Research Statement | Earlence Fernandes

I am excited by problems that span disciplines, and my research contributes cross-disciplinary insights and techniques. I work in computer security and privacy because of its many problems that inherently possess this cross-disciplinary flavor. This is perhaps best represented in the impending convergence of physical and digital systems (i.e., Internet of Things or IoT/Cyber-Physical Systems or CPS). To paraphrase Bruce Schneier, a noted security expert, "your fridge is a computer that keeps food cold, your car is a computer with wheels and an engine." Everyday objects are being enhanced with computational and networking capabilities, with the goal of driving economic efficiencies, improved energy utilization, enhanced healthcare, better security, and more convenience. From a computer security and privacy perspective, cyber-physical systems have the potential to introduce new threat models, and the related threats. From a cyber-physical systems perspective, computer security has the potential to introduce secure designs from the ground up. Exploring such bidirectional problems represents my general approach to research, and currently, my work focuses on the intersection of computer security and cyber-physical systems.

In particular, my work has explored new threat models and the related threats that cyber-physical systems bring to computer security and privacy research. Examples of the threats that my work has introduced include the first smart home malware for Samsung SmartThings [10], FM radio-based malware distribution [9], privacy threats in entity recognition systems that use IoT data [14], [15], physical adversarial examples for future self-driving cars [4], and platform compromise in trigger-action platforms [13]. I've structured my approach to discovering these threat models and threats based on a system builder's view of computing: at the lowest layer, we have devices and hardware systems, the next layer contains network communication protocols, followed by the platform layer that unifies heterogeneous devices and protocols, and finally the application layer that provides the promised benefits of the IoT: smart homes, cities, autonomous vehicles, etc.

On the flip side, I've invented computer security techniques that are helpful in the secure design of cyber-physical systems. Per my structured approach, these defenses have focused on the applications and platforms layer of the IoT computing stack. For example, FlowFence [11] and Heimdall [17] shows how developers can write secure IoT apps that respect a user's policies on data use. Appstract [15] shows how entity recognition can be performed in a more privacy respecting way through algorithmic improvements. Decentralized Trigger-Action Platform [13] shows a clean-slate design for emerging IoT trigger-action platforms that provides strong integrity guarantees in the presence of platform compromise.

Looking forward, my plan is to further explore these bidirectional problems and solutions. For example, at the applications later, I'm interested in investigating better end-user programming mechanisms for IoT, and defenses against physical adversarial examples for autonomous systems. At the platforms layer, I'm interested in exploring multi-user authentication. At the hardware layer, I'm interested in exploring fail-safe designs.

Security is a constantly evolving field, and I've often found that reinventing my research focus often helps me identify and address important emerging topics. For example, in the past I've worked on Android security problems that straddle the boundary between computer security and embedded operating systems. A cross-cutting element of my approach to past and current inter-disciplinary problems has been to independently seek out collaborators in related fields. For example, I helped bootstrap a collaboration between my group while I was at Michigan (Prof. Atul Prakash and Amir Rahmati) and UIUC (Prof. Darko Marinov and Alex Gyori) that led to a recently funded NSF proposal (CPS-1646392, CPS-1646305, $800K) on IoT security testing.

Impact. The empirical security analysis of smart home applications received the *Distinguished Practical Paper Award* at IEEE Security and Privacy in 2016 along with widespread press coverage [1]. Samsung has responded positively to our results and is currently implementing changes to its architecture based on recommendations from our study. The Appstract system has resulted in two patents for Microsoft.

IoT Platform Layer Security and Privacy

The IoT is very heterogeneous, and current work in studying the security problems of these systems has looked at individual devices or range-limited protocols in isolation. However, per the structured computing stack discussed above, there are platforms that unify these heterogeneous devices and protocols into a uniform layer. A compromise of such platforms can lead to long-range and large-scale threats, often with physical manifestation. Therefore, I've led several studies of IoT platforms with the goal of determining the appropriate threat models and related threats that such platforms pose, and how might we begin to secure these emerging platforms.

First Security Analysis of a Smart Home Platform [10]. I analyzed the security of Samsung SmartThings, a mature platform with wide support for devices and third party apps. It shares core design principles (privilege separation through capabilities, trigger-action programming through events) with other such platforms that are currently in nascent design stages. Therefore, the lessons we extracted from analyzing SmartThings will help inform the design of other systems of this kind. This work used black-box fuzzing techniques and custom-built static program analysis tools to determine that SmartThings and its apps do not adhere to the tried-and-tested security principles of least-privilege, sensitive data protection, and access control. The analysis revealed that SmartThings apps are *automatically* overprivileged (i.e., apps have access to device operations that they do not need for their functionality). Based on the discovered design flaws, I built long-range attacks that reprogrammed door locks and the first smart home malware that snooped on pincodes [10], [12]. This work provides example attacks for a threat model where attackers with software access to a home can eventually gain physical access, and cause physical damage. In traditional computer security problems, software access does not always imply the ability to cause physical damage.

This work received the Distinguished Practical Paper Award at IEEE Security and Privacy 2016, along with widespread press coverage. SmartThings is creating design changes to mitigate the automatic overprivilege in the capability system, inspired by recommendations in the paper.

Information Flow Control for Smart Home Platforms [11]. Based on the above analysis of a popular, representative smart home platform, a lesson is that even if an IoT platform is least-privileged, malicious apps can still misuse permissions in ways that are inconsistent with user expectations. This lesson is not unique to IoT, and has been observed in other types of computing systems such as mobile, desktop, and cloud. However, a difference is that the nature of IoT data changes the type of threat—The IoT has the ability to generate extremely sensitive data pertaining to people's activities, family members etc. This level of personal information collection is a tantalizing target for attackers. Thus, it is even more imperative that a principled defense against such kinds of apps exist. Towards addressing this problem, I and my co-oauthors designed and built FlowFence [3], [11], [18], a system where information flow control is a first class primitive.

Although information flow control is not a new concept, making it work in the IoT platform setting requires overcoming IoT-specific challenges. First, labels for sources and sinks must serve a functional purpose beyond representing sensitivity. Through my prototyping experience, I discovered that a label would have to be abstract at policy-writing time, because before an app is installed, the flow policy cannot refer to concrete devices. However, at runtime, these same labels should be concrete, and should refer to physical devices uniquely. The challenge is to reconcile these different label behaviors in a single system. Second, IoT systems are heterogeneous—a security solution for one system would not necessarily translate in a straightforward manner to another system. Therefore, FlowFence builds on a small set of ubiquitous security mechanisms: process isolation, and secure IPC. Arguably, these two primitives are widely available on most operating systems, including those used in IoT platforms. Using these security primitives, it is relatively straightforward to construct a FlowFence-like system. Third, IoT app developers favor rapid development. Learning a new security-oriented language will likely be a barrier to adoption. To this end, FlowFence does not introduce new language extensions. Rather, it introduces a set of APIs and app-structuring tools that leverage existing developer IDEs, debuggers, and deployment solutions.

Analyzing and Fixing Security Issues in Trigger-Action IoT Platforms [13]. SmartThings app development requires programming knowledge, and therefore, home occupants might not be able to create their own automations. *End-user* programming is a popular paradigm in the personal IoT domain, and there is a vibrant ecosystem (e.g., Microsoft Flow, Stringify, Zapier) of cloud-based platforms allowing home occupants to program automation rules on their own using simple user interfaces. These rules take the form if-trigger-then-action. For example, "If smoke is detected, then turn off my oven." To deepen my analysis of risks from IoT platforms, I conducted an empirical analysis of If-This-Then-That (IFTTT), a mature and popular trigger-action platform with over 11 million users in total and approximately 1 billion rule executions per month.

The main finding from this analysis is that if trigger-action platforms like IFTTT are compromised, then OAuth tokens for a large number of users are exposed, allowing attackers to arbitrarily manipulate physical and digital resources of users. Trigger-action platforms face this risk because of their logically-monolithic architecture. Indeed, even well-designed and tested cloud services are not immune to persistent and sophisticated threats. Prominent examples of recent attacks include Equifax, Google Docs OAuth phishing, Dropbox, and the US voters database. Beyond this threat, the analysis also revealed that IFTTT's OAuth tokens are overprivileged, permitting the bearer to perform actions beyond the abilities of IFTTT itself. For example, the work shows how an attacker an steal OAuth tokens and reprogram firmware on an IoT chip using a single HTTP call. This overprivilege only exacerbates the consequences of a platform compromise.

To defend against such attacks, this work switches to a more reasonable threat model by assuming that a trigger-action cloud platform is untrusted, and can be compromised. Under this threat model, I introduced *Decentralized Action Integrity* [13]. This security principle ensures that an attacker who controls a compromised trigger-action platform: (1) can only invoke actions and triggers needed for the rules that users have created; (2) can invoke actions only if it can prove to an action service that the corresponding trigger occurred in the past within a reasonable amount of time; and (3) cannot tamper with any trigger data passing through it undetected.

IoT Application Layer Security and Privacy

Building on top of the platform layer, the application layer enables the promised benefits of the IoT. However, these new applications might pose security and privacy risks. In this section, I will summarize my work in securing the application layer, with the goal of pointing out the bidirectional problems at the intersection of IoT and computer security.

Physical Adversarial Examples in Autonomous Systems [4]. Deep Neural Networks (DNNs) have achieved state-of-the-art, and sometimes human-competitive, performance on many computer vision tasks. Based on these successes, they are increasingly being used as part of control pipelines in physical systems such as cars, UAVs, and robots. Recent work, however, has demonstrated that DNNs are vulnerable to adversarial perturbations. These carefully crafted modifications to the (e.g., visual) input of DNNs can cause the systems they control to misbehave in unexpected and potentially dangerous ways. As part of a larger team of collaborators in security and machine learning, I've recently started exploring the physical effectiveness of adversarial perturbations. This is an example of a new threat model that my work has explored. Current work in adversarial machine learning has mostly assumed digital access to input vectors. However, for cyber-physical systems like cars, having digital access is a strong attacker assumption. A more likely threat is that an attacker can manipulate the physical world perceived by the car. This work recognizes the gap in the threat models used by current work, and introduces a more reasonable threat model for vision in autonomous systems.

Our initial results indicate that DNN-based classifiers can be effectively tricked in the physical world [4]. Figure 1 shows an example attack, where a physical Stop sign is modified with simple black and white stickers, causing a DNN-based road sign classifier to output the label 'Speed Limit 45' instead of the expected label 'Stop.'

Analyzing Data in a Privacy-Respecting Way [14], [15]. One of the main IoT benefits is that it produces data at a scale that has not been achieved before. This data is very fine-grained, and can range from how users interact with smartphone apps that control devices, to actual logs of human activity within a home. Understanding the semantics of this data can enable new experiences. For example, if a user prefers a certain thermostat temperature setting when at home, a system that can learn this fact will be able to dynamically and automatically set the temperature when the user is at a different location (e.g., hotel room). The critical piece in enabling such features is to understand the semantics of data.

We are seeing an emergence of systems that offer data semantics interpretation services (e.g., Google Now-on-Tap and Bing Snapp). However, these systems transmit all data to third-party cloud back-ends. Often, this data could be privacy sensitive. Therefore, current systems do not help improve or maintain user privacy. I designed and built Appstract [14], [15], a system and a set of algorithms that efficiently and accurately extracts the semantics of text without transmitting that text to a cloud server. This is challenging because: (1) individual data items provide very little context and (2) understanding the semantics of text poses prohibitive computational and storage overheads. A key insight is to split the analysis into a user-agnostic cloud-phase and a user-specific device-phase.

Enabling IoT Data and Semantics Usage in a Privacy-Respecting Way [17]. Related to the notion of extracting semantics from data, such as that produced by the IoT and mobile systems, Heimdall is a system that enables accountable use of that data, particularly in the context of recommendation systems. Consider an app that uses power state information of devices around a home to generate energy saving recommendations. Such an app provides useful benefits, but also has the potential to perform privacy attacks by recording device power activity. Heimdall enables constructing such an app by limiting it to only collect and transmit information of user-specified high wattage devices. Although such guarantees could be provided by flow control systems like FlowFence, the key innovation with Heimdall is simplifying the type system to include a single privacy-oriented type, therefore making it easy for developers to adopt the system.

Contextual Integrity for Users [6]. This work incorporates Nissenbaum's property of contextual integrity into IoT applications. By using a combination of static and dynamic program analyses as proxies for context, the ContexIoT framework ensures that apps only perform actions in contexts the system has seen before. Any new contexts require user approval. For example, consider an app that opens the windows when the internal temperature is greater than a certain value. If an attacker manipulates the app into opening windows, when say, people are sleeping, the context under which the window opening operation occurs has changed. ContexIoT detects this change, and will prompt users to authorize the action.

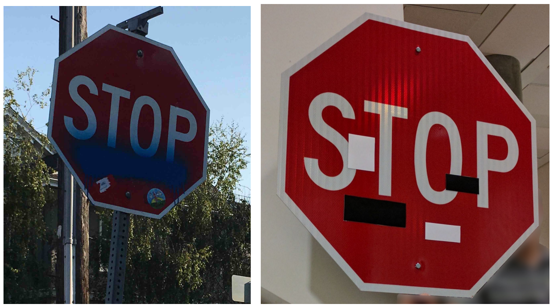

Fig. 1: The left image shows graffiti on a Stop sign in a city, a relatively common occurrence that most humans would not think is suspicious. The right image shows our example physical perturbations applied to a real Stop sign. We design our perturbations to mimic graffiti, and thus "hide in the human psyche." The right-side image is interpreted as Speed Limit 45 by DNN-based classifiers.

Past Work in Android Security

My past research has explored UI phishing defenses [8], systems that help apps reduce their trusted computing base [7], FM radio based attacks on phones [9], early techniques for contextual access control on smartphones [2], and mechanisms for bring-your-own-device (BYOD) solutions [19]–[21]. Mobile systems, in many ways, have been a precursor to the IoT phenomenon, and thus, many of these projects have helped me identify and address important emerging problems in IoT.

Looking Forward

My future research plans are to deepen my investigation into problems that cross disciplines. As before, these projects are structured around the system builder's view of the IoT computing stack: application layer, platform layer, network layer, and device layer.

Fail-Safe Designs. Traditional security research has investigated designs that protect a system from malicious actors, rather than designs that take steps to protect users when there is malicious activity or a breakdown of a system's core security mechanisms. This is reasonable because the threats to an individual from contemporary digital security breaches are virtual. In the cyber-physical setting, security breaches can result in physical and material harm to users. Fail-safe designs provide a degree of safety in the particular case of a breakdown in a system's defenses. For example, a traffic light system cannot be switched into an inconsistent state independently of any type of software control because it contains physical components to prevent that. Notice that fail-safe designs are not a defense in and of themselves. Rather, they are a fallback for when security does get compromised. In the long-term, I am interested in exploring fail-safe design for modern IoT systems. A challenge is the heterogeneity and customizability of IoT devices. In contrast to traditional cyber-physical systems where the notion of safety is fixed (e.g., garage door or traffic lights), IoT devices might not have preset notions of safe operation. For example, a thermostat's safe temperature maximum and minimum is dependent on the use case. The values for temperature might vary widely between a nursing home for adults versus a critical care unit for infants in hospitals. In the long-term, I plan on exploring how software verification techniques, and trusted hardware (e.g., SGX) might be useful in building a language and system architecture for flexibly specifying fail-safe behavior in heterogeneous IoT systems.

Physical Models of Devices for Intrusion Detection. In classic network intrusion detection settings, a challenge is that computing a model for "normal" behavior of a network node is tricky because nodes are general purpose computers performing a variety of tasks. By contrast, a node in the IoT network is a relatively fixed-function device whose behavior is more predictable. In the near-term, I am interested in exploring how physics-based models of IoT devices can be useful in creating network signatures for the purpose of intrusion detection. In the long-term, I'm curious about whether such models yield insights into whether a device is functioning as per a designer's parameters. Related to this, I am also interested scaling up the use of physics-based models to entire processes, with the goal of ensuring that an entire physical process runs correctly to completion.

Tackling the Device Update Problem. Many of today's attacks occur because IoT devices are unpatched, or have security bugs due to poor software practices. The standard method of dealing with such issues in computer security is to issue software patches using well-known mechanisms. However, the IoT setting presents new challenges: (1) Devices may fundamentally not be patchable because the manufacturer has simply not built the infrastructure for it. (2) IoT devices may run in harsh environments with intermittent access to power and network connectivity (e.g., sensors in a concrete bridge that harvest power through vibrations). (3) Performing updates might require careful co-ordination because IoT devices control physical processes. Unplanned downtimes due to software updates can have disastrous consequences. I am interested in exploring solutions to these challenges that might require changing our notions of how software updates are executed (e.g., transitioning from host-based updates to network-based patching).

Improved End-User Programming for IoT. Although end-user IoT programming allows flexibility, it also allows programming errors. Even simple trigger-action rules can be difficult for end-users to get right [5], [16]. These errors can lead to security, privacy, and especially relevant for IoT, safety issues. In the near-term, I plan on exploring techniques that reduce the potential for errors to occur in end-user programs. This exploration would include human subjects research, and novel user interface (UI) designs, including notions of UIs beyond the classically popular modality of a screen. For example, can emerging technologies like augmented reality help users better visualize trigger-action rules because these rules might be better physically-situated compared to today's web-based programming interfaces? Would emerging voice-based interfaces be helpful in reducing programming errors? In the long-term, I plan on exploring whether and how lessons learned from this exploration of simple trigger-action rules transfer to end-user IoT programming in more complex (multiple triggers and actions), and varied settings (e.g., ladder logic in process control).

Multi-User Authentication. In traditionally considered settings of authentication, such as online and mobile services, authentication protects users from remote strangers. In IoT settings such as the home, authentication operates under a different threat model: (1) Devices are shared. (2) Authentication rules traverse complex social ties ranging from parent-child relationships to roommate situations and temporary home-share visitors. (3) IoT devices do not necessarily have traditional interaction modalities (such as screens). This complicates the straightforward porting of existing authentication mechanisms like passwords. (4) Devices possess access duality that complicate access rules—some can be actuated remotely though software, some can be actuated only physically, and some can be actuated through software and physical controls. Clearly, existing concepts in authentication do not cleanly transfer to the IoT setting. In the long-term, I plan on exploring these authentication issues through user studies, and system building projects.

The Intersection of Cyber-Physical Systems and Machine Learning. As discussed earlier, my current work has explored the effectiveness of physical world attacks on deep learning models. In the long-term, I plan on investigating two types of problems in deep learning when applied to cyber-physical systems: (1) Enhanced attack techniques: My current work has explored attacks on classifiers—a single component of a larger control pipeline [4]. I'm interested in exploring the extent to which physical world attacks can compromise more components of the vision pipeline of autonomous cyber-physical systems. Object detection, segmentation, and steering-wheel angle prediction are a few examples. (2) Defense techniques: The larger vision of this work is to learn insights that would help construct effective defenses. I plan on exploring defenses against adversarial perturbations using a two-pronged approach. First, how can we re-introduce humans in the loop, and construct so-called centaur systems where a human can be brought efficiently and correctly into a control loop when the machine is unsure of its next steps? Second, can we leverage techniques from control theory to the analysis and removal of adversarial perturbations? Control theory has a rich set of analyses for the stability of functions. Can instability analyses point out regions in which adversarial examples exist? Consequently, can we leverage secure controller synthesis techniques to transform an existing DNN into a version that is resistant to classes of adversarial examples?

Overall, I am broadly interested in computer security and privacy, with a special interest in cross-disciplinary problems.

REFERENCES

[1] "Media coverage of my work," https://iotsecurity.eecs.umich.edu.

[2] M. Conti, B. Crispo, **E. Fernandes**, and Y. Zhauniarovich, "CRePE: A System for Enforcing Fine-Grained Context-Related Policies on Android," *IEEE Transactions on Information Forensics and Security (TIFS)*, 2012.

[3] M. Conti, **E. Fernandes**, J. Paupore, A. Prakash, and D. Simionato, "OASIS: Operational Access Sandboxes for Information Security," in *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM@CCS)*, 2014.

[4] I. Evtimov, K. Eykholt, **E. Fernandes**, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song, "Robust Physical-World Attacks on Deep Learning Models," in *arXiv preprint 1707.08945*, 2017.

[5] J. Huang and M. Cakmak, "Supporting mental model accuracy in trigger-action programming," in *ACM International Joint Conference on Pervasive and Ubiquitous Computing (Ubicomp)*, 2015.

[6] Y. Jia, Q. A. Chen, S. Wang, A. Rahmati, **E. Fernandes**, Z. M. Mao, and A. Prakash, "ContexIoT: Towards Providing Contextual Integrity to Appified IoT Platforms," in *21st Network and Distributed Security Symposium*, 2017.

[7] **E. Fernandes**, A. Aluri, A. Crowell, and A. Prakash, "Decomposable Trust for Android Applications," in *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2015.

[8] **E. Fernandes**, Q. A. Chen, J. Paupore, G. Essl, J. A. Halderman, Z. M. Mao, and A. Prakash, "Android UI Deception Revisited: Attacks and Defenses," in *Proceedings of the 20th International Conference on Financial Cryptography and Data Security (FC)*, 2016.

[9] **E. Fernandes**, B. Crispo, and M. Conti, "FM 99.9, Radio Virus: Exploiting FM Radio Broadcasts for Malware Deployment," *IEEE Transactions on Information Forensics and Security (TIFS)*, 2013.

[10] **E. Fernandes**, J. Jung, and A. Prakash, "Security Analysis of Emerging Smart Home Applications," in *Proceedings of the 37th IEEE Symposium on Security and Privacy (S&P)*, 2016.

[11] **E. Fernandes**, J. Paupore, A. Rahmati, D. Simionato, M. Conti, and A. Prakash, "FlowFence: Practical Data Protection for Emerging IoT Application Frameworks," in *Proceedings of the 25th USENIX Security Symposium*, 2016.

[12] **E. Fernandes**, A. Rahmati, J. Jung, and A. Prakash, "The Security Implications of Permission Models in Smart Home Application Frameworks," *IEEE Security and Privacy Magazine*, 2017.

[13] **E. Fernandes**, A. Rahmati, J. Jung, and A. Prakash, "Decentralized Action Integrity for Trigger-Action IoT Platforms," in *22nd Network and Distributed Security Symposium (NDSS)*, 2018.

[14] **E. Fernandes**, O. Riva, and S. Nath, "My OS Ought to Know Me Better: In-app Behavioural Analytics as an OS Service," in *15th Workshop on Hot Topics in Operating Systems (HotOS XV)*, 2015.

[15] **E. Fernandes**, O. Riva, and S. Nath, "Appstract: On-The-Fly App Content Semantics with Better Privacy," in *Proceedings of the 22nd ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, 2016.

[16] C. Nandi and M. D. Ernst, "Automatic trigger generation for rule-based smart homes," in *Workshop on Programming Languages and Analysis for Security (PLAS)*, 2016.

[17] A. Rahmati, **E. Fernandes**, K. Eykholt, X. Chen, and A. Prakash, "Heimdall: A Privacy-Respecting Implicit Preference Collection Framework," in *15th ACM International Conference on Mobile Systems, Applications, and Services*, 2017.

[18] A. Rahmati, **E. Fernandes**, and A. Prakash, "Applying the Opacified Computation Model to Enforce Information Flow Policies in IoT Applications," in *Proceedings of the 1st IEEE CyberSecurity Development Conference (SecDev)*, 2016.

[19] G. Russello, M. Conti, B. Crispo, and **E. Fernandes**, "MOSES: Supporting Operation Modes on Smartphones," in *Proceedings of the 17th ACM Symposium on Access Control Models and Technologies (SACMAT)*, 2012.

[20] G. Russello, B. Crispo, **E. Fernandes**, and Y. Zhauniarovich, "YAASE: Yet Another Android Security Extension," in *3rd IEEE Conference on Privacy, Security, Risk and Trust (PASSAT)*, 2011.

[21] Y. Zhauniarovich, G. Russello, M. Conti, B. Crispo, and **E. Fernandes**, "MOSES: Supporting and Enforcing Security Profiles on Smartphones," *IEEE Transactions on Dependable and Secure Computing (TDSC)*, 2014.