Mohannad Alhanahnah – Research Statement

Vision: I take a broad view of software security, which is beyond just developing scripts and then identifying the existence of vulnerabilities. Software is conquering the world. There is hardly a piece of equipment that does not have software in it today. Not only does equipment have software to help with internal functions, but it also has begun to relate externally to other software-aided devices. This observation entails software systems comprising several components, and these components are not working in isolation anymore. These emergent software systems interact not only internally but across systems. This interaction brings ample benefits to facilitate human society by enabling automation and digitalization. However, this interaction can hinder traditional security mechanisms, complicate conducting security analysis, and introduce new classes of threats. Addressing these challenges will help me to achieve my ultimate goal to make the use of software systems *safer* for our society and developers.

To achieve this vision, my research approach relies on three pillars: (1) developing appropriate abstraction to facilitate understanding the nature of interactions in emergent platforms, (2) defining threat models corresponding to the interaction these interactions (3) hardening software systems in isolation before exposing them by applying minimization and enforcing the classic principle of least-privilege. Through systems design and program analysis, my research seeks to improve security [5, 12], safety [9], privacy [1], and robustness [14] guarantees in these emergent platforms.

My Ph.D. research employs the first and second pillars to develop a rich understanding of the security and safety consequences as a result of *interactions across various components* on the IoT and Android platforms. While my postdoc research applies the third pillar to harden software systems by *enforcing minimization and normalization concepts*.

Impact: my findings are recognized for their practical significance and impact on the community, as evidenced by acceptance in top-tier security and software venues such as USENIX Security Symposium, IEEE European Symposium on Security and Privacy (EuroS&P), ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA), and Neural Information Processing Systems (NeurIPS). I believe that if there are ways to benefit society after publishing papers, technology transfer and commercialization are important channels to reflect such a positive impact. Consequently, the technology developed throughout my research has been accepted for technical transfer and received $500K pre-seed funding to launch a startup company.

In the following, I elaborate more on my Ph.D. and postdoc research. I conclude by briefly discussing my vision for future research work.

1 Research Themes

I will discuss recent research results that focus on two areas: (1) enforcing minimization and normalization concepts and (2) understanding the security and safety consequences of interactions in emergent platforms. 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 [11], and IoT malware [7].

1.1 Postdoc research

My postdoc research focuses on hardening software systems and involves two research thrusts: (1) applying software and data minimization, and (2) improving the robustness of Machine Learning (ML) systems.

1.1.1 Software and Data Minimization

- **LMCAS (EuroS&P'22 [5])**: a software debloating tool that minimizes applications 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. Our evaluation shows that LMCAS reduces the binary size of the debloated program while preserving the required functionalities. It also reduces the attack surface by removing known vulnerabilities among the unneeded components.

  Impact: 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 organized by ONR Total Platform Cyber Protection (TPCP). LMCAS constituted a cornerstone towards cofounding FitStack, a startup company to commercialize the research collaboration with my postdoc mentor Prof. Jha. This work also led to the establishment of collaboration with Prof. Eric Bodden's research group to automate the process of identifying the boundary between configuration and main logic. This collaborative work is currently under submission. Finally, Wisconsin Alumni Research Foundation (WARF) filed a patent corresponding to LMCAS technology.

- **minTAP (USENIX'22 [1])**: improves the privacy of TAPs by using language-based data minimization. It 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.

  Impact: WARF filed a patent corresponding to minTAP technology.

1.1.2 Machine Learning Robustness

- **N&P (NeurIPS'22 [14])**: this work is motivated by my prior work [7] that leverages ML for generating succinct signatures to detect IoT malware based on structural, statistical, and string features. We observed these ML detection systems are susceptible to semantic-preserving attacks that aim to evade detection.

  For addressing this line of attacks, my collaborators and I devised a learning framework that leverages input normalization to achieve provable learning robustness against relational adversaries, who create adversarial examples in a reflexive-transitive closure of a logical relation. Specifically, the adversary can manipulate the original test inputs via transformations specified by a logical relation. Therefore, these attacks are beyond ℓ_p-norm bounded attacks, because the relational adversary can apply an arbitrary sequence of transformations to the inputs as long as the essential semantics of the input is preserved. N&P solves the pain points of adversarial training against relational adversaries and can be combined with adversarial training for the benefit of both approaches. First, it converts each data point to a canonical form and subsequently restricts the training and testing of models on the normalized data. We then combine N&P with adversarial training in an attempt to achieve the optimal robust-accuracy trade-off. We empirically evaluated N&P against source code authorship attribution and malware detection attacks. The evaluation shows N&P significantly improves the robustness of models against relational adversaries.

  Impact: this work has been designated as an "Oral" at NeurIPS 2022.

1.2 PhD Research: Security and Safety analysis for Interactions on Android and IoT platforms

- **IoTCOM (ISSTA'20 [9], TSE'22 [10]):** a formal method tool to identify safety and security violations that can occur due to the interactions between IoT apps in smart home environments. IoTCOM is a compositional approach that empowers end users to protect a given bundle of cyber and physical components co-located in an IoT environment. It automatically discovers such complicated interaction threats. IoTCOM combines static analysis with lightweight formal methods to automatically infer relevant specifications of IoT apps in an analyzable formal specification language by taking into consideration the mapping between the cyber and physical channels. IoTCOM then checks the extracted specifications as a whole for interaction threats.

  Impact: IoTCOM received the ACM SIGSOFT Distinguished Paper Award.

- **Dina (INFOCOM'19 [12], TIFS'20 [13]):** exposes a new attack that leverages reflection and dynamic class loading features in conjunction with inter-app communication to conceal malicious attacks to bypass existing security mechanisms. We show that the interaction between apps can lead to privacy leakage and spoofing attacks. To identify such vulnerabilities, we designed, developed, and implemented Dynamic INter-App Communication Tool (Dina), a novel hybrid analysis approach for identifying malicious IAC behaviors concealed within dynamically loaded code through reflective/DCL calls. Dina appends reflection and DCL invocations to control-flow graphs and continuously performs incremental dynamic analysis to detect the misuse of reflection and DCL that obfuscates Intent communications to hide malicious IAC activities. Dina utilizes

string analysis and inter-procedural analysis to resolve hidden IAC and achieve superior detection performance.

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 use of open-source software and third-party libraries is rapidly growing. However, these libraries can be malicious (i.e., SolarWinds attack) and/or vulnerable (i.e., Log4j). Therefore, the US government issued a federal order that mandates the need for a Software Bill of Materials (SBOM), which will facilitate the identification of such libraries. Consequently, this order mandated that US government agencies work only with software vendors that provide SBOMs. In prior work, LMCAS [5], we developed a hybrid analysis approach for debloating C/C++ programs. This approach identifies necessary components (libraries) and removes unneeded components. This research thrust aims to leverage these concepts of software debloating to develop approaches to specialize applications that will mitigate software supply chain attacks. Specialized programs will obey the *principle of least privilege* (PLP) and the principle of *privilege separation* (PS). In my current collaboration with colleagues from the Chalmers University of Technology, we studied dependencies in the context of machine learning containers [15], determined the amount of bloated (unneeded dependencies), and quantified the security impact of these dependencies in terms of the number of CVEs. This work shows that ML containers are significantly bloated, but removing unnecessary files can speed up the provisioning by up to $3.7X$ and eliminate 98% of the vulnerabilities. Therefore, in the short term, it is necessary to develop a holistic dependency tree that captures the chain of dependencies across needed and unneeded components. These defense-in-depth strategies can also mitigate some classes of known and unknown vulnerabilities. In the long term, I plan to make the use of open-source software safer, as it is becoming a public good, which is motivated by the fact that approximately 90%[1] of software developments leverage open-source code. Consequently, governments established funding programs to support efforts to secure software supply chains and open-source software. Therefore, I plan to submit proposals to these programs like Pathways to Enable Open-Source Ecosystems (POSE) funded by NSF. This research direction constitutes a stepping stone for my career proposal.

- **Security of Robot Operating System:** in my previous work, I addressed the challenge of detecting unsafe interactions in the context of Android and smart homes. This challenge can also be studied in the robotics ecosystem, as robots comprise a set of sensing and actuation components that interact with each other. The robot operating system (ROS) is one of the prominent frameworks that is used in the robotics ecosystem. Therefore, the usage of ROS is expected to grow in industrial applications and academic research. However, the security of ROS constitutes a major concern that threatens to delay the development of robotics systems. In the short term, I plan to investigate and identify unique

[1] https://www.infoworld.com/article/3245308/rethinking-your-open-source-use-policy.html

challenges about ROS, including the complexity of ROS architecture, the different programming languages used to develop the ROS framework, and the reality that ROS is open source that incorporates third-party libraries. I plan to use static analysis and formal method techniques to pursue this work.

- **Generating Specialized Adversarial Attacks:** malware attacks these days are targeted. For example, during COVID-19, we observed several attacks targeting medical institutions or certain countries. A targeted malware attack means that the malware will be specialized to infect a certain environment and under a set of pre-defined settings. As a result, this specialized malware can bypass ML malware detectors. To enhance the robustness of these detectors, existing studies generate adversarial malware to train the detectors. However, these studies take advantage of perturbation-space-based statistical features but do not consider structural features or their conjunction. In this thrust, I improve the robustness by generating specialized adversarial malware, wherein the perturbation space is explored based on statistical and structural features. I will employ partial evaluation approach to generate the specialized adversarial malware. I will specialize certain APIs that are used to target a specific country (i.e., location/language APIs) or particular environment (i.e., framework or operating system). I envision this research as an extension of my prior work N&P (NeurIPS'22 [14]). Therefore, the impact of this work is devising mechanisms to normalize the use of APIs that can potentially be specialized to evade detection. I would also like to extend the scope of this work beyond malware detection to other domains such as code authorship. For funding this project, I plan to submit research proposals to DARPA and ONR.

- **Green Coding:** the challenge for society globally, however, is that our use of data is only going to increase as we continue to digitize entire sectors and even countries. Therefore, large companies have started supporting environmental sustainability initiatives to manage carbon emissions. For example, Google has set itself an ambitious target of running its whole data center estate on carbon-free energy by 2030. Meanwhile, Amazon Web Services (AWS) is on track to power its entire operations with 100% renewable energy by 2025. One way carbon management can be achieved is by adopting Green Coding practices. Green coding is a term recently popularized for its environmental intentions and refers to programming code that is written to produce algorithms that have minimal energy consumption. It consists of a collection of processes and principles aimed at reducing the energy consumption of software, which requires software engineers to be more considerate of the code they are writing and how efficient it is. Among these principles is the removal of unused features, which improves energy efficiency and makes software easier to maintain. This removal is necessary because approximately 90% of software developments leverage open-source code that is not tailored for specific uses and applications. Therefore, it contains redundant and unnecessary sections of code, using up more processing power and producing unnecessary carbon emissions. In this research, I am eager to study and boost the impact of applying software debloating techniques on reducing hardware requirements (i.e., CPU, memory, storage, network), thus reducing energy consumption and carbon emission. I plan to submit research grants to fund this project to programs that promote environmental sustainability and address climate change challenges.

References

[1] 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***, C. Stevens*, and H. Bagheri. Scalable Analysis of Interaction Threats in IoT Systems. In *ISSTA 2020*, 2020 (*Equal contribution) (**ACM SIGSOFT Distinguished Paper Award**).

[10] **M. Alhanahnah**, C. Stevens, B. Chen, Q. Yan, and H. Bagheri. IoTCOM: Dissecting Interaction Threats in IoT Systems. *IEEE Transactions on Software Engineering*, 2022.

[11] **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 WKSHPS)*, pages 1–6, April 2018.

[12] **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.

[13] **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.

[14] 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.

[15] 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.