ADWAIT NADKARNI - RESEARCH STATEMENT

I am internationally recognized for my research in mobile and IoT security. In particular, my research develops automated techniques for *systematic evaluation of security analysis tools*, thereby improving the state of security analysis and bringing tangible security gains to users of commodity software.

Motivation: Governments and standards bodies have only recently begun to grapple with the reality of billions of potentially vulnerable Internet of Things (IoT) products, and have responded with the promise of targeted security and privacy regulations. The practical effectiveness of these initiatives depends on their enforcement, which has led to an ever-increasing demand for *automated security/vulnerability analysis tools* from industry (*e.g.*, Coverity and LGTM), academia (*e.g.*, FlowDroid, CryptoGuard, and CogniCrypt), and the open source community (*e.g.*, SpotBugs). Open source platforms are also encouraging developers to integrate security analysis tools as a part of their CI/CD pipeline (*e.g.*, Github Code Scan). Thus, the ability of automated tools to detect vulnerabilities directly influences the security guarantees availed by the user. However, while security tools have proliferated, their *effectiveness in practice* is relatively understudied.

Research Overview: I am working towards establishing a new area within security research focused on the *systematic and practical evaluation of security analysis tools*. My research helps answer a fundamental question: how effective are security analysis tools at detecting vulnerabilities in practice? To this end, I develop *automated testing frameworks* that generate *test cases* for evaluating security tools thoroughly and systematically, in a manner that is contextualized to the tools' security goals and application domain.

My research explores this new area along three complementary research directions: (1) ***developing comprehensive test cases*** that not only evaluate a tool's ability to detect the most obvious expression of a vulnerability (*i.e.*, the *base* case), but also a broad space of possible *variants* that developers may introduce in end-user software due, (2) ***generating natural scenarios for testing*** in highly contextual domains such as IoT, where the behavior and effectiveness of a security tool may vary based on runtime context it is exposed to in the wild, and, (3) ***the analysis/design of security/privacy protections*** that grounds and complements the evaluation of security tools. The remainder of this statement will describe the importance of these three directions and their impact.

Comprehensive Test Cases using Mutation Testing

My research [1, 2, 3, 4] (supported by NSF-1815336, *SaTC small*, **PI**) leverages the well-founded approach of *mutation testing* from the software engineering (SE) domain to comprehensively test static analysis security testing (SAST) tools. The overall approach is intuitive, and as follows: I design mutation frameworks (*e.g.*, μSE [1, 2, 3] and MASC [4]) that generate variants or *mutants* of vulnerabilities that the targeted tool claims to detect and seeds the mutants in benign software. The tool then analyzes the mutated software, and any *uncaught/unkilled* mutants lead to the discovery of flaws in the tool's analysis. To make mutation testing useful for evaluating security tools, my research addresses the foremost challenge of *contextualizing mutation analysis to security*, and particularly, the unique intricacies of the problem (*e.g.*, cryptographic API misuse, private data leaks) and application domains (*e.g.*, Android, Java) targeted by SAST tools. Through this contextualization of mutation testing, my research makes novel contributions to both security and SE.

Developing novel mutation abstractions for security in μSE (USENIX'18, TOPS'21, ICSE Demo'21): I led the development of a framework for Mutation-based Soundness Evaluation (μSE, read as "muse") [1, 2, 3], which evaluates Android security analysis tools, and particularly data leak detectors (*e.g.*, FlowDroid, Argus) to uncover *unsound* design and implementation decisions. μSE systematizes the use of mutation testing for security, and introduces two novel abstractions: (1) *security operators*, which are security-focused mutation operators (*i.e.*, transformations on code) that *represent a high-level security goal* (*e.g.*, data leak detection), and hence define *what* the mutants should express, and (2) *mutation schemes*, *i.e.*, strategies for placing/seeding mutants in applications that further enhance their expressiveness, influenced by factors such as the security goal, Android-specific abstractions, and code-reachability, which define *how* the mutants are instantiated. μSE's novel approach is also highly effective: an evaluation of 7 popular data leak detectors for Android using thousands of *compilable* and *executable* mutants generated using μSE uncovered 25 previously undocumented design/implementation flaws. Our *flaw propagation* study demonstrated that several flaws were inherited from prior tools, and many were surprising to the tool developers themselves. Moreover, we were able to patch 2 flaws in FlowDroid, improving its analysis.

Comprehensive evaluation of crypto-detectors with MASC (S&P'22): Critical vulnerabilities aris-

ing from misuse of cryptographic APIs are common in end-user software. Thus, while μSE laid the groundwork for evaluating security tools with mutation testing, its findings motivated my research into the effectiveness of another highly security-critical class of tools: *crypto-detectors*, or tools that detect crypto-API misuse, which are commonly used by developers (*e.g.*, the CogniCrypt Eclipse plugin), industry security teams (*e.g.*, CryptoGuard in Oracle's internal test suite), and in CI/CD pipelines (*e.g.*, Github Code Scan).

I led the design of a framework for Mutation-based Analysis of Static Crypto-misuse detectors (MASC, pronounced as mask) [4], which addresses the key challenges for contextualizing mutation testing to evaluate crypto-detectors. First, MASC's mutants must represent *all relevant crypto-API misuse* cases arising in the vast and complex realm of crypto-APIs. Hence, we constructed the the first *comprehensive taxonomy of crypto-API misuse*, consisting of 105 crypto-API misuse cases along 9 semantic clusters, via a data-driven process that systematically identifies, studies, and extracts misuse cases from all academic and industrial sources from 1999-2021. Second, to expressively instantiate the misuse cases from the taxonomy in *all possible ways developers may misuse them*, we define the novel abstraction of *usage-based mutation operators*, *i.e.*, general mutation operators that leverage the common usage characteristics of diverse crypto-APIs (*e.g.*, the Java Cryptography Architecture (JCA)). The design of our mutation operators is guided by a *threat model* that represents realistic threat scenarios that crypto-detectors face in practice. Finally, we designed the novel abstraction of *mutation scopes* for seeding mutants of variable fidelity to realistic API-use and threats.

We evaluated 9 popular crypto-detectors from industry and academia using $20,303$ compilable mutants generated by MASC, and revealed 19 previously unknown flaws. These flaws directly impact the security experienced by the end-user: we found vulnerabilities represented by *all of the flaws* in popular open source software. Our interactions with tool designers during the flaw reporting process revealed key insights regarding the factors influencing the current design and evaluation of crypto-detectors. My hope is to use the lessons learned from MASC to further evaluate and improve security-critical SAST tools, particularly those used by third-party labs that offer security assessments and compliance certifications to product vendors.

Practical Evaluation with Natural Scenarios

Tools built to analyze security in the smart home are generally evaluated using either random permutations of smart home event, or event sequences from *IoT apps* (*i.e.*, developer-defined home automation routines), neither of which capture the realistic events that may actually happen in an end-user's home due to home automation usage. During my research on the smart home security [5, 6, 7], I observed that platform-provided user interfaces empower users to create *user-driven routines* by combining triggers and actions *without writing a single line of code*, a realization of the end-user programming paradigm in smart homes. This observation, coupled with the intuition that users may prefer easily created routines that represent their automation requirements over developer-defined IoT apps, motivated my research into developing a *realistic characterization of home automation* for a practical evaluation of smart home security tools.

Generating natural smart home scenarios with Hϵlion (S&P'20): My research developed the Hϵlion framework [7] (supported by NSF-2132281, *CPS medium*, **PI**), which uses statistical language models (LMs) to capture the regularities in user-driven home automation, and leverage them to generate *scenarios*, *i.e.*, event sequences that reflect home automation usage in the wild. Hϵlion's approach builds upon the notion of *naturalness* from the NLP and SE domains, which states that while languages (*i.e.*, natural languages like English or programming languages like Java) are extremely expressive in theory, their use in practice by people is "natural", *i.e.*, exhibits patterns that make it predictable. I observed that *user-driven routines* are effectively expressions of programs created by humans, and that the routines scheduled by the user to execute throughout the day is essentially an expression of the user's program. This key observation led to the *naturalness hypothesis for home automation*, which states that the *home automation event sequences* resulting from an ordered execution of routines scheduled by the user are inherently natural, *i.e.*, exhibit semantic patterns that make them predictable, and hence, we can use statistical LMs to to analyze corpora of such sequences and predict *natural* home automation scenarios *useful* for security.

We designed Hϵlion as an end-to-end framework that (1) collects routines, (2) serializes them into home automation sequences guided by user intuition, (3) models the generated corpus of home automation sequences from users using n-gram LMs, and (4) generates scenarios as a sequence generator, *i.e.*, by successively predicting the next most probable event given a history of events that has already occurred. Our evaluation with a corpus of home automation sequences generated from 40 users led to a key result, *i.e.*, we

demonstrated that home automation sequences are far more natural than language or software corpora (*i.e.*, exhibit low perplexity), and can be effectively modeled using statistical LMs. Our second evaluation with 18 external evaluators demonstrated that users find Hєlion's scenarios to be valid, or reasonable enough to occur in end-user homes. We designed two additional systems for using scenarios for security: a *snapshot module* that would allow an event-by-event analysis of a scenario's effect on various device/home states, and an *execution engine* based on the SmartThings platform for dynamically testing with scenarios. Our evaluation demonstrated that Hєlion's scenarios were useful for key tasks in designing and evaluating security systems for the smart home, *i.e.*, we automatically *generated 17 security policies* using the snapshot module, and *discovering 3 bugs* in SmartThings' runtime and a popular door lock using the execution engine.

Analysis and Design of Security and Privacy Protections

My experience in designing tools and evaluating existing security and privacy protections [8, 9, 5, 6, 10] continues to inform my research agenda on evaluating security tools, and I plan to continue research in this reciprocal research direction. More importantly, my prior work on building defenses for modern commodity platforms [11, 12, 13, 14, 15, 16, 17, 18] has shown that software security analysis must be complemented with strong guarantees in the operating system, to account for vulnerabilities or malware that escape analysis.

Smart Home Security/Privacy Evaluation (CODASPY'19 Best Paper, ACM TCPS, USENIX'22): My recent security evaluation of the NEST and HUE smart home platforms (*CODASPY 2019 Best Paper*) demonstrated the first instance of a *lateral privilege escalation* in the smart home [5, 6]. To elaborate, an attacker could compromise a vulnerable, low-integrity, component connected to the home, *e.g.*, steal an authentication token by performing a MiTM attack on the TP Link Kasa app with an SSL vulnerability, and leverage the compromised component's API-access to make changes to the home's state (*e.g.*, change "away" to "home"), thereby indirectly causing a device dependent on the state (via routines/automation) to be modified (*e.g.*, allowing us to remotely turn off the NEST security camera without having to directly attack it). This attack motivates my work on platform-level defenses that provides integrity guarantees for sensitive states against adversaries with API access. Further, our recent large scale evaluation of the privacy policies of smart home vendors [10] (to appear in USENIX Security'22) uncovered serious flaws, *e.g.*, 49% of vendors do not offer device-related privacy policies, whereas 11% do not offer privacy policies at all, and even when they do, privacy policies are often incorrect or incomplete. This result echoes prior finding, but has far worse implications given the sensitive nature of data collected by smart home products, and motivates work on building privacy policy analysis tools adapted to recognize smart home data in privacy policies.

HomeEndorser (major revision submitted to USENIX'22): To address the problem exposed by our evaluation of NEST and Hue [5, 6], my research developed the HomeEndorser framework [19], which enables integrity guarantees for changes to abstract home objects (*e.g.*, home/away) by third-party integrations via platform API. HomeEndorser significantly deviates from prior work as it does not rely on access to third-party integration/app code, which is beneficial given that most platforms (*e.g.*, SmartThings v3, NEST) treat third-party integrations as black boxes hosted on third-party clouds. Instead, HomeEndorser introduces the paradigm of *home abstraction endorsement* leveraging a unique opportunity given the cyber-physical nature of the smart home: it uses the ground truth observations from devices, *i.e.*, the "local context" of the home, to *endorse* and proposed change to an AHO by an untrusted integration. We propose an expressive policy model and a platform-based reference monitor that provides both complete mediation and tamperproofness, and implement HomeEndorser on the popular HomeAssistant platform. Our evaluation demonstrated HomeEndorser's effectiveness, low false positives and overhead, and minimal platform integration effort.

Finally, while laying the groundwork on effective evaluation and development of security tools, my research has led to several open-source tools and artifacts that are available for future research, and have led to measurable impact. Particularly, my Android Security Modules (ASM) framework [14] (US Patent 9,916 ,475 B2) has been used by 65 researchers from 10 industrial and 35 academic institutions. My privacy-preserving tool for Android, NativeWrap, has been downloaded over 10000 times since it was released on Google Play. My research on IoT security routinely receives press coverage, and my prior work on Android security analysis received an "Android Security Acknowledgement" and 3 CVEs. Finally, I was recently **nominated for the 2022 Commonwealth Cybersecurity Initiative (CCI) Impact Award** within the state of Virginia, which recognizes breakthrough security research and its impact.

Future Research Directions

I plan to continue my work in the broad area of systematic evaluation of security analysis tools along two research directions. I am particularly interested in understanding the root causes and impact of problems in SAST tools, and am qualitatively exploring this problem by interviewing both security experts that build them, and software developers that use them. At present, we have surveyed and interviewed over 30 software developers who use SAST tools, and uncovered several interesting insights regarding their use of the tools, expectations from them, and actions/impact resulting from detection failures. I plan to continue along these lines to interview tool designers as well, and identify any conflict in their understanding of what developers should expect from SAST tools. My vision is to understand and analyze the (often contrasting) perspectives from both developers and tool designers and identify concrete opportunities for improving SAST tools.

Further, I intend to build upon my work on evaluating automated security tools and leverage it to improve the state of the security compliance infrastructure, with a focus on IoT security and privacy compliance. Particularly, I intend to develop data-driven techniques for evaluating the vulnerability detection performance and general processes of Commercially Licensed Evaluation Facilities (or CLEFs), which are licensed by regulators to evaluate end-user products (*e.g.*, IoT apps, software, and devices). To enable this data-driven understanding of the vulnerabilities relevant in the IoT context, my team recently developed a method to collect mobile-IoT apps on a large scale, resulting in *the largest extant dataset of 37k mobile-IoT apps* [20]. In addition, I also plan to build an understanding of the CLEFs' scope of work (*i.e.*, what they claim to detect) using a survey/interview methodology, leveraging my collaborations with CLEFs such as *IoTInspector* and *Xanitizer*, and product vendors such as *IBM* and *Mojo Vision*. This data will guide a mutation framework that will generate non-trivial vulnerability instances (i.e., mutants) for analysis by CLEFs. In parallel, I plan to train a vulnerability detector on a large corpus of mutants, overcoming the key obstacle of the lack of non-trivial positive samples faced by machine learning-based vulnerability detectors, which will serve as a baseline for evaluating CLEFs and help vendors improve their products. This research will transform compliance enforcement by enabling the regulators and policymakers to incentivize accountability in CLEFs through empirical checks and balances grounded in real data, in addition to helping CLEFs and vendors improve their security posture.

Furthermore, I am also an active member of the *Philadelphia Smart City Task Force*, which will help me engage with policymakers on this topic and amplify impact. Finally, I recently co-led a Birds of a Feather (BoF) session at the 2022 NSF Secure and Trustworthy Cyberspace (SaTC) PI meeting, with the goal of building a community with a common interest towards improving the state of security tools used in security critical use-cases such as compliance.

Mentoring

I lead the *Secure Platforms Lab* at William & Mary, consisting of four PhD students: Kaushal Kafle (lead graduate student), Amit Seal Ami, and Prianka Mandal, and an undergraduate student, Sayyed Razmjo. Sunil defended his dissertation in June 2022, and recently joined the IBM T.J. Watson Research Center as a research scientist in August 2022. I have also graduated a Masters student (Brian Burns), and five undergraduate students: Kunyang Li (Charles Center Scholar 2021, Honors thesis, currently PhD student at University of Wisconsin CS), Ruhao Tang (Charles Center Scholar 2018, Park Award 2019, Honors thesis, and currently employed at the MIT Lincoln Laboratory), Rozda Askari (contributed to μSE [1]), Caleb Atkins (independent study), and Hanqiu Peng (independent study).

As the advisor and research group leader, my role is not only limited to advising research, but also helping the students in achieving their short and long-term career goals. My experience has taught me the value of patience, diligence, foresight and the ability to bring people with a diverse array of personalities and skills together as a strong team. In recognition of my mentoring efforts, I received the **2021 Arts & Sciences Graduate Faculty Mentoring award by William & Mary**, which has generally been awarded to senior (tenured) faculty. I have also fostered a culture of effective mentoring in my lab; my lead graduate student, Sunil Manandhar, recently received the **2021 Sanderson Award for Undergraduate Mentoring** from the the college of Arts & Sciences, William & Mary. Through a far-reaching research vision coupled with effective mentoring, my goal is to develop my students into strong security researchers, and bring about tangible research advancements to the art of security analysis.

References

[1] Richard Bonett, Kaushal Kafle, K. Moran, **Adwait Nadkarni**, and D. Poshyvanyk, "Discovering Vulnerabilities in Security-Focused Static Analysis Tools for Android using Systematic Mutation," in *Proceedings of the 27th USENIX Security Symposium (USENIX)*, Baltimore, MD, USA, Aug. 2018, pp. 1263–1280, ***Acceptance rate: 100/524 ≈ 19%***.

[2] Amit Seal Ami, Kaushal Kafle, K. Moran, **Adwait Nadkarni**, and D. Poshyvanyk, "Demo: Mutation-based Evaluation of Security-focused Static Analysis Tools for Android." in *Proceedings of the 43rd IEEE/ACM International Conference on Software Engineering (ICSE'21), Formal Tool Demonstration Track*, May 2021.

[3] ——, "Systematic Mutation-based Evaluation of the Soundness of Security-focused Android Static Analysis Techniques," *ACM Transactions on Privacy and Security (TOPS)*, vol. 24, no. 15, Feb. 2021.

[4] Amit Seal Ami, N. Cooper, Kaushal Kafle, K. Moran, D. Poshyvanyk, and **Adwait Nadkarni**, "Why Crypto-detectors Fail: A Systematic Evaluation of Cryptographic Misuse Detection Techniques," in *Proceedings of the IEEE Symposium on Security and Privacy (Oakland)*, Apr. 2022.

[5] Kaushal Kafle, K. Moran, Sunil Manandhar, **Adwait Nadkarni**, and D. Poshyvanyk, "A Study of Data Store-based Home Automation," in *Proceedings of the 9th ACM Conference on Data and Application Security and Privacy (CODASPY)*, Dallas, TX, USA, Mar. 2019, pp. 73–84, ***Acceptance rate: 28/119 ≈ 23.5%, [Best Paper Award]***.

[6] Kafle, Kaushal, K. Moran, Sunil Manandhar, **Adwait Nadkarni**, and D. Poshyvanyk, "Security in Centralized Data Store-based Home Automation Platforms: A Systematic Analysis of Nest and Hue," *ACM Transactions on Cyber-Physical Systems (TCPS)*, vol. 5, no. 1, Dec. 2020.

[7] Sunil Manandhar, K. Moran, Kaushal Kafle, Ruhao Tang, D. Poshyvanyk, and **Adwait Nadkarni**, "Towards a Natural Perspective of Smart Homes for Practical Security and Safety Analyses." in *The IEEE Symposium on Security & Privacy*, 2020.

[8] B. Andow, **Adwait Nadkarni**, B. Bassett, W. Enck, and T. Xie, "A Study of Grayware on Google Play," in *Proceedings of the IEEE Mobile Security Technologies workshop (MoST)*, San Jose, CA, USA, May 2016, pp. 224–233, ***Acceptance rate: 10/35 ≈ 28.6%***.

[9] S. A. Gorski III, B. Andow, **Adwait Nadkarni**, Sunil Manandhar, W. Enck, E. Bodden, and A. Bartel, "ACMiner: Extraction and Analysis of Authorization Checks in Androids Middleware," in *Proceedings of the 9th ACM Conference on Data and Application Security and Privacy (CODASPY)*, Dallas, TX, USA, Mar. 2019, pp. 25–36, ***Acceptance rate: 28/119 ≈ 23.5%***.

[10] Sunil Manandhar, Kaushal Kafle, B. Andow, K. Singh, and **Adwait Nadkarni**, "Smart Home Privacy Policies Demystified: A Study of Availability, Content, and Coverage," in *The USENIX Security Symposium (USENIX)*, Aug. 2022.

[11] **Adwait Nadkarni** and W. Enck, "Preventing accidental data disclosure in modern operating systems," in *Proceedings of the 2013 ACM Conference on Computer & Communications Security (CCS)*, Berlin, Germany, Nov. 2013, pp. 1029–1042, ***Acceptance rate: 105/530 ≈ 19.8%***.

[12] **Adwait Nadkarni**, B. Andow, W. Enck, and S. Jha, "Practical DIFC Enforcement on Android," in *Proceedings of the 25th USENIX Security Symposium (USENIX)*, Austin, TX, USA, Aug. 2016, pp. 1119–1136, ***Acceptance rate: 72/463 ≈ 15.6%***.

[13] **Adwait Nadkarni**, V. Tendulkar, and W. Enck, "NativeWrap: Ad Hoc Smartphone Application Creation for End Users," in *Proceedings of the 7th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, Oxford, UK, Jul. 2014, pp. 13–24, ***Acceptance rate: 25/96 ≈ 26%***.

[14] Stephan Heuser*, **Adwait Nadkarni***, W. Enck, and A.-R. Sadeghi, "ASM: A Programmable Interface for Extending Android Security," in *Proceedings of the 23rd USENIX Security Symposium (USENIX)*, San Diego, CA, USA, Aug. 2014, pp. 1005–1019, ***Co-first Authors. *Acceptance rate: 67/350 ≈ 19.1%*.

[15] R. Shu, P. Wang, S. A. Gorski III, B. Andow, **Adwait Nadkarni**, L. Deshotels, J. Gionta, W. Enck, and X. Gu, "A Study of Security Isolation Techniques," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, Oct. 2016.

[16] **Adwait Nadkarni**, A. Verma, V. Tendulkar, and W. Enck, "Reliable Ad Hoc Smartphone Application Creation for End Users," in *Intrusion Detection and Prevention for Mobile Ecosystems*. CRC Press, July 2017, editor: George Kambourakis and Asaf Shabtai and Konstantinos Kolias and Dimitrios Damopoulos. [Online]. Available: https://www.crcpress.com/Intrusion-Detection-and-Prevention-for-Mobile-Ecosystems/Kambourakis-Shabtai-Kolias-Damopoulos/p/book/9781138033573

[17] **Adwait Nadkarni**, A. Sheth, U. Weinsberg, N. Taft, and W. Enck, "GraphAudit: Privacy Auditing for Massive Graph Mining," North Carolina State University, Department of Computer Science, Raleigh, NC, Technical Report TR-2014-10, Aug. 2014.

[18] **Adwait Nadkarni**, W. Enck, S. Jha, and J. Staddon, "Policy by Example: An Approach for Security Policy Specification," arXiv preprint arXiv:1707.03967, Jul. 2017.

[19] Kaushal Kafle, K. Jagtap, M. Ahmed-Rengers, T. Jaeger, and **Adwait Nadkarni**, "REDACTED: IN SUBMISSION," in *The 18th ACM ASIA Conference on Computer and Communications Security (AsiaCCS)*, 2023, ***In submission***.

[20] X. Jin*, Sunil Manandhar*, Kaushal Kafle, Z. Lin, and **Adwait Nadkarni**, "Understanding IoT Security from a Market-Scale Perspective," in *Proceedings of the 29th ACM Conference on Computer and Communications Security (CCS)*, Los Angeles, CA, USA, Nov. 2022, *Co-first Authors., To appear.