Earlence Fernandes – Research Statement

I. Overview

I take a broad view of computer systems security. The classical interpretation focuses on the host operating system and network levels. However, computer systems *themselves* have evolved. Today, they are distributed, embedded and capable of learning. Although these evolved computer systems provide benefits in many areas including energy efficiency, productivity, convenience and safety, unless they have the correct security foundations, they amplify traditional computer security threats, giving those threats wider reach and impact and also introduce new classes of threats. My research goal is to ensure that human society will gain the benefits of these evolved computer systems without the security and privacy issues.

To achieve this vision, I use a systems security approach — (1) apply the security mindset to deeply understand the new ways in which these evolved computer systems are vulnerable; and (2) construct principled systems-level defenses that offer end-to-end guarantees. In general, I take particular inspiration from the secure design principles of computer systems, as articulated by Saltzer and Schroeder [20]. My current research agenda focuses on two themes that I believe are critical enablers of the evolved computer systems discussed above.

(1) Controlled sharing in distributed systems. Due to the commoditization of cloud computing, all of our digital data and physical devices are now available on the Internet sitting behind vendor-specific cloud services and APIs (e.g., emails, personal files, messages, health data, smart home and building devices). The main benefit is that users can share their resources with other parties to achieve useful functionality. One of the most popular systems that use controlled sharing are so-called Trigger-action platforms that evolved to support user-specified connections between data and devices. Common examples include controlling a garage door based on location and saving emails of a certain subject to online storage. Controlled sharing protocols (e.g., OAuth) make all this flexible automation possible, yet, our collective experience has shown that these protocols are insufficient from a security standpoint. Fundamentally, they cannot control *how* sensitive access is used once privilege has been granted. Indeed, many of the problems of user resource misuse today are fundamentally because there are no controls on how third-parties use their privileged access. My research goal is to create techniques for controlled sharing in distributed systems where one can obtain strong guarantees on *how* that access is used by a third party. The core insight is to enforce the classic principle of least-privilege by co-designing sharing protocols and the systems that use the shared resources.

(2) ML-powered physical systems. Machine learning has supercharged many areas of computer systems. Simultaneously, the community has established that machine learning is fundamentally vulnerable to many kinds of attacks. This problem is particularly pernicious when we use ML in safety- and security-critical systems like autonomous vehicles or authentication systems in mixed reality. My key insight is that ultimately, machine learning is a piece in a larger computational pipeline and thus, understanding and building robustness *requires* an end-to-end *systems* perspective. Building on this insight, this theme characterizes the vulnerability of ML-powered physical systems with the final goal of creating robust computer systems that use ML in their pipelines.

Impact. I believe that if there are ways to benefit society *after* publishing papers, then it is important to facilitate such positive impact. Earlier in my career, I collaborated with Samsung to fix security issues in their smart home product line. Based on that experience, ideas from my subsequent work in secure smart home platforms (e.g., FlowFence [14]) was adapted into Samsung products. I am currently exploring how defense techniques from my work on secure Internet-scale automation can be applied to products from IFTTT (one of the largest user-centered automation providers). The Stop sign that we used in our physical adversarial example work is on display at the Science Museum in London (and has been added to their permanent collection of artifacts), and helps raise awareness in the general public about emergent security issues.

I often engage with the press when society at large would benefit from awareness about my results — my work on smart home security and physical adversarial perturbations work received widespread coverage, and raised awareness among key stakeholders. For example, I have advised a member of Congress on Internet of Things security, and several US government agencies including the FTC, the National Academies, and the JASONs on security issues related to the deployment of ML systems in critical infrastructure.

II. Research Themes

I will discuss recent research results that focus on two areas: (1) Secure controlled sharing in user-centered automation platforms and (2) Secure ML-powered physical systems. The interested reader may refer to the bibliography for pointers to my work in smart home security [6]–[8], [13], [14], [17], privacy-respecting data analysis systems [15], [16], and my early work in smartphone security [4], [10]–[12], [18], [19], [22].

A. *Secure controlled sharing in user-centered automation platforms*

There is a web-based API for everything — digital resources like email, health data, and online storage; physical resources like homes, buildings, cities, and industrial equipment. End-users, who are not trained in programming, create simple trigger-compute-action automations using these web APIs with the help of so-called Trigger-Compute-Action Platforms (TAPs). Common examples of TAP programs include routing emails with certain subject lines to specific folders, locking a garage door based on a user's location or an energy company monitoring the health of field devices and then triggering alerts for maintenance. TAPs have become a critical substrate for Internet-scale automation because of their widespread compatibility with diverse digital and physical resources and their ease-of-use that allows non-programmers to create automations (e.g., the popular IFTTT platform boasts 20 million users). Unfortunately, they are built using controlled sharing protocols like OAuth, that are fundamentally vulnerable because a user cannot enforce *how* access to their resources is used — users have no choice but to take it on faith that the third-parties (i.e., automation platforms) will abide by their promises to not misuse their access.

This research theme establishes techniques for controlled sharing of a user's digital and physical resources while respecting the principle of least-privilege. Concretely, I have the singular goal of ensuring that the TAP only executes user-created rules without doing anything else, such as stealing user data or manipulating their physical and digital resources beyond what the user intends. I will achieve this guarantee under a threat model where we assume that the TAP can abuse its access to user resources. This threat model serves as a convenient proxy for a range of attackers who might compromise the TAP cloud system to various degrees. My insight is that the unique TAP environment permits innovations in practically adapting powerful machinery from trusted computing, applied cryptography and program analysis.

Design Spectrum Exploration. The concrete research agenda is to explore the different trade-offs possible between functionality, security, usability and performance and ultimately contribute a controlled sharing framework that will help future TAP designers to make informed choices when building these systems. Below, I discuss concrete systems I have built that explore these trade-offs.

Practical resource access minimization (USENIX Security 2022) [2]. Consider the automation rule from Fig. 1. Based on the OAuth controlled sharing protocol, the automation platform will obtain OAuth tokens to access user emails and a Slack channel. The security issue of this design is twofold: (1) The OAuth tokens allow access to more data than what is needed to run the rule; (2) An attacker can misuse the tokens because the security system cannot control how an entity uses those credentials. In this example, the automation platform gets access to all emails for a user from the specified sender (due to issue 1 above) and if those tokens are leaked (e.g., through a simple implementation flaw), the attacker can use these tokens however they want (due to issue 2 above).

Stepping back, the ideal solution is to have a controlled sharing system that ensures: (1) Anyone who has the authorization tokens may *only use* them in accordance with a user-specified policy; (2) The tokens are least-privileged with respect to the rule. This is a challenging problem to solve in general. First, what is the appropriate access usage policy? Simply asking users is a non-solution. Second, how can we create least-privilege tokens that are specific to computations that a user wants to run with their data and devices?

My minTAP system addresses these two issues in the context of trigger-computer-action platforms (TAPs). The core insight is that the well-defined structure and semantics of TAP automations lend themselves to automatic techniques for the two challenges above. Specifically, minTAP shows how a controlled sharing system can automatically derive an access usage policy directly from the automations that end-users create. Using the derived policy, it shows how one can utilize lightweight program analysis techniques to enforce that the TAP can only access the user's resources in accordance with the derived policy.

Going back to the example in Fig. 1, minTAP offers the following guarantee, automatically: The TAP cloud server (where the user-created automations execute) will be able to use its OAuth tokens to access an email's subject line *only* when the email subject line contains the specified keyword, despite the fact that the tokens can themselves do a lot more than that. This effectively locks down the *use* of those tokens automatically and dynamically.

Cryptographically-enforced controlled sharing (IEEE S&P 2021) [3]. The minTAP project offers one notion of controlled sharing — a third-party with a security token may gain minimal access to user data, as specified by a user-created automation rule. Although this is a step in the right direction, ultimately, the TAP cloud server does get access to plaintext data, that it could potentially misuse. Is there a stronger form of controlled sharing possible where this type of misuse is not mitigated? My encrypted TAP system explores this question by adapting tools for computing on encrypted data. The core insight is that the TAP rules are simplistic and stateless compared to general-purpose software. This allows us to tailor cryptographic techniques like Garbled circuits to gain practicality, functionality and efficiency.

```
let str = NewEmail.Subject
if (str.indexOf('Confidential') ===
↪  -1) {
  Slack.postToChannel.skip()
} else {
  Slack.postToChannel.setMessage(
  NewEmail.Subject
  + ' just received!')
}
```

Fig. 1: Example TAP rule that computes on sensitive data, connecting an email service with Slack. It has access to ALL email data from a sender, but only needs the subject line, only when it contains the characters 'Confidential'. The 'skip' function call means that the entire rule is aborted. In that case, the rule does not need access to *any* sensitive email data.

Concretely, a trigger-action rule involves several operations on the trigger service data, and the computation follows a pre-defined path of: (1) sensitive data coming from the trigger service; (2) the TAP computing on that data to produce a result; (3) the action service performing tasks based on that output. Furthermore, the computation is unidirectional proceeding from the trigger service to the TAP to the action service without multiple rounds of communication. The trigger and action services do not have any dependency on each other — they are independent services. Thus, the TAP acts as a function evaluator that knows how to interact with the trigger and action services. My approach designates the TAP cloud service as a malicious function evaluator and uses garbled circuits (GCs) as a primitive to execute functions on encrypted trigger data. This will guarantee confidentiality of user data and integrity of the rule through the properties of garbled circuits as formalized by Bellare et al. [1].

To achieve this design, my work addresses the following open problems: (1) Although from a theoretical perspective, any computable function can be evaluated using GCs, performance matters. Therefore, what types of operations exist in trigger-compute-action rules and can we efficiently execute them using appropriate circuit representations? What innovative circuits do we need for the TAP domain? (2) The TAP is fully malicious implying that we must use malicious-secure protocols; however, they can become expensive and introduce multi-round communications between parties which drastically changes the TAP computing paradigm. Is there a design point where we can mitigate these issues while supporting real-world trigger-action rules?

At a high-level, this project found that indeed, it is possible to run TAP rules using garbled circuits in the presence of a malicious function evaluator using *only a semi-honest implementation of GCs*. This is possible because the TAP environment affords a trusted circuit generator and a trusted party that operates on the results of GC evaluation.

Current and future directions. Based on my experience with the above two projects, I have the following insights: (1) Trigger-action programming systems offer a lot of structure, allowing us to tailor strong security guarantees with good trade-offs in performance and functionality; (2) The deeper underlying problem is that bearer credentials (e.g., OAuth), the cornerstone of cloud-based automation, are fundamentally inadequate as a security enforcement mechanism. It only enforces *what* is accessed and not *how* the accessed data and devices are used. Thus, my current and future research agenda asks the question — How can we create decentralized credentials that govern how the resources they protect are used? Indeed, one can view the encrypted TAP and minTAP systems as point-solutions towards this problem. The common issue with them is that they are limited in the expressivity of the computation that governs how sensitive data and devices are being used. I am currently investigating how tailored trusting computing primitives might offer a solution that is both, expressive in computation and low in performance overhead. Conceptually, my goal is to create a system where a security credential can only be used in a trusted enclave. A key contribution, I expect, is that we can use a minimal set of trusted computing hardware primitives and a very small trusted computing base because TAP rules are well-structured and simplistic compared to general-purpose software (see insights above).

B. Securing ML-powered Physical Systems

AI has super-charged many areas of computer systems, including those that interact with the physical world. For example, self-driving cars use machine learning for perception and authentication systems in mixed reality devices use it to recognize biometrics. This theme's goal is to understand end-to-end robustness properties of computer systems when they use ML as part of the pipeline. Only then, will we be able to construct appropriate systems-oriented defenses. My current work focuses on the canonical ML-powered physical system: autonomous vehicles. In the future, I plan to expand the investigation to include systems like biometric authentication in augmented reality.

Realistic threat models (CVPR 2018 & 2021, AAAI 2021). The most realistic way to compromise an ML-powered autonomous vehicle is to manipulate its environment. When it senses this corrupted environment, its behavior will change from ideal and the attacker will achieve their goals. My early work showed the first example of how an attacker can throw a few stickers on a Stop sign to trick traffic sign classifiers [5]. However, this opens up several questions, such as: (1) What is the space of physical adversarial examples that are *effective* and *realistic*? (2) What are the end-to-end effects of attacks on machine learning models in a control pipeline?

For the first question on examining the attack space, my goal is to examine the differences between human and machine sensing and understand how attackers can exploit these differences. By establishing these new attack classes, we can begin to think about defenses. As preliminary work, we have recently demonstrated the first *invisible* physical attack that only manipulates the light shining on victim objects rather than the object's underlying texture [21]. This work exploits the fact that humans cannot perceive flickering light above 80 Hz. Fig. 2 shows an example. I contend that these types of attacks are more realistic and practical than current techniques (e.g., attacker flying a drone in front of a victim car or aiming lasers from the side of the road at moving objects with high precision) because the attacker only has to hack a smart LED bulb (e.g., in a tunnel) to achieve their goals.

For the second question on end-to-end effects, my goal is to determine how ML-based attacks affect the larger systems that use ML as part of their processing pipelines. This will help determine what parts of the pipelines are vulnerable and what kinds of defenses need to be in place to avoid the security issues. As preliminary work, we have recently contributed a sequential attack algorithm that can manipulate the behavior of Forward Collision Warning (FCW) systems — a common driver assistance feature available in most cars today. Using techniques from control theory, we demonstrate how an attacker can only compromise vision-related measurements to completely hijack FCW behavior [9]. For example, an attacker can force FCW to show no collision danger when in fact, there is an imminent collision.

With Attack Signal Without Attack Signal

Fig. 2: Example of an invisible physical attack: Images as seen by a human (without border) and as captured by a camera (in black border) with the attack signal (left two images) and without (right two images). The image without the attack signal is classified as coffee mug, while the image with the attack signal is classified as perfume. The attack is robust to camera orientation, distance, and ambient lighting.

Unfortunately, we discovered that attacking an end-to-end system such as FCW requires us to make threat model assumptions that would limit the attacker's ability to target multiple vehicles at scale, because they have to tailor an attack to a specific vehicle (concurrent work in the community has come to a similar conclusion through their own explorations). This drastically limits the applicability of the attack — for example, an attack that is only successful when a number of factors align is not a very widely effective attack and insufficiently informs future defenses. My hope is that the results on practical physical attacks will help us understand more realistic end-to-end attacks.

Current and future directions. Motivated by the above results, I am pushing our understanding of practical physical attacks along two directions: (1) Investigate other types of physical attacks that rely on differences between human and machine sensing. For example, global shutter cameras are vulnerable to electromagnetic interference — could an attack direct specially-crafted RF signals and manipulate what the camera is seeing? (2) Apply practical physical attacks to end-to-end self-driving pipelines to understand global robustness properties. The second direction will help focus the defense efforts of the community. Longer term, I plan on exploring how ML security issues affect biometric authentication that is used in augmented reality systems, as this is another crucial and emerging area where ML is an integral part of the larger computer system. I have recently received funding from Facebook on related issues in authentication for mixed reality.

REFERENCES

[1] M. Bellare, V. T. Hoang, and P. Rogaway, "Foundations of garbled circuits," in *Proceedings of the 2012 ACM conference on Computer and communications security*, 2012, pp. 784–796.

[2] Y. Chen, M. Alhanahnah, A. Sabelfeld, R. Chatterjee, and **E. Fernandes**, "Practical Data Access Minimization in Trigger-Action Systems," in *Proceedings of the 31st USENIX Security Symposium*, 2022.

[3] Y. Chen, A. Chowdhury, R. Wang, A. Sabelfeld, R. Chatterjee, and **E. Fernandes**, "Data Privacy in Trigger-Action Systems," in *Proceedings of the 42nd IEEE Symposium on Security and Privacy (S&P)*, 2021.

[4] 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.

[5] K. Eykholt, I. Evtimov, **E. Fernandes**, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust Physical-World Attacks on Deep Learning Visual Classification," in *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[6] W. He, M. Golla, R. Padhi, J. Ofek, M. Dürmuth, **E. Fernandes**, and B. Ur, "Rethinking access control and authentication for the home internet of things (iot)," in *27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, 2018, pp. 255–272. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/he

[7] W. He, V. Zhao, O. Morkved, S. Siddiqui, **E. Fernandes**, J. Hester, and B. Ur, "Sok: Context sensing for access control in the adversarial home iot," in *2021 IEEE European Symposium on Security and Privacy (EuroS P)*, 2021, pp. 37–53.

[8] 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.

[9] Y. Ma, J. Sharp, R. Wang, **E. Fernandes**, and X. Zhu, "Sequential Attacks on Kalman Filter-based Forward Collision Warning Systems," in *35th AAAI Conference on Artificial Intelligence (AAAI)*, Feb. 2021.

[10] **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.

[11] **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.

[12] **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.

[13] **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.

[14] **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.

[15] **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.

[16] **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.

[17] A. Rahmati, **E. Fernandes**, K. Eykholt, and A. Prakash, "Tyche: A risk-based permission model for smart homes," in *Proceedings of the 3rd IEEE CyberSecurity Development Conference (SecDev)*, 2018.

[18] 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.

[19] 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.

[20] J. Saltzer and M. Schroeder, "The protection of information in computer systems," *Proceedings of the IEEE*, vol. 63, no. 9, pp. 1278–1308, 1975.

[21] A. Sayles, A. Hooda, M. Gupta, R. Chatterjee, and **E. Fernandes**, "Invisible perturbations: Physical adversarial examples exploiting the rolling shutter effect," in *In proceedings of CVPR 2021*.

[22] 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.