stevens, M. Alhanahnah, Q. Yan, and H. Bagheri. Comparing Formal Models of IoT App Coordination Analysis. In SEAD 2020.
- [3] M. Alhanahnah, P. Bertok, and Z. Tari. Trusting Cloud Service Providers: Trust Phases and a Taxonomy of Trust Factors. *IEEE Cloud Computing*, 4(1):44–54, Jan 2017.
- [4] M. Alhanahnah, P. Bertok, Z. Tari, and S. Alouneh. Context-Aware Multifaceted Trust Framework For Evaluating Trustworthiness of Cloud Providers. *Future Generation Computer Systems*, 79:488 – 499, 2018.
- [5] M. Alhanahnah, R. Jain, V. Rastogi, S. Jha, and T. Reps. Lightweight, multi-stage, compiler-assisted application specialization. In 2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P), pages 251–269. IEEE, 2022.
- [6] M. Alhanahnah, A. Jhumka, and S. Alouneh. A Multidimension Taxonomy of Insider Threats in Cloud Computing. *The Computer Journal*, 59(11):1612–1622, 11 2016.
- [7] M. Alhanahnah, Q. Lin, Q. Yan, N. Zhang, and Z. Chen. Efficient Signature Generation for Classifying Cross-Architecture IoT Malware. In 2018 IEEE Conference on Communications and Network Security (CNS), pages 1–9, May 2018.
- [8] M. Alhanahnah, S. Ma, A. Gehani, G. Ciocarlie, V. Yegneswaran, S. Jha, and X. Zhang. autoMPI: Automated Multiple Perspective Attack Investigation with Semantics Aware Execution Partitioning. *IEEE Transactions on Software Engineering*, 2022.
- [9] M. Alhanahnah, P. Schubert, N. Gupta, T. Reps, S. Jha, and E. Bodden. Slash: Static configurationlogic identification. Under submission.
- [10] M. Alhanahnah*, C. Stevens*, and H. Bagheri. Scalable Analysis of Interaction Threats in IoT Systems. In ISSTA 2020, 2020 (*Equal contribution) (ACM SIGSOFT Distinguished Paper Award).
- [11] M. Alhanahnah, C. Stevens, B. Chen, Q. Yan, and H. Bagheri. IoTCOM: Dissecting Interaction Threats in IoT Systems. *IEEE Transactions on Software Engineering*, 2022.
- [12] M. Alhanahnah and Q. Yan. Towards best secure coding practice for implementing SSL/TLS. In IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WK-SHPS), pages 1–6, April 2018.
- [13] M. Alhanahnah, Q. Yan, H. Bagheri, H. Zhou, Y. Tsutano, W. Srisa-an, and X. Luo. Detecting Vulnerable Android Inter-App Communication in Dynamically Loaded Code. In *IEEE INFOCOM* 2019 - *IEEE Conference on Computer Communications*, pages 550–558, April 2019.
- [14] M. Alhanahnah, Q. Yan, H. Bagheri, H. Zhou, Y. Tsutano, W. Srisa-an, and X. Luo. Dina: Detecting hidden android inter-app communication in dynamic loaded code. *IEEE Transactions on Information Forensics and Security*, 15:2782–2797, 2020.
- [15] Y. Wang, M. Alhanahnah, X. Meng, K. Wang, M. Christodorescu, and S. Jha. Robust learning against relational adversaries. In Advances in Neural Information Processing Systems (NeurIPS), 2022.
- [16] H. Zhang, F. Abdulqadir Ahmed, D. Fatih, A. Kitessa, M. Alhanahnah, P. Leitner, and A. Ali-Eldin. Machine learning containers are bloated and vulnerable. arXiv preprint arXiv:2212.09437, 2022.
- [17] H. Zhang, M. Alhanahnah, and A. Ali-Eldin. Blafs: A bloat aware file system. Under submission.